0 + 0 = 1: The Appliance Model of Selling Software Bundled with Hardware

by

Bettina Hein

lic. oec. HSG
University of St.Gallen, Switzerland, 1998

ref.iur.
University of Constance, Germany, 2000

Submitted to the MIT Sloan Fellows Program in Innovation and Global Leadership
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Management of Technology

At the

Massachusetts Institute of Technology

June 2007

Signature of Author______________________________

Bettina Hein
MIT Sloan School of Management
May 11, 2007

Certified by______________________________

Michael A. Cusumano
Thesis Supervisor
Sloan Management Review Distinguished Professor of Management

Accepted by______________________________

Stephen Sacca
Program Director
MIT Sloan Fellows Program in Innovation and Global Leadership

0 + 0 = 1: The Appliance Model of Selling Software Bundled with Hardware

by

Bettina Hein

Submitted to the MIT Sloan Fellows Program in Innovation and Global Leadership
on May 11, 2007 in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Management of Technology

Abstract

The business model of selling software bundled with hardware is called the appliance model. As hardware becomes less and less expensive and open source software is being offered for free, the traditional business model of selling packaged software is being threatened. This disruption in the software industry is forcing software vendors to consider other business models such as advertising-based, transaction-based, software-as-a-service or appliance-based models in order to create additional value for customers. Most of these models have existed in variants for decades but are now gaining in popularity due to factors such as changing cost structures or the Internet as a delivery channel.

This thesis analyzes the economic drivers and barriers for the appliance model for both the consumer and enterprise software industry segments. Important drivers of the appliance model for both of these sectors are hardware commoditization, open source software and vertical integration in order to capture margins. In the enterprise software segment the complexity of traditional software integration and operation including unpredictability of total cost of ownership, rising IT personnel cost and maintenance fees are driving the adoption of the appliance model. In the consumer software segment, ease of use, limited battery life, disintermediated distribution and prestige are important economic factors. The appliance model also has a number of economic disadvantages that hinder its adoption. Among these are the additional competencies that a company needs to build, supply chain and distribution costs, as well as inflexibility and inconvenience for the customer. Decision criteria for companies considering adopting the appliance model are also discussed.

Industry examples in the consumer and enterprise software segment are examined and specific companies are used as case studies. Among these are the digital music player, digital video recorder, router and enterprise search markets.

The possible implementation paths for software companies transitioning to the appliance model are proposed and the virtual appliance model as a next adoption step is discussed.

Thesis Advisor: Michael A. Cusumano

Title: Sloan Management Review Distinguished Professor of Management

Acknowledgments

Foremost, I'd like to thank my thesis advisor Professor Michael Cusumano. Without his classes "Managing Technological Innovation and Entrepreneurship" and "The Software Business" my interest in this topic would not have been sparked. I would also like to thank him for his valuable comments and suggestions while I was writing this thesis.

I'm also grateful to my teammates Andreas Göldi, Paul Hsu, Caleb Hug and Tim Jones in "The Software Business" class for working with me on a team paper that prepared some of the case studies and analysis used in parts of this thesis. I also had many valuable conversations with other classmates in the same course that influenced my thinking on this topic.

I want to thank the entrepreneurs I interviewed on the appliance model for their time and help: Peter-Frans Pauwels, co-founder and CTO of TomTom NV, Todd Mozer, founder and CEO of Sensory as well as Bill Liao, founder and CEO of Finaxis AG.

My husband Andreas has been an invaluable resource for me. He was a diligent editor and excellent technical knowledge base. Discussions with him and his patience have helped me more than anything.

This thesis is dedicated to my grandmother, Maria Neveling, who knows little about computers or the appliance model but she taught me what strength of will and perseverance are.

Table of Contents

1 Introduction

1.1 How will the software industry survive the next decade?

The software industry is currently in a phase of disruption. After more than three decades of exploiting what investors call the "beauty of the software" model, i.e. the 99% gross margins of software licenses, buyers of software have caught up with this "scheme". They know that software can be duplicated infinitely without incurring any incremental cost (Snyder 2006). The open source movement and the Internet have given them access software like never before. Consumers and corporate buyers are coming back to a notion common in the 1960s: software is free.

Hardware manufacturers are struck by the same problem. As Moore's Law pushes computer performance higher and higher, commoditization and fierce competition have continually lowered the prices hardware manufacturers can command for their products.

At the same time, consumers and corporate buyers have had several decades to experience the real cost of software. Installing, customizing and learning to use software is a task commonly dreaded by both the CIO migrating thousands of users to new ERP (Enterprise Resource Planning) software and the home PC user installing the latest 3-D gaming software alike.

So as the software industry struggles with these phenomena, new (or newly reinvigorated) business models are emerging that try to salvage profitability. Selling software as a service based on subscription fees or advertising-based software usage are models companies are experimenting with. This thesis will focus on another new model: selling software bundled with hardware – referred to here as the appliance model.

In the consumer business, market success stories of products such as Apple's iPod or TomTom's personal navigation device have shown that bundling hardware with software can be very profitable. In the enterprise software market, firewall appliances, email appliances and, most recently, Google's search appliance have made a case that the software business may be returning to one of its original business models, i.e. selling dedicated hardware and software together.

By bringing two components back together that have separately eroded their business models by commoditization, software companies may be able to re-create the value lost to commoditization. Thus, the title of this thesis: 0 + 0 = 1.

1.2 Research Question

This thesis attempts to analyze an anecdotal trend in the software industry towards an appliance business model of packaging software with dedicated hardware. It addresses the following questions:

- What are the economic factors that explain the (re-)emergence of the appliance model?
- What kind of software companies should adopt this new business model?

- Once a software company has decided to pursue this business model, what hurdles does it face? How can companies structure the transition to the appliance model?

1.3 Research Methods

This thesis attempts to answer the research questions above by surveying the available academic literature, interviewing executives of companies that use the appliance model, assembling case studies of companies and products and analyzing publicly available data on this subject.

1.4 Outline

This thesis consists of three major sections:

- a framework of the appliance model describing economic drivers and barriers
- industry examples of the appliance model
- an implementation model for companies that are going to adopt the appliance model

After this introductory chapter, *chapter two* sets the foundation by briefly describing the history of selling software as well as current business model trends in the industry. *Chapter three* focuses on the appliance model. It describes the economic drivers behind the model as well as a framework of decision criteria for the adoption. *Chapter four* describes industry examples of the appliance model and how the framework of economic drivers applies to those examples. The examples are taken from the consumer was well as the enterprise software industries. *Chapter five* shows how companies can transition from selling software to offering an appliance. *Chapter six* then looks into the future of this business model by describing the virtual appliance model. Finally, *chapter seven* offers a summary and conclusion while making suggestions for further research on the topic.

2 Selling Software

2.1 Historical Development of Selling Software

Software has a short history, spanning barely sixty years. Until the 1970s, the notion of selling standardized software separately from the computers they ran on was an alien concept to most people in the computer industry.

Hardware is dumb

In the 1950s, only few computers existed worldwide. Most of these were in university or military research labs such as the ENIAC at the University of Pennsylvania, the Harvard Mark II or the MIT Whirlwind. They were called "giant brains" by the general population and were housed in rooms the size of tennis courts. In the public mind, computers were wondrous machines that magically performed tasks. But, as seems obvious to us today, no computer could function without receiving instructions. In those days, computers received their instructions by feeding punched cards into a reader. As Paul Niquette, who claims to have coined the word 'software', explained to his early audiences: "Without its 'software', a giant brain is less intelligent than a giant gall bladder." (Niquette 2006)

Unbundling of software and hardware

The awareness that the instructions and the machine where not inseparable came about only gradually as computer scientists started using instructions designed for one computer on another. Only in 1960 did the Oxford English Dictionary first include the term software as it pertains to computer programming. (Niquette 2006)

It took almost another two decades from the realization that making software was an activity separate from designing and operating hardware to the mainstream practice of selling standard software separately. While early pioneers such as Applied Data Research began selling a software product in the early 1960s (Cusumano 2004) only IBM's announcement in 1968 to unbundle its software from the mainframe hardware launched a large-scale industry that produced standard software. (Campbell-Kelly 2003) The personal computer (PC) revolution of the 1980s exponentially grew the mass market for standard packaged software through an entrepreneurial wave of consumer software, making the concept of software accessible and tangible to general public. (Cusumano 2004)

Up-front license model for packaged software

For the three decades between 1970 and 2000, most software companies have operated on the stable business model of selling packaged software products. This means selling perpetual licenses to users allowing them to install the software product locally on general-purpose machines (mainframes, servers, stand-alone PCs, Personal Digital Assistants (PDAs) etc.) and often combining the perpetual up-front license with an annual maintenance fee in return for software updates. Other companies sell services associated with software (custom software, adaptation of standard software). Some companies combine these models in a hybrid fashion (Cusumano 2004).

The dot-com crash of 2001 and the subsequent crisis that followed in the entire IT industry (Vincent et al. 2005) made evident that the trend of ever-growing IT and software expenditures might not continue.

2.2 Current Disruption in the Software Business Model

Disruption in technology-dominated industries has been described as the process of a new entrant technology or business model, initially perceived as inferior and therefore not threatening to high-margin business, overtaking the incumbent technology or model in the industry (Christensen 1997).

In the 1990s, enterprise software salespeople could strike riches by selling expensive per-seat licenses to corporate customers. Lucrative recurring revenues were achieved by selling annual maintenance packages at 15-25% of the up-front license price (Nayak 2006).

Today, companies and consumers have become more reluctant in purchasing expensive software due to the high costs associated with software ownership. This has led to disruption in the software industry. A recent report by Accenture describes the crisis in the software business as triggered by three factors (Vincent et al. 2005):

- *"The "good enough" crisis: A situation in which product-based differentiation is no longer rewarded, thus triggering the maturation of every product category. Ten years after the Internet, and 20 years after client/server and the PC, many enterprise software segments have been hit by the "good enough" crisis.*
- *The "IT does not matter" crisis: A general disillusionment with IT that was captured by Nicholas Carr's May 2003 Harvard Business Review article and produced a conservative spending environment for the past four years.*
- *The "complexity" crisis: A desire for simplification and reluctance to introduce new technologies in an enterprise environment already struggling to deal with the legacy of decades of IT experimentation (a term coined by research firm IDC)."*

While some established companies can afford to coast for years if not decades on maintenance fees and service revenue, new or growth-oriented established companies in the software industry have to find a different way to profitably operate in this new customer environment. The next section examines some of the new or newly invigorated business models that companies are using to cope with the disruption in the software industry.

2.3 New Business Models for the Software Industry

Business models in the software industry consist of a number of choices the vendor makes in several business dimensions. The three most important dimensions are (Göldi et al. 2006b):

- Revenue model: In what fashion do companies charge for the usage of the software? Examples are up-front license fees, recurring subscription fees, recurring maintenance fees, advertising-based usage, free usage with charges for services.

- Delivery model: How does the customer access the software? Examples are access on local clients or servers, remote hosting (proprietary or web-based), bundled with hardware.
- Customers targeted: What segment of customers should use the software? Examples are small businesses, early-adopter enterprises, mainstream enterprise customers, early-adopter consumers etc. Often the location of the customer on the technology adoption curve (Moore 1991) is a factor in defining the target customers. The target customers influence important decisions such as the distribution channel or the pricing structure.

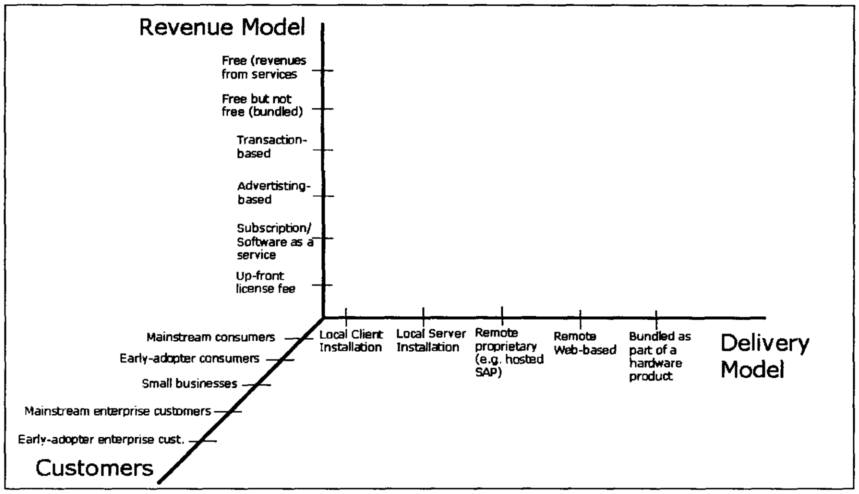

Exhibit 1: Dimensions of software business models

A business model chooses one or several variants along each dimension. For example, the most common business model in software, the upfront-license fee for packaged standard software, has long entailed not only the up-front component but also the installation on the local client or local server and, dependent on the software vendor, usually a target customer segment. The recombination of variants can lead to new business models.

In choosing from the dimensions above the following new (or simply newly revisited, in some cases) business models are currently gaining popularity in the software industry:

Software as a Service (SaaS)

Software as a Service is currently defined as a business model that separates the ownership of software from its use (Turner, Budgen D., and Brereton 2003). The customer accesses the software remotely via the Internet and generally pays a per-user subscription fee (Traudt and Konary June 2005). SaaS has gained popularity as broadband Internet access has become commonplace in companies and consumers' homes. Its advantages include the ease of use of Internet browsers, low initial purchase barriers (pay-as-you-go model) and lower IT personnel cost for the customer. Disadvantages such as security or privacy concerns due to remote hosting and the difficulty of disconnected use (for mobile users) are hampering the adoption of this business model (Göldi et al. 2006a). Successful examples of companies using this business model include salesforce.com (web-based customer relationship management software), Siebel On-Demand (web-based customer relationship management software) and 37signals (project management, group writing) (Göldi 2007).

Advertising-based model

Advertising-based business models for software have their origins in the traditional media businesses such as newspapers and television. The delivery model is generally remote via the Internet. Advertisers pay to have their messages shown to the software user for example in form of banner ads or as query-based paid placement. This model depends on either large user volume or very specialized and therefore targetable viewers. (Rappa 2007). Advantages of this model are that users can often use the software for free and therefore, barriers to user acquisition in the consumer market are low. Disadvantages are that corporate users do not like their employees to be distracted by advertising at work, strongly limiting the use of this business model for enterprise applications. The market for online advertising is currently estimated at $15 billion (2006) with growth rates of more than 35% per year (Kessler 2006). The most well-known example for an advertising-based model is Google, not only with search but also other applications such as Gmail or Google Calendar, Google Docs & Spreadsheets (Google Inc. 2007c). While some companies like Austin-based Spiceworks (IT management software) have started to enter into the enterprise space with advertising-supported business models, no clearly successful examples have emerged to date.

Transaction-based model

Transaction-based business models in the software industry are not new. This model involves an incremental fee for each function performed. This is also referred to as "pay-per use". The advantages of this model include a steady revenue stream for the software vendor and greater control over the use of the software. Traditionally, the delivery model has involved a hosted service. In the 1960s and 1970s, time-sharing services offered software on a per-use basis. The advent of the personal computer made time-sharing and thereby its business model obsolete. Until recently, the customer segments where this business model survived were high volume businesses such as airlines or financial services. One example is the airline reservation system SABRE, which was launched in 1964 after 10 years of planning. SABRE charges a fee per flight booking. Another example is Automatic Data Processing (ADP), which has always charged transaction fees for its payroll-processing service. ADP was founded in 1949 and began using computers for payroll services in 1961(Campbell-Kelly 2003).

Today, the new delivery channel of the Internet has made transaction-based business models viable again. Consumers and small companies can access usage-based software via the Internet that they would otherwise not purchase due to their high purchase or installation cost, complexity or requirement of a network effect. Examples include EBay (auction platform software), Amazon Web Services (historical pricing, human-intelligence services, e-commerce services) and WebEx (conferencing software). An interesting new development in the pay-per-use model involves software-powered utility-computing platforms such as Amazon's Compute Cloud (Amazon Inc. 2007a) or 3TERA (3TERA Inc. 2007). This appears almost like a return to the time-sharing business model where companies like Tymshare were more interested in selling computer time than in the software provided (Campbell-Kelly 2003).

Appliance model

The appliance business model involves selling software bundled with hardware. Again, the basic notion of this model is not new. As described in Section 2.1, before the packaged standard software model became prevalent, software was always sold together with hardware. Often, this meant that the bundle functioned like an appliance in the respect that it was only intended to perform one or few specified tasks. The software was usually custom-programmed to fit the specific business need of the computer purchaser. When the number of computers grew exponentially in the 1960s, the shortage of computer programmers led to more and more reuse by packaging programs into software products (Campbell-Kelly 2003). Today, commoditization of hardware and the availability of open source software have created opportunities for software vendors to package their software with dedicated hardware and command a premium for this offering. This "hardwarization" of software can be found in the consumer as well enterprise software segment. While technically, as described below in Section 3.2, an appliance refers to packaging all necessary elements of a solution stack into a coherent product, the actual economic model is less well-defined. The following common variants can be observed in the market:

- Subscription service

  This variant differs from Software as a Service only in the delivery model. Here, the application vendor delivers all service and maintenance to the customer in a pay-as-you-go fashion. This model allows the customer to save costs of managing multiple maintenance streams, licenses and service contracts (Wikipedia 2007).

- One-time fee

  This variant is often used in the consumer segment. The software is packaged into a ready-to-use device (e.g. Apple's iPod) and the customer only has to pay once to use the products basic functionality. Appliance companies try to achieve recurring revenue by selling upgrades or additional functionality to customer after the initial sale (e.g. TomTom's traffic services or Apple's iTunes Store). In the enterprise segment, the one-time fee is often scaled by number of users (e.g. Mirapoint's email appliance), number of documents (e.g. Google's Search Appliance) or other units of measuring the size of the computing task.

Hybrid variants can include a one-time fee combined with an update service or maintenance.

Common aspects of the new models

Overall, these new or newly invigorated business models have in common that they offer customers specific advantages that the currently still dominant up-front license model does not always offer. These advantages include:

- Ease of use

- IT personnel cost savings (no installation, less customization)
- Lower or no cost (in the case of open source software)
- Reduction of complexity

The essence of these new business models is the customer's combination of the unwillingness to pay for software coupled with the willingness to pay for the alleviation of current discomforts of software use. By providing their customers with a better offering, companies employing these business models are seeking out a competitive advantage over their established competitors that still employ the standard upfront license model.

The next chapter will examine closely the appliance model as one of the business models promising a competitive advantage and determine what factors could make it desirable (or not) for companies to choose this model.

3 The Appliance Model

3.1 Definition: Appliance

As little scholarly literature as been published on the topic of appliances in the software industry, a standard definition is not readily available.

The Oxford Dictionary of English defines appliance as "a device designed to perform a specific task" (Soanes and Stevenson 2003). While this definition can be used to describe a toaster as well as a washing machine, it also serves well as a guideline for the definition of an appliance in the software industry.

General definition

The Computer Desktop Encyclopedia lists the following definition (Freedman and Morrison 2007):

> *"A device that is dedicated to a specific function in contrast to a general-purpose computer. Many consider the router the first network appliance."*

The research firm Gartner offers this definition (MacDonald et al. 2006):

> *"A computing entity that delivers predefined service(s) through an application-specific interface, with no accessible operating software."*

Gartner specifies that their definition of an appliance includes that it is *not* a general-purpose device. Instead, it is a focused and partly inflexible device that reduces complexity by hiding underlying operating software, not allowing users or administrators to add arbitrary software.

