Detecting Dominant Designs
Applying the theory in Network Economies

By

John Jordan

A project presented as a requirement

for the degree of

Master of Applied Science

in

Management Sciences

Waterloo, Ontario, Canada, 2001

Contents

List of Tables

List of Figures

Abstract

Dominant designs have been shown to affect the ground rules upon which competitive forces behave within an industry (Suarez & Utterback, 1995). The effects of a dominant design can be so profound as to cause the demise of firms that do not detect the dominant design early and, as a consequence, do not adopt it for their products (Christensen et al., 1998). Therefore it is in the best interests of the management of technology producing firms to develop methods to detect the emergence of a new dominant design that may disrupt their business.

Existing research has added to our understanding of the forces contributing to the success and failure of technologies and firms, especially in fast-paced technology industries (Brown & Eisenhardt, 1997; Christensen et al., 1998; Cusumano & Yoffie, 1998; Eisenhardt & Brown, 1998; Suarez & Utterback, 1995; Teece, 1986; Tushman & Anderson, 1986). However, very little has been written to develop factors, based on technological change research, to assist managers in understanding the forces influencing the emergence of a dominant design.

The paper derives a framework for the emergence of a dominant design from the literature and applies it to the Internet software industry. This is accomplished through an in-depth examination of the web server industry. Particular attention is given to factors important in network-based economies including network effects, standards, complimentary assets and technological change. A complete examination of the web server industry from its origins in the Internet through to today's web server market dominated by the products of two main software development methodologies, open-source and proprietary-executable only provides interesting insights into the application of the emergence framework. The paper concludes with an emphasis on understanding the importance of standards, network effects and complementary assets in industries where open standards play an important role in the evolution of technology.

Chapter 1

Introduction

At the race track, betting on the wrong horse can be expensive, but at least there will be a new race to bet on. Technology producing firms that bet on the wrong technology can be eliminated from competing in the race forever (Christensen, Suarez, & Utterback, 1998; Schilling, 1998; Suarez & Utterback, 1995; Teece, 1986; Tushman & Anderson, 1986). In fact, Arthur (1996) suggests technologies that win in the marketplace can do so by "chance" and he likens competition in the technology industry to a casino and Bill Gates to a "wizard of precognition". Surely there must be a way for technology producing firms to improve the odds of this high stakes gamble.

Existing research has added to our understanding of the forces contributing to the success and failure of technologies and firms, especially in fast-paced technology industries (Brown & Eisenhardt, 1997; Christensen et al., 1998; Cusumano & Yoffie, 1998; Eisenhardt & Brown, 1998; Suarez & Utterback, 1995; Teece, 1986; Tushman & Anderson, 1986). However, very little has been written to develop factors, based on technological change research, to assist managers in determining the right technological horse to bet on. Adopting the winning technological design is a critical strategic choice, particularly in industries where the rate of change is rapid and the window of opportunity to win a technology battle is short (Christensen et al., 1998).

Research focusing on technology change and industry dynamics has described the establishment of a dominant design as a "watershed event" (Christensen et al., 1998) that dramatically affects the survival of firms involved in the industry. Signifying its importance to both researchers and practitioners, a dominant design has been described to be "at the intersection of both strategy and organizational theory as well as … theory and practice." (Tushman & Murmann, 1998). Introduced by Abernathy and Utterback (Abernathy & Utterback, 1978) in the mid 1970's, a dominant design has been described as "a specific path, along an industry's design hierarchy, which establishes dominance among competing design paths" (Suarez & Utterback, 1995). For the purposes of this research, the definition of a dominant design is "the distinctive way of providing a generic service or function that has achieved and maintained the highest level of market acceptance for a significant amount of time" (Lee, O'Neal, Pruett, & Thomas, 1995).

The dominant design literature focuses on several principal themes. First, the effect that the establishment of a dominant design has on an industry in terms of the number of firms entering and exiting the industry. Typically, the emergence of a dominant design is associated with a significant decline in the number of firms competing in the industry (Christensen et al., 1998; Suarez & Utterback, 1995; Tushman & Anderson, 1986). Second, the literature documents a shift in the innovative activities of firms in the industry from product innovation to process innovation. This shift occurs when the emergence of a dominant design signals an acceptance in the market of a core set of features. Subsequently, competition shifts away from features and begins to focus on price (Abernathy & Utterback, 1978; Teece, 1986; Tushman & Anderson, 1986). Dominant designs have been shown to affect the ground rules upon which competitive forces behave within an industry (Suarez & Utterback, 1995). The effects of a dominant design can be so profound as to cause the demise of firms that do not detect the dominant design early and, as a consequence, do not adopt it for their products (Christensen et al., 1998). Therefore it is in the best interests of the management of technology producing firms to develop methods to detect the emergence of a new dominant design that may disrupt their business.

Typically, technology producing firms and consumers focus on technological merits as the prime indicator of the potential success of a new technology (Arthur, 1987). However, current research points to important non-technological factors as having a greater influence on the establishment of a dominant design. These include:

1. possession of collateral assets;
2. industry regulation and government intervention;
3. strategic maneuvering at the firm level;
4. existence of bandwagon effects or network externalities.

Taken from (Suarez & Utterback, 1995)

Collateral or complementary assets, such as marketing, sales, and support are usually needed to successfully commercialize an innovation (Teece, 1986). For example, the availability of gasoline stations for refueling complemented the petroleum based internal combustion engine and thus assisted that particular engine design to become the dominant one (Henderson, 1998). Industry regulation or government intervention can also help establish a dominant design. This was the case in the selection of the RCA colour television standard (Suarez & Utterback, 1995). The classic battle of Beta vs. VHS is an example of strategic maneuvering. The liberal licencing of the VHS format by the Victor Company of Japan (JVC) overcame the highly restricted licencing approach used by Sony in its promotion of the Beta format (Cusumano, Mylonadis, & Rosenbloom, 1992). Finally, the existence of network externalities can significantly affect the adoption of a dominant design for technologies that are more valuable to individual adopters as the network of adopters grows (Katz & Shapiro, 1985; Katz & Shapiro, 1994; Suarez & Utterback, 1995). Lee et al (1995) develop a model describing the factors, including those just described, influencing the emergence of a dominant design.

Dominant Designs in Internet Software

Studying the new kinds of technology provides opportunities to use new types of data and examine new variables and theories (Van de Ven & Rogers, 1988). Suarez and Utterback (Suarez & Utterback, 1995) call for the examination of contemporary industries to further our understanding of dominant designs. The Internet software industry presents a contemporary example of the struggle for dominance of two very different approaches to the development of software based technology. The two competing designs are open-source software and proprietary executable-only software (PEOS).

PEOS is the most widely know software development and distribution design. Typically commercial software is developed by firms and distributed to end-users by licencing an executable-only program in a format that is only readable by a computer. The end-user is not able to view or modify the underlying source code. Microsoft is the most successful firm to employ this design of software development and distribution.

Open-source software is typically developed by a group of individuals, using the Internet as a communications mechanism, in the public view. Anyone can access the source code of the software and use the software at no cost. There are some firms who have attempted to use this process for commercial software such as Netscape with their web browser product.

While open-source software and proprietary executable-only software products provide similar functionality, they provide this functionality in quite different ways. As evident from the brief history in Chapter 4, open-source software and proprietary executable-only software development and distribution methods have followed different design paths. The roots of open-source software are closely tied to the development of the Internet. Proprietary executable-only software is closely tied to the development of personal computers. The meeting of these two different paths, the Internet and personal computing, has resulted in an interesting and complex interaction of two very different approaches to software development and distribution.

Technology evolves rapidly in the Internet industry (Cusumano & Yoffie, 1998; Iansiti & MacCormack, 1997). Open-source software, an Internet-based phenomenon, provides an excellent opportunity to explore the dynamics of a fast-paced and intensely competitive industry as it unfolds. Some open-source software projects promise to provide excellent sources of data to explore the evolution of a new technology. One such project is web server software. As previously mentioned, the open-source web server software product, Apache, is a significant player in the web server market. It is in the same market space as proprietary executable-only alternatives such as Microsoft Internet Information Server (IIS) and Netscape Web Servers. Extensive data pertaining to the adoption of web servers on the Internet can be obtained from several sources.

The combination of the rate of change engendered in Internet-based technologies, the availability of useful data, and the transparency of open-source software provides an opportunity to investigate some of the factors that are purported to illustrate the emergence of a dominant design. Suarez and Utterback (1995) note the need for more research on the establishment of dominant designs in contemporary industries or technologies. This opportunity is unique in that previous studies investigating the emergence of a dominant design have relied on *ex post* analysis of industry and product level data (Christensen et al., 1998; Suarez & Utterback, 1995; Tushman & Anderson, 1986). The premise of this research is that open-source software could provide an opportunity to observe the potential emergence of a dominant design as it happens. This could lead to useful insights on how to detect the emergence of a dominant design and thus provide firms with the opportunity to adjust strategies and improve survival rates.

Using the web server market as a case study, this research will investigate factors that may assist in the early detection of a dominant design in the Internet software industry. The research will compare the open-source software development and distribution approach to the more traditional proprietary executable-only development and distribution approach. These two approaches to software development are considered to be the two significant competing "design paths" in Internet software development. A combination of qualitative and quantitative techniques will be applied to the available data in an effort to isolate one or more of these factors. This will hopefully lead to the development of techniques for both researchers and practitioners in the continued effort to understand and react to technological change in technology based industries where network economic factors such as standards and network externalities are present

Document Structure

Chapter 2 provides and in-depth review of the literature introducing, within the context of the Management of Technology field of study, the concepts of technology adoption and diffusion, dominant designs, standards and network externalities. Chapter 3 draws on these concepts to develop a framework for detect or attempting to establish a dominant design in a network based industry. This framework is then applied, in Chapter 4, to the web server software market, which provides an excellent example of competing designs in the Internet software industry. To place this study in the context of the industry, a brief history of the Internet, its adoption by commercial businesses, the WWW and open-source software is given. Most people are very familiar with the history of proprietary executable-only software or can refer to other literature (Cottrell, 1997) to learn of its history. An in-depth look into the web server software market focusing on factors identified in the literature is presented to elucidate the framework developed in Chapter 3. The goal is to discover if there are appropriate methods for determining if there is an emerging dominant design in software development and distribution for Internet software. Based on the analysis conducted, conclusions are offered and future research opportunities described.

Chapter 2

Literature Review

Management of Technology is a discipline concerned with understanding innovation. Other disciplines that conduct research in the innovation field include economics, sociology and psychology (Gopalakrishnan & Damanpour, 1997). Management of Technology researchers are primarily concerned with "the process of generating new technology or improving upon existing technology" (Gopalakrishnan & Damanpour, 1997). Consistent with the focus of contextual technologists, the focus of this literature review is on technological change at the industry level (Gopalakrishnan & Damanpour, 1997).

Central to the study of technological change at the industry level is the concept of a **dominant design**. Consequently, this literature review examines research concerned with the concept of a dominant design and several related areas of study including diffusion of innovations, standards and network externalities.

This literature review places the study of dominant designs within the context of Management of Technology and Innovation research. This provides the context for examining research focusing on the emergence of dominant designs. Research exploring the factors influencing the emergence of a dominant design is a new and poorly understood area of research despite its obvious importance to industry (Lee et al., 1995). The limited research on the emergence of a dominant design is reviewed and several factors are identified for further exploration. The latter part of this paper reviews the research on standards and network externalities as they relate to the study of dominant designs.

Management of Technology

Rosenbloom and Cusumano (1987) define the strategic management of technology (MOT[1]) field as being concerned with the issue of how managers translate technological capability into competitive advantage. It is a field of study that combines the previously independently researched fields of business strategy and technological innovation (Rosenbloom & Cusumano, 1987). This definition has been more explicitly defined as the field of study that "links engineering, science, marketing, operations, human resources, and other management disciplines to formulate strategy, develop technological capabilities, and use them to achieve strategic objectives" (Husain & Sushil, 1997). Both definitions are clearly concerned with how a firm creates and deploys technological capabilities to foster competitive advantage. This is clearly an important management issue given the widespread use of technology in the economy and its far-reaching impact on the way firms operate in today's business environment.

Innovation Research

Innovation is widely regarded as an important capacity for firms to develop if they hope to remain competitive in today's rapidly changing marketplace (Brown & Eisenhardt, 1995;

---

[1] Similar terms refer to this area of study. These include "Management of Technological Innovation", "Strategic Management of Technology", "Management of Innovation and New Technology", "Technology Management," etc.

Brown & Eisenhardt, 1997; Roberts, 1998). Innovation research can be divided into categories, each consisting of several stages. From an organizational perspective, researchers categorize innovations as being either adopted by an organization or generated by an organization (Gopalakrishnan & Damanpour, 1997). Research into the adoption of innovation is concerned with determining the conditions, characteristics or factors that influence a particular adoption unit (individuals or organizations) to adopt an innovation (Rogers, 1995; Zaltman, Duncan, & Holbek, 1973). The generation of innovation is understood to have five stages: idea generation, project definition, problem-solving, design and development (Gopalakrishnan & Damanpour, 1997). The final two stages are concerned with the development and commercial exploitation of the innovation (Gopalakrishnan & Damanpour, 1997). Researchers have observed that the establishment of a technology design as the dominant one in a product class can provide an organization with a high degree of market power (Cusumano et al., 1992; Gabel, 1991; Tushman & Anderson, 1986). Consequently, researchers also focus on how an innovation is diffused throughout a population of organizations (Gopalakrishnan & Damanpour, 1997).

Technology Diffusion

In the fourth edition of his seminal work, *Diffusion of Innovations*, Rogers defines adoption as “a decision to make full use of an innovation as the best course of action available”. (Rogers, 1995, p21). Rogers’s approach to the study of innovation adoption, or the diffusion of innovations, is based on a social communications perspective. This perspective assigns five characteristics to the innovation. These characteristics are: relative advantage, compatibility, complexity, trialability, and observability. The theory describes an innovation-decision process where an individual, or other unit of adoption, assesses the innovation in terms of the characteristics described above and makes a decision to adopt or reject the innovation. The social aspect of this process recognizes that this innovation-decision occurs within the context of the adopting unit's particular social system. The formal definition for diffusion is "the process by which an innovation is communicated through certain channels over time among the members of a social system" (Rogers, 1995).

The major focus of diffusion of innovation research is to understand the diffusion process using the individual as the unit of analysis. However, it has been argued that organizations differ from individuals in the criteria they use and the process they follow to reach the innovation-decision (LaRose & Hoag, 1996; Rogers, 1995; Zaltman et al., 1973). As a result of these differences, research into the diffusion of innovations, in the organizational context, has developed into several different streams. These include:

- **The initiation phase of adopting an innovation within an organization.** According to Rogers (1995) this includes the "information gathering, conceptualizing and organizing leading up to the decision to adopt." Research in this tradition includes: (Ancona & Caldwell, 1990; Ancona & Caldwell, 1992; Leonard-Barton, 1994; von Hippel, 1987). The unit of adoption in this research is typically the individual within an organization.
- **The implementation of innovations within an organization.** (Note: the concept of implementation is used to describe the outcome of the innovation-decision process rather than the concept of adoption, which is more oriented towards the individual unit of analysis, as suggested in Zaltman (1973).) Implementation includes the decisions, events and actions taken to put an innovation into use (Rogers, 1995). Researchers examining this aspect of the innovation process include: (Agarwal & Prasad, 1997; Agarwal & Prasad, 1998; Agarwal, Tanniru, & Wilemon, 1997;

Astebro, 1995; Cooper & Zmud, 1990). This area is sometimes defined as user acceptance in the study of information technology adoption. The unit of adoption is the individual within an organization.

- A further area of study is **the adoption of innovations by organizations.** These studies consider the firm to be the unit of adoption and attempt to describe the diffusion of an innovation among firms, usually in a specific geographic area or industry sector. Examples of these studies include: (Chau & Tam, 1997; Dos Santos & Pfeffers, 1998; LaRose & Hoag, 1996; Lind, Zmud, & Fischer, 1989; Teo & Tan, 1998).

Diffusion of innovation studies typically focus on a single innovation. Characteristics of the innovation and/or the unit of adoption are often the variables examined. This research is valuable for determining, within the context of a social system, the characteristics of an innovation, which encourage its adoption. However, it does not help us understand the forces at play when there are multiple competing innovations. Nor does it help firms understand the conditions under which their innovation could become the dominant innovation in an industry. The next section, Dominant Designs, suggests just such an approach to understanding industry dynamics from a technology change point of view.

Dominant Designs

The concept of a dominant design stems from research examining patterns in industrial innovation and technological change conducted in the 1970’s at the Massachusetts Institute of Technology (MIT) (Abernathy & Utterback, 1978). Abernathy and Utterback (1978) introduce a model of technological development describing the relationships between the innovative focus of firms in an industry, the stage of an industry in terms of the amount of change in its products, and firm strategies. They suggest industries can be broadly described as being in one of three phases.

Industries typically begin in a fluid phase, characterized by a high degree of uncertainty in product functionality. Competition focuses on features and requires flexible production mechanisms, which may be inefficient. Innovation is more likely to be radical. As the features of the product stabilize, the industry moves into a transitional phase. In this phase, innovative efforts begin to focus on process rather than product. At least one product begins to gain market share and production facilities begin to grow now that more specific mechanisms for production are available. Finally, the industry moves into the specific phase. In this phase, competition focuses on reducing cost and production facilities become highly efficient, capital-intensive and rigid in order to exploit economies of scale. Figure 1 depicts the relationship between product and process innovation and the phases of industry evolution as proposed by Abernathy and Utterback (1978).

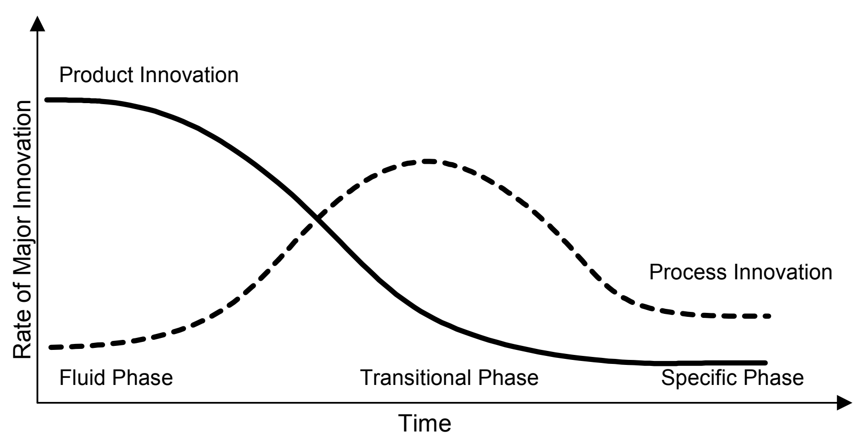

Figure 1: Abernathy – Utterback technological development model. (Abernathy & Utterback, 1978)

The event that triggers a transition from the fluid phase to the specific phase in this model is the establishment of a dominant design (Abernathy & Clark, 1985, Abernathy, 1978 #80; Smit & Pistorius, 1998). Dominant designs can also influence the survival rate of firms (Baum, Korn, & Kotha, 1995; Christensen et al., 1998; Suarez & Utterback, 1995) and can affect product class and firm performance (Hamel & Prahalad, 1994; Henderson & Clark, 1990).

