Application Platform Suite Software Vendors' Strategies
in Standards Driven Industry Networks

by

Boon Chung Phua
B.E., Tokyo Institute of Technology, 1994

Submitted to the Alfred P. Sloan School of Management
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in the Management of Technology
at the Massachusetts Institute of Technology

June 2004

© 2004 Boon Chung Phua. All Rights Reserved
The author hereby grants to MIT permission to reproduce and
distribute copies of this thesis document in whole or in part.

Signature of author

________________________________________________________
Sloan School of Management
May 7, 2004

Certified by

________________________________________________________
Henry Birdseye Weil
Senior Lecturer, Sloan School of Management
Thesis Supervisor

Accepted by

________________________________________________________
David Weber
Director, MOT Program
Sloan School of Management
Application Platform Suite Software Vendors’ Strategies

In Standards Driven Industry Networks

by

Boon Chung Phua

Submitted to the Alfred P. Sloan School of Management on May 7, 2004 in Partial Fulfillment of the Requirements for the Degree of Masters of Science in the Management of Technology

ABSTRACT

The emergence of industry standards often has disruptive impacts on the behavior of markets. It can drive commoditization, substitution and convergence. It also changes the industry structure creating new business models and value chains, coupled with the entrance of new firms and the exit of incumbents still entrenched in proprietary technology. Studies have been conducted on the impact of industry standards in the personal computer and microprocessor industries which resulted in the disintegration of the industries into a horizontally organized cluster of hundreds of firms, and the emergence of value-capture powerhouses within the industry.

This thesis studies the shifts in industry structure in the field of Application Platform Suite (APS) software resulting from the emergence of Java programming language as the industry standard for non-Microsoft-based application development. APS software runs and manages the critical e-business applications managing the interactions between users and the enterprise backend IT systems. BEA Systems and IBM are the two dominant Java-based APS value-capture powerhouses within this industry.

This thesis surveys and summarizes the existing body of research on industry dynamics, growth strategy, technology strategy and competitive strategy, applies it to the field of APS software industry to predict the dynamics of the value chains and propose future strategies in this industry. It predicts that (1) Success of Microsoft’s entry into enterprise APS market is low to medium; (2) Success of IBM in creating a unified development platform is medium to low; and (3) Success of Microsoft’s entry into SAP’s core packaged business application market is medium to low. Building on these forecasts, the thesis proposes the following strategies: (1) Microsoft should fundamentally change its business and technology strategy to shift the likely outcome towards its favor; (2) SAP should consider progressively outsourcing its back-end technology components to focus on the applications business, and BEA and IBM should consider positioning themselves as the preferred choice of the IT industry’s back-end system provider; (3) Microsoft and BEA should invest in small open-source experimentation to understand the open-source dynamics; and (4) BEA should plan for the contingency of being an acquisition target with the primary aim of sustaining its fundamental position of remaining as an independent and pure-play infrastructure software vendor.

Thesis Supervisor: Henry Birdseye Weil
Title: Senior Lecturer of Sloan School of Management
ACKNOWLEDGEMENTS

I would like to express sincere gratitude to my thesis supervisor, Dr. Henry Birdseye Weil, for his helpful and ceaseless advice.

I am extremely grateful to DSTA for providing me this once-in-a-lifetime opportunity to study at MIT. I am especially grateful to Mr. Leonard Nee for his support and encouragement throughout my study at the Management of Technology Program of MIT.

Last but not least, I would like to thank my wife, Kar Ming, and my daughters, Wan Cheng and Suet Cheng. My days in Cambridge could never be so substantial and memorable without them.
# Table of Contents

1. Introduction .................................................................................................................. 8

2. Structure and Method ................................................................................................. 11

3. Key Industry Players .................................................................................................. 12
   3.1. Introduction ........................................................................................................... 12
   3.2. IBM ....................................................................................................................... 12
   3.3. BEA ....................................................................................................................... 13
   3.4. Microsoft ............................................................................................................. 14
   3.5. SAP ....................................................................................................................... 16
   3.6. JBOSS .................................................................................................................. 17

4. Trends in the APS Software Industry ......................................................................... 19
   4.1. Introduction ......................................................................................................... 19
   4.2. Business Trends .................................................................................................. 19
      4.2.1. Stages of IT Architecture and the impact on the locus of control of IT Architecture and Standards ................................................................................. 19
      4.2.2. Increased adoption of open-source software ..................................................... 26
      4.2.3. Antitrust.......................................................................................................... 36
   4.3. Technological Trends ............................................................................................. 42
      4.3.1. Interoperability through Web Services ............................................................ 42
      4.3.2. Service Oriented Architecture ........................................................................ 45

5. Analytical Frameworks ............................................................................................... 50
   5.1. Introduction ......................................................................................................... 50
   5.2. Technology Acquisition Strategy .......................................................................... 50
   5.3. Multiple Dimensions of Acquisition Implementation ........................................... 52
   5.4. Ecosystem Driven Growth Strategy through Platform Leadership .................. 55
   5.5. Dynamics of Industry - Evolution and disintegration and reintegration of industry ........ 60
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6. Closed and Open Innovation Paradigm</td>
<td>63</td>
</tr>
<tr>
<td>5.7. Product Platform and the Architecture of Software Products</td>
<td>65</td>
</tr>
<tr>
<td>5.8. Open source innovation</td>
<td>70</td>
</tr>
<tr>
<td>5.9. Technology Diffusion – Crossing the Chasm</td>
<td>74</td>
</tr>
<tr>
<td>5.10. Value Chain Dynamics</td>
<td>79</td>
</tr>
<tr>
<td>5.11. The Innovator’s Dilemma</td>
<td>91</td>
</tr>
<tr>
<td>5.12. Scenario Analysis</td>
<td>94</td>
</tr>
<tr>
<td>6. Analysis – IBM versus BEA</td>
<td>97</td>
</tr>
<tr>
<td>6.1. Introduction</td>
<td>97</td>
</tr>
<tr>
<td>6.2. IBM</td>
<td>97</td>
</tr>
<tr>
<td>6.2.1. Corporate and Business Strategy</td>
<td>97</td>
</tr>
<tr>
<td>6.2.2. Technology Strategy</td>
<td>100</td>
</tr>
<tr>
<td>6.2.3. Analysis</td>
<td>101</td>
</tr>
<tr>
<td>6.2.4. Summary</td>
<td>105</td>
</tr>
<tr>
<td>6.3. BEA</td>
<td>105</td>
</tr>
<tr>
<td>6.3.1. Corporate and Business Strategy</td>
<td>105</td>
</tr>
<tr>
<td>6.3.2. Technology Strategy</td>
<td>109</td>
</tr>
<tr>
<td>6.3.3. Analysis</td>
<td>110</td>
</tr>
<tr>
<td>6.3.4. Summary</td>
<td>116</td>
</tr>
<tr>
<td>6.4. Summary</td>
<td>117</td>
</tr>
<tr>
<td>7. Analysis – Microsoft versus BEA</td>
<td>121</td>
</tr>
<tr>
<td>7.1. Introduction</td>
<td>121</td>
</tr>
<tr>
<td>7.2. Microsoft</td>
<td>121</td>
</tr>
<tr>
<td>7.2.1. Corporate and Business Strategy</td>
<td>121</td>
</tr>
<tr>
<td>7.2.2. Growth Strategy</td>
<td>123</td>
</tr>
<tr>
<td>7.2.3. Technology Strategy</td>
<td>125</td>
</tr>
</tbody>
</table>
8.4.5. Use of open-source as a competitive tool to disrupt the market ........................................... 172
8.4.6. Value chain dynamics ........................................................................................................... 174

   9.1. Introduction .......................................................................................................................... 177
   9.2. Strategic Uncertainties ....................................................................................................... 177
   9.3. Key Drivers Identified ......................................................................................................... 179
   9.4. Scenarios Identified ............................................................................................................ 182
   9.5. Scenarios Explored ............................................................................................................ 184
   9.6. Proposed Strategies for APS Software Providers ............................................................... 188

10. Conclusion ............................................................................................................................... 193

11. Appendices ............................................................................................................................. 203
    Appendix A – Acquisitions of IBM ......................................................................................... 203
    Appendix B – Acquisitions of BEA ....................................................................................... 206

12. References ............................................................................................................................... 211

13. Bibliography ............................................................................................................................ 213
1. Introduction

BEA Systems (BEA) is the company that popularized the “Application Server”, a piece of application integration and middleware\(^1\) software critical to managing the web sites for companies that conduct business online. Founded in 1995 by Bill Coleman, Ed Scott and Alfred Chuang (their first initials make up BEA), they made their first major acquisition in 1998 of a firm called WebLogic, which sold an early version of the Java application server. WebLogic technology became the clear leader in Java application servers by 1999. BEA's revenues tripled between mid-1999 and mid-2001, making it the fastest software firm ever to reach $1 billion in revenues with a customer base of more than 13,000 enterprises around the world.

However, by the end of 2001, BEA's strength of focusing on the application server started to become its liability. BEA's customers' projects grew beyond the scope of the application server model. BEA users were buying their portal technology, integration technology, systems management technology and development tools elsewhere. Meanwhile, the maturing Java 2 Platform, Enterprise Edition (J2EE) standard and improving vendor execution enabled a number of vendors to offer competitive, alternative J2EE application servers. Most notably, IBM's WebSphere grew from a rudimentary to an increasingly powerful product\(^2\).

To gain market share, IBM deployed a penetration pricing tactic by pricing its product for 10 to 20% lower than BEA. IBM has 8,000 engineers working on its WebSphere application server products, compared to just 700 engineers at BEA\(^3\). IBM WebSphere grew from an initially work-in-progress product to enterprise class offerings. Also, IBM can discount heavily because it sells so many different pieces of hardware, software, and services — and typically makes its profits in services. That puts BEA — which only sells software — at a tactical disadvantage. Moreover, IBM has strong sales connections with many companies where the IT infrastructures are established using IBM hardware (IBM Mainframe, RS6000, AIX, etc) and software (Lotus, Tivoli, MQ, DB2, etc). Naturally, the easiest path to migrate to e-businesses for these companies is to install IBM

---

\(^1\) Application Integration and Middleware is defined as the system software or runtime infrastructure used to provide intra- and inter-application communication. Middleware is typically layered between an application program and the operating system and network transport service. (Gartner)


WebSphere.

IBM's share of the market for the software moved from 31 percent in 2001 to 37 percent in 2002. BEA slipped from the top spot, dropping from 2001's 34 percent to 29 percent in 2002. IBM and BEA collectively control 66 percent of the market for application servers that adhere to the J2EE specification, according to Gartner Dataquest numbers. Besides IBM, BEA is also facing competition from other large IT vendors such as Oracle, Sun and SAP, which offered increasingly competitive Java application server technology, aggressive pricing and a broad portfolio of other surrounding technologies. Also, open-source (JBOSS) and commercial but free Java application servers (Sun ONE Application Server, Platform Edition) put further price pressure on BEA.

Another key competitor is Microsoft. Microsoft's .NET⁴ has become a viable substitute to J2EE Application Platform Suite (APS). Although .NET is new to the market, released at least four years after the first J2EE product, it is expected to have a rapid adoption by the many loyal Microsoft developers worldwide, especially among Small Enterprises. Over the long run, Microsoft may turn out to be an even more daunting a competitor for BEA attacking from below.

On the customer side, business leaders are facing unprecedented challenges in meeting their customers’ and shareholders’ expectations. The time to market for new ideas and business plan has reduced tremendously. There is a need for timely and well-informed decisions to translate into responsive, competitive actions in line with their vision and strategy. However, many businesses are in the vicious cycle where projects are obsolete before they are finished because IT has to spend so much time integrating new functionality and aggregating data within the company’s legacy, heterogeneous, decentralized and complex environment.

Now the business pressure is cascading to IT management. The IT departments are tasked with providing the technical underpinnings allowing the business to move forward, delivering on its objectives. The IT team is continually asked to do more with less, and to deliver yesterday. There is little to leverage between projects, projects do not scale and the business objective continue to evolve more quickly than IT can respond.

⁴ Microsoft .NET is Microsoft’s strategy and implementation of connecting people, information, systems and devices through the use of Web Services. It includes everything needed to develop and deploy a Web Service-connected IT architecture (2003 Microsoft Annual Report).
The landscape of companies’ application infrastructure software is typically complex, with difficult-to-use tools, rigid legacy technologies, and expensive and time-consuming processes for making things work together. Customers increasingly demand a better way to build and integrate mission-critical applications. They want a platform with the strength to handle the biggest projects, and a software foundation that will work backwards with previous investments and forward in delivering faster time-to-value for new IT projects. Ideally, customers want an enterprise application infrastructure that will make them a more productive, efficient, and responsive business.

One of the root problems is the artificial separation between business application development and the integration with the legacy environment. There is a need for more seamless business integration between the various business processes and systems within the organization. There is a business need for application platform vendors to provide pre-integrated products to help reduce technical complexity and risks. Vendors need to continue to invest in the development of innovative products to improve functionality, reduce integration complexity and to enable ease of development. Many embarked on a journey to build vertically integrated APS to meet the new market needs through internal development, acquisitions, alliances, licensing and educational acquisitions.
2. Structure and Method

As shown in Figure 2-1, this thesis is organized as follows. In chapter 3, I identify and provide some background of the key players in the APS software industry. Chapter 4 describes some of the key trends in the software industry. In chapter 5, I present frameworks to analyze the competition and summarize prior research to identify key implications that will have bearing on the analysis. Following chapter 5, chapter 6 and 7 provide comparative analysis on IBM versus BEA and Microsoft versus BEA respectively. Chapter 8 attempts to identify the key market drivers and capture some of the key market dynamics in the software industry. Chapter 9 explores the potential future scenarios in the APS software industry using scenario analysis and identifies strategic options for companies in the industry in preparation for future challenges and in the event of strategic changes in the industry.

Figure 2-1  Structure of this thesis
3. Key Industry Players

3.1. Introduction

The major categories of related players in the APS software industry are Java platform providers (IBM, BEA, Sun and Oracle), Microsoft platform provider (Microsoft), open-source provider (JBoss) and packaged business application providers (SAP, PeopleSoft, Oracle and Microsoft). Some of the important complementors are Independent Software Vendors (various ISVs) and Consulting/Systems Integrators/Outsourcing providers (IBM Consulting, Cap Gemini Ernst & Young, Accenture, KPMG Consulting, Deloitte Consulting, IBM Global Services, Electronic Data Systems, Computer Sciences and HP).

In this chapter, I’ll provide a brief company background of the key players: IBM, BEA, Microsoft, SAP and JBOSS.

3.2. IBM

IBM, a diversified IT vendor, is the world’s largest computer hardware company and a leading provider of business software, IT services and financing. IBM is headquartered in Armonk, New York and employs more than 316,000 people worldwide. For the fiscal year ended December 2003, the company generated revenues of $89 billion.

IBM develops and manufactures computer systems, software, networking systems, storage devices and microelectronics. It is increasingly focusing on IT services, which accounted for over 48% of sales, and software, which accounted for 16% of 2003 revenues. IBM’s Software business consists primarily of middleware and operating systems. The company’s middleware brands include WebSphere, Data Management, Tivoli and Lotus.

The WebSphere application server and deployment platform is built on open standards such as Java and XML with support for more than 25 operating systems ranging from IBM’s own eServer systems to various versions of Unix, Linux and Microsoft Windows. There are more than 40,000 WebSphere developers and more than 35,000 customers.

With its 8,000 engineers working on its WebSphere application server products, its ability to bundle and discount heavily on software and make its profits in services, and its 175,000 Global Services employees with end-to-end capabilities, deep contacts in companies as well as its decades of
experience selling mainframes and other hardware, to market its platform to enterprise customers, IBM is the greatest challenger to BEA in the Java APS software market.

3.3. BEA

BEA is one of the world’s leading application infrastructure software companies with more than 15,000 customers around the world, including the majority of the Fortune Global 500 companies. BEA employs more than 3,100 people, is headquartered in San Jose, California, and has 93 offices in 34 countries. For the fiscal year ended Jan 31, 2002, the company recorded revenues of $976 million.

With the initial success of BEA’s application server and building up of a critical market share, BEA is in a position to achieve a dominant position in the enterprise application software industry. Similar to Microsoft’s strategy of encouraging ISVs to develop based on Microsoft Windows platform, BEA strived to nurture, attract and retain these ISV complementors to enhance BEA’s offerings. BEA defined an open architecture and made available integration interfaces for any third party best-of-breed component provider to integrate seamlessly with the BEA platform. With its initial critical market penetration, many end-users demand other third party best-of-breed component provider to be integrated with BEA platform. The enhanced BEA’s offering which provides a broad horizontal suite of products further attract more customers, and result in increasing growth for both BEA and its partners. BEA is one of the platform leadership wannabe trying to achieve the Holy Grail of all high-tech industries.

BEA’s WebLogic Enterprise Platform delivers a highly reliable, scalable software infrastructure designed to bring new services to market quickly, to lower operational costs by automating processes, and to link up with suppliers and distributors. The platform brings together six application infrastructure technologies, namely portal, personalization, application integration, business process management, security and application management, into one unified, simplified and extensible platform. A single unified architecture reduces complexity and total IT cost.

Companies turn to BEA to help them evolve their existing enterprise software applications from inflexible, redundant, legacy architectures to highly responsive, mature Web infrastructures. Companies built on BEA software aim to use IT to affect rapid change within their organizations and achieve breakthrough levels of efficiency and responsiveness. BEA’s customers use BEA products as a development platform for Internet-based applications, including custom-built and
packaged applications, and as a means for robust enterprise application integration among mainframe, client/server and Internet-based applications.

3.4. Microsoft

Microsoft Corporation was founded in 1975 and incorporated in 1981. Microsoft employs approximately 55,000 people and is headquartered in Seattle, Washington. For the fiscal year ended June 30, 2003, the company recorded revenues of $32 billion with operating income of $13 billion. Microsoft is a software focused company with a wide range of software products, which include operating systems for servers, PCs, and devices; server applications for client/server environments; information worker productivity applications; packaged business applications; and software development tools for a multitude of computing devices. Microsoft’s seven product segments are:

- Client (Window XP, Windows 2000, and other standard Windows operating systems);
- Server and Tools (Windows Server, SQL Server, Exchange Server, and other servers);
- Information Worker (Suite of personal productivity products such as Microsoft Office XP consisting of Outlook, Excel, PowerPoint, Word and Access, and a suite of collaboration products designed to increase personal, team and organization productivity such as SharePoint Portal Server);
- Microsoft Business Solutions (Suite of packaged business applications designed to help small and mid-sized businesses (SMB) become more connected with customers, employees, partners, and suppliers);
- MSN (Internet access, email, online instant messaging, search, content);
- Mobile and Embedded Devices (Windows Mobile software that powers PDA and mobile phone, and Windows Embedded software used in advanced consumer electronics including digital televisions, IP-based set top boxes, etc);
- and Home and Entertainment (Microsoft Xbox video game system, PC games, TV platform and other home products).

Microsoft has been a very profitable company and its desktop PC operating system has a dominant market share installed on over 90% of all desktop PC. Referring to Microsoft’s FY03 financial results shown in Table 3-1, the Client and Information Worker segments are Microsoft’s two most profitable segments, contributing over 60% of Microsoft’s FY03 revenue of $32 billion and all of FY03 operating income of $13 billion including covering the losses of other business segments. The only other profitable segment is Server & Tools segment, the unit that includes the .NET
platform which competes with the Java platform providers. All the other segments are currently in the red, with a total loss of $4.7 billion in FY03.

Table 3-1: Microsoft’s Product Segment Revenue and Operating Income for FY 2003

<table>
<thead>
<tr>
<th>(In millions)</th>
<th>Revenue</th>
<th>Operating Income/(Loss)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Year Ended June 30</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client</td>
<td>9,360</td>
<td>10,394</td>
</tr>
<tr>
<td>Server and Tools</td>
<td>6,157</td>
<td>7,140</td>
</tr>
<tr>
<td>Information Worker</td>
<td>8,212</td>
<td>9,229</td>
</tr>
<tr>
<td>Microsoft Business Solutions</td>
<td>308</td>
<td>567</td>
</tr>
<tr>
<td>MSN</td>
<td>1,571</td>
<td>1,953</td>
</tr>
<tr>
<td>Mobile and Embedded Devices</td>
<td>112</td>
<td>156</td>
</tr>
<tr>
<td>Home and Entertainment</td>
<td>2,453</td>
<td>2,748</td>
</tr>
<tr>
<td>Other</td>
<td>192</td>
<td>—</td>
</tr>
<tr>
<td><strong>Consolidated</strong></td>
<td>28,365</td>
<td>32,187</td>
</tr>
</tbody>
</table>

The increasing popularity and adoption of open-source Linux server and the potential future threat in its dominant client desktop and productivity applications business from open-source Linux desktop and open-source OpenOffice desktop productivity suite is expected put price pressure on Microsoft’s three most profitable segments.

The key challenge for Microsoft is whether it can extend its dominance from Windows desktop to other areas of its business and improve the performance of these other divisions in order to replace the potential loss or reduction of revenues of its Client and Information Worker segments due to the commoditization effect from open-source software.

---

5 2003 Microsoft Annual Report.
3.5. **SAP**

Germany-based SAP is the leading maker of business application software, with $7.7 billion in annual sales in 2003. The company's core business is in developing complex software programs designed to streamline corporate bookkeeping, order processing, customer service, human resource administration and manufacturing.

SAP's key competitor in the packaged business application market is PeopleSoft and Oracle. In the past year in 2003, SAP has increased its market share in the US to 37 percent from 26 percent, winning ground against Oracle and PeopleSoft. Worldwide, SAP has a market share of 59 percent.

The other up and coming competitor is Microsoft, which entered the market through the acquisitions of two ERP vendors that target midsized firms: Great Plains Software, purchased in April 2001 for $1.1 billion, and Navison, a Danish company, purchased in May 2002 for $1.3 billion. In response Microsoft's entry, SAP announced its intention to rekindle an initiative it had abandoned several years ago to sell software to small businesses. SAP has since introduced two new sets of products, one for companies with less than $1 billion in annual revenue, called SAP All-In-One, and one for companies with fewer than 150 employees, called Business One. With these products, SAP wants to grow revenue from sales to companies in the $1 billion-and-under segment – the same SMB market that Microsoft targets – from around 6 percent of software license revenue to at least 15 percent by 2005.

The implementation of large scale SAP business applications is complex and risky. To make its line of business applications easier to modify and better able to work with other systems, SAP designed a new software architecture called NetWeaver 2004. NetWeaver, first unveiled in 2003, is the new technical foundation for SAP's application software. The updated version is SAP's most significant product release since it introduced its flagship R/3 programs in the mid-1990s, which marked the company's move from mainframe to client-server technology.

NetWeaver 2004 is a collection of new and revamped products packaged to run on one server. Previous versions of the programs required separate servers and different underlying technologies.

---

Included in the latest NetWeaver package are SAP's portal software and application server program, integration tools, data mining systems, workflow programs and the company's mobile computing infrastructure.

However, some components of NetWeaver, including the application server and data sharing programs, may put SAP at odds with its longtime partners, IBM, Microsoft, BEA and others who make products that compete in the same market.

3.6. JBOSS

JBOSS Inc. is a small, privately owned company (fewer than 30 employees) and its principal software product is the open-source J2EE application server with the same name as the company, called JBOSS. The company revenue is derived from consulting and services sold in conjunction with the deployment of JBOSS, as well as from training and documentation for the users of JBOSS. While the partner companies distribute the product and provide first-level support, JBOSS is the exclusive source of expert (Level 2 and 3) support for JBOSS. JBOSS Inc. was formed in 2001 and is now widely recognized as a growing alternative to commercial J2EE application servers of low and medium grade.

There are more than 250 organizations that have directly selected and deployed JBOSS for production use. Hundreds more deployed packaged applications or tools with JBOSS embedded inside. The product is not yet well-established as a reliable platform for mission-critical applications at mainstream enterprises (although there are some showcase examples of such applications), but it is growing in adoption by ISVs as the default embedded J2EE platform for the ISVs' applications or tools.

There is a key difference between JBOSS and most other open-source products such as Linux. JBOSS adopted a modified open-source strategy where the company both develops and distributes the product. It combines the roles of the open-source consortium and open-source distributor. This gives JBOSS much greater control of the business evolution of the product, provides for the services and support revenue, and, most importantly, exclusively associates the brand of the JBOSS

---

product with the JBOSS company. In addition, the Lesser General Public License (LGPL) open-source licensing allows embedding of JBOSS into commercial products, broadening the commercial impact of JBOSS.
4. Trends in the APS Software Industry

4.1. Introduction

In this chapter, we will identify some of the key business and technology trends related to the APS software industry and conduct an analysis to identify key implications with potential major impact to the industry.

4.2. Business Trends

4.2.1. Stages of IT Architecture and the impact on the locus of control of IT Architecture and Standards

Introduction

In the research by Ross on “Creating a Strategic IT Architecture Competency: Learning in Stages\textsuperscript{9}”, she finds that most firms’ IT capabilities limit rather than create new business opportunities. Their limitations come from their history of applying IT as a response to specific business needs. The business needs are isolated and their solutions rarely combine to create a strategic capability. As a result, most firms see their IT architectures\textsuperscript{10} as competitive liabilities in times of the changing market conditions.

Through the study of 40 case studies of firms, conducted by Ross and her colleagues\textsuperscript{11} from 2001 to 2003, that are evolving their IT architectures from sets of isolated solutions to planned capabilities that support their strategic business processes, Ross finds that firms do not necessarily see their IT architectures as competitive liabilities. The case studies illustrate that firms hone their ability to define and align IT and business strategy by accumulating architecture-related experiences. When used to enrich organizational learning, these experiences can create enterprise IT architecture competencies. The following is the executive summary of this research.

---


\textsuperscript{10} Ibid. Consultants and researchers often refer to an enterprise IT architecture as a kind of city plan that details policies and standards for the design of infrastructure technologies, databases, and applications. At the enterprise level, an IT architecture is the organizing logic for applications, data, and infrastructure technologies, as captured in a set of policies and technical choices, intended to enable the firm’s business strategy. Accordingly, the enterprise architecture implies certain IT capabilities. These capabilities are the objectives of the IT architecture, specifying what the architecture enables the business to do.
IT architecture is often assumed to follow business strategy, to align IT with the business’s strategic objectives. Increasingly, though, many business strategies depend on specific underlying IT capabilities. To develop a synergy between business strategy and IT architecture, firms must develop organizational competencies in IT architecture. My research has identified four IT architectural stages, each with its own requisite competencies. The “application silo architecture stage” consists of IT architectures of individual applications. The “standardized technology architecture stage” has an enterprise-wide IT architecture that provides efficiencies through technology standardization. The “rationalized data architecture stage” extends the enterprise-wide IT standards to data and processes. And the “modular architecture stage” builds onto enterprise-wide global standards with loosely coupled IT components to preserve the global standards while enabling local differences. Each stage demands different organizational competencies to implement the architecture and prepare the firm to move to the next stage.

The four architectural stages

Figure 4-1 presents an overview of the four architectural stages. It shows how IT resources are allocated during each stage.

11 Peter Weill, David Robertson, George Westerman, Nils Fonstad, Cynthia Beath and John Mooney.
Figure 4-1: Changing Resource Allocations across Architecture Stages

Copyright 2003 MIT Sloan School Center for Information Systems Research. Used with permission.

Another CISR Research Briefing article on “Stages of IT Architecture: Pursuing Alignment and Agility”\(^\text{12}\) describes the characteristics of each architecture stage as follows:

\((1)\) The Application Silo Architecture Stage

*In the first stage, Application Silos, a firm invests most of its IT resources in functional applications with a small investment in a data center that performs centralized...*\(^\text{12}\)

---

transaction processing. They funded applications based on the individually calculated costs and benefits of inserting technology into a narrowly defined process.

The first stage allows functional optimization, but firms burdened with Application Silo legacies have found that their IT capability inflexible to changing business needs. Application silos are expensive to maintain and slow to build. As firms create complex webs of applications, they become progressively slower and support becomes more expensive.

(2) The Standardized Technology Architecture Stage
In an effort to reduce IT costs and enable integration across functions and business units, firms move into the second stage, Centralized Core. In this stage, resources are shifted from application development to infrastructure development as the firm extracts infrastructure from applications and introduces enterprise-wide standards for technologies and systems. The firm also invests in data, creating data warehouses to collect historical data for analytical purposes.

Firms of all sizes, realize significant benefits from standardizing on a core set of technologies. Technology standards enable them to reduce costs by limiting skill requirements and technical complexity. Reduced technical complexity also facilitates cross-functional integration. Firms implement a Centralized Core architecture to align with existing or intended organizational structure rather than strategy. A firm's strategy is embodied in its processes, and the Centralized Core architecture does not align with business processes.

(3) The Rationalized Data Architecture Stage
Firms in the third stage, Hardwired Business, build their key IT capabilities around their core business processes, and thus align architecture with business strategy. In this stage, firms further diminish the relative proportion of resources spent on applications because core transaction processes become "wired" into the infrastructure. Investment in data increases as firms create large databases of transaction data for use in integrated systems.

Hard-wired firms look for opportunities to leverage their IT capability. Many
manufacturing firms have hardwired their supply chain processes through packaged enterprise resource planning, supply chain and customer relationship management systems.

(4) The Modular Architecture Stage
Hardwiring provides speed to market with services that extend the core business. Hardwiring does not provide strategic agility. To position itself for a broader range of strategic options, a firm must move into the fourth stage, Modularity. This stage involves isolating data from existing processes and creating reusable technology components. Then, when new business opportunities arise, a firm can leverage its data and technology components.

Resources in the Modularity stage are focused in developing reusable components and building and maintaining data stores wrapped in middleware for easy access. Applications in the Modularity stage are focused on experiments for testing new business concepts. This allocation of resources allows a firm to expand rapidly into adjacent businesses by assembling appropriate components to follow up on successful experiments.

Architecture stage of the forty firms in the study
As reflected by Ross, in the early 1990s, most firms were in the first Application Silo Architecture stage. Most of the forty firms involved in the study are now in the Standardized Technology Architecture stage. As the development of components, both internal and vendor-provided, is in early stages, few firms have moved into the fourth stage.

However, there is probably a skew of the statistics towards the right as the forty firms involved in the study are typically large, mature, and leading enterprise firms. Therefore, many other firms are probably in the Application Silo stage transiting into Standardized Technology Architecture stage.

Ross highlighted that different business units and processes within a firm will likely move through the architecture stages at different points in time. She recommends that firms move core processes first because they will provide the greatest opportunity for leveraging IT capabilities. Large, diversified firms may never find value in moving beyond the Centralized Core as an enterprise (although business divisions or customer segments will benefit from migration to more mature states).
How locus of IT Architecture and Standards decision making change across the architecture stages

Figure 4-2 provides a summary of the characteristics and types of learning in each architecture stage. Each stage defines a dramatically different relationship between IT and business executives and between IT architecture and business strategy.

<table>
<thead>
<tr>
<th>IT Capability</th>
<th>Application Silo</th>
<th>Standardized Technology</th>
<th>Rationalized Data</th>
<th>Modular</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT applications serve isolated business needs</td>
<td>Firm-wide technology standards</td>
<td>IT focused on wiring core process</td>
<td>Modules enable business model extensions</td>
</tr>
<tr>
<td>Key Management Innovation</td>
<td>Technology-enabled change management</td>
<td>Standardization and exception management, refresh</td>
<td>Recognizing essence of the business</td>
<td>Practices facilitating reusability</td>
</tr>
<tr>
<td>Business Case for IT</td>
<td>ROI of applications</td>
<td>Reduced IT costs; interoperability</td>
<td>Improved business performance; integration</td>
<td>Speed to market; Strategic agility</td>
</tr>
<tr>
<td>Locus of Control</td>
<td>Local control</td>
<td>Senior management support of CIO</td>
<td>Senior management, IT, and process leadership</td>
<td>Senior mgmt, IT, process, and local leadership</td>
</tr>
<tr>
<td>Key Governance Issues</td>
<td>Estimate, measure, communicate value</td>
<td>Establish (local/ regional/ global) standard setting, exception &amp; funding processes</td>
<td>Determine core processes and funding priorities</td>
<td>Define boundaries for business experiments</td>
</tr>
</tbody>
</table>

Figure 4-2: Characteristics of the Architecture Stages

Source: Jeanne W. Ross, “Creating a Strategic IT Architecture Competency: Learning in Stages,” CISR WP No. 335 and Sloan WP No. 4314-03.

Copyright 2003 MIT Sloan School Center for Information Systems Research. Used with permission.

One important factor impacting software vendors marketing and sales strategy is the change in locus of control of IT architecture and standards as firms evolve through the architecture stages.
The Application Silo Architecture stage allows an arms-length relationship between IT architecture and business strategy. Business people can define strategy without IT input, and IT can deliver solutions without understanding the business strategy.

In the Standardized Technology Architecture stage, business and IT managers make decisions on IT standards based on their negotiated understanding of the impact of IT on business strategy. They also negotiate funding models for the shared infrastructure, including replacing and upgrading technologies before they become obsolete. At this stage, IT and business managers develop governance structures, such as executive committees, to formalize funding for the shared infrastructure, both for new infrastructure development and replacement. The executive committee also debates the appropriate organizational level for IT standards. Architecture committees, typically populated with IT people, establish processes for developing, monitoring, and granting exceptions to standards.

In the Rationalized Data Architecture stage, IT capabilities shape, as well as respond to, business strategy. IT and business managers clarify strategic intent, critical IT capabilities, and the target enterprise architecture. These discussions eventually produce consensus on the firm’s core processes and the data that drives them. The executive committee continues to address strategic IT prioritization and investment issues.

Finally, the modular architecture stage introduces the challenges of componentization, customization, strategic experiments, and reuse. In this stage, IT and business management introduce new governance mechanisms to encourage component reuse, and they retain governance mechanisms that support funding, standardization, and IT value assessment.

**Increasing influence and power of CIO on technology standards**
One of the findings of the study is that in the Standardized Technology Architecture stage, business managers rarely participate in developing the enterprise IT architecture. They defer to the IT organization to set the technology policies and standards. However, senior business managers are anxious for IT cost savings, and they support the CIO’s efforts to standardize and centralize infrastructure technologies by mandating compliance with the technology standards. Therefore, the CIO is empowered to establish and enforce technology standards.
Implications

# 1: Business case for IT shift from ROI of individual applications, to reduced total cost of ownership and integration, to improved business performance and integration and then to speed to market and strategic agility.

# 2: Specialized products for niche requirements increasingly have to go through an exception process before adoption.

# 3: Infrastructure software vendors marketing and sales target audience changes as IT standards approval shift from business unit IT manager to CIO and business executives.

# 4: Due to senior business managers desire to control cost, CIO is increasingly empowered to establish and enforce technology standards.

4.2.2. Increased adoption of open-source software

In this section, I will be discussing an important trend impacting all commercial software companies - the increasing adoption of open-source software. I will cover some background on open-source license and the implications to commercial software production, the adoption trend and potential threat to commercial software companies, some of the challenges of open-source software development and the evolution of the open-source model, and some business and political factors driving the adoption of open-source software.

Introduction

Most commercial software is distributed only in binary form (encoded for use directly by the computer hardware and operating environment). For open-source software, the underlying source code (the high-level programming instructions that tell the computer what functions to perform) is distributed along with the binary form. Having access to a program’s source code allows the user to understand how a program operates and, if they should desire, to make changes to that program.