For the purpose of this thesis I will use the following general definition:

An appliance is a *dedicated physical or virtual unit* that *houses software to perform a specific computing task*. An appliance is to be differentiated from general-purpose computing platforms that accommodate several types of software performing a variety of tasks. In an appliance, the task to be performed does not have to share computing (or battery, for mobile devices) resources with other applications.

Subcategories

In technical journals, trade press and product literature one finds subcategories of the appliance definition such as (MacDonald et al. 2006; Novell Inc. 2007):

- Hardware appliance

  A hardware appliance includes one or several pieces of physical hardware bundled with a software appliance. The hardware may be customized to or optimized for the application.

- Software appliance

  A software appliance contains a specific software application combined with a tailored solution stack including an operating system, data management and a programming environment. It can be easily installed on standard hardware or a virtual machine.

- Virtual appliance

  A virtual appliance includes all elements of a software appliance but it additionally simulates the necessary hardware by running the software appliance on a virtual host operating system.

3.2 Technical Concepts

In technical terms, the appliance model completely integrates various hierarchy layers of a computer hardware and software solution into a single package for the end user. The hierarchy layers needed to implement a fully functional computing solution are also referred to as a solution stack (Freedman and Morrison 2007).

A simplified solution stack consists of the following components (Göldi et al. 2006a):

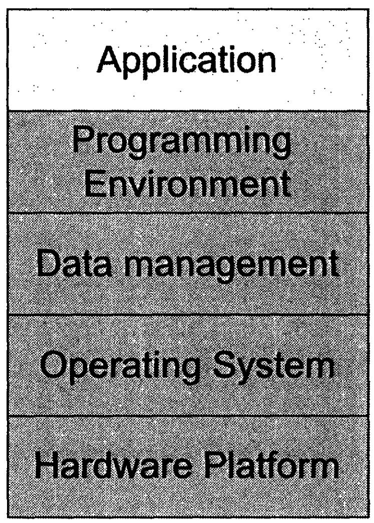

Figure 1: Simplified solution stack

Hardware platform

The hardware platform is the physical basis of the solution stack. Examples of a hardware platform are general-purpose personal computers, mobile devices such as PDAs (personal digital assistants) or specialized hardware devices.

Operating system

The basic hardware functions are controlled by an operating system (OS). An OS is the "software component of a computer system that is responsible for the management and coordination of activities and the sharing of the resources of the computer." (Parker 2004) A solution stack can use a standard OS such as Windows or Linux, or an application-specific OS.

Data management

A data management layer includes the part of the stack that manages the physical storage and retrieval of data but also software that makes it possible to interactively create, store and change files (Freedman and Morrison 2007). For example, network-oriented file management systems or database management systems are part of this layer.

Programming environment

Software developers can implement the desired functionality of a program through a programming environment. This environment contains application programming interfaces (APIs). APIs are a "language and message format used by an application program to communicate with the operating system or some other control program such as a database management system (DBMS) or communications protocol" (Freedman and Morrison 2007). The APIs therefore link the programmer to the lower-level functionalities of the stack hierarchy. This layer creates a run-time environment for the controlled execution of applications. Examples are the Java environment (available in several different, application-specific versions such as J2EE) and Microsoft's .NET environment.

Application

At the top of the hierarchy is the application, i.e. the software program that delivers the intended end-user functionality.

3.2.1 Packaged Standard Software

The currently prevalent packaged standard software model supplies only the application layer to the end customer. The customer then installs the application software on a defined general-purpose stack where it normally shares resources with other applications (Göldi et al. 2006a).

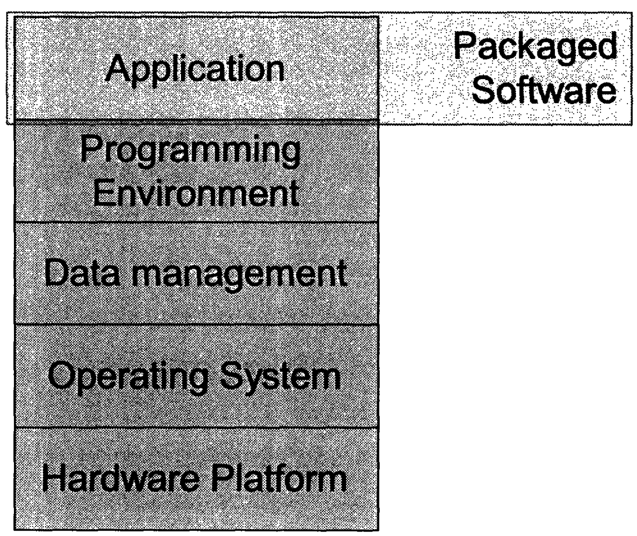

Figure 2: Packaged software in the solution stack

As described in Chapter 2 packaged software became prevalent through IBM's unbundling decision and the standardization achieved through the PC revolution with the dominance of a Windows/Intel stack. In theory, this allowed programmers to concentrate on providing a solution for a business need without worrying about lower-level functionality. Customers have the advantage that they only need one defined lower-level stack (such as an Intel processor-driven PC with Microsoft Windows OS) and can run several applications on it. This created efficiencies that were previously not available to owners of customized hardware/software bundles.

While in theory large-scale efficiencies should be possible and sustainable, competition in all layers of the stack has created multiple hurdles for their full exploitation.

Competition between stacks

Concentrating solely on the business application has been hampered by the existence of several competing and incompatible stacks (e.g. Microsoft Windows and Apple Macintosh in desktop computing; Windows CE/Mobile and Symbian in mobile phones). Software companies that want to appeal to a broad set of customers have to support several stacks. The efficiency gained by having a common stack can easily be lost by having to port to several platform stacks or by relinquishing the market segment that uses an incompatible stack. It is estimated that software companies spend 25-40% of research and development expenses on non-differentiating maintenance, testing and quality assurance associated with supporting multiple platform stacks (Marshall 2006a). Customers also lose efficiency because they are limited in their software choice by the stack they have purchased.

Incompatibilities within stacks

Competition has also lead to incompatibilities within the stacks, thereby increasing complexity further for software vendors. For example, certain application programs need a specific version of a programming environment, database management software or even specific hardware. This is why shrink-wrapped software often specifies on the packaging requirements for the stack (e.g. available memory, processor speed, graphics cards). For instance, in game software specific graphics cards are often needed to run the software. If such a card is not present, the application will not run. The customer is usually responsible for integrating and configuring the elements, often leading to considerable additional costs. Even if there is no deviation from the currently most common stack, software companies and their customers may find their application software malfunctioning because of an unexpected interaction between different application software installed on the same stack.

Performance issues

Because several applications are running on the same platform stack, users may encounter performance issues. Even if users run only one application on a given general-purpose platform, the overhead of modern programming environments and operating systems is considerable. Today, a typical PC can easily spend more CPU time supporting the stack than actually running applications. Also, for mobile applications battery performance still limits the usefulness of a general-purpose stack.

The customer therefore ends up with a compromise: limitations in the selection of available applications, integration problems, and a less-than-perfect performance are the price for the economic efficiencies of standardized stacks (Göldi et al. 2006a).

Scalability issues

When an application has reached a certain capacity it can to prone to failure. Unless completely redundant systems are built, this can lead to a single point-of-failure. To prevent failure, near-capacity applications have to be scaled up. To scale an application to more users or a greater number of transactions, conventional packaged software has to be integrated on another host. For enterprise solutions this causes difficulties because business processes may have to be interrupted.

3.2.2 Appliance Model

The appliance model capitalizes on these known difficulties by combining all necessary stack elements into an integrated product.

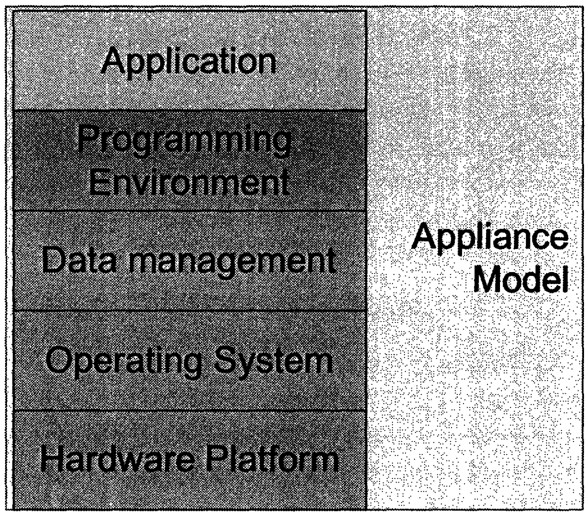

Figure 3: Solution stack in the appliance model

The vendor (usually, but not always, the software application company) chooses all components of the stack in order to optimize the solution. For example, in consumer electronics products, the vendor can optimize screen size, buttons or battery options that best fit their product (cf. Chapter 4 industry examples). In enterprise software, this can mean that the hardware is optimized to deliver exactly the performance promised (e.g. search query performed in 10ms). The customer receives a "ready-to-go" solution that requires no installation or integration. Ideally, only a few configuration options are available to the end user (consumer or corporate IT department). This configuration should require no knowledge of the lower-level implementation.

The lower levels of the appliance model stack can be custom-designed (i.e. custom chips, specially designed molded plastic etc.) but they don't have to be. In fact, most appliances use a standard hardware, OS, programming environment and data management component that is then optimized or configured for the specific application. For example, for Internet application servers, a standard stack with the acronym LAMP has become popular. LAMP stands for (Dougherty 2001):

- Linux, the operating system
- Apache, the web server
- MySQL, the database management system (or database server) and

- PHP (or Perl or Python), the programming language.

The popularity of this stack can be explained by the low or zero purchasing cost because these are all open source programs. Also, Apache, MySQL and PHP are bundled into most Linux distributions, creating easy access to them.

Practical implementation

In the consumer segment, appliances can take many shapes and forms, ranging from the tiny iPod nano MP3 player to large set-top boxes like Digital Video Recorders (DVRs).

In the enterprise segment, most appliances come in the form of a server with an x86 Intel processor that is stackable in a rack configuration. This also called the "pizza box" form factor (cf. Figure 4). The individual appliances can run different applications, can be individually exchanged as needed, and can be rapidly reconfigured simply by attaching the appliances to different IP addresses. To scale up capacity for most appliances, one can easily add additional "pizza boxes" to the rack.

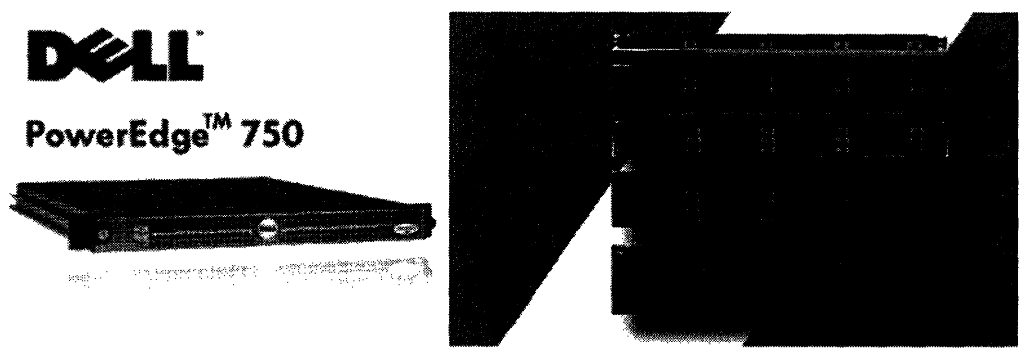

Figure 4: "Pizza box" form factor for corporate servers (Source: Dell Corporation)

The configuration and management is done via a simple graphical user interface (GUI) that allows for little customization. The example below shows the Google Search Appliance's GUI.

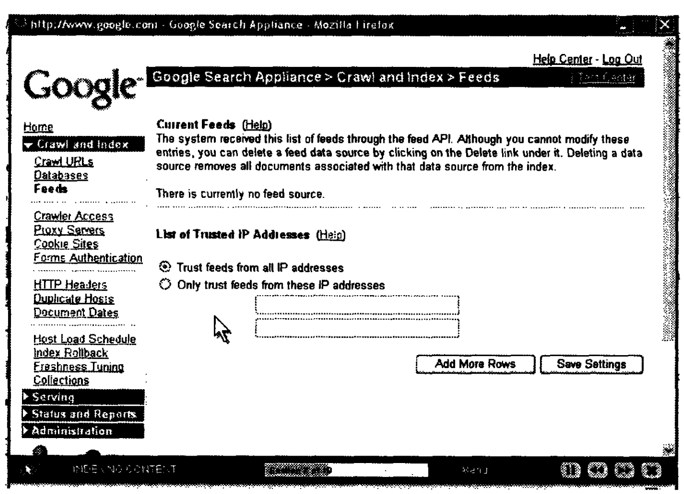

Figure 5: Google Search Appliance's configuration GUI

For well-designed appliances, the most important technical differences between packaged software and hardware appliances are summarized in the table below (8e6 Technologies 2005). It must be noted, though, that not all products marketed under the "appliance" label have all the characteristics listed (MacDonald et al. 2006).

Device Type	**Hardware Appliance**	**Packaged Software**
Platform	Single-purpose	Multi-purpose
CPU	Dedicated	Shared
Memory	Dedicated	Shared
Other processing resources	Dedicated	Shared
Potential bottleneck	No	Yes
Enterprise segment only		
Single point-of-failure	No	Yes
Scalability	Simple, add another appliance	Difficult, integrate software on another host
Configuration	Simple; via dedicated GUI	Complex
Management	Simple; via dedicated GUI	Difficult

3.2.3 Virtual Appliance Model

The appliance model itself is evolving and for some applications, the virtual appliance model may be the next logical step.

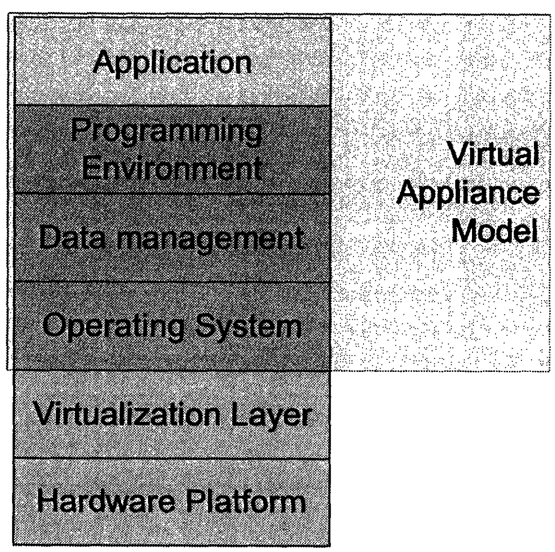

Figure 6: Solution stack for virtual appliances

In a virtual appliance all layers of the solution stack are provided, with the exception of the hardware. To simulate the presence of dedicated hardware, a "virtualization layer" is introduced (Göldi et al. 2006a). The virtualization layer creates a standard interface to the layers above so that the operating system functions as if it were running on a specific industry-standard hardware, e.g. an x86 PC. With the rise of utility computing, virtual appliances could run on a large mainframe just as well as on a single server located anywhere. The virtual appliance therefore allows for a fully pre-installed and pre-configured application and operating system environment with all the advantages of eliminating the installation, configuration and maintenance costs associated with running complex stacks of software. In addition, the company selling the virtual appliance is still able to control all of the software components of the solution stack.

3.3 Economic Drivers for the Appliance Model

For the appliance model to be viable and become a mainstream business model in the software industry, there has to be a strong economic rationale behind it. In this section, the economic drivers of the appliance model will be analyzed.

From a top-level perspective, it can be hypothesized that the economic value created by a bundle of software and hardware has been turned on its head. In the 1960s, software was considered an add-on necessary to sell hardware. (Campbell-Kelly 2003; Cusumano 2004) Today, the rationale behind bundling hardware with software comes from the opposite side: hardware is a necessary add-on to sell software.

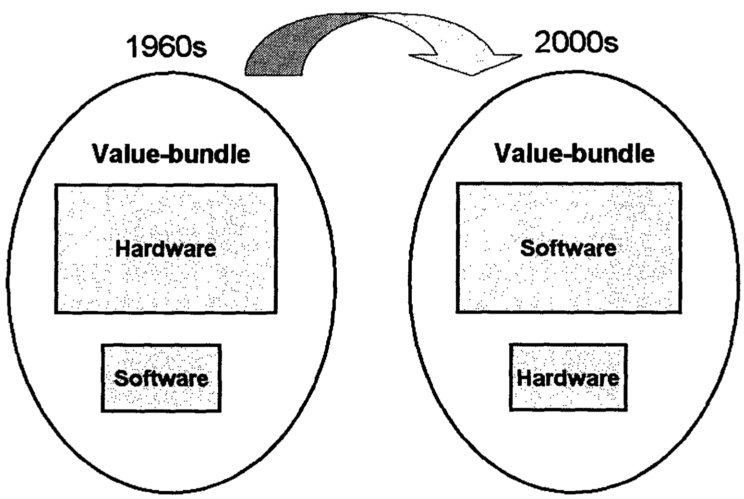

Figure 7: Economic value creation turned on its head

3.3.1 Example Enterprise Segment

How strongly the economic drivers in the enterprise software space have changed only in the past ten years can be shown by a simple hypothetical pricing example in the enterprise software segment (Göldi et al. 2006a). The example assumes that a software firm sells a proprietary enterprise application for $30,000. The necessary solution stack ten years ago is described in the "1997" column.

	1997			2007		
	Description	Approx. cost	Percentage	Description	Approx. cost	Percentage
Initial investment						
Application	proprietary	$ 30'000	30%	proprietary	$ 30'000	86%
Programming Environment	BEA or similar	$ 15'000	15%	Tomcat	$ -	0%
Data Management	Oracle or similar	$ 15'000	15%	MySQL	$ -	0%
Operating System	Unix	$ 5'000	5%	Linux	$ -	0%
Hardware Platform	High-end Unix server	$ 20'000	20%	x86 server	$ 5'000	14%
Service costs for integration	Systems integrator	$ 15'000	15%	none (pre-integrated)	$ -	0%
Total initial investment		**$ 100'000**			**$ 35'000**	
Recurring cost (yearly)						
Personnel cost	Systems administrator	$ 50'000	82%	IT manager (configuration)	$ 5'000	50%
Maintenance fees	entire stack, 17%	$ 11'050	18%	only application, 17%	$ 5'100	50%
Total recurring cost		**$ 61'050**			**$ 10'100**	

Figure 8: Hypothetical pricing example for enterprise software solution

1997: proprietary enterprise software solution

A company in 1997 would have had to buy all of the elements of the solution stack described in Section 3.2. In the example shown in the left column above, a high-end Unix server with Unix OS is complemented by an Oracle database. The proprietary application could then have been built on a powerful application server programming environment such as BEA WebLogic. The proprietary application would then have been integrated with this stack by a systems integrator. Although this example shows a fixed cost of integration, in practice systems integrators fees can seldom be estimated accurately. To continuously operate the application, each year the salary for a systems administrator plus a yearly maintenance fee on the entire software stack would have added to the cost. The result would have been a very feature-rich platform with high performance but also a high initial investment with additional recurring costs. While 70% of the initial cost was being spent on the non-application part, every year another 60% of the initial cost would have to be invested to keep the system running. Therefore, a customer would want to reuse this expensive infrastructure for additional applications. The cost structure 10 years ago thus advantaged a packaged software business model.