In earlier research, technological change was typically considered to be an exogenous variable or taken to be constant (Romer, 1998). However, introduction of the dominant design concept has focused the attention of researchers on how significant the effects of technological change can be on firm evolution, structure and design. Today, firms ignore technological change at their peril (Christensen et al., 1998; Suarez & Utterback, 1995). Various disciplines have researched the effects of a dominant design (Lee et al., 1995) including technology and innovation research (Abernathy & Clark, 1985; Suarez & Utterback, 1995; Teece, 1986), organizational research (Tushman & Rosenkoph, 1992) and product standardization in economics (Farrell & Saloner, 1987; Gabel, 1987; Hergert, 1987; Katz & Shapiro, 1985).

As a result of the wide range of research into dominant designs there is confusion over the definition, the unit of analysis and causal factors (Henderson, 1998; Tushman & Murmann, 1998). For example, the concept of a dominant design is similar to a standard. At times, the terms dominant design and product standard (or dominant standard) are used interchangeably (Afuah, 1998; Schilling, 1998). As stated in Suarez and Utterback (1995) and Henderson (1998), the notion of a dominant design is related to the notion of a standard. However, the notion of a dominant design is intended to encompass a broader scope focusing on the design architecture of a technology at an industry level (Suarez & Utterback, 1995) which may include many standards (Henderson, 1998). The literature on standards is reviewed in a subsequent section of this paper.

A further issue to clarify is the level of analysis at which the dominant design is being examined. Smit and Pistorius (1998) suggest two different levels at which a dominant design may be examined. A dominant design may be studied at an industry level where several designs are competing for dominance. In this case, there are often both an incumbent technology and a challenger, and the issue is one of substitution. For example, the eventual replacement of sail ships with steamships. An alternative level of analysis occurs at the product class level. At this level, uncertainty occurs as the dominant set of features and functionality are determined over time. This is particularly important where the product is a complex assembled product and various subsystems of the product may also be in periods of ferment and thus affect the higher level system designs (Tushman & Murmann, 1998). An example of product class design competition is the personal computer industry where various hardware and operating system architectures compete for dominance.

Various definitions for a dominant design exist in the literature. The following definition of a dominant design, taken from Lee et al (1995), provides a broad foundation to build on:

> *A dominant design is "the distinctive way of providing a generic service or function that has achieved and maintained the highest level of market acceptance for a significant amount of time." (Lee et al., 1995)*

Other definitions of a dominant design tend to be narrower in scope thus limiting their applicability to all industries. For example, "A dominant design is a specific path, along an industry design hierarchy, which establishes dominance among competing design paths" (Suarez & Utterback, 1995). Suarez's definition is derived from research focusing on products assembled for the mass market and thus loses some of the breadth of the Lee definition.

Following the introduction of the concept of a dominant design by Abernathy and Utterback (1978), its scope has been expanded to include the social and physical infrastructure that grows up around the dominant design (Henderson, 1998). The definition put forward by Lee et al (1995) encompasses this broader view of a dominant design.

Emergence of a Dominant Design

Given the importance of the dominant design in influencing the structure of an industry and the effect it can have on survival, firms have a strategic interest in ensuring that their offerings conform to the dominant design. However, the processes by which firms can develop strategies to detect an emerging dominant design are not well understood. Consequently, the process of how a dominant design emerges is often considered to be a “black box” involving a wide range of factors that are difficult to identify and measure (Lee et al., 1995; Suarez & Utterback, 1995). Often, firms consider the technological merit of the innovation in question to be the most important factor in terms of establishing a dominant design. This point of view is not substantiated by the literature or actual observation (Arthur, 1987). It is possible for technologically superior design alternatives to fail to be established as the dominant design (David, 1985).

Suarez and Utterback (1995) note that the emergence of a dominant design may be influenced by non-technological factors. These include:

1. *possession of collateral assets (equivalent to complementary assets (Teece, 1986))*
2. *industry regulation and government intervention (standards setting)*
3. *strategic maneuvering at the firm level (time pacing, standards strategies)*
4. *existence of bandwagon effects or network externalities (net effects)*

In 1995 Lee et al. (1995) published a paper proposing a framework for understanding the process by which a design may emerge as dominant. This framework synthesizes previous research including Teece (1986) and Suarez and Utterback (1995). Again, this framework stresses the importance of the non-technological factors in the process of establishing a dominant design.

The Lee et al. framework, as depicted in Figure 2, specifies a number of factors that may influence the emergence of a dominant design. However, in applying the framework to a particular industry or product class, some of the factors may or may not be relevant. Some customization of the framework is required as the relevant forces may vary widely from case to case (Lee et al., 1995).

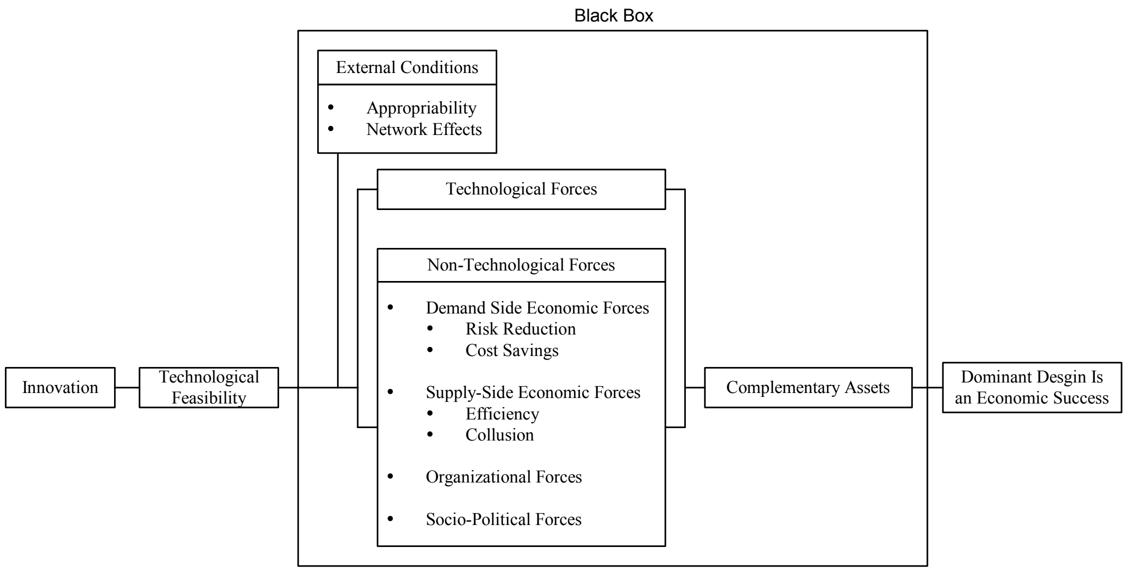

Figure 2: Framework for the Emergence of a Dominant Design (Lee et al., 1995)

Based on the Lee et al. framework, Smit and Pistorius (1998) explore the emergence of a dominant design in electronic ignition systems in the South African mining industry. Smit and Pistorius provide a more concrete list of factors in their extension of the Lee et al. framework. These factors are listed in Table 1.

Factor	Indicator
Technological factors	Technological feasibility Technical capability Rate of technological change Momentum of the established practice
Market factors	Marketing Strong brand name Distribution network

	Network Externalities First-mover advantages
Economic factors	Demand-side factors Supply-side factors
Social and behavioural factors	Ease of use Resistance to change Communication between producer and user

Factor	Indicator
Standards and regulations	Current standards and industry regulation Government regulations Industry standards and committees Safety, health, and environmental concerns
Political factors	Sociopolitical factors
Organizational factors	Interactive learning Powerful user Powerful producer Alliances
Strategic factors	Complementary assets Patents Appropriability

Table 1: Factors influencing the Emergence of a Dominant Design, (Smit & Pistorius, 1998)

The factors listed in the two frameworks above are not necessarily applicable to all scenarios. Consequently, it is necessary to develop a framework that is applicable to the industry under study. Such a framework should include factors from the Lee et al. framework and the Smit and Pistorius extension combined with other factors derived from the literature relevant to the industry of concern. Certain factors will play a more critical role in influencing the emergence of a dominant design, depending on the technology in question.

The following two sections of this paper examine the research on two such factors, standards and network externalities, in greater detail. These factors play a particularly important role in industries where the installed base, networks and compatibility are critical to the emergence of a dominant design. Examples of such industries are typically information technology oriented, for example: telephony, fax, email, computer operating systems and the Internet (Hergert, 1987). However, other industries such as: railways, typewriters and stereophonics are also influenced by these factors. In all these industries, dominant designs have been established and are strongly based on factors such as compatibility standards, the size of the installed base of users and expectations of potential adopters on the future size of the network (Christensen et al., 1998; Cottrell, 1997; Shapiro & Varian, 1998)

Standards

The literature on standards is broad and diverse, ranging from economic analysis to competitive strategy formulation. For the purposes of this research, the standards literature will be reviewed with respect to competitive strategy and its effect on adoption decisions in the information technology industry.

Standardization is a coordination process which results in the production of goods that are interchangeable or compatible (Farrell & Saloner, 1987). These goods can then be mixed and matched in order to create a system. Standards can arise from this coordination process or standards can be agreed upon or set by regulatory agencies, governments or industry bodies in order to develop compatible systems (Farrell & Saloner, 1987).

According to Afuah (1998) there are two kinds of standards: “interchangeable standards” (analogous to compatibility standards) and “product standards.” However, the concept of a “product standard” is not well defined and may well conflict with the concept of dominant designs (Henderson, 1998; Suarez & Utterback, 1995) as noted earlier. The type of standard that is important to the information technology industry is a compatibility standard (Besen & Farrell, 1994). A summary of compatibility standards is presented in Farrell and Saloner (1987), some of the leading researchers in the economic and strategic effects of standards. These compatibility standards are:

- Physical compatibility: objects physically designed to fit together such as camera bodies and lenses, nuts and bolts.
- Communications compatibility: devices that are able to communicate with each other such as telephones, fax machines, computer networks.
- Compatibility by convention: coordinated efforts that result in some benefit but not embodied in a physical object: standard time, currency, measurement.

According to Farrell and Saloner (1986) three main benefits of compatibility standards are:

- Interchangeability of complementary products
- Ease of communication (people to people, people to machines, machine to machine)
- Cost savings (interchangeability of parts allows for mass manufacturing)

Clearly the Internet[2] is an excellent real world example of all these benefits.

Standards are essential to the functioning of the Internet and continue to foster the creation of billions of dollars worth of innovation and economic growth in the information technology industry (Farrell & Saloner, 1987; Hergert, 1987). A significant amount of research describes the varying effects and important strategic role of standards in the information technology industry, particularly the software industry (Besen & Farrell, 1994; Church & Gandal, 1992; Farrell & Saloner, 1986; Gandal, 1995; Katz & Shapiro, 1985). This research has been presented in a more accessible format in books aimed at practitioners such as *Information*

---

[2] See Chapter 4 for a brief history of the Internet and how the Internet standardization process has evolved.

Rules (Shapiro & Varian, 1998), and in other books such as *Standards, Strategy and Policy* (Grindley, 1995) and *Competitive Strategies for Product Standards* (Gabel, 1991).

Standards and standards-making organizations have existed for some time. For example, during the construction of the railroads in the 19th century, standards were an issue in terms of the gauge of the rail tracks (Farrell & Saloner, 1987). The standards making process has had a significant impact both in terms of the new standards that it has created and a new model for standard setting (Rutkowski, 1995).

Historically most standards were set via two major standards setting approaches (Farrell & Saloner, 1987; Rutkowski, 1995; Shapiro & Varian, 1998, p237). At some point these mechanisms were given the labels de jure, meaning "legitimate, just or imposed as a matter of law" (Rutkowski, 1995) and de facto, meaning "illegitimate, condoned or accepted for practical purposes" (Grindley, 1995, p25; Rutkowski, 1995). De jure standards are considered to be set by formal, often governmental organizations such as ANSI, CSA, the Organizations for International Standards (ISO), or the ITU, an international, intergovernmental telecommunications standards arm of the United Nations. Firms or industry consortia are normally considered to set de facto standards (Farrell & Saloner, 1992). Rutkowski's contention (1995) is that no standards making body, whether voluntary, governmental or an independent organization can claim its standards as legitimate, legally binding or superior. In fact, the de jure standards-setting bodies such as the ISO and the ITU may not be so "legitimate". The ISO is a for-profit organization and the ITU has no legally binding authority to enforce its standards (Rutkowski, 1995).

Formal standards setting processes, often managed by international bodies, are slow and expensive. De facto standards are regarded as inefficient, and re-enforce the power of a single or small number of firms that have created the standard (Rutkowski, 1995). In contrast, the Internet standards process is open, flexible, efficient and timely. It is a process that resembles the time-honoured academic review process for scientific research. Rutkowski, a member of the Internet Society, has proposed the following explanatory factors for the success of the Internet standards process:

- Individual participation
- Direct open participation by experts and innovators
- Output consists of demonstrated working standards
- Emphasis on meeting real user needs
- A well-managed development process
- Minimal institutional ossification
- Standards approved via a robust expert review process
- Standards and related material are instantly and universally accessible and browseable
- Activities are network based
- Creating the right culture

See (Rutkowski, 1995) for a more detailed description of the factors.

Given the importance of standards-setting to competitive strategy in the information technology industry, a number of different frameworks have been proposed to assist in developing a standards based competitive strategy. Gabel (1991) has proposed a 2 x 2 matrix (Figure 3) to describe competitive strategies when dealing with the creation of standards and the positioning of the standard to maximize its utility.

	Restricted Access	Open Access
Proprietary Standard	1 Maximize the standard's rent	2 Maximize the standard's market share
Public Domain Standard	3	4 Non-competitive standard

Figure 3: Types and strategies of standards (Gabel, 1991, p12)

Gabel's matrix deals with the creation of a standard and how to gain benefit from it from the producer's point of view. In this context (note quadrant 3) it is not logical to have a public domain standard with restricted access. In the context of Internet standards, however, this may be a possible outcome. While all Internet standards are publicly accessible, a great deal of specialized knowledge is needed to understand these highly technical specifications. Similarly, if one were interested in participating in the standards setting process for the Internet this same specialized knowledge would be necessary. It also appears that Internet standards setting organizations like the W3C (responsible for the HTML, HTTP, and XML standards among others) are highly influenced by a select group of individuals and companies therefore, having industry contacts appears to be another barrier to participation.

An approach similar to Gabel's matrix relating competitive strategy to standards is given by Grindley (1995) as shown in Figure 4.

		Access	
		Proprietary	Open
Leadership	LEAD (Develop)	1 Sponsor/Defend	2 "Give Away"
	FOLLOW (Adopt)	3 License in	4 Clone

Figure 4: Strategic positioning decisions (Grindley, 1995, p30)

Grindley's matrix is very similar to Gabel's, especially in quadrants 1 and 2 where Sponsor/Defend is analogous to maximizing rent and "Give Away" is analogous to maximizing market share. Using these matrices, producers can begin to understand what type of strategy may lead to a greater acceptance of the standard they are trying to develop or promote. For example, Sony chose the "maximize the standard's rent" (quadrant 1 of Figure 3) strategy while JVC chose the "maximize the standard's market share" (quadrant 2 of Figure 3) in VCR beta vs. VHS standards battle (Cusumano et al., 1992). Applying the framework given in Figure 4 we see that JVC's strategy treated VHS much more like an open standard and thus followed a "give away" strategy. Sony considered their beta standard to be proprietary and followed the "sponsor/defend" strategy. In the end, treating the VHS standard as open and maximizing it market share proved to be the winning strategy. Sony may have realized this as they were not the leaders/creators of video recording technology and thus should not have pursued a strategy that positioned Sony as a leader/developer (Cusumano et al., 1992).

The standards literature provides insight into the standards-setting process and proposes a number of generic strategies firms' may employ when competing in markets where standards are important. A further factor of importance which arises from the existence of compatibility standards and can play an important role in the success of a firm in establishing its design as dominant is the network externality (Church & Gandal, 1992; Hergert, 1987; Katz & Shapiro, 1985). Network externalities are an important factor in the establishment of dominant designs in the information technology industry (Brynjolfsson & Kemerer, 1996; Church & Gandal, 1992; Gandal, 1995; Hergert, 1987; Sheremata, 1997). The next section introduces the concept of network externalities within the context of the information technology industry.

Network Externalities

Network externalities occur when the benefits that arise from the consumption of a good increase with the number of users of the good (Katz & Shapiro, 1985). These consumption externalities can occur in the following situations (Katz & Shapiro, 1985):

- Directly through physical compatibility such as in telephone, fax or email networks. In this case the benefit to the user is directly related to the total number of users (Gandal, 1995).
- Indirectly, when a hardware-software dependency exists and the variety of available software will increase with the consumption of the supporting hardware. For example, in video games, video players or audio equipment. In this case, the benefit to the user arises from the variety of complementary products available as a result of compatibility standards (Gandal, 1995).
- Or, in the case of a durable good, when the quality of the service network may depend on the number of units sold. For example, in automobile service and parts.

Since the Katz and Shapiro article published in 1985 there have been many articles written examining various aspects of network externalities and their effect on pricing, strategy and market efficiency. This paper reviews only those factors relevant to the establishment of dominant designs in the information technology industry.

A key characteristic of a market in which network externalities arise, is the existence of compatibility standards (Church & Gandal, 1992; Hergert, 1987; Katz & Shapiro, 1985). The computer software industry is particularly subject to network effects (Brynjolfsson & Kemerer, 1996; Church & Gandal, 1992; Gandal, 1995; Hergert, 1987; Sheremata, 1997). For example, Brynjolfsson (1996) notes that the purchase price of software reflects only a small portion of the total consumer expenditure. The learning and conversion costs associated with switching software packages account for a majority of the price of software and help create strong network effects (Farrell & Saloner, 1986).

One might wonder why, in the face of network externalities, there exists variety of product designs in industries such as the computer software industry. Katz and Shapiro (Katz & Shapiro, 1985) examined the question whether or not firms develop the proper incentives to produce compatible goods and services. They found that firms with good reputations or large existing networks tend to ignore compatibility. Firms with small networks or weak reputations tend to favour compatibility. Therefore, we have a partial explanation as to why different “networks” or groups of computer systems exist. Large firms have the market power to ignore compatibility when introducing new systems, as IBM did with the PS/2 or Apple continues to do with new MacOS versions. Small firms may also disregard compatibility when introducing specialized systems. However, they are usually targeting niche markets and may include compatibility with existing systems to reduce switching costs. (e.g. the BeOS operating system which includes Windows compatibility).