The famous open-source GNU General Public License (GPL) is written by Richard Stallman, the founder of Free Software Foundation\textsuperscript{13}. Stallman, a computer programmer formerly working at

\textsuperscript{13} Free Software Foundation is a tax exempt charity organization dedicated to promoting computer users' right to use, study, copy, modify, and redistribute computer programs; promoting the development and use of free (as in freedom) software; and helping to spread awareness of the ethical and political issues of freedom in the use of software.
MIT’s Artificial Intelligence Laboratory, who frequently wrote and share his own programs, was frustrated in his attempts to improve several commercial programs purchased by his lab because the manufacturer refused to share its source code with him. Stallman was also concerned that the rise of proprietary software threatened the ability of programmers to share ideas and advance programming according to the norms of open science.\textsuperscript{14}

To ensure that his code would always be freely modifiable and distributable, he created the GNU GPL. The GPL specified that users of the source code could view, change, or add to the code, provided that they made their changes available under the same license as the original code.

**Key implications to commercial software productions by GPL terms and conditions**

In the preamble of the GNU GPL description\textsuperscript{15}, it describes the license as:

\begin{quote}
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software – to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.
\end{quote}

The key implications to commercial software productions are:

\begin{itemize}
  \item This License applies to any program or other work which contains a notice placed by the
\end{itemize}


copyright holder saying it may be distributed under the terms of this GPL. The “Program” refers to any such program or work, and a “work based on the Program”: a work containing the Program or a part of it, either verbatim or with modifications and/or translated into another language.

- You may modify the source code of a Program distributed under the GPL license, provided you agree to make the work that you create, that in whole or in part contains or is derived from the Program or any part thereof, to be under the same GPL license. This means that you must agree to make your work freely distributed at no charge to all other users under the terms of this License.

- The requirements apply to the modified work as a whole. That is to say, if you include a GPL license program in your own program, it makes the whole program to be under the terms of GPL license to be freely distributed to other users.

- This GPL does not permit incorporating the Program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License (GNU LGPL) instead of GNU GPL. That is, licensing your work under GNU LGPL allows it to be linked and used by proprietary applications without making the proprietary application to be under the License.

Other important points about GPL are:

- Mere aggregation of another work not based on the Program with the Program on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

- Activities other than copying, distribution and modifications are not covered under this License; they are outside its scope. The act of running the GPL licensed Program is not restricted.

- You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Increasing Adoption Trend

The best-known programs whose source code is openly available and free, for the most part, are the Linux operating system, the Apache Web server, the SendMail program for using the Internet to send and receive e-mail, the Mozilla Internet browser (the open-source version of Netscape
Navigator), and basic desktop applications such as the OpenOffice suite.

In 2004, Apache powers 68% of active Web sites, up from 54% two years ago, according to Netcraft, a company that surveys software used on Web sites. Apache has consistently captured market share at the expense of two formidable commercial competitors, Microsoft with its Internet Information Server, and Sun Microsystems with its SunOne Web server\textsuperscript{16}.

The popularity of Linux in the server can be seen from the growth of the Linux server shipments. Linux share of the worldwide server OS market share (% of all units shipped) has grown from 6.5% in 1996 to about 27% in 2000\textsuperscript{17}. In the fourth quarter of 2003, Linux server shipments grew 63% to $950 million from the fourth quarter of 2002 to the fourth quarter of 2003. Linux servers outgrew Windows server which has a growth of 16% in the same period with $3.9 billion revenue\textsuperscript{18}.

Potential threat to commercial software companies' profitability

The price of commercial software, or for that matter for any products, is determined by how much customers are willing to pay based on the value to the customers. Values, however, is affected directly by what other companies offer for similar product or service, as well as by what is freely available.

All commercial software companies faced direct competition with firms adopting alternative business models to the commercial software model. Firms adopting the open-source model typically provide customers with open-source software at nominal cost and earn revenue on complimentary services and products, without having to bear the full costs of research and development for the open-source software.

The growth of Linux server has prevented Microsoft from capturing more value from firms


transiting out of the higher cost Unix servers to Intel-based low cost servers. A lot of growth that Microsoft has anticipated from Windows Server 2003 is at risk because of the increasing popularity of Linux. In addition, Microsoft is also facing potential future threat in its dominant client desktop operating system and productivity applications business. Red Hat, the leading Linux server distributor, announced that it plans to enter the Linux client marketplace sometime in 2004. In August of 2003, Sun Microsystems announced the Sun Java Desktop System (JDS), a software package that consists of a fully integrated client environment based on open-source components and industry standards, including open-source Linux desktop and OpenOffice desktop productivity suite.

The continuous improvement of the Linux Desktop has made it become a viable alternative to Windows desktop. In May 2003, the city of Munich, Germany, awarded a contract to IBM to install Linux desktop for its 14,000 computers instead of installing Microsoft Windows. In Sep 2003, the State of Massachusetts decided to adopt a broad-based strategy of moving its computer systems towards open standards, the argument of which is not only to reduce licensing fees but also "by a philosophy that what the state has is a public good and should be open to all." And in Nov 2003, Sun announced that it has signed a deal with the China Standard Software Co. for 1 million licenses to deliver its Sun JDS in that country. Following that in Dec 2003, Sun announced a strategic five-year agreement with the United Kingdom's Office of Government Commerce to establish Sun JDS as the desktop solutions for the public sector. There is a growing momentum for Linux on the

---

desktop. However, it remains to be seen how widespread and successful the adoption will be.

Nevertheless, the competition from Linux and other open-source software appears to be helping enterprises negotiate a better deal with Microsoft. Sun is offering Sun JDS at a special price of $100 per desktop, or $50 per employee whereas a Microsoft Office suite cost about $400 to $50025. James Governor, principal analyst at RedMonk, expects Microsoft to "take a hit" when it comes to profit margins and said this is a good time to negotiate a better deal with the emergence of a real alternative. Gary Barnett, research director at Ovum, agreed that Microsoft is finding it increasingly difficult to maintain its high profit margins in Office with increasing pressure to offer special deals on price.26

The price pressures caused by open-source software do not only affect Microsoft but also all other commercial software vendors. MySQL, another open-source database product, grew 30 percent in 2003 compared with a 6 percent growth rate for Microsoft’s SQL Database Server. "Although MySQL is unlikely to displace any vendor in the near term, it is likely to put pressure on the top commercial database players [Oracle, IBM, Microsoft]," said Noel Yuhanna, a database analyst at Forrester Research.27 Another open-source product, JBOSS, which competes with BEA and IBM in the J2EE application server space, is also growing in popularity. According to the "Third Annual Java Use and Awareness Study," conducted in 2003 by BZ Research on 15,000 subscribers of Software Development Times, almost 27 percent of respondents reported using JBOSS in their company or at companies for which they consult, compared with 14% in 2002. The 93 percent increase represents the largest percentage growth among major J2EE application servers, including BEA WebLogic and IBM WebSphere28, putting pressure on the top commercial J2EE players.

Challenges of traditional open-source model

---

Despite the growing success of open-source movement and several open-source products, there are many more open-source products that the user community remains within the small developers' community with little widespread adoption. The article by Michelle Levesque discusses some of the issues with open-source software development that makes open-source software inaccessible to most of the general public. The five issues she identified are:

- The lack of focus on user interface design causes users to prefer proprietary software's more intuitive interface.
- Open-source software tends to lack the complete and accessible documentation that retains users.
- Developers focus on features in their software, rather than ensuring that they have a solid core.
- Open-source programmers also tend to program with themselves as an intended audience, rather than the general public.
- Lastly, there is a widely known stubbornness by open-source programmers in refusing to learn from what lessons proprietary software has to offer.

For the open-source product to be widely used by the general public, Levesque said that the above issues will have to be overcome.

**Emergence of second-generation of open-source model**

The success of first-generation of open-source software, example Apache, Linux and SendMail, are all similar in that these softwares formed part of a computer system's infrastructure. This was a natural focus for open-source projects, given that users of these softwares typically have strong technical skills that allowed them to find and fix problems with the software. However, due to the issues identified above for pure community-based open-source software development, the suitability for other types of product is still in doubt.

---

As a result, a second-generation of open-source model emerges which adopted a dual-license mode of offering products (both MySQL and JBOSS are under this model) under both an open-source license and a commercial license. This allows open-source projects to use the software at no cost, which contributes to widespread use and testing of the software and the fast growth of a large installed user base. Companies redistributing the software as part of commercial products can also get the benefits of the open-source software by purchasing a commercial license, which releases them from requirements to publish their source code. Commercially-licensed customers generate revenues for the open-source vendors, which contributes to the rapid development of high-quality software. Also, successful second-generation open-source vendors usually make the majority of their revenue from selling software licenses. This license-based business model offers higher margins than services-based businesses. Historically, most open-source companies such as Red Hat and SUSE LINUX which distributes Linux, have tried to make money by selling services and support.

Beside the benefits of open-source software such as the ability to tap on an established community of open-source developers and freedom from vendor lock-in, dual-licensed products gain all of the benefits traditionally associated with commercial software, including:

- A paid, full-time, dedicated development team, providing continuity, a roadmap of upgrades and professional documentation;
- Enterprise-level, production and developer support available 24/7/365; and
- Confidence that the vendor has full rights and/or ownership of all intellectual property in the software.

Challenges of second-generation open-source model

Although there are some initial successes of second-generation open-source model, the longer term dynamics and evolution remains unclear. Some of the challenges faced by the second-generation open-source companies are:

- Due to the typically narrower and specialized scope, it is not able to tap into the vast Linux professional and voluntary talent pool. The companies are largely responsible for its programming needs and rely on the open-source community for debugging, testing and other tasks. That means it essentially competes with commercial competitors’ on product
development.

- One key difference this second generation of open-source products is different from its predecessors is that it may lack the support of an established technology company to help in the products adoption. The endorsements from influential technology companies such as IBM helped propel Linux and Apache. But some of the newer open-source offering threaten the market share of these influential technology companies. JBOSS performs functions similar to IBM's WebSphere, while MySQL is a relational database that competes with Microsoft SQL Server, IBM's DB2, and Oracle's flagship system.

- Alternatively, the companies need to obtain endorsement from influential technology companies with no large vested interest in their products’ market. MySQL’s recent success can be partly attributed to the endorsement by SAP which turned over the source code for its SAP DB relational database, its transactional and high-performance database running SAP business applications, to MySQL AB after SAP decided that it is not a database vendor.\(^{32}\)

Business and political factors driving the adoption of open-source software

There are business, political and anti-monopolistic factors that provide motivations and incentives for organizations and commercial companies to support open-source software adoption.

1) As a competitive tool to disrupt market

Some vendors use open-source as a weapon for disrupting markets that they do not control in order to weaken competitors or suppliers position. For example, IBM supported Linux and Apache to disrupt Microsoft’s dominance of the operating system. SAP supported MySQL to commoditize the database tier to first, attack the profit margin of Oracle and Microsoft as competitors in the business application market, and second, to weaken the power of Oracle as a database supplier for some of SAP’s applications that rely on Oracle database.

2) Political and anti-monopolistic actions

Open-source is gaining popularity, particularly in developing nation, who thinks that the cost of proprietary software is too high, and who are worried about security and being beholden to a

proprietary vendor. A case in point is the collaboration between China Standard Software Co., Ltd. (CSSC) and Sun on Linux desktop. The CSSC is a consortium of Chinese technology companies supported by the Chinese government to produce a nationwide standard desktop software system to help bridge the digital divide among the nation's 1.3 billion citizens, and help to fuel the technology growth in China. The desire for more choice, affordable price and higher level of information security is reflected in the following statement by a Chinese official.\textsuperscript{33}

"Linux and open-source software bring new opportunities not only to China, but also to other countries around the world," said Li Wuqiang, deputy director-general, Department of High and New Technology Development and Industrialization, Ministry of Science and Technology (MOST) People's Republic of China. "A desktop solution based on open standards means more choice, an affordable price and a higher level of information security. China warmly welcomes international cooperation in this area, such as this agreement between CSSC and Sun."

The collaboration between three North Asian countries, China, Japan and Korea, to co-develop an open-source operating system to replace Microsoft Windows is another initiative by Asia to leapfrog the Western countries in the development of Linux-based applications, especially in the area of embedded Linux that powers electronic devices such as video recorders, cell phones and network routers. For countries with a strong manufacturing base such as China, Japan and Korea, Linux gives them a chance to create an operating system free of licensing fees and with full control over the source code.\textsuperscript{34}

Conclusion
The challenges posed by open-source software to commercial software companies are multi-faceted and complex. Some of the innovation-related incentives and the benefits that an innovator may derive from freely revealing an innovation back to the open-source community will be discussed in


Section 5.8 on Open Source Innovation.

Implications

# 5: Microsoft’s profit margin is the most affected by open-source products as it competes mainly in the lower SMB market, the market segment most likely to consider open-source alternative due to cost considerations.

# 6: Second-generation open-source companies essentially compete with commercial competitors’ on product development.

# 7: For widespread adoption, open-source companies need to obtain endorsement from influential technology companies.

# 8: Open-source is increasingly being used as a tool to disrupt market.

# 9: Political and anti-monopolistic factors, especially from countries outside of US, may ultimately drive the adoption of a wide spectrum of open-source products.

4.2.3. Antitrust

Introduction

Determining what kind of an organization you want to be is an important decision for companies in a platform leadership position, which often comes with a monopoly position (defined as 70 percent or more of a market). In their drive to compete, platform leaders need to be careful not to cross the line into unethical behavior that will get a company into trouble not only with the antitrust authorities and the Securities and Exchange Commission but also with customers, employees, industry analyst, investors, and business partners.35

Microsoft, the dominant player in the desktop PC operating system with over 90% market share is definitely in a monopolistic position. The competitive actions by Microsoft not only have major impact to Microsoft’s effort to sustain its dominant position in various software areas, but also greatly affect the entire software ecosystem.

Cases of Microsoft’s competitive actions

Microsoft has many encounters with antitrust authorities in its many controversial handling of competitors as well as even partners in order to protect or expand its market. Some of the cases

highlighted in the book “The Business of Software” by Michael A. Cusumano related to Netscape, Sun, Intel, IBM and Compaq are as follows:

- **Netscape.** Microsoft bundled its Internet Explorer browser with Windows 95 “for free,” and then strong-armed PC manufacturers to accept this bundle in order to preserve their Windows licenses. The reason for Microsoft’s aggressive response to Netscape was that the Navigator browser could also be a foundation for using Web-based applications, making Windows less valuable or even unnecessary for many PC users and potentially harming Microsoft’s desktop applications business as well.

- **Sun.** Microsoft’s decision not to include Sun Microsystems’s Java technology with Windows XP was controversial. Java is a neutral programming language that enables the PC to run on different operating systems, making computers less dependent on Windows to run applications. But, because many Web sites already used Java for animation and other small applications, lack of support for Java in Windows XP meant that users would have to take additional step and go to a special site to download a utility to view Java-enabled Web sites properly.

- **Intel.** Microsoft clashed with Intel in 1995 when the microprocessor producer tried to introduce programming tools that bypassed Windows. Intel developed its own Native Signal Processing (NSP) technology to help developers build applications with advanced video and graphics capabilities without the use of special signal processing chips. In addition, NSP interfaces were platform neutral – which meant that developers could more easily port their applications to different operating systems. Microsoft pressured PC manufacturers not to install NSP software on their computers. Ultimately, Intel gave in and agreed not to promote its NSP technology and to curtail its software development efforts. In turn, Microsoft incorporated some of the NSP features into Windows.

- **IBM.** Microsoft clashed with IBM repeatedly over the years, such as when IBM attempted to promote OS/2 as an alternative operating system to Windows, and when it promoted its own applications (Lotus 1-2-3 and the SmartSuite of office applications). As a major PC producer, IBM also had to purchase millions of copies of Windows. To pressure IBM, according to the Department of Justice documents, “Microsoft punished IBM PC Company with higher prices, a late license for Windows 95, and the withholding of technical and

---

36 Ibid.
marketing support”.

- **Compaq.** Microsoft even clashed with its long-time partner Compaq, the largest PC manufacturer in the world during the 1990s and a preferred Microsoft customer, when the PC manufacturer decided in 1996 to continue featuring Netscape Navigator on the Windows desktop and to remove the Internet Explorer and MSN icons. Microsoft threatened Compaq with a discontinuation of its Windows license. Compaq relented.

The problem with Windows XP is Microsoft’s decision to include various Internet-related technologies within the new operating system. The bundled technologies included Microsoft’s version of instant messaging (incompatible with AOL’s technology, which was the market leader) and its multimedia player (incompatible with RealNetworks’ technology, another market leader in this area). Through the bundling, Microsoft hoped that customers would find it more convenient to use the default application provided by Microsoft even if the competitor product is marginally better, and that these technologies would then become the de-facto standard through Microsoft’s existing monopoly of the desktop operating system.

As described by Cusumano, the main dilemma faced by antitrust regulators is Microsoft’s argument that customers do want the new features and they do want them bundled with the operating system to simply installation and management of the software. As part of the antitrust law suit settlement, Microsoft made policy changes to allow PC manufacturers to load more non-Microsoft software on new versions of Windows, such as XP. The courts left it somewhat unclear, however, to what extent it was legal for Microsoft to continue bundling new features into future versions of Windows. The courts were likely to evaluate challenges on a case-by-case basis, trying to weigh potential harm to competitors versus potential benefits to consumers.

**Recent changes in Microsoft’s partner strategy**

Microsoft’s recent decision in Apr 2004 to pay Sun Microsystems, one of Microsoft’s greatest staunch enemies resulting from their disputes on Java, $1.95 billion to settle antitrust and patent issues may be the greatest indications that Microsoft intends to resolve long-standing legal and

---

regulatory disputes\textsuperscript{38}. This deal follows a string of settlements with former rivals such as Apple and AOL Time Warner. The following are some of the recent settlements.

- **Apple.** In its 1997 settlement with Apple Computer, Microsoft invested $150 million in the struggling company and pledged to continue developing future versions of Office, Internet Explorer and various development tools for the Mac. That ended a long-standing tiff over whether Windows had infringed on any of Apple's patents. It also helped keep alive one of the few rivals to the Windows operating system, giving Microsoft a counterargument to monopoly charges in its antitrust cases.\textsuperscript{39}

- **Nine States in US.** In 2001, Microsoft convinced 9 of the 18 state attorneys general involved in the antitrust case to end their antitrust battle with Microsoft. Under the terms of the five-year consent decree, Microsoft must disclose to the computer industry the APIs and protocols that its middleware products and servers use to communicate with Windows. The vendor must also stop using its software licensing agreements with PC makers and independent software vendors to punish or threaten companies that work closely with Microsoft's competitors. And Microsoft agreed to a ban on contracts that require exclusive use of its products.\textsuperscript{40}

- **AOL.** In May 2003, Microsoft settled with AOL Time Warner (now Time Warner), paying the one-time browser rival $750 million and promising to cooperate on software distribution and digital media. Time Warner's Netscape Communications unit had filed the antitrust complaint in January 2002 alleging that by bundling its own browser into Windows, Microsoft used its dominance in the OS market to crush Netscape's browser business.\textsuperscript{41}


\textsuperscript{41} Karen Southwick and Charles Cooper, "Microsoft reveals a softer side," CNET News.com, Apr 2, 2004,
Sun, Microsoft and Sun Microsystems have settled antitrust and patent disputes and signed a 10 year technology-sharing agreements. The deal helps both vendors and promises benefits for customers. Microsoft will pay Sun $1.6 billion to settle their antitrust and intellectual property disputes, and Sun, the prime complainant in the antitrust case in European Union (EU) against Microsoft, will drop its case in the EU. The vendors will license each other's server technologies. Microsoft will pay Sun $350 million for technology licenses, and Sun will pay fees when it uses Microsoft technology.\(^{42}\)

Of all the settlements, the agreement with Sun is the most significant in the following ways. First, it allows Microsoft and Sun to share patents and proprietary information for better interoperability between Microsoft .Net and Java, and between Microsoft’s and Sun’s products.\(^{43}\)

Second, such kind of cross license technology-sharing framework allows engineers from both organizations to know the rules in which to start collaborative work and to subsequently build enough trust for deeper collaboration.

Lastly, and perhaps the most important, is that the agreement allows both party to tap on innovations outside the company boundary, and to license their intellectual property to others on reasonable terms with appropriate compensation for the investment made by the company.

The above settlements reflects the desire of Microsoft, with its large current asset of more than $50 billion, to settle its legal disputes and move on, instead of spending a lot of time and effort in legal battles, and being portrayed publicly as a bullying monopolist. It allows Microsoft to channel its energy to focus on the immediate challenges such as the threat pose by open-source.

Although Microsoft has managed to settle many of its disputes in the US, it still has some pending


disputes outside the US, the most prominent and important of which is the antitrust action by EU. The European Commission, with its ruling, fined Microsoft $613 million and ordered Microsoft to offer a version of its Windows operating system without its Media Player. On top of this, the European Commission went a step further than what the US antitrust settlement with Microsoft, which required the company to broaden its disclosure of desktop protocols to other server vendors, by calling for “full interoperability” of non-Microsoft servers with Windows PCs and servers.

In an interview with Lee Patch, vice president of legal affairs at Sun, he said that the ruling defines the level or degree of interoperability required of a monopolist. It sets full interoperability “as the standard for competition,” meaning that competition should be based on price, performance, stability and security that's "not to be distorted by the interoperability to the Microsoft desktop monopoly product."

Microsoft is appealing against the ruling. Microsoft’s chief legal counsel, Brad Smith, derided that aspect of the decision and said that by forcing the company to disclose server protocols, the European Commission is asking it to give away valuable parts of its intellectual property to its competitors.

Conclusion
The outcome of this case and the actual implications is still unclear as the European Commission has yet to disclose the specific requirements stating how far Microsoft must go. With the settlement with Sun to cooperate on interoperability, Microsoft will attempt to show antitrust attorney that there is a way for competitors to access its proprietary interfaces through licensing based on mutually agreed and reasonable terms. However, it remains to be seen what “reasonable” means to different party, and how Microsoft will treat its partners in future dealings.

Implications
# 10: Microsoft plans to continue bundling applications with Windows operating system arguing that it is for consumers’ benefits and it is what they want.
# 11: The courts will have to evaluate antitrust challenges on a case-by-case basis, trying to weigh

potential harm to competitors versus potential benefits to consumers.

# 12: Microsoft shows intention to resolve long-standing legal and regulatory disputes and move on to focus on impending challenges.

# 13: Through the agreement with Sun, there will be better interoperability between .NET and Java.

# 14: There will be more cooperation between Microsoft and Sun, to counteract the power of IBM (for example in setting standards in standards body).

# 15: The cross-licensing agreement with Sun signals Microsoft’s attempt to tap on innovations outside the company boundary.

# 16: There is a way for competitors to access Microsoft’s proprietary interfaces through licensing based on mutually agreed and “reasonable” terms.

4.3. Technological Trends

4.3.1. Interoperability through Web Services

Introduction
Web Services is an important set of technology and standards that has the potential to help deliver the long desired aim of achieving interoperability between different kinds of system. The essence of Web Services is described as:\footnote{Rupert Goodwins, "Everything you need to know about Web Services," ZDNet UK, Jul 21, 2003, accessed Apr 22, 2004.}

Web Services is a set of XML-based standards and programming guidelines that allow applications to swap data over the Internet. With it, you can take any aspect of your company’s data management and computing processes and link it to another division, another company, end users or whoever you like, pretty much independently of whatever systems they are using (can be either Microsoft .NET or J2EE or any other platform that comply with the Web Services standards). By mediating and automating the process of sharing data across the Net between heterogeneous systems, Web Services has successfully combined some of the most powerful ideas in the history of computing, and have done so in an open, non-proprietary way -- in theory and, to some extent, in practice.

Extensible Markup Language (XML) emerged in late 1996 as a way to create structured Web
documents that could contain arbitrary data in standard ways. By agreeing on schema — sets of rules in XML — two or more companies wishing to share data in machine-readable ways could do so over the Net, using standard tools. Besides XML, the other main protocols in Web Services are Simple Object Access Protocol (SOAP), Web Services Definition Language (WSDL), and Universal Description, Discovery, and Integration (UDDI). In simple terms, XML describes the data, UDDI says where services are, WSDL describes what they do and SOAP talks to them.

Understanding the concepts

A more elaborate description of the four main protocols (WSDL, UDDI, XML and SOAP) that currently define Web Services, and how they work together to accomplish a specific task is as follows.

- **WSDL (describe it).** WSDL is an XML dictionary for describing a Web Service, its functionality specifications, inputs, outputs, and accessible methods. By standardizing the way to describe remote Application Programming Interfaces (APIs), WSDL provides a new opportunity to automate how developers catalog, manage, and document application components.

- **UDDI (find it).** UDDI is a mechanism by which providers and consumers of Web Services can find each other. The UDDI project, which is a multi-company initiative, has resulted in a set of specifications by which registries can be created to match WSDL-described service requirements with providers of those services. The main focus of UDDI is on creating public registries, but many companies are now looking at UDDI as a way to generate internal, private registries.

- **XML (call it).** XML is being universally adopted as a language to define application-independent data structures and storage. With Web Services, XML provides a mechanism to both define the interface and include the parameters for a specific procedure call.

- **SOAP (sends and receives it).** SOAP is an XML messaging protocol that was developed by Microsoft and submitted to the W3C as an ongoing standard. As indicated by its name, SOAP is a language- and platform-independent protocol whose simplicity has eased its general adoption. It is the communication mechanism for delivering API calls and

---

Figure 4-3 depicts the process to link two processes through Web Service: (1) The Web Service Provider would describe the service it provides in WSDL specification and register with the UDDI registry; (2) The Consumer Service would search the UDDI registry for service providers that have services that matched its needs; (3) The UDDI registry would then return the pointer, which indicates the location, to the matching Web Service Provider; and (4) The Consumer Service would then communicate directly with the Web Service Provider to execute the required service. Thus, in theory, an application can be created by combining various Web Services registered in the registry to perform a complex task.

![Diagram of the process to link two processes through Web Service]

**Figure 4-3: Web Services Technologies**


**Conclusion**

Web Services has matured a long way since it first started with XML in 1996. More enterprise level adoption and deployment is expected with the establishment of other core standards such as Business Process Execution Language (BPEL) used to automate complex business processes, Web Services Security (WS-Security) for securing transaction between the two communicating systems, and WS-Reliable Messaging (WS-RM) to ensure completion of transaction between the systems.
Companies will start to look at how these technologies help to automate processes within the organization and with business partners.

**Implications**

# 17: Web Service enables interoperability between .NET and Java, allowing co-existence of the two differing technologies.

### 4.3.2. Service Oriented Architecture

**Introduction**

Service-oriented architecture (SOA) is a way of designing software to make it easier to automate business processes and share information between disparate systems. To construct an SOA, companies write applications as a collection of interoperable components that communicate using industry standards, such as Web Services. This modular application approach can be more cost-effective because it lets companies create a software function and reuse it in many instances throughout a corporation.\(^{47}\)

Although the concept has been around for decades, adoption of SOA is picking up for both business and technical reasons. Corporate customers are demanding more cost-efficient computing systems. The growing adoption of Web Services, a set of XML-based protocols for easily exchanging information between disparate systems, makes the appropriate tooling for SOA widely available as well.\(^{48}\)

Increasingly, corporations are coming to the same conclusion: SOA offers a promising design approach for making computing systems more flexible and cost-effective. More companies are choosing SOA as a way to redesign their technology foundation and improve interoperability

---


between disparate systems.\textsuperscript{49}

\textbf{Understanding SOA}

The fundamentals of SOA are described in the paper “Raising EAI standards”\textsuperscript{50}:

\textit{In a SOA model, the principle is that separate business operations or process steps are packaged as much as possible as self-contained components or ‘services’. This has strong similarities to the old object-oriented programming concepts, but, instead of the encapsulated component being a program or a piece of code, it is a set of pieces of functionality making up the service in question. In addition, an SOA model frequently operates on an event-driven basis, very much as a traditional business process does.}

\textit{The SOA approach has some fundamental advantages when applied to integration solutions:}

- It offers a high degree of reuse
- It contributes to raising the overall quality of service for the new solution
- It provides a highly flexible and adaptable implementation
- It allows IT operational flows to more closely reflect the way the business operates

Every business operation in an SOA environment is broken down into the basic set of services, which in turn form building blocks that can be used to assemble new solutions. The connectivity and integration of the components within these encapsulated services, and indeed of the services themselves, is provided by the Enterprise Application Integration (EAI) technology. Once the business services have been created, they can be viewed as self-contained black boxes with well-defined inputs and outputs. This is what delivers the flexibility, adaptability, quality and reuse. In an SOA environment it becomes a simple matter to switch a particular service for a new one as part of an overall business


\textsuperscript{50} Steve Craggs, “Raising EAI standards: Looking at the development of the EAI market and the effects of the emergence of standards,” EAI Industry Consortium, Apr 2003.
process. Indeed, because of the power of EAI, the new service might not even be internal but could be provided by a third party instead.

Developments around the SOA context over the last few years are proving very fruitful, and are already influencing the direction of the EAI market. Perhaps one of the most interesting ones is the concept of an Enterprise Service Bus (ESB) – an example of what Gartner Group calls an enterprise nervous system (ENS) backbone. This is the melding of SOA principles with EAI technology, taking advantage of developments in the broader standards area too, to produce an enterprise integration backbone throughout the value chain with services socketed onto it as required. These services can be running in any environment (J2EE, .NET, Legacy, Packaged application, etc.), either within the enterprise or in a partner company. Connectivity to these environments, presentation of data structures and navigation between the services are controlled by the EAI technology.

The diagram in Figure 4-4 illustrates the ESB concept:

![Diagram showing Enterprise Service Bus (ESB) concept with MOM facilities, transformation services, and content-based routing.]

**Figure 4-4: Enterprise Service Bus Concept**
Essentially, an ESB delivers a powerful, affordable, standards-based backbone throughout the enterprise and partner companies that smooths the operational path of the processes running the business and reduces the time, effort and cost of integrating the different components that underpin these process steps. One of the most powerful benefits that this type of approach can deliver is that it allows in-house development teams to build new applications that are already ‘integration enabled’ and can easily be socketed into the ESB as required. Not only does this save a lot of investment on expensive skills, but also it has a significant impact on time-to-market for new initiatives.

Industry support\textsuperscript{51}

The growing interest in the revitalized SOA enabled by Web Services has created a rush of interest from technology providers trying to position themselves as "SOA providers." Whoever becomes the preferred provider for SOA infrastructure software "has huge leverage within that customer," said Joanne Correia, an analyst at Gartner.

The next version of Windows, code-named Longhorn, will include connectivity software called Indigo, based entirely on Web Services. BEA in May 2004 is launching a program called Project Sierra to encourage customers to better demonstrate the business value of SOA, a design approach meant to bring greater flexibility to corporate computing system, and to use its Java tools and server software to build an SOA. IBM, too, has tools and a self-designated "Center of Excellence" dedicated to promoting SOAs and Web Services.

Conclusion\textsuperscript{52}

In reality, the latest products – essentially the latest generation of middleware and development tools – are still relatively immature. But marketing from large vendors is beginning to raise awareness, said Ron Schmelzer, an analyst at research company ZapThink. SOAs are still used primarily by leading-edge customers, relatively sophisticated consumers of technology. However, if implemented successfully, SOAs could break the tradition of costly, underperforming IT projects


\textsuperscript{52} Ibid.
and give businesses better tools for releasing new applications quickly.

**Implications**

# 18: More companies are choosing service-oriented architecture as a way to redesign their technology foundation and achieve higher level of reuse and flexibility which allows IT operational flows to more closely reflect the way the business operate.

# 19: Enterprise Service Bus allows integration of heterogeneous applications without building a single end-to-end package software.
5. Analytical Frameworks

5.1. Introduction

This thesis applies the findings of a wide body of research to its analysis of the APS software industry. This chapter summarizes each piece of prior research that will have a bearing on the analysis.

5.2. Technology Acquisition Strategy

This section presents a review of the Acquisition Strategy as described in the "Entering New Business: Selecting Strategies for Success" paper\(^5\) with the focus on growth through acquisitions.

Entry into new product-markets represents diversification for the existing firm with the purpose of achieving future growth and profitability. However, such endeavors are risky and often fail. Two basic strategic questions are thus posed: (1) Which product-markets should a corporation enter? and (2) How should the company enter these product-markets to avoid failure and maximize gain?

Entering of new business can be achieved through various mechanisms such as internal development, acquisition, joint venture, and minority investments of venture capital. Each of these mechanisms makes different demands of commitment and involvement from corporation. It is important to understand the relative benefits and costs of each entry mechanism, and when should each be used.

Some New Business Development Mechanisms

Internal Developments. Internal development exploits internal resources as a basis for establishing a new business. In a study by Biggadike on Fortune 500 companies that had used this approach in corporate diversification, he found that companies typically take eight years to generate a positive return on investment, and performance did not match that of a mature business until a period of ten to twelve years had elapsed\(^4\).


Acquisition. In contrast to internal development, acquisition can take weeks rather than years to execute and it offers a much lower initial cost of entry into a new business or industry. Salter and Weinhold point out that this is particularly true if the key parameters for success in the new business field are intangibles, such as patents, product image, or R&D skills, which may be difficult to duplication via internal developments within reasonable costs and time scales\textsuperscript{55}.

Educational Acquisitions. In a targeted small acquisition, the acquiring firm immediately obtains people familiar with the new business area. Staff acquired in this manner may even be used by the parent as a basis for redirecting a corporation’s primary product-market thrust. It is necessary to ensure that key people do not leave soon after the acquisition as a result of the removal of entrepreneurial incentives.

Optimum Entry Strategies
Roberts and Berry’s Familiarity Matrix provides a uniform framework for assessing the acquisitions in terms of their initial market and technology-relatedness to the acquiring company. The Familiarity Matrix, depicted in Figure 5-1, is constructed with the two axes of market and technical familiarity of the acquisitions relative to the acquirer, using the three measures of base, new familiar, and new unfamiliar on each axis. The alternative strategies shown in each of the nine blocks of Figure 5-1 reflects Roberts and Berry’s arguments for the most “appropriate” organizational options for efforts aimed at new business development that are characterized by the indicated degree of market and technical newness. Within the lower left ‘base/familiar’ sectors, acquisitions are assumed to be capable of adding depth to the acquirer’s existing skills and knowledge, with only incremental broadening. Empirical studies of acquisition strategies show that such ‘related acquisitions’ tend to become successful additions to the firm\textsuperscript{56}. In contrast, if they were located in the upper right ‘familiar/unfamiliar’ sectors, any acquisitions would be essentially unrelated to the acquirer’s core knowledge. While such acquisitions might have the potential of dramatically adding new skills and new market opportunities to the acquirer, the research literature


\textsuperscript{56} P Healy et al., “Which takeovers are profitable? Strategic or financial?”, Sloan Management Review, Summer 1997, p. 45-57.
indicates that such ‘unrelated acquisitions’ usually fail and are frequently divested.

<table>
<thead>
<tr>
<th>Market Factors</th>
<th>New Familiar</th>
<th>New Unfamiliar</th>
<th>Base</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-Joint Ventures</td>
<td>-Venture Capital</td>
<td>-Venture Capital</td>
</tr>
<tr>
<td></td>
<td>-Internal Market Developments</td>
<td>-Venture</td>
<td>-Venture</td>
</tr>
<tr>
<td></td>
<td>(Acquisitions)</td>
<td>Nurturing</td>
<td>Nurturing</td>
</tr>
<tr>
<td></td>
<td>(Joint Ventures)</td>
<td>-Educational</td>
<td>-Educational</td>
</tr>
<tr>
<td></td>
<td>-Internal Ventures</td>
<td>Acquisitions</td>
<td>Acquisitions</td>
</tr>
<tr>
<td></td>
<td>-Acquisitions</td>
<td>Licensing</td>
<td>Licensing</td>
</tr>
<tr>
<td></td>
<td>-Internal Product Developments</td>
<td>&quot;New Style&quot;</td>
<td>&quot;New Style&quot;</td>
</tr>
<tr>
<td></td>
<td>(Acquisitions)</td>
<td>Joint Ventures</td>
<td>Joint Ventures</td>
</tr>
</tbody>
</table>

Figures 5-1: Familiarity Matrix: Optimum Entry Strategies


Implications

# 20: Acquisition provides a much faster time-to-entry into new markets.

# 21: Educational acquisitions can be used as a basis for redirecting a corporation’s primary product-market thrust instead of a full scale acquisition.