2007: enterprise solution appliance

Today, an inexpensive x86 server can deliver the same performance as the high-end Unix server in 1997. Free open source software such as Linux and MySQL has replaced the proprietary software on all basic levels of the stack. With the appliance model, no systems integrator is needed and a simple interface for configuration requires no full-time systems administrator. While a maintenance fee might still apply for the proprietary software (depending on the exact business model variant), the maintenance for the rest of the stack is included. The total value of the entire solution now can be dominated by the proprietary software (86% of initial cost vs. 30% of initial cost in 1997). By vertical integration, the software vendor can capture more of the value created. This shift in value creation has made the appliance model economically feasible and attractive for both the vendor and the customer.

3.3.2 Example Consumer Segment

To illustrate the economic drivers of the appliance model in the consumer space from a user's perspective, the personal digital assistant as a multi-purpose mobile computing platform will be contrasted with focused-use appliances.

2001: iPAQ

In this example, a hypothetical user purchases a high-end iPAQ in 2001 for $650 (Brown 2001). The iPAQ is a personal digital assistant and was introduced in 2000. It was originally a Compaq product and is currently made by Hewlett-Packard. The iPAQ comes with Microsoft Windows Pocket PC as its operating system and allows the user to freely install other applications on the device.

The user intends to use the iPAQ not only for calendar and address book functions but also to receive and send email, listen to music and navigate in the car. To do this the user does the following:

- On a Saturday morning, the user attempts to configure the email. The user has to manually configure the server addresses for POP, IMAP and SMTP protocols. He also has to set up the mobile data access by configuring the GPRS access points. Not technically versed, our user calls a friend with more technical knowledge to help him. When he wants to enter email, the user either uses the stylus on the touch screen keyboard or he attaches an external accessory keyboard to speed up his typing. *(Time spent: 3 hours)*
- The user wants to take his CD collection and transfer it to the iPAQ. To do this, he has to first find software on the Internet to convert the songs on the CD into MP3 format. The software costs $18.95. He then has to manually copy the MP3 files from his PC to the iPAQ. He thereby has to make sure that he places the MP3 files into the proper directory path. When he tries to test this, he notices that because he has used too much memory, other applications stopped working. To finally listen to music when he's on his way to work, he has to take the stylus to start the media player application, scroll through a long list to find the songs he wants to hear. Unfortunately, his playlists couldn't be copied from the PC to the PDA. After 3 hours of listening to music, the battery of the iPAQ is exhausted. *(Time spent: 5 hours)*
- Next, our user tries to install GPS-enabled navigation software on the iPAQ. In addition to the navigation software (which already contains the necessary map data) he has to purchase an additional GPS receiver, an SD memory card and a windshield halter (cf. Figure 9). The entire package costs him $300. The user then has to connect the iPAQ to his PC, copy the navigation software and the map data on to the SD memory card and install the software. In his car, the user has to enter the destination with the stylus, place the iPAQ in the halter and start the navigation. He notices that the voice guidance is not very intelligible at higher speeds because the loudspeakers on the iPAQ are quite small. *(Time spent for accessory purchase and installation: 3.5 hours)*

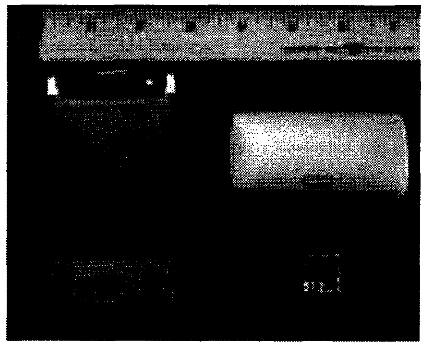

Figure 9: General-purpose PDA (HP iPAQ) with SD card and GPS receiver

Overall, our user has spent his entire day solving technical details and he is still not satisfied with his user experience. He quickly loses interest in some of the applications but they are just too tedious for him to use. He has spent nearly $1000 on this technology.

2007: Blackberry, iPod and Garmin

Our hypothetical user tries again in 2007 to get the same functions. This time around, he tries a different strategy. He purchases 3 single use devices to get the same functionalities:

- He buys a Blackberry phone that is configured for him in the store. Because he signs up for a service plan with Cingular AT&T, the device costs him nothing. Otherwise, he would have to pay $400 for the device (Amazon Inc. 2007b). He starts using it immediately with the provided email address. Typing is easy because of the Blackberry's alphanumeric keyboard. He doesn't have to actively download his email because the Blackberry pushes new email onto his device constantly. *(Time spent: 30 minutes)*
- Next, he buys an Apple iPod nano MP3 player with 8 GB memory for $240 (Amazon Inc. 2007c). He installs the easy-to-use iTunes software which converts his CDs into MP3s. The software then synchronized the music and his playlists with the iPod. Our user also buys 10 new songs for $0.99 each by clicking on the iTunes Store icon on his PC. He also subscribes to some free podcasts (sound files on specific topics, similar to talk radio shows) that are automatically synchronized to his iPod from his PC. He easily accesses the songs, can listen to music for up to 14 hours and the device is almost as small as a cigarette lighter. Additionally, the user likes the styling of the device – it impresses his friends. *(Time spent: 1.5 hours)*
- Finally, the user orders the StreetPilot c330 personal navigation device by Garmin for $280 on the Internet (Amazon Inc. 2007c). When it is delivered, he takes it out of the box, enters the destination on the touch screen, mounts the halter on the windshield and attaches the device. He navigates to a friend's house and hears clearly spoken instructions due to the large loudspeakers. *(Time spent: 30 minutes)*

Overall in 2007, the user has spent 2.5 hours enabling the functionality he wanted vs. 11.5 hours in 2001. He spent $530 and still continues to use the devices daily. He's so thrilled with the navigation device that he even buys one for his 65-year-old father. Sometimes though, he wishes he didn't have to carry around these three devices.

In this consumer example, the focused-use appliance model has enabled an optimized user experience that gives more people (non-technical users) access to the desired functions. While the user has also reduced his overall total cost by almost 50%, the time gained and the continuous user experience are of equal, if not greater importance.

3.3.3 Key Economic Drivers

The examples above show that appliances exist in both consumer and enterprise segments. Some of the economic drivers behind their growing popularity are common to both segments while some are relevant only in one category. These are the economic drivers derived from the examples above as they pertain to the enterprise and consumer segments:

Economic driver	**Enterprise segment**	**Consumer segment**
Hardware commoditization	Yes	Yes
Open source software	Yes	Yes
Margins gained by vertical integration	Yes	Yes
Less vulnerability to software piracy	Yes	Yes
Maintenance fees	Yes	No
Predictability of total cost of ownership	Yes	No
IT personnel costs	Yes	No
Ease of use	No	Yes
Battery life	No	Yes
Single-feature marketing	No	Yes
Disintermediated distribution	No	Yes
Prestige	No	Yes

Balancing these drivers are economic barriers that slow the adoption rate of the appliance model. These impeding factors are:

Economic barrier	**Enterprise segment**	**Consumer segment**
Company capabilities	Yes	Yes
Distribution costs	Yes	Yes

Need to evolve	Yes	Yes
Up-front investment	Yes	Yes
IT security	Yes	No
Inconvenience of multiple devices	No	Yes

Below, these factors will be analyzed in detail to show how the individual factors have changed over the last decade.

3.3.3.1 Common Drivers

The factors described in this section are common to both the enterprise and the consumer segments.

3.3.3.1.1 Hardware Commoditization

Part of the equation of putting together a full appliance stack involves buying hardware to bundle with the software stack. While at the beginning of the computer era software was a negligible part of the total price of a computer, today the hardware has been largely commoditized. For example, the price for processing power has continuously eroded over the last three decades. The cost of one million instructions per second (MIPS) decreased from $480 per MIPS in 1978 to $50 per MIPS in 1985 to $4 per MIPS in 1995 and $0.01 per MIPS in 2006 (Anderson 2006b). The commoditization in the PC market has gone so far that IBM, the giant of the PC era, finally exited the market with the sale of its PC business to Lenovo in 2004 (Bulkeley 2004). These developments allow companies using the appliance model to assemble a total product including hardware at a relatively low cost.

Even though the price of computer hardware has gone down in comparison to other parts of the solution stack and especially in comparison to the labor costs involved with the deployment of IT solutions, hardware's price will never sink to zero. Therein lay several opportunities for the appliance model:

- Customers will always pay for hardware because it is intuitive to them that a physical good costs money to produce (in contrast to software, which can be reproduced at zero cost).
- The "black box" nature of a hardware appliance can additionally make pricing segmentation easier. For example, Google sells its Search Appliance for small and medium firms in a blue box for $3000, for large corporations it sell the same functionality (albeit for more documents) in a yellow box starting at $35'000.
- Cheap hardware lowers the barriers to entry into the appliance model. An appliance vendor today has relatively low costs of inventory compared with 10 years ago (cf. pricing example above of high-end Unix-server in 1997 vs. standard x86 server in 2007)

3.3.3.1.2 Open Source Software

It was common among programmers from the 1950s until the 1970s to share software freely because it was considered only an add-on to the hardware. The change to charging license fees for software (discussed in Chapter 2) changed this practice completely. Companies selling software had an economic incentive to not give their customers access to source code but rather deliver only object code. This protected their investment into the software, now treated economically as a capital good.

While the free software movement founded by Richard Stallman (Williams 2002) had little impact on reversing this paradigm change, more recently, the open source movement has started to effect commercial software producers substantially. The open source movement was founded in 1998 after Netscape had released the source code of its browser to the public under the name of Mozilla (Raymond 1998). Open source software makes its source code available to the public under a license agreement that allows for almost unrestricted usage, modification and redistribution. Often, large numbers of developers collaborate in creating or improving open source software.

A recent blogger's (i.e. someone that regularly writes Internet commentaries on specific topics) comment exemplifies what many in the software industry are thinking (Urlocker 2007):

> *"To me the really interesting thing about open source software is that it can challenge the existing perpetual software license model. Today, companies spend millions of dollars for enterprise software. They pay all the money up front, and then they commit to spend 18-22% in maintenance every year they use it. Not surprisingly, most enterprise software ends up bloated with new features year after year, most of which are never used. I've seen dozens of IT projects fail after spending millions of dollars on overly-complex enterprise software. And let's face it, few enterprise software vendors are loved by their customers."*

While most corporate IT managers considered open source software like Linux or MySQL to be inadequate only five years ago and only allowed it for niche applications like print servers, today many e-commerce, data warehousing or reporting applications are using open source software (Urlocker 2007). This market description exemplifies the disruption process depicted by Christensen. A new, far more inexpensive market entrant with initially inferior quality slowly overtakes the incumbents from behind in more and more sub-segments (Christensen 1997).

The implications of open source software for the appliance model are:

- Software vendors have to find new ways of charging for their products
- Software vendors considering the appliance model are able to use inexpensive open source distributions to build their appliance stack.

Software vendors can capitalize on the new levels of acceptance among IT managers to launch software appliance products powered by open source software that save customers money while protecting the vendor's margins.

By using inexpensive open source software that is optimized for their software application, software vendors can build a stable solution stack that is easy for them to maintain. Regarding internal costs, vendors can also save research and development expenses by eliminating the need to port to, test and maintain several stacks. Although some of those costs have to be re-directed to maintaining the basic open source stack, significant cost savings should be attainable.

3.3.3.1.3 Capturing Margins by Vertical Integration

Vertical integration can be an economic driver if current market solutions have inefficiencies that can be removed by better coordination and adaptation of the parts of the system to each other. Enterprise customers are always looking to reduce the complexity of their business processes. Consumers are seeking to reduce the complexity of their interaction with devices.

One of the complexities of IT management is that corporations have to interact with a large number of entities when they deploy a software solution. Not only do they have to buy the software from different vendors, coordinate the implementation and the operation but they also have to administrate the different software licenses with probably slightly different pricing models (per seat, per CPU, per telephony channel etc.). These myriads of connections exponentially drive costs because often, interaction between all elements of a solution is necessary (Dembo 2004). This creates an economic opportunity for a "one stop shop" vendor to capture the value created by saving coordination costs in the enterprise.

The same is true for consumers. As shown in the initial example above, the end consumer is willing to pay for products that save him time by doing the integration for him.

3.3.3.1.4 Less Vulnerability to Software Piracy

Software vendors of all sizes have to worry about their software being copied illegally. Copyright infringement and creating copies of commercial software is often considered to be a peccadillo, a harmless crime. The Business Software Alliance's 2006 study conducted by IDC estimated that 35% of the packages software installed on PCs worldwide was illegally copied. This would have the losses due to software piracy amount to $34 billion worldwide (IDC 2006, 1-21).

While even companies that sell appliances have to worry about piracy (cf. Cisco's lawsuit against Huawei Technologies for copying their IOS software and using it on Huawei's routers in section 4.2.1.2), combining software with hardware creates a much greater barrier for software customers to illegally copy vendors' software.

3.3.3.2 Enterprise Drivers

The drivers described in the following section are particular to the appliance model in the enterprise segment.

3.3.3.2.1 IT Personnel Costs

For an end consumer, IT personnel cost is irrelevant, but corporations spend a significant part of their IT budget on personnel. Compared to other employed workers, personnel in the information technology sector is expensive. The U.S. Department of Labor's Bureau of Labor Statistics show that the average hourly earnings of non-supervisory workers in the information sector were $23.23 in 2006, over one-third above the average of $16.76

for production and non-supervisory workers the private sector overall (U.S. Department of Labor Bureau of Labor Statistics 2007).

Gartner analysts aptly observed (Prentice et al. 2006, 1-13): "Every year, a euro buys you more infrastructure but less IT labor." There is therefore a strong economic incentive to replace IT labor by infrastructure investment. This is especially true in regards to IT personnel involved in the integration and deployment of new software applications.

The appliance model promises IT managers lower IT personnel costs because the model eliminates a large percentage of the labor associated with installation and maintenance of the solution stack. Currently, there are no large-scale studies to verify this claim. There have been studies that compare individual appliance solutions to their conventional counterparts. For example, Radicati's study comparing the Mirapoint Message Server email appliance to Microsoft Exchange showed that Mirapoint needed 80% less administration staff (Radicati Group 2004). Some analysts predict higher administration costs in the data center because appliances' own administration and management interfaces can often not be integrated into broader data center management tools (Enck, Dawson, and Phelps 2006). On the whole therefore, it remains to be seen whether appliances actually lower IT personnel budgets.

3.3.3.2.2 Maintenance Fees

Software maintenance is defined by the ANSI/IEEE Standard 729-1983 and IEEE Standard 1219-1998 as the "modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment." While correction of faults and adaptation of software is certainly necessary, maintenance and update fees are sometimes seen as the "dirty little secret" of the software industry.

For a software vendor, producing a hit software product is difficult but producing several hit software products is a feat performed by very few companies. A software company can nonetheless survive nicely for decades if it manages to continue to collect maintenance fees on a hit product. As a company matures, service and maintenance fees make up a continuously growing share of total revenues (Cusumano 2004; Nayak 2006).

Currently, software vendors charge between 15% and 25% of the initial up-front perpetual license for maintenance with a recent study showing an average of 17 (Nayak 2006).

The problem with maintenance is that it sets incentives for software vendors to produce an ever-growing set of features, many of which are used by only a tiny fraction of users. From release to release, the feature set increasingly "bloats" (as the blogger mentioned in Section 3.3.2 puts it) the software application. This in turn increases the complexity and size of the application. The results for the software customer are:

- More integration costs for new installations
- More internal IT resources to manage the application
- More memory and other hardware resources are required, therefore higher hardware costs

Maintenance fees therefore may save costs by correcting faults but they increase costs in many other categories relevant for a purchaser of software. In summary, the business model of up-front software licenses with a maintenance component sets economic incentives for the software vendor that are in several ways opposed to the interests of the software customer.

The economic appeal of the appliance model is that it offers to correct some of these incentives. The following conflicts of interest may be solved:

- If software vendors have to supply the hardware, their interest in bloating applications' memory and processing power requirements is diminished because more expensive hardware lowers their margins.
- If software vendors have to create easy-to-use configurations, their interest in overly increasing feature sets is decreased.

While customers may find this compelling, the strategic challenge for software vendors remains that most of them need to find a source of recurring revenue to sustain operations beyond a hit product. Therefore, it would be strategically advisable to software vendors to continue to charge a recurring component.

3.3.3.2.3 Predictability of Total Cost of Ownership

In 2005 organizations and governments spent an approximately $1 trillion on IT hardware, software, and services worldwide. Companies on average spend about 4 to 5 percent of revenue on IT, with those that are highly IT dependent—such as financial and telecommunications companies—spending more than 10 percent on it (Charette 2005). To keep IT costs predictable and in check overall is therefore a significant concern for executive management on all levels.

Research shows that IT projects are routinely late and costs run over budget. A large percentage of projects even fail outright. The annual Standish CHAOS report of 2004 showed that 18% of IT projects failed. Those that did succeed were on average 56% more expensive than budgeted and took 84% longer than expected. In addition, 64% of IT managers also rated themselves poorly or only moderately skilled in their ability to accurately estimate IT project time and costs. (The Standish Group Inc. 2005)

Even worse, the costs incurred *after* the installation of an enterprise IT system can account for up to 90% of its total lifetime costs (Vincent et al. 2005). These so-called "downstream" costs consist of maintenance expenses, hardware replacement, user support costs, and changes to a system. The resulting "total costs of ownership" (TCO) are therefore an important metric for the management of IT systems.

Appliances can help to increase the predictability of TCO through the following factors:

- Since the appliance comes as a preconfigured system, there is far less integration effort. By definition, common problems in the installation of software such as hardware incompatibilities or unexpected interaction between software modules can't happen when installing a pre-tested appliance.

- Appliances are easier to administer and maintain, since all necessary components fit together. Therefore, there are less surprises from skyrocketing system administration costs (Radicati Group 2004)
- Thanks to the optimized matching of hardware and software components, appliances tend to be more reliable than modular systems. This significantly reduces downtime costs and repair expenditures.
- There are no unexpected additional costs for third-party software licenses. For instance, many conventional enterprise software products require additional licenses for a relational database, such as Oracle or Microsoft SQL Server. Whenever the vendor of such third-party components changes prices, additional costs are incurred. In contrast, in a pre-bundled appliance, all necessary components are already installed and fully licensed.

3.3.3.3 Consumer Drivers

The economic motivation driving the adoption of the appliance model in the consumer segment is described below.

3.3.3.3.1 Ease of Use

As the consumer example above showed, a well-designed user interface that is tailored to the focused use of the device can drive adoption of single use appliances.

Usability of computing devices has long been an issue of contention. In the 1990s studies claimed that computers hindered rather than enhanced productivity (Landauer 1996). While professional users are often forced to interact with less than optimal user interfaces, end users will abandon devices (or partial functions of devices) that are hard to use. A classic example of this is the flashing "12:00" on many home video recorders because, according to surveys, one-third of users never invested the time to be able to program a VCR. Instead they used it only to replay bought or rented videos. Research shows that the more uses a device has, the more complex the user interface becomes and the less likely it is that users will be able to perform more than the most basic tasks. (Cooperstock 1997). Because of this research, more and more devices during the last decade are being designed with usability in mind.

Unfortunately, multipurpose devices like PDAs often transfer user interfaces originally designed for the desktop computer to a completely differently structured task (e.g. on Windows Pocket PC, the desktop interface is applied to mobile usage scenarios). This often frustrates users and creates economic opportunity for single-use appliances.