Katz and Shapiro (1992) suggested further that installed bases (networks) are critical objects of strategic rivalry in industries with rapid technological development processes and buyers concerned with compatibility issues. They showed that firms introducing new technology are biased against compatibility. (i.e. they try to lock buyers into their installed base). They also find that firms with new incompatible technology may introduce their product sooner than desirable (e.g. beta software).

From the consumer perspective, the adoption of a certain system (i.e. owning a particular brand of personal computer) will be partially dependant on the number of other consumers purchasing similar systems (Hergert, 1987). This condition arises from the consumers’ perception that the size of the network affects the number and variety of peripherals and software available. Standards act as insurance, where utility of the system is dependant on the availability of compatible goods. Also, the more the base system is used the more likely compatible goods will continue to be available as needs change (Hergert, 1987).

In a critical review of this literature, Liebowitz and Margolis (1994) suggest that while network effects may be widespread, network externalities are likely very uncommon. The thrust of their argument is based on the distinction between "network effects" and "network externalities". Network effects, according to Leibowitz and Margolis, occurs when "the net value of an action (consuming a good, subscribing to a telephone service) is affected by the number of agents taking equivalent actions" (Liebowitz & Margolis, 1994). A "network externality" is reserved for "a specific kind of network effect in which the equilibrium exhibits unexploited gains from trade regarding network participation" (Liebowitz & Margolis, 1994). In other words, the market is not fully efficient when there is profit that is not realizable by the producers of a good. Their arguments are based on a very strict interpretation of the economic models used to develop the concept of network externalities. For the purposes of understanding the strategic effects of this phenomenon on the information technology industry the concept of "network effects" is sufficient as we are not concerned with the efficiency of the market.

Summary

The establishment of a dominant design is a watershed event in an industry (Christensen et al., 1998). Both consumers and producers are greatly affected by the establishment of a dominant design and therefore have an interest in developing methods for making an early determination of which of many competing designs will become the dominant one.

This literature review places the study of the dominant design clearly within the field of Management of Technology. It then examines the links between innovation, diffusion of innovations, and the concept of a dominant design. These links show a connection between the study of technology as a strategic asset contributing to the success or failure of a firm, and earlier research on innovation in the firm and the diffusion of innovations from both an individual and organizational perspective. The concept of a dominant design assists in explaining industry-wide evolution of technological innovation and change. However, little research has been done on how firms can either detect or influence the emergence of a dominant design. The emergence of a dominant design is clearly critical to a firm's technology strategy and ultimately contributes to its success or failure.

This review introduces several frameworks from the literature, exploring the "black box" of how a dominant design emerges. Several constructs from these frameworks are expanded upon and given a context in the information technology industry. From this review it is apparent that the establishment of compatibility standards and the existence of network externalities or network effects can greatly influence the establishment of a dominant design in an information-technology-based industry. Firms, through sponsorship, reduced prices, provision of complementary assets or other methods reviewed in this paper can influence the adoption of their technologies and thus improve their chances of success. The next chapter and subsequent case study will focus on a subset of the factors presented the frameworks and examine, in detail, an industry where a struggle for design dominance is on-going.

Chapter 3

Detecting a Dominant Design

This chapter presents a framework for detecting dominant designs in information technology industries, particularly ones affected by network economic factors. As indicated in Lee et al. (1995) the general framework that is presented in their research requires customization to fit the particular industry under examination. The framework in Figure 5 merges the Smit and Pistorius extensions into the Lee et al framework. The merged framework has also been supplemented with factors from the literature relevant to technology adoption and diffusion.

This study will concentrate on factors within the “black box” of the emergence framework. Since the technology being studied is widely used we can assume that the adoption factors have been satisfied. To understand the impact of various factors influencing the emergence of a dominant design in the Internet software industry, this chapter will focus on the factors listed in Table 2.

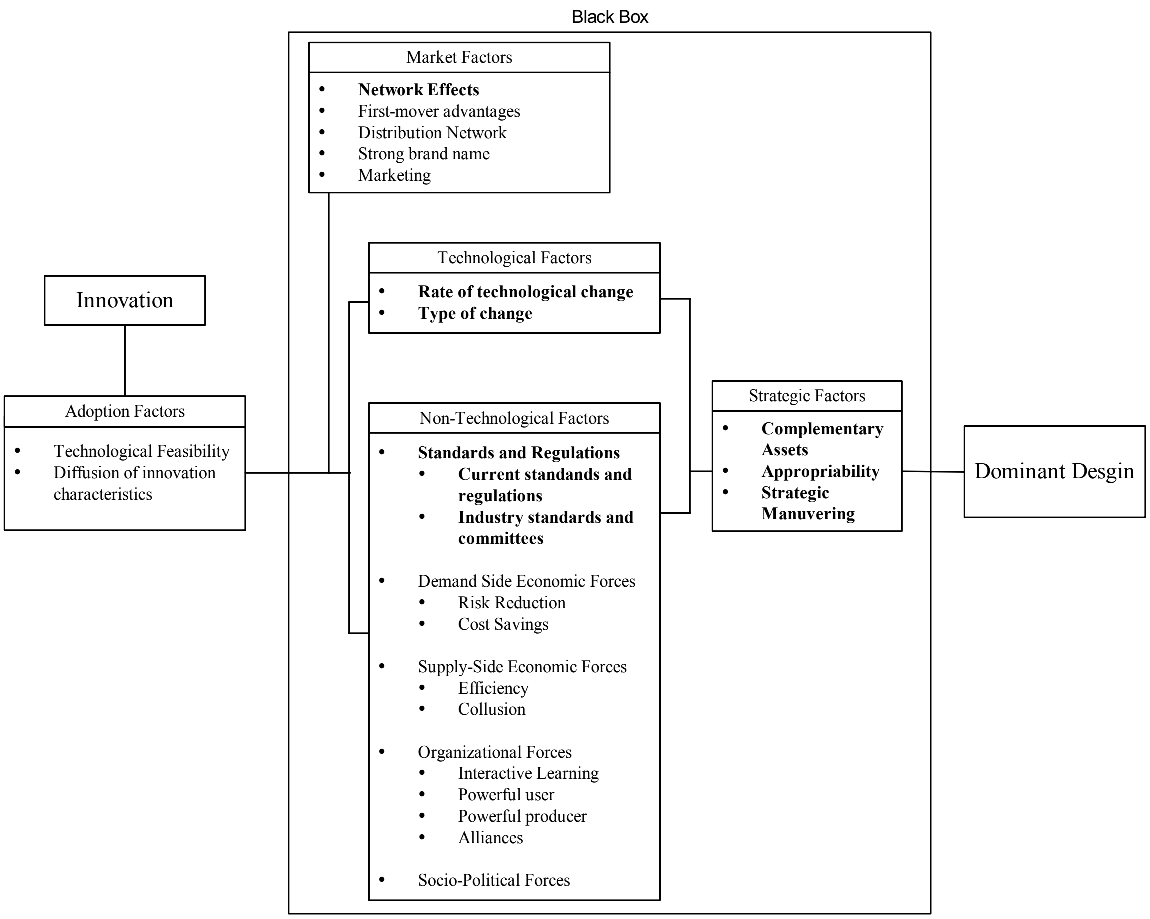

Figure 5: Modified Framework for detecting a dominant design

Market Factors
Network effects

Technological Factors
Rate of technological change
Type of change

Non-Technological Factors
Standards and regulations

Strategic Factors
Complementary assets
Appropriability
Strategic maneuvering

Table 2: Factors for Internet software industry

The list of factors presented in Table 2 is not an exhaustive the list of all possible dominant design influencing factors. However, the list is based on research highlighting these factors as important success factors in information technology based industries (Christensen et al., 1998; Cusumano et al., 1992; Gabel, 1991; Shapiro & Varian, 1998). Several of the factors have been explored in detail in the literature review including standards and network effects. Those factors that have not been presented in detail will have a brief description provided.

Factor Descriptions

This section briefly describes each of the factors listed in Table 2. These descriptions are not intended to provide an exhaustive explanation of the factor but to provide a link from the research literature to the case study presented in this paper.

Market Factors

Network effect or externalities can be either direct or indirect. Direct network effects arise as a result of the number of users on the quality of the product (Katz & Shapiro, 1985). In the web server and browser market, direct network effects stem from the ability of the browsers and servers to interoperate. In this case, the ability to interoperate is clearly linked to standards such as HTTP and HTML. If the web servers and browsers remain standards based and are able to interoperate then the addition of each new web server benefits all browser users. This is a direct externality as benefits accrue to the browser users that they do not pay for. The existence of these direct externalities creates a strong incentive for server producers to maintain a high degree of compatibility with existing browsers.

Indirect effects are derived from benefits that accrue to users of a technology as a result the availability of complementary services or products (Katz & Shapiro, 1985). In the web server industry, the availability of customer and third-party support, the existence of user-groups, the amount of knowledge about a product that exists in the form of trained users (i.e. touch-typing for QWERTY or Microsoft certified engineers) are all indirect network effects. These indirect network effects play an important role in the adoption of software products and are linked to the existence of complementary assets (discussed below). An illustration of this is the fact that individual users want to know that their skills are marketable and therefore have an interest belonging to an established and enduring network. Similarly, companies want to ensure that there is an ample supply of knowledgeable workers for the technology they are investing in as this helps them with lower labour costs and protects their investment for a longer period of time.

Other market factors are less complex and will be touched on in the exploration of other factors. On such factor, first-mover, does not require an in depth review and analysis. First-mover advantages have been shown to not influence the emergence of a dominant design in disk drive industry (Christensen et al., 1998). As well, in the VHS vs. Beta standards battle, being first did not lead to Beta becoming the established standard for the home video industry (Cusumano et al., 1992). Distribution networks, marketing and brand name may have also contributed to the study. However, these factors seem to have been relatively equal given the revolutionary nature of the Internet at the time of the study and the leveling effect it had on the ability to distribute the opposing products at very low costs. Also the marketing might of Microsoft was relatively neutralized as the “buzz” created in the media for open-source software and “free” software such as Netscape likely exceeded the capabilities of any marketing effort to inform people of these product options.

Technological Factors

Technological factors, in this instance, do not refer to the performance characteristics of the technologies in question. Performance characteristics are often disputed and subjective in nature. For example, popular belief is that Beta was a “superior” technology to VHS. However, in independent professional reviews the superior standard could not be agreed upon(Cusumano et al., 1992). In the context of this study, technological factors refer to the characteristics of change in the industry in question. Consumers can be concerned that the choice of technology they make today may be obsolete tomorrow. As well, the variety of choices and the rate at which they change can cause consumer confusion leading to delays in adoption decisions. This can be observed in the PC industry or the cell phone industry.

The rate and type of technological change in software affects the propensity of buyers to invest in a technology. In software, switching costs can be expensive, hence the importance of backward compatibility particularly in terms of data formats and user interfaces (Brynjolfsson & Kemerer, 1996; Cottrell, 1994). If new versions of software are not backward compatible users are less likely to upgrade. Conversely, users will be more likely to select technologies that demonstrate an upgrade path that protects their investment in data and knowledge of how to operate the software (Brynjolfsson & Kemerer, 1996). Overall, users seem to prefer incremental improvements to software that maintain the investment they have made rather than adopting radical new technologies that incur high switching costs which are typically borne by the user.

Non-Technological Factors

The major component examined in the non-technological factors category is Standards and regulations. The literature review clearly shows the linkages between standards and dominant designs, particularly interface standards. The case study presents an extensive history of the development of Internet standards. Attention is given to the processes by which the standards involved arise and the effect this process has had on the evolution of the World Wide Web. This study does not examine all Internet standards and is not intended to be a comprehensive technical review of those standards examined.

Other factors such as supply and demand side economic forces and organizational forces are outside of the scope of this study. However, it seems likely that standards are the major factor in this category of influencing forces on the establishment of a dominant design.

Strategic Factors

Strategic factors refer to those factors that can be influenced or controlled by the producing firm. In this study we will examine complementary assets, appropriability regimes, and strategic maneuvering. These are factors that may be employed by firms in an effort to influence the outcome of the dominant design competition.

Complementary assets are capabilities or assets that must be combined with the innovation in question in order for the producing firm to generate profits (Teece, 1986). Marketing, competitive manufacturing and after-sales support are all examples of complementary assets. In the case of the web server market there are several significant complementary assets influencing to outcome of a dominant design competition including:

- operating systems,
- scripting languages,
- modules (smaller programs or “plug-ins” which add new features to the server),
- the availability of technical support in the form of user communities and professional services.

Complementary assets give rise to indirect network externalities as users seek to belong to the larger network in order to leverage existing investments in data and knowledge (Christensen et al., 1998; Katz & Shapiro, 1985; Suarez & Utterback, 1995; Teece, 1986). In the case of the web server market the producing firms do not necessarily own the complementary assets. For example, the open-source developers of Apache do not own, develop or market any operating system. However, the existing installed base of UNIX servers, the first platform Apache was available on, serves as a complementary asset. Microsoft was not able to exploit this complementary asset since they do not develop software for the UNIX platform. This limited the ability of Microsoft to penetrate the existing Internet server market as most servers, especially early in the WWW history where UNIX based servers. Figure 6 provides data on user platforms, which in the early days of the Internet is a good proxy for the available server platforms as Internet Service Providers selling dial-up access to the Internet had yet to take off.

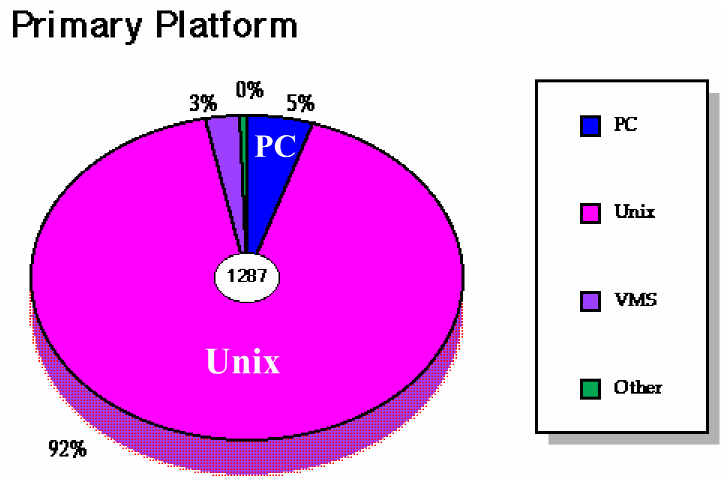

Figure 6: Internet Platform 1994 (GVU's WWW Survey Team, 2001)

Appropriability regime "refers to the environmental factors, excluding firm and market structure, that govern an innovator's ability to capture profits generated by an innovation" (Teece, 1986). Table 3 identifies key dimensions of an appropriability regime that may enable an innovating firm to capture profits from its innovation.

Legal Instruments	Nature of Technology
Patents	Product
Copyrights	Process
Trade Secrets	Tacit
	Codified

Table 3:Appropriability Regime: Key Dimensions (Teece, 1986)

In software industry strong appropriability regime is important due to the high fixed cost and near zero marginal cost of developing software. If software developers are not able to protect their innovation through one of these mechanisms then there will be little incentive for these firms to continue producing new software programs.

Contrary to most software markets, the web server market appears to have a very weak appropriability regime. The two major web server products available are both offered to consumers at no cost. It appears that the existence of a significant open-source software product removes the ability of firms to charge a price for a similar software product. It therefore seems counterintuitive that Microsoft would actively develop a web server if it is unable to charge a price for its software. However, if Microsoft did not offer a web server for its server operating system Windows NT (now Window 2000 Server), it would lose valuable market share in the server market to those users wishing to host a web site. Therefore, in some sense, the inclusion of a web server in Microsoft's server operating system at no extra cost to the consumer has become table stakes in the operating system industry.

Strategic maneuvering refers to the actions which firms may undertake to establish their design as dominant. For example, in the VHS vs. Beta competition, JVC (VHS proponent) employed a "humble" strategy of licencing to many manufacturers and programming producers while Sony (Beta proponent) employed a "greedy" did not licence their technology. It is widely acknowledged that this "strategic maneuver" buy JVC enabled VHS to overcome the significant installed base of Beta machines and become the industry standard (Cusumano et al., 1992).

Strategic maneuvering in the web server market has undertaken a number of forms. Foremost is the pricing of the web server software. Clearly the values of the open-source community have forced the price of web server software to zero. Microsoft has bundled their web server offering with their network operating system, Windows NT (now Windows 2000). As well, Microsoft has made many attempts to discredit open-source software through a tactic industry pundits term FUD (Fear, Uncertainty, Doubt). Through these tactics Microsoft attempts to undermine consumer confidence in software other than their own in various ways such as casting doubt on the reliability of the opposing software, by claiming that no support exists, or raising security concerns. Clearly strategic maneuvering plays a role in the establishment of a dominant design.

Summary

The factors detailed above all have a role to play in the establishment of a dominant design in a network market. These factors and the role they have played will be highlighted in the analysis of the web server market.

Chapter 4

Case Study

This case study examines the contest between open-source software and proprietary binary-only software in the Web server software industry. The web server software market provides an excellent opportunity to observe the dynamics between these two competing designs for several reasons. First, data on the growth in the number of web servers on the Internet is readily available. Second, the open nature of Internet standards facilitates the gathering of qualitative data on its evolution. Third, the rapid evolution of the Internet and the World Wide Web since the early 1990's provides a compact period of time to study.

Chapter 3 outlines the various factors that this study is examining with respect to the establishment of a dominant design in the web server software market. Many of these factors are qualitative in nature or are difficult to measure in a quantitative manner. For this reason the case study is a descriptive study supplemented with quantitative data where available. Throughout the narrative, the influence of various emergence factors becomes evident. The importance of standards and the unique standards process of the Internet, the relationship between the number of users and the frequency of new versions of the web server software packages, and the interdependencies between the Internet, computing platforms, user knowledge and web server software will be presented. While it is difficult to determine the winner of a dominant design contest before it is over, this study gives some guidance on what factors are important and how they may be observed.

The narrative begins with a brief history of the Internet. It is particularly important to note the standards setting approach used by the pioneers of the Internet as it has a significant effect on its evolution and the evolution of open-source software. A brief discussion on the adoption of the Internet by commercial entities is provided to establish a link to industry. The rise World Wide Web or WWW is documented providing the background for a more in-depth examination of the web server industry. The history of the WWW documents the evolution of the main protocols providing the technological platform for the web server software market. The use of these protocols by software developers allows the interoperability of browsers and servers and establishes the foundation for the now large network of web browsers and servers.

A detailed history of the Open Source software movement provides an understanding of the differences between this design path and the proprietary executable-only software development and distribution design path. This history describes the origins of open-source software and draws the links to the Internet and the convergence of personal computing, the Internet and the availability of software programming skills worldwide. The case is completed with a detailed examination of the web server market, a model case for exploring the dynamics of the battle between open-source software and proprietary executable-only software.