# 22: Entry mechanisms requiring high corporate involvement should be reserved for new businesses with familiar market and technological relatedness.

5.3. Multiple Dimensions of Acquisition Implementation

This section presents a study on how the multiple dimensions of acquisition implementation affect the successful appropriation of technologies and capabilities by the acquirer.
Gaining new technologies and capabilities through acquisition is likely to be a challenging task. Problems with post-acquisition integration are oft-cited reason for the failure of many acquisitions. In the study by Lord and Ranft\textsuperscript{57}, they developed an empirically grounded model of technology and capability transfer during acquisition implementation. They assess how the nature of the acquired firms' knowledge-based resources, as well as multiple dimensions of acquisition implementation, have both independent and interactive effects on the successful appropriation of technologies and capabilities by the acquirer. The model provides a consistent framework for assessing the degree of success of the acquisitions.

The proposed model presented in Figure 5-2 has five major components: (1) the nature of the underlying knowledge, (2) multiple dimensions of the acquisition implementation process, (3) the acquisition context, (4) management practices, and (5) the transfer of technologies and capabilities to the acquiring firm.

Some of the propositions put forth by the authors are:

1. During acquisition implementation, greater autonomy facilitates the preservation of an acquired firm's tacit and/or socially complex knowledge;
2. During acquisition implementation, greater autonomy inhibits the transfer of the acquired firm's technologies and capabilities that are based on tacit and/or socially complex knowledge;
3. During acquisition implementation, rich communications facilitate the preservation of the acquired firm's technologies and capabilities by creating a more favorable climate between the two organizations;
4. Slow acquisition implementation is (a) positively associated with the preservation of tacit and/or socially complex knowledge, but is (b) curvilinearly (inverted U shape) associated with the transfer of these technologies and capabilities to the acquirer;
5. During acquisition implementation, retention of key acquired employees facilitates the preservation of the acquired firm's technologies and capabilities that are based on tacit and/or socially complex knowledge; and

(6) During acquisition implementation, (a) greater tacitness and/or social complexity of knowledge underlying an acquired firm's technologies and capabilities is positively associated with the use of financial incentives aimed at retention, and (b) use of such financial incentives enhances retention of key acquired employees.

Figure 5-2: Acquiring New Technologies and Capabilities: An Expanded Model of Acquisition Implementation


Implications

# 23: Greater autonomy facilitates preservation of an acquired firm’s knowledge but inhibits the transfer of the knowledge to the acquirer.

# 24: Slow acquisition implementation facilitates preservation of an acquired firm’s knowledge but inhibits the transfer of the knowledge to the acquirer.
5.4. Ecosystem Driven Growth Strategy through Platform Leadership

In the book “Platform Leadership”\textsuperscript{58}, Gawer and Cusumano describe how the best high-tech firms such as Intel, Microsoft and Cisco win by driving industry innovation, and establish dominant market positions in their respective industries by becoming platform leaders\textsuperscript{59}. Successful platform leaders can expect huge financial reward and at the same time exert strong platform influence over the direction of innovation in their industries and thus over the network of firms and customers participating in the “ecosystem”. However, not all industries are suitable for platform leadership strategies. The fundamental condition is that the firm’s product has limited value when used alone but gains in value when used along with complements. Once a platform leader wannabe recognized this, it can then works and creates a strategy to deal with issues such as how to encourage complementary innovation and how to evolve the platform interfaces and architecture. The aim is to collaborate and achieve win-win scenarios for the platform leader, complementors, and customers.

Gawer and Cusumano present their Four Levers Framework for designing and implementing a successful platform strategy:

1. \textit{Determine the scope of the firm}: Is it preferable to create product complements internally or let the “market” produce them?

2. \textit{Design product technology strategically}: What degree of modularity is appropriate? Should product interfaces be open or closed? What information should leaders disclose to outside firms?

3. \textit{Shape relationships with external complementors}: How can the company balance competition and collaboration with outside players?

4. \textit{Optimize internal organization structures}: What processes and systems will allow the company to manage internal and external conflicts of interest most effectively.


\textsuperscript{59} Companies that provide the technological foundation on which other products, services, and systems are built.
(1) *Determine the scope of the firm*

Platform leaders depend on the ecosystem or network of innovation to produce complements so as to increase the value of the platform. Determining the scope of the firm – in other words, what complements to make inside and what to leave to external firms – is the most important decision that platform leaders have to make. One of the important determinants in this decision is the capabilities of the firm – whether the platform leader has the technical and organizational skills or the financial capacity to compete in the relevant markets.

The other important lever that platform leaders can exercise involves how platform leaders can stimulate external innovation, and how they can choose appropriate levels of investment and involvement in venture capital activities or acquisitions aimed at evolving the platform or helping the complements business. One of the proposed ways for the platform leaders to encourage companies to come on board of the platform is the “rabbit” strategy where a promising complement from the target industry is selected and supported in a very visible way so that its success can encourage other firms to follow. The rabbit strategy also signals to the world that the platform leader intends to stay out of this complementary market while encouraging competition in that market.

The experience of Intel and the like suggests that even if a firm decides to rely on external developers for complements, it should maintain the in-house capability to enter the complementary areas in future as it provides three possible benefits:

(a) *the expertise necessary to define the interface between its product and components;*

(b) *the ability to enter complementary market directly, should outside firms decline to enter or fail in the attempt; and*

(c) *some bargaining power with complementors.*

The use of venture investments or merger and acquisitions are mechanisms to change the scope of the firm and enable it to influence the actions of complements. However, acquisitions may have side effects and result in conflict of interests as the platform leader now becomes a competitor in the marketplace of former partners.

(2) *Design product technology strategically*

*Architecture*

*Platform architecture – the high-level design of the system and the interface designs that determine*
how components or subsystems work together – can have a profound and lasting impact on the structure of an industry and the nature of innovation. Modular designs reduce the costs of innovation and encourages the investment and development of creative complements by niche players.

**Interfaces**

Open interfaces encourages third-party innovation. Moreover, the large number of complementors integrated through these interfaces provides a huge barrier of entry for potential competitors that are considering introducing a competing architecture with different interfaces. On top of providing better product architecture, the potential competitor would have to convince the complementors to adapt to their design which could incur huge switching costs for the complementors.

**Intellectual Property (IP)**

Most platform leaders are very careful not to release internal architectural information about their product which is their core competence and provides the competitive advantage. However, they have willingly disclosed information regarding external interfaces, as this encourages external innovation. The decision in charging royalties on the use of the external interface specifications provided by the platform leaders will determine the rate of diffusion of the platform and interface standard. Intel made an early decision to relinquish royalties on the interface IP so that the industry could adopt without huge concerns about infringing IP or having to pay high royalties and this policy encourages the adoption of the standards proliferated by Intel.

(3) Shape relationships with external complementors

One of the tasks of the platform leaders is to work with both established complementors and new potential complementors, and helping the latter to grow and assisting them with technical issues.

**Consensus and control**

To be successful, platform leaders need to obtain consensus among key complementors on the interface technical specification and standards and also try to maintain control over those critical design decisions made by the complementors that may affect future interoperability of the new product generations.

**Collaboration and competition**

To ensure success, it is important that the platform leaders play multiple roles:
(a) driving architectural, systemic innovations;
(b) stimulating innovation on complements; and
(c) coordinating.

The first role is of a firm deeply concerned with the common good, in other words, with expanding the “pie” for everyone. The second role is that of industry enabler. At certain times, platform leaders have to make tradeoffs and sacrifices against their own interest in favor of the common good. A third role that platform leaders must play is similar to the classical position of firms that pursue their own best interest, such as to maximize their own profits or market share.

The platform leader need to play this coordinating and balancing role as their incentives are not always align with the complementors for several reasons. Firstly, in cases where the complementors bear large proportion of the risks, the authors believe that the platform leader should try to share the risks with potential complementors. Secondly, there is a conflict of interest due to the existence of the dual role of both a partner and a competitor when the platform leader decides to enter complementary markets due to reasons cited earlier under the benefits of maintaining in-house capability. However, such actions might erode the trust of complementors and affect future collaborations. As such, platform leaders have to be extremely careful with decisions that extend the functional scope of their products or extend their lines of business.

(4) Optimize internal organization structures
The problem of platform leaders is usually how to compete and collaborate simultaneously where some groups might compete with complementors while others might try to encourage those same complementors to cooperate and adopt the platform’s technical standards.

Structure
The authors believe that these internal tension must, and can, be managed by having a clear separation of units charged with achieving different goals such as stimulating innovation by complementors (e.g. R&D lab) versus investing directly in complementary markets (e.g. the product group). The separation facilitates trust with external partners.

“System” mindset and neutrality
Based on the various case studies, the authors noted that platform leaders had a better chance of
success if the employees perceive the platform as a system and have a vision for how to evolve that system. Being neutral is also an important attribute as their job often involves acting as a neutral broker between external party and internal group in co-development tasks.

Gawer and Cusumano succinctly summarize the essence of platform leadership:

*In conclusion, we can say that the essence of platform leadership begins with a vision that extends well beyond the business operations of one firm or the technical specifications of one product or one component. It is a vision that says the whole of the ecosystem can be greater than the sum of its parts, if firms work together and follow a leader. The vision and decisions of platform leaders can affect not only the immediate competitive environment but also the evolution of technologies and entire industries... The decisions they make and do not make, can greatly influence the degree and kind of innovations that complements create.*

The authors also highlighted three closely related issues of the platform leadership strategy. First, even successful platform leaders can fall prey to problems that arise from too much of a platform centric mentality. There is the danger for the platform leaders becoming entrenched in on one vision of an industry and be blindsided by the emergence of radical changes in the environment or product line as in the case of Microsoft being caught off guard by the emergence of Netscape’s Internet browser that once threatened its operating system platform leadership.

Second, platform leaders can become so tied to certain technologies that they find it difficult to evolve their platforms. In industries where technology is rapidly evolving, evolution is often important to long-term survival. A dominant position in one technology is more likely to constrain a firm’s ability or its incentives to invest in a new ecosystem that can threaten current sales and profits.

And third, there may be some confusion as to whether market leadership is a necessary condition of platform leadership. It is usually the case that platform leaders first establish its market leadership before attempting to evolve into being the technical leader for its platform. The authors highlighted that although a high market share and a high degree of innovative capabilities are important and necessary conditions to achieve platform leadership, they alone do not suffice. The company has to be able to engage complementors and place itself in the center of the network of innovation around its platform.
Implications

# 25: One of the factors in determining the scope of the firm is the capabilities (technical, organizational and financial) of the firm.

# 26: One of the ways to encourage companies to come on board of the platform is the “rabbit” strategy where a promising complement from the target industry is selected and supported in a very visible way so that its success can encourage other firms to follow.

# 27: Changing the scope of the firm by entering complementary markets may have side effects and result in conflict of interests as the platform leader now becomes a competitor in the marketplace of former partners. Such actions might erode the trust of complementors and affect future collaborations.

# 28: Modular designs reduce the costs of innovation and encourages the investment and development of creative complements by niche players.

# 29: A large number of complementors provides a huge barrier of entry for potential competitors that are considering introducing a competing architecture with different interfaces.

# 30: The decision in charging royalties on the use of the external interface specifications provided by the platform leaders will determine the rate of diffusion of the platform and interface standard.

# 31: Having a clear separation of units charged with achieving different goals such as stimulating innovation by complementors versus investing directly in complementary markets facilitates trust with external partners.

# 32: There is the danger for the platform leaders becoming entrenched in on one vision of an industry and be blindsided by the emergence of radical changes in the environment or product line.

# 33: Platform leaders can become so tied to certain technologies in order to sustain current sales and profits that they find it difficult to evolve their platforms which is essential for long-term survival.

5.5. Dynamics of Industry - Evolution and disintegration and reintegration of industry

Fine⁶⁰ hypothesized that industry evolution follows a model based on an infinite double loop that cycles between vertically integrated industries inhabited by large companies and horizontally

---

integrated industries populated by many innovative companies (see Figure 5-3).

His idea to study fruit fly companies originated from the use of fruit flies, which has similar genetic structure similar to that of humans but has a short life span of less than two weeks, in the study of genetic changes. He reasoned that the study of fruit fly companies might enable us to study the dynamics of business in industries similar to that of the fruit fly company. And that the lessons drawn from examining the experience of the fast clockspeed\(^6\) company can be applied to others, even those with medium or slow clockspeeds.

![Diagram: The Double Helix, Illustrating How Industry/Product Structure Evolve from Vertical/Integral to Horizontal/Modular, and Back](image)

**Figure 5-3: The Double Helix, Illustrating How Industry/Product Structure Evolve from Vertical/Integral to Horizontal/Modular, and Back**


One of the most information-rich fruit flies is the PC industry – the story of how IBM’s decision to outsource the microprocessor and operating system to Intel and Microsoft respectively – where the

\(^6\) The rate at which the industry evolve.
lion's share of profits and industry clout flowed from IBM to these hand-picked suppliers. The lesson learnt on the importance of not to outsource core competencies can be applied to slower clockspeed industries such as automobiles where the electronic components and systems suppliers are increasingly gaining power and capturing larger portion of the profit pool.

Fine hypothesized that all competitive advantage is temporary. He emphasized that while exploiting current capabilities and competitive advantages, there is a need for companies to build new capabilities for the inevitable moment when the old ones no longer provide an advantage. Even Microsoft which appeared to have an absolute lock-in based on the company's proprietary technological standard in PC operating system was blindsided by the emergence of Internet browser and the upstart Netscape, which threatened the future of the traditional operating system. Microsoft had to build up new capabilities in Internet browser to respond to the attack by Netscape. Thus, the ability to analyze and anticipate where the lucrative opportunities will arise and investing in the capabilities and relationships to exploit them differentiates the top-performing companies from the rest.

He further hypothesized that the analyses of the capability chains of the companies helps us to understand their mutation, evolution, and eventual survival or demise, and that the business genetics is the industrial equivalent of the double helix – a model based on an infinite double loop that cycles between vertically integrated industries inhabited by large companies and horizontally integrated industries populated by many innovative companies seeking to grab a niche in the many opportunities that emerged with the demise of the giants.

The rationale for this model is explained by the following forces of disintegration and integration in the vertically and horizontally integrated industries respectively.

*When the industry structure is vertical and the product architecture is integral, the forces of disintegration push toward a horizontal and modular configuration. These forces include:*

- The relentless entry of niche competitors hoping to pick off discrete industry segments;
- The challenge of keeping ahead of the competition across the many dimensions of technology and markets required by an integral system; and
- The bureaucratic and organizational rigidities that often settle upon large, established companies. These forces typically weaken the vertical giant and create pressure toward disintegration to a more horizontal, modular structure.

On the other hand, when an industry has a horizontal structure, another set of forces push toward more vertical integration and integral product architectures. These forces include:

- Technical advances in one subsystem can make that the scarce commodity in the chain, giving market power to its owner;
- Market power in one subsystem encourages bundling with other subsystems to increase control and add more value; and
- Market power in one subsystem encourages engineering integration with other subsystems to develop proprietary integral solutions to increase profitability.

**Implications**

# 34: Industry structure evolves in an infinite double loop that cycles between vertically integrated industries and horizontally integrated industries.

# 35: There is a need for companies to build new capabilities for the inevitable moment when the old ones no longer provide an advantage.

### 5.6. Closed and Open Innovation Paradigm

This section describes the innovation paradigm shift that is occurring in many companies in order to monetize both internal and external innovations.

John Seely Brown, Director Emeritus, Xerox Palo Alto Research Center, in the foreword of the book “Open Innovation”\(^62\)” described the difficulty of sustaining the generation of innovative ideas:

> I still find it amazing just how difficult innovation\(^63\) continuous to be. But today we are


\(^63\) Brown describes innovation as invention implemented and taken to market.
faced with the extra problem that our ideas of innovation have gone stale. So we need to be innovative in the area of innovation itself, which is what this book will help us to do and what I mean by calling this foreword “Innovating Innovation.”

Brown explains that the Open Innovation model that the author, Chesbrough, describes shows the necessity of letting ideas both flow out of the corporation in order to find better sites for the monetization, and flow into the corporation as new offerings and new business models. It is important to find the right balance and mechanisms to enable the flow of ideas. Brown further explains it is much more risky for a rising corporate star to champion a risky innovation that ends up failing, than squelching one that could have been a winner. The Open Innovation model diminishes both the error of squelching a winner and that of backing a loser, and moves us closer to a world where protective moves and face-saving mechanisms no longer cause potentially great innovations to be shelved.

The author describes that in the Closed Innovation Paradigm, it implicitly assumes that all R&D activities are conducted within the firm. However, the new approach of Open Innovation Paradigm assumes that valuable ideas can come from inside or outside the company and can go to market from inside or outside the company as well. This approach places external ideas and external paths to market on the same level of importance as that reserved for internal ideas and paths to market during the Closed Innovation era.

The importance of architecture
In the Open Innovation paradigm, there remains a critical role for internal R&D: the definition of an architecture\textsuperscript{64} to organize the many subcomponents into a new useful system. In the early life of a promising new technology, the complexities of the new approach create many ambiguities about how best to incorporate it into the larger system to reduce its overall complexity and improve performance. Relying completely on external technologies to determine the optimal interconnections in such uncertain and complex circumstances is difficult as the companies making these technologies will all differ on the best way to utilize their technology. Moving the resolution of the interconnection problem within the firm allows the firm to bypass the possible holdup tactics by outside companies who may perceive that their component is a key part of the system based on

\textsuperscript{64} A hierarchy of connections between disparate systems.
how they would have defined the relationships among its parts.

The author emphasized that as challenging as that is, crafting connections between technologies inside a system is only a portion of the task of the innovating firm because companies’ architectures also have powerful implications for how the value chain and surrounding ecosystem will be structured. A valuable architecture not only creates opportunities for others to contribute their expertise to the system but also reserves opportunities for the firm to own a piece of the chain for itself and profit from its research contribution to the creation of the new system.

Over time, as the technology matures, interdependencies become clearer and more manageable, and the dominant design\(^{65}\) emerges. The vertical character of technological competition, in which internal R&D was critical to reduce complexity and achieve optimal performance, gives way to a more horizontal phase of technological competition, in which external technologies compete within the partitions of an established architecture. In fact, the firm has established itself as the platform leader described in Section 5.4 on Platform Leadership.

**Implications**

\# 36: To maximize monetization of ideas, there is an innovation paradigm shift from Closed to Open Innovation.

\# 37: Even in the Open Innovation era, internal R&D is central to retaining a portion of the value created by the research contribution of the firm and the activities of external firms.

**5.7. Product Platform and the Architecture of Software Products**

Many companies fail to make the difficult transition from one generation of technology to the next and sustain their excellence through eras of changing technology and leadership. This section

\(^{65}\) A dominant design is, by definition, the one that wins the allegiance of the marketplace, the one that competitors and innovators must adhere to if they hope to command a significant market following (Utterback, J.M., “Mastering the Dynamics of Innovation”, Harvard Business School Press, Boson, Massachusetts, 1994).
summarizes the product platform approach to product development described in the book “The Power of Product Platforms”\textsuperscript{66} as a mechanism to sustain technological leadership. Building robust product platforms can be a market domination strategy and can provide access to new markets.

**Focusing on single products**

The long-term success of an enterprise depends on a stream of new successful products to replace old ones and to enter new markets. However, traditional methods for managing this vital business function usually fail to deliver in the long run or are highly costly for the following two reasons. First, every single product must compete for resources against other projects in the corporation’s portfolio, and the return on investments, time-to-market and NPVs are typically calculated based on single product.

Second, the end result of this single-product focus is a failure to embrace commonality, compatibility, standardization, or modularization among different products and product lines. This results in a lack of commonality in the products themselves and their component parts making the use of the product complex and cumbersome.

**Product platform approach**

The author proposed that a better way of product development is to build an entire family of products that leverage a common market understanding, common product technologies, and a common set of highly automated production processes. Rather than have separate development teams each working on single products, it would be better to have them join forces in building a common platform from which a host of derivative products could be effectively and efficiently created. The author calls this foundation of core technology the “product platform” which is a set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently developed and produced.

**Challenges of sustaining consistent platform strategy**

Succeeding in one generation of product family once is not sufficient for the long term. Product platforms need to be rejuvenated continuously to incorporate new functions and newly established

standards, embrace technological changes as they occur and making each new generation of a product line more exciting and value-rich than its predecessors.

However, new product strategy is as much a mindset as a process, a fact often lost as small companies grow larger over time. Some of the challenges identified by the author are:

- As growth continues, product portfolios expand and the company may diversify into new technologies and markets. To improve efficiency, a functional division of labor may also take form, where R&D, manufacturing, sales, services and other functions find themselves in separate departments, in separate buildings, and often in different cities.

- Diversification is often unplanned where products and product lines are added one product at a time and without the benefit of overarching strategic principles. Typically, the overall system complexity increases resulting in greater complexity in manufacturing, procurement, and distribution, and the attention of senior managers is consumed in managing that complexity. Once senior managers lose touch with the development of new products, new product strategy is guided by the initiatives of middle managers. The notion of a centrally rational product strategy guided by the top executives gets lost in the bureaucracy.

- Relying on acquisitions as a primary source for introducing innovation to the firm is insufficient; innovation can and should come from within. The mindset also demands that different parts of the corporation need to consider how they can work more closely together – designing their respective product lines on common platform and manufacturing technology.

- Senior managers sense their feel for the business slipping away and respond with various processes to maintain the semblance of order and control: approval committees, stage-gates, and procedural rules of various types. Unfortunately, many such measures merely reinforce the bureaucracy and make the important business of product development slower and more difficult. In such an environment it is very difficult for executives, entrepreneurial middle managers, or innovative employees to marshal the energy and resources needed to introduce common subsystems and manufacturing processes across the corporation.

The architecture of software products
Besides applying the product platform development approach to tangible products, the author believed that the same benefits can be derived for intangible products such as software:

*A software product family can provide great choice and variety to the user by providing users with customized modules that can be incorporated into the software engine at no additional integration cost. This approach is presented in Figure 5-4. The software platform is the design and implementation of the engine, a core set of programs that propel the entire system. Modules are developed and plugged into the engine to provide variety, either as new versions of existing products or as entirely new derivative products. Thus each new derivative product represents either a viable upgrade or a new product purchase for the customer.*

![Diagram of Add-in Modules Creating Derivative Products and Operating System Services]

**Figure 5-4: Generalized Architecture of a Software Product Platform**

Managing the evolution of a software products family

In a product platform approach, the entire product family can be upgraded to a new computing environment without having to substantially reprogram all the various add-in modules that were developed for an older platform – provided a clear and robust interface between the underlying engine and the add-in modules has been developed and sustained. R&D efforts can then be focused at the platform level when migrating the software to a new operating system or networking environment or to use a new set of software development tools.

The author also highlighted how a platform strategy for software can also have compelling benefits to end users. Typically, end users will develop their custom applications on top of the add-in modules described above. If those add-in modules are clearly separated from the underlying software engine, they can be carried forward across generations of the product platform, often without radical revision. Thus, end users can be shielded to a significant extent from changes in the underlying technology as it evolves over time.

The future of software development

There are several implications of product platform strategy on the future of software development. Some of the recommendations put forth by the author are:

- *In today's constantly changing business and technological environment, the platform approach to software development is the best approach for keeping pace. Software products are simply too complex to develop on a piecemeal basis. It is far more efficient to develop the underlying platforms and to launch derivative applications from them. The firm can then renew its software product platforms while it is creating derivative products from existing platforms.*

- *Common interfaces, rather than myriad unique interfaces, are essential for maintaining a fast pace of renewal. That is all the more important at a time when alliances between software companies are becoming commonplace. For those alliances to work well, their products must also work together. Achieving that with some degree of elegance requires a high degree of clarity and robustness in the interfaces between a company's software and systems to be combined with it. Indeed, design and development of software products boils down to questions about how firms can best manage the interfaces within their software architectures.*

- *The product family concept as we have applied it here to software provides a powerful differentiation between new generations of software products and those embodied in legacy systems of the past. These ideas should not only guide the developers of new systems but also*
help corporate and individual consumers better consider which particular package or system will best suit their needs not only today but in the future.

Implications

# 38: In today’s constantly changing business and technological environment, the platform approach to software development is the best approach for keeping pace with new functional requirements, newly established standards and new technologies.

# 39: Unplanned or over diversification increases complexity and redirects senior managers’ attention to manage that complexity resulting in the dilution of the centrally rational product strategy.

# 40: Firms relying on acquisitions as a primary source for introducing innovation to the firm still requires internal collaboration to align and integrate the myriad products on common platform and manufacturing technology.

# 41: In the event that the product portfolio environment has become overly complex and senior managers has lost control of the product development strategy, it is very difficult for executives, entrepreneurial middle managers, or innovative employees to marshal the energy and resources needed to introduce common subsystems and manufacturing processes across the corporation.

# 42: In a world where alliances between software companies are common, having a robust, clear and common interfaces is essential for smooth migration to new operating system or networking environment, or to use a new set of software development tools.

# 43: A platform strategy shields end users to a significant extent from changes in the underlying technology as it evolves over time and allows custom systems built to be independent of the operating system.

# 44: The product family concept provides a powerful differentiation for new generations of software products and can help corporate and individual consumers better select package or system that suit their needs not only today but in the future.

5.8. Open source innovation

Software innovation through open-source projects as horizontal innovation networks has been studied by Von Hippel (2002)⁶⁷. In this study, the author describes how innovation development,

---

production, distribution and consumption networks can be built up horizontally – with actors consisting only of innovation users (more precisely, “user/self-manufacturers”) – and that user innovation networks can function entirely independently of manufacturers when (1) at least some users have sufficient incentive to innovate, (2) at least some users have an incentive to voluntarily reveal their innovations, and (3) diffusion of innovations by users is low cost and can compete with commercial production and distribution. When only the first two conditions hold, a pattern of user innovation and trial and improvement will occur within user networks, followed by commercial manufacture and distribution of innovations that prove to be of general interest.

Some of the great advantages highlighted by the author of horizontal user innovation networks over the manufacturer-centric innovation development systems are:

- They enable each using entity, whether an individual or a corporation, to develop exactly what it wants rather than being restricted to available marketplace choices or relying on a specific manufacturer to act as its (often very imperfect) agent.
- Individual users do not have to develop everything they need on their own: they can benefit from innovations developed by others and freely shared within and beyond the user network.

**Economics of innovation by users**

The author then states that research on innovation-related incentives and capabilities provides a theoretical basis for the empirical observations regarding innovation by users in general and by lead users in particular. Some of the key findings highlighted by the author are:

- With respect to innovation by users rather than manufacturers, it has been shown that in some product categories users may reasonably expect a higher reward from innovating than can manufacturers. He cited the example of IBM, where IBM may profit from developing improvements to the open-source program GNU/Linux, if these improvements enhance Linux’s functioning with a complementary good, such as proprietary software or hardware, that IBM does sell.
- User innovation costs can be significantly lower than manufacturer innovations costs when the problem-solving work of innovation developers requires access to “sticky”- costly to transfer – information regarding user needs and the context of use. Such information is located predominantly at user sites and can be mostly cheaply accessed by problem-solvers
located at those sites. The concentration of innovation activity among lead users within the user population can also be understood from an economic perspective. Given that innovation is an economically motivated activity (Schmookler 1966, Mansfield 1968), those users expecting significantly higher economic or personal benefit from developing an innovation – one of the two characteristics of lead users – have a higher incentive to and so are more likely to innovate. Also, given that lead users experience needs in advance of the build of a target market, the nature, risks, and eventual size of that target market are often not clear to manufacturers. This lack of clarity can reduce manufacturers’ incentives to innovate, and increase the likelihood that lead users will be the first to develop their own innovative solutions for needs that later prove to represent mainstream market demand. 

- In the specific instance of open-source software, software users can profit by using the software improvements that they develop. In contrast, there is no commercial market for open-source software – because open-source software developers make their innovations freely available as a public good. This eliminates manufacturers’ direct path – not ignoring the indirect path – to appropriating returns from private investment in developing open-source products, and so often eliminates their incentive to innovate in this arena.

Sources of benefit from free revealing
The author then discusses about the previous research explored by Harhoff et al. (2000) on the benefits that an innovator may derive from freely revealing an innovation. A brief overview of the

---


research on this topic, abstracted from Harhoff et al, begins with Allen (1983)\(^\text{72}\). Allen proposed that such behavior could be economically justified by profit-seeking firms on several grounds:

\begin{itemize}
  \item[(1)] gains in reputation for the firm or firm manager are sufficient to offset a reduction in firm operating profits caused by free revealing;
  \item[(2)] so many people know the information that it could not have been kept secret in any case;
  \item[(3)] the innovation is to some extent specific to the innovator and so free riders would not gain advantage equal to that of the innovator;
  \item[(4)] gains in the value of assets complementary to the user or production of the innovation exceed losses associated with free revealing;
  \item[(5)] free revealing may increase the innovator’s profit by enlarging the overall market for the product under consideration.
\end{itemize}

Harhoff et al (2000) also proposes that an innovation that is freely revealed and adopted by others can become an informal standard that may preempt the development and/or commercialization of other versions of the innovation. If the innovation that is revealed is designed in a way that it is especially appropriate to conditions unique to the innovator, this can result in creating a permanent source of advantage for that innovator. Note that being first to reveal a given type of innovation increases a user’s chances of having its innovation widely adopted, other things being equal. This may induce innovators to race to reveal first.

**Implications**

\# 45: For user innovation networks to function entirely independently of manufacturers, at least some users have sufficient incentive to innovate.

\# 46: For user innovation networks to function entirely independently of manufacturers, at least some users have an incentive to voluntarily reveal their innovations.

\# 47: In some product categories users may reasonably expect a higher reward from innovating than can manufacturers.

\# 48: User innovation costs can be significantly lower than manufacturer innovations costs when the problem-solving work of innovation developers requires access to “sticky” information regarding user needs and the context of use.

\# 49: The nature, risks, and eventual size of the target market for products that meet lead user needs

are often not clear which reduced manufacturers' incentives to innovate, and increase the
likelihood that lead users will be the first to develop their own innovative solutions for needs that
may later prove to represent mainstream market demand.

# 50: There is no direct path - not ignoring the indirect path - to appropriating returns from private
investment in developing open-source products since open-source developers make their
innovations freely available as a public good and this eliminates the manufacturers’ incentives to
innovate in this arena.

# 51: The market for APS software is huge and therefore great motivations exist for manufacturers
to innovate in this arena.

5.9. Technology Diffusion – Crossing the Chasm

In Moore’s Technology Adoption Life Cycle\textsuperscript{73} depicted in Figure 5-5, the types of consumers the
products attract throughout its useful life are different. The model says that the way to develop a
high tech market is to work the curve left to right, focusing first on the innovators, growing that
market, then moving on to the early adopters, followed by early majority, late majority, and even to
the laggards. In this effort, companies must use each “captured” group as a reference base for going
on to market to the next group. Innovators and early adopters are people who find it easy to
imagine, understand, and appreciate the benefits of a new technology, and to relate these potential
benefits to their other concerns. They typically buy into new product concepts very early in their
life cycle. The rest of the segments are driven by a strong sense of practicality and tend to adopt
products at a much later stage or not buy at all.

Figure 5-5: The Technology Adoption Life Cycle


Each group in the technology adoption life cycle represents a unique psychographic profile – a combination of the psychology and demographics that makes its marketing responses different from those of the other groups. Understanding each profile and its relationship to its neighbors is a critical success factor of high-tech marketing.

The chasm
To cross the chasm, it is especially important to understand the profile of innovator, early adopters and early majority. Innovators pursue new technology products aggressively. Early adopters (the visionaries), like innovators, buy into new product concepts very early in their life cycle, but unlike innovators, they are not technologists. Rather they are people who find it easy to imagine, understand, and appreciate the benefits of a new technology, and to relate these potential benefits to their other concerns. The early majority (the pragmatists) shares some of the early adopter’s ability to relate to technology, but ultimately is driven by a strong sense of practicality. They want to see well-established references before investing substantially. This pragmatist segment consists of about one-third of the whole adoption life cycle and winning their business is necessary for any substantial profits and growth.

75
The author further describes that between any two psychographic groups there are gaps which symbolize the dissociation between adjacent groups – that is, the difficulty any group will have in accepting a new product if it is presented in the same way as it was to the group to its immediate left. Out of all the gaps, there is the deep and dividing chasm that separates the early adopter from the early majority. The early adopter is some kind of change agent. By being the first to implement this change in their industry, the early adopters expect to gain some business advantage. They expect a radical discontinuity between the old ways and the new, and they are prepared to champion this cause against entrenched resistance. By contrast, the early majority want to buy a productivity improvement for existing operations. They do not want to be pioneer, and who have learned the hard way that the “leading edge” of technology is all too often the “bleeding edge.” They are looking to minimize the discontinuity with the old ways. By the time they adopt it, they want it to work properly and to integrate appropriately with their existing technology base. Because of these incompatibilities, early adopters do not make good references for the early majority. And because of the early majority’s concern not to disrupt their organizations, good references are critical to their buying decisions. So what we have here is a catch-22. The only suitable reference for an early majority customer, it turns out, is another member of the early majority, but no upstanding member of the early majority will buy without first having consulted with several suitable references.

Thus, the biggest problem during this transition period is the lack of a customer base that can be referenced at the time of making the transition into a new segment. To cross the chasm, it is recommended by Moore to target a very specific niche market where you can dominate from the outset, force your competitors out of that market niche, and then use it as a beachhead for broader operations. Concentrate an overwhelmingly superior force on a highly focused target. The key to this strategy is that it allows the company to win over pragmatist customers in advance of broader market acceptance by focusing an overabundance of support into a confined market niche. By simplifying the initial challenge, the enterprise can efficiently develop a solid base of references. The efficiency of the marketing process, at this point, is a function of the “boundedness” of the market segment being addressed. The more tightly bound it is, the easier it is to create and introduce messages into it, and the faster these messages travel by word of mouth. One of the keys in breaking in a new market is to establish a strong word-of-mouth reputation among buyers.

The pragmatists and the “whole” product

76
The pragmatists market segment is an important one though it requires much effort to win over. They are loyal once won, and often establish and enforce a company standard to use only your product for a given requirement which increases subsequent sales volume and lower future cost of sales. Pragmatists care about the company they are buying from, the quality of the product, the infrastructure of support and the reliability of the service they are going to get – they are planning to live with the decision personally for a long time to come. Because pragmatists are in it for the long haul, and because they control the bulk of the dollars in the marketplace, the rewards for building relationships of trust with them are very much worth the effort.

One of the key product characteristics that pragmatists value is the availability of whole product from the outset. The concept of whole product is as follows:

*There is a gap between the marketing promise made to the customer – the compelling value proposition – and the ability of the shipped product to fulfill that promise. For that gap to be overcome, the product must be augmented by a variety of services and ancillary products to become the whole product.*

At the introduction of any new type of product, competition focuses at the level of the generic product. But as the market matures, the generic product become more and more alike, and the value-add increasingly comes from augmented services and ancillary products. Investing in additional R&D at the generic level has a decreasing return compared with investments at the levels of the other elements of the whole product. The author summarizes that to cross the chasm and win over the skeptical pragmatists, marketing focus has to shift from product-centric value attributes to market-centric ones (refer to Table 5-1 for the representative list of each).