Makers of appliances with a single or very focused uses have the luxury of designing the user interface around that use. By optimizing the user interaction for that single use, consumers can interact more efficiently with the device. This efficiency promotes prolonged and more intensive use, thereby creating more value for the consumer. Value for the consumer translates into higher willingness to pay. This makes usability a strong economic driver for single-use appliances.

3.3.3.3.2 Battery Life

Still inadequate life of rechargeable lithium-ion batteries makes multipurpose devices less useful than they could be. For example, on a long-haul economy flight a user can work on his or her laptop for only about 3 hours before the battery runs out. Experienced users therefore carry a mobile DVD player, an MP3 player, maybe even an e-book reader to cover their media consumption needs.

For example, recent research whether on music-enabled cell phones will overtake dedicated MP3 players found (Gartenberg and Wood 2007):

> *"While vendors continue to innovate and introduce new music-enabled phones, these phones will not compete directly with dedicated media players in the near term. With limited internal storage capacity, proprietary technologies, and* ***battery issues*** *plaguing phones, the music consumption experience will not parallel that of the dedicated device."*

In consumer surveys of desired features for mobile devices, long battery life is often at the top of the list (Vasquez 2006). Appliances profit from the continuing inadequacy in battery technology. Even though much research (e.g. by MIT spin-off A123) is being conducted in this field, commercially viable breakthroughs for mobile devices are not on the horizon yet. Therefore, single-use devices may continue to offer advantages in regards to usage length for a longer period of time.

3.3.3.3.3 Single-feature Advantages

Some products need only one single feature that differentiates them to make them a market success. For example, a camera that is completely waterproof may not differ in any other way from other cameras but in this single aspect. Still it can command different prices and target a different segment than all other cameras in the market.

For companies with appliances in the consumer segment, this can create a large market opportunity. For example, Apple has recently launched a new device called "Apple TV". Inside the box, the Apple TV is simply a lower-end Apple desktop computer. But it is now packaged as a set-top box. The only function it performs is that it connects to your TV and wirelessly to your desktop computer and streams photos, music and videos from there. Of course, it performs this function with an easy-to-use interface and it is elegantly designed.

The economic driver for appliances is finding the single feature that creates so much value that the end user is willing to buy an entire new device just to get this one feature.

3.3.3.3.4 Disintermediated Distribution

Disintermediation is term used to define the elimination of one or more organizations from the distribution channel (Gallaugher 2002). In the past decade the Internet has allowed some companies to distribute their products more efficiently.

While most consumer products are still sold through traditional retail outlets, the U.S. Census Bureau reported that in the last quarter of 2006 already 3% of all retail purchases were conducted online. In 2006, $108 billion was spent on online purchases in the United States, 23% more than 2005 (U.S. Census Bureau 2007). In the segment of consumer electronics, a larger percentage of products are sold online. For example, in 2006 22% of MP3 players were sold online (The NPD Group Inc. 2007).

Because they are not limited by shelf space or other physical inventory constraints, e-commerce channels can offer a broader range of products to consumers. For example, Amazon.com currently offers 3.7 million book titles while the average Border's retail store stocks 100,000 book titles. Some consumers are increasingly buying products that are not top sellers but rather cater to their individual tastes. 25% of Amazon's book sales revenue comes from outside its top 100,000 titles. This phenomenon has been described as "The Long Tail" (Anderson 2006a).

For appliance companies this new trend can be an economic driver. In the past, companies without strong ties to retail distribution networks stood no chance of marketing their products to the end consumer. Today, a compelling appliance can be marketed entirely online and distributed directly. Appliance companies can use this mechanism to create a new market that wouldn't have been possible with traditional retail distribution channels.

3.3.3.3.5 Prestige

In addition to ease of use, consumers' purchases are often also triggered by the prestige factor of a device. A sleek appliance is a more visible statement of technological "hippness" compared to just installing new software on an existing general-purpose device. Social prestige generated by a sleek design can be translated into an economic advantage because it increases the consumer's willingness to pay.

Apple's iPod is prototypical example where "coolness" was a major factor for adoption. A recent research study showed that while 6.2% of iPod users bought the device because it had an easy-to-use interface, almost 45% of buyers said they bought it because the device looked cool or it made them look cool (Gilliam 2005).

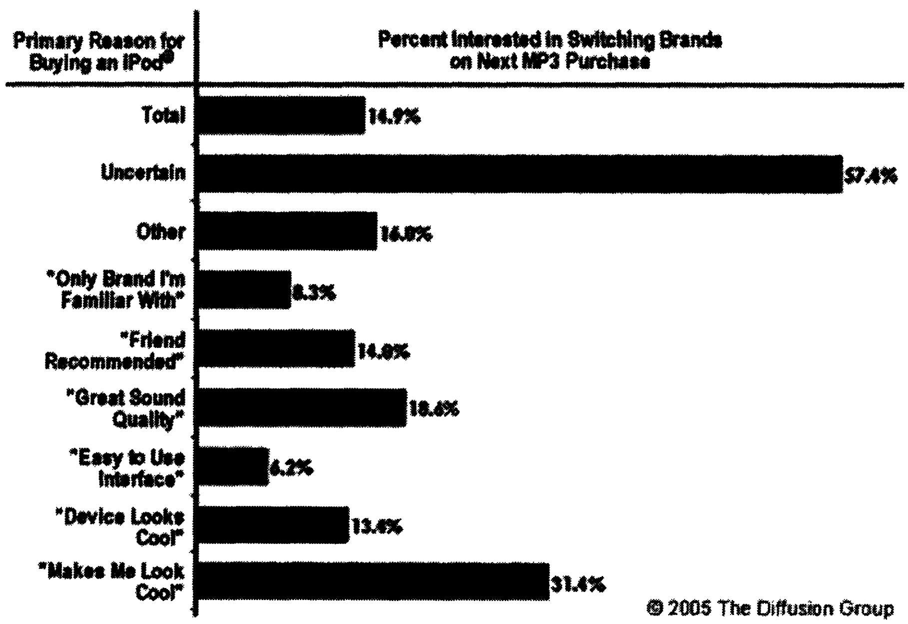

Figure 10: Primary reason for purchasing an iPod

Because appliances are focused around a specific use, they have to make less design compromises than multi-purpose devices that need to allow for numerous user interactions. If an appliance vendor succeeds in creating a coolness factor (e.g. by a sleek design) this can be a strong driver for device sales.

3.3.4 Key Economic Barriers

What made the packaged software model so attractive is that relatively little capital investment was needed to start a software company. While it may take between several months and several years to roll out a first product, a traditional software company needs little other resources than a business idea, a few capable software developers and general-purpose computing hardware to get started. Marketing and sales and other functions can usually be added in later stages. This is not the case for appliance companies. The physical aspect alone creates economic hurdles for an appliance company ranging from hardware design to distribution to up-front investment in inventory.

In the following these barriers to the adoption of the appliance business model are discussed. First, barriers that are common both to enterprise and consumer appliances are analyzed. Subsequently, barriers that are relevant only to one of the segments are treated.

3.3.4.1 Common Barriers

3.3.4.1.1 Company Capabilities

One of the key economic barriers that companies face when implementing the appliance model is the necessity to build additional capabilities. This can be an expensive and time-consuming endeavor. Today, a prototypical software company may be organized like this:

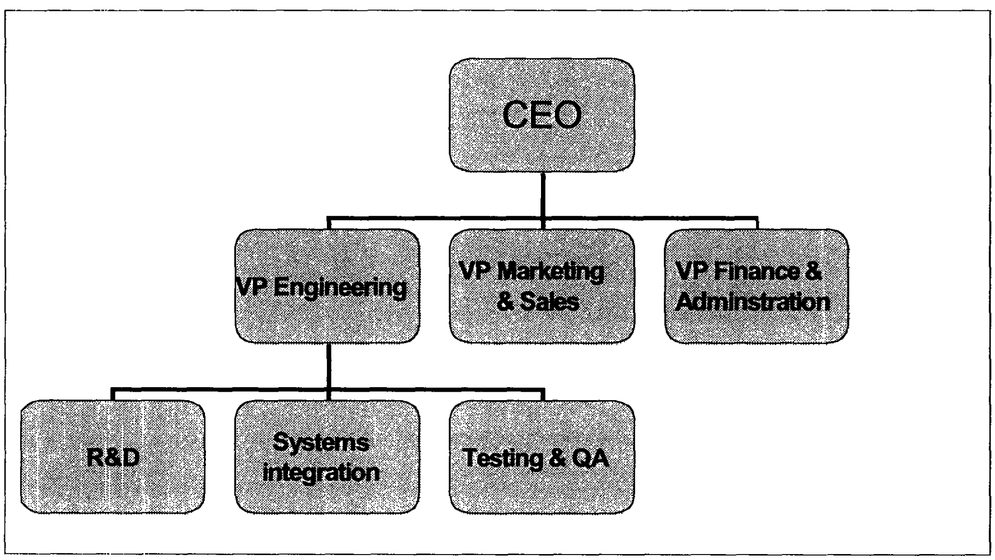

Figure 11: Prototypical organization chart of a packaged software company

The research and development department is the core of the software creation. Systems integration ports and packages the software on standard platforms like Windows, Macintosh, Unix or other operating systems. Testing and quality assurance assess whether software can be released to customers. In the appliance model, several functions have to be added to the organization. A prototypical organizational chart of an appliance company could look like this:

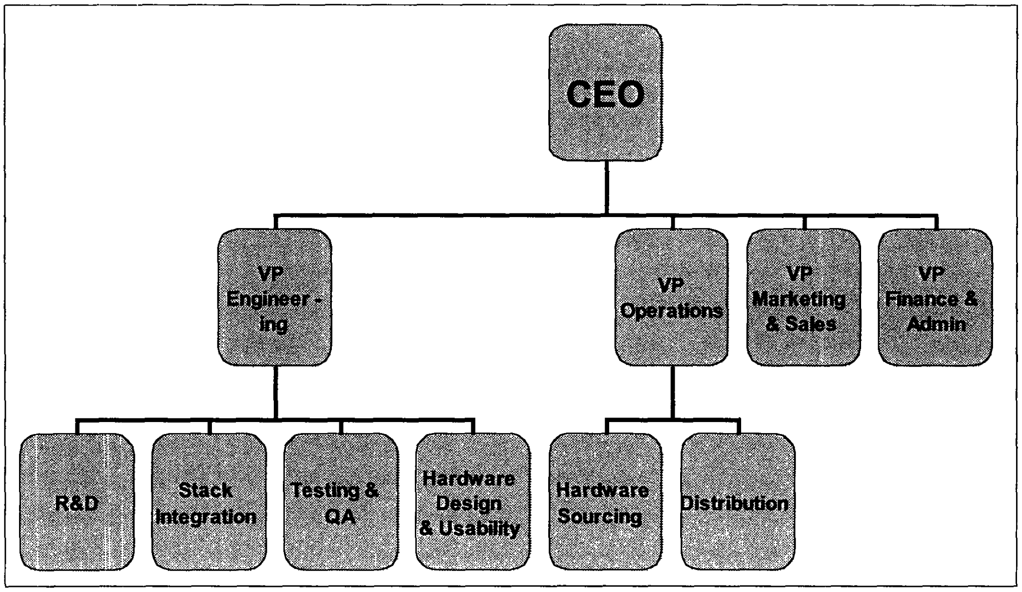

Figure 12: Prototypical organizational chart of an appliance company

While the systems integration department has simply been replaced by the stack integration department, an appliance company would also have to create a department for usability (to design the easy-to-use graphical user interface) and maybe also hardware design (if the company is designing consumer devices). In addition, an operations function is needed to source either commodity or specially designed hardware. Furthermore, physical distribution has to be organized by a distribution department.

These additional functions add a level of complexity that creates relatively large economic hurdles comprised of:

- Time and expense of hiring additional qualified personnel
- Management team has to master more areas of expertise
- Interaction between more departments adds to complexity of management function

It is therefore to be expected that fewer appliance companies will form than traditional packaged software companies. Just as with any new technology or business model, the adoption of the appliance model will be slowed by the lack of experienced managers that know how to successfully manage an appliance company's operation.

3.3.4.1.2 Supply Chain and Distribution Cost

The organizational chart for an appliance model company (cf. Figure 12) shows that a distribution department often has to be created when companies want to pursue the appliance model. While consumer software companies that sell through retail outlets have always had a physical distribution network, most enterprise software companies have no or only have a rudimentarily developed physical distribution function.

Building a distribution system for a physical product is far more challenging for a software vendor than allowing a software download over the Internet or allowing access to a hosted solution (like SaaS solutions). Because today hardware components are manufactured all over the world, an appliance company may have to become involved in the complexities of global supply chain management including demand forecasting, capacity management and logistics. Maintenance of the delivered hardware also poses another difficulty.

The magnitude of this barrier depends on each appliance company's business model (cf. Section 3.4.6). One company may only need to source commodity rack servers while another may design its own integrated circuits, hardware casing etc. In the last years, this hurdle has become much more surmountable for medium-sized companies due to contract manufacturers like Solectron, Flextronics, SCI, Celestica and Jabil Circuits (Lüthje 2002). Some of these companies now also offer additional services such as supply chain management. They will help with design, source the parts, manufacture the product, transport it to a company's distribution center, to the retail outlets or even to end customers and service the products (Flextronics Inc. 2007). By contracting these services out, though, the appliance company loses margins. This lowers the economic incentive for the adoption of the appliance model.

3.3.4.1.3 Up-front Investment in Inventory

The traditional software business model has allowed companies to produce their products on demand. The appliance model usually implies building an inventory of physical goods. Any business requiring physical inventory brings with it an up-front investment. This investment can range from:

- Buying commodity servers to be re-sold immediately
- Having a parts inventory in manufacturing plants
- Having finished products inventory on commission at retail outlets

One of the interviews for this thesis revealed for example how hard it is for software companies to switch their mindset to managing the investment in physical inventory (cf. interview Peter-Frans Pauwels in the Appendix):

> *"If you have product in the stores and you introduce the next generation product you need to manage that properly. When we first started shipping our own products, one quarter we announced a new product prematurely and the retailers still had thousands of units of the old model. They returned the old model and only wanted to sell the new one. We had to write off all of those units."*

The capital investment in inventory represents a large barrier to entry to the appliance model for small and start-up companies. The investment needed can represent a substantial fraction of a firm's revenues. Comparables are here consumer electronics manufacturers. For example, the Sony Corporation in 2006 invested almost 10% of its total revenues in inventory (Sony Corporation 2006a).

Because in the appliance model gross margins are no longer 99% (as in the packaged software business model) but rather between 40 and 75% (cf. Chapter 4 for industry examples), it may take longer to recuperate the liquidity lost through inventory investments from reserves formed through net income. The investment in inventory ties up cash that otherwise could be invested in more development or other company activities. This cash drain therefore can hinder growth of appliance companies.

The more inventory-intensive variants of the appliance model a decade ago would therefore have been limited to well-financed large companies. Falling hardware prices and recent changes in distribution have simplified market entry for small companies somewhat. In Section 3.3.3.3.4 the disintermediation of distribution is described. With an efficient distribution, less up-front investment is needed. Nonetheless, the need for physical inventory remains a formidable economic hurdle that strongly impedes the adoption of the appliance model.

3.3.4.1.4 Need to Evolve

Appliance solutions lack one of the key advantages of multi-purpose computing platforms: their ability to adapt and evolve when the corporate or consumer environment changes. Appliances are meant to be stable and not to change rapidly because they are deployed to perform a specific function. This makes them inherently less flexible than general-purpose computing platforms. This inflexibility leads to higher costs due to the resulting economic inefficiencies. If a hardware appliance is not serving its purpose, it will need to be replaced. In contrast, on general-purpose computing platforms obsolete software can simply be replaced by new software.

Some consumer appliances are even meant not to evolve for economic reasons. They therefore have obsolescence planned into them (cf. conflict around iPod battery replacements).

In the enterprise segment, some commentators say that appliances will be part of the 80/20 rule. They assume that the well-understood, commoditized tasks will be wrapped into appliances. These constitute 80% of all computing tasks. The other 20% are the evolving business applications that create the highest value and allow the enterprise to innovate. These 20% are high-margin software applications while the appliances will be commodities (Gilfix 2007). The counterargument to this reasoning is that high-value applications can also be deployed as appliances but may need more customization than more standard applications.

Another voiced complaint is that appliances don't offer the granularity of features customers need and therefore, appliance customers often have to settle for a solution that is suboptimal for their business requirements (Robinson 2007).

But in general, appliances will most likely not be adopted for applications that are highly customized and need to be changed or adapted very frequently. In these segments, general-purpose computing platforms with packaged software will prevail. If this only leaves 80% of the software market to the appliance model, there will still be a very large market for appliances.

3.3.4.2 Enterprise Barriers

3.3.4.2.1 IT Security

In the enterprise sector, IT security can be an economic barrier to the adoption of the appliance model. While appliances are less under scrutiny for their security risks than software as a service solutions are, the "black box" nature of appliances is discomforting to some IT managers. Appliances have the advantage that they can be physically located on the enterprise's premises or in a secure data center but the open source software stacks that many enterprise appliance solutions are comprised of don't always pass muster when examined by corporate security compliance officers. The reason for this is that the stack running on the appliance may contain security vulnerabilities that can't be fixed without the appliance vendor stepping in and providing a patch. This usually requires that the appliance vendor has remote access to the appliance, which not all enterprises are willing to provide.

On the other hand, appliances can also be *more* secure than traditional software because the operating system and the application stack can be "hardened" more easily before deployment, i.e. the appliance uses only features and functions necessary for its use (Cummings 2005). This gives less room to hackers and viruses that exploit little-used functionalities with security loopholes.

3.3.4.3 Consumer Barriers

3.3.4.3.1 Inconvenience of Multiple Devices

As described in the example at the beginning of this section, one barrier to the ubiquitous adoption of the appliance model in the consumer segment is the willingness of consumers to carry yet another device around with them.

Economically, the cost of multiple devices is evident. Multiple devices imply multiple CPUs, multiple screens, multiple input variants such as styluses, keyboards, dials etc. The user needs bigger or multiple pockets or bags to carry the devices around or store them in. All of this amounts to higher cost but, often more important to the consumer, more inconvenience. The more devices one has the more likely it is that one is missing the most fitting device just when it is needed.

General-purpose devices that serve a number of functions are much more likely to create the value necessary to be deemed worth of the precious space in the consumer's briefcase or bag. Therefore, new appliances have to demonstrate much higher value to a consumer before he or she is willing to add yet another device to his or her daily life. In 2007, U.S. households already owned an average of 25 consumer electronics products. But each year, the average American adult also spends $1,200 on new consumer electronics (Consumer Electronics Association 2007a).

3.3.5 Summary

This analysis of the economic drivers and barriers to the appliance model has shown that this business model has recently become more viable. The main reasons for the viability lay in development of costs associated with software and with deployment of appliances.

Cost increases of software for the *software purchaser* are caused by the following factors:

- Labor (cf. sections on IT personnel cost and maintenance fees)
- Complexity of software (cf. sections on predictability of TCO)
- Deficiencies of general-purpose hardware (cf. sections on ease-of-use, battery life, prestige)

The appliance model lowers these costs by, for example, implementing easy-to-use single-feature devices.

Cost decreases or margin improvements for the *software vendor* are caused by the following:

- Hardware commoditization
- Open source software
- Vertical integration
- Disintermediated distribution

Lower costs in these areas are the driving factors that advance the adoption of the appliance model in the software industry.

Several important factors, though, slow the adoption rate of the appliance model. These economic barriers involve rising costs for either the *software purchaser* or the *software vendor*.

