Communications of the Association for Information Systems

Volume 19   Article 1

March 2007

Monitoring Platform Emergence: Guidelines from Software Networks

Bala Iyer
Boston University, bala@bu.edu

Chi-Hyon Lee
George Mason University

N. Venkatramen
Boston University, venkat@bu.edu

Dan Vesset
IDC

Follow this and additional works at: https://aisel.aisnet.org/cais

MONITORING PLATFORM EMERGENCE: GUIDELINES FROM SOFTWARE NETWORKS

Bala Iyer
Babson College
biyer@babson.edu

Chi-Hyon Lee
George Mason University

N. Venkatraman
Boston University

Dan Vesset
IDC

ABSTRACT

In this paper we explore how platforms emerge and evolve due to independent actions by companies providing them or launching products on them. We use the software industry as the setting for our study. We analyze the pattern of evolution for Windows, Unix, and Linux over 14 years. Based on this, we derive some lessons for companies aspiring to compete in settings where platforms and complementors play a major role. We support our analysis using visualizations.

Keywords: architecture, platforms, complementors, strategy, visualization

I. INTRODUCTION

As we shift away from the industrial age to a setting characterized by increasing reliance on information and communication technologies, new concepts become central to how companies craft their business strategies. Competencies are not limited to what a firm has inside, as can be seen when firms strive to complement their internal competencies with those of its alliances and partnerships. In such settings, the focus is no longer about competition between products, but about competition between platforms. For example: in the video-game sector, the competitive battle is not between Sony PS3 and Microsoft Xbox in terms of product features. It is between the PlayStation versus Xbox platforms, where the strength of the platform is based on the degree of third-party support that the platform enjoys via the availability of videogame titles that run on these platforms. So, Microsoft and Sony compete not just on the technical features that differentiate their consoles but also on their ability to orchestrate their network of third-party support. This is the emergent idea of platform-based competition that calls for innovation ecosystems [Adner 2006].

In this paper, we argue that this type of competition is not limited to videogames but has relevance to many other industries, and that managers need to pay particular attention to

understanding platform emergence and developing appropriate responses. [Bresnahan and Greenstein 1999] pointed out analytically that competition in the computer industry is between computer platforms with their associated network of complements. [Gawer and Cusumano 2002] described the lessons learned by companies such as Microsoft, Cisco and Intel that compete based on platforms. Using Intel as a case in point, Gawer and Henderson [2005] described how a platform provider chooses to enter complementary markets. These dynamic markets require an astute understanding of how platforms emerge and evolve through the interdependent actions of platform players and supporting component developers. What makes this particularly important is that the interactions are not static but evolve dynamically over time. By studying platforms in the software industry, we develop a set of implications for managers who may be finding their settings under transition to platform-based competition.

The software industry serves as an excellent setting to study platform emergence because of the rise, maturity, and fall of platforms such as Mainframe, Windows, Unix, and Linux over the last three decades. It is also timely because the rise of Linux as a viable alternative to Windows is not just due to the open source movement but also due to the support provided by firms such as IBM, HP, Sun, and Novell. To understand these software platforms, we observe and analyze the patterns of evolution of Windows, Unix, and Linux platforms and their associated ecosystems of third-party support over 14 years. Our choice of Windows recognizes its dominance over the last decade; our choice of Linux is because it is heralded as the platform of the future. We include Unix as it had a significant market share prior to Windows emerging as a dominant platform and shares many technical features with Linux. Although we study software platforms within the packaged software industry, our implications have broader applicability in setting such as the mobile devices, video games and automotive industry.

Our paper is organized as follows. In the next section we describe software platforms and the role of complementors. In section 3, we describe our logic for visualizing software networks, and follow that with a description of the software industry as a network at five different points in time in section 4. In section 5, we present the network as seen from the perspective of a software stack. In sections 6 and 7, we draw some implications and identify some leading indicators to track. We end with conclusions in section 8.

II. ANALYSIS OF SOFTWARE PLATFORMS

Software companies deliver a product that functions with complementary components from other companies to deliver business value. This calls for software companies to understand their positions within the ecosystem [Iyer, Lee et al. 2006]. Two dominant strategies emerge within the software sector: 1) they make their products as modules that subscribe to the architecture specified by another company (i.e., a platform provider); or 2) they seek to become a platform architect by gaining the support of complementary developers.

A platform is successful only if a set of complementors support it; this phenomenon is often referred to as network effects or system-based competition [Shapiro and Varian 1999; Messerschmitt and Szyperski 2003]. The success of a platform can be seen through the interaction of two forces: the degree of acceptance and adoption by customers ("customer network effects") and the availability of supporting software modules ("complementary network effects"). Often, complementary network effects act as a lead indicator of the future success of a platform, although the network effects are highly interdependent. Research work done under the multi-sided market stream describes how platform owners can use differential pricing to promote network effects and adoption [Rochet and Tirole 2003]. By this logic, we should be able to see if a platform is viable by looking at the support it enjoys within the set of companies that design and deliver software products on a platform. [Venkatraman and Lee 2004] examined how network effects and other variables such as newness impacted a developer's platform adoption decision. They found that software developers were less likely to launch products on a platform when other developers already had a strong presence on the platform.

While we find information about sales to the end user of various platforms from the business press and analysts, we know very little about complementors that build software that works on top of these platforms. So, we began by looking at the data on the number of software products sold by companies supporting different platforms since 1990. Our analysis is based on data that has been collected by the International Data Corporation (IDC). Since 1990, IDC has systematically tracked the global software sales of more than 1,200 software companies. This database is widely considered the best source on the software industry; it has been used in the U.S. DOJ case against Microsoft and in academic research. The IDC database contains the global revenue generated by software companies at the level of products (e.g., SAS/Stat or PeopleSoft EnterpriseOne) for specific platforms. We excluded revenue from other sources (e.g., services) and focused on nine different platforms: We consider Unix, Linux, and Windows separately and aggregate the rest into "other." In effect, we focus on the network consisting of Unix and its variants (e.g., IBM's AIX, Sun's Solaris, and Hewlett-Packard's HP-UX), Windows and its variants (e.g., Windows 3.x, 32, and 64), and Linux.