<table>
<thead>
<tr>
<th>Product-Centric</th>
<th>Market-Centric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fastest product</td>
<td>Largest installed base</td>
</tr>
<tr>
<td>Easiest to use</td>
<td>Most third party supporters</td>
</tr>
<tr>
<td>Elegant architecture</td>
<td>De facto standards</td>
</tr>
<tr>
<td>Product price</td>
<td>Total cost of ownership</td>
</tr>
<tr>
<td>Unique functionality</td>
<td>Quality of support</td>
</tr>
</tbody>
</table>

*Table 5-1: Representative list of product-centric value attributes versus market-centric ones*

Securing the channel for crossing the chasm into the mainstream market

The number-one corporate objective when crossing the chasm is to secure a channel into the mainstream market with which the pragmatist customer will be comfortable. The author explained the various types of channels and what they are optimized for:

- **Demand creators versus demand fillers.** Direct sales forces, for example, are optimized for creating demand, while retail superstores are optimized for fulfilling it. When crossing the chasm, our immediate goal is to create demand, but we must also look ahead toward putting in place a channel that can fulfill it.

- **Role in providing the whole product.** System integrators and Value-Added Resellers (VARs) are optimized for playing a very large role in providing or developing the whole product, and make much of their profits from this service. By contrast, retail and Internet channels take a low-cost position, based on the assumption that the whole product is already “institutionalized” and can be fully assembled from off-the-shelf parts. In the chasm case, the goal is to take the burden of whole product off of the channel in order to free it up to spend more time creating – and fulfilling – demand for the product.

- **Potential for high volume.** In some ways, this is simply the obverse of the previous category. Channels optimized for whole product development are not effective for high-volume delivery. There is too much labor in their business mix, so that when business is booming, they tend to slow down their selling efforts to work off some of the backlog – thereby flattening what could otherwise by meteoric growth.

The author examined various potential channels for crossing the chasm and one of them is System Integrators. However, **Systems Integrators usually focus on servicing early market opportunities sponsored by visionary customers, running in advance of the institutionalization of the whole product and providing custom solution to deliver the strategic advantage promised by newer technologies. Because they do not usually serve pragmatist customers, they are not suitable as a prime channel for crossing the chasm.**

Another option is direct sales force which is the optimal channel for crossing the chasm since it is optimized for creating demand. However, as product matures and the “price/performance” increases, the average selling price drops making it increasingly difficult to sell through a direct sales force.
A reasonable alternative proposed by the author is to field a direct sales force and support as a demand-creation channel to penetrate the initial target segment and then, once the segment has become aware of your presence and leadership, to transition to the most efficient fulfillment channel.

**Implications**

# 52: To cross the chasm, concentrate an overwhelmingly superior force on a highly focused niche market to win over pragmatist customers in advance of broader market acceptance.

# 53: Shift marketing focus from product-centric value attributes to market-centric ones as market matures and value-add shifts from the generic product to augmented services and ancillary products.

# 54: To cross the chasm and subsequently support a large mainstream market, a reasonable approach is to field a direct sales force and support as a demand-creation channel to penetrate the initial target segment and then transit to the most efficient fulfillment channel later.

### 5.10. Value Chain Dynamics

In the thesis by Masahisa Kawashima\(^4\), he describes the need for telecommunication carriers, which are still relying heavily on the revenues from traditional telephone calls, to re-build their business models and strategies in preparation for the eventual migration of telephony calls to IP-based converged networks. Inspired by the case of IBM, which was once the dominant computer "manufacturer" where the company shifted its business model successfully to a "service" model and in the process achieved increase profitability, the author studied the dynamics of the business in the PC industry and applied the insights in the telecommunication industry which has similar characteristics albeit at a slower clockspeed.

**Frameworks used for analysis**

The author presented frameworks, consisting of customer classification, technology/market lifecycle and a framework for multi-dimensional chain analysis, to help analyze and identify the shift of companies' competitive advantages and power dynamics in an industry. The customer

classification is adapted from the customer segmentation in the Technology Adoption Life Cycle, a model for understanding the acceptance of new products. The technology/market lifecycle, with three phases - phase F (Ferment), phase E (Early Adoption) and phase M (Majority Adoption) – is a combination of the technology lifecycle as defined by Henderson\textsuperscript{75} and Utterback\textsuperscript{76} with the Technology Adoption Life Cycle. And the framework for multi-dimensional value chain analysis is adapted from supply chain design and analysis using the three dimensions, namely technology supply chain, business capability chain and organizational supply chain proposed by Fine\textsuperscript{77}.

**Framework for Multi-Dimensional Value Chain Analysis**

The author uses a graphical representation shown in Figure 5-6 to effectively analyze and communicate changes in the value chain. In this diagram, each horizontal box represents an organization. A bus represents a chain, and its thickness indicates the integration force of a chain. An oval with “I” stands for integration, and its location indicates the organization that is integrating the chain. A rectangle with “D”, “S”, or “M” represents design, supply, and management process, respectively. An oval with “A” represents customer acquisition.

![Figure 5-6 PC Hardware value chains](image)

\textsuperscript{75} Rebecca Henderson, Technology Strategy (15.912) (Lecture taught at MIT Sloan School of Management, 2002, Spring).


80
PC Value Chain Dynamics

Using the above framework, the author analyzed the past value chain dynamics in the PC hardware industry as it evolved from the Ferment Phase to Early Adoption Phase and then to the Majority Adoption Phase, with disintegration and reintegration of the design, supply and management chain, as the power and capabilities of the players in the industry changes. He described how the strategies of system providers change in accordance with the value chain dynamics.

Five types of value chains were identified by the author: (1) Phase F/Type F – Tightly integrated supply chain (Phase for architectural innovation); (2) Phase E/Type E – Disintegrated supply chain (Phase for application innovation); (3) Phase M/Type M1 – Reintegrated design chain under platform dominance (Phase for market expansion and business model innovation); (4) Phase M/Type M2 – Reintegrated supply and management chains; and (5) Phase M/Type M3 – Disintegration of System Providers. The details are as follows:

(1) Phase F/Type F – Tightly integrated supply chain

In the 1970s and the early 1980s, the PC industry was in the ferment phase. Because the cost and performance of computers were not good enough for personal use, computer vendors such as IBM and DEC still had to continue architectural innovations. The system designs were based on integral architectures and the value chains were tightly integrated as shown in Figure 5-7.

![Figure 5-7 Value Chain in Phase F](image)

(2) Phase E/Type E – Disintegrated supply chain

Phase F terminates when the industry has achieved architectural innovation that realize “good enough” systems and the system architecture have converged to a dominant design. During the transition from phase F to phase E, the PC industry disintegrates tightly integrated value chains into the type E value chain depicted in Figure 5-8. The system is designed with modular architecture
and value chains are disintegrated in all of the three chains (D, S and M). The author refers to this value chain as a type E value chain.

![Diagram of PC Hardware value chain in Phase E (Type E Value Chain)]

The competitive advantage of system providers is still system integration capability in this phase. However, a capability for architectural innovations is not required in this phase. Instead, a capability to provide systems tailored to individual customers is required, because class E customers, potential adopters of systems in this phase, are very specific about their requirements. It is noteworthy that system integration capability can become the competitive advantages of system providers all the more because it is difficult to secure enough system integrity under disintegrated value chains with proliferated module suppliers.

(3) Phase M/Type M1 – Reintegrated design chain under platform dominance
In Type M1 model, systems are designed with modular architecture but the design chain is reintegrated by a platform dominator, in this case by the CPU vendor Intel.

During the transition from phase E to M, a dominant module supplier re-integrates the design chain. One factor that pushes the industry to re-integrate the design chain is the need for manageability. Because of the proliferation of applications and modules in phase E, system integrity or manageability is relatively low in this transition phase. On the other hand, unlike class E customers, class M customers require high system manageability. This need for system manageability becomes a force to re-integrate the design chain into the type M1 value chain shown in Figure 5-9.
As Gawer and Cusumano explored in “Platform Leadership”, dominant platform suppliers achieve this re-integration, using several ways including partnerships and acquisitions. The author describes below two major ways, which are “Virtual integration” and “Vertical integration”:

*In virtual integration, which is a form of partnership, a platform dominator specifies some requirements on other modules and certifies module vendors that comply with the requirements. The certificate issued by a platform dominator endorses the interoperability with the platform. This surely influences the vendor selections of potential buyers. Module suppliers naturally comply with the requirements. In this way, platform dominators can influence the design of other modules and improve the manageability of entire systems.*

*In vertical integration, a platform dominator extends its business and starts supplying other modules. An obvious advantage of vertical integration is that the system integrity between platform and modules is greatly improved, because they are supplied by a single organization. In addition, modular system architectures create separate modules for different applications. On the other hand, some applications can create additional values by interacting with other applications and performing atomic operations. In other words, some applications should desirably converge into one application. However, the separation of modules by applications hinders application convergence. In such cases, a platform dominator can effectively achieve application convergence, by integrating modules of different functions at the same time.*

In this phase, system integration capability no longer contributes to the competitive advantages of
system providers because (1) the platform dominator screens module vendors and secures system integrity and thus the required system integration skill of system providers is much lower in phase M than in phase E; and (2) class M customers do not require customizations as much as class E customers do as class M customers typically select systems that match their preference to minimize cost.

Thus, the author proposed that system providers should rebuild their competitive advantages in two areas: customer intimacy\textsuperscript{78} and process superiority\textsuperscript{79}. He cited the example of today’s IBM as a good model of a company that has built its competitive advantage based on customer intimacy and Dell as a good model of a company with process superiority.

(4) Phase M/Type M2 – Reintegrated supply and management chains
The inefficiency of a disintegrated value chain creates a force to reintegrate the supply and management chain (in this case by Dell) into a type M2 value chain.

Re-integrating the supply and management chain is exactly what Dell has successfully done to build its process superiority. The critical success factor in supply chain management is the ownership of business process design for the entire supply chain. Dell is designing its end-to-end supply process and asking module suppliers to comply with Dell’s process. It is noteworthy that the economy of scale is important for Dell to accomplish this supply chain re-integration. Since Dell does not own module suppliers, Dell’s supply chain management can be considered as virtual integration. As the result of the re-integration of the supply and management chains, a type M2 value chain as shown in Figure 5-10 emerges.

\textsuperscript{78} Primarily consists of a deep knowledge about customers and a capability to deliver systems that can fit customers’ business.

\textsuperscript{79} The efficiency in supply and management processes
(5) Phase M/Type M3 – Disintegration of System Providers

A system providers based on customer intimacy strategy faces a dilemma between scale and focus; it cannot re-integrate the supply and management chains without having economy of enough scale. However, seeking economy of scale may jeopardize its focus on customers as shown in Figure 5-11.

This dilemma faced by the system providers between scale and focus creates a force to disintegrate system providers into front-end providers with customer intimacy and back-end providers with process superiority. This results in a type M3 value chain depicted in Figure 5-12. An example of type M3 value chains can be found in today’s notebook PC industry. Several companies, including Dell, HP, Compaq, Gateway, and Apple, are providing their notebook PCs with type M3 value chains in which Quanta, a Taiwanese contract manufacturer, is playing the role of a back-end PC...
manufacturer\textsuperscript{80}.

As system providers proliferate, some system providers based on customer intimacy strategy try to emphasize their customer focus and choose to outsource their back-end capabilities. This creates a feedback loop reinforcing the strength of a back-end company; the back-end provider accumulates its knowledge on back-end processes and innovates in its processes. This widens the gap in back-end capabilities between front-end system providers and back-end system providers. The increased gap presses more front-end system providers to outsource back-end capabilities. As expected, Quanta has been reinforcing its position with its accumulated know-how on flexible manufacturing with efficient supply chain management.

![Figure 5-12 Type M3 PC value chain](image)

Although the dilemma between customer focus and economy of scale creates a force to disintegrate system providers, the author highlighted that there is a way to achieve both customer intimacy and process superiority without disintegrating system providers. He author cited the example of Dell which has successfully built its competence by achieving both customer intimacy and process superiority on its flexible build-to-order supply system\textsuperscript{81}. This competence is enabling Dell to hold the leadership position in the PC hardware market.


Summary
The author has described the dynamics of IT value chains, pointing out factors that change the value chains, and how the strategies of system providers change in accordance with the value chain dynamics. The following diagrams summarize the overall dynamics. Figure 5-13 shows the value chain dynamics with transition forces over the technology/market lifecycle. Figure 5-14 shows the changes in the strategies of system providers.
Figure 5-13 the value chain dynamics over the technology/market lifecycle
Figure 5-14 the shift of the competitive advantages of system providers

Application to Telecom Value Chains
Based on the above 5 types of value chains, the author then applies the multidimensional value chain analysis to the telecom value chains to identify their current status and ongoing dynamics, and proposes future strategies for telecommunication carriers.

It was assessed that the enterprise networking service is currently in phase M where the value chain is type M1. A few IP router vendors, such as Cisco, are integrating the design chain. The author predicts that carriers are expected to continue with its vertical integration activities and build up competence in the efficiency of supply and management to eventually create type M2 value chain. He further predicts that as large carriers create type M2 value chains and provide services with efficient supply and management, niche network service providers will feel the pressure to improve their supply and management efficiency and eventually choose to outsource network operations to these large carriers forming type M3 value chains.

Building on these forecasts about the future value chain dynamics in the telecom industry, the author proposes the strategies for how carriers should build sustainable competitive advantages in future M2 and M3 value chains. He first analyzed the business model of Dell, a system provider with sustainable competitive advantage in a M2 value chain, and the business model of Quanta, a
back-end notebook PC manufacturer with sustainable competitive advantage based on manufacturing process superiority in a M3 value chain. He then combined the sustainable competitive advantages of Dell and Quanta and proposed that telecom carriers should strive to be both the customer front-end service provider and back-end network provider in the M3 value chain.

The specific proposed strategies are: First, carriers should do business as both a front-end service provider in a M2 value chain and a back-end network provider in a M3 value chain. Second, as a front-end service provider’s strategy, carriers should reinforce their base of loyal customers by providing tailored supply and management services like “Dell Premier”. Third, as a back-end network provider’s strategy, carriers should create the value of a back-end network like “VISA”, by providing services for the interoperation between front-end service providers. Fourth, carriers should also build complementary assets, such as “design-for-manageability” know-how/patents and the position to aggregate contents/applications/ASPs, taking advantage of their operation volume in back-end network services.

However, one of the key problems of the proposed strategy to be both a front-end service provider and back-end network provider in the M3 model is that of conflict of interest with customers. As highlighted in the author’s analysis of Quanta business model as a back-end provider, in order to avoid the conflict with customers, Quanta keeps its policy not to have products with its own brand. Also, after conducting a further research on Dell’s notebook business, it is clear that even Dell, a system provider in the M2 value chain which handles final assembly for nearly all of its desktop PCs and servers, has made some changes to its M2 model towards a M3 model for its notebook PC business. Dell has decided to outsource to Quanta to produce its notebook PC in 2001 and Quanta produces 55% of its notebook PC in 2001. The growth in notebooks manufactured in Taiwan rose from 58% in 2001 to around 65% in 2002 and the reason for the increasing growth lies partly in the changing nature of the work performed. In the past, they primarily manufactured systems from blueprints forwarded by their multinational clients. Now, brand-name companies submit performance specifications, and companies such as Quanta design the notebook and provide other engineering services. That is, such manufacturing players are increasingly playing the role


of back-end providers in the M3 value chain even for players strong in both front-end and back-end competencies such as Dell. Thus, the carrier will have to choose a primary focus on either customer front-end service provider or back-end network provider.

**Implications**

# 55: The evolution of the value chain dynamics progresses from Type F to Type E and then to Type M1, M2 and M3 as industries mature.

# 56: In a M3 value chain, although a firm can play both the front-end and back-end provider role, the dilemma between customer focus and economy of scale forces the firm to choose a primary focus on either customer front-end provider or back-end provider.

### 5.11. The Innovator’s Dilemma

In the book “The Innovator’s Dilemma”\(^{84}\), the authors discuss why the logical, competent decisions that are critical to the success of their companies are also the reasons why they lose their positions of leadership. The authors highlighted that managing better, working harder, and not making so many dumb mistakes is not the answer to the innovator’s dilemma. There are times at which it is right not to listen to customers, right to invest in developing lower-performance products that promise lower margins, and right to aggressively pursue small, rather than substantial, markets.

The failure framework proposed by the author to resolve the dilemma is build upon three findings from his study.

- The first is that there is a strategically important distinction between sustaining technologies and those that are disruptive. Most new technologies foster improved product performance. Some sustaining technologies can be discontinuous or radical in character, while others are of an incremental nature. While all sustaining technologies have in common is that they improve the performance of established products, along the dimensions of performance that mainstream customers in major markets have historically valued. Most technological advances in a given industry are sustaining in character. Occasionally, however, disruptive technologies emerge: innovations that result in worse

---


product performance, at least in the near-term. Disruptive technologies bring to a market a very different value proposition than had been available previously. Generally, disruptive technologies under perform established products in mainstream markets. But they have other features that a few fringe customers value. Products based on disruptive technologies are typically cheaper, simpler, smaller, and frequently, more convenient to use.

- Second, the pace of technological progress can, and often does, outstrip what markets need. In their efforts to provide better products than their competitors and earn higher prices and margins, suppliers often “overshoot” their market. They give customers more than they need or ultimately are willing to pay for. And more importantly, it means that disruptive technologies that may under perform today, relative to what users in the market demand, may be fully performance-competitive in that same market tomorrow.

- And third, customers and financial structures of successful companies color heavily the sorts of investments that appear to be attractive to them, relative to certain types of entering firms. Investing aggressively in disruptive technologies is not a rational financial decision for them to make for the following three reasons: Disruptive products are simpler and cheaper, they generally promise lower margins, not greater profits; disruptive technologies typically are first commercialized in emerging or insignificant markets; and leading firms’ most profitable customers generally don’t want, and indeed initially can’t use, products based on disruptive technologies. Hence, most companies with a practiced discipline of listening to their best customers and identifying new products that promise greater profitability and growth are rarely able to build a case for investing in disruptive technologies until it is too late.

However, the authors caution about discarding the capabilities, organizational structures, and decision-making processes that have made them successful in their mainstream markets just because they don’t work in the face of disruptive technological change. This is because the vast majority of innovations that the companies face are sustaining in nature. Managers of these companies simply need to recognize that these capabilities, cultures, and practices are valuable only in certain conditions.

The authors further summarized the key insights for managing disruptive technologies into the following:

- First, the pace of progress that markets demand or can absorb may be different from the
progress offered by technology. This means that products that do not appear to be useful to our customers today (that is, disruptive technologies) may squarely address their needs tomorrow. Recognizing this possibility, we cannot expect our customers to lead us toward innovations that they do not now need. Therefore, while keeping close to our customers is an important management paradigm for handling sustaining innovations, it may provide misleading data for handling disruptive ones.

- Second, managing innovation mirrors the resource allocation process: Innovation proposals that get the funding and manpower they require may succeed; those given lower priority, whether formally or de facto, will starve for lack of resources and have little chance of success. With the focus on profitability, it means that until other alternatives that appear to be financially more attractive have disappeared or been eliminated, managers will find it extraordinarily difficult to keep resources focused on the pursuit of a disruptive technology.

- Third, successful companies have a practiced capability in taking sustaining technologies to market, routinely giving their customers more and better versions of what they say they want. This is a valued capability for handling sustaining innovation, but it will not serve the purpose when handling disruptive technologies. If, as most successful companies try to do, a company stretches or forces a disruptive technology to fit the needs of current, mainstream customers, it is almost sure to fail. Historically, the more successful approach has been to find a new market that values the current characteristics of the disruptive technology.

- Fourth, the capabilities of most organizations are far more specialized and context-specific than most managers are inclined to believe. This is because capabilities are forged within value networks. Hence, organizations have capabilities to take certain new technologies into certain markets. They have disabilities in taking technology to market in other ways.

The author concluded that established companies can surmount this barrier of managing the conflicting demands of sustaining and disruptive technologies by first understanding what the intrinsic conflicts are. They then need to create a context in which each organization’s market position, economic structure, developmental capabilities, and values are sufficiently aligned with the power of their customers that they assist, rather than impede, the very different work of sustaining and disruptive innovators.

**Implications**
This means that products that do not appear to be useful to our customers today (that is, disruptive technologies) may squarely address their needs tomorrow.

For the successful diffusion of disruptive technology, the more successful approach has been to find a new market that values the current characteristics of the disruptive technology.

The capabilities of most organizations are specialized and context-specific and organizations may only be able to take certain new technologies into certain markets but have disabilities in taking the technology to market in other ways.

The first step to managing the conflicting demands of sustaining and disruptive technologies is to first understand what the intrinsic conflicts are.

5.12. Scenario Analysis

In the chapter on “Strategic Uncertainty and the Future of Online Consumer Interaction” of the book “Sense & Respond”, the authors describe the use of scenario analysis as a tool to explore strategic alternatives and possible strategic responses in response to strategic uncertainties. They highlighted that in many cases of strategic uncertainties, not only are we surprised by unforeseen changes, we are surprised by the speed with which those changes occur. Many companies will be surprised by the means through which their operations, their profitability, and their competitive positioning will be altered.

The authors first explained the implications of strategic uncertainty to companies, and then described the procedure of exploring strategic uncertainty through scenario analysis, followed by the appropriate strategic response to uncertainty and the benefits of using scenario analysis.

Implications of strategic uncertainty

Many technologies exhibit slow adoption at first, and then suddenly achieve nearly universal adoption when some “tipping point” is reached. The possibility that adoption of a new technology may suddenly accelerate suggests that corporations will need to make some investment in new emerging technology, even when customers appear indifferent to it. The most critical decision may be to make strategic commitments under considerable strategic uncertainty by making critical

---

85 Chapter authored by Eric K. Clemons and Stephen P. Bradley.

choice on which partners to keep and which to support or abandon.

Exploring strategic uncertainty through scenario analysis

The authors describe scenario analysis as:

Scenario analysis is useful in strategic planning during times of rapid change when discontinuities in the business environment make extrapolation from available historical data misleading or meaningless. Scenario analysis attempts to identify the environment which a firm may have to operate, expressed as a set of scenarios that covers virtually any eventuality the firm may encounter. These scenarios are not intended to represent good, bad, or average cases. Rather, they are based on those fundamental driving forces that will determine the business environment.

Scenario analysis does not rely on data. It relies instead on the identification of uncertainties on which data would be useful, and on an exploration of the implications of assuming the most extreme values for such data. In scenario analysis, generally the following steps are performed:

1. **Surface the key uncertainties.** Identify questions that cannot be answered, but which appear to matter greatly.
2. **Rank the key uncertainties to determine the key drivers.** Identify the two or three most important unanswerable questions — the things that cannot be known, that if known, would tell strategic planners precisely what they need to know.
3. **Combine the key uncertainties to yield concrete scenarios.** That is, define alternative futures in which each uncertainty is assumed to take an extreme value.
4. **Provide details.** Turn each scenario into a plausible story, explaining how it might come about and how customers, executives in the firm, and executives in competing firms would feel about it.

Strategic responses to uncertainty

The author discusses about the need for companies to respond immediately to the advent of emerging technology or event, or at least with initial preparations, despite the high degree of strategic uncertainty. Companies have various choices in responding to strategic uncertainty but the optimal choice may be to invest in options for strategic positioning, learn more, and wait until the
environment becomes clearer. The authors argued that as the passage of time reduces strategic uncertainty, companies can invoke or cancel contingent strategies as appropriate. Although some strategic investments may be wasted, they are likely to have been small since they were investment in the preparation for rapid implementation of a strategy rather than in full implementation of a strategy. More importantly, investing in strategic options allow companies to be prepared to respond quickly when the tipping point is reached.

Benefits of using scenario analysis
The authors described several benefits of using scenario analysis. First, it can serve as the basis for a discussion of alternative views of the future, providing a common basis for understanding and a common vocabulary for exploring implications for the future. Second, when properly extended and tailored to the needs of individual companies or industries, it can also provide the basis for strategic planning, including contingency planning both for threatening scenarios and for promising scenarios that offer real business opportunities. Third, by making a company more sensitive to early warning signs and indicators of emerging scenarios, scenario analysis can facilitate environmental scanning and accelerate a company’s strategic responses. Most importantly, early environmental cues may be so indistinct, and run so counter to prevailing expertise and intuition in the firm, that they would be discounted without the heightened awareness created by scenario analysis.

Implications
# 61: In the rapidly changing IT business, scenario analysis combined with system dynamics is a useful tool to explore strategic alternatives and possible strategic responses in response to strategic uncertainties.
6. Analysis – IBM versus BEA

6.1. Introduction

This chapter provides a comparative analysis of acquisition activities and their impact on the technology portfolio of IBM and BEA, the two leaders in the Java APS software market. I will compare the technology-based business development strategies of IBM and BEA using the familiarity matrix proposed by Roberts and Berry, and offer general conclusions regarding technology strategy in the APS market.

This section is organized as follows: In Section 6.2 and 6.3, I present the corporate, business and technology strategies, and an analysis of the acquisitions made by IBM and BEA respectively. Specifically, the paper probes the following five key technology issues for each firm: (1) The motivation and rationale behind the acquisitions, and the organizational considerations that promote technology transfer within the integrated firm; (2) The impact of the acquisitions upon the company’s portfolio of technology competencies; (3) The penetration of new markets and access to new technology areas provided by the acquisitions; (4) The results of the acquisitions; and (5) Any change in technology acquisition strategy. In Section 6.4, I compare and contrast the acquisition strategies of IBM and BEA and draw conclusions for developing new business and technology strategies in the APS software industry.

6.2. IBM

6.2.1. Corporate and Business Strategy

Until early 1990’s, IBM’s corporate strategy centered on its mainframe computer business. The company had an integrated offering of hardware, software and services for mainframe customers and growth was organic, driven by internal R&D investment. The advent of distributed and network computing, client-server model and PCs challenged IBM’s mainframe stronghold in the late 1980’s and lead to rapid decline in profits. However, a change in leadership brought a change in strategy. In 1993, Louis Gerstner took over the CEO post at IBM. The new strategy set by Gerstner was to keep IBM together, to focus on customers and to create new businesses in services and software. His strategy was to leverage IBM’s R&D breadth and its ability to be a

---

“one-stop-shop” for different kinds of hardware and software products and services. The turnaround proceeded well and IBM returned to profitability in 1994. The improved financial position gave the company the flexibility to enter into new growth businesses through acquisitions. Growth through acquisition continues to be the primary strategy for IBM in their software and service businesses.

According to an interview with an IBM executive, the key to the growth through acquisition strategy is to capitalize on IBM’s strengths in brand marketing and global sales channels. Large enterprise customers are usually not comfortable investing in new technology from small, unknown companies and IBM is able to provide the stability and worldwide support. IBM also embarks on what they call “opportunistic consolidation”. The primary objective in this type of acquisitions is to acquire the installed based of acquired company and then up-sell or cross-sell. In these cases IBM will typically not pay above the market valuation for the acquisitions. However, there are not many such opportunities left and it is increasingly difficult to identify companies for such kind of growth acquisition.

According to Samuel J. Palmisano, Chairman, President and CEO, IBM’s current corporate strategy is centered around “e-business on demand”, with three main thrusts:

1. Helping our customers become “on demand businesses”.
2. Evolving the computing model to an “On Demand Operating Environment”.
3. Establishing utility computing - computing on demand - as a viable and attractive alternative method for accessing and paying for IT.

IBM’s Software Group and the WebSphere platform are essential part of this corporate strategy. The details of the strategy are elaborated:

1. Helping our customers become "on demand businesses"

   Through IBM Global Services, we are applying IBM’s considerable business process and industry expertise to help customers build businesses that are almost intuitive in their responsiveness to changes in demand, supply, pricing, labor, capital markets and

---

88 2002 IBM Annual Report.
customer needs. This requires a great deal of integration of business processes and operations, and of applications and the underlying IT systems. It means making them resilient in the face of changes and threats, from hackers to hurricanes. And it means helping them focus on what differentiates them, on their core competencies and outsource or tightly integrate with strategic partners to supply the rest.

(2) Evolving the computing model to an "On Demand Operating Environment"

On demand business creates new rules for IT infrastructure. Computing must be integrated and must support integration of business processes and operations, which is why our WebSphere software is growing so rapidly. Computing must be built on open technical standards and platforms, which is why IBM will continue to be a leader of the open standards movement leader in Linux, Web Services and other emerging technical standards. Applications must be developed for this new, open model, which is why we acquired Rational; it gives software developers a compelling alternative to proprietary approaches.

In addition, an emerging technology called grid computing, built around another set of open specifications, allows the sharing and managing of separate computing resources as if they were one huge, virtual computer. This will dramatically increase utilization rates and give customers access to enormous computing capacity. Finally, IBM technologists are also pioneering ways to make IT systems "autonomic": more self-managing and self-healing. This, too, is critical as the increasing complexity of systems is making them unrealistically costly to manage and maintain.

(3) Establishing utility computing - computing on demand - as a viable and attractive alternative for accessing and paying for IT.

This effort has gotten a lot of attention. Yes, we intend to be a leader in utility computing services, so that customers can acquire computing and applications and pay only for what they use. IBM Global Services is already pioneering such services for server and storage capacity, as well as business processes like procurement and claims processing for companies such as American Express, The Dow Chemical Company and Mobil Travel Guide. But we also want to equip and help customers to build their own internal utilities software to manage and balance workloads, and server and storage systems to provide additional capacity on demand.
6.2.2. Technology Strategy

As the largest computer system vendor, IBM's technology strategy covers everything from microprocessors and storage technologies to Web Services and industry specific application software. To provide a relevant comparison with BEA, this section focuses on IBM's middleware strategy.

According to the IBM executive, IBM is constantly faced with build versus buy decisions in their middleware software business. During the 1990's, the acquisition decisions were ad hoc and opportunistic, especially during the dot.com boom with many companies wanted to be bought. From early 2000's, the company started a proactive process for identifying technology gaps against IBM's middleware product roadmap and evaluating acquisition opportunities. Various teams were formed focusing in areas such as content management, storage management, security management, etc., with the objective of educating management, segmenting the market, and identifying opportunities and market players.

The key criterion for making the build versus buy decision is often time to market. With internal development it may take 6-9 months to form a team through reallocation of resources and recruiting before actual development work can be started, while in an acquisition the acquired development team is already up and running and focused on the specific technology. Another important consideration is the existence of any ecosystem of ISVs around the acquired company’s product that provide value to the acquired firm. One significant issue that has grown in importance in the recent years is the use of open-source, especially if it is under General Public License (GPL).\textsuperscript{89} In these cases, IBM has to spend effort in swapping out and replacing the GPL code. Therefore, the checking on use of open-source has become an important part of the due diligence process.

\textsuperscript{89} The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software - to make sure the software is free for all its users. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
6.2.3. Analysis

Motivation and rationale, and organizational considerations

In order to gain insight into IBM’s acquisition strategy, seven key acquisitions were analyzed in more detail. Appendix A – Acquisitions of IBM lists the acquisitions and their primary motivations and rationale.

The first significant software acquisition was made in 1995 when IBM acquired Lotus Development Corporation, making IBM the largest software vendor worldwide at that moment. The Lotus acquisition was a bold and visible major step into PC based office productivity application and away from IBM’s base market in mainframe software. According to the IBM executive, the key considerations during the integration phase were retention of Lotus engineers and customers. To ensure retention of engineers, IBM was careful to keep their people away from Lotus in order not to overwhelm or inhibit Lotus. IBM also did not try to integrate the backend processes and systems such as HR, IT, order management, etc. For customer retention, the key decision concerned the product roadmap. It was announced 2 weeks after closure that all IBM products in workgroup area were to be discontinued and focus would be solely on Lotus. This gave an important signal to the market and customers and did not allow angst to build up, especially among the IBM staff that was impacted by the decision.

The second major acquisition was Tivoli Software in 1996. In this case, IBM had been trying to expand their host management business into distributed space through internal development, but due to different architectures between host-based and distributed environment this would have required significant investments in new competencies and progress was slow. As an alternative, IBM decided to acquire Tivoli, that had $50m revenue, at 15 times revenue multiple. It was difficult to justify the business case at 15 times multiples. However, Tivoli’s market then was largely fragmented and small, and IBM reasoned that if they can help to increase its market and revenue with IBM’s branding and resources, it could make economic sense.

The other analyzed acquisitions (NetObjects, CrossWorlds, Access 360, Holosofx and Rational) were primarily point technology acquisitions to fill specific gaps in IBM’s product portfolio with

---

fast time to market.

Impact on Technology Portfolio
IBM’s technology portfolio is summarized in Table 6-1. The first column lists IBM’s cumulative technology portfolio following its 7 technology acquisitions. There are mainly two types of technologies in this portfolio. First, there are technologies that reinforce the Java application server itself. Second, there is software suite like Lotus, DB2, Tivoli and MQ that are complementary components to the Java application server. However, all of the listed technologies contribute to the adoption of WebSphere.

One issue worth noting is that IBM has not been shying away from acquiring technologies that it already possesses, such as integrated development environment (IDE) and business integration platform, if the acquisition improves IBM’s competitiveness. For example, in the case of CrossWorlds, IBM already had an integration product called MQSI. However, IBM relied on a third party called NEON for its adaptors to enterprise business applications. But when Sybase, an IBM competitor in the integration market, acquired NEON, IBM needed a new source for these adaptors and with CrossWorlds’s acquisition IBM was able to fill this gap fast.

<table>
<thead>
<tr>
<th>IBM’s total technology portfolio after all acquisitions</th>
<th>Inhouse</th>
<th>Lotus</th>
<th>Tivoli</th>
<th>NetObjects</th>
<th>CrossWorlds</th>
<th>Access360</th>
<th>Holosofx</th>
<th>Rational Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Groupware</td>
<td>+++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relational database</td>
<td>+++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java Application Server</td>
<td>+++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message Queing</td>
<td>+++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System management &amp; monitoring</td>
<td>+++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network management &amp; monitoring</td>
<td>+++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrated Development Environment</td>
<td>+++</td>
<td>+++</td>
<td></td>
<td>+++</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Business Integration Platform</td>
<td>+++</td>
<td>+++</td>
<td></td>
<td>+++</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Directory Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J2EE development environment</td>
<td>+++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6-1: Portfolio of technology competencies for IBM and acquired companies

Familiarity Matrix Analysis
Next, I analyze IBM’s acquisitions by positioning them on the Familiarity Matrix. Figure 6-1 depicts the assessments of IBM’s acquisitions on the matrix. The Lotus acquisition is positioned in the top right-hand corner, as the acquisition provided access to new network-centric as opposed to
mainframe-centric technology (New Unfamiliar technology) and in the desktop software market (New Unfamiliar Market). The rest of the acquisitions can be considered to be in the lower left base-familiar sectors, as they often replaced or complemented existing products or technologies in IBM’s portfolio and IBM used their existing sales channels to market the acquired products to their existing enterprise customers.

![Familiarity Matrix for IBM acquisitions](image)

**Figure 6-1: Familiarity Matrix for IBM acquisitions**

**Results of acquisitions**

Overall, the growth by acquisition strategy in middleware software has been a success for IBM. In 2003, the revenue contribution from software acquisitions is estimated to be approximately $3 billion.

Lotus acquisition did not end Microsoft’s dominance of the PC desktop as intended, but in the long term the unit has still been able to provide IBM steady revenue growth. 2000 was its best revenue
year ever with estimated revenues of $1.5 billion\textsuperscript{91}. The number of Lotus Notes seats has grown from 15 million at the time of the acquisition to 110 million in 2003. In terms of integration, the success has been somewhat mixed. According to the IBM executive, IBM did not fully integrate Lotus until a few years ago and in retrospect, they should have integrated faster.