For the *software vendor*, the following factors increase costs:

- Building company capabilities
- Distribution costs
- Up-front investment in inventory

For the *software purchaser* costs are increased by the following factors:

- Less flexibility (cf. section on the need to evolve)
- Concerns about IT security
- Inconvenience of multiple devices

Overall, while the appliance model has been enabled by a number of shifts in cost structure over the last decade, there are still serious hurdles for companies that want to adopt the appliance model. Because the appliance model creates more hurdles for companies to launch into this field, it is unlikely that it will achieve the ubiquity of the packaged software model.

3.4 Business Criteria for the Adoption of the Appliance Model

The section above has shown on a macro-industry level the economic drivers and barriers for the adoption of the appliance model. That analysis has shown that the appliance model is not a panacea for all of the malaises in the software industry. While the appliance model may be very profitable for some vendors, others could greatly harm their business by adopting it. From the analysis above and the evidence gathered in the case studies (cf. Chapter 4) this section attempts to design a decision framework for companies seeking to adopt the appliance model.

3.4.1 Decision Funnel for the Appliance Model

Deciding to implement the appliance model is an important strategic decision for a software vendor. To assess whether the appliance model makes business sense, this section proposes a decision funnel that starts from the broad criterion of an industry assessment, then assesses the customers and at the end finally narrows the question down to company competencies (cf. Figure 13).

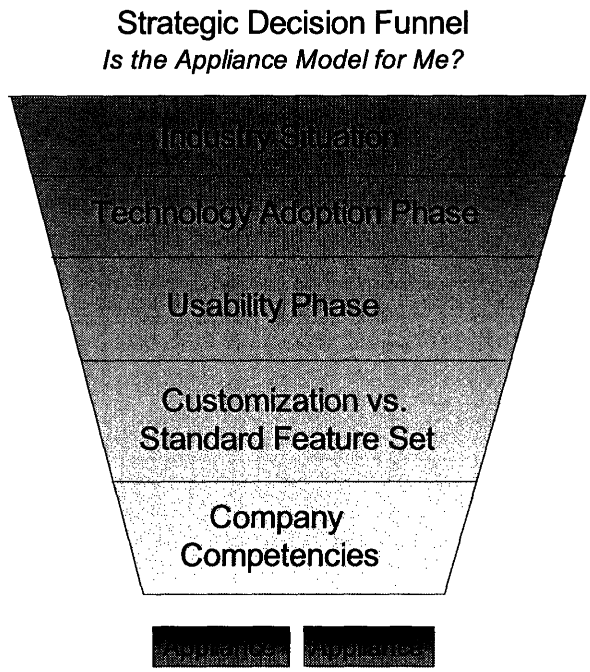

Figure 13: Strategic decision funnel for the appliance model

In the following section, the elements of this decision funnel are described in detail.

3.4.2 Industry Situation

As a starting point, a software company can use the elements of Michael Porter's Five Forces to understand the situation of the segment it is operating in (Porter 1980). Questions that need to be answered are:

- What are the current competitive dynamics? How strong is industry rivalry?
- What power do suppliers and customers currently have on pricing and access to technology?
- What are current barriers to entry? Are there recent new entrants?
- What are substitutes for the software used today?

Important parts of this analysis are the current break-down of value creation in this segment. What are the current margins for the software application developers in this segment? Which other players capture the value created? If, for example, a significant part of the value is captured by the lower parts of the solution stack and viable open source alternatives are available, this segment could be a good candidate for the appliance model.

A simple guideline for the enterprise segment is that if today's price point of a company's software is far lower than the hardware required for the bundle, only an extremely well-designed and convenient appliance can hope to penetrate the market. For example, if a company sells software for less than $100 to home-office users, it will be hard to find a value proposition for a successful appliance if the current software doesn't involvement significant installation hassles and is essential to the functioning of the user's office (Crowley 2006).

In the consumer segment, this equation is more difficult because simple software can create a large value-add if the appliance is well designed (cf. section on usability below and Chapter 4 for industry examples such as Apple's iPod).

Should a company conclude that the value creation in the industry supports the appliance model in regard to the potential for vertical integration, cost savings and industry dynamics, it should proceed to the next level of the decision funnel.

3.4.3 Phase in the Technology Adoption Cycle

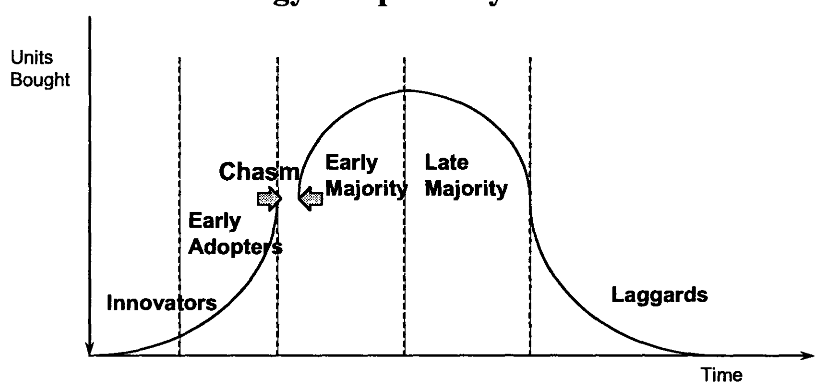

Figure 14: Phases of technology adoption (cf. Moore, graph adapted from Cusumano)

Geoffrey Moore's Crossing the Chasm describes the phases of technology adoption (Moore 1991). For a company considering the appliance model, it is necessary to analyze which phase of the technology adoption cycle its products are currently in. The attractiveness of the appliance model varies from phase to phase:

- **Innovators:** For companies in the innovators phase, the appliance model is largely unattractive. Its customers are technically competent users that derive satisfaction from the integration and installation involved with new devices. They are mostly not interested in having functionality hidden away. A company planning to become an appliance company can use innovators as alpha testers for its products. In this phase, the size of the addressable market does not support hardware design, sourcing and production. Therefore, only consumer products companies with a large investment budget would be able to enter the market with an appliance at this stage.
- **Early adopters:** Once a technology has reached the early adoption phase, many of the early problems and bugs have been corrected. Companies in this phase can

begin to think about the appliance model because the market size is expanding rapidly. The appliance model can help companies in this phase "cross the chasm" because complicated interfaces are a large hurdle to adopters in the early majority.

- **Early majority:** The early majority is the main market for the appliance model. Appliances usually address an already well-known need from a different angle. Buyers in the early majority may not want all the complex features early adopters need but they are willing to pay for a low-maintenance, easy-to-use solution. The market size supports all the investment necessary in hardware design and sourcing as well as distribution and physical inventory management.
- **Late majority:** The late majority is more price sensitive than customers in other phases. Here, the appliance model can enter if the demonstrated cost savings are high. Should an appliance achieve an order of magnitude cost savings, a significant disruption of this market is possible.
- **Laggards:** This segment presents a marginal opportunity for the appliance model. If a company can strip down a well-known product to its bare minimum and find a very low price point, it may be able to access the laggard market.

Once a company has determined which phase of technology adoption their appliance product would be located in and concluded that its user base is open to an appliance, it can continue the analysis with the usability requirements. Because the adoption phase of technology often coincides with its usability, these two requirements are closely connected.

3.4.4 Usability Phase

Section 3.3 above discussed that usability can be a strong economic driver to accelerate adoption of the appliance model. For companies that are considering the appliance model it is important to analyze what level of user competency is necessary to use their software.

The following levels can serve as guidelines:

1. Expert software engineer: many years of development experience
2. Expert system admin: years of experience in system administration
3. Versed technology-affine user: technically competent knowledge
4. Educated computer user: knowledge of basic to intermediate computer configuration
5. Unsophisticated computer user: knowledge of basic applications like email, web browser, text processing
6. Non-computer literate user: doesn't use computers but can operate TV, cell phone

How a company proceeds after this has been ascertained depends on the segment. Consumer and enterprise segments have different dynamics of when it makes sense to introduce an appliance.

Level of competence	Enterprise segment	Consumer segment
1	Degree of integration and configuration effort is still too high. **Action:** It is too early to bundle. Stay with traditional model.	Degree of integration and configuration effort is still too high. **Action:** It is too early to bundle. Only experimental bundles with level 1 testers could be used for pre-release tests.
2	Is it possible to simplify use and configuration via GUI to reach level 3? **Action:** If yes, an appliance development could be started for high-end corporate market.	Is it possible to simplify use and configuration via GUI to reach level 3 and 4? **Action:** If yes, an appliance development could be started for testing in the high-margin but low volume tech-enthusiast consumer market.
3	GUI still needs versed user for configuration. Is further simplification possible? **Action:** Launch appliance for high-end corporate market. Continue simplification to reach medium-sized corporate buyers	GUI still needs versed user for configuration. Is further simplification possible? **Action:** If financing is available, launch appliance in the high-margin but low volume tech-enthusiast consumer market to gain feedback for broader market.
4	Software only needs to be bundled with hardware to enter appliance market. **Action:** Launch appliance for general corporate IT market.	Software only needs to be bundled with hardware to enter appliance market. **Action:** Launch appliance for early-adopter consumer (often also called "prosumer") market. Achieve further simplification or reduce feature set.
5	Software is already easy-to-use for average office employee and could access a broader market if bundled. **Action:** Launch appliance for small office/home office (SOHO) market.	Software is already easy-to-use for the general computer literate user and could access a broader market if bundled. **Action:** Launch appliance for targeted to early majority market.
6	Software can operated without training by non-computer literate	Software can operated without training by non-computer literate

	users. **Action:** Software bundled with hardware is applicable for uses by manually-skilled field workers such as tradesmen, mechanics, warehousemen etc.	users. **Action:** Appliance for late-majority mass-market possible.

Once a company has determined what phase of usability its software is in, further customer requirements have to be analyzed. Whether software is easy to use or not is often a function of the need for customization.

3.4.5 Customization vs. Standardized Feature Set

This section discusses what level of individualization still supports the appliance model. The more customization an application needs, the less it is suited for the appliance model. Section 3.3 identified the need to evolve as one of the economic barriers. From a strategic point of view, a company has to consider where the key added value of the software lies:

- Does the current business model involve a lot of high-level consulting services?
- Do your customers need to adapt the software to their business processes?
- Or do you mainly need technicians to install and configure the software?

If customers use the software to perform a key business function that is very specific to their business, it may be in the software vendor's best interest to continue with a traditional packaged software model combined with high-margin integration services. If many customers solve the same problem in an almost identical fashion, the software may be a candidate for the appliance model.

This criterion sounds as if appliances where only destined for low-level uses such as firewall or email appliances but today you can already find appliance solutions in once highly individualized software categories such as Enterprise Resource Planning (ERP) or Customer Relationship Management (CRM) software (Cummings 2005). Therefore, before dismissing the opportunity too quickly it may be worthwhile to think intensively about whether a software application can be adapted to the appliance model. The most important questions for a software vendor in this respect are:

- Is there a "good-enough" feature set that appeals to a broad customer segment?
- Can feature set levels be created to sell variants of an appliance to different segments?

If a software vendor has the prospect of gaining a significantly higher market share or opening up previously inaccessible market segments by offering a much less customizable appliance, the next and final step of the strategic decision funnel involves whether the company has the right competencies to enter into the appliance model.

3.4.6 Company Competencies

In Section 3.3 above one of the major economic barriers discussed are company capabilities. Even if all levels of the decision funnel returned a positive analysis for the appliance model, lacking competencies can be a knock-out criterion for the appliance model. Depending on whether a company is simply looking to put its software on a standard LAMP stack in a rack configuration or wants to design its own hardware, different know-how is needed within the company.

The following questions need to be addressed:

- What organizational structure will the company need to change business models?
- Do the current employees have the competencies necessary to fill the positions in the new structure?
- Are enough management and financial resources available to fill the knowledge and experience gaps?

Competencies that companies need to transition to an appliance will vary individually but these are some of the areas of competence that may have to be added or strengthened:

- Open source stack solutions
- Hardware design
- Components sourcing
- Distribution
- Logistics
- Product management

A software vendor's management team has to honestly assess whether their current backgrounds and experience will be enough to support the change in business model.

4 Industry Examples for the Appliance Model

The last chapter described definitions, economic drivers and a decision framework for the appliance model. This chapter takes this as a basis and applies the concepts outlined there to examples from the consumer electronics and enterprise applications market. The aim is to provide details about the markets and individual companies to show how the economic drivers from Chapter 3 influenced the success of different appliances.

4.1 Consumer Electronics

The market for digital consumer devices is the sector where the appliance model currently plays the biggest role. Consumer markets are under similar pressure as the enterprise computing market. Margins for commodity consumer electronics products are thin, and shrink-wrapped consumer software is increasingly threatened by free web-based services and freeware.

Therefore, a number of manufacturers are trying to differentiate their products by combining hardware with advanced software functionality. The resulting products often achieve very attractive margins. Well-known examples include Apple's iPod and Sony's Playstation 2 game console, both highly profitable products. These products fit the definition of an appliance: They bundle hardware and software, but derive most of their end-user value from the software component.

For consumers, ease of use and ease of installation are crucial points when shopping for electronics products. But at the same time, today's savvy consumers expect a lot of functionality, particularly for the management of digital media. Like enterprises, consumers often have a choice between different solution approaches with different degrees of integration (Göldi et al. 2006a).

In this section, we examine three industry examples where fully integrated appliances have won against software solutions that run on top of a general-purpose solution stack (such as a PC or PDA): Digital audio players, personal navigation devices and digital video recorders. A fourth example describes speech recognition for toys. It shows how a small company can carve out profitable market leadership in a niche market by integrating hardware and software.

4.1.1 Digital Audio Players

This section gives an example of how companies who use their advantage in software integration in combination with hardware design can successfully penetrate a new consumer market segment.

Digital audio players allow users to listen to digital content. These players are software applications that decode and play compressed audio files. The most common format for audio files is called MP3. MP3 (officially called MPEG-1 Audio Layer 3) is a compression algorithm developed in 1991 by a team of European engineers from the Fraunhofer Institute, Philips, CCETT and IRT. MP3 became an ISO/IEC standard in 1993 (Fraunhofer Institut 2007). Digital audio players are therefore widely referred to as MP3 players (Consumer Electronics Association 2007b).

While digital audio players, strictly speaking, are software applications, colloquially they are understood to be portable devices for audio consumption. The software inside the player has become the name for the entire appliance.

4.1.1.1 The Market for Digital Audio Players

Predecessors to the today's MP3 players were Sony's Walkman (released in 1979) and portable CD players. They created the market of personal stereo music devices in the 1980s. The Sony Walkman had sold 100 million units by 1992 and 150 million by 1995 (Sony Corporation 2006b). The first MP3 players were Diamond Multimedia's "Rio 100" and Saehan Information System's MPMAN, both introduced in 1998 (Fraunhofer Institut 2007).

Consumer adoption of MP3 players was quite sluggish at the beginning. As described in the example in Section 3.3.2, this was due to the expert knowledge necessary to convert and transfer digital music files from a CD via the desktop computer to the player. Also, there was a trade-off between a pocketsize product that stored only 20-30 songs and a larger device that could carry an entire song collection. The user interfaces of early MP3 player didn't provide convenient access to the songs. Subsequently, in 1999 and 2000 only around half a million devices were sold (Consumer Electronics Association 2004; Consumer Electronics Association 2006a; Consumer Electronics Association 2006b).

The market needed a product that was able to "cross the chasm" from early adopters to the early majority. This turned out to be the Apple iPod. It was the perfect appliance: the iPod provided an elegant product with excellent software integration and a simple user interface.

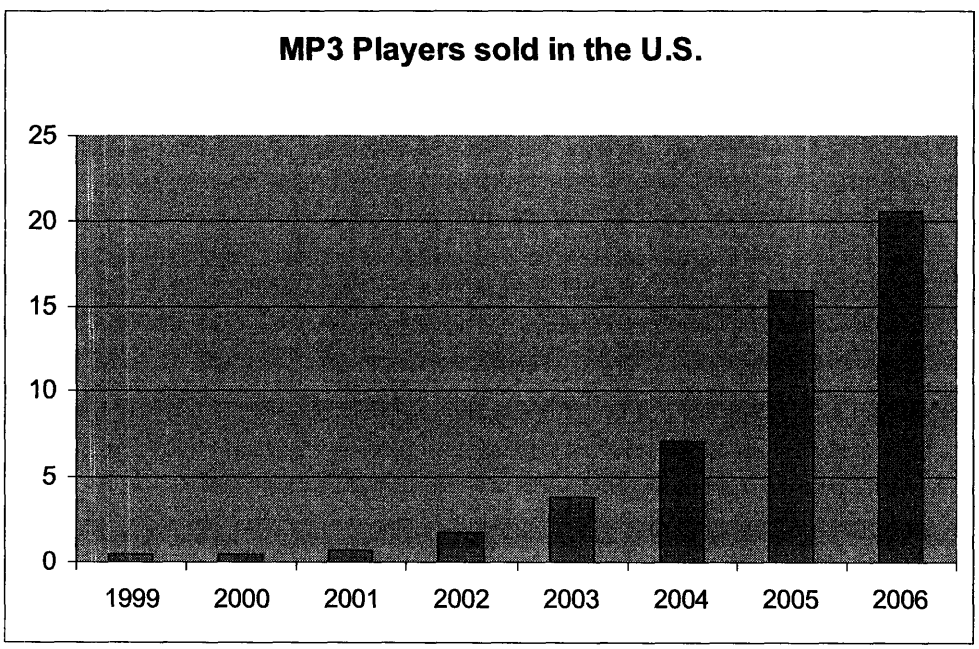

Figure 15: Number of MP3 players sold in the U.S. in millions (Consumer Electronics Association 2004; Consumer Electronics Association 2006a; Consumer Electronics Association 2006b)

Since the introduction of Apple's iPod in late 2001 the adoption rate of MP3 players has increased exponentially as shown in Figure 15. In January of 2005, 15% of U.S.

households owned an MP3 player. By April 2007 that number had increased to 32% of households, making MP3 players a mass-market, common household item (Consumer Electronics Association 2007a).

4.1.1.2 Case study - Apple iPod

The Apple iPod is a portable MP3 music player made by Apple Inc., a company located in Cupertino, California. As of April 2007, over 100 million iPods have been sold worldwide. This makes the iPod the fastest selling music player in history (Apple Inc. 2007a) – reaching the 100 million mark after only five and half years compared to the 11 years it took the Sony Walkman. The iPod has become a mainstream product that has changed the way people consume music.

The iPod was launched in November 2001 and currently sells in three basic product variants: the iPod (storage on hard disk), iPod nano (storage on flash memory) and iPod shuffle (random-play device with no display, storage on flash memory) (Apple Inc. 2007b).