Given the complex nature of technology based industries it is appropriate to present the context of the battle for dominance through a detailed description of the underlying technologies and markets. It is important to note the lack of discussion relating to the technological performance or features of the web server software under study. It is a main tenant of this study that these factors do not play a significant role in the establishment of a technology as the dominant design. The assumption being if one of the choices was clearly inferior to the point that it did not meet overall end-user needs then this choice would not gain market share for any sustained period. For this reason we can assume that the main competitors in this battle are substitutes from a technology feature and performance point of view and that there must be other factors influencing the establishment of a dominant design in the web server software industry.

History of the Internet

The Internet as it stands today is the result of over 30 years of research and development in communications and networking. It is an amazing technological as well as social innovation, and it may be one of the most successful examples of sustained research in the information technology field (Leiner et al., 1998).

The Internet originates from research at MIT that began in the early 1960s with the publication of a paper on packet switching theory by Leonard Kleinrock in July, 1961 (Leiner et al., 1998). Researchers worked in parallel on packet switched networking during the mid-1960s as it became apparent that the circuit-switched based telephone network was inadequate for computer-to-computer communications. As the various parties researching packet switching became aware of each other's work a more centralized effort to develop a packet switched wide-area network was established by DARPA, the Defense Advanced Research Projects Agency in the US. This collaboration resulted in the creation of ARPANET, the first wide-area computer network that would eventually evolve into the Internet of today. Figure 9 shows the evolution of the funding agencies (top row), the engineering standards bodies (second row) and major technology milestones in the evolution of ARPANET to the Internet.[3]

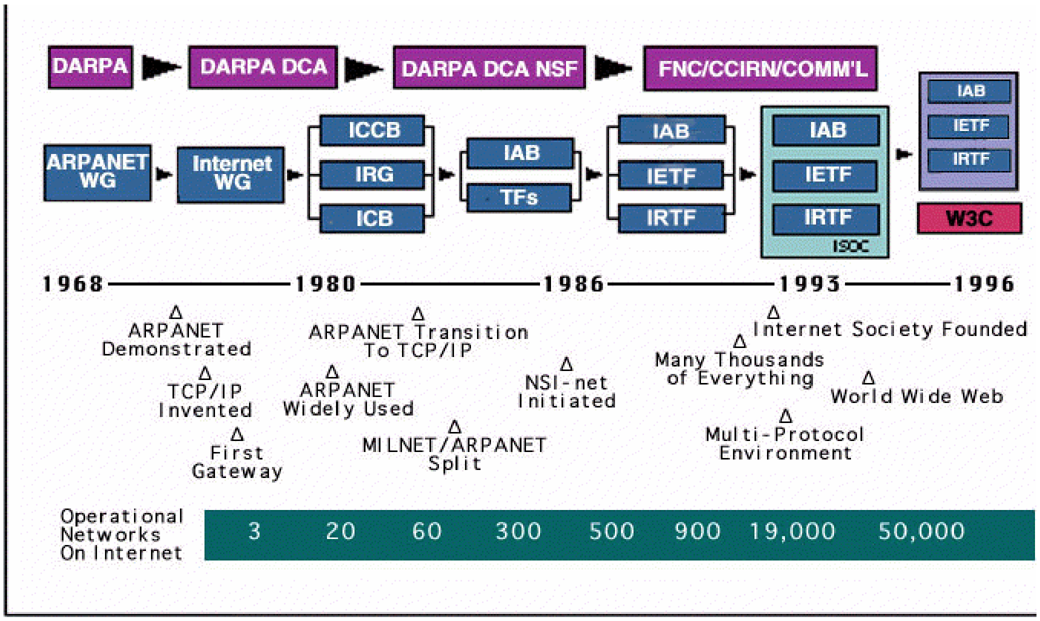

Figure 7: ARPANET to Internet Timeline (Leiner et al., 1998)

Early in the development of ARPANET the concept of an open architecture for networking was introduced by Robert Kahn, a researcher at DARPA (Leiner et al., 1998). Four key ideas guided Kahn's thinking and the subsequent development of the networking protocols that were to become the bedrock of the Internet:

- Each distinct network would have to stand on its own and no internal changes could be required of any such network to connect it to the Internet.
- Communications would be on a best effort basis. If a packet didn't make it to the final destination, it would shortly be retransmitted from the source.
- Black boxes would be used to connect the networks (these would later be called gateways and routers). There would be no information retained by the gateways about the individual flows of packets passing through them, thereby keeping them simple and avoiding complicated adaptation and recovery from various failure modes.
- There would be no global control at the operations level.

[3] An excellent source for information on the history of the development of the Internet is available at: http://www.isoc.org/internet-history.

(Leiner et al., 1998)

These key technological concepts in developing an open architecture network were accompanied by equally important methods for collaboration between the parties involved in creating and developing these protocols. The primary collaboration mechanisms were established by S. Crocker (from UCLA) in the form of Request for Comments notes or RFCs (Leiner et al., 1998). Still used today, the RFC is a document pertaining to a specific protocol or informational aspect of the Internet's operations. Originally RFCs were distributed, via the postal service, as a means of speeding up the collaboration process between researchers. As the infrastructure became available, the exchange of RFCs became electronic. RFCs are now available to anyone with Internet access at no cost[4]. The free and unrestricted availability of the detailed specifications for Internet protocols and the ability for anyone to comment on the development of these protocols is a critical factor in the success of the Internet (Leiner et al., 1998). Today, the Internet Engineering Task Force (IETF), an open membership organization for any individual (not organization) who wishes to contribute to the development of the Internet's technological underpinnings (Bradner, 1998), administers this process. The continued openness of both the development of the Internet's technology and the accompanying documentation are critical to the future success of the Internet (Leiner et al., 1998).

While the Internet originated in the early 1960s its recent rise to popularity began in late 1991 - early 1992. Since that time the Internet has grown in both the number of hosts and the number of users at an astonishing rate. A host is a device, usually a computer, connected to the Internet on a permanent basis. A host can support a number of users. Table 4 and Figure 7

---

[4] A complete set of Request for Comments is available at http://www.rfc-editor.org/

document the growth of the Internet in terms of the number of hosts. This growth has been a major factor in the acceptance of the Internet by many organizations and individuals.

Date	Survey Host Count	Adjusted Host Count	Replied To Ping*
Jul 99	56,218,000		
Jan 99	43,230,000		8,426,000
Jul 98	36,739,000		6,529,000
Jan 98	29,670,000		5,331,640
Jul 97	19,540,000	26,053,000	4,314,410
Jan 97	16,146,000	21,819,000	3,392,000
Jul 96	12,881,000	16,729,000	2,569,000
Jan 96	9,472,000	14,352,000	1,682,000
Jul 95	6,642,000	8,200,000	1,149,000
Jan 95	4,852,000	5,846,000	970,000
Jul 94	3,212,000		707,000
Jan 94	2,217,000		576,000
Jul 93	1,776,000		464,000
Jan 93	1,313,000		

{first NEW Survey}

{last OLD Survey}

{NEW and OLD refer to the methods by which the host counts were obtained. Details on these methods can be found at the web site http://www.isc.org}

{* estimated by pinging a sample of all hosts}

Table 4: Internet Domain Survey, July 1999 (http://www.isc.org/)

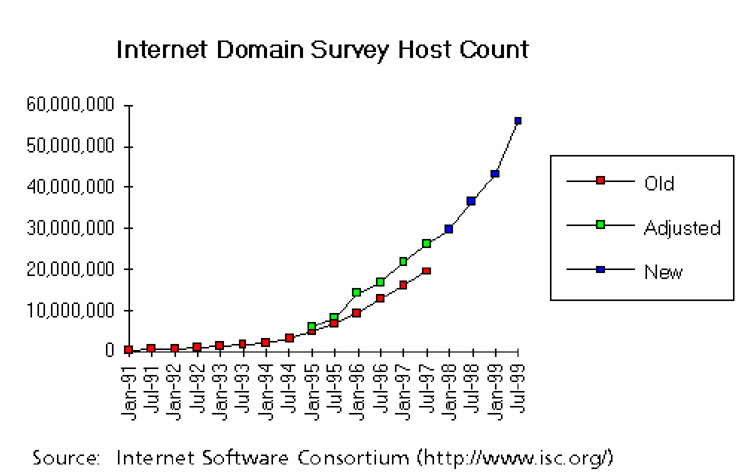

Figure 8: Internet Domain Survey Host Count

The Internet would not have been so amazingly successful both the technological and social (standards-setting) aspects of its development. An essential element to the current success of the Internet is the fact that its technological evolution was conducted in the open with the explicit goal of including as broad a range of networking technologies as possible. It is important to note that the underlying protocol to the Internet, Transmission Control Protocol/Internet Protocol, or more commonly TCP/IP was not the only solution to packet switched wide area networking. Many other proprietary protocols were developed simultaneously, mostly by for-profit organizations, such as SNA by IBM, DECnet by Digital, and XNS from Xerox (Leiner et al., 1998). However, it now seems clear that TCP/IP and the protocols that form the Internet are the winners in this technological race.

In October 1995 the Federal Networking Council (FNC), a forum for networking collaborations among Federal agencies to meet their research, education, and operational mission goals (Federal Networking Council, 1999), passed a resolution defining the term “Internet”.

> Definition of "Internet" 10/24/95
> --
> On October 24, 1995, the FNC unanimously passed a resolution defining the term Internet. This definition was developed in consultation with the leadership of the Internet and Intellectual Property Rights (IPR) Communities.
> --
>
> RESOLUTION:
>
> "The Federal Networking Council (FNC) agrees that the following language reflects our definition of the term "Internet".
> "Internet" refers to the global information system that --
>
> (i) is logically linked together by a globally unique address space based on the Internet Protocol (IP) or its subsequent extensions/follow-ons;
> (ii) is able to support communications using the Transmission Control Protocol/Internet Protocol (TCP/IP) suite or its subsequent extensions/follow-ons, and/or other IP-compatible protocols; and
> (iii) provides, uses or makes accessible, either publicly or privately, high level services layered on the communications and related infrastructure described herein."
> (Federal Networking Council, 1999)

Adoption of the Internet by Commercial Organizations

The acceptance of the Internet and Internet technology by organizations has been no less dramatic that the growth of the Internet in general. In fact, commercial or “dot.com” domains have been one of the main drivers of this growth, representing 33% of all hosts on the Internet as of July 1999. Table 5 provides data on the top 12 top-level domains, which represents 86% of all Internet hosts according the Internet Software Consortium’s domain survey (http://www.isc.org).

Rank	Domain	Hosts	Percent of Total	Full Domain Name
1	Com	18,773,097	33%	Commercial
2	Net	12,432,542	22%	Networks
3	Edu	5,141,774	9%	Educational
4	jp	2,072,529	4%	Japan
5	uk	1,599,497	3%	United Kingdom
6	Mil	1,561,756	3%	US Military
7	us	1,555,882	3%	United States
8	de	1,426,928	3%	Germany
9	ca	1,294,447	2%	Canada
10	au	907,637	2%	Australia
11	Org	821,933	1%	Organizations
12	Gov	683,363	1%	Government
⋮				
252	bi	0	0%	Burundi
TOTAL		56,218,330	86%	

Table 5: Top-Level Domain Name by Host Count, July 1999 (http://www.isc.org/)

Commercial organizations are adopting Internet technologies for a wide variety of purposes. In a 1997 study Liu et al. (1997) found that almost two-thirds of Fortune 500 companies had set up corporate home pages. These home pages provided a range of information to customers or potential customers including product/service descriptions, company overviews, interactive feedback and new announcements (Liu et al., 1997). During the nine-month study period of this research the provision of online business functionality on Fortune 500 home pages increased from 10% to over 25%, signifying the rapid and increasing use of more sophisticated Internet technologies by these information technology leading firms (Liu et al., 1997). A later study, conducted in June 1998, analyzing the web features used by Fortune 500 home pages found that 400 out of the 500 companies had home pages (Turau, 1998).

Organizations can employ a wide variety of Internet technologies in a wide variety of situations. Technologies such as Telnet (character based terminal protocol), FTP (file transfer protocol) and email have been available since the early 1970's. However, it was the introduction of the World Wide Web (WWW) and its HTML (HyperText Markup Language) and HTTP (HyperText Transfer Protocol) standards that sparked the rapid acceptance of Internet technologies in organizations. The WWW technologies simplified both the creation of content and the ability for people to access that content. WWW traffic accounts for a significant amount of Internet activity today. Table 6 gives an idea of the size and growth of the WWW since 1997. The WWW is a significant driving factor in the continued rapid growth of the Internet.

	1997	**1998**	**1999**
Web Sites:	1,570,000	2,851,000	4,882,000
Unique Sites:	1,230,000	2,035,000	3,649,000
Unique Public Sites:	800,000	1,457,000	2,229,000

% Change:	**'97 to '98**	**'98 to '99**	**'97 to '99**
Web Sites:	82	71	211
Unique Sites:	65	79	197
Unique Public Sites:	82	53	179

Table 6: Web Growth, Source: (Online Computer Library Center, 1999)

The World Wide Web

The World Wide Web (WWW), a term coined by Tim Berners-Lee in 1990 (World Wide Web Consortium (W3C), 1995b), is an application that uses the Internet as its communications system. The basic components of the WWW system are a server and a browser. The browser is a software program that allows users to access information, on their local computer, that is stored on various local or remote servers connected to the Internet. Servers are computers running a program that responds to the browsers' requests by sending standards-based encoded data to the browser using a standards-based protocol. The browser is then able to decode the data and display the results on a computer screen. The program that enables the servers to respond to the browser requests is web server software. Both the server and the browser implement the communications protocol, Hypertext Transfer Protocol (HTTP), and the data-encoding standard, Hypertext Markup Language (HTML). These two programs are the fundamental technologies of the WWW. (see Figure 10)

The WWW has its roots in a technology called Hypertext. Hypertext can be defined as "links between different parts of a document or between different documents creating a branching or network structure that can accommodate direct, unmediated jumps to pieces of related information." (Encyclopædia Britannica Online, 2000)

Tim Berners-Lee and Robert Cailliau, both working at the high energy physics labs, at CERN in Switzerland, were independently devising hypertext systems to facilitate the exchange of data and documents between scientists at various high energy physics labs throughout the world (Cailliau, 1995). In March of 1989, Tim Berners-Lee proposes a networked Hypertext system for CERN in a document, circulated at CERN for comments, titled *Information Management: A Proposal* (Berners-Lee, 1989). In the same time frame, Robert Cailliau independently proposes a hypertext project for documentation handling inside the CERN laboratory (Cailliau, 1995). By October of 1990, development had begun on the first hypertext GUI browser + editor. As well, Tim Berners-Lee and Robert Cailliau reworked the original Berners-Lee proposal for a presentation to CERN management. It was during this time in casual discussions between Cailliau and Berners-Lee that Berners-Lee devised the name World Wide Web for this new hypertext system (Cailliau, 1995). Figure 8 is a screen capture of the first World Wide Web browser – editor developed by Berners-Lee in 1990. The version show is actually from a 1993 version of the program, however the only differences in appearance are:

- The rendering would have been in grey scale as the NeXT computing platform on which it was developed did not display colour at the time;
- The inline images, such as the world/book icon and the CERN icon, would have been displayed in separate windows, the original version didn't render inline images. (Berners-Lee,)

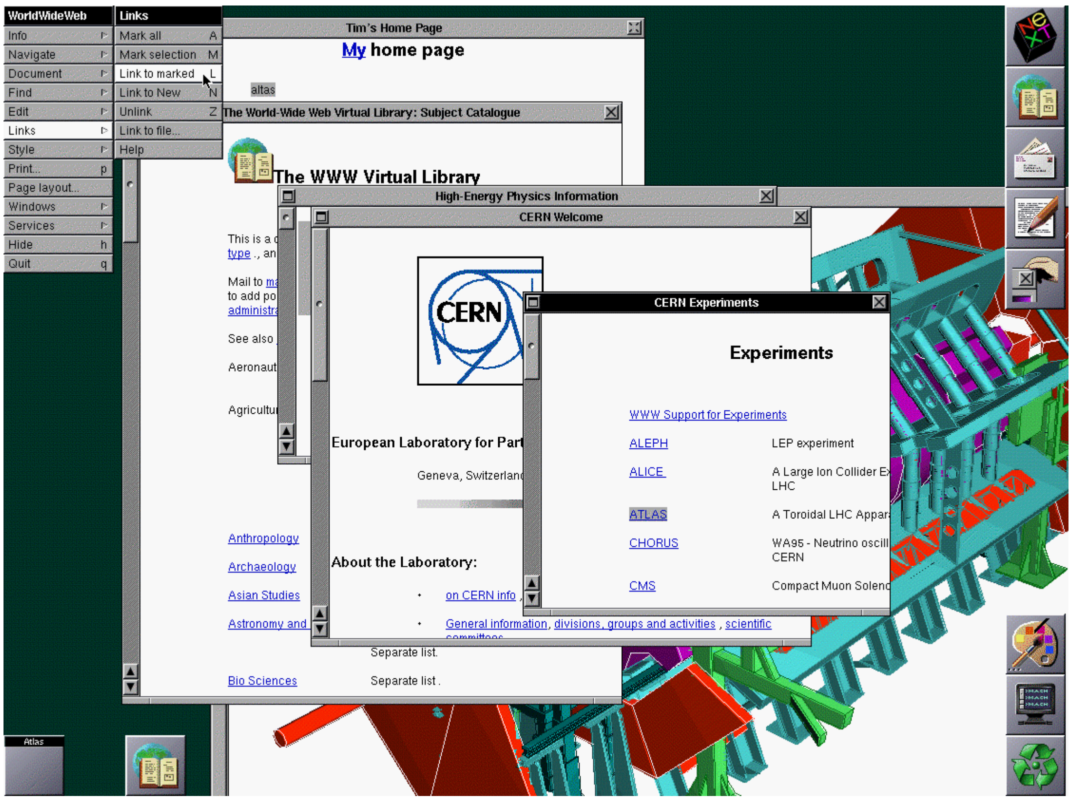

Figure 9: Original World Wide Web Browser – Editor (Berners-Lee,)

Work continued on the World Wide Web system within CERN during the first 8 months of 1991. In June 1991, version 0.1 of the CERN web server software is developed (World Wide Web Consortium (W3C), 1995a). In August of 1991 the small CERN team announced the availability of the World Wide Web system on the USENET newsgroups alt.hypertext (6, 16, 19th Aug), comp.sys.next (20th), comp.text.sgml and comp.mail.multi-media (22nd) (World

Wide Web Consortium (W3C), 1995b). This includes free and open access to CERN software source code for browsing and serving WWW pages. By December of 1991 the first web server in the USA was made operational at the Stanford Linear Accelerator Center (SLAC) in California. It served the contents of an existing, large database containing physics papers abstracts (Cailliau, 1995). It is likely that these first servers used the CERN web server program, the first publicly available web server.