Our first quest was to see how many companies offer software products on various platforms. Figure 1 indicates that there is a steady increase in the percentage of software companies supporting Linux. As of December 2004, nearly 21 percent of the firms in the software industry support Linux. Similarly, when we look at the Windows platform, we find that over 80 percent of the independent software vendors [ISVs] support the platform by having at least one product that runs on the platform. Unix, on the other hand, is seeing its support base gradually eroding. While it once enjoyed the support of nearly 80 percent of the independent software companies, today its share has been reduced to just over 60 percent. It is also clear from our analysis that most companies *derive only a small percentage of their revenues from Linux*. Windows [along with Unix], as expected, provides a big share of the revenue stream. We go further to explore how the network of complementors for the various platforms has evolved between 1990 and 2004.
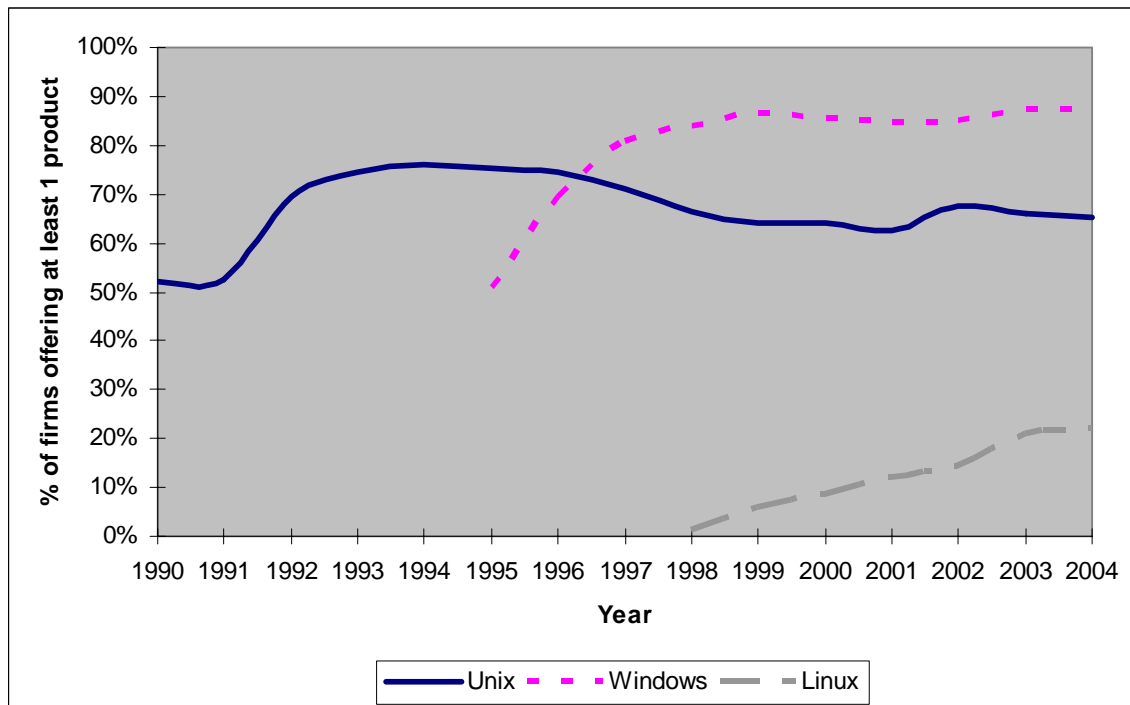

Figure 1. Software Product Companies Supporting Linux, Unix and Windows

III. ANALYZING THE NETWORK

In the U.S. DOJ case against Microsoft, a key contention was that Microsoft enjoyed *applications barriers to entry*.   Judge Thomas Penfield Jackson based his ruling against Microsoft on the

claim that the company's monopoly in operating systems is protected by an applications barrier to entry made up of over 70,000 Windows-based software programs. As a result, to enter the operating-system market a newcomer would need a large and varied base of compatible applications like those available to consumers who might otherwise choose Windows. So, to understand platforms, we need to look at the network of complements supporting it.

While we can show the impact of complementary products using statistical techniques, we chose visualization for its explanatory power and ease of understanding. We build the network using the following components: software developer firms that build complementary products; platforms that provide the basis to build complements; and links that connect a platform to a firm. Firms are depicted as circles, platforms as squares, and links as lines. A link connects a firm to a platform if and only if the firm sells a product on that platform. When a firm sells products on more than one platform, a separate link is created to each platform.

Our network is color coded to show sales percentages as follows: We draw a link between a developer and a platform environment if the developer sells any package that runs on that platform. The link color is defined as follows: light green if the developer generates less than 15 percent of sales from that platform; green if it generates between 15 percent and 30 percent of sales on that platform; spring green if the sales are between 30 percent and 45 percent; golden red if sales are between 45 percent and 60 percent; yellow if sales are between 60 percent and 70 percent; orange if sales are between 75 percent and 90 percent; and red if the percentage of sales from that platform is over 90 percent.

The size of the node indicates the total revenue of a software product company. The software company nodes are color-coded as follows: first time supporters of a platform are colored yellow; purple denotes companies that have supported the platform in previous years; and finally, magenta denotes firms that have yet to support the platform at that time. Platforms are represented by green squares. The size of the platform is based on the total revenue generated for that platform (a natural log transform of that).

Using Pajek network visualization software (http://vlado.fmf.uni-lj.si/pub/networks/pajek/), we present a series of discrete-time images of the evolution of the software network. Pajek employs "spring-embedded" drawing algorithms to represent network data in two-dimensional Euclidian space. It has been recently used to model business networks [Venkatraman and Lee 2004; Powell, White et al. 2005].