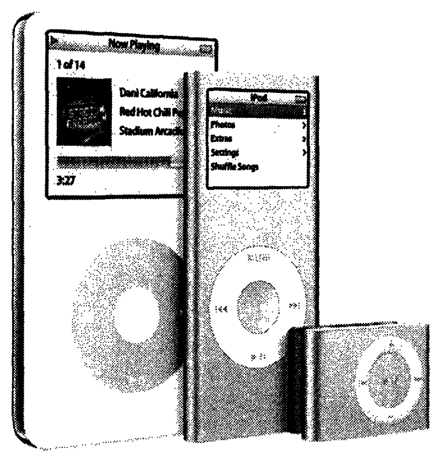

Figure 16: Apple iPod, iPod nano and iPod shuffle (Apple Inc. 2007a)

iPod Development

In January 2001, Apple launched iTunes, a digital media player application software that plays and organizes digital music, as a free download. The software was originally developed by a company called C&G. iTunes was released as part of Apple's "Digital Hub" strategy to extend into digital media like music and photography. Originally, iTunes was intended to work on Apple's desktop machines as a jukebox application. At its launch, Apple CEO Steve Jobs said (Young and Simon 2005):

> *"Apple has done what Apple does best – make complex applications easy and make them more powerful in the process. iTunes is miles ahead of every other jukebox application, and we hope its dramatically simpler user interface will bring even more people into the digital music revolution."*

From the early days of the Apple II and the Macintosh, Steve Jobs had always understood that ease of use through superior software design was the best way to sell hardware. But

Jobs attempting to do more than sell more desktop computers, he wanted to create a mobile device for music. As described above, usable MP3 players were already on the market but the sales numbers were disappointing. Steve Jobs analyzed that there was "plenty of evidence from the MP3 players already out there that consumer electronics makers don't know diddly about software." (Young and Simon 2005) So Jobs set his proven talents onto developing an appliance that combined software and hardware into an easy-to-use device.

The development team had the job to create an elegant device that could hold 1,000 songs on a hard disk. Assembled from off-the-shelf components, the new player was, in essence, a small computer with a 5 GB hard drive from Toshiba. Its novel scroll-wheel user interface allowed quick access to all songs. The iPod seamlessly integrated with the iTunes software on the desktop (Levy 2006). In November of 2001, the white iPod was launched at a price of $399, making it significantly more expensive than competitive products. Trade press reactions to the announcement were mixed, with analysts fearing that (Fried 2001):

> *"Apple may take some heat for entering the consumer electronics market, which typically has lower profit margins than Apple gets from its computers."*

Due to the high price and because it was originally only available to the community of Mac users using the iTunes software, initial sales were good but not overwhelming. In July of 2002, a Windows version became available. The Windows version of iTunes never became popular for Windows users per se (i.e. those that didn't have iPods) but this strategic move enabled PC users to buy iPods. In the fourth quarter of 2002, Apple sold 200,000 iPods. The iPod was already vastly outselling the company's core product, the Macintosh. According to industry reports, profit margins on the iPod at almost 50% were much higher than in Apple's other business segments. Apple Inc's overall gross margins have been between 26% and 30%, already high compared to Dell Computer's 18% (The iPodObserver 2005).

Completing the Solution

In 2003, Apple completed the total customer solution in regard to accessing and buying music by launching the iTunes Store. The iTunes software connects via the Internet to the iTunes Store, where users can purchase and download songs in a digital file format to their computers and synchronize these with the iPod device. In an industry ravaged by conflicts around Napster and Kazaa, the iTunes Store offered an easy solution to consumers to legally purchase and transfer music to their mobile players. Apple wanted to make it easy on consumer's wallets also by offering songs at $0.99, further reducing the incentive to steal music. Apple reportedly doesn't earn much money on the sale of songs, instead relying on the high gross margins on selling hardware to turn a profit.

In addition, there are currently over 4,000 accessories offered to complement the different iPod models – ranging from carrying cases to stereo-substituting loudspeakers.

The iPod has even enabled new media forms. A new format called podcasts, spoken segments similar to talk radio shows but downloadable in MP3 format, can be easily subscribed to via the iTunes synchronization feature.

Expanding the Scope

Recently, Apple has been working on expanding its reach beyond mobile media consumption. In March 2007 Apple launched an appliance that uses the iTunes software to stream content into users' living rooms onto the television set, a so-called media extender. As described in the section on single-feature marketing, the product called "Apple TV" underneath is a simple desktop computer that has only limited functionality opened up to the user. It connects wirelessly to the user's desktop or laptop computer and streams content like photos, music and videos onto the TV. Apple TV uses the iTunes Store to offer media content like single episodes of popular TV shows that are then streamed to the TV (Apple Inc. 2007b). With this appliance, Apple is proposing a simple solution for the long unsolved video-on-demand business model. It thereby also enters into the market territory for digital video recorders described in the section below. The Apple TV, though, still relies on the user's computer as the "central command headquarters" for media usage. Meanwhile, a new competitor, the start-up company Vudu, has announced an appliance for video-on-demand that is completely stand-alone (Stone 2007).

Additionally, Apple is responding to the threat posed to its iPod product series through cell phones containing MP3 players by announcing the iPhone, due to be launched in June 2007 (Apple Inc. 2007b). This deviates somewhat from the appliance model and it remains to be seen whether Apple can transfer the iPod's success into the cell phone market. If so, Apple will have gone full circle from music on a general-purpose computer to music on a dedicated device back to music on a, albeit smaller and mobile, general-purpose device.

Summary

Overall, the iPod exemplifies how the appliance model can bring tremendous business success. Apple Inc. has turned from a computer company into a consumer electronics company, albeit using its software knowledge heavily to triumph over more hardware-oriented companies. Originally built around a software solution, the iPod exploits most of the economic drivers described in Chapter 3:

Economic driver	iPod
Hardware commoditization	The iPod has a Toshiba hard drive with 5 GB of storage sourced by Apple for significantly less than $100 in 2001 (Levy 2006)
Open source software	Not applicable
Margins gained by vertical integration	Gross margins of ca. 50%, compared to 20% of hardware manufacturers
Less vulnerability to software piracy	iPod's features were designed to make illegal copying of music harder. iTunes software itself is free, therefore piracy was not an issue here.

Ease of Use	Seamless customer experience, easy installation, download of music, synchronization, access to music via scroll wheel
Battery life	Despite challenges of the trade-off between battery life and performance before launch, the first iPod launched with max. 10 hours of battery life.
Single-feature marketing	Portable music player (not a phone, not a PDA); newer product generations are watering down this aspect with calendar/contact features and iPod photo/video.
Disintermediated distribution	Music download via online iTunes Store; iPods can be bought online and through traditional retail channels
Prestige	45% of buyers attribute purchase to the coolness of the product (cf. Figure 10 in Section 3.3.3.3.5)

Overall, Apple is using its knowledge about software platforms to build an entire family of appliances (iPod, Apple TV, iPhone) that each perform a specific function and provide high added value by themselves but also can be linked with one another to provide even more value through seemless media consumption.

4.1.2 Personal Navigation Devices

This section shows how dedicated mobile navigation devices launched by software companies have achieved commercial success. These companies successfully combined hardware and software and thereby drove adoption rates in the navigation market (Göldi et al. 2006a).

Until a few years ago, mobile satellite navigation was limited to sophisticated commercial and military applications. Today it has become a consumer-oriented technology with strongly increasing adoption rates. In recent years, dedicated mobile devices for navigation have emerged. They are often referred to as "Personal Navigation Devices" (PNDs). While the first generation of mobile navigation devices were used in the marine and hiking markets, today the largest market segment is automotive navigation devices.

A case study on TomTom, a leading manufacturer of navigation solutions, is described below to show how a software company can become a successful player in the consumer space by aligning the necessary components like hardware, marketing and distribution. TomTom's example shows that if successful, an appliance company can achieve gross margins that rival those of hybrid software/service firms and achieve even higher profitability. For instance, TomTom's gross margin in 2006 was 42% (TomTom NV

2007a). TomTom's main competitor Garmin even achieved a gross margin of 50% (Garmin Inc. 2007)

4.1.2.1 The Market for Navigation Devices

Since the mid-1990s many consumers have used route-planning applications on the Internet such as Mapquest and Google Maps for their day-to-day navigational needs. To add the convenience of automatic spoken and visual route guidance and to eliminate the need for advance planning, more and more consumers are opting for GPS navigation systems for use in their cars.

The distinctive value-add of navigation solutions today is delivered by proprietary navigation software and map data. The functionality of navigation can be delivered in a variety of formats (appliance, embedded software, software-only). The market for automotive navigation is currently segmented as follows:

- Built-in navigation systems in luxury and mid-range cars
- Dedicated mobile navigation devices for in-car use
- Software packages (for use on PDAs, mobile phones, laptops etc.)

The largest providers of map data worldwide are Teleatlas, a Dutch company, and NAVTEQ, a U.S. company. These map data providers license their data to either the navigation software vendors or the hardware navigation unit manufacturers, depending on who brings the bundle to market. Currently, global navigation is provided through the only fully operational global navigation satellite system the United States NAVSTAR Global Positioning System (GPS) (National Space-Based Positioning, Navigation, and Timing Coordination Office 2007). The European Union's Galileo system is scheduled to start operation at the end of 2010 (Commission of the European Communities 2006). In 2005, over 40 million Global Navigation Satellite System receivers were shipped (ABI Research 2006).

4.1.2.2 Consumer Adoption of Navigation Software

Navigation software has been adopted at widely varying rates worldwide. The market is most developed in Japan, where 50% of new cars are equipped with built-in navigation systems (Nakahara 2004). In Europe, built-in adoption rates have grown significantly and are expected to rise from 12 % of new cars sold in 2005 to 20% in 2010 (SBD 2006). Revenues of navigation system manufacturers are currently increasing worldwide at about 20% per year (Strategy Analytics 2005). Built-in navigation in cars is still considered an optional feature in nearly all serially produced cars. Because of the high prices of EUR 1,500 to EUR 2,500 for the built-in navigation option, adoption rates have not increased as much as vehicle manufacturers would like (Göldi et al. 2006a).

A more inexpensive way to get almost the same navigation functionality are personal navigation devices (PNDs). In Europe, early-majority consumers are rapidly adopting this technology. At retail prices from EUR 200 to 800, these windshield-mounted devices have experienced significant market growth: in 2004, 2.6 million of these devices were sold in Europe, increasing to 3.8 million in 2005 and 10.1 million devices in 2006 (Canalys.com 2006; Canalys.com 2007). The United States is lagging the trend towards adoption of built-in navigation and PNDs. While in 2005 only 800,000 PND units were sold, this number increased to 2.9 million in 2006 (Canalys.com 2007). Figure 17 shows an overview of the worldwide markets for navigation devices.

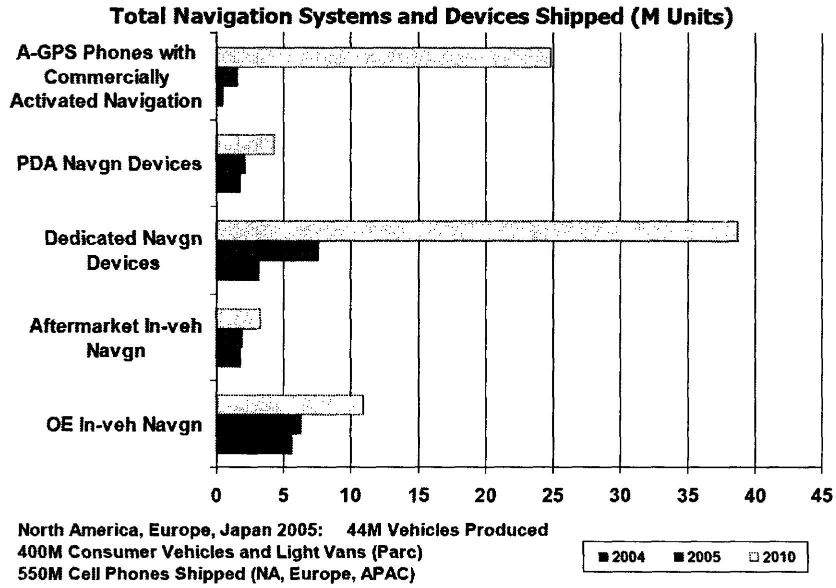

Figure 17: Worldwide market for navigation devices 2004, 2005, 2010 (Blight 2006)

In terms of the technology adoption cycle, recent European market data suggests that PNDs have reached the mainstream early majority. This data shows that PNDs are bought by all segments of the population, with 31% bought by households earning net less than EUR 2000 per month, 34% bought by households earning net between EUR 2000 and 3000 per month, with the remaining 35% bought by those earning more than EUR 3000. The age groups buying PNDs seem to confirm that navigation devices have "crossed the chasm": 34% are younger than 39 years, 40% are between 40 and 50, while 26% are over 60 years old (Polifke 2006).

4.1.2.3 Emergence of Personal Navigation Devices

As described in the consumer example in Section 3.3, until 2004 most navigation software available to consumers was sold separately or together with a multi-purpose PDA. Consumers had to purchase an additional GPS receiver and an SD memory card, and they had to install the software on the PDA themselves.

The market data in Figure 17 shows that PDA navigation devices have fallen behind in market growth and are expected to stagnate in the future. Some manufacturers have given up on the PDA segment completely. For example, Dell recently announced the discontinuation of its Axim PDA (Fried 2007).

When dedicated devices that were easy to use and had a moderate price appeared in 2004, they quickly gained in consumer interest. The value proposition for consumers of a dedicated device is expressed tersely by an industry analyst (Canalys.com 2006):

> *"Consumers have flocked to transferables in droves. They typically present a simple proposition, in an optimized form factor, at attractive price points. They are*

> *easy to demonstrate and sell. As a result, they have stolen a lot of retail shelf space away from handhelds, which has hastened the latter's decline."*

This is one of the examples where successful appliances can crowd out a general-purpose computing device.

Currently, there are several categories of players in the PND market:

- **Pure software players**: e.g. Navigon, Destinator, Map& Guide, ALK

  These players sell their software to hardware manufacturers like Acer, Mio, Medion etc. who then bundle them into offerings

- **Dedicated devices sold by software players:** e.g. TomTom, Navman, ViaMichelin

  These are companies that developed hardware capabilities but are originally software companies.

- **GPS technology companies:** e.g. Garmin, Magellan (Thales)

  These companies were technology pioneers in the GPS field, first supplying devices to the military and aviation industry.

- **Late-comer consumer/car electronics players:** e.g. Sony, Becker, Blaupunkt, Packard Bell, Pioneer, VDO Dayton

  These companies are traditionally strong in consumer or car electronics and have recently jumped on the PND bandwagon, mostly by sourcing software from the pure software players mentioned above.

The dedicated devices sold for navigation are mostly devices manufactured in Taiwan (e.g. by HTC) and other Asian countries. Technically, the hardware required is a commodity PDA running Linux or Microsoft CE, occasionally equipped with extra buttons for easy access to certain features.

Although PNDs' market success has been built on their single-feature marketing, competitive pressure has recently led many companies in the segment to increase the feature set of the devices to foreign language phrasebooks, MP3 players, hands-free phone dialing and other features unrelated to the original designated use of the device. Analysts see this trend as more of hindrance to adoption than a progression towards a multi-purpose device (Wood and Aksoy 2007):

> *"Converged devices present a lesser opportunity to expand demand and use of GPS devices and services. In order to capitalize on this niche interest, vendors must proceed with caution. They [PND vendors] must not attempt to create an all-in-one device; they must focus on creating devices that provide excellent GPS services."*

4.1.2.4 Case Study – TomTom

TomTom is an example of a software company that was moderately successful selling software and became an overnight hit when it started bundling its software in an easy-to-use consumer device. The appliance model has allowed TomTom to achieve high gross margins and uniquely differentiate itself from other players in the market.

Today, TomTom N.V. is a Dutch-based globally active company selling mobile navigation devices and mobile navigation software in 20 countries. In 2006, TomTom sold 5 million units, up from 2.2 million in 2005 and 680,000 in 2004. TomTom expects to sell between 7 and 8 million units in 2007 (TomTom NV 2007a). The company's revenues increased from only EUR 8 million in 2002 and EUR 39 million in 2003 to EUR 192 million in 2004, EUR 720 million in 2005 and EUR 1.36 billion in 2006. TomTom had gross margins of 42% and a net income of 16% in 2006 (TomTom NV 2007a). Quoted on the Amsterdam Stock Exchange since May 2005, TomTom currently has a market cap of EUR 3.5 billion (TomTom NV 2007a).

Company development

TomTom was founded as the software company Palmtop in 1991 by Peter-Frans Pauwels (cf. interview transcript in the Appendix) and Pieter Geelen, who had both recently graduated from Amsterdam University. Their goal was to develop applications for the first generation of handheld computers.

After many years of moderate success with business and consumer applications for Psion, Palm and Microsoft-based devices, the founders of TomTom hired several senior managers from Psion, including Harold Goddijn, the current TomTom CEO and former CEO of Psion and Mark Gretton, former CTO of Psion, to drive their business forward. Also, Corinne Vigreux, an early member of the management team, brought connections to retail distribution channels to the competence portfolio at TomTom.

TomTom's big software hit was the TomTom Navigator, launched in 2002. The TomTom team soon realized that software installation, the external GPS device needed and the less-than optimal PDA user interface which required a stylus were large hindrances in mainstream adoption of consumer navigation. It therefore became Gretton's task to build a hardware team in the UK. This team was to design an all-in-one navigation product, the TomTom GO, which was launched in spring of 2004. The hardware itself was sourced from an Asian manufacturer that produced TomTom's custom design. In addition to this hardware know-how, TomTom knew it needed to become a consumer brand with a sophisticated product management strategy. They therefore hired Alexander Ribbink, former vice president of brand development at Mars Inc, in November 2003 to drive the brand marketing of TomTom (TomTom NV 2007b). Today, he is TomTom's Chief Operating Officer (COO).

All these efforts combined helped TomTom launch a nicely designed dedicated device whose touch screen interface is still unrivalled in its user-friendliness almost three years after market introduction.

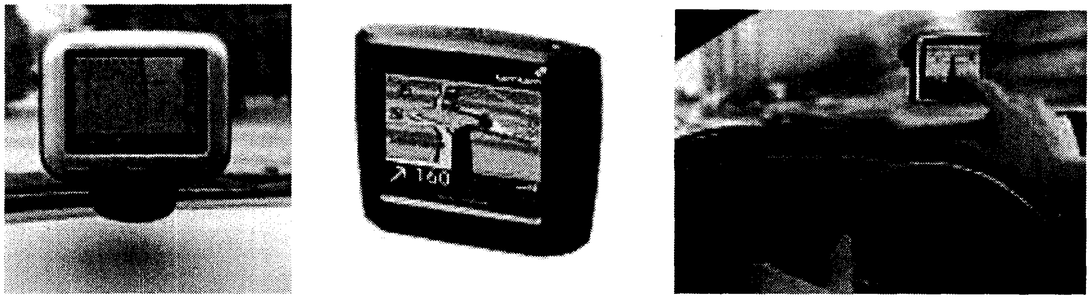

Figure 18: TomTom products GO and ONE (TomTom NV 2007b)

Building company competencies

Looking back at his company's development, Peter-Frans Pauwels described the following steps of competence-building at TomTom (cf. interview in the Appendix):

- Custom software: during the first years, TomTom programmed software for individual customers to specifications.
- Standard software: TomTom made PDA applications sold under the Psion label and received royalties for the software. Psion shrink-wrapped and distributed the software.
- Own brand: TomTom's management realized that they could get more of the margins if they created their own brand. This also meant creating a distribution network. Corinne Vigreux was responsible for building this network because she had previous experience in retail distribution.
- Create a package: TomTom decided to make it easier for consumers to use their software by bundling off-the-shelf accessories like memory cards, GPS mouse, dashboard mount for the PDA, cigarette lighter adapter and maps into one package.
- Design own hardware: Mark Gretton's hardware team and an industrial design consultancy helped from TomTom's unique look and feel that combined the ease-of-use of the software with ergonomic hardware design. This step also involves building a product management team to coordinate the hardware development with market needs.