Tivoli acquisition was initially even a bigger success than Lotus. It was kept as a standalone business unit, and revenue grew from approximately $50 million to $500 million over the first 2 years. In addition, IBM transferred its own $500 million revenue generating middleware and system management products to the Tivoli Unit (so called “reverse merger”), making it a billion dollar business for IBM by end of 1997. This segment’s revenue growth rate (approximately 35%-40%) was the highest of any of the top 10 independent software companies around the world, with the exception of SAP A.G. (60%-65% growth estimated by IDC)\textsuperscript{92}. However, in 1998-1999 Tivoli’s start-up culture, especially the “hit and run” sales culture started threatening IBM’s long-standing relations with its enterprise customers. Also, integration and re-engineering of the Tivoli product architecture have required significant investments from IBM in the past few years.

**Changes in Acquisition Strategy**

As a result of the lessons learnt from the Lotus and Tivoli acquisitions, IBM has shifted their acquisition approach from slow integration to rapid and immediate integration. The primary driver for this shift is the emergence of customer problems that impact the long and deep relationship with enterprise customers. As part of due diligence, the people and culture are assessed on the possibility of rapid integration. Another reason for the change in strategy is that with rapid integration, IBM is able to take the product through their worldwide channel more effectively. Also, customers want integrated solutions and there is a massive effort to evolve the various products into an integrated solution through modularization, componentization and standardization that will allow for the reuse of well-tested components, modules, architecture and technology resulting in better quality of new product development, faster time-to-market and evolution of the products.

Another recent change in IBM’s acquisition strategy is leveraging their corporate ventures. IBM has invested in several VC funds and established relationships that will allow IBM to understand


their portfolio and what companies are doing. When IBM is interested in acquiring a certain technology or capability, IBM is now increasingly approaching the VC community to source for potential candidates.

6.2.4. Summary

In summary, the acquisitions of Lotus and Tivoli were motivated by IBM’s corporate strategy to enter the fast growing middleware market, while the later acquisitions were primarily incremental acquisitions of point technologies or capabilities that were either missing or weak in IBM’s technology portfolio.

With its sometimes overlapping technology acquisitions and customers’ demand for more integrated solution, IBM is trying to reconcile products and technologies from its own labs and the diverse companies it has acquired. However, integrating these into a cohesive architecture is a daunting technical task and will probably take longer than anticipated. ISVs and global system integrators might be reluctant to adopt IBM’s current loosely integrated platform when they can use more stable and less technically risky products from IBM’s competitors. Moreover, customers may be unwilling to risk a major product migration exercise that will become a necessity after IBM integrates its platform and retires duplicate components in its current technology portfolio.

6.3. BEA

6.3.1. Corporate and Business Strategy

A succinct summary of BEA’s vision emanates from the statement by Alfred S. Chuang, Founder, President, and CEO of BEA Systems in the company’s 2002 annual report.

“We have more than 13,000 customers worldwide, including the majority of Fortune Global 500 companies. We do not succeed unless they succeed literally: our growth plan is based on the revenue of relationships, not only recurring but increasing revenues, as our customer partnerships deepen and expand. In other words, our customers’ business is our business. Their customers are, ultimately, our customers. Our vision is to provide the enterprise software platform that enables our customers to execute most effectively on their vision. And our success is increasingly powered by long-term partnerships rather than short-term product sales.”

In order to implement this vision, BEA has devised the following corporate and business strategies:
(1) Customer-Focused strategic planning and product development, (2) Building a ecosystem through strategic relations with key partners and building of developer community, (3) Sales and marketing to both senior executives and information technology department personnel, (4) Maintaining its technology leadership, (5) Continuing its standards leadership, and (6) Staying focused and remain as an independent and pure-play software vendor. The details of the strategies are:

(1) Customer-Focused strategic planning and product development
In response to customers’ needs, BEA invested aggressively to leverage its leadership in the Java application server market into leadership in the much larger application infrastructure market. This market is emerging in response to the distress calls coming from the enterprise as a result of the complexity of IT. Companies want to make information easily available to whomever needs it in their enterprise, serve their customers better, and do more with less. And they want their enterprise IT architecture to help them achieve these goals rather than hindering them. They wanted simplicity in their application infrastructure and do not want to evaluate these core infrastructure technologies such as application server, integration, portal, development and deployment, security, and application management separately because they are fundamentally connected and interdependent.

To meet these needs, BEA invested heavily for two years to extend BEA WebLogic Server into an underlying software foundation for the enterprise that simplifies building, integrating, and extending applications.

(2) Building a ecosystem through strategic relations with key partners and building of developer community
As a vendor without a large consulting, professional services and system integration division, BEA has two choices: grow services expertise organically, or partner with firms that can bring those services skills to bear. It will be a great, if not impossible task, to take on IBM Global Services directly in a service play. Moreover, BEA does not have IBM’s product breadth, which includes both hardware and software, to cover the entire horizontal segments. Thus, BEA competes through ecosystem driven partnership.

With the initial success of BEA’s application server and building up of a critical market share, BEA is in a position to achieve a dominant position in the enterprise infrastructure software industry.
Similar to Microsoft's strategy of encouraging ISVs to develop based on Microsoft Windows platform, BEA strived to nurture, attract and retain these ISV complementors to enhance BEA’s offerings. BEA defined an open architecture and made available integration interfaces for any third party best-of-breed component provider to integrate with the BEA platform. With its initial critical market penetration, many end-users demand other third party best-of-breed component provider to be integrated with BEA platform. The enhanced BEA’s offering which provides a broad horizontal suite of products further attract more customers, and result in increasing growth for both BEA and its partners.

An important element of BEA’s sales and marketing strategy is to increase the indirect sales capacity by expanding their relationships with third parties and strategic allies such as system platform companies, independent software vendors (ISVs), application service providers (ASPs), system integrators (SIs), independent consultants, and distributors to increase the market awareness, demand and acceptance of BEA and their solutions.

BEA also markets their software directly to system and application developers through their developers’ website. The website is designed to provide a forum for BEA developer’s community to exchange technical information and sample code, as well as provide feedback to BEA on their products and industry directions.

As at the end of 2002, the 13,000 customers, including the majority of the world’s largest and most successful companies, 2,200 partners and alliances, and more than 400,000 registered developers in BEA’s “dev2dev” community, formed a very strong enterprise software ecosystem centered around BEA’s WebLogic Enterprise Platform.

(3) Sales and marketing to both senior executives and information technology department personnel
One of BEA’s strategies\(^{93}\) is to pursue opportunities worldwide within large organizations and organizations that are establishing e-businesses, through their direct and indirect sales channels. Once penetrated, BEA will attempt to proliferate within those customers, servicing higher usage volumes and selling additional products.

The company typically uses a consultative, solution-oriented sales model that entails the collaboration of technical and sales personnel to formulate proposals to address specific customer requirements. Because the solutions are typically used as a platform or integration tool for initiatives and applications that are critical to a customer’s business, BEA focuses its initial sales efforts on senior executives and information technology department personnel who are responsible for such initiatives and applications.

(4) Maintaining its technology leadership

Through BEA’s research and development efforts or acquisitions of complementary companies, products and technologies, BEA has embraced new standards and added important features and functionalities to its product line. BEA products have won several key industry awards and have received strong recommendations from key industry analysts.

One of the key strengths of the product is that the platform comprised of a core set of Java-based products designed from the ground up to run on the Web and are highly integrated with each other. This strength of integration can be seen by how different components can be centrally managed through the same administrative console and how skills in one product carry over to the other products.

In BEA’s attempts to penetrate the application integration market against IBM, it will not adopt the key tactic employed by IBM of acquiring small and niche integration specialists, struggling for survival. BEA believes that acquisition would fail to enhance BEA’s WebLogic Platform from a technology perspective.

(5) Continuing its standards leadership

One of the competitive advantages of BEA is the ardent support of the open-standards strategy on which all BEA products are based on. With an open architecture and integration interfaces made available for third-party best-of-breed component provider to integrate with the BEA platform, it offers customers more options and greater flexibility in their technology choices to meet their needs.

---

specific needs. For BEA, open standards have proven to be a key differentiator. Customers know
that they can count on BEA for a neutral, interoperable platform that isn’t designed to lock them
into a proprietary technology or a particular vendor. BEA continued to devote significant resources
in supporting open standards, including taking leadership positions on standards boards, partnering
with other industry leaders on standardization planning, and supporting open standards in all
product development.

(6) Staying focused and remain as an independent and pure-play software vendor
BEA positioned itself as an independent and pure-play application infrastructure software vendor
and leverages this focus as a competitive advantage. BEA’s customers, especially the ISVs who are
developing package applications, prefer to purchase the application infrastructure software from a
pure-play vendor who would not become a direct competitor in the package applications software
market.

6.3.2. Technology Strategy
BEA’s total research and development expenses were approximately $61.0 million (13% of
revenue), $89.2 million (11% of revenue) and $120.9 million (12% of revenue) in fiscal 2000, 2001
and 2002, respectively. The market for their products is highly fragmented and competitive with
alternative computing architectures and characterized by continuing technological developments,
evolving and competing industry standards, and changing customer requirements. The introduction
of products embodying new technologies, the emergence of new industry standards or changes in
customer requirements could render their existing products obsolete and unmarketable. As a result,
their success depends upon their ability to timely and effectively evolve their products to keep pace
with technological and market developments, and emerging industry standards.95

Up to 1998, BEA positioned itself as a middleware company providing transactional, messaging,
and distributed object-based software for developing and deploying e-commerce applications and
for connecting e-commerce applications to legacy and mainframe applications. In Sep 1999, BEA
had a major strategic re-think. It decided to make a very conscious effort to reposition itself as an
e-commerce company. BEA committed more than $20 million to re-branding the company as “The
e-commerce-transaction company”, a shift away from its earlier identity as a middleware producer.

95 BEA Systems, Inc, 2002 Annual Review.
In 2000/2001, based on customers' needs as well as the need to combat the increasingly commoditized Java application server market and regain its leadership position, BEA embarked on a key pioneering initiative: To develop the first unified platform converging application development and application integration into one framework, reducing the effort and complexity of application development and system integration. BEA developed a vertically integrated platform suite through a series of concerted internal development and educational acquisition efforts. The new platform product was delivered in Mar 2003 and it allowed all development (portal, application and integration) to be done using a single visual development environment. The new product has attracted much positive review from the industry but the actual adoption of the platform by the majority of existing client base will determine whether the BEA can return to double digit revenue growth rates in the late FY04 and early FY0596.

6.3.3. Analysis97
Motivation and rationale, and organizational considerations
Since its inception, BEA has acquired several companies and added additional product lines and functionalities to its existing products as well as expanded its direct distribution and service capacity. The details of the acquisitions are summarized in Appendix B – Acquisitions of BEA. The motivation and rationale of the acquisitions can be divided into two groups, namely technology acquisitions, and consulting and educational resource acquisitions.

Within the former group, Tuxedo from Novell and Object Broker/MessageQ from DEC are the earliest BEA's acquisitions. Tuxedo became a core product of BEA at that time which would meet the growing demand for the mission critical middleware when the industry shifted towards distributed computing. Object Broker/MessageQ also augmented the required functions for the middleware.

In 1998, when the company acquired WebLogic Inc., BEA started to focus on web application server business rather than backend middleware business. At that time, Java server side technologies were just established and BEA wanted to enter the Java application server business.

97 With inputs from interview with a BEA executive, 18 Nov, 2003.
They acquired WebLogic Inc. due to its Tengah application server’s superior quality to similar products such as Netscape Application Server, Sybase Java CTS and Microsoft MTS.

BEA acquired several other companies to augment certain key components of its APS: portal (The Theory Center in Nov 1999), business process workflow automation (Workflow Automation Corp in Mar 2000), security (CrossLogix in Feb 2003) and performance accelerator (Appeal Virtual Machines in 2002). To enhance its development tool capability, it invested in WebGain in Dec 99 and subsequently embarked on an educational acquisition of CrossGain in Jul 2001. Besides augmenting its APS, there were additional motivations for the acquisitions of Appeal Virtual Machine and CrossGain.

While Java Virtual Machine (JVM) is the core infrastructure technology required to run the Java application server, BEA did not have its own JVM. As a result, BEA had several problems. For example, availability on its product had been sometimes delayed due to the functional change of new version of JVM, and when problems occurred in its system, it had difficulty to determine whether the problem lies in its application server or the third party JVM. BEA had to keep close contact with third party JVM providers like Sun Microsystems. Considering this situation, BEA’s acquisition of Appeal Virtual Machines seemed to be just a matter of course. Moreover, the acquired JVM is co-developed with Intel to run on the new Intel Architecture 64 (IA64) chip, the next generation Intel chip. This is the only JVM that runs on IA64 and will provide BEA with some competitive advantage for a while.

The acquisition of CrossGain is another key strategic move by BEA, realizing that there is a need to enter the IDE market directly in order to reduce the J2EE technologies complexity and simplify the development process. Through CrossGain, BEA gained the expertise of a 40 employees startup staffed by many former Microsoft executives and developers, including Tod Nielsen, a 12-year Microsoft veteran and former vice president, and Adam Bosworth, a 10-year Microsoft evangelist responsible for driving the database and XML efforts within Microsoft. CrossGain has assembled an enterprise class management and technical team with an incredible track record of success, and enabled BEA to deliver on its integrated tool strategy, developing a user friendly development tool, the hallmark of Microsoft’s development tool. In an interview with a BEA executive, he said: “In the case of CrossGain, the acquisition of talented human resource is more important than the product itself.”
In technology related acquisitions, BEA pays much attention to retain the motivation of the engineers. Especially when the acquired technology is slated to be the core technology for BEA as in the case of Tuxedo, WebLogic and CrossGain, BEA tries to preserve the culture and environment of the acquired companies by keeping their existing offices and not forcing the relocation of engineers which may destroy the firms’ tacit and/or socially complex knowledge bases. It was only in 2001, 3 years after WebLogic acquisition, that BEA decided to merge the Tuxedo team and the WebLogic team to enable tighter integration of WebLogic and Tuxedo technologies. Also, the autonomy given to CrossGain allows Adam Bosworth, the previous Microsoft’s XML evangelist, to attract other talents and help establish a research center in Seattle, which has since become the center of R&D for BEA.

The approach for consulting and educational resource acquisition is slightly different. The management and arrangement is individually tailored according to each employee’s skills and industry experience. There is more flexibility in deciding on their physical location and other matters.

Impact on Technology Portfolio
BEA’s technology portfolio is summarized in Table 6-2. The first column lists BEA’s cumulative technology portfolio following its 8 technology acquisitions. An examination of the technology portfolio yields several interesting observations. First, BEA is a company primarily built through acquisitions of startup or small companies. The primary objectives of the acquisitions are for market entry and technology instead of purely appropriating the market of the acquired firm. Second, there is no overlap in technological expertise in the acquisitions and lastly, the acquired technologies are subsequently integrated through ground up development to evolve into a fully integrated platform.
<table>
<thead>
<tr>
<th>BEA's total technology portfolio after all acquisitions</th>
<th>Inhouse</th>
<th>Novell Inc. - Tuxedo Technology</th>
<th>Digital Equipment Corporation - Object Broker, DECoMmessageQ, etc</th>
<th>WebLogic Inc</th>
<th>The Theory Center</th>
<th>Workflow Automation Corporation</th>
<th>Crossgain Corporation</th>
<th>Appeal Virtual Machines AB</th>
<th>CrossLogix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java Application Server</td>
<td>+++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrated Development Environment</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workflow framework</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portal framework</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security framework</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Java Virtual Machine</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data view wrapper for data sources</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Online Transaction Monitor</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message Queueing</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object Request Broker</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reusable software component</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 6-2: Portfolio of technology competencies for BEA and acquired companies**

**Familiarity Matrix Analysis**

I next analyze BEA's acquisitions by placing each one into the Familiarity Matrix. Figure 6-2 depicts the assessments of BEA's acquisitions on this grid. From the Familiarity Matrix perspective, Tuxedo, Object Broker/Message Q, WebLogic, Workflow Automation Corp and CrossLogix appear to reside in the new familiar market and new familiar technology sector. Tuxedo and WebLogic formed the core acquisitions by BEA to enter the middleware and Java application server market respectively and were highly successful. Object Broker/Message Q, Workflow Automation Corp and CrossLogix augmented BEA's APS capabilities in messaging, workflow and security respectively. These acquisitions have mixed results. Both the messaging and workflow products were originally acquired for the purpose of bundling for sale with Tuxedo and WebLogic respectively. According to the BEA executive, the workflow acquisition is also for the purpose of entry into the business-to-business and enterprise application integration market, especially for integration with the then popular ARIBA e-Market Place product. However, the realization that these products do not fit the open standards approach and coupled with lower than expected sales, BEA decided to stop selling them and diffused the messaging and workflow framework technologies into BEA APS. Aligned with the theory, the results of acquisitions in the marginal sectors are mixed.
Different from all other acquisitions, The Theory Center (TTC) acquisition is positioned in the new unfamiliar market and new familiar technology sector. In this case, BEA attempted to enter the new and unfamiliar EJB reuse component business. But BEA soon faced the problem that the fixed specification of components did not match the various customers’ requirements. Finally, BEA concluded that it was difficult to make a profit by selling components. It stopped the sales of the components and evolved the TTC portal components into the current BEA APS Portal framework.

Appeal Virtual Machines acquisition is positioned in the base market and new familiar technology sector. The technology is seamlessly integrated in the platform removing BEA’s dependency on third party JVM providers.

Finally, CrossGain acquisition is located in the new familiar market and new unfamiliar technology sector. It is an educational acquisition not only for the IDE technology but also its XML capability. The Roberts/Berry arguments highlight the diversity of CrossGain technologies relative to BEA base capabilities and suggest that it is risky to rapidly integrate CrossGain into its own core activity.
BEA kept the team in its original location in Seattle and provided much autonomy to it. This acquisition proved to be highly successful as the team evolved to become the nucleus of the new centre of R&D for BEA.

Despite the many technology acquisitions, BEA fairly successfully integrated them into a unified architecture. The BEA executive highlighted the importance of constantly aligning the technical direction and sharing the concept of the architecture with the various technology units.

Results of acquisitions
WebLogic is the monumental acquisition. "The acquisition expenditure for WebLogic is fairly large, compared to other acquisitions. Out of all the acquisitions, this acquisition clearly contributed to the revenue increase." said the BEA executive. With WebLogic, BEA's revenues tripled between mid-1999 and mid-2001, making it the fastest software firm ever to reach $1 billion in revenues with a customer base of more than 13,000 enterprises around the world.

The rest of the horizontal technology acquisitions resulted in the new unified platform. According to The Buckingham Research Group⁹⁸, recent discussions with several developers and SIs suggest that the new WebLogic 8.1 platform is a substantially improved product which clearly differentiates BEA from its competitors, especially IBM. Its well-integrated development and deployment environments, and robust integration capabilities has clearly put the company ahead of its peers. However, the actual adoption of the platform by the majority of the existing client base will determine whether the company can return to double-digit revenue growth rates in the late FY04 and early FY05.

One of the key impediments to the adoption of the new BEA platform is the concern of lock-in to BEA. For BEA to build an easy-to-use tool, the company had to include new commands that are not yet part of the Java language standard. Java proponents and customers of BEA fear that the new tool will result in Java software that does not comply with the current standard making it incompatible with other servers, and potentially breaking the "write once, run everywhere" promise of Java. BEA, however, plans to submit the new technology to the standards process and to allow its rivals to use the same commands in their tools. Analysts say software makers, including IBM,

---

have historically submitted their new Java technology to the Java Community Process standards body. BEA will likely have to show prospective customers exceptional productivity gains to justify lock-in to its design model.

Changes in Acquisition Strategy

The BEA executive highlighted that after the recent acquisition of CrossLogix, there is a change in BEA’s acquisition strategy. The core framework is already established and BEA will not be changing the core architecture dramatically. No future major acquisition is expected and BEA will continue to build strategic alliances with key partners to reinforce its product offerings.

6.3.4. Summary

In summary, Tuxedo and WebLogic were the core acquisitions by BEA to enter the middleware and Java application server market respectively and were highly successful in terms of revenue contribution, while the later acquisitions contributed to the breadth of BEA’s technology portfolio. With its core technology in place, BEA’s management is now focusing on its sales force transition in an attempt to improve its sales force execution. The integration market will require a different sales approach than has been used to sell the core application server. In entering the application server market, BEA successfully targeted the developer community that embraced its product for its superior technical quality. However, penetration to the enterprise integration market requires decisions to be made by the Line of Business managers as well as the C-level executives within organizations, neither of whom BEA has traditionally called upon. Hence, the entry into the integration market by BEA will require a change in its sales forces’ go-to-market strategy.

As the industry matures further, even broader horizontal integration of the APS is expected, including performance management, security, database management, development process management, and operations management. IBM already has the products and domain expertise to offer the various components. To compete effectively, BEA will have to either acquire or develop these products and technologies in the near future, or continue with its strategic partnership with


ISVs to provide the missing components and the SIs to integrate the components for the customers.

6.4. Summary

From the study, there are fundamental shifts in business and technology strategy of IBM and BEA. IBM has changed its acquisition approach from slow to rapid and immediate integration in order to provide consistent support to its customers, to enable it to take the acquired product through its worldwide channel more effectively, and to enable tighter integration of the various products. BEA, on the other, has achieved a unified platform of core APS components and is now shifting its focus on its sales force transition to improve its sales force execution. Both IBM and BEA benefited substantially from the technologies and new market opportunities through their acquisitions of complementary companies that enabled technology gaps to be filled, existing capabilities to be reinforced and the creation of new business revenue streams. The comparison of the two companies with respect to the five key technology attributes is summarized in Table 6-3.

Both IBM and BEA have placed great emphasis on the preservation of the acquired firms' tacit and/or socially complex knowledge base by avoiding unnecessary relocation of employees and offices, controlling the pace of integration, providing much autonomy to the acquired companies and having appropriate financial incentives for retention of key staffs. However, there are natural trade-offs between autonomy and effectiveness of the transfer of values and best practices, and between slower integration to preserve the knowledge asset and the speed of transfer for these technologies and capabilities to the acquirer. IBM has shifted from a slow integration to a rapid and immediate integration strategy to resolve some of the problems brought about by long period of autonomy and slow pace integration.

In the Java APS software industry, BEA faces great competitive pressures from IBM, which is much larger, has very good account control and has been aggressively pushing WebSphere that has significantly improved over the years. However, Microsoft, with its .NET architecture that competes directly with enterprise Java, may turn out to be the more daunting competitor for both BEA and IBM.

IBM, BEA and Microsoft occupy one of the most attractive technology real estate in the Java and non-Java enterprise software industry – leading market share in application server software – that forms the platform that provides for all the building blocks for application development and management, database connectivity, and network and security functions. IBM and BEA have
reached points of critical mass and any new entrant to the Java APS market is likely to have a difficult time achieving a significant level of market acceptance.

In conclusion, we have learned that for successful companies, the corporate, technology and business development strategies must be closely aligned. Acquisition provides companies with a means to acquire new capabilities and enter new markets and the success depends not only on the preservation of culture and environment of the acquired firm but also the alignment with the corporate vision and strategy for optimal long-term results.

**Table 6-3: Comparison of technology strategy issues for IBM and BEA**

<table>
<thead>
<tr>
<th>No</th>
<th>Issue</th>
<th>IBM</th>
<th>BEA</th>
</tr>
</thead>
</table>
| 1. | Motivation and Rationale   | • Growth through acquisition, capitalizing on IBM’s strengths in brand marketing and global sales channel, is one of the primary strategies for IBM in the software and service business.  
   • Horizontal incremental acquisitions of point technologies that were missing or were weak in IBM’s technology portfolio. | • Can be divided into technology and consulting and educational resource acquisitions.  
   • Horizontal integration in the field of portal, business process workflow automation, security, performance accelerator and development tool.  
   • Vertical integration to remove dependency on third party JVM.  
   • Delivery of its integrated tool strategy. |
| 2. | Impact on technology portfolio | • Does not shy away from acquiring technologies that it already possesses if the acquisition improves IBM’s competitiveness or revenues.  
   • Has the broadest horizontal APS software portfolio including end-to-end performance management, | • No overlap in technological expertise in the acquisitions.  
   • Acquired technologies are subsequently integrated into a cohesive platform. |
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| 3. | Penetration of new markets | * Lotus and Tivoli were motivated by IBM’s corporate strategy to enter the fast growing middleware market.  
  * Embarks on “opportunistic consolidation” with objective of acquiring installed based of acquired company and then up-sell or cross-sell. | * Tuxedo and WebLogic formed the core acquisitions for successful entry into middleware and Java application server market respectively.  
  * The results of acquisitions in the marginal and familiar/unfamiliar sectors are mixed. |
| 4. | Organizational challenges | * With its sometimes overlapping technology acquisitions, IBM is trying to reconcile products and technologies from its own labs and the diverse companies it has acquired. | * In core technology acquisitions, BEA tries to preserve the culture and environment of the acquired company.  
  * The various research centers have been established with the center of R&D in Seattle. |
| 5. | Change in technology strategy | * Shifted from slow to rapid and immediate integration.  
  * Leveraging on VC community to source for potential business opportunities and acquisition candidates. | * Core framework is already established and BEA does not foresee future major acquisition. |
| 6. | Overall assessment | * The focus on acquisitions in the base/familiar sectors of the Roberts/Berry matrix yields positive financial results.  
  * Till date, growth through acquisition has been successful in generating new revenue stream. However, this | * Primary objective of acquisitions are for market entry and horizontal technology acquisition instead of purely appropriating the market of the acquired firm.  
  * Forays into the marginal sectors of the Roberts/Berry matrix |
| | strategy may have its limit as the potential pool of acquisition target shrinks over time. | yields mixed business success. |
| | • IBM has acquired the broadest horizontal APS software portfolio. However, with its sometimes overlapping technology acquisitions, IBM is trying to reconcile the products and technologies and evolve them into a more integrated platform. | • Despite the many technology acquisitions, BEA is fairly successful in integrating them into a unified architecture. |
7. Analysis – Microsoft versus BEA

7.1. Introduction

Being the platform of choice, such as Microsoft in desktop PC software, Palm in handheld computing devices and NTT DoCoMo in mobile services, that brings with it huge financial returns is the Holy Grail of high-tech industries. There are many things that platform leaders and platform-leader wannabes can learn from Intel in its effort to build communities of customers and third parties who are interested in building complements or offering complementary services. The four levers, proposed by Gower and Cusumano, which all platform leaders need for designing and implementing a successful platform strategy are:

1) Determine the scope of the firm: Is it preferable to create product complements internally or let the “market” produce them?

2) Design product technology strategically: What degree of modularity is appropriate? Should product interfaces be open or closed? What information should leaders disclose to outside firms?

3) Shape relationships with external complementors: How can the company balance competition and collaboration with outside players?

4) Optimize internal organization structures: What processes and systems will allow the company to manage internal and external conflicts of interest most effectively.

This chapter provides a comparative analysis of the platform leadership strategies and their impact on the platform adoption of Microsoft’s .NET and BEA’s Java APS using the above Four Levers framework. In Section 7.2 and 7.3, I present the corporate, business and technology strategies, and an analysis of the platform strategies adopted by Microsoft and BEA respectively. In Section 7.4, I compare and contrast the strategies of Microsoft and BEA and draw conclusions for developing platform leadership in the APS software industry.

7.2. Microsoft

7.2.1. Corporate and Business Strategy

Microsoft’s mission is to enable people and businesses throughout the world to realize their full

\footnote{2003 Microsoft Annual Report.}
potential, and its vision is empowering people through great software – any time, any place, and on any device. And one of its key business value propositions is:

Creating opportunities through innovation
At Microsoft, focusing on the needs of customer fuels our passion for innovation. We believe innovative technology has the power to eliminate obstacles and create opportunities. Opportunities for people and businesses to realize their full potential. But there remains a great deal of work to make technology more affordable, easier to use, and less complex. Discovering new ways to help individuals be more productive and organizations more efficient is the inspiration for our investments in research and development. Delivering technology that enhances everyday life – at work and at play – is the core promise behind our comprehensive, integrated software platform. And since we know we can’t accomplish these goals alone, we’re building relationships with the world’s largest ecosystem of developers, channel partners, and hardware manufacturers who share our vision. Everything we do begins with a focus on customers. Every idea. Every solution. Every line of code.

Some of Microsoft’s key thrusts in its business plan are: (1) Provide breakthrough, high-quality integrated innovation; (2) Deliver best-in-class responsiveness to customers; (3) Make our platform the best for developers; and (4) Deliver simple, high-value experiences and services.

(1) Integrated innovation
Microsoft believed that customers will find enormous value in a truly integrated, affordable software platform that takes the complexity out of computing. To continue to strengthen its value proposition, Microsoft focuses on the following objectives:

- Continue to advance the development of an end-to-end technology platform that seamlessly integrates hardware and software from the enterprise to the desktop to wireless mobile devices;
- Address security, manageability, and reliability;
- Provide high-value and easy-to-use technologies that support customer scenarios “right out of the box” – with little or no customization;
- Achieve strong integration between our products and services, and applications on other systems, through standards of interoperability and XML Web Services; and
- Ensure that we protect and derive value from the intellectual property we create.

(2) Responsiveness to customers
Microsoft thinks that providing value to customers means not only building great products, but also listening carefully to customers, responding quickly, and being more transparent and accountable. And in response to customers’ concerns about product quality, Microsoft initiated a companywide drive for Trustworthy Computing, aimed at constant improvement in reliability, security, privacy, and business integrity.

(3) Supporting developers
The success of Windows derived in large part from its support for the broadest array of applications. Microsoft has a long history of providing programmers with state-of-the-art tools and services that make it easy to develop for Window's platform. It provides developers with an advanced and complete set of software development tools, enabling them to greatly improve their productivity with a single programming model for PC, mobile, and Web applications, and for building Web Services with the .NET Framework.

(4) Simplicity, value, understanding
One of Microsoft’s primary objectives in the innovation of the integrated platform is to provide simpler, more compelling, complete, and targeted customer experiences. Microsoft continues to add more capabilities to existing products, and segments its product offerings in order to deliver the simplest, most valuable package of its technologies for each specific customer segment at the right price.

7.2.2. Growth Strategy
Many companies fail to make the difficult transition from one generation of technology to the next and sustain its excellence and leadership. In the book “Microsoft Secrets” by Cusumano and Selby, the author identified two key principles that allow Microsoft to sustain its platform leadership over more than one generation of products and extend the dominance to new markets: (1) Continual cycle of incremental and occasionally more radical innovations and (2) Continually integrating,

linking, repackaging and often simplifying its products.

(1) Continual cycle of incremental and occasionally more radical innovations
Microsoft frequently makes incremental improvements, through packaging of many incremental innovations, and occasionally introduces major advances that makes older product version obsolete. With continual improvements, competitors have little opportunity of challenging the market leader. Moreover, Microsoft has accumulated enormous financial and technical resources to sustain this level of R&D.

Through the introduction of occasional radical changes and the manipulation of the standards – that it controls – that defined some areas of compatibility, Microsoft “forced” its customers to upgrade their applications and continue to use Microsoft’s software assets as Windows evolved. This technical lock-in was essential to Microsoft’s continued domination of the desktop PC software platform. Microsoft reconciled its commitment to applications compatibility with its need to generate revenue from upgrade sales by maintaining “backward compatibility” but did not always guarantee “forward compatibility”. The author cited the example where users of programs written for Windows 95 or later versions could generally read files and run programs written for Windows 3.1 or DOS. However, users who were still running Windows 3.1 or DOS could not run applications designed for Windows 95 or newer versions. Similarly, if a user wrote a report using Office 97, a colleague using Office 95 could not read the report unless the Office 97 user had saved them in the old file formats.

(2) Continually integrating, linking, repackaging and often simplifying its products.
Microsoft is continually integrating, linking, repackaging, and often simplifying its products. The main objectives are to enter new markets with products that combine multiple functions that were once separate, and to make products more accessible to broader sets of users. This pattern of competition and innovation has enabled Microsoft to extend its reach to the enormous mass markets of the computer novice and the home consumer. Microsoft is leveraging its existing technologies and products as well as impressive capabilities in new product development. It is entering new but related mass markets by creating linkages among products and taking advantage of a vast customer network.

Cusumano and Selby identified four key Mass Market strategies that help Microsoft sustain its growth. First, Microsoft has aggressively moved from one mass market and one technology
generation to another through linking of products and technology, relationship with hardware and software retail vendors, and similar marketing practices to promote new products and new product generations. Second, it creates standards that generate new markets and new demand which enhances the growth opportunities for Microsoft as long as it competes in these new arenas. Third, it expands aggressively to avoid dependence on saturated markets such as moving from programming language to operating systems and stand-alone desktop productivity applications, and then to various types of office and networking software, home-consumer products, multimedia publishing, and on-line products and services. And lastly, it continues to blur distinctions among different layers and types of software to preempt competition and expand market shares by combining and integrating features to make it easier for Microsoft to market products and attract large numbers of new customers while at the same time help to retain the existing base of customers.

7.2.3. Technology Strategy

To deliver the business value proposition of providing customers with a truly integrated, affordable software platform that takes the complexity out of computing, Microsoft invests heavily on R&D innovation. It plans to spend $6.8 billion, about 20% of FY03 revenues, on R&D innovation in FY04104.

Microsoft’s technology strategy can be summarized into the following 3 key points: (1) Heavy reliance on internal development for maximum control; (2) Shift from client-based to server-based computing in the Internet era; and (3) Market entry through acquisitions.

(1) Heavy reliance on internal development for maximum control105

Most of Microsoft’s software products are developed internally, although it also purchase technology, license intellectual property rights, and oversee third-party development and localization of certain products. Internal development allows Microsoft to maintain closer technical control over its products and gives them the freedom to designate which modifications and enhancements are most important and when they should be implemented.


105 Ibid.
(2) Shift from client-based to server-based computing in the Internet era

The advent of the internet presented huge challenges to Microsoft’s dominance of the desktop software platform and applications business. First, it is possible for users to access the Internet through non-Windows PCs and workstations, as well as through non-PC based devices such as PDAs and cell phones. Second, it was possible for companies to host applications that can be accessed through browsers running on any operating system and eliminates the need for Windows and Microsoft’s desktop applications. As a result of this broad threat to its business, Microsoft in mid-2000 initiated a five-year project to evolve from the client-based centric to server-based centric computing through the development of Microsoft .NET. The meaning of .NET has evolved over time as Microsoft adjusted its strategy and the current definition\textsuperscript{106} is defined as:

\begin{quote}
Microsoft .NET is our strategy and implementation of connecting people, information, systems and devices through the use of Web Services. It includes everything needed to develop and deploy a Web Service-connected IT architecture: servers to host Web Services (Windows Server System and Windows Server 2003), development tools to create them (Microsoft Visual Studio .NET 2003 and the .NET Framework), applications and smart devices that use them (Microsoft Office System, smart phones, Pocket PCs and PCs), and a worldwide network of more than 35,000 Microsoft Certified Partner organizations – people whose skills and experience can help businesses get the most from their IT investments. Built on industry standards, Web Services enable applications to communicate and share data over the Internet or an intranet, regardless of operating system or programming language.
\end{quote}

(3) Market entry through acquisitions\textsuperscript{107}

Microsoft pursued a strategy of following other firms and incrementally introducing innovations. Over the years, Microsoft acquired many companies (though usually small companies) to enter new emerging markets. Some examples are DOS, PowerPoint, FrontPage (a tool for designing Web pages). Even for the new Java-like programming language for .NET, named C# (pronounced “C sharp”), Microsoft adapted many key features from Java.

\textsuperscript{106} Ibid.

In recent years, Microsoft made some large acquisitions to enter the applications market – which include customer-relationship, human-resource, and supply-chain management software – for small business, a $19 billion market, through the acquisitions of Great Plains Software for $1.1 billion in April 2001, and Navison for $1.3 billion in May 2002.\(^{108}\)

### 7.2.4. Analysis

The analysis below is largely based on the analysis of Microsoft’s strategies in achieving platform leadership, conducted by Gawer and Cusumano in the book “Platform Leadership”\(^{109}\), supplemented by other sources of information.