1992 saw the introduction of several GUI browsers including the Finnish "Erwise" GUI client for X and Pei Wei's "Viola" GUI browser for X. Early in the year, the CERN Line mode v 1.2 browser was announced on alt.hypertext, comp.infosystems, comp.mail.multi-media, cern.sting, comp.archives.admin, and mailing lists. The WWW project group made presentations throughout the year to various High Energy Physics groups in the world (World Wide Web Consortium (W3C), 1995b). By the end of the year there exists about 50 known web servers on the Internet (Cailliau, 1995). In a posting to the new WWW-talk mail list, a mail list for people interested in developing software for the fledgling WWW, first mention of the NCSA web server software is made (Andreessen, 1992). Free and open access to the source-code was given for the NCSA web server software.

1993 marks the beginning of the dramatic rise of the WWW. In February, Marc Andreessen and Eric Bina, students working in the Software Development Group at the University of Illinois Champana-Urbane’s National Center for Supercompting Applications (NSCA), write the GUI WWW browser called Mosaic after seeing the Viola and Midas browsers. Mosaic is designed to be easy to install and robust. A major new feature, the ability to display images in the same window as text, or “in-line colour images”, is provided in Mosaic. This new ability dramatically increases the visual appeal and design flexibility of WWW pages.

By September of 1993, NCSA releases versions of the Mosaic browser for all common platforms: X, PC/Windows and Macintosh (World Wide Web Consortium (W3C), 1995b). The release of Mosaic is one of the key events that brings the WWW into the mainstream. Mosiac is licenced, several years later, by NCSA to Spyglass who then licences the source-code to Microsoft. The two most popular browsers today, Netscape Navigator (developed from scratch by the same people who wrote Mosaic) and Microsoft Internet Explorer (licenced Mosaic source-code) share the same source-code ancestry (Cusumano & Yoffie, 1998).

A number of key events occurred in 1993 that would shape the future of the WWW. One of the most important events was the announcement by CERN's directors that WWW technology would be freely usable by anyone, with no fees being payable to CERN (World Wide Web Consortium (W3C), 1995b). This is clearly an important milestone that paves the way for the unprecedented level of innovation occurring around the freely available open-standards based technology of the WWW. By September of 1993, WWW traffic represents 1% of traffic on the NSF backbone, the main backbone of the Internet. This is a ten-fold increase in WWW traffic in only 6 months. By the end of 1993 there are 250 known web servers and the first world conference on the WWW is announced (Cailliau, 1995).

Continuing its rapid rise to ubiquity, the WWW began to attract the attention of business. In January of 1994, O'Reilly, Spry and others announce the "Internet in a box" product based on the NSCA Mosaic software. These companies are the first commercial companies to attempt to bring the WWW into the consumer market (World Wide Web Consortium (W3C), 1995b). In March of 1994, Jim Clark founder of Silicon Graphics, founded Mosaic Communications Corporation along with six of the seven programmers from the NCSA group who developed the Mosaic browser and the NCSA web server software. The company name is later changed to

Netscape Communications in a name dispute with NCSA. Netscape's browser software became one of the most rapidly diffused software products in history and changed the terms of competition in the software market (Cusumano & Yoffie, 1998). In one short year, Netscape was able to generate 80 million dollars in revenue and reach a market valuation of 7 billion dollars (Cusumano & Yoffie, 1998). This marked the beginning of what has become an enormously innovative and incredibly lucrative industry.

On the heals of Netscape's rapid rise to success, Microsoft stepped into the Internet fray. On December 7th of 1995, held a special Internet Strategy Workshop where Bill Gates stated that Microsoft was "hard-core about the Internet" (Microsoft Corporation, 1999). Microsoft's main strategy was to "embrace and extend" the Internet by including Internet standards in all of their product lines and adding "improved" functionality to these standards. The entry of Microsoft into the Internet software market space both legitimized the Internet as a very important business technology and significantly altered the competitive landscape of the industry. While some industry observers consider Microsoft's entry into the Internet market to have been late, few would question the impressive speed with which Microsoft was able to transform their organization into an Internet-centric one.

In December of 1994, Tim Berners-Lee left CERN for the Massachusetts Institute of Technology (MIT) where a new organization is formed called the World Wide Web Consortium (W3C). The role of the W3C is to foster "the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability" (World Wide Web Consortium (W3C), 2000). The W3C is a vendor neutral, international industry consortium, led by Tim Berners-Lee and Jean-François Abramatic. It is funded by Member organizations and works with the global community to produce specifications and reference software that is made freely available. The W3C is jointly hosted by the Massachusetts Institute of Technology Laboratory for Computer Science (MIT/LCS) in the United States; the Institut National de Recherche en Informatique et en Automatique (INRIA) in Europe; and the Keio University Shonan Fujisawa Campus in Japan. The services provided by the W3C include: “a repository of information about the World Wide Web for developers and users; reference code implementations to embody and promote standards; and various prototype and sample applications to demonstrate use of new technology.” (World Wide Web Consortium (W3C), 2000)

At the end of 1995 the WWW was out of its infancy, with an estimated 73,500 web servers (Cailliau, 1995) up from 2,500 server the year before. The WWW was becoming synonymous with the Internet. This amazing innovation, in five short years, created one of the biggest David vs. Goliath battles in the software industry’s history: Netscape, the young Internet start-up, vs. Microsoft, the 15 year veteran that virtually defined the rules of the personal computer software industry. This battle, pitting Netscape with their Navigator product against Microsoft with their Internet Explorer product, brought in a new era of rapid technological development and rapid product cycles. New versions of these software products were being released every six months. This furious pace of change and competition has inspired new strategies for competing in fast-paced market places (Brown & Eisenhardt, 1997; Cusumano & Yoffie, 1998; Eisenhardt & Brown, 1998).

Meanwhile, whilst industry and the media were focused on this very competitive “battle of the browsers”, a new competitive landscape was developing, one that could have the potential of changing the rules of competition in the Internet software industry yet again. The web server software market, with its open-source roots continued to grow. Starting from the modest beginnings of the CERN and NCSA open-source web server software offerings, the web server software market space has grown to many millions of servers (E-Soft Inc., 2001; Netcraft, 2001). It is this market that will be focused on to illustrate the dominant design emergence framework.

Open Source Software

The term open-source[5] software came into being in February of 1998, however its heritage is much older than that. The roots of open-source software stem from the tradition of scientific advancement through the sharing of knowledge. Traditionally, in science, this sharing of knowledge is achieved in various ways including publishing the results of studies in journals, teaching, and conferences. This sharing is a primary tenant of the scientific method and allows other scientists to attempt to replicate or improve upon existing knowledge. In computer science, the sharing of results and the accompanying knowledge is achieved by sharing the source code to the research (DiBona, Ockman, & Stone, 1998). The sharing of source code helps ensure the advancement of knowledge in the field of computer science.

There are more specific roots to the Open Source movement and the subsequent rise in popularity of open-source software such as the Linux Operating System Kernel, the GNU tools, and the Apache web server. Two major traditions have contributed to the evolution of open-source software, the tradition of the scientific method and the open method of developing the technologies for the Internet primarily through RFCs.

The first major tradition, the sharing of knowledge for the advancement of science, is directly responsible for open-source software through the creation of the Free Software Foundation in 1984 by Richard Stallman. Stallman, commonly referred to in Internet discussions

[5] As in (Dyson, 1998), the term Open Source is used to refer to the Open Source Movement and the term open-source is used to refer to the open-source model of software development and distribution.

and the media by his email signature RMS, was a staff member at the AI labs of MIT during the 1970's and early 1980's. In the 1970's the culture at the AI lab, and the computing community at large, was very much of a software-sharing one (Stallman, 1998). People and organizations freely shared the source code to the software they wrote so that others could improve or adapt it to more closely fit their needs. This allowed people the freedom to innovate in ways that would not have been possible if they did not have access to the source code of the software they were using.

During the late 1970's and the early 1980's, coupled with the dramatic increase in the availability of computing power, a new era in software was rapidly gaining momentum. This movement was based the close control of the source code for software and on the selling of software for money without transferring the right to copy this software to users. The major difference with this movement was not the charging of a fee for software, but the restriction of the copying of this software and the withholding of the source code from users. This is the beginning of the rapid and very profitable rise of proprietary executable-only software. The idea of stealing software or "software piracy", critical to establishing the success of proprietary executable-only software, was made popular in the now famous "Open Letter to Hobbyists" written in February 1976 by the young software entrepreneur Bill Gates and published in the MITS Computer Notes journal. In that letter he "accuses hobbyists of stealing software and thus preventing good software from being written" (Gates, 1976). Gates concludes the letter with the statement "...Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software." (Gates, 1976). Here we can clearly see that the proprietary executable-only software proponents believe that "good software" is written by people or organizations who maintain close control over the source code for the software they licence for a fee. The socialization of users away from software-sharing to software piracy greatly disturbed the people who believed in software-sharing, especially Stallman. In 1984, seeing proprietary executable-only software as a threat to the spread of knowledge in the still fledgling field of computer science, Stallman established the Free Software Foundation to achieve the ultimate goal "…to provide free software to do all of the jobs computer users want to do--and thus make proprietary executable-only software obsolete." (Free Software Foundation, 1999)

The Free Software Foundation (or FSF) is responsible for the concept of "free software". In this context free has the meaning of freedom rather than free from monetary cost. In order to protect the rights of users the FSF created a legal document, now referred to as the the GNU Public Licence (GPL) or "copy-left" which clearly stipulates the three rights of the users:

1. the freedom to copy the program and give it away to your friends and co-workers;
2. the freedom to change the program as you wish, by having full access to source code;
3. the freedom to distribute an improved version (that must include the source code) and thus help build the community (If you redistribute GNU software, you may charge a fee for the physical act of transferring a copy, or you may give away copies.)

(Free Software Foundation, 1999)

The proponents of the free software movement, empowered by the strength of the GPL, felt that their approach to software development was superior to the software proprietary executable-only software vendors developed from both a moral (Stallman, 1998) and technical standpoint. This strong stance, coupled with the vocal reprimand of the proprietary executable-only software vendors as harmful to the science of software development, inhibited the wide spread adoption of free software and its accompanying development methodology. As the use of free software began to grow beyond a small population of experienced computer users it became apparent to some people within the free software movement that these ideological stances were limiting the wider adoption of their software and their methods.

Eric Raymond, one of the most well known proponents of free software, authored a paper titled “The Cathedral and the Bazaar” which was delivered at the Linux Kongress in May 21 1997 (Raymond, 1999). This paper describes the evolution of an open-source software development project based on the author’s observation of the development of the Linux Operating System Kernel. This paper provides some of the key reasoning behind the success of the free software development model and strongly influenced the future adoption of the free software development methods.

On January 31st 1998 a landmark announcement was made by Netscape Communications Corporation. Strongly influence by the reasoning presented in “The Cathedral and the Bazaar” and guided by Eric Raymond, Netscape was going to release the source code to their web browser and embrace the free software development model. Several days after this announcement, a group of well known contributors to the free software movement, including Raymond, got together to discuss how to best capitalize on this tremendous opportunity. They coined the term “Open Source” as a way of moving away from the rhetoric of the free software movement and began developing a more pragmatic and business oriented set of reasons why open-source software and its accompanying methods should be adopted by proprietary executable-only software vendors. In order to provide a forum for discussion and promotion of the newly created Open Source Definition and to advocate its adoption to software developers of all kinds, this group formed an organization called the Open Source Initiative (OSI).

The use of the open-source method of software development by commercial software developers is seen as an important goal of the Open Source software movement. To this end, the OSI created a set of criteria that software licences must meet in order to be officially (according to the OSI) open-source (Open Source Initiative, 1999). Although there is a vigorous debate within the free software community as to the necessity of the term open-source, there is an increasing acceptance of the Open Source method by previously strictly proprietary executable-only software developers. For example, IBM has adopted the open-source Apache web server as their core web server (Apache Software Foundation, 1998) and have made Linux (a popular open-source operating system) one of their supported operating systems for their hardware and software platforms (www.ibm.com/linux). The specific criteria a software licence must meet, as set out by the OSI, to be considered open-source is included in Appendix A.

Notwithstanding the recent success of several open-source software packages, the rhetoric around the pros and cons of proprietary executable-only software versus open-source software continues. Large proprietary executable-only software producers continue to vigorously pursue their markets. The “software piracy” issue is as important to the proprietary executable-only software vendors as ever (Banks, 1996). Microsoft continues to produce studies detailing the costs of piracy (Microsoft Corporation, 1999) while the open-source community continues to espouse the virtues of open-source software. While the answer to this debate is likely a matter of perspective, what has become clear is that both methods of software development and distribution are able to provide users with high quality software. Users and organizations are increasingly able to have a choice between using proprietary executable-only software or open-source software to solve software oriented problems. This is made very apparent by the wide spread use of open-source software such as the GNU/Linux operating system, the Apache web server, the BIND program, and sendmail by individuals and organizations around the world.

Further confirmation that open-source software is perceived as a viable approach to developing mission critical and competitive software is evident in both the stock market and venture capital investment. Leading venture capital companies such as Kleiner Perkins Caufield

& Byers, CMEA Ventures and Benchmark Capital are funding open-source start-ups (Raynovich, 1999). Wall street reacted very favourably to the Initial Public Offer (IPO) of Red Hat Inc. Traded on the NASDAQ stock exchange, the stock was sold for $14 on the IPO and quickly shot up to over $100, giving the company a market value of over $7 billion. While valuations for these companies, along with the entire stock market, have dramatically dropped they remain going concerns and many have announced a turn to profitability by fiscal year 2002.

Open-source software has made a significant impact in the continuing evolution of the information technology landscape. It is now clear that there are viable alternatives to proprietary executable-only software when building the infrastructure for an Internet based software system. What is not clear is the impact open source software development and distribution methods will have on the software industry as a whole. In order to gain some understanding of these effects, the web server software industry is examined. The web server industry provides an excellent example of these two software design and distribution methods in competition with each other.

The Web Server Software Industry

Core to the operation of the WWW is the web server. The web server is a computer that runs software enabling the delivery of information between web browsers and web servers via the HTTP protocol. Figure 9 provides a high level schematic the protocols and components involved in a typical WWW transaction. The software enabling these computers to deliver HTML content via the HTTP protocol is web server software. The technical details of the HTTP protocols and HTML are beyond the scope of this study. These specifications can be found at the W3C website (http://www.w3c.org).

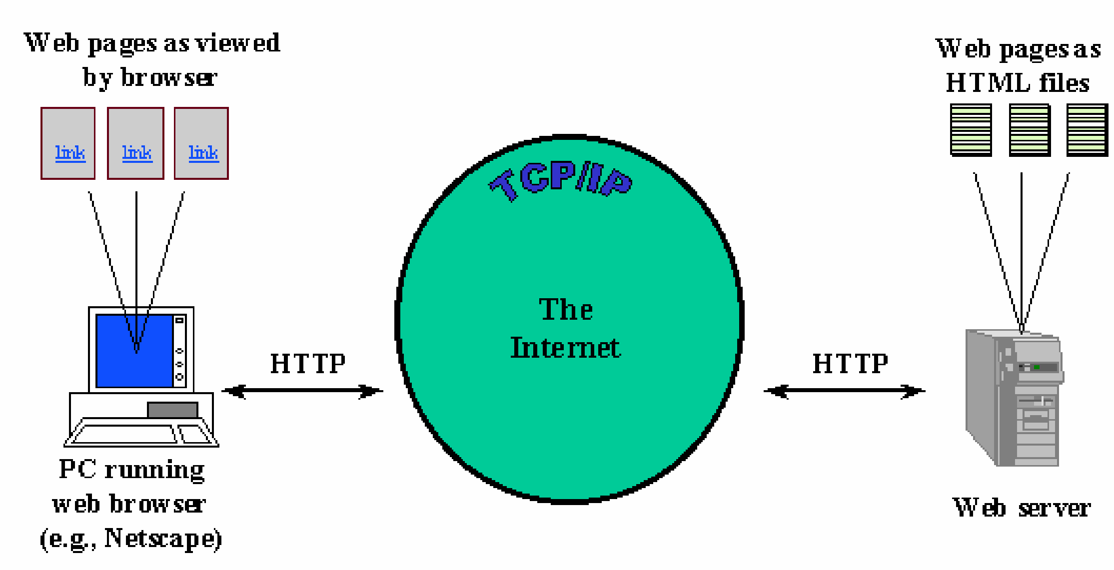

Figure 10: Overview of WWW protocols and components

Today three web server software programs dominate the web server market, Apache, Microsoft Internet Information Server and iPlanet (formerly Netscape). There are many other web server software programs in use, however, none of these programs command more than one percent of the total market. All combined, the market share of all other web servers software programs does not exceed 10 percent (E-Soft Inc., 2001; Netcraft, 2001) of the total market. Therefore this study will focus exclusively on the dominant web server software programs, Apache and Microsoft IIS representing approximately 86% of the total market. iPlanet is not investigated in detail since it only represents approximately 3% of the market and is not aggressively being developed..

Two companies track web server usage and make their results available to the public at no cost. Figure 10 depicts the market share of web servers as of April 2001 based on a survey of 28,669,939 sites by Netcraft.

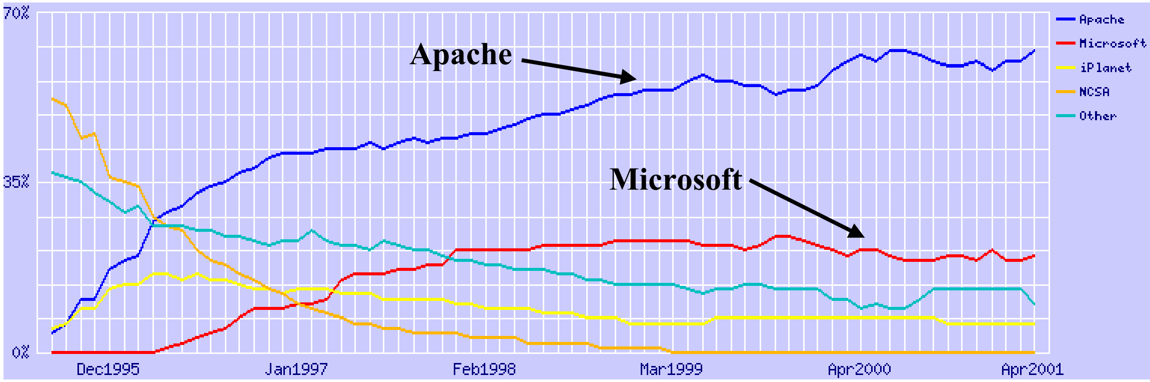

Figure 11: Market Share for Top Servers Across All Domains August 1995 - April 2001 (Netcraft, 2001)

Figure 11 depicts the market share of web servers for April 2001 based on a survey of 2,599,595 servers by E-Soft Inc. Both of these surveys are completely independent of each other and gather their samples in different ways. The methodologies they employ can be reviewed on their respective websites.