IV. VISUALIZING THE NETWORK OF PLATFORMS AND COMPLEMENTS

We present five network pictures representing the years 1995, 1998, 2000, 2002, and 2004. Each picture captures the state of the network at that point in time and we discuss the evolution of the network in terms of how the platforms become dominant through supporting complements. If the platform has an increasing number of complementors, it is gaining in network effects. Similarly, if more complementors are dependent on the platform for a major percentage (100 percent if they support it exclusively) of their sales, the platform has greater staying power.
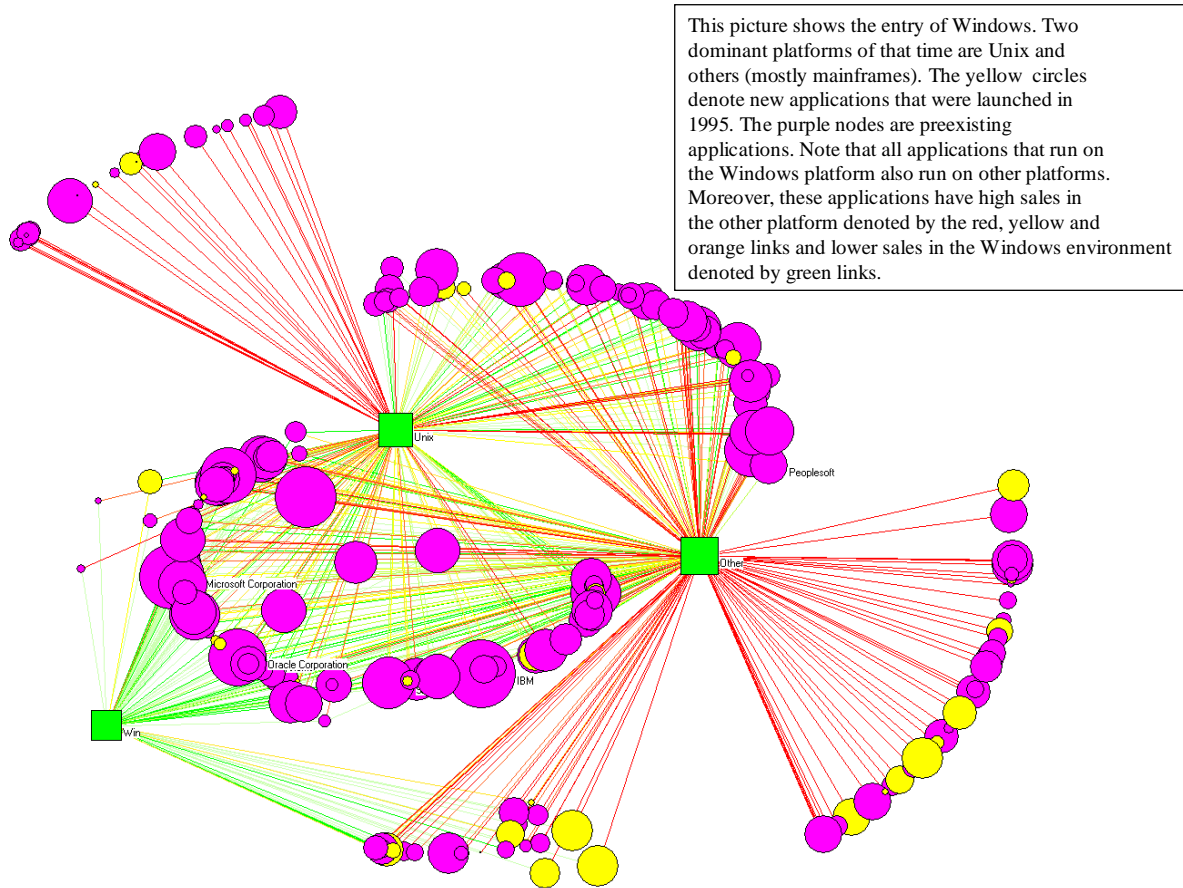
This picture shows the entry of Windows. Two dominant platforms of that time are Unix and others (mostly mainframes). The yellow circles denote new applications that were launched in 1995. The purple nodes are preexisting applications. Note that all applications that run on the Windows platform also run on other platforms. Moreover, these applications have high sales in the other platform denoted by the red, yellow and orange links and lower sales in the Windows environment denoted by green links.

Figure 2. Software Network -- 1995

Figure 2 shows the software network in 1995. The two dominant platforms are Unix and other mainframe based platforms; Windows was a new platform. It is clear that most software product companies played it safe by porting their existing applications from Unix and other mainframes to the new entrant's [Windows] platform. Although Windows dominated the desktop, none of the enterprise applications ran exclusively on the Windows platform. A possible explanation is that since Windows was a newcomer to the enterprise market, software companies that had successful products on other platforms did not see the value in risking the current user base. At the same time, some made a strategic bet that Windows would expand beyond the desktop into the enterprise market by developing applications that operated on both preexisting platforms and Windows. *In 1995, the installed base of enterprise applications is with Unix and mainframes; software vendors supported Windows but not with exclusive software applications.*