(1) **Determine the scope of the firm**

Microsoft continues to dominate the desktop software platform through the building of a huge complement ecosystem around Windows. However, as discussed previously under Microsoft’s Growth Strategy, Microsoft not only stimulated the growth of external complements but also competes aggressively in the new markets in order to avoid dependence on saturated markets and to generate demand and ensure the success of future generations of the Windows. Unlike Intel, Microsoft’s entry into the complements business is substantial. In FY03, the contribution from applications (Information Worker segment: $9.3 billion) is about the same as that from desktop Windows (Client segment: $10.4).

Similar to its client strategy, Microsoft applied the same strategy, of blurring the distinctions among different layers and types of software, in the server side. Microsoft developed its own suite of products including portal, content management, database, security and integration, and most significantly, into the business applications realm to compete with SAP, PeopleSoft and Oracle in their lower-end SMB market.


(2) Design product technology strategically

Architecture

Unlike Windows, a largely proprietary and closed standard, Microsoft is trying to leverage on open standards, such as Web Services, in .NET for interoperability and communications over the Internet with systems based on other technology such as J2EE. Despite the adoption of open standards, Microsoft did not abandon its attempts to “embrace and extend” the Internet and draw it closer to the Windows software platform – unlike Java which runs on multiple platforms, applications written for .NET only run on Microsoft operating system.

Microsoft also deviated from past practices of often setting its own standards and expected the industry to follow. The new emerging Web Services standard is in fact first proposed by Microsoft to the neutral World Wide Web Consortium. IBM decided to support the Web Services standards after Microsoft incorporated some IBM suggestions\textsuperscript{110}. Web Services is fast becoming a widely supported open standard.

Interfaces

Microsoft has been found to use its Lever 2 strategy (control platform interfaces and technical information) to gain market advantage in those complementary markets that it had decided to enter (Lever 3). Although Microsoft’s DOS and later the Windows programming standards are proprietary, Microsoft considered them as open because it distributed specifications and programming information for free or a minimal charge to companies that it considered complementors. However, the Windows programming interface standards were not open in that Microsoft – not any group of companies or a standards body – controlled their design and future evolution. Microsoft has been found to withhold technical information from firms that it considered competitors. As a result, developers outside the company faced disadvantage in developing products that integrate seamlessly with Windows versus Microsoft’s internal product groups.

One aspect of interface that has worked against Microsoft is the need to maintain “backward

compatibility", for example the ability to read files made using older versions of the applications. The commitment to backward compatibility is highly valued by users and is an essential element of Microsoft’s growth strategy. However, this commitment prevents Microsoft from making radical changes to Windows even though it control the interfaces, constraining Microsoft into making incremental and evolutionary innovations.

(3) Shape relationships with external complementors

Consensus and control
Since the Windows programming interface standards were controlled by Microsoft, it enables Microsoft to move relatively quickly compared with technologies that are controlled through standards body which need to seek consensus from various parties before the acceptance of any specifications.

Collaboration and competition
As described under Microsoft’s growth strategy, Microsoft continues to expand its market scope into any mass market business of large market potential and compete with its software and even hardware complementors. The company might also try to acquire firms that made an important complementary technology or potentially threatening substitute technology.

Microsoft has placed a lower priority on gaining and maintaining the trust of its complementors over its priority of protecting the exclusivity of Windows software platform, particularly software applications complementors. One of the major reasons highlighted by Gawer and Cusumano is that Microsoft has the financial and technical capability to create its own software complements, if necessary.

(4) Optimize internal organization structures

Structure
Microsoft application developers had an indirect advantage over its competitors by having direct access to detailed Windows roadmap for planning and to Windows developers for technical consultation. They also better understood the intricacies and idiosyncrasies of the windows operating system. Gates and other executives have openly encouraged the movement of people and sharing of technical knowledge across the different product groups. It is not unusual for one
Microsoft executive to head both the systems and applications divisions.

Microsoft long maintained that it did not have a “Chinese wall” to separate its operating systems group and applications group. In one interview with Gates\textsuperscript{111}, he said that “\textit{We don’t block input going in either direction}”. One of Microsoft’s fundamental arguments for “integration” of different applications, systems, and networking technologies is that it is good for customers as it reduces complexity and minimizes TCO.

\textbf{Process and culture – Focus on security}

One of the key weaknesses of Microsoft’s style of product development, the synch-and-stabilize\textsuperscript{112} and feature-driven approach, is that it underemphasize the importance of the underlying product architecture. Without a good architecture to define how components interact, it can become extraordinarily difficult and time-consuming to develop, re-work, test, and debug all the different features that teams add or change. This may be one of the major causes of the numerous security flaws that appear over time in Windows software platform, especially the client Windows operating system.

As a result of the cost of virus attack to enterprises, Microsoft has realized the importance of security over features to its future success and initiated a companywide Trustworthy Computing initiative, aimed to ensure a safe and reliable computing experience for customers that is both expected and taken for granted.

One sign that Microsoft’s initiative has begun to pay off is the relatively low number of flaws uncovered in its latest Windows Server 2003 so far, compared with Windows 2000 at the same stage. As at Jul 2003, just four security bulletins have been released for Windows Server 2003


\textsuperscript{112} The essence is to continually synchronize what people are doing as individuals and as members of different teams, and periodically stabilize the product in increments as opposed to doing once at the end.
compared with 14 for Windows 2000 in the same period\textsuperscript{113}.

However, it remains to be seen to what extent can Microsoft eliminates the security flaws. There is a recent news article\textsuperscript{114} covering the release of Microsoft fixes that cover at least 20 Windows flaws, several of which could make versions of the operating system, including the latest Windows XP and Windows 2003, vulnerable to new worms or viruses. In the article, Marc Maiffret, chief hacking officer for eEye Digital Security, claims that Microsoft took as many as 216 days to fix the latest set of flaws, an indication of a tightly coupled product architecture that makes identification and fixing of bugs difficult and time-consuming. Gerhard Eschelbeck, chief technology officer for vulnerability assessment company Qualys, said: “A lot of the flaws in this release are derivative of ones that we have seen before. Typically, someone finds a flaw in a particular area and a lot of researchers start looking in that code.” It makes you wonder what else is in the millions line of codes.

“System” mindset and neutrality

Microsoft’s dual role as both a platform leader and substantial complements producer reduces the effectiveness of Microsoft’s execution as a neutral broker, a critical role that a platform leader undertakes, between external party and the internal group in co-development tasks. Microsoft addressed these internal conflicts by manipulating Lever 4 – the internal organization. Gates placed well-respected executives at the top of both the software platform and applications businesses and got directly involved in resolving disputes when necessary\textsuperscript{115}.

However, Gates and other Microsoft executives did not completely succeed in resolving internal conflicts. Prior to deciding on .NET, a number of senior Microsoft executives wanted the company to introduce a cross-platform framework that is able to run on multiple operating system, similar to


the cross-platform Java programming language. Gates vetoed the proposal as this would have undermined directly the importance of Windows and against Microsoft’s fundamental strategy of building around Windows. As a result, several key executives and engineers left the company during 1998 to 2000. Although the organization seemed to be rallying effectively around the .NET strategy, the departures of talented people did not bode well for Microsoft’s future. In fact, the group that left and setup CrossGain, a company focusing on XML, was later acquired by BEA and became one of BEA’s strategic acquisition in helping BEA to develop an integrated platform suite that has the potential of competing with Microsoft’s fundamental competitive advantage against J2EE, simplicity and the ease of development.

7.2.5. Summary
In summary, Microsoft’s fundamental value proposition is to make technology more affordable, easier to use and less complex by developing comprehensive and integrated software platform, through the building of an ecosystem with developers, channel partners, and hardware manufacturers who share its vision. Notice the absence of application complementors in the ecosystem. Microsoft placed a lower priority in gaining and maintaining the trust of its application complementors as it has the financial and technical resources to create its own software complements.

The objective of Microsoft is to build an end-to-end integrated technology platform that spans from the enterprise to the desktop to wireless mobile devices and is able to interoperate with systems based on non-Microsoft technology through XML Web Services.

Microsoft attempts to sustain its growth and platform leadership through (1) Continual cycle of incremental and occasionally more radical innovations and (2) Continually integrating, linking, repacking and often simplifying its products, and the manipulations of standards to “force” its customers to upgrade to its latest products version.

Its business entry strategy is usually to acquire small companies with emerging complementary or substitution technologies, not necessary mature ones, of large market potential, and then incrementally enhances them to achieve market leadership. Similar to its client strategy, it has also

116 J. Greene, “Microsoft’s Big Bet,” Business Week, 30 October 2000, 162.
entered into the applications market at the server end through the acquisitions of Great Plains Software and Navison.

The result of Microsoft's business diversification strategy is aligned with Roberts and Berry's Familiarity Matrix business entry framework. Microsoft's entry into related product-markets in the mass market is highly successful while its new entry into the new unfamiliar market of building business applications for enterprise is not very positive: Microsoft suffered a $176 million loss for revenue of $308 million in FY02 and a $254 million loss for revenue of $567 million in FY03. But the ratio of loss has decreased from 57% (176/308) in FY02 to 45% (254/567) in FY03, an incremental improving trend typical of Microsoft's entry into new markets before it achieves market leadership.

Microsoft has also started to embrace open standards while at the same time trying to bring the Internet closer to Windows. It continues to control Windows interfaces strategically and is able to make .NET runs optimally on Windows compared with Java on Windows. With the evolution of .NET under its direct control, Microsoft will be more agile compared with companies that rely on Java, which is controlled by Sun but managed under the Java Community Process (JCP).[^117]

### 7.3. BEA

#### 7.3.1. Corporate and Business Strategy

To recap, BEA's vision is

> "Our vision is to provide the enterprise software platform that enables our customers to execute most effectively on their vision. And our success is increasingly powered by long-term partnerships rather than short-term product sales."

And BEA has the following corporate and business strategies: (1) Customer-focused strategic planning and product development, (2) Building a ecosystem through strategic relations with key partners and building of developer community, (3) Sales and marketing to both senior executives

[^117]: The Java Community Process (JCP) is an open, participative process to develop and revise the Java technology specifications, reference implementations, and test suites in cooperation with the international Java developer community.
and information technology department personnel, (4) Maintaining its technology leadership, (5) Continuing its standards leadership, and (6) Staying focused and remain as an independent and pure-play software vendor.

7.3.2. Growth Strategy
There are two key principles that drive the growth of BEA application server in the past and will continue to play key roles in driving the adoption of its new Java APS: (1) Scale through indirect sales force and (2) Build ecosystem through alliances.

(1) Scale through indirect sales force
As a company with only 3,100 employees, compared with IBM which has more than 316,000 employees worldwide, BEA has to rely on partners such ISVs, system integrators and value-added resellers, and consulting firms to increase the market awareness, demand and acceptance of BEA and its solutions.

(2) Build ecosystem through alliances
BEA has about 700 engineers working on its products compared with 8,000 engineers working on IBM WebSphere application server products. 118 Its annual R&D budget of about $120 million is a tiny fraction (about 2%) compared with the $3 billion-plus in annual R&D spending on .NET by Microsoft. To compete effectively in the infrastructure software market, in which both IBM and Microsoft have broad product breadth, BEA has to build many alliances in order to provide comprehensive solutions to enterprise customers.

7.3.3. Technology Strategy
BEA acquired key missing technologies fundamentally through acquisitions of small companies and then reengineers the acquired technology from ground up to fit well into the core platform.

The objective is to develop a Unified Application Platform that removes the complexity and risk of integrating the various infrastructure components, and reduce the effort of developing and

deploying applications. The vision of the new BEA architecture team – led by Adam Bosworth, the former Microsoft technology architect amounts to the following principles\textsuperscript{120}:

- **Integration and development become one.** "Most projects see integration and development as equally essential elements of the process. BEA will deliver the platform where integration and development are organically fused."

- **Mass-market J2EE: Make a leap in J2EE ease of use.** "J2EE is too hard. BEA will deliver a development environment to offer the best of both Microsoft and J2EE worlds: ease of use and platform independence — at the same time."

- **Make service-oriented architecture (SOA) the foundation for BEA-based applications.** "SOA design is too complex for mainstream software projects. WebLogic Workshop (WLW) will use the SOA concept as the core of its vision for application architecture. Combined with its vision of ease of use, WLW will bring SOA to mainstream software projects."

- **Use the tool to sell the platform.** "A J2EE application server alone is not sufficient to meet all software project requirements. The development process must be equally comfortable with design of components and services, design of multi-channel user interactions and design of integrated processes. Such an integrated tool will sell its underlying integrated middleware to a massive new pool of developers."

### 7.3.4. Analysis

(1) **Determine the scope of the firm**

As described above in BEA's growth strategy, the reliance on partners as the indirect sales force and software complementors to provide comprehensive solutions due to its limited sales force size and product breadth compared with other software giants many times its size.

In the interview with the BEA executive, he said that since BEA's core framework is already established and will not be changing the core architecture dramatically, BEA is expected not to make major acquisition in the near future but will continue to build strategic alliances with key partners to reinforce its product offerings.

---

However, as the industry matures further with business application vendors (SAP and Oracle) moving into infrastructure business and infrastructure vendors moving into business application business (Microsoft), even broader horizontal integration of the APS is expected, including performance management, security, database management, development process management, and operations management.

(2) Design product technology strategically

Architecture
BEA, or in fact all J2EE vendors, uses the product platform approach in developing their product offerings based on J2EE specifications.

So what is J2EE? The J2EE is a set of coordinated specifications and practices that together enable solutions for developing, deploying, and managing multi-tier server-centric applications\(^\text{121}\). The J2EE platform uses “containers” to simply development by enabling the separation and shielding of the business logic from the underlying infrastructure complexity and changes. Adapted from the Generalized Architecture of a Software Product Platform presented in Section 5.7 (see Figure 7-1), the “containers” is equivalent to the box surrounding the add-in modules. And the J2EE specifications specify the standard way for the container to interact with the add-in modules as well as to interact with the underlying components within the “Engine”. The task of the many J2EE vendors is to provide the implementation for the container and the underlying components in the “Engine”, either internally or through third party components. The vendors are free to compete on implementations but not on standards or APIs.

Figure 7-1: Generalized Architecture of a Software Product Platform: To explain J2EE concepts


Interfaces
One of the key strength of BEA is the ardent support of the open-standards to develop frameworks that provides robust interfaces that enable (1) BEA to evolve with and implement new standards effectively and (2) To integrate effectively with many third party component providers. Two of the key frameworks are WebLogic Security Framework that allows BEA to integrate with specialized security solutions from major security vendors, and the Network Management and Monitoring Framework that allows integration with major infrastructure management vendors.

In Sep 2003, BEA launches a “Developer Extensibility Program” to help ISVs easily and quickly
integrate with BEA WebLogic Platform\textsuperscript{122}. As of Sep 2003, thirty-three industry leading ISVs, across various markets such as application management, business intelligence, content management, enterprise instant messaging, and packaged applications, have signed up to offer BEA WebLogic Workshop extensions controls – reusable software components that can unite development, integration and third-party software assets through a single development environment.

(3) Shape relationships with external complementors

Consensus and control
As a member of the JCP, a community that manages the evolution and standardization of the Java specifications, BEA will have to work closely with members of the Java community, and especially IBM and Sun, to effectively evolve the standards to sustain its competitive advantage.

Collaboration and competition
The importance and reliance on partners is critical to BEA and BEA strived to nurture, attract and retain complementary partners.

The Developer Extensibility Program provides ISVs with free software, technical resources, testing services and marketing support to help make it easier for ISVs to reach millions of Java developers and accelerate the adoption of their applications within the BEA customer base. The program also provides ISVs with a standard infrastructure for building and integrating applications and technologies within BEA WebLogic Platform. The enhanced BEA’s offering which provides a broad horizontal suite of products further attract more customers, and result in increasing growth for both BEA and its partners.

The wide spectrum of alliances with many partners, both large software vendors and as well as smaller specialized firm, in various parts of the software value chain is one of the key strengths of BEA. Table 7-1 shows some of BEA’s alliances in various domains.

Table 7-1: BEA’s wide spectrum of alliances

<table>
<thead>
<tr>
<th>Area</th>
<th>Partner</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J2EE Standards</td>
<td>IBM</td>
<td>IBM and BEA Systems, usually staunch rivals, are collaborating on ways to smooth out technical differences between their respective Java software lines in response to request by customers and ISVs to create more common technical underpinnings for their Java server software(^{123}).</td>
</tr>
<tr>
<td>Web Services Interoperability (WS-I) standards</td>
<td>Microsoft</td>
<td>Customers use BEA to create Interoperable Web Services between Java and .NET. As a founding member of the WS-I organization, BEA is committed to enabling developers to build applications that can easily interoperate with applications deployed on other platforms, including .NET(^{124}).</td>
</tr>
<tr>
<td>Linux</td>
<td>Red Hat and Novell</td>
<td>Global Linux leaders choose BEA WebLogic JRockit (JVM product) to deliver superior price and performance. The alliances with Red Hat and Novell to co-develop solutions that bundle BEA WebLogic JRockit with their Linux solutions to help customers build and deploy a cost-effective, high-performing application infrastructure further cement BEA’s role as the leading global provider of JVM for the booming Linux operating system(^{125}).</td>
</tr>
<tr>
<td>HP Itanium Server</td>
<td>HP and Intel</td>
<td>HP and BEA optimize BEA WebLogic Server for HP Itanium-based solutions. BEA and HP announced the availability of BEA's WebLogic Server on HP's Itanium-based (co-developed by HP and Intel) server platform, jointly providing a proven</td>
</tr>
</tbody>
</table>


\(^{126}\) BEA Systems, Inc, “HP and BEA Optimize BEA WebLogic Server for HP Itanium 2-based solutions,”
<table>
<thead>
<tr>
<th>Utility Computing</th>
<th>Veritas</th>
<th>BEA, Veritas to tune tools in utility alliance. BEA and Veritas have penned a partnership designed to further align the two companies around utility computing. The alliance calls for the two companies’ engineering groups to collaborate to make their respective products work better together in corporate data centers. The partnership will also involve joint marketing and sales arrangements to foster cooperation between the two organizations’ sales forces.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Wily Technology</td>
<td>Wily Technology Announces Completion of BEA Validation Program. Wily technology announced that its market-leading application management solution has completed the BEA Validation Program and is integrated with BEA WebLogic Platform.</td>
</tr>
<tr>
<td>Management</td>
<td>Technology</td>
<td></td>
</tr>
<tr>
<td>Mainframe</td>
<td>GT Software</td>
<td>GT Software Completes BEA Validation Program. GT Software, a leading provider of mainframe integration solutions, announced that GTConnect has completed the BEA Validation Program and is now verified to integrate with BEA WebLogic Workshop (unified development product). GTConnect is designed to provide a fast path to mainframe information for BEA WebLogic developers.</td>
</tr>
<tr>
<td>Integration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Portal</td>
<td>Accenture</td>
<td>BEA, Accenture build portal package. BEA and Accenture are</td>
</tr>
</tbody>
</table>


joining forces to help enterprises streamline a jumble of Web sites into fewer, and better structured Web portals\(^{130}\).

(4) Optimize internal organization structures

"System" mindset and neutrality
The focus and positioning to remain as an independent and pure-play application infrastructure software vendor has provided BEA some competitive advantage. First, application independence makes BEA an attractive partner to application vendors – Packaged application vendors are more likely to support BEA than some of its competitors that offer packaged applications of their own. Second, systems integration independence makes BEA an attractive partner to systems integrators – Systems integrators are more likely to support BEA than some of its competitors that offer systems integration services of their own.

7.3.5. Summary
In summary, BEA's growth principle is to rely on partners to drive sales and to augment its own offerings in order to provide comprehensive solutions to enterprise customers. The inherent nature of J2EE and the robust implementation frameworks developed by BEA to implement the J2EE specification allow BEA to timely and effectively evolve their products to keep pace with the rapid technological and market developments, and emerging industry standards\(^{131}\). The frameworks together with the Developer Extensibility Program, enables BEA to effectively integrate and support many external market-leading component providers so as to build an attractive ecosystem. The co-marketing and co-selling with market-leading vendors help to drive the adoption of BEA as well as the partners’ products.

The key strengths of BEA are its engineering capability, wide spectrum of partnership, and the focus and positioning to remain as an independent and pure-play application infrastructure software vendor. These are important factors for building a large and healthy ecosystem.


\(^{131}\) BEA Systems, Inc, 2002 Annual Review.
7.4. Summary

From the study, there are great similarities and at the same time great differences in the platform leadership strategies of Microsoft and BEA. The comparison of the two companies with respect to the four platform leadership levers is summarized in Table 7-2.

The value proposition of both Microsoft and BEA is to make technology more affordable, easier to use and less complex by developing an integrated platform suite through a continual cycle of incremental and occasionally radical innovations to integrate, link, package and often simply their products.

Also, the technology acquisition strategies are very similar in that both usually acquire small companies with emerging technologies, not necessary mature ones, and then integrate the acquired technologies into the integrated product suite through ground up re-development if necessary to ensure seamless integration.

However, there are major differences in the execution. While both companies attempt to build integrated platform, the objective of Microsoft is to build an end-to-end integrated technology platform that spans from the enterprise to the desktop to wireless mobile devices and covers both the applications and the infrastructure. BEA, on the other hand, is a pure-play application infrastructure software vendor focusing only on the core infrastructure components and relies on partners to augment its product offerings.

Although Microsoft has also started to embrace open standards, its .NET platform only runs on Windows operating system. In contrast, the Java platform is architecturally separated from the operating system and is thus able to run on multiple operating systems. Thus, .NET applications can only run on servers, PCs and devices installed with Microsoft’s operating system whereas Java applications can run on almost any hardware appliances installed with various types of operating systems.

In conclusion, although Microsoft and BEA compete aggressively to be the preferred application development platform standards for enterprise and ISVs, the major threat may actually be something else. Microsoft and BEA face the constant challenge from IBM which has the market power to change the dynamics of the market and has the mindshare of large enterprises. However, the critical threat may actually be from SAP attacking from the top and open-source attacking from
below. SAP, with its recent entry into the infrastructure software market, might capture a large infrastructure software market share from the existing enterprise customers who are already committed and entrenched into SAP applications. At the other end of the spectrum, the SMB, the increasing popularity and adoption of open-source threatens the margin of the low end market and thus the profitability of commercial software companies.

Table 7-2: Comparison of platform leadership strategies of Microsoft and BEA

<table>
<thead>
<tr>
<th>No</th>
<th>Lever</th>
<th>Microsoft</th>
<th>BEA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Determine the scope of the firm</td>
<td>• Do not hesitate to enter complements market with large market potential.</td>
<td>• Reliance on partners as the indirect sales force to support wider expansion.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Compete significantly in complements market with revenues about the same as the platform revenue.</td>
<td>• Reliance on software complementors to augment product offerings to provide comprehensive solutions of customers.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Continues to blur the distinctions among different layers and types of software.</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Design product technology strategically</td>
<td>• Largely proprietary and closed standard.</td>
<td>• Inherently a product platform approach that shields the add-in modules from underlying changes in technology.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• However, rely on open standards such as Web Services for interoperability with systems based on other standards.</td>
<td>• Robust interfaces enable BEA to effectively evolve with and implement new standards and to integrate effectively with many third party component providers.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Deviated from past practices of setting its own standards and expecting the industry to follow. Instead, it promoted Web Service standards through submission to standards body.</td>
<td></td>
</tr>
</tbody>
</table>
| 3. Shape relationship with external complementors | • Internal product developers continue to have an advantage over external developers who do not have access to detail Windows roadmap, complete technical information, and direct technical advice.  
• More agility as Microsoft itself controls Windows programming interface standards.  
• Continues to expand market scope into any mass market with large market potential.  
• Placed a lower priority on gaining and maintaining trust of complementors over protection of Windows dominance. | • Wide spectrum of alliance with many partners in various parts of the software value chain.  
• Developer Extensibility Program helps ISVs to integrate with BEA platform and expand the market for both the partners and BEA through co-marketing and co-selling. |
| 4. Optimize internal organization structures | • Encourages sharing between application and platform group.  
• Shifted focus on security over features as a result of increased cost of virus attack to enterprises.  
• Significantly improved security of latest Windows but remains to be seen on the extent of elimination of security flaws.  
• Dual role as both platform | • Focus and positioning to remain as an independent and pure-play application infrastructure software vendor makes BEA an attractive and trusted partner to application vendors and systems integrators.
| 5. Overall assessment | • Objective is to build end-to-end integrated technology platform.  
• Major expansion into business applications market to replace potential loss of revenues of Client and Information Worker segments as a result of increasing commoditization from open-source Linux Desktop and OpenOffice. | • Primary growth principle is to rely on partners to drive sales and to augment its own offerings.  
• Robust J2EE implementation framework allows BEA to timely and effectively evolve its products to keep pace with rapid technological and market changes.  
• Enterprise customer segment is also under long term threat from commoditization by open-source. |
8. Analysis – Key Market Dynamics

8.1. Introduction

In this chapter, we will analyze the key market dynamics that affects the APS software industry using System Dynamics - Causal Loop Diagrams (CLD). We will first identify the key market drivers and their implications, both from external environment as well as those that are internal to the companies driven from their business and technology strategies. The external environmental drivers are synthesized from the various business and technology trends discussed in Chapter 4. We will then identify some key events or activities that have potential major impact to the future evolution of the industry and analyze their dynamics.

Notes:

- In the CLD diagrams, the key drivers are underlined and in Italic Bold while the key implications are in Italic Bold.
- Only the negative links\textsuperscript{132} are indicated in the CLD diagrams. All other links are positive links\textsuperscript{133}.

\textsuperscript{132} A negative link means that if the cause increases, the effect decreases below what it would otherwise have been, and if the cause decreases, the effect increases above what it would otherwise have been.

\textsuperscript{133} A positive link means that if the cause increases, the effect increases above what it would otherwise have been, and if the cause decreases, the effect decreases below what it would otherwise have been.
8.2. Key Market Drivers – External Environment

8.2.1. Stages of IT Architecture

**Stage 1: Application Silo Architecture Stage**
- Firm invests mostly in functional applications → Need for ease of development and time-to-market. → Attractiveness of rapid application development (RAD) tools. → Adoption of RAD tools.
  - Application silo.
  - Cost of application maintenance.
  - Pressure on profit margin.
  - Senior business managers desire to control cost.

**Stage 2: Standardized Technology Architecture Stage**
- Entry of competitors or the emergence of substitutes or alternatives → Business case for IT focus on ROI of individual applications.
  - Business case for IT focus on integration.
  - Importance of interoperability.
  - CIO's empowerment to establish and enforce technology standards.
  - Needs for manageability.
  - Needs for specialization of server-based over client-based products.
  - Needs for specialized products (catering for niche requirements) to go through an exception process before adoption.
  - Adoption of integrated APS.

**Stage 3: Rationalized Data Architecture Stage**
- Customers' demand of a single point of contact as number of business units grows → Business case for IT focus on improved business performance and integration.
  - Degree of hardwired business.
  - Agility of business.
  - Adoption of package applications.

**Stage 4: Modular Architecture Stage**
- Business need strategic agility to respond effectively for sustain competitiveness. → Business case for IT focus on speed to market and strategic agility.
  - Need for fast time-to-market and flexible architecture.
  - Adoption of SOA.

**Figure 8-1: Stages of IT Architecture**

147
In the first Application Silo Architecture Stage, firms invest mostly in functional applications and the business case for IT investments focus on ROI of individual applications. The productivity and ease of development of Rapid Application Development (RAD) tools drive the adoption of RAD tools in this stage.

In the second Standardized Technology Architecture Stage, the relative inflexibility of application silo and the entry of competitors or the emergence of substitutes that put pressure on companies’ profit margins cause the IT investment focus to shift to integration and cost reduction. These led to the adoption of pre-integrated APS for improved manageability and more importantly, CIO’s empowerment to establish and enforce technology standards to reduce heterogeneity in hardware and software to achieve economy of scale and reduce management complexity and cost. The key implication to software vendors is the shift in IT products standards approval from business unit IT manager to CIO and business executives.

In the third Rationalized Data Architecture Stage, customers’ demand of single point of contact as the number of business unit grows cause the shift in IT focus to improving business performance and integration within the enterprise leading to the adoption of package applications.

In the fourth Modular Architecture Stage, the business need for strategic agility to respond effectively to the faster paced of business brought about by the advent of the Internet in order to sustain competitiveness drives the IT focus towards agility. This makes the inherent flexible and nature of SOA an attractive option.
8.2.2. Increased adoption of open-source software

The five key drivers are: (D1) The use of open-source as a tool to disrupt market; (D2) The emergence of second-generation open-source model based on dual-license mode; (D3) The political and anti-monopolistic factors, especially from countries outside of US, driving the support of
open-source; (D4) The number of windows desktop complements and their degree of locked-in to Windows API; and (D5) Microsoft’s reliance on the SMB market.

First, open-source has been used as a competitive tool by IBM to disrupt the Unix and Windows Server market. This is to reduce the relative power of Sun so that IBM can better drive Java standards and to reduce the relative power of Microsoft in its entry to enterprise software market. SAP also supported open-source database MySQL in its effort to reduce the power of Oracle and Microsoft which compete with SAP in its key business application market.

Second, the emergence of second-generation open-source model helps to address the challenges of community-based traditional open-source model that prevents the wide-spread adoption of open-source software. However, one drawback of this new model is the reliance on the companies own development resources, instead of the entire open-source developer community, to compete with commercial product companies.

Third, political and anti-monopolistic factors will continue to push for the adoption of desktop open-source, putting great competitive pressure on the future profitability of Microsoft’s most profitable Windows Client and Office segments. However, the fourth factor, the existing locked-in to Windows API, acts as a great barrier of entry for desktop Linux with the high switching cost to migrate from desktop Windows to Linux.

Lastly, as a result of Microsoft’s relative focus and reliance on the SMB market, the market most attractive to low-cost open-source alternatives, the degree of impact on Microsoft’s profitability from open-source may be the greatest which will reduce the relative power of Microsoft over its competitors in future.
8.2.3. Antitrust

The three key drivers are: (D1) Microsoft’s existing current account asset; (D2) Existence of competitors with IBM as common “enemy”, and (D3) Microsoft’s compliance of fair disclosure to
First, with the existing large current asset of more than $50 billion, Microsoft has engaged in a series of resolution of long-standing legal and regulatory disputes, so as to be able to focus on other impending challenges and to tap outside innovation through cross-licensing. Second, the existence of IBM as the common "enemy" has even made possible the collaboration with Sun, a former Microsoft's staunch rival, in order to counteract the power of IBM. Lastly, Microsoft's compliance of fair disclosure to competitors on its proprietary interfaces will have major impact on the likelihood of future antitrust action against Microsoft and the improved interoperability between .NET and Java.

8.2.4. Web Services and Service-Oriented Architecture

![Diagram of Web Services and Service-Oriented Architecture]

The adoption of Web Services and SOA will enable easier integration between heterogeneous applications and provides higher level of flexibility and reuse. These will reduce the need for single end-to-end monolithic software packages, facilitate co-existence of .NET and Java environment and allows IT to more closely reflect the way business operates and improve the agility to create new business opportunity and respond to competition.
8.2.5. Summary of key market drivers from external environment

The above diagram summarizes the drivers and implications from key business and technological trends in the APS software industry.
8.3. Key Market Drivers – Internal business and technology strategies

8.3.1. IBM

Two of IBM’s key businesses are IBM Products Group and IBM Global Services. The two key drivers are (D1) Growth through acquisitions and (D2) The needs for integration and manageability of the broad product portfolio.

The growth through acquisitions capitalizes on IBM’s strength in brand marketing and global sales channels to increase sales of newly acquired products (R1). Also, the acquired products contribute to IBM’s product portfolio increasing the attractiveness of IBM’s ability to provide “whole”
product to customers end-to-end needs leading to greater adoption of IBM’s APS (R2). IBM Global Services, with its deep contacts in companies is also able to market and sell IBM’s software products (R3).

However, as the acquisitions continue, organizational and product portfolio complexity leads to challenges in products integration effort affecting the optimal degree of integration (B1). The second key driver of the needs for manageability of IBM’s broad product portfolio leads to the change in acquisition integration strategy from slow to rapid integration, which limits the choice of suitable acquisition target that can be integrated quickly, and thus limits the growth through acquisition (B2). Also, the need for IBM Global Services to provide optimal business value to customers posed the potential conflict of interest that limits the effort of marketing IBM’s products (B3).
8.3.2. Microsoft

Figure 8-6: Microsoft's business and technology strategies

There are many drivers in Microsoft's business and technology strategies that led to its phenomenal growth (see Figure 8-6). Microsoft's financial and technical resources enable Microsoft to have a vision of delivering an integrated end-to-end software platform (R1) and to cultivate a large ecosystem of developers, channel partners and hardware manufacturers around its platform (R2). Its relentless growth effort through diversification into new product area with large market potential, and the bundling with the operating system to provide more complete solution increases the adoption of the new product (R3). Microsoft also makes occasional architectural innovation to provide substantially new capabilities in its effort to drive upgrade sale (R4). Finally, the strategy to ensure backward compatibility but not forward compatibility forces users to upgrade to newer versions to avoid the inconvenience of using older versions of the software (R5).
However, there are also several challenges that Microsoft faces which limit its future growth (see Figure 8-7). First, its strategy of continuous improvement through feature-driven development approach reduces the focus on the underlying product architecture which undermines the robustness of the underlying product architecture leading to the exploitation by hackers to create viruses that wreak havoc across many enterprises which spend huge effort in containing the virus attack (B1). Although Microsoft has initiated a companywide drive for Trustworthy Computing to improve reliability, security and privacy (R7) that has improved the security track record of Windows Server 2003, it remains a challenge for Microsoft to sustain the effort across all its product lines against its fundamental strategy of feature-driven development approach.

Second, the strategy of making occasional architectural innovation to provide new capabilities often times require huge migration effort resulting in increase TCO for companies reducing the
attractiveness of Microsoft’s platform over more stable platform (B2). The huge migration effort resulting from the periodic upgrade of Microsoft’s platform coupled with the security vulnerabilities and effort required to contain virus attacks negatively affect CIOs’ and business executives’ perception of Microsoft that dampens the overall attractiveness of Microsoft’s products.

Third, Microsoft’s large investment in software complements reduces the application complementors trust in Microsoft that limits the number of external application complements (B3). This is especially challenging when Microsoft is trying to extend its dominance from the desktop to the server where it does not yet have a dominating position and requires many more external application complements to fill the product gaps while it builds up its end-to-end capabilities.

Lastly, Microsoft’s continuous improvement and adding of new features (R6) may have caused the overshoot of capabilities over and above what is needed by the majority of Microsoft’s products users resulting in diminishing returns of additional new features. This may set the pre-condition necessary for the success of disruptive technology. Coupled with the expected major migration effort that may be required to move to the next version of Windows, companies may decide to take a big step and migrate to the cheaper Linux desktop alternative (B4).
BEA’s focus on scalability, security, stability and manageability (D5) and the needs to keep pace with technological and market developments (D6) drives the effort to build a robust platform implementation framework. The standards-based product development (D3) and the effort in Developer Extensibility Program (D4) coupled with the modular design of the framework makes third party product integration easier which encourages a large ecosystem of application complements. The focus on application infrastructure (D8) and the partnership with system integrators also encourages the buildup of a large ecosystem of channel complements willing to market and support BEA’s products. The focus on ease of development (D7) and the sales force transition effort (D1) help to drive the adoption of BEA’s APS in the developer’s community and the acceptance of CIOs and executives of BEA’s products respectively.