Market success

Nr.	Brand, Model	PG	EU 6 Sales Value %	EU 6 Price EUR
1	SONY,KDL-32U2000	LCD/PLASMA	0,8	1.164
2	SONY,KDL-32V2000	LCD/PLASMA	0,7	1.781
3	APPLE,MA146/MA002 IPOD	DIG. PORT VIDEO	0,7	295
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
7	SONY,KDL-32S2010E	LCD/PLASMA	0,6	1.408
8	APPLE,IPOD NANO	MP3	0,5	171
9	PANASONIC,TX 32 LXD 60	LCD/PLASMA	0,5	1.422
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
11	APPLE,IPOD NANO	MP3	0,5	216
12	SAMSUNG,LE-32 S 71 B	LCD/PLASMA	0,5	1.015
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
14	SAMSUNG,LE-32 R 74 BD	LCD/PLASMA	0,5	1.278
15	SONY,KDL-40 V 2000	LCD/PLASMA	0,5	2.621
16	SONY,KDL-26 U 2000	LCD/PLASMA	0,5	914
17	LG,32LC2R	LCD/PLASMA	0,4	906
18	PANASONIC,TH-42PX60E	LCD/PLASMA	0,4	2.161
19	APPLE,IPOD NANO	MP3	0,4	140
20	SONY,KDL-32S2000E	LCD/PLASMA	0,4	1.382

Figure 19: Top-selling consumer electronics in the biggest 6 European countries in September 2006 by sales value (Polifke 2006)

Figure 19 shows how successful TomTom has become in the consumer electronics space. Only Sony's LCD flat-screen TVs and the Apple iPod command a higher share of the sales volume generated in September 2006 in the 6 largest European countries.

Summary

Economic driver	**TomTom PNDs**
Hardware commoditization	Asian manufactured hardware with custom casing that uses flash memory or hard drive (GO 700). Pauwels describes that the TomTom product was only even made possible by the strong decline of prices for memory cards and the increase of PDA performance (cf. interview Pauwels in the Appendix).
Open source software	TomTom uses Linux OS in its appliances to save on software licenses.
Margins gained by vertical integration	TomTom's gross margins of 42% are relatively high compared to other consumer electronics companies. Apart from map data, TomTom pays almost no other license fees. All other costs are hardware-related.

Less vulnerability to software piracy	The functionality and usability delivered by the mobile hardware makes piracy of TomTom software much less attractive.
Ease of Use	TomTom is renowned for its easy to use touch-screen interface, unrivalled even three years after introduction.
Battery life	TomTom GO originally had a battery life of 4-5 hours, but all TomTom appliance products come with a car charger.
Single-feature marketing	Navigation is the one single feature why customers buy PNDs. Like other companies, TomTom is also watering down this single-mindedness by offering hands-free dialing, music players etc.
Disintermediated distribution	Early in the company history TomTom gained popularity and built its brand through online sales. This lowered the hurdle to the mainstream for TomTom but the company spent considerable efforts building retail competence step-by-step. Today, TomTom's products and especially the add-on functionalities are available online, but TomTom PNDs are sold primarily through traditional retail channels.
Prestige	Car navigation is still early enough in the adoption cycle to bring "bragging rights" to buyers of PNDs.

4.1.3 Digital Video Recorders

Entertainment and media consumption in the home has become increasingly digitized in the last decade. Digital music, video, photos and other media can be viewed and listened to on the PC, but the location of the PC in the home office is not ideal for most users. Therefore, consumer electronics manufacturers makers have worked on bringing more computing power into the living room. Because the interface needed to be simplified, many appliances are appearing in this market.

While the first generation of digital entertainment devices, such as DVD players, was relatively simple and hardware driven, the latest generation of these devices is often complex and offers powerful features. In many cases, devices can be networked and customized to a user's preferences.

Software is the major value-adding factor in most of these new home entertainment devices. Many use entirely proprietary software, but increasingly more open general-purpose components are deployed. For example, Blu-ray disc players run the Java platform and many set-top boxes are being deployed with the Windows-based Microsoft

TV Foundation Edition operating system. Also, open source systems based on Linux (cf. case study TiVo below) are gaining in popularity.

Software has been particularly relevant in the development of the device category of digital video recorders (DVRs). The industry example below highlights how the appliance model has been used in the DVR market and which business models have turned out to be successful. The case study on TiVo shows how a software-based company successfully launched a home appliance but is now (partially) returning to software-licensing (Göldi et al. 2006a).

4.1.3.1 The Market for Digital Video Recorders

A DVR is a device that records TV or video content directly to a hard disk or similar storage medium. The first consumer DVRs were launched in 1999 (Kohler 2006). DVRs can be programmed to record broadcasted television or can rewind or pause during live viewing. Because the devices can store dozens or even hundreds of hours of content, consumers can conveniently record many episodes of their favorite TV programs and view them at a later time. This time-shifting capability, as well as the possibility to skip TV ads when watching a show, have led to a rapid market penetration. According to the Consumer Electronics Association, rapid growth has let the share of U.S. households that own a DVR rise from 18% to 25% between 2006 and 2007 (Consumer Electronics Association 2007a). This share is (conservatively) forecasted to increase to 34% in 2011 (Best and Wood 2006).

Three categories of DVRs can be found on the market:

- Stand-alone systems, e.g. TiVo or ReplayTV.
- PC-based systems, e.g. Microsoft Media Center.
- Set-top boxes (for cable or satellite TV) with built-in DVR capabilities.

Between these categories software platform approaches and component integration vary widely. One technical difference is the connection to electronic program guides or use of the European Video Programming Signal (VPS) (Schmidt 2007).

Stand-alone DVR Systems

The first generation of DVRs were stand-alone systems, i.e. complete devices that offer all necessary DVR functionality, but are independent from the distribution medium (such as cable or satellite). Stand-alone DVRs and DVD players are the technological successors of the analog video recorder. This type of DVR in 2006 had a market share of approximately 13% (Best and Wood 2006).

The market leaders were TiVo (manufactured by TiVo, Inc.) and Replay TV (owned by Japanese electronics company D&M Holdings). Both vendors were marketing hardware devices under their own brand and each used a proprietary software platform. End of 2005, Replay exited the stand-alone market to concentrate on software again.

In Europe and Asia, Samsung, Panasonic and other consumer electronics manufacturers also sell DVRs that instead of electronic program guides use the VPS system that was originally put in place for analog video recorders.

PC-based DVRs

DVR solutions based on standard PC hardware with a built-in TV tuner card are sold in the high-end market. In 2006 14% of DVRs in the U.S were PC-based (Best and Wood 2006). These systems offer the advantage of having all the usual features of a PC in addition to the DVR. They are therefore often also used as home PCs.

The most popular software used on these PCs is Microsoft Windows XP Media Center Edition. It contains DVR software and other pre-configured software modules for digital media consumption. Media Center is sold only in bundles with specialized PC hardware. Most big PC hardware manufacturers currently sell Media Center PCs, but also several smaller high-end vendors offer specialized Media Center products. Other software vendors also sell DVR solutions for Windows PCs, Macintosh and Linux-based systems.

All of these systems require relatively versed users and a computer-friendly home setting (keyboard, mouse etc.).

Set-Top Box DVRs

Cable and satellite TV companies have begun to integrate DVR functionality into the set-top boxes required for digital TV subscribers. These products are marketed as add-ons to customers' cable or satellite subscription. For example, Comcast offers a DVR product for an additional $10-$12 fee per month (Mossberg 2006).

The strong established distribution channels of cable and satellite operators have made this type of DVR very successful in a short time frame. Cable and satellite DVRs now account for 74% of the U.S. market (Best and Wood 2006).

Some of the software vendors active in this market are:

- TiVo offers its DVR software for OEMs, used e.g. in DirecTV set-top boxes.
- Microsoft's TV Foundation Edition operating system with DVR functionality is often used for set-top boxes.
- Digeo, a company owned by Microsoft co-founder Paul Allen, offers its Linux-based Moxi platform directly to cable operators.
- Scientific Atlanta, recently acquired by Cisco Systems, manufactures set-top boxes based on its PowerTV operating system that are distributed by cable companies.

These vendors all allow their platform to be customized to the needs of the individual cable operators. They aim to increase the functionality of the platform by encouraging third-party applications that can be developed using a software development kit (SDK).

These companies use very different business models. While some concentrate on software (Microsoft), others sell their software exclusively bundled with their own hardware (Scientific Atlanta), and some use hybrid models that combine software licensing with services (TiVo, Moxi).

DVR Business Models

The different types of DVR devices are sold based on very different business models. The following table summarizes these models (Göldi et al. 2006a).

Device Type	Acquisition cost for consumer	Subscription costs	Openness
Stand-alone	Free (for low-end models) to several hundred dollars.	Approx. $15/month	Program guide limited to provider, but open to several program sources
PC-based	High (cost of a high-end PC)	Mostly free	Very open, can be used as a general-purpose PC
Set-top box	Free or very low price	Included in cable/ satellite subscription, or low additional fee	Limited to cable or satellite provider's program offering

PC-based solutions use a traditional product-oriented model, cable and satellite providers focus on a pure subscription model, and vendors of stand-alone solutions typically use a mixed model with some subscription elements.

It is currently hard to determine the financial success of the different business models. Apart from TiVo (cf. case study below) financials are not published for this product category. Market shares, however, manifest the success of the fully integrated appliance model. Stand-alone and cable/satellite DVRs currently command an 87% market share that is expected to remain constant over the next few years despite the integration of DVR functionality in the new Microsoft Windows Vista.

4.1.3.2 Case Study – Tivo Inc.

U.S.-based TiVo Inc. was founded in 1997. In 2006, the company had revenues of $259 million, with gross margins of 33% and a net loss of $48 million (TiVo Inc. 2005). Despite relatively attractive gross margins (compared to hardware or consumer electronics manufacturers), the nine-year-old company is still not profitable.

TiVo Inc.'s main product TiVo is a custom-manufactured hardware appliance that is a stand-alone digital video recorder with an integrated program guide. TiVo devices are connected to a central TiVo server that supplies the electronic programming guide via a modem line or a broadband Internet connection.

Technically, the TiVo appliance is a low-end general-purpose computer with a Motorola PowerPC or MIPS processor and a modified Linux kernel as the operating system. As in most other appliances, the main value-adding DVR application is proprietary software (Barton 2006).

TiVo Inc.'s marketing strategy has been to sell the hardware at low prices and earn recurring subscription revenue from the electronic programming guide service (TiVo Inc. 2006).

After early successes with a proprietary appliance that built significant brand recognition, TiVo Inc. has decided to increase market penetration by pursuing a licensing strategy – albeit to other appliances, not for general-purpose computers. In the course of this strategy TiVo has licensed its software to manufacturers such as Philips, Sony, Pioneer and Toshiba for their own hardware platforms. In the near future, TiVo's technology and user interface will power the DVR functionality in some of cable and satellite TV operators' (e.g. DIRECTV and Comcast) set-top boxes (TiVo Inc. 2006).

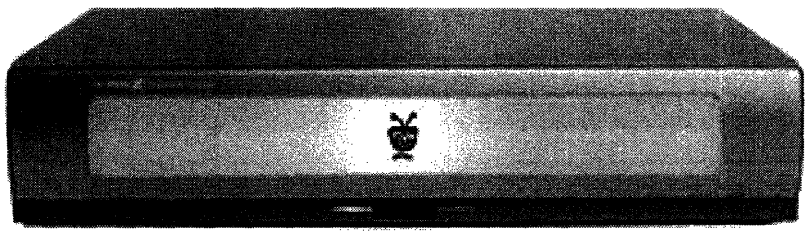

Figure 20: TiVo DVR device

ReplayTV, TiVo's main competitor, announced in 2005 that it would exit the hardware business and concentrate on selling its PC-based DVR platform (ReplayTV 2005). ReplayTV had pursued a similar strategy as TiVo but was less successful. ReplayTV's appliance was based on a modified Linux operating system, but the new software-only product will run on Windows.

In comparison, both TiVo and Replay came to the market at the same time with nearly identical products but different business models. While Replay sold its hardware at a relatively high price with a free program guide, TiVo charged for the guide but subsidized the hardware. TiVo generated hardware revenue of $89 million in 2006, but had $112 million in hardware costs (TiVo Inc. 2006). This implies that TiVo's ability to generate recurring revenues was essential to success.

Summary

While TiVo succeeded with an appliance-based model that included a major service component and is progressing to licensing, ReplayTV failed with the pure appliance model, had to exit the hardware business completely and is now concentrating on software.

TiVo's strategy to license its software platform will likely destroy some of the margins its gets from selling a complete solution but will, at least for some time, enable recurring revenue. Given the large market hurdles for a small company where distribution channels are dominated by monopolistic cable operators, this was probably the most pragmatic decision. TiVo's development shows why the appliance model can at times not be sustainable for software companies.

Economic driver	**TiVo DVR**
Hardware commoditization	TiVo uses commodity Motorola or MIPS-based hardware.
Open source software	TiVo uses a Linux-based kernel for its hardware appliance.
Margins gained by vertical integration	Gross margins of ca. 33%, compared to approx. 20% of hardware manufacturers.
Less vulnerability to software piracy	Several factors make the TiVo software very resistant to piracy: it is encapsulated in the appliance, the programming service can't be accessed by pirated software and the typical location in the living room (no mouse, keyboard etc.) makes piracy inconvenient.
Ease of Use	TiVo's user interface is easy to learn and has appeal to customers that previously did not use video recorders. It is easier to install and use than PC-based DVRs.
Battery life	Not applicable because TiVo is not a mobile device.
Single-feature marketing	TiVo was marketed originally as a device for TV content recording. In the past years, more features like music and photo center have been added to compete with PC-based media centers. But this may have watered down TiVo's core value proposition.
Disintermediated distribution	TiVo's success has been mitigated by the lack of disintermediated distribution due to the near-monopolies of the cable/satellite operators.
Prestige	Prestige is not a factor typically associated with TiVo ownership – apart from the novelty gadget status when it was first introduced.

4.1.4 Speech Recognition for Toys

This section intends to show that even fairly small companies can bundle hardware and software into a package that increases their margins significantly. While small companies might be not have the resources to launch into mass consumers markets by themselves, a step-by-step approach can facilitate market entry and, as shown above in TomTom's case, later large-scale success.

Speech recognition software uses algorithms to automatically identify the linguistic content of speech signals (Rabiner and Juang 1993). Speech recognition is a technology that has been hailed as soon-to-be ubiquitous for many years, but adoption has been slower than expected. The advances of recent years have allowed speech recognition to be deployed in a number of productive uses such as in interactive voice response systems (IVRs), cars, mobile phones or office dictation.

4.1.4.1 Market for Speech Recognition

In 2006, the market for speech recognition technology reached $1 billion, increasing 100% in two years (Borzo 2007). The overall market is divided into many subsectors. The most important are (Hong 2006):

- Telephony: software for served-based systems for call center automation, email reading, directory assistance etc. The software is generally accessed via the telephone network. The total telephony speech recognition market was estimated at $600 million in 2006 according to Opus Research, with the speech recognition software licenses only constituting $117 million according to the Gartner Group. The balance is made up of dialog and other application software as well as professional services needed for deployment.
- Embedded: software for voice command applications in cars, mobile phones, toys, consumer electronics and industrial machinery. The software is embedded into the devices. This market in 2006 was worth approximately $125 million according to Datamonitor.
- Desktop: software for uses like dictation (legal, medical, other professional and home use) or language learning. The software is installed on a PC.

Among the companies selling speech recognition software are Nuance Communications Inc., with $270 million in speech-related revenue currently the market leader (Nuance Communications Inc. 2006), as well as IBM and Microsoft.

Speech recognition software is part of one layer needed to deploy a speech-enabled system. Figure 21 shows the different components and layers.

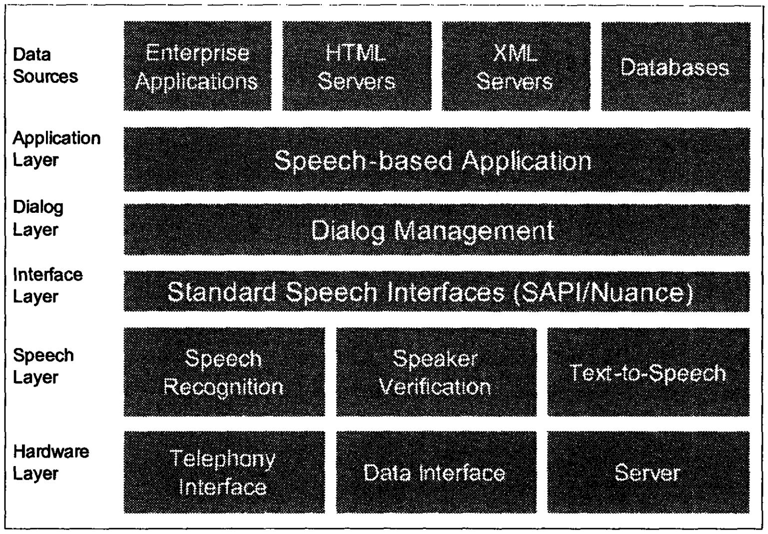

Figure 21: Layers of a speech-enabled system (SVOX AG and Hein 2001)

The current complexity of the different components makes the introduction of an appliance difficult. Most companies sell extensive professional services or work with integrators to be able to offer a full solution to customers.

The enabling-software components of these solutions have often suffered from price pressures during the last years. The following case study illustrates how a small company combined hardware and software expertise to escape these competitive pressures.

4.1.4.2 Case Study – Sensory Inc.

Sensory Inc. makes speech recognition for embedded applications such as toys and price-sensitive consumer electronics devices. The Silicon Valley-based company is located in Sunnyvale, California. It was founded in 1994 by Todd Mozer, his brother Mike Mozer and father Forrest Mozer. The company was financed by venture capital investors and currently has around 30 employees with offices in Portland (Oregon), Vienna, Tokyo and Hong Kong (Sensory Inc. 2007).

As a relatively small company, Sensory has been able to capture the market for speech-enabled toys almost completely. For example, the popular toy "Furby" was released in 2005 in its second version speech-enabled with Sensory's technology referred to as "emoto-tronics." Sensory's RSC-4128 IC product provides the function for motor controls and speech recognition and synthesis, so Furby can communicate in his own language called "Furbish"or in one of seven other languages. Furby uses facial expressions and body motions to display his emotions (Meisel 2005).

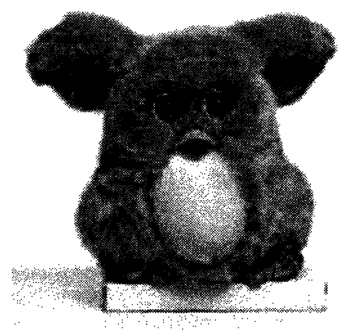

Figure 22: Hasbro's Furby toy with Sensory chips

The company has achieved market leadership by making their own integrated circuits (ICs), one example is shown below in Figure 23. Sensory is currently the only speech technology company to make their own integrated circuits. This is economically feasible because Sensory uses the fabless semiconductor model, i.e. they design their own chips while outsourcing the fabrication to a specialized semiconductor foundry.

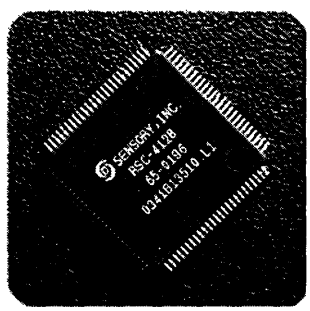

Figure 23: Integrated circuit produced by Sensory Inc.