Figure 3 shows the network in 1998 when Linux is first tracked as a new software platform. By now, we can see that many software developers support Windows exclusively. Only one independent software firm supports Linux exclusively; one supports both Linux and Unix; and five firms support Linux, Unix and Windows. Except for the exclusive developer, all other links to Linux are shaded green (i.e., they generate less than 15 percent of their revenue from Linux). Look at the support for Windows; it has close to 100 software firms that support it exclusively. Moreover, many firms support Windows along with other platforms such as Unix. It is interesting to note that these links are all either orange or red indicating strong commitment to Windows (i.e., greater than 90 percent of the firm's sales comes from Windows). At the end of 1998, *the Windows platform is becoming dominant and many big firms are beginning to support the three dominant platforms (mainframe, Unix and Windows)*.
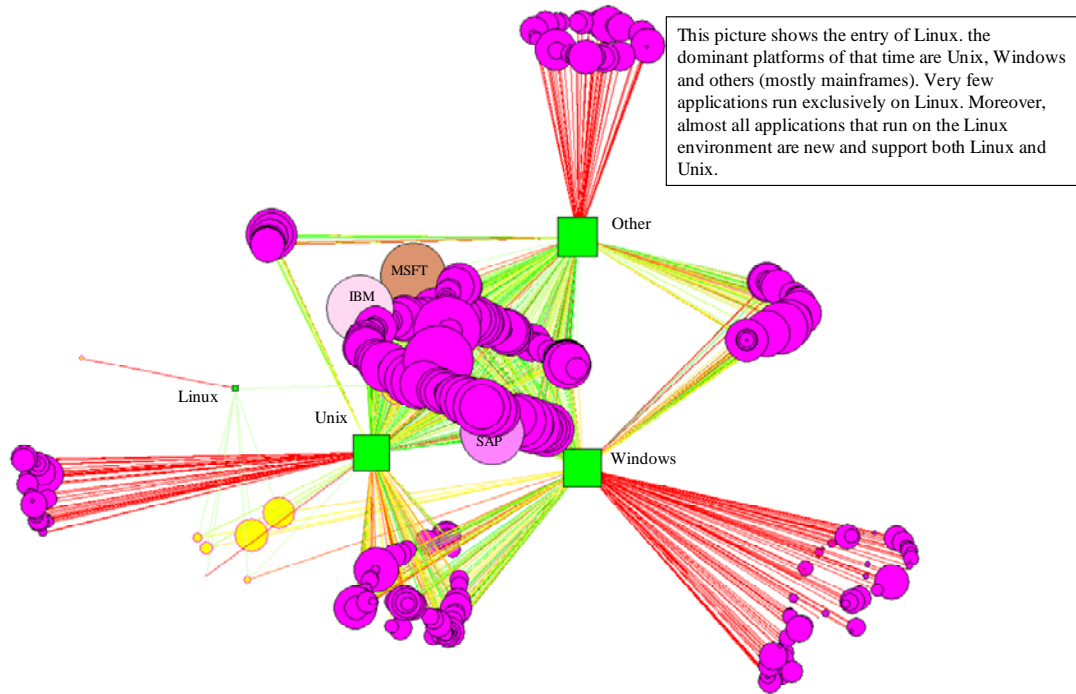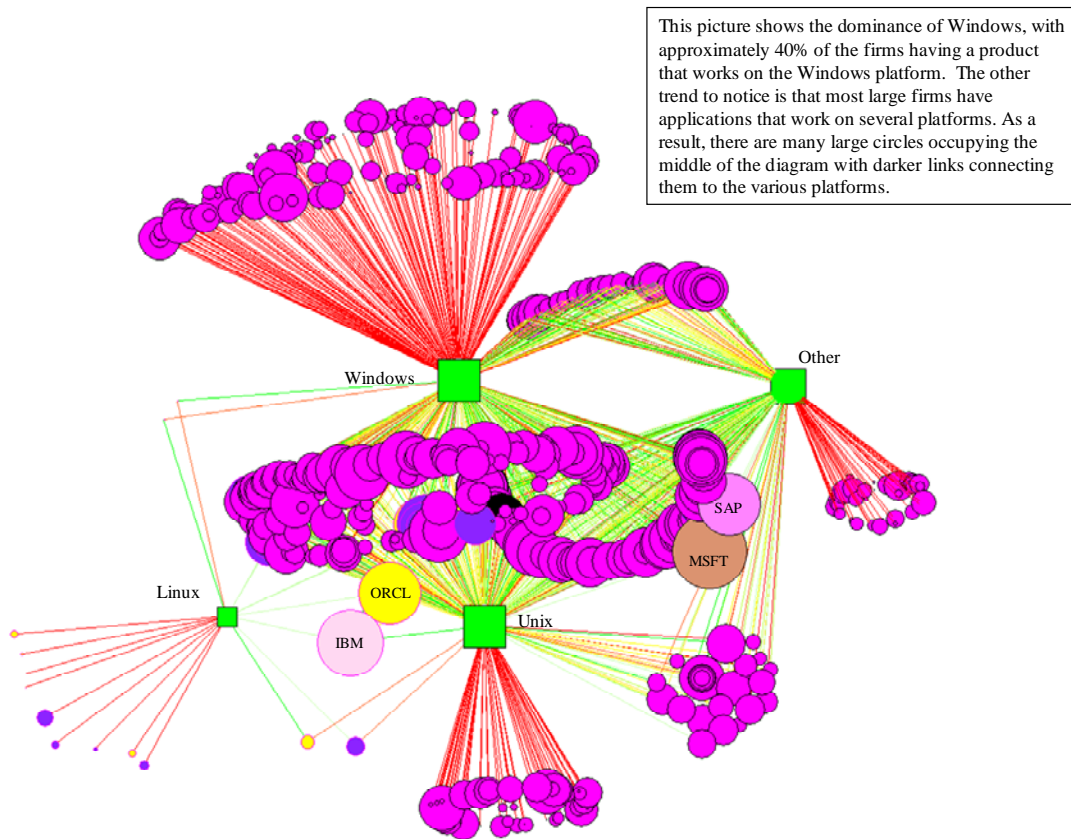
This picture shows the entry of Linux. the dominant platforms of that time are Unix, Windows and others (mostly mainframes). Very few applications run exclusively on Linux. Moreover, almost all applications that run on the Linux environment are new and support both Linux and Unix.

Figure 3. Software Network -- 1998

This picture shows the dominance of Windows, with approximately 40% of the firms having a product that works on the Windows platform. The other trend to notice is that most large firms have applications that work on several platforms. As a result, there are many large circles occupying the middle of the diagram with darker links connecting them to the various platforms.