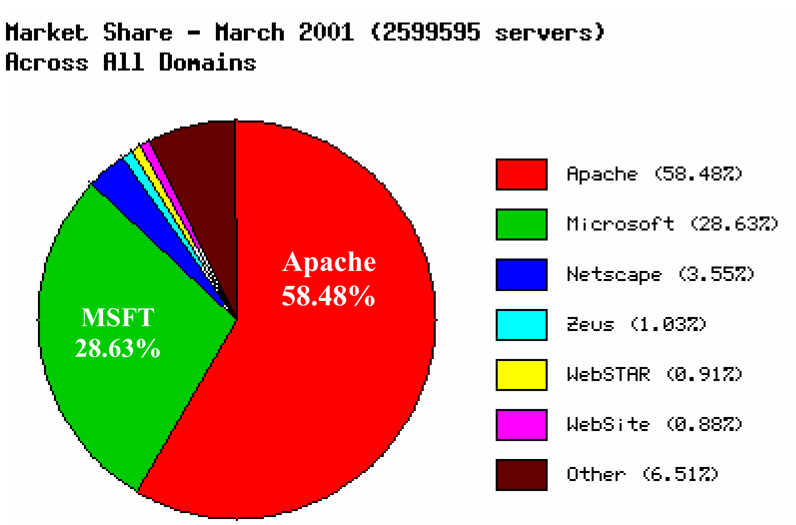

Figure 12: Market Share for Top Servers Across All Domains April 2001 (E-Soft Inc., 2001)

Earlier in this chapter the history of the WWW was presented. This history introduced the beginning of the web server market originating from the original CERN web server software. Since those early beginnings, the web server market has evolved into a battle between two main competitors, one open-source and one proprietary executable-only. The open-source web server software program is called Apache and the proprietary executable-only software program is call Microsoft Internet Information Server or IIS. The next two sections will describe the origin and evolution of the two competing web server software programs. These descriptions will include version release data, market share data, outline complementary technologies and identify required supporting technologies which act as an installed base for each program. In Chapter 5, this data will be related back to the emergence framework developed in Chapter 3.

The Apache Web Server Software Project

The Apache Project is an open-source software project aimed at creating a robust, commercial-grade, feature rich, and freely available source code implementation of a web server. As stated on the Apache Software Foundation web site, "The project is jointly managed by a group of volunteers located around the world, using the Internet and the Web to communicate, plan, and develop the server and its related documentation. These volunteers are known as the Apache Group. In addition, hundreds of users have contributed ideas, code, and documentation to the project (Apache Software Foundation, 2001)." The Apache server project led to the creation of The Apache Software Foundation (ASF). The ASF is a membership-based, not-for-profit corporation comprised of individuals who have demonstrated a commitment to collaborative open-source software development, through sustained participation and contributions within the Foundation's projects. An individual is awarded membership after nomination and approval by a majority of the existing ASF Members. The ASF provides a focal point and administrative structure from which to manage the resources necessary to develop complex software products.

The following history of the Apache project is adapted from the Apache Software Foundation web site and is included here for completeness (Apache Software Foundation, 2001).

In February of 1995, the most popular server software on the Web was the public domain web server developed at the National Center for Supercomputing Applications (NCSA), University of Illinois, Urbana-Champaign. However, development of that web server had stalled after a key developer left NCSA in mid-1994, and many webmasters had developed their own extensions and bug fixes that were in need of a common distribution. A small group of these webmasters, connected via e-mail, decided to band together for the purpose of coordinating their changes (in the form of "patches", hence the name a "patchy" server which evolved to Apache server). Two of these developers put together a mailing list, shared information space, and logins for the core developers on a machine in the California Bay Area, with bandwidth donated by one of the developers employers. By the end of February, eight core contributors formed the foundation of the original Apache Group.

Using the freely available, but no longer actively developed, NCSA web server version 1.3 as a base, all of the available bug fixes and worthwhile enhancements were merged into the existing software from NCSA. This software was tested on the servers of the original developers. The first official public release (0.6.2) of the Apache web server was made in April 1995.

This early Apache server was successful, however the Apache group felt that the software needed a general redesign. During May-June 1995, while development continued on the existing software to support the rapidly growing Apache user community, a new server architecture designed to be more robust, scalable, and modular was being developed. The Apache group switched to this new server software architecture in July 1995 resulting in the release of Apache version 0.8.8 in August 1995. Release 0.8.8 become the base for all subsequent releases of the Apache web server software.

After extensive beta testing, many ports to different operating systems, a new set of documentation, and the addition of many features in the form of modules (software features that can be easily added or removed from the main program by end-users), Apache 1.0 was released on December 1, 1995. Less than a year after the release of the first version, the Apache web server passed NCSA's web server as the most widely used web server on the Internet. Figure 12 and Table 7 detail the release history of the Apache Web server since version 1.0 was released on December 1, 1995. Table 8 provides some descriptive statistics for the release history of the Apache web server. The data for the Apache Release history was gathered from the apacheweek (editors@apacheweek.com, 2001) weekly web publication focused exclusively on the Apache web server. Each week a report is issued stating the current status of the Apache web server, the current released version, current beta activities and the type of release if a new one has been issued.

Week of Release	Version	Type of Release	Weeks since last Release
01-Dec-95	1.0.0	Major	-
09-Feb-96	1.0.2	Bug Fix	10.0
23-Feb-96	1.0.3	Bug Fix	2.0
26-Apr-96	1.0.5	Bug Fix	9.0
05-Jul-96	1.1.0	Major	10.0
12-Jul-96	1.1.1	Bug Fix	1.0
17-Jan-97	1.1.3	Bug Fix	27.0
06-Jun-97	1.2.0	Major	20.0
11-Jul-97	1.2.1	Bug Fix	5.0
22-Aug-97	1.2.4	Bug Fix	6.0
09-Jan-98	1.2.5	Bug Fix	20.0
27-Mar-98	1.2.6	Bug Fix	11.0
05-Jun-98	1.3.0	Major	10.0
24-Jul-98	1.3.1	Bug Fix	7.0
25-Sep-98	1.3.2	Minor	9.0
09-Oct-98	1.3.3	Bug Fix	2.0
15-Jan-99	1.3.4	Minor	14.0
26-Mar-99	1.3.6	Minor	10.0
20-Aug-99	1.3.9	Minor	21.0

Week of Release	Version	Type of Release	Weeks since last Release
25-Feb-00	1.3.12	Bug Fix	5.0
13-Oct-00	1.3.14	Minor	33.0
02-Feb-01	1.3.17	Minor	16.0
02-Mar-01	1.3.19	Minor	4.0

Table 7: Release History of Apache Web Server (editors@apacheweek.com, 2001)

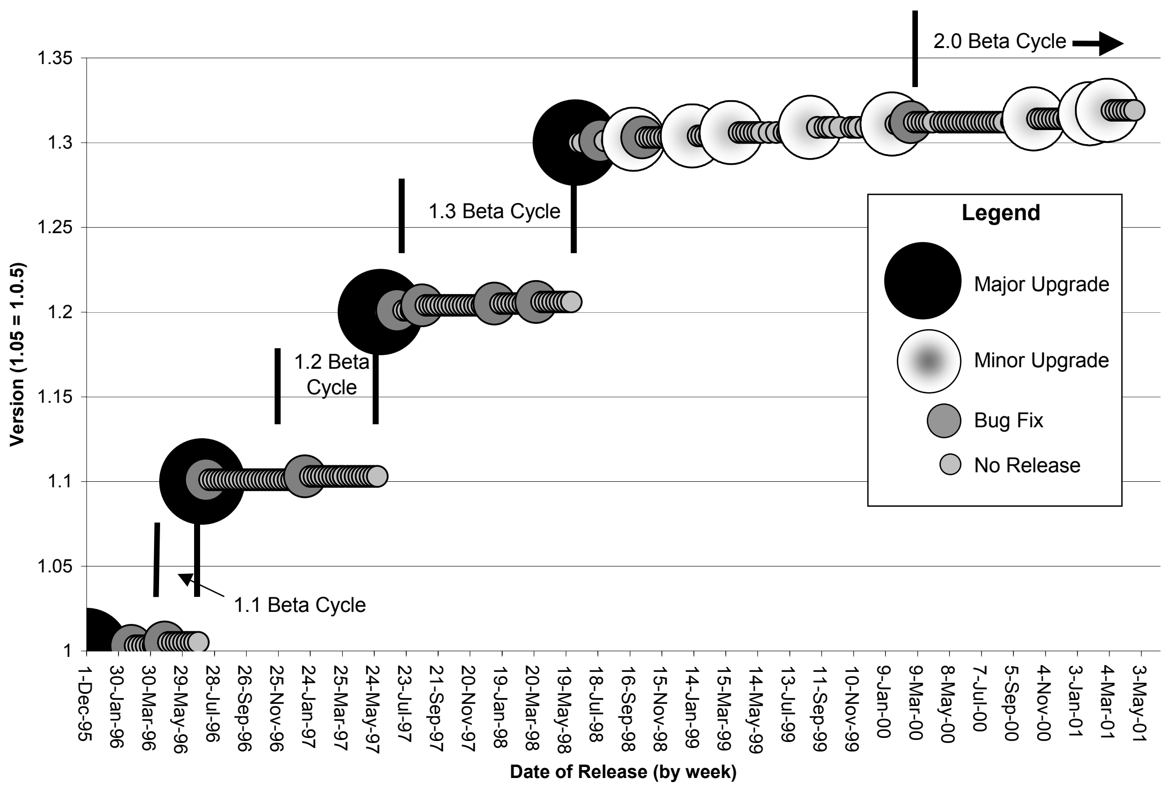

Figure 13: Apache Web Server Release History (editors@apacheweek.com, 2001)

Description	Weeks
Average time between releases	11.9
Average time between major releases	43.7
Average time between minor releases	17.9
Average time for bug fix releases since last release	8.8
Frequency of release (any type)	Once every 11.4 weeks

Table 8: Descriptive statistics of Apache Release History

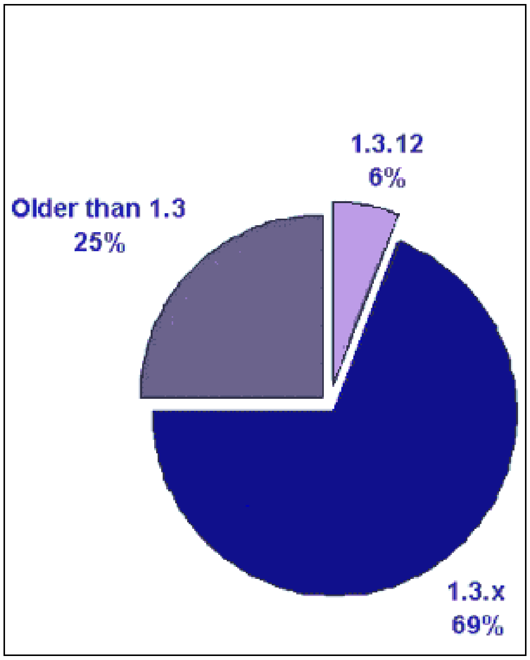

Figure 14: Apache Releases in Use in May 2000 (editors@apacheweek.com, 2001)

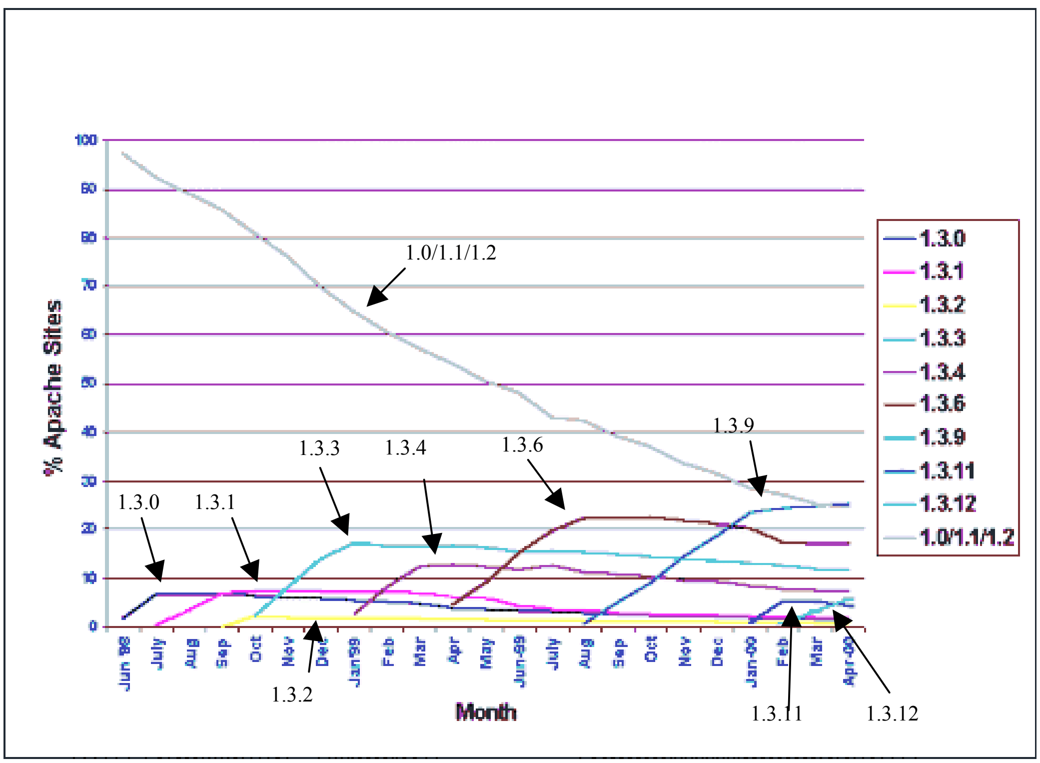

Figure 15: Take up of Apache by Release as of May 2000 (editors@apacheweek.com, 2001)

Figures 13, 14, and 15 depict the release history and subsequent adoption of the 1.3.x versions of Apache web server. While there are quite frequent releases of Apache the latest release is not taken up by the majority of the user community immediately (Figure 15). It is difficult to determine the cause for the lag in time between the release and the adoption of a new version of the Apache web server without performing a survey of the user population.

Apache web server is known to work on at least 28 different operating systems. It difficult to know exactly how many operating systems the Apache web server has been ported to as there is no known central list of ports. However, Table 9 provides a list of the operating systems for which a pre-compiled version of Apache web server can be downloaded from the Apache Software Foundation web site (Apache Software Foundation, 2001).

Operating System	Operating System
aix	netbsd
aux	netware
beos	openbsd
bs2000-osd	os2
bsdi	os390
darwin	osf1
dgux	qnx
digitalunix	reliantunix
freebsd	rhapsody
hpux	sinix
irix	solaris
linux	sunos
macosx	unixware
macosxserver	win32

Table 9: List of Operating Systems known to support Apache web server

The Apache web server has a "modular" architecture. Apache's 'modular' architecture makes is possible for anyone to add new functions to the server. Most of the functionality that comes as part of the Apache web server distribution is in the form of modules, and can be removed or replaced. There are a large number of modules written for Apache. Besides those included with the distribution, modules are also written to add functions not already in the code, or to do things which are needed on some web sites but are not of widespread use. Core Apache developers write some of these modules, however, most modules are written by other users of Apache who want to adapt its functionality to their needs. The ability to easily customize the Apache web server to specific applications is a very important capability of the software. These modules allow the Apache web server software to interact with a wide variety of other computer systems such as databases, scripting languages and specialized hardware. Modules provide a compatibility interface allowing end-users to leverage their existing systems, saving money and reducing learning curves.

An another important criterion in the adoption of software programs is the availability of technical support. End-users and organizations are generally very reluctant to adopt software that does not have reliable technical support or appears to be in risk of being abandoned by its developers. Fundamentally, these users to not want to be stranded with an obsolete, unsupported software product, particularly if the software is used in critical business systems.

Apache has no official software support organization. The developers of Apache accept bug reports and will fix these software bugs. However, this does not mean that there is no support for Apache web server. Actually, it is exactly the opposite. There exists a wide range of support options for users of the Apache web server. These options include, documentation, newsgroups, email list, web sites, and third party books. Given the large number of users of the Apache web server, it is very likely that one of the on-line sources of information will contain an answer or will have a contributor that is willing to post an answer to a question. There are a number of companies, including IBM, which will support the Apache web server for a fee. While this form of support is different than the traditional support model for proprietary executable-only software it appears to be working very well.

In summary, the Apache web server is the most widely used web server in the world with market share of over 58%. Apache supports a wide variety of operating systems, is frequently updated to add new features and fix bugs, is compatible with existing systems through the use of modules and has excellent support options. New features are delivered to the user community in the form of minor feature releases much more frequently than Microsoft Internet Information Server. It is interesting to note, however that Version 2 of Apache has taken quite a long time to develop while Microsoft has released its latest web server. Version 2 of the Apache web server is a significant rewrite of the software project and it is unknown how different Version 5 of Microsoft's web server is from Version 4, although it seems unlikely that it would be as significant a change as the Apache server is undergoing. Given all these positive attributes it appears that Apache will continue to do very well in the web server market.

Microsoft Internet Information Server

Microsoft Internet Information Server (IIS) is Microsoft's web server software product. First released in February 1996, IIS was initially provided as a free download from Microsoft's web site. In later versions, IIS is bundled with Windows NT Server. To this day, IIS is bundled with Microsoft's server operating systems software.

Being a proprietary executable-only product, information about IIS is not as readily available to the public as it is for Apache. Extensive efforts were made to get release information and development effort from Microsoft directly. However, given the highly competitive nature of the industry and the on-going legal issues Microsoft faces with respect to anti-competitive behaviour, people within Microsoft that were contacted were unable or unwilling to provide the same kind of information which is publicly available for Apache. Therefore, public sources were used to obtain release history data for IIS. Microsoft's public press release and support web sites were primary sources of data. Press releases from January 1996 to April 2001 were reviewed for information pertaining to release dates for IIS. Microsoft releases bug fixes and new functionality to there operating system software via a mechanism call service packs. The Microsoft support site was referenced to determine the contents of the service packs and to confirm release dates based on the date of the executable file on the support web site. Often the data was confirmed in the trade magazines covering Microsoft technologies. Tables 10, 11 and Figure 13 document the release history of IIS.