However, one key weakness is the relative size of BEA compared with IBM and Microsoft (D2) that affects its ability to optimally drive standards. This coupled with the needs for interoperability constrained BEA’s ability to innovate at a faster pace.
8.4. Key Market Dynamics

8.4.1. SAP’s entry into APS software market

D1: Saturation in SAP’s core market in large corporate software overhauls.

D2: Growth in enterprise package business application market.

D3: Competition with Microsoft and Oracle in SMB package business application market.

D4: Conflict with host of partners, especially IBM, BEA and Microsoft, which SAP traditionally relied to make its programs work.

D5: Introduction of a suite-based licensing for NetWeaver components.

D6: Needs to support tight integration of SAP’s business application.

Figure 8-9: SAP’s entry into APS software market

In an effort to expand into new markets, SAP introduced NetWeaver in 2003, to compete with IBM and Microsoft in the APS software market. There are many drivers for SAP’s entry into this market. First, SAP’s core market of large corporate software overhauls may be saturated (D1). Second, as the large companies completed the implementation of major ERP systems, they begin to focus on achieving agility through the use niche package applications and the development of custom applications using SOA to meet their specific business requirements (D2). In the SMB package

---


business application market, SAP faces intense competition from Microsoft and Oracle (D3). All these factors limit SAP’s overall potential growth in the package business application segment which led it to its entry into the application infrastructure software market.

However, there are a few challenges that SAP faces in this endeavor. Its entry into the infrastructure software business puts SAP in direct conflict not only with IBM, BEA and Microsoft, the key players in the infrastructure software market, but also with a host of partners on whom SAP traditionally relied to make its programs work (D4). The conflict may limit the support of partners in positioning SAP’s APS as the core enterprise infrastructure.

Although, SAP’s introduction of a suite-based license (D5) may help drive the adoption of NetWeaver since customers may just use it since they already own it through the purchase of SAP’s business applications, SAP may place its effort on integrating the various components in making NetWeaver a suite rather than focusing on maturing the individual components.

Moreover, the need to support tight integration with SAP’s business application will channel most of SAP’s effort in making NetWeaver to work best and be integrated with SAP business applications (D6).

All these challenges and the fact that the APS software market already has well-entrenched players like IBM, BEA and Microsoft, may result in NetWeaver becoming a part of every SAP application user’s infrastructure but may not be the primary infrastructure provider within the SAP installed based.
8.4.2. IBM's consortium-based open-source strategy

**Figure 8-10: IBM's consortium-based open-source strategy**

**Introduction**

The use of open-source by IBM as an external source of innovation is an important initiative where the dynamics needs to be better understood. It is not only important to create value from external source of innovation but also be able to capture the value created. The following covers IBM's
transition from Closed Innovation to Open Innovation paradigm, the use of open-source as an external source of innovation as well as a strategic tool for developing a development platform to integrate IBM’s broad portfolio of technologies and products, and the dynamics and success of achieving that objective.

IBM’s transition from Closed Innovation to Open Innovation paradigm

IBM has made a transformation from Closed Innovation to and Open Innovation mind-set in its difficult transition from the 80s to the mid 90s. In the Closed Innovation era from 1945-1980, IBM dominated the computer industry and the business model was build on internal innovation, proprietary control over the architecture and all its key elements, and internal integration to provide customers a complete solution to their needs. This model allowed IBM to invest in R&D with confidence, knowing that it would capture a significant portion of the value its R&D created.

By 1992, IBM’s business was facing tremendous competitive pressures on many fronts and IBM gave careful consideration to the idea of splitting into smaller more focused companies to compete with the many niche players entering into its various business segments. However, the new CEO Lou Gerstner, brought in from outside of IBM in Apr 1993, chose not to break up IBM and decided to keep the company together as an integrated corporation.

To maintain IBM as an integrated organization, IBM needed a vision and mechanism to connect the many different parts without the logic of complete control that had pervaded the organization which made IBM become a closed and inward-looking organization. With the advent of the Internet, IBM realized that hardly any of the core Internet technologies revolutionizing the computer industry were coming out of IBM’s corporate R&D labs which reflected a powerful shift in mind-set within IBM. To deliver value to its customers, IBM would need to turn to external technologies, even those that were not created in any particular company’s lab. IBM would leverage technologies that it did not own and could not control – technologies what were open to its competitors as well as to IBM. However, to create and capture value from external innovations, IBM would have to identify the best technologies out there from whatever source and integrate the technologies into effective solutions since it cannot make money directly from these external technologies themselves.

Open-source as a source of external innovation

IBM started the use of open-source as a source of external innovation with the support and marketing for Apache and Linux. It managed to capture great value by gaining a large portion of the Linux server market share. More recently, IBM started an open-source consortium-based project called Eclipse in Nov 2001 with a $40 million donation and has since contributed significantly to its development. The aim of Eclipse is to tap on the external innovation through collaboration with other software companies (D1) to create a software platform that allows different development tools, such as a code editor and an application modeling tool, to be integrated into a single development environment. This effort may be one of the key strategic initiatives by IBM to experiment with open-source innovation.

Open-source Eclipse as a strategy to develop a cohesive and integrated development environment

One of IBM’s key challenges, following its strings of acquisitions, is to develop a centrally rational product strategy (D2) to realign its overlapping and myriad technologies and integrate them into a cohesive integrated platform to reduce complexity and improve manageability.

Dynamics of the Eclipse initiative

Driven by the desire to maximize monetization of ideas (D1) and to encourage broader adoption and support of Eclipse platform by the industry, IBM decided to transit the original Eclipse group from an organization backed by IBM to an independent foundation modeled after other successful open-source organizations such as Apache Foundation (D1-1) which helps to reinforce the open-source ecosystem loop (R1) leading to greater adoption of Eclipse. However, the move to an independent foundation may have two negative effects. First, it may add complexity to the product as more companies get involved in its design which affects the ease of use of IBM’s products. Second, it may diminish IBM’s ability to steer Eclipse direction leading to less than optimal integration of IBM’s products. Both of these negative effects reduce the overall attractiveness of IBM’s products which diminishes the strength of R1.

The needs for a centrally rational product strategy (D2) with Eclipse as the foundational software platform for integrating IBM’s different development tools motivated IBM to strongly support

---

Eclipse’s development with both financial and manpower contribution (D2-1) that reinforces R1. But its inherent motivation to continue to push Eclipse agenda (D2-2) may dampen the vibrancy of the open-source network.

So far, the Eclipse open-source adoption has been widespread and successful with about 50 software companies that have joined the Eclipse consortium resulting in a strong ecosystem (R1). The vibrancy of this open-source network has a positive effect of encouraging all Java tool industry players – including those with Java application server products, such as Sun, BEA, Oracle and Borland, who compete with IBM – to support and adopt Eclipse (R2). But the relative larger value captured by IBM from Eclipse’s success compared to other members resulted in the industry’s perception that IBM dominated the setting of Eclipse’s agenda to push sales of its own Java software (B1) which dampens the potential adoption by all the Java community.

Conclusion
Overall, I think IBM has created much value from the Eclipse initiative tapping on the broad external open-source innovation. However, the long term effects of how much IBM can capture the value created, in pursue of its primary objective of creating a unified development platform to effectively integrate IBM’s broad portfolio of technologies.

8.4.3. Microsoft’s sustenance of its integral product architecture
Introduction
The current structure of IT architecture is shown in Figure 8-11. It is a horizontally integrated industry with multiple industry players in each layer. Microsoft is the only company with direct presence in all software layers attempting to build an integral end-to-end suite. One important dynamics in the APS software industry is to examine the various forces that may impact the sustenance of the power of leaders in various platforms, for example Client side PC hardware (Intel), Client side PC software (Microsoft Windows XP), Server side OS (Microsoft Server and Linux OS) and Server side application platform (Microsoft .Net, BEA WebLogic and IBM WebSphere). The many strong disintegration forces may put pressure on Microsoft to become more modular and horizontal.
<table>
<thead>
<tr>
<th>Client Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Client Devices</strong></td>
</tr>
<tr>
<td>PC</td>
</tr>
<tr>
<td>Mobile Devices</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Client OS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
</tr>
<tr>
<td>Linux Desktop</td>
</tr>
<tr>
<td>Windows Mobile</td>
</tr>
<tr>
<td>Linux Symbian</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Network Layer</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Server Layer</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application server</strong></td>
</tr>
<tr>
<td>Microsoft .NET</td>
</tr>
<tr>
<td>BEA WebLogic</td>
</tr>
<tr>
<td>IBM WebSphere</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Server OS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Server</td>
</tr>
<tr>
<td>Linux</td>
</tr>
<tr>
<td>Unix</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Server Hardware</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
</tr>
<tr>
<td>HP</td>
</tr>
<tr>
<td>Sun</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Services</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Consulting/Systems Integrators/Outsourcing</td>
</tr>
<tr>
<td>IBM</td>
</tr>
</tbody>
</table>

- Microsoft's Integral Strategy

**Figure 8-11: Current Structure of IT Architecture**
Figure 8-12: Microsoft's sustenance of its integral product architecture

Dynamics
As hypothesized by Fine, the industry dynamics results in an infinite double loop that cycles between vertically integrated industries inhabited by large companies and horizontally integrated industries populated by many innovative companies (see Figure 8-12).
- When the industry structure is vertical and the product architecture is integral, the forces of disintegration push toward a horizontal and modular configuration (B1). The many dimensions of technology increase the need for more structured organization which overtime results in bureaucratic and organizational rigidities that leads to entry by niche competitors.

- On the other hand, when an industry has a horizontal structure, another set of forces push towards more vertical integration and integral product architectures (B2). Technical advances in one subsystem result in increasing market power to its owner and encourage integration by the owner with other subsystems to develop proprietary integral solutions for higher profitability.

There are many forces, both internal and external, pressuring Microsoft’s integral product architecture to disintegrate into a more modular product architecture. The first internal force is the need to tie nearly all key product releases to one another in Microsoft’s next wave of product, called Longhorn, as a result of the integral product nature. This results in cascading delays in delivery of all of Microsoft’s products as a result of delay of any key components (B3). The second force is the result of the high dimensional complexity and organizational rigidity that reduces the ability to control all products development strategy, design and schedule, which in turn increases the possibility of major delays and reduces the ability to predict ship dates of key components. This has impact to both Microsoft’s own delivery of a series of products tied to those key components (B4) as well as to customers’ own product planning (B5). It also affects Microsoft’s effort to convince large companies to switch from one-time software purchases to Software Assurance license agreements to stabilize revenue stream, since customers may not get an expected free major upgrade within the three years licensing period which is the fundamental benefit of the licensing scheme (B6). All these internal forces put pressure on Microsoft to modularize and reduce the interdependencies between various products.

There are also several external disintegration forces. First, the inherent multi-layer and multi-device nature of the Internet (D1), made it possible for users to access the Internet through non-Windows PCs and workstations, as well as through non-PC devices, including Web-enabled cell phones and PDAs. Second, the support and co-ownership of Symbian mobile OS from some of the world’s largest mobile phone and wireless infrastructure companies, including Nokia, Sony Ericsson and Samsung, increases the ability and likelihood of Symbian retaining market leadership or substantial
market share in mobile phone software (D2) in future. These two factors make it necessary for Microsoft to support multi OS devices and platforms in its overall solution to customers. Third, the inherent OS independence nature of browser-based application made it possible for companies to run applications on any OS and thus eliminating the need for Windows and Microsoft’s desktop applications (D3). Lastly, browser-based applications bypass the large entry barrier of the need to install applications on the PC and allow many innovative niche competitors such as Yahoo and Google to enter the applications market (D4).

Conclusion

All these forces, together with antitrust and anti-monopolistic pressure (D5) on Microsoft make it highly unlikely for Microsoft to establish or sustain its dominance in the various platforms, similar to the level of desktop OS, and remain truly a pure integral product. Microsoft would need to support other platforms, such as Symbian OS in mobile phone, other than its own.

**8.4.4. Microsoft’s ability to extend from desktop to server dominance**

![Diagram of Microsoft's ability to extend from desktop to server dominance](image_url)

*Figure 8-13: Microsoft's ability to extend from desktop to server dominance*
As described in Section 8.4.3, to sustain Microsoft's integral product architecture, Microsoft would need to extend its Windows desktop dominance to the server and device end. Here, we will examine the various forces affecting Microsoft's effort in extending its dominance from desktop to server.

First, the fact that many enterprises are in the Application Silo Architecture Stage (D2) implies that the environment of most enterprises, including those in Microsoft's SMB market, are heterogeneous with many types of legacy hardware and software. This necessitates the support of heterogeneous environment in the server end which increases the complexity of Microsoft's products that over time reduces the comparative advantage of .NET over J2EE in ease of use leading to higher adoption of J2EE over .NET (R1). Second, Microsoft .NET only runs on Windows platform (D1) and is not able to run on heterogeneous environment reduces some flexibility of choice for enterprises making the choice of J2EE more attractive. Third, BEA's focus on ease of development through development of a unified platform and IBM's effort in Eclipse to unify its development environment (D3) reduces the advantage of .NET over J2EE in ease of use. Fourth, the fear of ISVs with Microsoft's potential future expansion into their market (D4) and competing directly with them, and Microsoft's product group having the advantage in access to detail knowledge and internal support from .NET platform group, reduces the attractiveness of .NET to ISVs. All the above factors increase the relative attractiveness of J2EE over .NET (D5) that reduces the .NET adoption rate.

There are two reinforcing loops that typically drive the adoption of new technology. In the case of .NET, they are the network effect (R2) that improves the support structure, availability of trained resources and interoperability between systems deployed on .NET, and the complementary goods effect (R3) that increases the number of complementors as a result of the attractiveness of the .NET market size which in turns reinforces the attractiveness of .NET itself (R2). However, the entrance and growth of Linux server (D6) has lowered the expected Windows Server and .NET market size - Microsoft has expected increased in Windows Server deployment from firms transiting out of the higher cost Unix servers - reducing the overall effect of these two reinforcing loops.
8.4.5. Use of open-source as a competitive tool to disrupt the market

Figure 8-14: Use of open-source as a competitive tool to disrupt the market

Open-source is increasingly being used as a competitive tool to disrupt the market with great ramifications to the entire commercial software industry. It is important to understand the dynamic forces at play that drive this trend. First, open-source is an attractive option, over .NET and commercial J2EE offerings, to ISVs and startups which need low-cost, flexible and multi-platform environment (D5) to enable broad market acceptance of their own products. Moreover, the technical capability within the ISVs (D6) are generally higher than typical user enterprises which increases their ability to resolve technical issues on their own and reduces the resistance to adopt
open-source products due to concern of lack of comprehensive open-source vendor support. The adoption by ISVs – the lead user of APS – is the first step towards the broader adoption of open-source APS.

Second, the maturity of J2EE application server (D7), as a result of the establishment of core J2EE specifications, together with the emergence of open-source alternative brings about the commoditization of the application server that leads to multiple other effects that in turn support the adoption of open-source products. The commoditization reduces the value of the application server market to commercial software companies (R1) which increases their willingness to support open-source low margin products if they can shift their focus to services and ancillary products where they can still make healthy profits (R2) – IBM and SAP can still profit from alternative businesses in services and business application market respectively. The resulting shift from product-centric value attributes to market-centric ones also makes it an attractive option for companies to be seen as supporting open-source (R3) that provides lower cost and less lock-in to end-user customers. All these factors increase the possibility of open-source endorsements by influential technology companies as a tool to disrupt the market.

Lastly, and the most important, is the occurrence of triggering events leading to the use of open-source as a disruptive tool that not only put pressure on competitors profit margin but also on the initiator’s own margin. I’ve identified four key triggering events: (D1) The success of Microsoft’s entry into SAP’s business application market; (D2) The success of SAP’s entry into APS market; (D3) The success of Microsoft’s entry into enterprise APS market; and (D4) The success of IBM in sustaining its Java APS market share against BEA. These events also formed the basis of the analysis of future scenarios in the APS software industry in the next chapter.
8.4.6. Value chain dynamics

-- Diagram --

**Dilemma between customer focus and economy of scale.**

**Efficiency of Back End System Provider.**

Competitive pressure on niche System Providers.

Disintegration force on System Providers.

**Proliferation of System Providers.**

**Phase M/Type M1:** Reintegrated under Platform Dominance (Phase for Market Expansion and Business Model Innovation).

**Phase M/Type M3:** Disintegration of System Providers (Phase for Choice between Customer Intimacy and Process Efficiency).

**Phase M/Type M1 players**

- Siebel (packaged CRM)

**Packaged application software companies**
- SAP (Package business application and APS software)
- Microsoft
- Oracle

**Phase M/Type M3 players**

- Siebel (SOA) - all major integration vendors such as IBM, Microsoft, BEA, etc
- PeopleSoft (packaged business application) - IBM and BEA (APS)
- SAP (database) - MySQL

**Services and software companies**
- IBM (Services and APS software)
  - HP (Services and hardware) - Partner with Best of breed software players
  - Accenture (Services) - Partner with best of breed players
- BEA (APS software) - Partner with services companies

---

Figure 8-15: Value chain dynamics

174
As described in one of the analytical framework in Section 5.10 on Value Chain Dynamics, in a M3 value chain, although a firm can play both the front-end and back-end provider role, the dilemma between customer focus and economy of scale coupled with the improved efficiency of back-end provider forces the firm to choose a primary focus on either customer front-end provider or back-end provider.

The current state of companies in the packaged application software, and services and software categories are listed in Figure 8-15. The state of the IT industry is a hybrid mode of Phase M/Type M1 and Phase M/Type M3 with some companies transiting from M1 to M3.

Packaged application software
Siebel used to be in Type M1 but is currently in transition to Type M3 in its development of the Universal Application Network (UAN), Siebel’s new enterprise application integration initiative based on SOA. In contrast to all major ERP packaged application vendors’ enterprise application integration strategies, Siebel’s approach with UAN relies heavily on partnerships with established EAI vendors (e.g. IBM, Microsoft, BEA, SeeBeyond, Tibco, and webMethods). With the partnerships, Siebel will be able to focus specifically on extending its domain expertise and delivering vertical- and industry-specific applications.\textsuperscript{138}

PeopleSoft is a Type M3 company with strategic partnerships with IBM and BEA to utilize their respect infrastructure software stacks to deliver complete solution to PeopleSoft’s customers. PeopleSoft applications support a broad array of technologies including hardware platforms, database servers, web/application servers and browsers.

SAP has partially transited from Type M1 to Type M3 with the outsourcing of its SAP DB relational database to MySQL to focus on the packaged business application business while at the same time developing its own back-end NetWeaver APS software.

Microsoft and Oracle continues to be Type M1 companies attempting to develop end-to-end suite

consisting both business application and infrastructure software.

Services and software
IBM, which has both the financial and technical resource, provides both the services and software infrastructure to customers to deliver whole solution for custom software development platform in a Type M1 model.

HP, Accenture and BEA, on the other, each focuses on their own niche areas and relies on partnerships with best-of-breed providers to deliver complete solution to customers in a typical type M3 model.
9. Potential Future Scenarios and the Proposed Strategies for APS Software Providers

9.1. Introduction

In this chapter, we will explore the potential future scenarios in the APS software industry using scenario analysis and identify strategic options required for companies to be prepared in the event of strategic changes in the industry.

We will use the steps proposed in Section 5.12 on Scenario Analysis for conducting the analysis:

1. **Surface the key uncertainties.** Identify questions that cannot be answered, but which appear to matter greatly.
2. **Rank the key uncertainties to determine the key drivers.** Identify the two or three most important unanswerable questions – the things that cannot be known, that if known, would tell strategic planners precisely what they need to know.
3. **Combine the key uncertainties to yield concrete scenarios.** That is, define alternative futures in which each uncertainty is assumed to take an extreme value.
4. **Provide details.** Turn each scenario into a plausible story, explaining how it might come about and how customers, executives in the firm, and executives in competing firms would feel about it.

9.2. Strategic Uncertainties

We will first identify the strategic uncertainties by analyzing the key market drivers and dynamics explored in Chapter 8 and categorized them according to the key players in the APS software industry.

**Microsoft**

- How much and how fast will Microsoft’s profitability in the two most profitable segments, Client and Information Worker segments (the two segments collectively generate 60% of Microsoft’s FY03 revenue of $32 billion and all of FY03 operating income of $13 billion including covering the losses of other business segments), be impacted by open-source Linux desktop and OpenOffice alternatives?

- How much and how fast can Microsoft build up its market share and profitability in its other money-losing business segments such as Microsoft Business Solutions ($567 million
revenue and $254 million in operating loss in 2003) and Home and Entertainment ($2,748 billion revenue and $924 million in operating loss in 2003)?
- How successful will Microsoft be in entering the packaged business applications market and pose a threat to SAP?
- How successful will Microsoft be in proliferating and sustaining the Trustworthy Computing initiative to improve reliability, security and privacy across its product lines against its fundamental strategy of feature-driven development approach?
- How much impact does security vulnerabilities and periodic “forced” migration effort has on TCO and the effect on CIOs’ and business executives’ impression of Microsoft? What is the degree of impact to potential sales as a result of the above IT product standards decision makers’ negative impression of Microsoft?
- How will Microsoft products’ ease of use be impacted by the need to support heterogeneous environment in its effort to expand into the server platform?
- How much external application complements do Microsoft need in its expansion into other platform? How much trust can Microsoft gained from application complementors?
- What will ISVs choose as the software platform to develop their products on?
- To what degree will Microsoft comply with fair disclosure to competitors on its proprietary interfaces and the likelihood of future antitrust action against Microsoft?

IBM
- Can IBM manage the product portfolio complexity and continue to grow through acquisitions?
- Will IBM succeed in creating a unified development platform through open-source mechanism to effectively integrate IBM’s broad portfolio of technologies and compete with Microsoft and BEA in ease of use?

BEA
- Will BEA continue to optimally drive standards and innovate at a fast pace given its relative smaller size compared with IBM and Microsoft?

SAP
- Can SAP’s NetWeaver penetrate the infrastructure software market and become the primary infrastructure provider within the SAP installed based?
9.3. Key Drivers Identified

Ranking the key uncertainties identified above, the three most critical unanswerable questions are: (1) Success of Microsoft’s entry into enterprise APS market; (2) Success of IBM in creating a unified development platform; and (3) Success of Microsoft’s entry into SAP’s core packaged business application market.

(1) Success of Microsoft’s entry into enterprise APS market
Microsoft’s other profitable segment, besides Client and Information Worker, is the Server and Tools division with annual revenue of about $7 billion and operating income of $2.5 billion in 2003. It is essential that Microsoft continue to grow this business to become the core of its business as server side applications is the engine that drives the business in the Internet era.

To grow, Microsoft needs to expand from its current focus on lower-end SMB market to the larger SMB and enterprise APS market. Although the reliability, performance and security of Microsoft’s latest Windows Server 2003 has greatly improved over its predecessors, it remains unclear how successful will this expansion be in future, with increasing competition from Linux server, especially after the current migration trend out of higher-cost Unix servers is completed. The security vulnerabilities and periodic “forced” migration effort has major effects on TCO that greatly influences decisions of cost conscious CIOs and executives. Moreover, the need to support heterogeneous environment in larger organizations may increase the complexity of Microsoft’s products suite that reduces its key advantage over J2EE products in ease of use. And the adoption of its new server platform by ISVs may also be greatly hampered by the competition, now or in future, with Microsoft in Microsoft’s expansion into the business application market.

In summary, the lack of trust by developers, complementors and IT product decision makers in Microsoft is the key factor that affects Microsoft’s penetration success into the enterprise APS market. The degree of success is likely to be low to medium.

(2) Success of IBM in creating a unified development platform
IBM has been capitalizing on its strength in brand marketing and global sales channels to grow the IBM Product division through acquisitions. However, in recent years, the change in acquisition strategy from slow integration to rapid and immediate integration limits the choice of suitable acquisition target and thus slows IBM’s growth through acquisitions. As an alternative growth strategy, IBM has initiated a major program in 2003 called ISV Advantage Initiative to expand
business into the $300 billion SMB technology market through ISV partnerships and plans to invest $1 billion in 2004 in working with the partners. ISVs represent the fastest growing segment of the IBM business partner world, and the SMB space is fastest growing among that. So far, 200 ISVs have joined in the program's first year.\(^{139}\)

The aim of this program is to realign other IBM ISV programs to better enable the company's ISV partners to go after business industry-by-industry. Under the program, IBM is delivering Web-based communities that will facilitate the development of industry-specific solutions. There will be a total of 16 industries, with the initial first 6 industries covering banking, financial markets, healthcare, life sciences, retail and communications. Each industry will have a specific architecture optimized for the industry's needs.

ISV Advantage provides ISVs with comprehensive technical, marketing and sales support to help meet the specific needs of medium sized businesses. The plan is for ISVs that have enabled their applications on IBM's pre-configured and industry-optimized infrastructure software to provide customers with flexible, reliable, secure and cost effective solutions that span multiple computing platforms, including Linux. In addition, customers are provided with industry-specific solutions that will help them automate and manage the business processes that are unique to their needs, enabling them to respond to customers, partners and suppliers faster.

However, this initiative still depends very much on the success of IBM's effort in creating a unified development platform to effectively integrate IBM's own broad portfolio of technologies onto which the ISV partners build their own solutions on. IBM has just announced in Mar 2004 that over the next year and a half, the company will rework the Rational design tools that it acquired in Dec 2002 to become fully integrated with Eclipse as the first phase of its IBM Software Development Platform plan to meld IBM products\(^{140}\).

In summary, it remains unclear at this point the degree of success this initiative, of using


consortium-based open-source Eclipse for internal integration effort, will be in bridging the gap in ease of use of its products to compete with Microsoft and BEA. The fact that IBM still contributes the majority of the development resource to the newly formed independent foundation, with more than 100 researchers focused on enhancing the Eclipse platform, increases the success rate. The degree of success of this initiative is likely to be medium to low.

(3) Success of Microsoft’s entry into SAP’s core business application market\(^{141}\)

The success of Microsoft’s entry into the SMB packaged business application market is critical for Microsoft in order to replace the potential loss of revenue from its two most profitable segments of Client and Information Worker as a result of the emergence of viable open-source alternatives.

Microsoft intends to grow its Business Solutions group from the current $0.5 billion in annual revenue to $10 billion by 2011. This is about the same size as the Client division which sits on 94% of corporate desktops worldwide. It would also make Business Solutions bigger than current-day SAP or Oracle, longtime business application players that will compete aggressively with Microsoft for a slice of the SMB growth market.

Unlike its typical entry into new markets through acquisitions of small companies and then evolving the products to attain market leadership, Microsoft built the Business Solutions group through its largest acquisitions ever, of Great Plains Software for $1.1 billion in April 2001, and Navision for $1.3 billion in May 2002.

Microsoft initiated a multi-year project called Project Green in an attempt to unify and replace the separate Great Plains, Navision and Microsoft CRM with applications built on a single codebase on top of the new suite of Microsoft Longhorn tools targeted to be shipped in 2006. Project Green is still several years away and Microsoft announced that the existing business applications would be supported until 2013, which under Microsoft product lifecycle formula would mean that active development of the applications would cease in 2008 and presumably replaced by the Project Green software. This poses a major hurdle for Microsoft to convince ISVs, resellers and customers to buy into the current set of products and having to migrate into the new platform a few years down the road.

However, with Microsoft’s track record of developing many award winning products after three or four versions of enhancements before “getting it right” and having a fair share of Microsoft’s $6.9 billion annual R&D budget, Microsoft may be able capture market share over a period of time. Therefore, the degree of success in Microsoft’s effort to enter SAP’s core packaged business application market and reach its $10 billion target is likely to be medium to low.

**Summary**

Table 9-1 shows the summary of the likely outcome of the key uncertainties.

<table>
<thead>
<tr>
<th>Strategic Uncertainties</th>
<th>Description</th>
<th>Likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Success of Microsoft’s entry into enterprise APS market.</td>
<td>Low to medium.</td>
</tr>
<tr>
<td>2.</td>
<td>Success of IBM in creating a unified development platform.</td>
<td>Medium to low.</td>
</tr>
<tr>
<td>3.</td>
<td>Success of Microsoft’s entry into SAP’s core packaged business application market.</td>
<td>Medium to low.</td>
</tr>
</tbody>
</table>

**9.4. Scenarios Identified**

Combining the above key uncertainties, five strategic scenarios are presented as shown in Figure 9-1. Each of the key drivers is represented on the horizontal or vertical axis of a four-quadrant matrix. The horizontal axis represents the most critical uncertainty, the degree of success in Microsoft’s entry into enterprise APS market. The vertical axis represents the second critical uncertainty, the degree of success of IBM in creating a unified development platform. The region indicated by the notation in the top right hand corner of the figure represents a high degree of success in Microsoft’s entry into SAP’s core business application market.

The market share notation in the bottom right hand corner of the figure represents a pictorial estimation of the market share captured by the various players analyzed here – IBM, Microsoft, BEA and SAP. The horizontal axis represents the customer segment: SMB on the left towards large enterprises on the right. The vertical axis represents the types of application: package application on the bottom and custom application on the top.
Based on the summary of the likely outcome of the key uncertainties in Table 9-1, the base case scenario is depicted as Scenario 1 in Figure 9-1. The major winners of the five scenarios are: (1) BEA, IBM and SAP; (2) BEA and SAP; (3) IBM and SAP; (4) Microsoft, IBM and SAP; and (5) Microsoft.

**Scenario 1: BEA, IBM and SAP**

The base case scenario, Scenario 1, is characterized by Microsoft failing to enter the APS market due to lack of trust in Microsoft by developers, complementors and IT product decision makers, but is partially successful in its assault into SAP’s core packaged application market. IBM partially succeeded in developing integrating its broad and overlapping product portfolio to develop a simplified platform that competes with Microsoft and BEA in ease of use, and managed to maintain its J2EE market share sharing the APS market with BEA.
Scenario 2: BEA and SAP

Scenario 2 is characterized again by Microsoft failing to gain the trust of developers, complementors and IT product decision makers coupled with increased complexity of its product suite as it is being pressured to support the inherent heterogeneous server environment of most companies. Microsoft lost great market share to alternatives provided by open-source and J2EE. IBM is also unable to deliver its integrated platform suite and BEA and SAP emerged as great winners in custom application and package business application market segments respectively.

Scenario 3: IBM and SAP

In this scenario, Microsoft failed in entering both the enterprise APS market as well as into SAP’s packaged business application market. IBM managed to push Eclipse development towards its agenda of unifying its platform and yet support a vibrant open-source network, capturing a substantial portion of the value created from external open-source innovation. IBM emerged as the greatest winner and split the market with SAP.

Scenario 4: Microsoft, IBM and SAP

Scenario 4 is characterized by Microsoft’s success in enterprise APS market but failure in enterprise packaged business application market, with IBM succeeding in developing its unified platform. The power and market share is co-shared by Microsoft, IBM and SAP.

Scenario 5: Microsoft

In this scenario, Microsoft emerged as the sole winner extending its dominance from desktop operating system to the server and devices end in developing a comprehensive, integrated software platform. Not only is Microsoft successful in the platform segment, it also succeeds in rebuilding the acquired Great Plains and Navison packaged application into an integrated and yet flexible business application suite and capturing much market share from SAP.

9.5. Scenarios Explored

In this section, we will explore how customers, executives in the firm, and executives in competing firms would feel and react to the scenarios.

Scenario 1: BEA, IBM and SAP

To regain the trust of developers, complementors and customers, Microsoft will use its remaining huge cash reserves to further improve its products’ security, change its growth strategy from one of
making periodic architectural innovation at the operating system level to drive upgrade sales to one of providing an underlying stable software platform and innovating at the higher application level. It will continue to focus on improving its packaged business application to compete with SAP and spend much resource on building better relationships with complementors as well as marketing to improve its corporate image.

IBM, with additional growth and market share captured from Microsoft, will not overreact to BEA overtaking IBM slightly in the J2EE APS market share, since it still gain substantial value from the sale of its APS software.

SAP, to counter the partial successful assault by Microsoft into its core packaged business application market, SAP may decide to outsource its underlying infrastructure software stack to infrastructure specialists such as IBM or BEA to transit to a Type M3 value chain model so as to focus on the applications business.

**Scenario 2: BEA and SAP**

In this scenario, the action by Microsoft is likely to be similar to Scenario 1, spending effort on building trust, relationships and marketing to improve its image.

IBM, in its attempt to gain J2EE market share from BEA, is likely to disrupt the APS market and try to capture as much value from its services and hardware businesses instead, since it has nothing much to lose in the APS software market without substantial market share. Similar to Eclipse, IBM may disrupt the market by supporting or contributing its APS software stack to the open-source community to gain broader industry acceptance, commoditize the APS software market to reduce the profitability and relative power of BEA and Microsoft, and profit as the integrator and distributor of open-source software supporting enterprises in its overall IT solution that incorporates much open-source software. In essence, IBM transits to a Type M3 value chain model focusing on customer needs and solution delivery.

For this scenario, the industry wide software product profit margins shrink, open-source flourishes with many companies moving into second generation open-source model by placing a free version of their commercial software for non-commercial use (similar to MySQL), and revenues shift from products to mainly services.
Scenario 3: IBM and SAP
Similar to Scenario 2, Microsoft will try to improve its position through the use of its cash reserves while BEA struggles to compete with IBM. To remain viable and be able to compete with IBM, BEA is likely to seek a closer relationship with strategic partners, such as HP, Accenture or Intel, or even be acquired to increase its financial strength and size to take on IBM’s support and sales division with strong sales connection to high level executives in enterprises. Microsoft and BEA may seek closer co-operation in their collective confrontation with IBM.

Scenario 4: Microsoft, IBM and SAP
In this scenario, BEA is surrounded by stronger players each occupying their own niche market segments. BEA is most likely to be acquired. Some of the potential candidates are HP, SAP and Oracle. HP would be a perfect fit for BEA as the acquisition would strengthen the integration between their complementary offerings with BEA supplying the software for HP environments. SAP or Oracle may want to acquire BEA for its unified platform suite to complement or replace their own weaker components in order to strengthen their positions in the packaged application market and prevent Microsoft from encroaching further into their core enterprise territory. Microsoft and IBM are unlikely to be able to acquire BEA for antitrust reasons.

Scenario 5: Microsoft
With both IBM and SAP losing power and market share to Microsoft, it is the most likely scenario that both would compete aggressively with Microsoft and disrupt the APS software market while seeking profits in their other revenue generating business in services and packaged application respectively. Similar to Scenario 2, open-source is likely to be the tool used to disrupt the market given the successful precedence of IBM using Linux to attack Windows and SAP using MySQL to put pressure on Microsoft’s and Oracle’s database business.

Similar to Scenario 1, SAP may also outsource its software infrastructure stack to transit to a Type M3 value chain model so as to focus on the applications business.

Summary
Table 9-2 summarizes the likely responses of the various key industry players to each of the five scenarios.
<table>
<thead>
<tr>
<th>Scenario</th>
<th>Major winners</th>
<th>Likely responses</th>
</tr>
</thead>
</table>
| 1.       | BEA, IBM and SAP | • Microsoft changes its fundamental growth strategy and make strategic shift to innovate at the higher level software stack above the operating system.  
• SAP shifts to a Type M3 value chain model to focus on the applications business to compete more effectively with Microsoft. |
| 2.       | BEA and SAP   | • IBM disrupts the APS market by supporting or contributing its APS software to the open-source community and profit as open-source integrator and distributor.  
• Many commercial software companies move to second-generation open-source model and shift revenue source from products to services. |
| 3.       | IBM and SAP   | • BEA seeks closer relationship with strategic partners, or even contemplates being acquired to increase its financial strength and size to take on IBM.  
• Microsoft and BEA seeks closer co-operation in their collective confrontation with IBM. |
| 4.       | Microsoft, IBM and SAP | • BEA is acquired by one of the potential candidates such as HP, SAP and Oracle.  
• Microsoft and IBM are unlikely to join the bid to acquire BEA for antitrust reasons. |
| 5.       | Microsoft     | • IBM and SAP uses open-source to disrupt the APS market.  
• SAP shifts to a Type M3 value chain model to focus on the applications business to compete more effectively with Microsoft. |
9.6. Proposed Strategies for APS Software Providers

Reviewing the likely responses of the key industry players to each of the five scenarios in Table 9-2, I have identified four strategic options that the APS software industry players can consider to invest and position themselves strategically and be able to respond quickly in the advent of any of the likely scenarios, while waiting for the environment to become clearer.