Figure 4. Software Network -- 2000

Monitoring Platform Emergence: Guidelines From Software Networks by B. Iyer, C. Lee, N. Venkatraman, and D. Vesset

Now look at Figure 4, representing the network at the end of 2000. Most large firms are supporting more than one platform. Software vendors supporting Linux have grown to about 50 firms. Oracle and IBM now support all major platforms including Linux. It is not surprising that Microsoft and SAP still do not support Linux. Microsoft was still trying to get exclusive commitment to its platform (this has changed recently with their announcement to partner with Novell and make their version of Linux interoperable with Windows). Since SAP competes in the application layer of the software stack, their adoption decision depends on the lower layers being interoperable with Linux. Since not many products exist in the lower layers of the Linux stack, SAP has to delay their adoption decision. During 1998-2000, Linux as a platform attracted few exclusive developers: it appears that developers that do support it provide symbolic support— denoted by green links (showing low share of sales) that connect developers to Linux. As a comparison, look at the links to Windows or Unix: the number of software firms supporting each platform is higher and the links show substantive commitment (as shown by red links and several links with yellow/orange/red colored links from firms that support multiple platforms). Now, look at the center of the picture: many larger and important firms are at the center. This region shows firms that support many platforms. To the right are firms (Microsoft, SAP) that support the Windows, Unix, and other platforms. To the left are firms (Oracle, IBM) that support all four platforms including Linux. However, these firms are connected by green links to Linux, showing weak commitment. IBM announced a one-billion-dollar commitment to Linux but this has not yet translated into software product sales. *Windows and Unix/mainframe platforms dominate the software network; Linux has some support from complementors but is not yet a strong contender as a platform.*
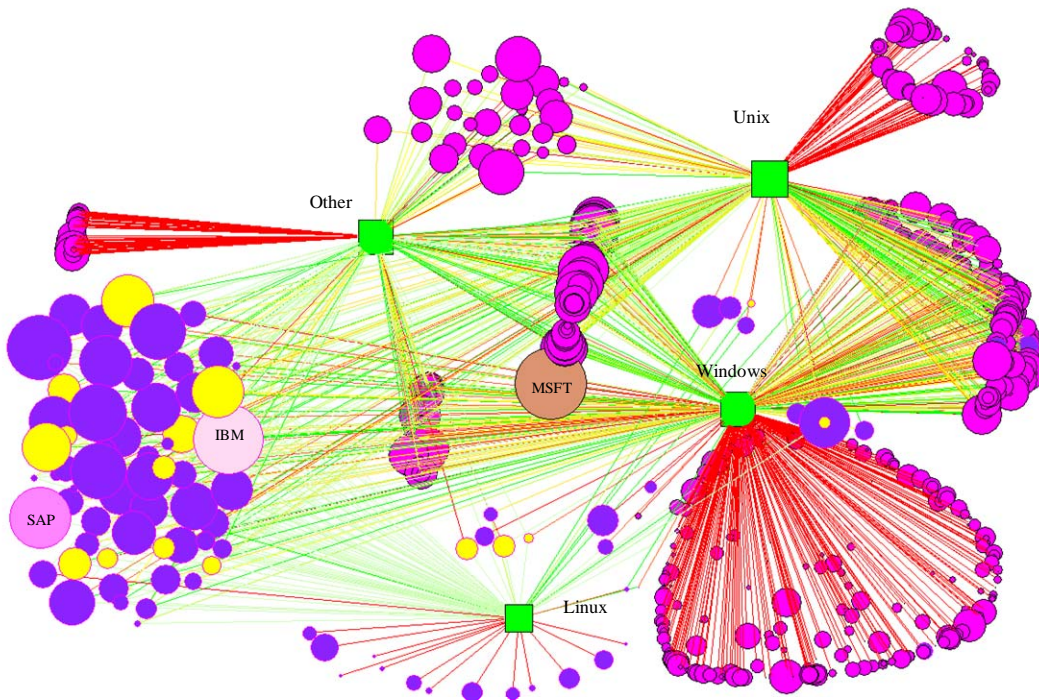

Figure 5. Software Network -- 2002

Now, look at the network at the end of 2002 (Figure 5). Linux is beginning to show stronger network effects: more firms are linked to it. Very few software firms are exclusive to Linux (< 2%).