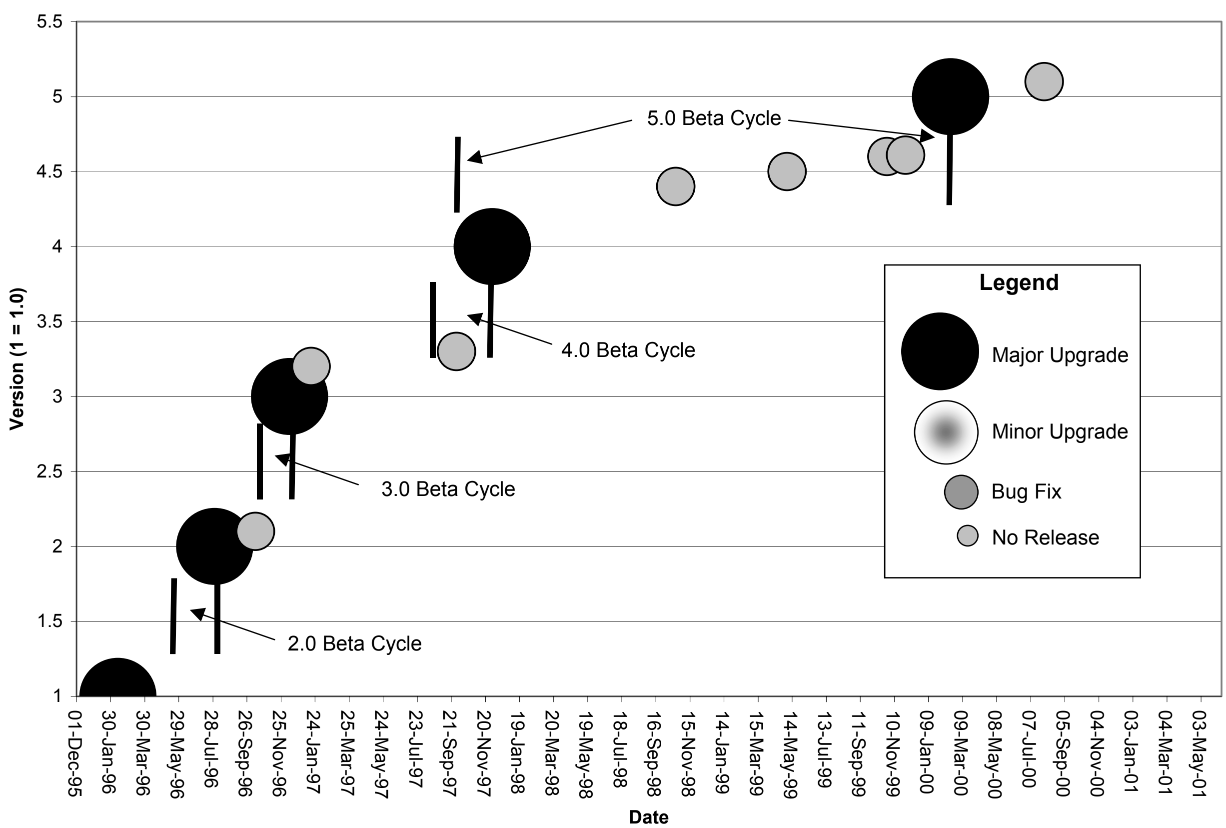

Figure 16: Release History of IIS (from various sources)

Week of Release	Version	Type of Release	Weeks since last Release
12-Feb-96	1.0	Major	
31-Jul-96	2.0	Major	24.3
11-Oct-96	2.0 SP1	Bug Fix	10.3
10-Dec-96	3.0	Major	8.6
17-Jan-97	3.0 SP2	Bug Fix	5.4
30-Sep-97	3.0 SP3	Bug Fix	36.6
02-Dec-97	4.0	Major	9.0
21-Oct-98	4.0 SP4	Bug Fix	46.1
05-May-99	4.0 SP5	Bug Fix	28.0
28-Oct-99	4.0 SP6	Bug Fix	25.1
30-Nov-99	4.0 SP6a	Bug Fix	4.7
17-Feb-00	5.0	Major	11.3
31-Jul-00	5.0sp1	Bug Fixes	23.6

Table 10: Release History of IIS (from various sources)

Description	Weeks
Average time between releases	19.4
Average time between major releases	52.4
Average time between minor releases	n/a
Average time for bug fix releases since last release	22.5
Frequency of release (any type)	Once every 17.4 weeks

Table 11: Descriptive statistics of IIS Release History

Unlike Apache, IIS only runs on a single operating system. This operating system is Microsoft Windows NT Version 4 and Windows 2000 (the new name for Windows NT Version 5). Although IIS is provided as no extra cost, Windows NT is not a free operating system. Therefore there end-users wishing to use IIS must purchase Windows NT. End-users wishing to use Apache may choose one of the freely available Unix operating systems such as Linux or BSD Unix. The tight bundling of IIS to Windows NT limits the growth of IIS as a standalone product as many Internet facing servers are Unix based.

End-users are able to develop sophisticated web based applications using IIS as the core technology. IIS has the ability to connect to legacy systems and databases through Microsoft and third party extensions. While users do not have the ability to modify the source code of IIS there is an application programming interface which allows software developers to control certain aspects of IIS's behaviour. This functionality allows new features to be added to IIS by non-Microsoft programmers. This customization capability fulfills a similar role to the Apache modules, albeit in a different way.

Support for Microsoft IIS, and Microsoft products in general, is provided through an extensive network of value added resellers, certified training programs for Information Technology specialists and by Microsoft itself. Further to these support mechanisms there are many online resources, book, magazines and some TV shows which cover Microsoft technologies and are sources of information for end-users.

In Summary, IIS is the second most popular web server on the Internet and is the only serious challenger to Apache with approximately 28% of the total market. IIS is only available on one platform and does not have the flexibility to be customized to run on other operating systems or devices. Since IIS is tied to the Windows operating system is appears that bug fixes, minor releases and major releases are decreasing in frequency. Windows is a very complicated program and as it gets more complicated and is required to support more devices, it seem likely that Microsoft will not be able to adapt to new standards and features with the frequency that Apache will be able to.

In the next chapter, this extensive history of the Internet, open-source software and the Web server market will be summarized using the emergence framework detailed in Chapter 3.

Chapter 5

Discussion and Conclusions

"The emergence of a dominant design in the disk drive industry was a process that spanned a decade – it was not a discrete event" (Christensen et al., 1998)

Chapter 3 described a framework for detecting the emergence of a dominant design. The main purpose of this discussion and subsequent conclusions is to provide guidance on how to apply the framework to a real-world scenario, in this case the web server software market or more generally, the Internet standards-based software industry. As is evident from the length of the case study material, the amount and type of data is large and diverse. Therefore it takes a variety of methods combined with experience and knowledge in the industry under study to begin to understand the forces shaping the next dominant design. Given the significance of the establishment of a dominant design to an industry and its constituent firms, it is important to continue to improve the ability of managers to evaluate the stages of technological evolution in their industry. This paper provides an example of how to apply a dominant design emergence framework for managers attempting to understand technological evolution in industries where standards and network effects play strong roles.

Applying the Emergence Framework

Market Factors

The primary variable explored in the Market factors portion of the emergence model was network effects. Network effects can be both direct and indirect and have strong links with compatibility standards. Network effects clearly have a strong influence in the Internet software Industry.

Direct network effects arise from the requirement that web browsers and web servers must adhere to the same transaction (HTTP) and content (HTML) standards. This affects both Apache and IIS. Neither software product is able to dictate these protocols as neither the Apache Software Foundation nor Microsoft control both the browser and server market. It is possible for Microsoft to customize protocols for their web browser and web server products such that they will only work in conjunction with each other. This does occur to some extent, however Microsoft has strong incentive to support standard HTTP and HTML because most web servers are not using Microsoft software. If Microsoft were to stop supporting these standards in either their browser or server products they would be removing themselves from a large and well-established network of users. As long as one organization does not control the market for both the browser and servers then direct network effects have the effect of enforcing compliance to a minimum set of open transaction and content standards.

Indirect network effects arise in the web server software industry in several ways. In the case of the Apache server, indirect network effects were triggered when the developers decided to maintain compatibility with the NCSA web server configuration and log files. This provided a smooth upgrade path for existing web servers and allowed web server administrators to maintain their investment in knowledge as well as content. New Apache software users would also have had access to a pool of people through newsgroups, mailing lists and websites of experienced users, lowering their startup costs. The nature of open-source software and the modular architecture of the Apache web server also provided a fertile environment for the development of extensions to the basic server. Clearly this was a successful strategy as the Apache web server has been ported to over 28 operating systems and supports many different modules allowing Apache perform a wide range of services from secure web sites to dynamic content to database driven applications.

Similarly, IIS is subject to various indirect network effects. Given that Apache is primarily a Unix based implementation of a web server, Microsoft needed to provide a web server solution that preserved the knowledge Windows NT server operators had developed. IIS integrates into the Microsoft server management framework accomplishing this goal. Integrating IIS into the Windows NT operating system allowed Microsoft to include IIS in Microsoft certification curriculums and provided a support network for users through Microsoft's solution provider program.

Clearly network effects, both direct and indirect, have played an important role in the development of the web server market. Direct effects allowing the growth of the web by re-enforcing the importance of adhering to open standards and indirect effects causing the two opposing design choices, open-source and proprietary executable only, to maintain their users investments in knowledge, content and support.

Technological Factors

Contrary to common opinion, features and performance factors are not the significant predictor of a technology rising to dominance. Both Apache and IIS provide similar functionality and performance and can be consider substitutes for each other. Therefore there must be other factors effecting the adoption decisions of end-users with respect to the technological choices they face. In the framework presented these factors are type of technological change and the rate of change. Both of these factors are documented in the case study (see figures 13-16 as well as tables 7,8,10, and 11). Apache appears to adhere to an incremental frequent minor and bug fix release model and IIS appears to favor an occasional major release model with some bug fix releases.

From Figure 17, it appears that Apache is behind IIS in the latest release, however as mentioned in the case study, the 2.0 version of Apache is a significant re-write of the software and the 5.0 version is likely an incremental version of IIS. As well, Figure 17 does not depict the eight minor releases of Apache providing new features to the user community during the same time period that Microsoft provided four bug fix releases and one major release (bundled into Windows 2000).

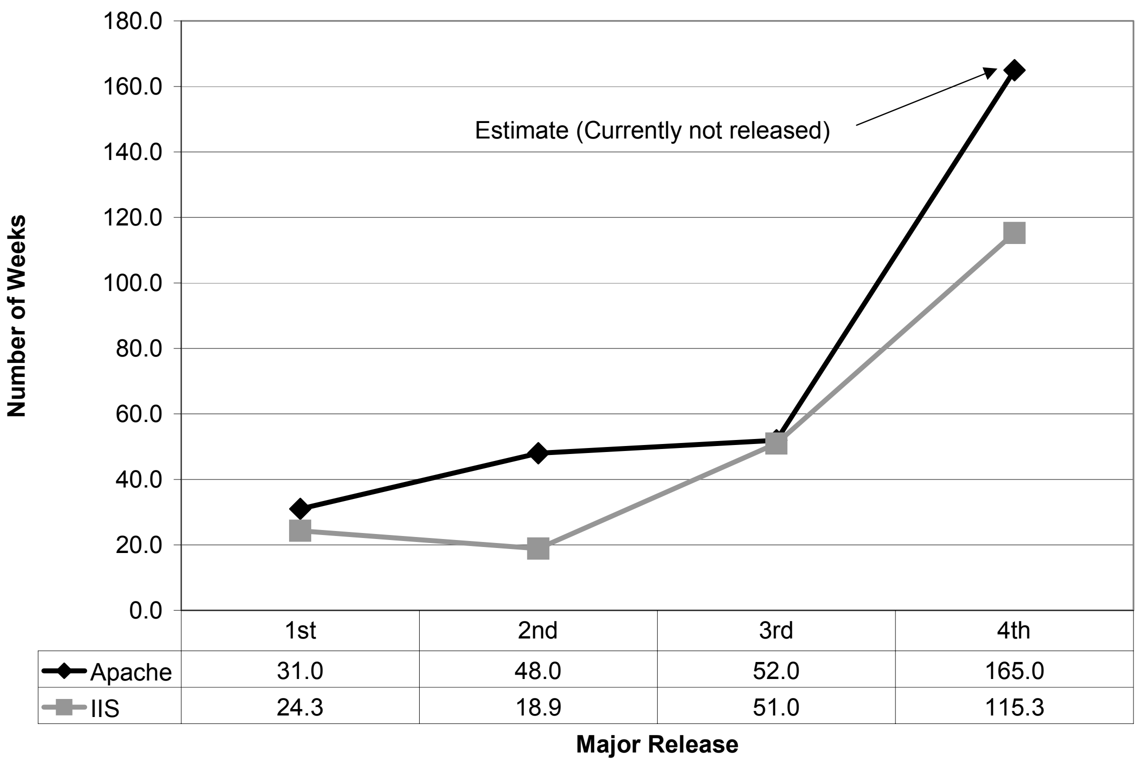

Figure 17: Comparison of Time between Major Releases for Apache and IIS

The 5.0 release of IIS is bundled with Windows 2000 server requiring the user to upgrade the entire operating system and not just the web server, a significant undertaking in many cases. The open-source Apache web server approach to version releases appears to fit the model where users prefer incremental improvements to major changes despite the fact that the entire community doesn't adopt the latest release. This approach of more frequent increment releases indicates to the user community continued support for their technological choice and provides strong signal that their investments in both learning and technology will be maintained.

Non-Technological Factors

Standards have played a critical role in the evolution of the web server market and in the evolution of the Internet software industry as a whole. Clearly an enabling factor, standards provide the glue that allows the Internet to remain a cohesive technological suite of capabilities. Two standards have been highlighted in the study, HTTP and HTML. HTTP defines the communications mechanism between browser and server and HTML defines the display characteristics of the content, allowing browsers to correctly display the data retrieved from web servers. Both of these standards have evolved in the open.

The Internet standard setting approach has ensured the continued independence of these core web standards from the control of a single, powerful organization. Further re-enforcing the open nature of the standards setting process is the availability of an open-source implementation of the standards. The developers of open-source implementations do not have an incentive to attract customers and generate revenue. Their main incentive is to provide a standards-compliant implementation. The widespread availability of the open-source implementation forces the proprietary providers to adhere to standards as the network becomes defined by software that is compliant to the standards. This also applies to proprietary implementations of the standards as long as the developing firm does not try to evolve the standard outside of the process. Netscape generally followed an open standards strategy.

Overall, the adherence to standards by all the major providers of web server and browser software has given rise to tremendous amounts of innovation and growth. The WWW gained widespread popularity very rapidly for such a young technology. It seems quite clear that open standards and open-source software played and important role in this growth.

Strategic Factors

The strategic factors examined in this study were complementary assets, strategic maneuvering and appropriability regimes. Normally these factors would be employed by the competing firms in an attempt to establish their technology as dominant. However, the structure and motivation of the Apache Software Foundation and open-source developers in general differs from for-profit firms. Nonetheless, developers of open-source software clearly make decisions about licencing, supported platforms, distribution, and promotion of their software project. The ASF has developed a clear mandate to guide the creation and maintenance of open-source software including the Apache web server and other related software projects. Microsoft has a different objective, to promote the use of their software solutions to generate profits for their shareholders. These differing perspectives require a slightly different interpretation of the strategic factors.

Open-source developers are most likely to initially develop software for platforms in wide use and with appropriate development tools available. For example, hardware device drivers for the Linux operating system are typically available first for popular hardware followed by less popular devices. Since Apache web server was a direct evolution of the already available Unix-based NCSA web server, the primary operating system for Internet servers in the early 1990's was Unix, and there are freely available development tools for Unix (compilers, editors, debuggers) the open-source developers created Apache for Unix first. This decision, essentially a strategic maneuver, had the effect of forming one of Apache's primary complementary assets, an existing base of users familiar with the administration of Unix servers and the configuration of their web server. Microsoft did not have any significant presence in the Internet server market and therefore needed to build an installed base of users familiar with Windows NT technology as an Internet server.

A second strategic maneuver by the ASF was to implement a modular architecture early in the release cycle (at Version 0.8.8) of the web server. This redesign had the effect of allowing other developers to create modules enhancing the capability of the basic Apache web server. This decision has clearly enhanced the usefulness of the Apache web server and has allowed it to evolve as new needs arise in the user community without requiring modification to the core web server software. The open-source development model leverages the ever-increasing knowledge base of software developers worldwide, another important complementary asset that is difficult for proprietary executable-only developers to leverage. Microsoft IIS also has the capability to be expanded via add-ins using Microsoft's application interface. However, users are not able to customize the core product and have a less opportunity to have bugs fixed and smaller new features made available as is evident in the release history of IIS.

The final strategic maneuver made by the ASF was to develop the web server as an open-source project. This fundamentally altered the appropriability regime for the web server market. If the original developers of the Apache web server had decided to form a company and continue the development of a web server as a proprietary-executable only software product they would likely have not been able to defend against Microsoft. Clearly this is what happened to Netscape. By remaining an open-source project the ASF removed the revenue-generating proposition from the market. Microsoft's only response was to provide their web server software as a feature of Windows NT in order to develop a market share for NT in the Internet server market. Subsequent bundling of the Apache web server into products competing with Windows NT by organizations such as IBM, Red Hat and Sun further increased the need for Microsoft to continue development of IIS without increasing the cost of Windows NT to the end-user.

Conclusion

As the opening quote to this chapter indicates, the emergence of a dominant design is a process that spans a long period of time. It is unlikely that any model can predict the winning design, however it is still of strategic importance to firms to track competing designs and to attempt to influence the outcome of a design competition. The main purpose of the framework is to provide guidance to managers and researchers on factors that influence the emergence of a dominant design. The application of this framework requires in-depth knowledge of the industry and technologies involved in the competition. This study has provided an in-depth analysis of one industry showing the breadth of information required to operationalize the factors in the context of the emergence framework.

At this point in the evolution of the Internet software industry, it appears open-source software development and distribution methods are favoured. Most of the major Internet infrastructure technologies, including email, domain name resolution, and web servers are implemented with open-source software. In applying the framework to the web server industry several factors arose which seem to provide an understanding of why open-source software has emerged as a significant design path in software development and distribution. In summary, these factors are:

1. Open-source software is typically implemented on Unix based platforms, which is the primary Internet Server platform. The availability of free development tools, the installed base, and the level of industry knowledge in Unix administration creates strong network effects which influences the adoption decisions of end-users and is a type of complementary asset to the open-source program.
2. An open standard backed by a viable high-utility open-source software implementation creates a weak appropriability regime and strengthens the power of open standards. This limits the ability of a single organization to dominate a technology and lock-in users to their specific implementation.
3. The ability to modify or enhance the core software is a strength of the open-source model and signals to users the long term viability of the software. In the worst case, a user of open-source software has access to the source code and can continue to develop or modify the software as required.

Notwithstanding these observations, the software industry and particularly the Internet software industry is highly dynamic. Technological change is a constant and firms need to be mindful of this change. The emergence framework developed in the dominant design literature and explored in this study provides a basis from which a firm can make the necessary observations and linkages. Open standards and open-source software provide a powerful combination which proprietary executable-only software firms have to contend with. At the very least, the Apache web server has defined a market and increased the entry cost for firms to compete in the Internet server industry.

Future Research

In order to increase the understanding of the dominant design emergence framework and to improve its ability to be generalized, further case studies need to be conducted. This implementation of the framework is unique. Future case studies may have access to a wider range of data and could develop further qualitative and quantitative techniques for each of the factors. In the Internet software industry, other interesting studies could examine the email software market, the web browser market, or Internet server operating system market. All of these markets have both open-source and proprietary executable-only software available.

Appendix A
Open Source Definition

The hypertext version of this document can be found at

http://www.opensource.org/osd.html.