The four strategic options are: (1) Microsoft should fundamentally change its business and technology strategy to shift the likely outcome towards Scenario 4: Microsoft, IBM and SAP; (2) SAP should consider progressively moving to a Phase M/Type M3 value chain model, and BEA and IBM should consider how to position themselves as the preferred choice of the IT industry’s back-end system provider; (3) Microsoft and BEA should invest in small open-source experimentation to understand the open-source dynamics; and (4) BEA should plan for the contingency of being an acquisition target with the primary aim of sustaining its fundamental position of remaining as an independent and pure-play infrastructure software vendor.

(1) Microsoft should fundamentally change its business and technology strategy to shift the likely outcome towards Scenario 4: Microsoft, IBM and SAP

Microsoft’s greatest enemy is with itself, not IBM, BEA, SAP or even open-source. It has to start with the review of its business and technology strategy that are the actual drivers behind many of its current challenges and dilemmas. Moving forward, the fundamental challenge of Microsoft is in gaining trust from developers, complementors and IT product decision makers.

The first issue is the integral nature of Microsoft’s technology strategy and the focus of making architectural innovation on the existing software platform foundation to encourage user upgrade sales in order to meet its growth objective. The integral product architecture blurs the distinction between operating system and the software platform\footnote{The middle layer “engine” that sits between user defined custom applications and the underlying operating system and shield end users from changes in the underlying technology as it evolves over time and thus reduces TCO over the useful life of custom application.} and result in major migration effort when architectural innovation is applied at the existing software platform foundation, which increases TCO for both enterprises that uses Microsoft’s platform to develop custom applications and ISVs that build applications for Windows platform. Microsoft’s customers has gotten weary of the
periodic upgrade cycle, especially if it is not really architectural innovation but upgrade due to poor product architecture of previous versions of Windows resulting in major changes of Windows API interfaces over the different generations of Windows.

The second issue is Microsoft’s over diversification and declining market focus. Although Microsoft has identified security as fundamental to the acceptance of Microsoft’s products in enterprises and initiated a company wide Trustworthy Computing initiative to improve reliability, security and privacy of its products, the diversification into hundreds of products in many domains may prevent Microsoft from proliferating and sustaining the focus on security, especially with its rapid prototyping and feature-driven development approach across many of its product lines. To project the security conscious image effectively, Microsoft not only have to be security conscious in developing its platform products but also in other products.

A third issue is Microsoft’s attempt to influence the needs of users through technology push to encourage adoption of the new versions of existing products. The incremental and continuous improvement strategy of adding new features may have already overshoot what is needed by majority of Microsoft’s products users resulting in diminishing returns of additional new features.

To regain the trust of developers, complementors and IT product decision makers, Microsoft has to decouple the integral architecture, at least internally within Microsoft, to clearly define the operating system and software platform layer and innovate by adding on top of the existing platform foundation instead of drastically changing the existing foundation. In this way, customers will be more than happy to pay Microsoft to enjoy the benefits of new innovation from Microsoft without the huge migration cost.

Microsoft should also change its approach of using three or four versions to “get it right” for enterprise applications. Unlike consumer products like PDA or mobile phones where users simply discard and buy new ones periodically, enterprises build applications with the aim of sustaining its useful life for as long as possible. For customers to go through several versions of upgrade before becoming stable may not be a viable strategy when there are alternatives available. This will be one of the key factors affecting the success of Microsoft’s entry into enterprise packaged business application market.

(2) SAP should consider progressively moving to a Phase M/Type M3 value chain model, and BEA
and IBM should consider how to position themselves as the preferred choice of the IT industry’s back-end system provider

SAP’s investment and entry into the APS software market may be too late when the market has well-entrenched players like IBM, BEA and Microsoft. SAP may only realize this, to its detriment after investing much effort into APS and losing its existing competitive edge in package applications to Microsoft and many other niche package application players – niche application may become a very viable alternatives to enterprises that will enable them to both standardize their business processes and yet achieve strategic agility with the use of maturing SOA technology that enables easier integration between heterogeneous applications.

SAP’s internal strategy may be similar to SAP DB database which it developed internally to a certain maturity level before licensing it to MySQL as open-source software to reduce the dependence on and the power of database component supplier such as Oracle and Microsoft. Regardless, it may overtime finds that it is more economical to outsource the increasingly standardized infrastructure software stack to infrastructure software specialists and focus more on its core packaged application business and shift to a Phase M/Type M3 value chain model.

Besides investing in the core foundational infrastructure software technology, BEA and IBM should invest much engineering effort to position itself as the integrator of back-end technology components, both from itself as well as from external commercial product partners and even those open-source components with licenses that allow them to be integrated with commercial products.

IBM has already initiated this process through its ISV Advantage Initiative which plans to develop more than 16 industry specific architectures optimized for each industry’s needs. However, IBM needs to balance the scale and scope of this initiative, which it aims to capitalize on the vast external innovations of ISVs, with the need to invest on internal R&D to integrate and capture the value created.

BEA also has similar initiatives, one of which is the development of a carrier-grade service delivery platform, announced in June 2003, with pre-integrated hardware and third-party applications, targeted at helping service providers increase profitability by enhancing their ability to deliver an optimal customer experience quickly and cost effectively. Another initiative is with HP in developing the HP Mobile Service Delivery Platform based on BEA’s APS. Due to BEA’s relatively smaller size as compared with IBM, BEA needs to resist the temptation of responding with similar
broad programs and instead focus on certain niche industries to make them highly successful before broadening the scope into more industries. This is especially important, since the APS is a relatively new product on the market and the fundamental objective is to cross the chasm first.

(3) Microsoft and BEA should invest in open-source experimentation to understand the open-source dynamics
The emergence of second-generation open-source model, which adopted a dual-license mode of offering products under both an open-source license and a commercial license to enjoy both the benefits of open-source and commercial products, and the potential use of open-source as a disruptive tool by influential technology companies makes it necessary for all commercial software companies to devise a plan to deal with such scenarios if it occurs. Open-source component, either external or internally sponsored ones, may become an essential element of the integrated offerings provided by back-end system provider in the Model M/Type M3 value chain model.

Both Microsoft and BEA should prepare itself adequately in understanding the dynamics of open-source in the event that they may also need to move into the second-generation dual-license model for some of their technology components. This is especially so for BEA, which may be required to take this step, either as a requirement to become the back-end system provider of SAP, or to counteract the potential move by IBM to put some of its software stacks as open-source for broader adoption, similar to the strategy of Eclipse. BEA needs to review its architecture to determine the architecture boundary from which certain components of its APS can be separated, submitted and managed as separate open-source projects and then re-integrated back into the suite while considering the potential impact to its competitive advantage.

(4) BEA should plan for the contingency of being an acquisition target with the primary aim of sustaining its fundamental position of remaining as an independent and pure-play infrastructure software vendor
BEA, with a market cap of only about US$5.6 billion and an annual revenue of US$0.9 billion in 2003, is the smallest company among the key APS software industry players such as Microsoft, IBM, Oracle, SAP and HP (see Table 9-3).
Table 9-3: Annual software/services revenue and market value of key players in APS software industry


<table>
<thead>
<tr>
<th>S/No</th>
<th>Company</th>
<th>Annual software/services revenue (US$ billion)</th>
<th>Market value (US$ billion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Microsoft</td>
<td>28</td>
<td>327</td>
</tr>
<tr>
<td>2.</td>
<td>IBM</td>
<td>49</td>
<td>179</td>
</tr>
<tr>
<td>3.</td>
<td>Oracle</td>
<td>9.7</td>
<td>70</td>
</tr>
<tr>
<td>4.</td>
<td>SAP</td>
<td>7.7</td>
<td>49</td>
</tr>
<tr>
<td>5.</td>
<td>HP</td>
<td>10</td>
<td>35</td>
</tr>
<tr>
<td>6.</td>
<td>BEA</td>
<td>0.9</td>
<td>5.6</td>
</tr>
</tbody>
</table>

In the scenario where IBM successfully integrates its overlapping technologies to develop a unified platform coupled with its broad range of technologies, BEA may need to seek closer cooperation with strategic partners or be acquired to take on IBM. Oracle may also seek to acquire BEA, especially if Oracle fails in its current acquisition bid for PeopleSoft to consolidate its packaged business application position against SAP.

BEA would need to make contingency plans for such events, especially if it is a hostile takeover, in order to keep its status of being a pure-play infrastructure software vendor which is one of the fundamental strengths that makes BEA an attractive partner to both application vendors as well as systems integrators who competes with Microsoft and IBM.
10. Conclusion

The ultimate objective of this thesis is to identify APS software vendors' strategic options in preparation for future challenges and in the event of strategic changes in the industry. I first identified the key trends in the software industry and analyzed the key players' current strategies and challenges to identify the key market drivers and dynamics. Based on the key drivers, I explored the future scenarios in the APS software industry. Then, building on the analysis of the future scenarios, I proposed strategic options in preparation for future challenges and strategic changes in the industry.

I describe below the key points of each chapter and show how these key points are interrelated in Figure 10-1.

Chapter 4, I described three business and two key technological trends in the APS software industry. First, as firms progress across the four architecture stages (Application Silo, Standardized Technology, Rationalized Data and Modular Architecture Stage), the business case for IT shift from ROI of individual applications, to reduced total cost of ownership and interoperability, to improved business performance and integration and then to speed to market and strategic agility. One important factor impacting infrastructure software vendors' marketing and sales strategy is the change in target audience from business unit IT manager to CIO and business executives. Second, there is a trend of broader adoption of open-source products in enterprises beyond the traditional community-based open-source projects such as Apache and Linux with the emergence of second-generation open-source model that combined the benefits of open-source software and commercial software. Also, there is a push of open-source Linux into the desktop and devices, especially in countries outside of US. Third, Microsoft is working to settle its legal and regulatory disputes and move on to focus on impending challenges, and setting up collaborative cross-licensing agreements with companies in a shift to Open Innovation paradigm. Lastly, the adoption of Web Services and SOA will enable easier integration between heterogeneous applications and provides higher level of flexibility and reuse. These will reduce the need for single end-to-end monolithic software packages, facilitate co-existence of .NET and Java environment and allows IT to more closely reflect the way business operates and improve the agility to create new business opportunity and respond to competition.

In chapter 5, I presented frameworks to analyze the competition and summarize prior research to identify key implications that will have bearing on the evolution of the industry.
I used the Roberts and Berry’s Familiarity Matrix as a uniform framework for assessing the technology acquisition strategy of IBM and BEA. The Four Levers Framework proposed by Gawer and Cusumano for designing and implementing a successful platform strategy is used for assessing the platform leadership strategies of Microsoft and BEA.

Also, several other prior research provided insights into the drivers of high-tech industry. The infinite Double Helix loop that cycles between vertically integrated industries inhabited by large companies and horizontally integrated industries populated by many innovative companies, illustrates the key disintegration and integration driving forces.

Chesbrough described how the desire for optimal monetization of innovation triggered the shift from Closed to Open Innovation paradigm, and emphasizes that even in the Open Innovation paradigm, there remains a critical role for internal R&D to integrate and align the diverse innovations in order to capture a portion of the value created by the research contributions of the firm and the activities of external firms.

Meyer and Lehnerd described how the use of product platform can be an effective mechanism to enable the difficult transition from one generation of technology to the next and sustain their excellence through era of changing technology and leadership. They also highlighted the danger of unplanned or over diversification which increases complexity that may result in the dilution of the centrally rational product strategy, and the difficulty of introducing common subsystems and processes across the corporation once management lost control of the product development strategy.

The research by von Hippel et al on software innovation through open-source projects as horizontal innovation networks provided insights into the conditions necessary for the user innovation networks to function entirely independently of manufacturers, in this case commercial software companies. Two important incentives that commercial software companies may invest in open-innovations are: the companies may invest and contribute to open-source projects, even if it leads to commoditization of their own products, if the improvements result in higher reward from other areas of their business; and the freely revealed innovation leads to broad adoption resulting in the open-source software becoming an informal standard, especially if the innovation is designed in a way that is especially appropriate to conditions unique to the innovator.
Moore described one key challenge of many companies in their attempt to expand high-tech products markets from the early adopter to the mainstream early majority customer segment. The dilemma faced in crossing the chasm is a catch-22 situation in that good customer reference are critical to the risk averse early majority customers, but the only suitable reference for an early majority customer, is another member of the early majority, but no upstanding member of the early majority will buy without first having consulted with several suitable references. To cross the chasm, Moore suggested concentrating an overwhelmingly superior force on a highly focused niche market to win over pragmatist customers in advance of broader market acceptance.

The thesis by Kawashima analysis the value chain dynamics of the PC industry and applied the insights to the telecommunication industry. He presented a framework for multi-dimensional value chain analysis and used it to help analyze and identify the shift of companies’ competitive advantages and power dynamics as the industry evolves through the technology/market lifecycle. In the Phase M/Type M1 phase, systems are designed with modular architecture but the design chain is reintegrated by a platform dominator. However, the dilemma faced by the system providers between customer focus and economy of scale creates a force to disintegrate system providers into front-end providers with customer intimacy and back-end providers with process superiority into the Phase M/Type M3 phase.

The insights provided by Christensen and Raynor, in their book “The Innovator’s Dilemma” are important in understanding the weakness of Microsoft’s growth strategy and the future impact of open-source alternatives to all commercial software companies. The fact that the pace of technological progress can, and often does, outstrip what markets need limits customers’ willingness to pay and causes diminishing returns on subsequent improvements. And more importantly, it means that disruptive technologies that may under perform today, relative to what users in the market demand, may be fully performance-competitive in that same market tomorrow. Also, the customers and financial structures of successful companies prevent investment in disruptive technologies, which provides lower margins, has insignificant markets, and is not attractive to the companies’ most profitable customers. The focus on providing what their best customers needs and identifying new market that promise greater profitability and growth often result in companies not investing in disruptive technology until it is too late. The author concluded that established companies can surmount this barrier of managing the conflicting demands of sustaining and disruptive technologies by first understanding what the intrinsic conflicts are.
Finally, Clemons and Bradley provided a description on the use of scenario analysis as a tool to explore strategic alternatives and possible strategic responses in response to strategic uncertainties. Scenario analysis is particular useful in times of rapid change when discontinuities in the business environment make extrapolation from available historical data misleading or meaningless. Scenario analysis attempts to identify the environment which a firm may have to operate, expressed as a set of scenarios that covers virtually any eventuality the firm may encounter. In the rapidly changing IT business, scenario analysis combined with system dynamics is a useful tool to predict future scenarios.

In chapter 6, I conducted a comparative analysis on IBM and BEA focusing on their technology acquisition strategies.

Both IBM and BEA benefited substantially from the technologies and new market opportunities through their acquisitions of complementary companies that enabled technology gaps to be filled, existing capabilities to be reinforced and the creation of new business revenue streams. They have placed great emphasis on the preservation of the acquired firms’ tacit and/or socially complex knowledge base by avoiding unnecessary relocation of employees and offices, controlling the pace of integration, providing much autonomy to the acquired companies and having appropriate financial incentives for retention of key staffs. However, there are natural trade-offs between autonomy and effectiveness of the transfer of values and best practices, and between slower integration to preserve the knowledge asset and the speed of transfer for these technologies and capabilities to the acquirer.

IBM’s key challenge is the organizational complexity brought about by the sometimes overlapping technology acquisitions. To resolve some of the problems brought about by long period of autonomy and slow pace integration, IBM has shifted from a slow integration to a rapid and immediate integration strategy when acquiring new companies.

On the other hand, BEA’s strategy of acquiring smaller companies with complementary non-overlapping technologies enables BEA to successfully evolve the various products into a unified architecture and platform despite the many technology acquisitions.

In chapter 7, I conducted a comparative analysis on Microsoft and BEA focusing on their strategies
in achieving platform leadership.

The value proposition of both Microsoft and BEA is to make technology more affordable, easier to use and less complex by developing an integrated platform suite through a continual cycle of incremental and occasionally radical innovations to integrate, link, package and often simply their products. Also, the technology acquisition strategies are very similar in that both usually acquire small companies with emerging technologies, not necessary mature ones, and then integrate the acquired technologies into the integrated product suite through ground up re-development if necessary to ensure seamless integration.

However, there are major differences in the execution. While both companies attempt to build integrated platform, the objective of Microsoft is to build an end-to-end integrated technology platform by itself. It placed a lower priority on gaining and maintaining trust of complementors over protection of its Windows dominance.

BEA, on the other hand, is a pure-play application infrastructure software vendor focusing only on the core infrastructure components and relies on partners to augment its product offerings. The focus and positioning to remain as an independent and pure-play application infrastructure software vendor makes BEA an attractive and trusted partner to application vendors and systems integrators. BEA’s robust J2EE implementation framework allows BEA to timely evolve its products to keep pace with rapid technological and market changes effectively.

In chapter 8, I analyzed the key market dynamics that affects the APS software industry. I first identified the key market drivers and their implications, both from external environment as well as those that are internal to the companies driven from their business and technology strategies. Then, I identified some key events or activities that have potential major impact to the future evolution of the industry and analyzed their dynamics.

In chapter 9, I explored the potential future scenarios in the APS software industry using scenario analysis and identified strategic options for companies in the industry in preparation for future challenges and in the event of strategic changes in the industry.

I first identified the strategic uncertainties by analyzing the key market drivers and dynamics explored in Chapter 8 and ranked them according to the degree of uncertainty and potential impact.
to select the three most critical unanswerable questions. The critical questions are: (1) Success of Microsoft’s entry into enterprise APS market; (2) Success of IBM in creating a unified development platform; and (3) Success of Microsoft’s entry into SAP’s core packaged business application market.

A qualitative assessment was done to determine the likely outcome of each of the three uncertainties and is summarized as follows: (1) The lack of trust by developers, complementors and IT product decision makers in Microsoft affects Microsoft’s penetration success into the enterprise APS market and the degree of success is assessed to be low to medium; (2) Although it remains unclear on the effectiveness of using consortium-based open-source Eclipse for IBM’s internal integration effort, the fact that IBM still contributes the majority of the development and financial resource increases the success rate and the degree of success is assessed to be medium to low; and (3) Although Microsoft will faced great challenges in unifying the separate business application products acquired through major acquisitions, its track record of developing winning products after several iterations and having a substantial R&D budget increases the probability of Microsoft establishing a foothold in the currently underserved SMB packaged business application market that acts as a beachhead for expanding into SAP’s enterprise market. The degree of success of Microsoft’s entry into SAP’s core packaged business application market is assessed to be medium to low.

The likely outcome of the key uncertainties formed the base case scenario from which alternative views of the future were developed. Combining the key uncertainties, five strategic scenarios were presented with each having different major winners who are able to build and sustain their competitive advantages and capture large portions of the market share. The major winners of the five scenarios are: (1) BEA, IBM and SAP; (2) BEA and SAP; (3) IBM and SAP; (4) Microsoft, IBM and SAP; and (5) Microsoft. For each of the scenarios, I explored how customers, executives in the firm, and executives in competing firms would feel and react to the scenarios. The likely responses have been summarized in the following table.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Major winners</th>
<th>Likely responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>BEA, IBM and SAP</td>
<td>• Microsoft changes its fundamental growth strategy and make strategic shift to innovate at the higher level software stack above the operating system.</td>
</tr>
</tbody>
</table>
| 2. | BEA and SAP | - SAP shifts to a Type M3 value chain model to focus on the applications business to compete more effectively with Microsoft.  
- IBM disrupts the APS market by supporting or contributing its APS software to the open-source community and profit as open-source integrator and distributor.  
- Many commercial software companies move to second-generation open-source model and shift revenue source from products to services. |
| 3. | IBM and SAP | - BEA seeks closer relationship with strategic partners, or even contemplates being acquired to increase its financial strength and size to take on IBM.  
- Microsoft and BEA seeks closer co-operation in their collective confrontation with IBM. |
| 4. | Microsoft, IBM and SAP | - BEA is acquired by one of the potential candidates such as HP, SAP and Oracle.  
- Microsoft and IBM are unlikely to join the bid to acquire BEA for antitrust reasons. |
| 5. | Microsoft | - IBM and SAP uses open-source to disrupt the APS market.  
- SAP shifts to a Type M3 value chain model to focus on the applications business to compete more effectively with Microsoft. |

In conclusion, I have identified four strategic options that the APS software industry players can consider to invest and position themselves strategically and be able to respond quickly in the advent of any of the likely scenarios, while waiting for the environment to become clearer. First, Microsoft should fundamentally change its business and technology strategy to shift the likely outcome towards Scenario 4: Microsoft, IBM and SAP. Second, SAP should consider progressively moving to a Phase M/Type M3 value chain model, and BEA and IBM should consider how to position themselves as the preferred choice of the IT industry’s back-end system provider. Third, Microsoft
and BEA should invest in small open-source experimentation to understand the open-source dynamics. And lastly, BEA should plan for the contingency of being an acquisition target with the primary aim of sustaining its fundamental position of remaining as an independent and pure-play infrastructure software vendor.

This thesis did not present implementation strategies. However, I would suggest that the following key points, which I believe are essential in formulating a sustainable growth strategy in the software industry, be taken into considerations. First, Christensen and Raynor highlighted that the pace of technological progress can, and often does, outstrip what markets need limits customers' willingness to pay, and makes it possible for disruptive technologies that may under perform today to one day become fully performance-competitive and eliminate the market share and profitability of existing high-performance premium products. The power of open-source software should not be underestimated by commercial software companies. Eric S. Raymond, one of the co-founder of Open Source Initiative, wrote in the closing of the article "The Cathedral and the Bazaar", which analyzes the critical success factors of Linux and other successful open-source projects:

"Perhaps in the end the open-source culture will triumph simply because the commercial world cannot win an evolutionary arms race with open-source communities that can put orders of magnitude more skilled time into a problem".

Second, trust plays a fundamental role in the sustenance of long term relationship and customer retention. For sustainable growth, it is important to understand the true meaning of customer ownership. In the foreword by McKenna in the book "Crossing the Chasm", he aptly defines the essence of ownership.

"Customers do not like to be "owned," if that implies lack of choice or freedom. The open systems movement in high tech is a clear example of that. But they do like to be "owned" if what that means is a vendor taking ongoing responsibility for the success of their joint ventures. Ownership in this sense means abiding commitment and a strong sense of mutuality in the development of the marketplace. When customers encounter this kind of

ownership, they tend to become fanatically loyal to their supplier, which in turn builds a stable economic base for profitability and growth.

And lastly, I think that IBM’s and BEA’s strategy of virtual integration to build a vast partner ecosystem in order to provide complete solutions to customer needs, instead of Microsoft’s vertical integration strategy, may be the only sustainable growth model that can potentially break the infinite double loop that cycles between vertically integrated industries inhabited by large companies and horizontally integrated industries populated by many innovative companies.
Chapter 6 & 7: Comparative Analysis
- IBM versus BEA
  - IBM's key challenge is the organizational complexity brought about by the sometimes overlapping technology acquisitions.
  - BEA's strategy of acquiring smaller companies with complementary non-overlapping technologies enables BEA to successfully evolve the various products into a unified architecture and platform.
- Microsoft versus BEA
  - Similar technology acquisition strategies of usually acquiring small companies with emerging technologies, and then integrate the acquired technologies into the integrated product suite through ground up re-development.
  - The objective of Microsoft is to build an end-to-end integrated technology platform by itself. It placed a lower priority on gaining and maintaining trust of complementors over protection of Windows dominance.
  - BEA is a pure-play application infrastructure software vendor focusing only on the core infrastructure components and relies on partners to augment its product offerings. The focus and positioning to remain as an independent and pure-play application infrastructure software vendor makes BEA an attractive and trusted partner to application vendors and systems integrators.

Chapter 8: Key Market Dynamics
- Business & technology strategy, company challenges

Chapter 9: Potential Future Scenarios and Proposed Strategies
- The three most critical unanswerable questions are:
  1. Success of Microsoft's entry into enterprise APS market.
  2. Success of IBM in creating a unified development platform.
  3. Success of Microsoft's entry into SAP's core packaged business application market.
- The four strategic options are:
  1. Microsoft should fundamentally change its business and technology strategy to shift the likely outcome towards Scenario 4: Microsoft, IBM and SAP.
  2. SAP should consider progressively moving to a Phase M/Type M3 value chain model, and BEA and IBM should consider how to position themselves as the preferred choice of the IT industry's back-end system provider.
  3. Microsoft and BEA should invest in small open-source experimentation to understand the open-source dynamics.
  4. BEA should plan for the contingency of being an acquisition target with the primary aim of sustaining its fundamental position of remaining as an independent and pure-play infrastructure software vendor.

Figure 10-1 Key points of this thesis
## 11. Appendices

### Appendix A – Acquisitions of IBM

<table>
<thead>
<tr>
<th>No</th>
<th>Company (product)</th>
<th>Time</th>
<th>Expertise</th>
<th>Motivations and Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lotus Development Corporation</td>
<td>1995</td>
<td>Workgroup collaboration software</td>
<td>IBM shifts software focus from the declining mainframe related software to fast growing distributed software and gets presence on the desktop with Lotus Notes, which IBM is able to sell in large quantities to its large existing customer base. &quot;In acquiring Lotus ... we simultaneously filled a hole in our portfolio, made a bold move into the world of networked computing, and announced that IBM was back.&quot; (2002 annual report).</td>
</tr>
<tr>
<td>2</td>
<td>Tivoli Systems Inc.</td>
<td>1996</td>
<td>System management software</td>
<td>IBM's acquisition of Tivoli extends the company's strength in host-based systems management to multiplatform distributed systems. The merger will combine Tivoli's advanced technology with IBM's host-based systems management products and global sales, service and support. Together, the companies will provide customers with the most comprehensive, open systems management solution in the industry for network-centric computing. &quot;Demand for distributed systems management products is exploding, and customers are enthusiastically adopting the Tivoli Management Environment. IBM's global reach and complementary management products for hosts and networks will quickly allow us to achieve a worldwide leadership position&quot; (Tivoli CEO Frank Moss).</td>
</tr>
<tr>
<td></td>
<td>Company</td>
<td>Year</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------</td>
<td>-------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>NetObjects, Inc.</td>
<td>1997</td>
<td>IBM’s purchase of a majority interest in NetObjects, a leading provider of website development tools (Fusion) for designers and intranet developers, was motivated by the importance of development tools in controlling the developer base and the web standards. By acquiring a leading website development tool provider IBM aimed to prevent Microsoft from setting de-facto standards that would favor Microsoft architecture and operating systems.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CrossWorlds Software, Inc.</td>
<td>2001</td>
<td>IBM acquired CrossWorlds Software Inc., a developer of enterprise application integration (EAI) technology through a $129 million deal. CrossWorlds would fill holes in IBM’s environment for e-business EAI. CrossWorlds supplies software that enables companies to automate business processes that integrate multiple applications such as customer relationship management (CRM) and supply chain management (SCM).</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Access360</td>
<td>Sep-02</td>
<td>IBM integrated Access360 into its Tivoli systems management portfolio. IBM integrated functionality from its own Tivoli Access Manager and Privacy Manager products to form a new identity management offering with Access360’s enRole software at its core. With the combined Tivoli capabilities and Access360 capabilities, IBM had a world class offering for end-to-end security identity management. In the future, IBM Tivoli will take care more of security management than network management.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Holosofx</td>
<td>Sep-02</td>
<td>IBM acquired Holosofx to extend its portfolio of business integration infrastructure software. Holosofx's business process modeling (BPM) and monitoring tools, as part of IBM's WebSphere Business Integration platform, helps customers implement powerful business integration solutions. With the addition of Holosofx products, IBM's WebSphere Business Integration platform provides end-to-end capability for a customer to design, deploy, monitor and improve business processes across the enterprise and with its business partners. This acquisition gives IBM product capability in business process modeling - tools used by analysts to define the logic within a business process.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Rational Software Corp</td>
<td>Dec-02</td>
<td>Rational provides open, industry standard tools, best practices and services for developing business applications and building software products and systems. This deal extends IBM's ability to help customers into the 'on demand' future with tools built on industry standards to develop, integrate and manage their business processes. IDC estimates the market opportunity for application development software will grow from $9 billion in 2002 to $15 billion in 2006. Additionally, Rational gives IBM a means of reconciling the disparate development worlds of J2EE.</td>
<td></td>
</tr>
</tbody>
</table>
## Appendix B – Acquisitions of BEA

<table>
<thead>
<tr>
<th>No</th>
<th>Company (product)</th>
<th>Time</th>
<th>Expertise</th>
<th>Motivations and Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Technology</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>acquisitions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Novell Inc.</td>
<td>Feb 1996</td>
<td>Transaction Processing Monitor</td>
<td>BEA's first acquisition is in Transaction Processing Monitor. BEA had a vision that the age of centralized mainframe computing was giving way to a client-server environment. And it recognizes the importance of &quot;middleware&quot; which coordinates application running over heterogeneous operating systems. BEA wasn't planning to build anything from scratch. It bought the rights to TUXEDO as well as much of its preciously independent distribution network. BEA was confident that the shift to distributed computing will peak soon and the product will meet the growing demand.</td>
</tr>
<tr>
<td></td>
<td>- Tuxedo Technology</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Digital Equipment Corporation</td>
<td>Feb 1997</td>
<td>Message Oriented Middleware and Object request broker</td>
<td>Besides transaction processing monitor, the distributed computing required other functions such as message queuing and object request broker. BEA acquired these capabilities from DEC.</td>
</tr>
<tr>
<td></td>
<td>- Object Broker, DECmessageQ, etc</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>WebLogic Inc</td>
<td>Sep 1998</td>
<td>Java Application Server</td>
<td>Preparing for the rapid growth of eBusiness, BEA had to acquire two technologies, JAVA and web application server. BEA aimed to integrate its existing M3 middleware platform (a platform developed using technologies acquired from Novell and DEC) with web application server. WebLogic’s Tengah Application Server was one of the most advanced enterprise Java application servers</td>
</tr>
<tr>
<td></td>
<td>The Theory Center</td>
<td>Nov 1999</td>
<td>Reusable EJB components</td>
<td>&quot;This move is as strategic as our acquisition of WebLogic&quot;, BEA CEO Bill Coleman said. The Theory Center's technology, which includes tools, templates and methodologies, along with the EJB components, will &quot;catapult BEA into the market for quickly delivering e-commerce-on-demand solutions based on Java components,&quot; Coleman said. The Theory Center's component line includes EJBs for customer relationship management (CRM), online order management, product catalogs, trouble ticket management, and workflow management.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>5</td>
<td>Workflow Automation Corporation</td>
<td>Mar 2000</td>
<td>Workflow technology</td>
<td>Many companies were opting for the outsourcing model to focus on their core capabilities. In the outsourcing model, companies have to aggregate their partners, add partners to their network dynamically, and collaborate with them efficiently. To meet these needs, BEA acquired from Workflow Automation Corporation primarily for its jFlow business process engine - a Java workflow management system that performs application integration across the Internet. BEA planned to integrate this technology into its business-to-business collaboration architecture through an initiative called</td>
</tr>
<tr>
<td></td>
<td>CrossGain Corporation</td>
<td>Jul 2001</td>
<td>Development tool</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------</td>
<td>----------</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To provide easy-to-use IDE is one of key strategy to expand the share of server applications by attracting new and mid-skilled programmers to develop using BEA’s platform.

BEA acquired CrossGain Corporation, and retained the services of its CEO and former Microsoft executive Tod Nielsen and XML guru Adam Bosworth. BEA aimed to provide an excellent IDE that leverages on XML to deliver Web Services and turn the IDE into the Java equivalent of Microsoft's easy to use Visual Studio tools.

<table>
<thead>
<tr>
<th></th>
<th>Appeal Virtual Machines AB</th>
<th>Feb 2002</th>
<th>Product run-time performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Java Virtual Machine (JVM) is one of the critical run-time components that determine the performance of the application server. BEA sought JVM expertise in delivering performance on Intel-based systems. Appeal's JVM product, known as JRockit, has very high performance and is optimized for every type of platform. The acquisition enables BEA to further boost the performance of its platform.

Project e-Collaborate, and re-launch the product in the third quarter of 2000 as BEA e-Process Integrator.
<table>
<thead>
<tr>
<th>#</th>
<th>Company Name</th>
<th>Date</th>
<th>Acquisition Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>CrossLogix</td>
<td>Feb 2003</td>
<td>Enterprise authorization infrastructure</td>
<td>&quot;Everybody that is selling application platforms has found that people want some kind of access management built in,&quot; said John Pescatore, an analyst at Gartner. The increasing demands to robust and secure infrastructure required BEA to integrate security within their product line.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Consulting and educational resource acquisitions</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Component Systems</td>
<td>Jun 1999</td>
<td>Architectural expertise for Component-based application development</td>
<td>BEA's customers were increasingly using the technology for building their next generation e-commerce applications, and they requested BEA to provide more in-depth services to help them architect these new systems. Thus BEA chose to acquire Component Systems because BEA had been working with consultants from Component Systems on major customer projects over the past two years and experienced highly successful developments.</td>
</tr>
<tr>
<td>10</td>
<td>Technology Resource Group, Inc. (TRGI)</td>
<td>Jul 1999</td>
<td>Education Resource</td>
<td>BEA invested not only in Java technology, but also in its ability to educate and assist companies to transform themselves into e-businesses. TRGI is a leading Java information technology education, mentoring, and consulting organization. The acquisition brings a proven Java educational force in-house to strengthen BEA's industry leadership in providing end-to-end education for customers who seek to build e-commerce applications using cutting-edge, Web-optimized languages and technologies.</td>
</tr>
<tr>
<td></td>
<td>Company</td>
<td>Date</td>
<td>Activity Description</td>
<td>Quote or Details</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------</td>
<td>-------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>11</td>
<td>Avitek, Inc.</td>
<td>Aug 1999</td>
<td>Development of Java solutions for front-office e-commerce applications such as sales automation and order entry</td>
<td>&quot;With the acquisition of Component Systems, TRGI and Avitek, BEA provides an end-to-end solution for building and deploying component-based applications for e-commerce,&quot; said Deborah Stanley, president of BEA’s eSolutions business unit.</td>
</tr>
<tr>
<td>12</td>
<td>The Object People</td>
<td>Apr 2000</td>
<td>Consulting and Educational Resource</td>
<td>BEA acquired the consulting and educational arms of The Object People Inc to further strengthen its resources in these areas.</td>
</tr>
<tr>
<td>13</td>
<td>Softport Systems, Inc</td>
<td>Jun 2000</td>
<td>Consulting Resource, especially in the financial and banking areas</td>
<td>BEA aimed to become a leader of total software solutions for anyone who wants to do business on the Internet. BEA bought Softport Systems, Inc. for its prowess in e-commerce consulting and application development services, particularly in the financial services and banking markets.</td>
</tr>
<tr>
<td>14</td>
<td>Bauhaus Technologies, Inc.</td>
<td>Oct 2000</td>
<td>Consulting Resource</td>
<td>Bauhaus provided design, development, and integration services using technologies such as Java and XML. The addition of Bauhaus’s 27 consultants enhances BEA’s professional service practice.</td>
</tr>
</tbody>
</table>
12. References

Information sources for research on IBM's technology acquisition strategy in Chapter 6


Information sources for research on BEA's technology acquisition strategy in Chapter 6

13. Bibliography


Books


Other readings