Interestingly first-time supporters of Linux support multiple platforms as firms to the left of the page show commitment to all platforms including Linux. As in 2000, these firms are connected by green links to Linux indicating that they exhibit token commitment. Almost 75 firms including SAP and IBM support all platforms. Microsoft shows symbolic commitment to Unix and no commitment to Linux. Thus, by 2002, Linux—despite the buzz in the trade press—has not yet emerged as a serious alternative platform. The third-party complementor network is small and weak even six years after introduction of this platform. Over 80 percent of software product companies do not yet have any product on Linux. While Linux may have received symbolic support from many of the bigger software companies, the support is not substantive yet. At the same time, the networks supporting Windows and Unix do not seem to be getting weaker. *Thus, the new entrant (Linux) is gaining support, but the complementor network is not big enough to make it a serious contender to the other platforms.*
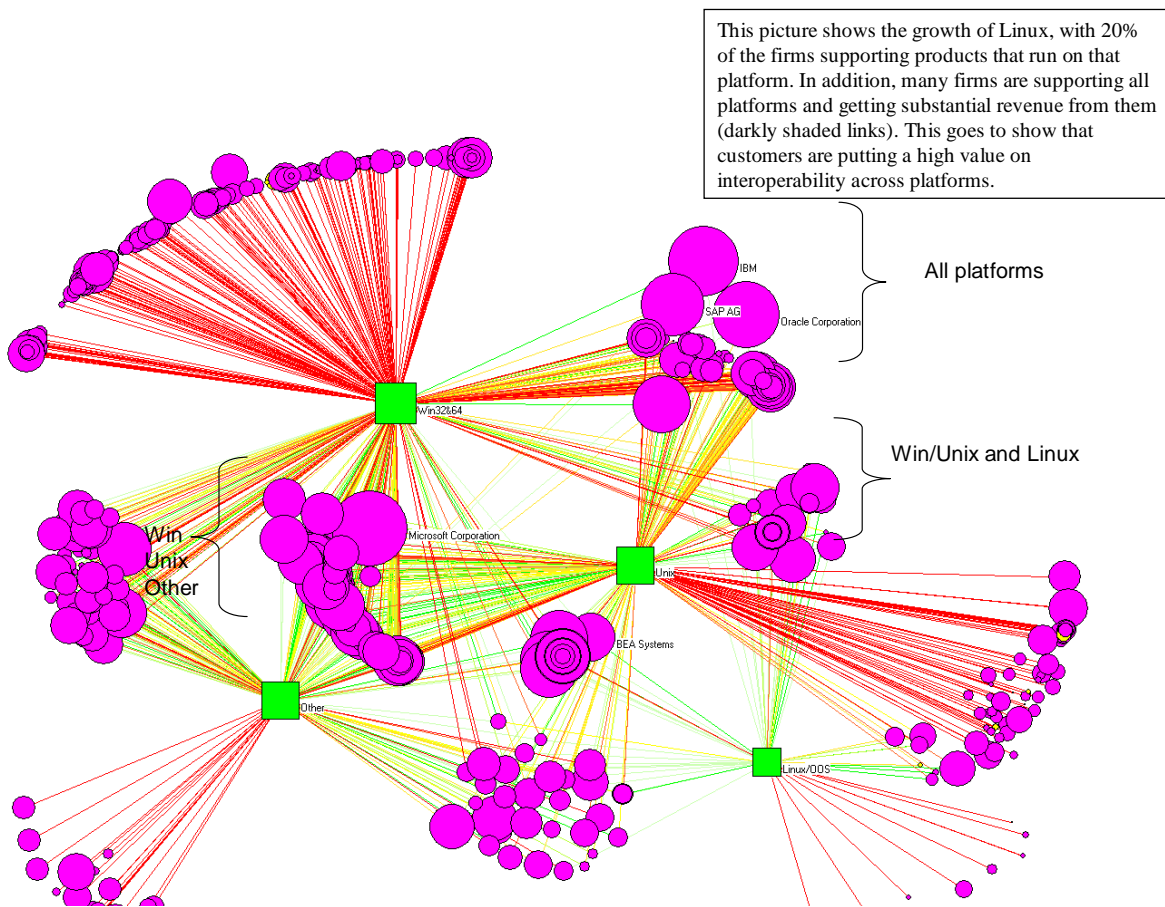
Figure 6. Software Network – 2004

Finally, let us look at the network at the end of 2004 (Figure 6). Linux looks like Microsoft in 1995 – most of the complementors have products that work on Windows and other platforms like Unix and mainframes. Many links to the Linux platform are yellow or green in color indicating that some companies are getting over 15 percent of their revenue from Linux. Meanwhile, number of developers for the Windows platform has shown a slight growth. While the number of software firms committed to Unix has steadily declined, there is a steady increase in the number of firms offering applications that work on many platforms. *Linux still has not gained any substantive traction from complementors.*

V. MONITORING PLATFORM EMERGENCE THROUGH SUPPORT IN COMPLEMENTARY SOFTWARE STACKS

Thus far, we have treated the support from complementors as if they are all at the same level of abstraction in the information technology architecture. It is generally accepted that the computer industry has evolved from being vertically integrated to horizontal interconnections, which can be represented using four layers (See Figure 7): services, application software, software operating systems, and hardware [Gerstner Jr. 2003; Gao and Iyer 2006]. Understanding platform emergence also calls for analyzing the pattern of support for a platform across these different layers which impact the basis of competition [West and Dedrick 2000].
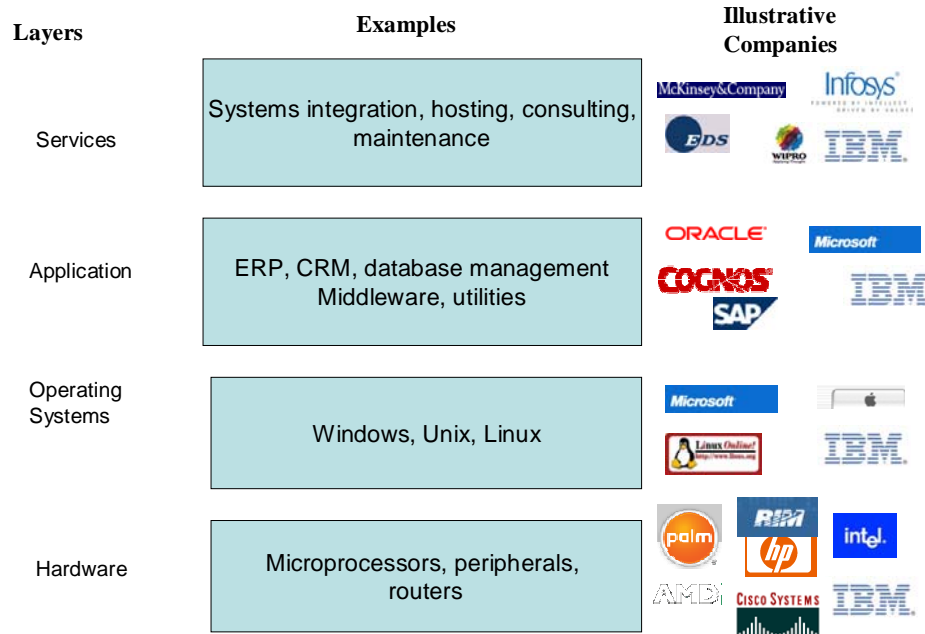
Figure 7. Software Stack