The Open Source Definition

(Version 1.7)

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution
The license may not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license may not require a royalty or other fee for such sale. (rationale)

2. Source Code
The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost -- preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed. (rationale)

3. Derived Works
The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software. (rationale)

4. Integrity of The Author's Source Code.
The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software. (rationale)

5. No Discrimination Against Persons or Groups.
The license must not discriminate against any person or group of persons. (rationale)

6. No Discrimination Against Fields of Endeavor.
The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research. (rationale)

7. Distribution of License.
The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties. (rationale)

8. License Must Not Be Specific to a Product.
The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution. (rationale)

9. License Must Not Contaminate Other Software.
The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software. (rationale)

Change history:
1.0 -- identical to DFSG, except for addition of MPL and QPL to clause 10.
1.1 -- added LGPL to clause 10.
1.2 -- added public-domain to clause 10.
1.3 -- retitled clause 10 and split off the license list, adding material on procedures.
1.4 -- Now explicit about source code requirement for PD software.
1.5 -- allow ``reasonable reproduction cost" to meet GPL terms.
1.6 -- Edited section 10; this material has moved.
1.7 -- Section 10 replaced with new "Conformance" section.

Conformance
(This section is not part of the Open Source Definition.)
We think the Open Source Definition captures what the great majority of the software community originally meant, and still mean, by the term "Open Source". However, the term has become widely used and its meaning has lost some precision. The OSI Certified mark is OSI's

way of certifying that the license under which the software is distributed conforms to the OSD; the generic term "Open Source" cannot provide that assurance, but we still encourage use of the term "Open Source" to mean conformance to the OSD. For information about the OSI Certified mark, and for a list of licenses that OSI has approved as conforming to the OSD, see this page.

Bruce Perens wrote the first draft of this document as `The Debian Free Software Guidelines', and refined it using the comments of the Debian developers in a month-long e-mail conference in June, 1997. He removed the Debian-specific references from the document to create the `Open Source Definition'.

Bibliography

Abernathy, W. J., & Clark, K. B. (1985). Innovation: Mapping the Winds of Creative Destruction. *Research Policy, 14*, 3-22.

Abernathy, W. J., & Utterback, J. M. (1978). Patterns of Industrial Innovation. *Technology Review, June-July*, 40-47.

Afuah, A. (1998). *Innovation Management: Strategies, Implementation and Profits*. New York: Oxford University Press.

Agarwal, R., & Prasad, J. (1997). The role of innovation characteristics and perceived voluntariness in the acceptance of information technologies. *Decision Sciences Journal, 28*(3), 557-582.

Agarwal, R., & Prasad, J. (1998). Antecedents and consequents of user perceptions in information technology adoption. *Decision Support Systems, 22*(1), 15-29.

Agarwal, R., Tanniru, M., & Wilemon, D. (1997). Assimilating Information Technology Innovations: Strategies and Moderating Influences. *IEEE Transactions on Engineering Management [IEEE TRANS ENG MANAGE], 44*(4), 347-358.

Ancona, D. G., & Caldwell, D. (1990). Beyond Boundary Spanning: Managing External Dependence in Product Development Teams. *The Journal of High Technology Management Research, 1*(2), 119-135.

Ancona, D. G., & Caldwell, D. (1992). Bridging the Boundary: External Activity and Performance in Organizational Teams. *Administrative Science Quarterly, 37*, 634-665.

Andreessen, M. (1992, Dec 1, 1992). *NCSA http server*, [Web]. NCSA. Available: http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1992.messages/382.html [2000, March 9].

Apache Software Foundation. (1998, June 22, 1998). *IBM joins the Apache Project, Plans to bundle and support the Apache HTTP Server*, [Website]. Apache Software Foundation,. Available: http://www.apache.org/press/22Jun98.html [2001, April 25].

Apache Software Foundation. (2001, February 1999). *The Apache Project*, [Website]. Apache Software Foundation,. Available: http://httpd.apache.org/ABOUT_APACHE.html [2001, April 25].

Arthur, W. B. (1987). Competing Technologies: An Overview. In G. Dosi (Ed.), *Technical Change and Economic Theory* (pp. 590-607). New York, New York: Columbia University Press.

Arthur, W. B. (1996). Increasing Returns and the New World of Business. *Harvard Business Review, 74*(4), 100-111.

Astebro, T. (1995). The effect of management and social interaction on the intra-firm diffusion of electronic mail systems. *IEEE Transactions on Engineering Management, 42*(4), 319-332.

Banks, R. G. (1996). *Altair History Site*, [Web]. Online Ethics Center for Engineering and Science. Available: http://www.onlineethics.org/text/contest/altair/main.html [1999, Sept 7, 1999].

Baum, J. A. C., Korn, H. J., & Kotha, S. (1995). Dominant Designs and Population Dynamics in Telecommunications Services: Founding and Failure of Facsimile Transmission Service Organizations, 1965-1992. *Social Science Research, 24*, 97-135.

Berners-Lee, T. . *WorldWideWeb*, [Web]. Tim Berners-Lee. Available: http://www.w3.org/People/Berners-Lee/WorldWideWeb.html [2000, March 9].

Berners-Lee, T. (1989). *Information Management: A Proposal*, [Web]. Tim Berners-Lee. Available: http://www.w3.org/History/1989/proposal.html [2000, March 9].

Besen, S. M., & Farrell, J. (1994). Choosing How to Compete: Strategies and Tactics in Standardization. *Journal of Economic Perspectives, 8*(2), 117-132.

Bradner, S. (1998). The Internet Engineering Task Force. In C. DiBona, S. Ockman, & M. Stone (Eds.), *Open Sources: Voices from the Open Source Revolution* (Vol. 1998, pp. 47-52). Sebastopol, CA: O'Reilly.

Brown, S. L., & Eisenhardt, K. M. (1995). Product Development: Past Research, Present Findings, and Future Directions. *Academy of Management Review, 20*(2), 343-378.

Brown, S. L., & Eisenhardt, K. M. (1997). The Art of Continuous Change: Linking Complexity Theory and Time-paced Evolution in Relentlessly Shifting Organizations. *Administrative Sciences Quarterly, 42*(1), 1-34.

Brynjolfsson, E., & Kemerer, C. F. (1996). Network externalities in microcomputer software: An econometric analysis of the spreadsheet market. *Management Science, 42*(12), 1627-1647.

Cailliau, R. (1995, November 2, 1995). *A Short History of the Web*, [Web]. CERN. Available: http://www.inria.fr/Actualites/Cailliau-fra.html [2000, March 9].

Chau, P. Y. K., & Tam, K. Y. (1997). Factors affecting the adoption of open systems: an exploratory study. *MIS Quarterly, 21*, 1-24.

Christensen, C. M., Suarez, F. F., & Utterback, J. M. (1998). Strategies for survival in fast-changing industries. *Management Science, 44*(12), S207-S220.

Church, J., & Gandal, N. (1992). Network Effects, Software Provision, and Standardization. *Journal of Industrial Economics, 40*(1), 85-103.

Cooper, R. B., & Zmud, R. W. (1990). Information technology implementation research: a technological diffusion approach. *Management Science, 36*(2), 123-39.

Cottrell, T. (1994). Fragmented standards and the development of Japan's microcomputer software industry. *Research Policy, 23*(2), 143-174.

Cottrell, T. (1997). Computer Platforms: Competition in Microcomputer Software. In L. Deutsch (Ed.), *Industry Studies* (pp. Chapter 13): M.E. Sharpe.

Cusumano, M. A., Mylonadis, Y., & Rosenbloom, R. S. (1992). Strategic maneuvering and mass-market dynamics: the triumph of VHS over Beta. *Business History Review, 66*, 51-94.

Cusumano, M. A., & Yoffie, D. B. (1998). *Competing on Internet time : lessons from Netscape and its battle with Microsoft*. New York, NY: Free Press.

David, P. A. (1985). Clio and the Economics of QWERTY. *Economic History, 75*(2), 332-337.

DiBona, T., Ockman, S., & Stone, M. (1998). Introduction. In T. DiBona, S. Ockman, & M. Stone (Eds.), *Open Sources: Voices from the Open Source Revolution* (Vol. 1998, pp. 1-16). Sebastopol, CA: O'Reilly.

Dos Santos, B. L., & Pfeffers, K. (1998). Competitor and vendor influence on the adoption of innovative applications in electronic commerce. *Information & Management, 34*(3), 175-184.

Dyson, E. (1998). Open Mind, Open Source. *Release 1.0, 1998*(November), 1-3.

editors@apacheweek.com. (2001). *ApacheWeek Website*. Red Hat Inc.,. Available: http://www.apacheweek.com/ [2001, April 25].

Eisenhardt, K. M., & Brown, S. L. (1998). Time pacing: Competing in markets that won't stand still. *Harvard Business Review, 76*(2), 59-69.

Encyclopædia Britannica Online. (2000). *Hypertext*, [Web]. Encyclopædia Britannica Online. Available: http://www.eb.com:180/bol/topic?eu=1637&sctn=1#s_top [2000, March 13].

E-Soft Inc. (2001). *Web Server Survey*. E-Soft Inc. Available: http://www.securityspace.com [2001, April 24].

Farrell, J., & Saloner, G. (1986). Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation. *American Economic Review, 76*(5), 940-955.

Farrell, J., & Saloner, G. (1987). Competition, Compatibility and Standards: The Economics of Horses, Penguins and Lemmings. In H. L. Gabel (Ed.), *Product standardization and competitive strategy* (pp. 1-21). Amsterdam ; New York

New York, N.Y., U.S.A.: North-Holland ;

Sole distributors for the U.S.A. and Canada Elsevier Science Pub. Co.

Farrell, J., & Saloner, G. (1992). Converters, Compatibility, and the Control of Interfaces. *Journal of Industrial Economics, 40*(1), 9-35.

Federal Networking Council. (1999, October 24, 1995). *Definition of "Internet"*, [Web]. "Federal Networking Council". Available: http://www.fnc.gov/Internet_res.html [1999, August 18].

Free Software Foundation. (1999). *Overview of the GNU Project*, [Web]. Free Software Foundation. Available: http://www.gnu.org/gnu/gnu-history.html [1999, Sept 7, 1999].

Gabel, H. L. (1987). *Product Standardization and Competitive Strategy*. Amsterdam ; New York

New York, N.Y., U.S.A.: North-Holland ;

Sole distributors for the U.S.A. and Canada Elsevier Science Pub. Co.

Gabel, H. L. (1991). *Competitive Strategies for Product Standards: The Strategic Use of Compatibility Standards for Competitive Advantage*. London ; New York: McGraw-Hill.

Gandal, N. (1995). Competing Compatibility Standards and Network Externalities in the PC Software Market. *Review of Economics and Statistics, 77*(4), 599-608.

Gates, B. (1976). *An Open Letter to Hobbyists*, [Web]. Microsoft Corporation. Available: http://www.microsoft.com/billgates/bio/1976.htm [1999, Sept 7, 1999].

Gopalakrishnan, S., & Damanpour, F. (1997). A Review of Innovation Research in Economics, Sociology and Technology Management. *Omega International Journal of Management Science, 25*(1), 15-28.

Grindley, P. (1995). *Standards, strategy, and policy : cases and stories*. Oxford [England] ; New York: Oxford University Press.

GVU's WWW Survey Team. (2001). *GVU WWW User Survey*. GVU Center, College of Computing, Georgia Institute of Technology. Available: http://www.gvu.gatech.edu/gvu/user_surveys/ [2001, April 27].

Hamel, G., & Prahalad, C. K. (1994). *Competing for the future*. Boston, Mass.: Harvard Business School Press.
Henderson, R. M. (1998). S-Curves, Dominant Designs and Technological Trajectories. *MIT 15.932 Course Notes*.
Henderson, R. M., & Clark, K. B. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly, 35*, 9-30.
Hergert, M. (1987). Technical Standards and Competition in the Microcomputer Industry. In H. L. Gabel (Ed.), *Product standardization and competitive strategy* (pp. 67-89). Amsterdam ; New York
New York, N.Y., U.S.A.: North-Holland ;
Sole distributors for the U.S.A. and Canada Elsevier Science Pub. Co.
Husain, Z., & Sushil. (1997). Strategic management of technology - A glimpse of literature. *International Journal of Technology Management, 14*(5), 539-578.
Iansiti, M., & MacCormack, A. (1997). *Living on Internet Time: Product Development at Yahoo!, NetDynamics, and Microsoft* (Teaching Case 9-697-052). Cambridge, Mass: Harvard Business School.
Katz, M. L., & Shapiro, C. (1985). Network Externalities, Competition, and Compatibility. *American Economic Review, 75*(3), 424-440.
Katz, M. L., & Shapiro, C. (1992). Product Introduction with Network Externalities. *Journal of Industrial Economics, 40*(1), 55-83.
Katz, M. L., & Shapiro, C. (1994). Systems Competition and Network Effects. *Journal of Economic Perspectives, 8*(2), 93-116.
LaRose, R., & Hoag, A. (1996). Organizational adoptions of the internet and the clustering of innovations. *Telematics and Informatics, 13*(1), 13-25.
Lee, J.-R., O'Neal, D. E., Pruett, M. W., & Thomas, H. (1995). Planning for dominance: a strategic perspective on the emergence of a dominant design. *R&D Management, 25*(1), 3-25.
Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., Postel, J., Roberts, L. G., & Wolff, S. (1998, 20 Feb 98). *A Brief History of the Internet* (3.1), [Web Document]. Internet Society. Available: http://www.isoc.org/internet/history/brief.html [1999, .
Leonard-Barton, D. (1994). *Commercializing Technology: Imaginative Understanding of User Needs* (Working Paper 9-694-102): Harvard Business School.
Liebowitz, S. J., & Margolis, S. E. (1994). Network Externality: An Uncommon Tragedy. *Journal of Economic Perspectives, 8*(2), 133-150.
Lind, M. R., Zmud, R. W., & Fischer, W. A. (1989). Microcomputer adoption - the impact of organizational size and structure. *Information & Management [INF. MANAGE.], 16*(3), 157-162.
Liu, C., Arnett, K. P., Capella, L. M., & Beatty, R. C. (1997). Web sites of the Fortune 500 companies: Facing customers through home pages. *Information & Management, 31*(6), 335-345.
Microsoft Corporation. (1999). *Protection Against Software Piracy*, [Web]. Microsoft Corporation. Available: http://www.microsoft.com/piracy/ [1999, Sept 7, 1999].
Netcraft. (2001). *Web Server Survey*. Netcraft. Available: http://www.netcraft.com/survey [2001, April 24].

Online Computer Library Center. (1999, September 9). *OCLC Office of Research: Web Characterization Project : Statistics*, [Online]. Online Computer Library Center,. Available: http://www.oclc.org/oclc/research/projects/webstats/statistics.htm [1999, September 15].

Open Source Initiative. (1999, no date). *History of the Open Source effort*, [Online]. Open Source Initiative. Available: http://www.opensource.org/history.html [1999, Sept 8].

Raymond, E. (1999). *Version and Change History of The Cathedral and the Bazaar*, [Web]. Eric Raymond. Available: http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar-16.html [1999, Sept 9, 1999].

Raynovich, R. S. (1999). *Wall Street likes that Linux Spirit* (Sept 9, 1999), [Web]. Redherring.com. Available: http://www.redherring.com/insider/1999/0909/news-linux.html [1999, Sept 12, 1999].

Roberts, R. (1998). Managing innovation: The pursuit of competitive advantage and the design of innovation intense environments. *Research Policy, 27*(2), 159-175.

Rogers, E. M. (1995). *Diffusion of innovations*. (4th ed.). New York: Free Press.

Romer, P. (1998). *In The Beginning was the Transistor*. Available: http://anasazi.umsl.edu/fin455/paul_romer.htm.

Rosenbloom, R. S., & Cusumano, M. A. (1987). Technological pioneering and competitive advantage: the birth of the VCR industry. *California Management Review, 29*, 51-76.

Rutkowski, A. M. (1995). Today's Cooperative Competitive Standards Environment and the Internet Standards-Making Model. In B. Kahin & J. Abbate (Eds.), *Standards policy for information infrastructure* (pp. xiv, 653). Cambridge, Ma: MIT Press.

Schilling, M. A. (1998). Technological lockout: An integrative model of the economic and strategic factors driving technology success and failure. *Academy of Management Review, 23*(2), 267-284.

Shapiro, C., & Varian, H. R. (1998). *Information rules : a strategic guide to the network economy*. Boston, Mass.: Harvard Business School Press.

Sheremata, W. A. (1997). Barriers to Innovation: A Monopoly, Network Externalities, and the Speed of Innovation. *Antitrust Bulletin, 42*(4), 937-972.

Smit, F. C., & Pistorius, C. W. I. (1998). Implications of the Dominant Design in Electronic Initiation Systems in the South African Mining Industry. *Technological Forecasting and Social Change, 59*, 255-274.

Stallman, R. (1998). The GNU Operating System and the Free Software Movement. In T. DiBona, S. Ockman, & M. Stone (Eds.), *Open Sources: Voices from the Open Source Revolution* (Vol. 1998, pp. 53-70). Sebastopol, CA: O'Reilly.

Suarez, F. F., & Utterback, J. M. (1995). Dominant designs and the survival of firms. *Strategic Management Journal, 16*(6), 415-430.

Teece, D. J. (1986). Profiting from Technological Innovation: Implications for Integration, Collaboration, Licensing and Public Policy. *Research Policy, 15*(6), 285-305.

Teo, T. S. H., & Tan, M. (1998). An empirical study of adoptors and non-adopters of the Internet in Singapore. *Information & Management, 34*(5), 34-40.

Turau, V. (1998). The Fortune 500 Web. *Computer, 31*(11), 119-120.

Tushman, M. L., & Anderson, P. (1986). Technological Discontinuities and Organizational Environments. *Administrative Science Quarterly, 31*(2), 439-465.

Tushman, M. L., & Murmann, J. P. (1998). *Dominant Designs, Technology Cycles, and Organizational Outcomes.* Paper presented at the Academy of Management Meeting, San Diego.

Tushman, M. L., & Rosenkoph, L. (1992). Organizational Determinants of Technological Change: Toward a Sociology of Technological Evolution. *Research in Organizational Behaviour, 14*, 311-347.

Van de Ven, A. H., & Rogers, E. M. (1988). Innovation and Organizations: Critical Perspectives. *Communication Research, 15*(5), 632-651.

von Hippel, E. (1987). Cooperation between rivals: Informal know-how trading. *Research Policy, 16*, 291-302.

World Wide Web Consortium (W3C). (1995a, November 28, 1995). *Change History for httpd*. W3C. Available: http://www.w3.org/Daemon/ [2000, March 8].

World Wide Web Consortium (W3C). (1995b). *A Little History of the World Wide Web*. W3C. Available: http://www.w3c.org/history [2000, March 8].

World Wide Web Consortium (W3C). (2000, March 01, 2000). *About The World Wide Web Consortium*. W3C. Available: http://www.w3.org/Consortium/ [2000, March 13].

Zaltman, G., Duncan, R., & Holbek, J. (1973). *Innovations and organizations*. New York: Wiley.