We used the layered model representation to further understand the degree of support for a platform. The visualizations show that mature platforms enjoy strong support from ISVs and complementary products that work within many layers of the stack. In Figure 8, we depict the platform ecosystem based on the layer of the industry stack [Gerstner Jr. 2003; Messerschmitt and Szyperski 2003] and the revenues generated for each layer. The picture for Windows shows that the platform enjoys widespread support for every layer of the stack. For Linux, support for the platform came from applications and systems software. Even for these, the support was symbolic as indicated by the links shaded green in color. During the same year, Unix enjoyed broader support across the stack and serious commitment from the ISVs as indicated by many links with yellow and a few with red shading. In addition, many layers of the stack are supported. However, while the prognosis for Linux is not encouraging, there are some characteristics reminiscent of how Windows emerged in the early 1990s.
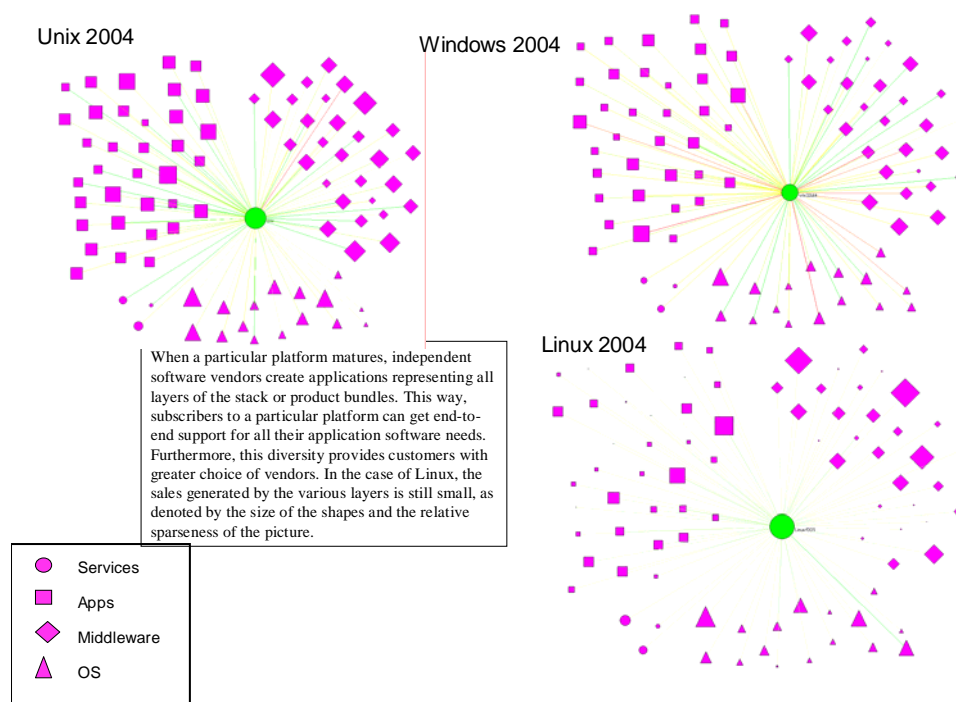
Unix 2004

Windows 2004

Linux 2004

When a particular platform matures, independent software vendors create applications representing all layers of the stack or product bundles. This way, subscribers to a particular platform can get end-to-end support for all their application software needs. Furthermore, this diversity provides customers with greater choice of vendors. In the case of Linux, the sales generated by the various layers is still small, as denoted by the size of the shapes and the relative sparseness of the picture.

● Services
■ Apps
◆ Middleware
▲ OS

Figure 8. Support for the Software Stacks in 2004

VI. IMPLICATIONS

The software industry is different from other industries because of the requirement of interdependence across different products. A platform succeeds because of the degree of network effects of complementors, which has been typically modeled analytically and/or statistically. In this paper, we have used network visualization to offer insights on the evolution of the software network to assess the growing importance of new platforms like Linux and the strong presence of incumbent platforms.

Our representation over time tells a compelling story. Mature platforms stay strong over the years. New platforms are not gaining traction, with ISVs only making symbolic commitments. If you look at the links to the Windows environment, you will notice a large set of exclusive links (represented in red), where these firms are "betting the farm" by developing products that run only on Windows. Viewing the pictures from the perspective of nodes, we notice that large companies such as IBM and SAP still hedge by supporting many platforms. Also, new entrants (shown in yellow) do not form exclusive links with any new platform: they hedge.

Four points are worth making. One: While exclusive support for Windows has been going down, close to 50 percent of all applications run on the Windows platform. Meanwhile, not many firms have made substantive commitments to a new platform, although 20 percent of the firms have signaled their intent to support Linux by having at least one product that runs on it. Support for the Unix platform, meanwhile, is showing a steady decline. Two: we have not seen droves of new software product companies entering the market to only support new platforms at the expense of other platforms. Three: most firms that embrace a new platform make a symbolic commitment at first and do not abandon their support for any other platforms. Most new entrants hedge their bets by supporting more than one platform. Four: when we delved deeper into the layers of the software stack, we found broad support across layers for the mature platforms such as Windows and Unix. Linux on the other hand, did not see broad-based support in all layers until 2004. Given that there is growth in the number of companies supporting Linux (See Figure 1), Linux may

emerge as a platform due to possible nonlinear growth since in the networked economy, the network can tip at any time [Gladwell 2000].

VII. EARLY WARNING SIGNALS OF PLATFORM EMERGENCE BASED ON THE SOFTWARE SECTOR

Increasingly, managers may find themselves in platform-based competition. Here are some of the signals that we can discern based on the software sector.

1. **Is your industry facing platform-based competition?** It is worthwhile to see if your industry has characteristics of the shift from vertical integration toward horizontal interconnected layers of functionality (Figure 7). When functionality is delivered by building on preexisting products, there is room for platform-based competition. One of the implications of the shift away from a vertically integrated, tightly controlled business architecture is the possibility of divided technical leadership. Just as Microsoft dominated with the Windows platform, Apple is striving to dominate the music industry through iTunes.

2. **Map platforms with links to key applications that support it**. In some cases, the platform providers have the power while in other situations the power rests with critical application providers. An early example of this was the relationship between hardware vendors and operating system providers. IBM created the Wintel ecosystem but Microsoft ended up appropriating most of the value. And in the videogame sector, game software companies such as Electronic Arts are jockeying to get a greater share of the value because they have critical applications demanded by end users.

3. **Track the level of porting of applications across platforms.** When software that is exclusive to a platform is ported to another, there is a significant shift in business power and leverage. SAP is trying to do this by supporting MySQL. Most of SAP's implementations support the Oracle database management system. By eroding Oracle's base within the database market, SAP is trying to stymie the progress of enterprise applications that Oracle is trying to sell to its database users.

4. **Watch for announcements from key industry players and entities to support specific platforms**. IBM's announcement of one-billion-dollar commitment to Linux in 1999, or up to $20,000 for business partners to switch from Microsoft Exchange to Lotus on Linux in 2006 has galvanized the open-source software movement. They serve as lead indicators of competitive shifts among alternative platforms. Governments could also play roles in changing the competitive characteristics of platform-based competition.

5. **Incorporate platform thinking in dynamic terms through management dashboards**. Current executive dashboards take real-time information from enterprise systems and look for anomalies or things that break a particular business rule. For example, companies may have a particular service level agreement with a vendor and the vendor will monitor delivery to conform to the agreements. As a result, dashboards are very good at tracking general process performance and in some cases companies have developed dashboards for verticals. What we have proposed here is a dashboard for strategic decision making. Such a dashboard should have the ability to render pictures like the ones depicted in this paper (Figures 3, 4, 5, and 6). These figures will help senior management from product companies to track competitors who launch products on their current platform or make substantive commitment to a competing platform. To get maximal insights from such networks, we recommend companies to make investment in decision support systems that will collect, organize and present information in the form of networks.

We hope our work stimulates companies to think in network terms using platforms and supporting complements.

VIII. CONCLUSION

In this paper we have presented the results of our analysis of the software industry. We find that firms can either be component providers that subscribe to a platform, or a platform provider that supports other component providers. Either way, firms have to be aware of the network effects. In particular, they should look for exclusive commitments, announcements by major vendors, porting solutions, and track these using dashboards. Our results apply to settings that are broader than the software industry. Today, in settings like mobile devices and automobiles, the product is dominated by software and the principles that we have identified apply. For example, when a product company plans to launch on a mobile device, they should first assess the operating platform that runs on the device. Having done that, they have to determine if other vendors are committed to it, and if so, how deep is the commitment. Based on this analysis, they have to make a strategic decision whether or not to adopt a platform.

ACKNOWLEDGEMENTS

REFERENCES

Adner, R. ([2006)]. "Match Your Innovation Strategy to Your Innovation Ecosystem." *Harvard Business Review*. 84([4)]: 98-107.

Bresnahan, T. and S. Greenstein. (1999). "Technological Competition and the Structure of the Computer Industry." *Journal of Industrial Economics.* 47(1): 1-40.

Gao, L. and B. Iyer. (2006). "Using Software Stacks to Explain Complementarities: The Case of Mergers and Acquisitions in the Software Industry." *Journal of Management Information Systems.* 23(2): 121-149.

Gawer, A. and M. Cusumano. (2002). *Platform Leadership*. Boston, Massachusetts: Harvard Business School Press.

Gawer, A. and R. Henderson. (2005). "Platform Owner Entry and Innovation in Complementary Markets: Evidence from Intel," *National Bureau of Economic Research*.

Gerstner Jr, L. V. (2003). *Who Says Elephants Can't Dance? Inside IBM's Historic Turnaround*. New York: Harper Business.

Gladwell, M. (2000). *The Tipping Point: How Little Things Can Make a Big Difference*. New York: Little Brown & Company.

Iyer, B., C.- H. Lee, et al. (2006). "Managing in a 'Small World Ecosystem': Lessons from the Software Sector." *California Management Review*. 48(3).

Messerschmitt, D. and C. Szyperski. (2003). *Software Ecosystems: Understanding an Indispensable Technology and Industry*. Cambridge, MA: The MIT Press.

Powell, W. W., D. R. White, et al. (2005). "Network Dynamics and Field Evolution: The Growth of Interorganizational Collaboration in the Life Sciences." *American Journal of Sociology*. 110(4): 1132.

Rochet, J. C. and J. Tirole. (2003). "Platform Competition in Two-Sided Markets." *Journal of the European Economic Association.* 1(4): 990-1029.

Shapiro, C. and H. R. Varian. (1999). *Information Rules: A Strategic Guide to the Network Economy.* Boston: HBS Press.

Venkatraman, N. and C. -H. Lee. (2004). "Preferential Linkage and Network Evolution: A Conceptual Model and Empirical Test in the U.S. Video Game Sector." *Academy of Management Journal.* 47(6): 876-892.

West, J. and J. Dedrick. (2000). "Innovation and Control in Standards Architectures: The Rise and Fall of Japan's PC-98." *Information Systems Research* 11(2): 197-216.

ABOUT THE AUTHORS

Bala Iyer is an associate professor in the Technology, Operations, and Information Management Division at Babson College. Professor Iyer received his Ph.D. from New York University with a minor in computer science. His research interests include exploring the role of IT architectures in delivering business capabilities, designing knowledge management systems using concepts from systems design, hypertext design and workflow management, querying complex dynamic systems, hypermedia design and development and model management systems.

Chi-Hyon Lee is an assistant professor in the management department at the School of Management, George Mason University. He received his doctorate from Boston University. His research interests include software ecosystem dynamics, alliances and partnerships, and network analytics and methods.

N. Venkatraman is the David J. McGrath Jr. Professor of Management at the Boston University School of Management. He received his doctorate from the University of Pittsburgh. His research interests are at the intersection of strategic management and information technology with an emphasis on competing through networks.

Dan Vesset is a research director for IDC's Analytics and Data Warehousing Software service. Mr. Vesset's research is currently focused on the business intelligence and analytic applications markets, which encompasses multi-dimensional analysis, end-user query and reporting, data mining and other related business intelligence tools, and supply chain and operational analytic applications.

.