Competitors in this field have to battle with several handicaps because they do not have the integrated knowledge of both technologies. Usually, a speech recognition software vendor and a chip manufacturer together produce a speech chip. This has the following disadvantages:

- Speech software is usually not designed with chips in mind and therefore needs more resources
- Semiconductor companies do not have their ICs optimized to run speech recognition and synthesis algorithms
- Both together don't have specialized knowledge of the requirements of the toy industry.
- Both players have to command a margin for their services

Because Sensory's CEO Todd Mozer had previous experience both in the semiconductor and speech technology fields he was able to build competencies for both of these areas in the company from the beginning (cf. interview with Todd Mozer in the Appendix). Sensory, for example, not only provides ICs for speech recognition but these can also be used as microcontrollers to control motor and other functions of the toy. This saves the toy manufacturer the cost of buying another chip and extends the battery life of the toy because one chip consumers less resources than two. In addition, Sensory offers easy-to-use integration tools to customers to engineer the speech application, saving manufacturers labor costs by reducing the interaction needed.

Due to these cost savings Sensory can command higher prices in the very price-sensitive toy market. Also, Sensory has higher margins because through vertical integration it was able to capture most of the margin that formerly went to the semiconductor company. In addition, by selling a hardware component, Sensory was able to avoid the crisis most speech technology vendors experienced after 2001. Their customers knew that their price could never sink to zero.

Summary

Sensory has not completed all the steps to becoming an appliance company but their business strategy shows how a small company can combine hardware and software knowledge to produce a higher-margin product that is cost-saving and easier to use for its customers.

Economic driver	**Sensory Speech Recognition ICs**
Hardware commoditization	Many steps of chip fabrication and even design can today be outsourced where formerly high investments were necessary
Open source software	Not applicable
Margins gained by vertical integration	Gross margins of ca. 40-60%, compared to 20% of hardware manufacturers, achieved because customers can save labor and additional hardware.
Ease of Use	Toy manufacturers do not have to deal with 2 different parties
Battery life	By using one chip that contains speech recognition and controls other functions, manufacturers have better life than if they had to use two chips.
Single-feature marketing	Speech-enabled toys
Disintermediated distribution	Not applicable
Prestige	Not applicable

4.2 Enterprise Applications

Just as in the consumer segment, appliances have started emerging in many subsegments of the enterprise market. Some appliances, for example Cisco routers, have been in the market for many years – even though they may not have been always perceived as appliances they do derive most of their value from the software, not the hardware.

Most appliances in the enterprise software segment have been used in areas that are important for the enterprise but the requirements do not vary widely from enterprise to enterprise. In areas where the needs are easily standardized, more appliances started appearing in the late 1990s as the price for computing power continued to drop. The industry examples in this section show that how appliance companies have successfully competed against software-only solutions in the enterprise software market.

4.2.1 Network Appliances

Computer networks offer a number of examples for successful use of the appliance model. Modern computer networks need a number of components to function securely. Among these are hubs, routers, firewalls, switches or more recently, network-attached storage.

While today these components are often delivered in proprietary boxes, most of them could just as well be located on general-purpose computing platforms. The reason why many companies in this market are offering their products in dedicated boxes is not just because they have optimized the hardware for their applications but also because it allows them to lock-in customers' willingness to pay and makes them less vulnerable to software piracy. The following section describes network routers as an example of a network appliance.

4.2.1.1 The Market for Routers

A router links two or more computer networks by buffering and transferring data packets between them (Gawer and Cusumano 2002). Routers were pioneered at Stanford University in California. The first multiprotocol router software was written in 1980 by Stanford staff researcher William Yeager (Dix 2006). Routers experienced explosive growth as computer networks and the use of the Internet has expanded in the last 15 years. Today routers may be combined with security features such as firewalls. While router software does not necessarily need dedicated hardware and most network operating systems include all necessary software to perform routing, most routers today are sold as a bundle of hardware and software.

In 2006, the market for basic enterprise routers was worth $ 6.1 billion. Cisco Systems dominates this market with a share of 75% of world revenue. Huawei Technologies and Juniper Networks Inc. were the second and third largest vendors of enterprise routers in 2006 according to Infonetics Research (Hickey 2007).

4.2.1.2 Case Study – Cisco Routers

Cisco Systems Inc. was founded in 1984 out of Stanford University in California by Sandra Lerner and Leonard Bosack. Today, Cisco has revenues of $28.5 billion (2006) and around 55,000 employees. Even though the majority of Cisco's current revenues come from other sources, Cisco's original product was router software (Cisco Systems Inc. 2007).

Cisco has always primarily sold its routers in packaged boxes but the main differentiator is its IOS software platform. This software is sold by packaging different sets of features depending on the customer segment. Even as Cisco has expanded into other segments of networking, the IOS software has remained the core of its platform strategy (Gawer and Cusumano 2002). As many other appliances, Cisco's routers today are easily configured through a web administration tool.

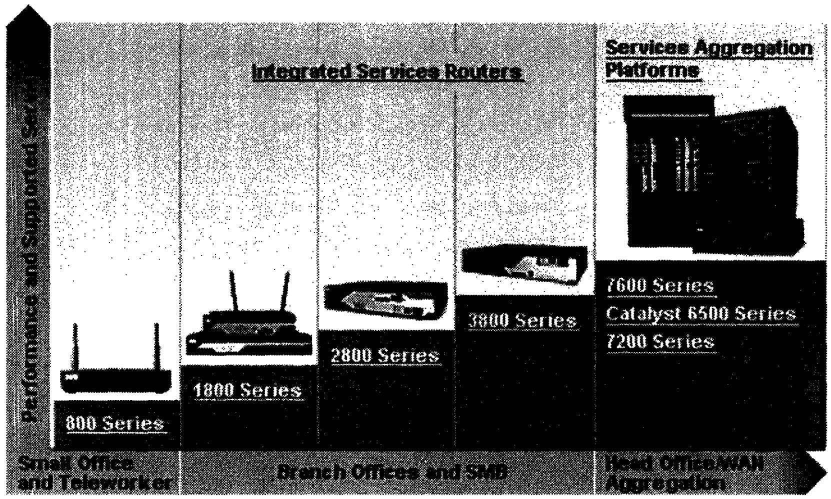

Figure 24: Cisco router product series

Software is the key ingredient in Cisco's routers, hardware packaging is often used purely for price differentiation. This becomes evident in the fact that while the hardware used is not exactly the same for lower-end routers compared to higher-end routers (cf. Figure 24) in some cases it is similar enough that software "hacks" will allow knowledgeable users to "upgrade" their routers to higher functionality. For example, Linksys WRT54G wireless routers (Linksys is a subsidiary of Cisco) with a retail price of approximately $50 are designed for home office use. They can be converted into the functional equivalent of a powerful, highly configurable $600 router by simply changing the firmware to an open source version (Pash 2006).

In an article discussing Cisco's counter-measures against counterfeit routers, Gartner analysts noted where the real value is derived (Hochmuth 2007):

> *"Increasingly, Cisco's emphasis is on its IOS software, and the tighter controls it has exerted over it are an effort to squeeze more value out of the hardware Cisco sells — whether new or old. Value is in the software for Cisco."*

This becomes overt in Cisco's recent market action. They introduced a software license manager tool and a new distribution system for IOS software. All new Cisco routers are shipped with full-load versions of IOS, which must be activated via a software key to unlock whatever level of features the user purchased.

For more than 20 years, Cisco has successfully used its lead in software and combined it with hardware to dominate the router market it created. It can be hypothesized that while now Cisco is being (marginally) threatened by counterfeiting and other copyists, the appliance model was a significant part of the strategy that allowed Cisco to remain dominant for such a long period. By offering customers the convenience of an appliance that included the best software in the segment, Cisco could defend its high gross margins of 65-70% (Cisco Systems Inc. 2007).

Economic driver	**Cisco Routers**
Hardware commoditization	Cisco's high gross margins and its increasing focus on protecting its software points to decreasing role that hardware plays in its pricing structure.
Open source software	Cisco's low-end consumer brand Linksys sells routers running Linux.
Margins gained by vertical integration	Overall, Cisco maintains high gross margins of 65 to 70%.
Less vulnerability to software piracy	The fact that Cisco only recently has had trouble with counterfeit routers points to its low vulnerability in the past due to combination with hardware.
Maintenance fees	Cisco routers do not have maintenance fees but customers can sign service contracts for support.
Predictability of total cost of ownership	Since routers have nearly always been bundled, a cost calculation is difficult. It can be assumed though that installing and servicing a software router has less predictable costs.
IT personnel costs	Cf. category above

4.2.2 Email Appliances

Since the 1990s electronic messaging, referred to as electronic mail or email, has become standard in most large and mid-sized organizations. In 2006 there were estimated to be over 170 billion messages sent per day, 70% of which were bulk unwanted messages called spam (Radicati Group 2006).

Currently, standard corporate email that has not been outsourced requires email server software, email client software, operating system software, email servers and storage resources. This requires intensive personnel resources for installation, integration and administration.

In this section an industry example is described where an appliance company has taken on two of the largest companies in the IT sector, Microsoft and IBM, with a strong cost-savings value proposition.

4.2.2.1 The Market for Corporate Email Software

Enterprise e-mail provides messaging capabilities for large organizations. The products in this sector cover both internal messaging, often based on a proprietary protocol, and external messaging, typically based on open Internet protocols.

The market for enterprise e-mail currently has around 400 million seats (Radicati Group 2006). More than 250 million of these are used by large companies with more than 1000 employees. The total number of seats is expected to grow to 456 million in 2009. The revenues generated in this sector of the software market are currently \$2.5 billion (Radicati Group 2006). By 2009 the market size is estimated to increase to \$3.3 billion.

The enterprise e-mail market is strongly concentrated with over 90% of revenues being generated by the top three players. Market leaders in the market for corporate email software are:

- Microsoft: In the enterprise e-mail server market, Microsoft leads the market with Microsoft Exchange Server. A new version, Exchange 2007, has just been introduced. Microsoft's market share is currently 42% in North America (Frost & Sullivan, Inc. 2006). Microsoft is also dominant in the e-mail client market with Outlook, part of the Microsoft Office suite.
- IBM: The Lotus Notes/Domino product suite ranks second in the enterprise e-mail market commanding a market share of 35%.
- Novell: The GroupWise e-mail product has a market share of 13%, ranking third. Novell is positioning the GroupWise 7 product as a more cost-effective and open alternative to Microsoft Exchange.
- Others: All other market participants together have 10% market share.

One of the fast-growing companies in the "others" category is Mirapoint Inc., a company that offers an email appliance to mid-sized and large companies.

4.2.2.2 Case Study – Mirapoint Inc.

Mirapoint Inc., based in Sunnyvale, California, was founded in 1997. It offers several product lines of email appliances as part of an "appliance-based secure messaging infrastructure". Mirapoint proposes the appliance as an alternative to conventional email server software. In 2006 Mirapoint served 115 million mailboxes worldwide, making it the second largest provider worldwide in terms of seats. But due to the low cost of its solution the company's revenues were only approximately \$50 million in 2005 (Cain 2006).

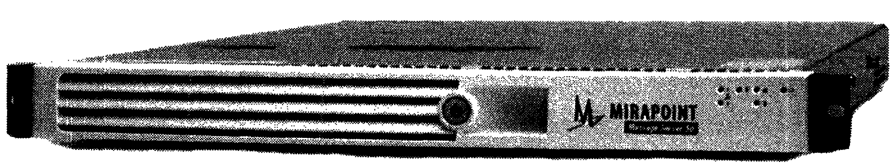

Figure 25: Mirapoint Message Server 50

Mirapoint's appliance products wrap email application, operating system, hardware infrastructure and storage resources into one package, bundling hardware with messaging and security software. Mirapoint markets the following advantages of its solution (Mirapoint Inc. 2007):

- 99.999% uptime and leading performance,
- fast deployment and simple management,
- secure, hardened operating system with no known exploits,
- network-based software updates,
- single vendor support

Mirapoint's entry-level product, Message Server M50, provides a complete e-mail solution for companies with between 100 and 500 users. The basic appliance with 50 user licenses costs around $14,000, and additional web-mail users can be added for as little as $12.90 (compared to $67 per Microsoft Exchange user). This cost advantage has enabled Mirapoint to gain market share in the Internet Service Provider (ISP), educational, and government market. Large corporations such as Ford, Mitsubishi Motors or British Telecom are also Mirapoint customers (Mirapoint Inc. 2007).

According to a study by Radicati, a Mirapoint deployment has a very strong total cost advantage over conventional vendors, since not only capital expenditures are lower, but maintenance is far simpler due to Mirapoint's all-in-one appliance model. The study comparing the Mirapoint Message Server email appliance to Microsoft Exchange showed that Mirapoint needed 80% less administration staff (Radicati Group 2004).

In analysts' perspectives the company's offering is primarily attractive to customers that require limited functionality and do not need deep email and calendar functionality. According to these reports Mirapoint's products are primarily attractive to the government and educational sectors (Cain 2006).

Summary

Economic driver	**Mirapoint Email Appliance**
Hardware commoditization	Mirapoint's appliances are rack-mountable industry-standard servers with a custom front panel.
Open source software	Mirapoint uses some open-source software in its stacks, for example the SpamAssassin spam filter.
Margins gained by vertical integration	There is no public information available about Mirapoint's gross margins, but the prices of its appliance package suggest high gross margins.
Less vulnerability to software piracy	Mirapoint's appliance is delivered in the hardware box, deterring software piracy.
Maintenance fees	Mirapoint point charges a one-time cost for its hardware plus an annual support fee that includes remotely administered updates.

Predictability of total cost of ownership	Due to a fixed hardware price plus a fixed support fee, total cost becomes more predictable.
IT personnel costs	The Radicati study cited above claims 80% reduction in IT personnel administration costs.

4.2.3 Search Appliances

From year to year corporations generate more and more digital files that are often not even printed and filed any more. For compliance, documentation and knowledge management purposes, companies are seeking mechanisms to organize and access these digital files efficiently. Retrieving the information contained in these files easily can create value for enterprises by saving employees time in searching or recreating information. One option to access this information is to index all documents and databases automatically and search them via a search engine.

4.2.3.1 The Market for Enterprise Search

Until recently, search was a function incorporated into individual enterprise software applications but most corporations invested little into enterprise-wide search capabilities (Andrews 2006a). Because the market for search across enterprise databases and applications is in an early stage, the market size is still limited. The market for enterprise search was estimated at $760 million worldwide for 2005 by IDC but is expected to rise to $1 billion in 2007 (Feldman 2006). In 2006, the search software market grew by 30% (Feldman 2007, 1-23). Currently, small vendors like Thunderstone, Index Engines, Arexera, Convera, Autonomy, Endeca, and FAST are still dominant but large vendors like IBM, Oracle, Microsoft and Google are entering this market. In 2006 Forrester Research assessed FAST and Autonomy as market leaders with IBM and Google in fourth and fifth place (Brown 2006).

Most of the above mentioned vendors use a traditional software deployment model requiring the software to be installed on general-purpose hardware within the company's network. Several vendors have adopted the appliance model. Among these are Convera, Thunderstone, Index Engines, Arexera and Google.

The following case study shows how Google as a consumer-oriented company is attempting to enter the enterprise software market via the appliance model.

4.2.3.2 Case Study – Google Inc.

As the Internet has evolved, search engines have become an effective means to access vast quantities of data available. In the last decade Google has become the leader in the Internet search market by indexing Internet documents. Google uses its Googlebot indexing software to crawl the public Internet and store the contents found in indexes.

For Internet searches, Google finances its operations very profitably with an advertising-based business model. In 2006, Google Inc. had total revenues of $10.6 billion (Google Inc. 2007b). Google is now trying to enter the enterprise segment in several areas. For example, Google is offering the Google Desktop for Enterprises, a PC-based personal search engine (Google Inc. 2007d). This expansion of its product offerings is congruent with the company goal of "organizing the world's information and making it universally accessible and useful" (Google Inc. 2007a).

Because much of the world's information is stored behind corporate firewalls Google had to adapt its Internet-based search approach to corporations' concerns about letting their internal information be accessed and indexed via the Internet. Instead of using the traditional packaged software approach, Google's answer to the enterprise need to keep search in-house was the Google Search Appliance (GSA). It was introduced in February 2002 in the "pizza-box" format as shown in Figure 26 (Perez 2006). The Google Search Appliance's value proposition is to use Google's existing technology to enable companies to navigate the troves of corporate data with the same ease and speed as querying the Internet.

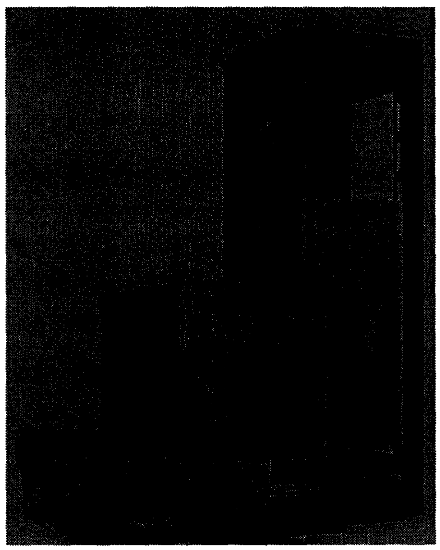

Figure 26: Google Search Appliance

The GSA is a self-contained unit that can be mounted on a standard server rack, plugged into a corporate network, and easily configured via a web browser. It provides an uncluttered, easy-to-use interface to finding the company's documents and information. The appliance supports important features such as security enforcement that prevents users from accessing documents that they have insufficient privileges to view (Göldi et al. 2006a). The appliance does not only index simple unstructured data such as Word documents. It can also index a variety of databases such as Oracle, MySQL and IBM DB2 (Google Inc. 2007e). Starting at a fixed cost of $30,000, a Google Search Appliance box can be purchased that is capable of indexing up to 500,000 documents.

Google has used the GSA to expand its enterprise search appliance offering even further. Google OneBox, a software extension for the GSA that enables users to securely access even more types of information on the corporate network, was released in 2006. Google has partnered with Cisco, Cognos, Employease, Netsuite, Oracle, Salesforce.com, and SAS in order to access their applications (Google Inc. 2007e).

The appliance also supports customization such as specifying corporate synonyms to consider for search queries. The search interface for the GSA is also easily integrated with Google Desktop for Enterprises (which allows search for files, email, IM chats, web pages viewed etc.) and the traditional Google search engine. This allows users to use a single common toolbar to access personal, corporate, and public information. The familiarity of its interface gives Google a strong advantage, even though it lags behind some competitors in regards to categorizing information (Claburn 2006).

Analysts believe Google to be emerging as a very strong player in this field due to its strong brand name and reputation for excellence. Gartner forecasts that Google will sell more than 40 percent of new unit licenses in the enterprise search market by the end of 2007 (Andrews 2006b). However, according to Gartner, Google's big advantage of simplicity is also its biggest limitation in more complex situations. Although Google's enterprise business is growing rapidly, this segment still is very small compared to the company's core business. In the third quarter of 2006, the enterprise business contributed only about one percent of Google's revenues (Google Inc. 2006).

Summary

This case study shows how a consumer software player can use the appliance model to transfer its brand name into the enterprise segment and win the trust of corporate IT buyers concerned about the security of their internal information.

Economic driver	**Google Search Appliance**
Hardware commoditization	The GSA is a rack-mountable industry-standard server in custom yellow or blue Google-branded "pizza-box".
Open source software	The GSA runs a special version of the Linux operating system (Google Inc. 2007e).
Margins gained by vertical integration	Google's gross margin in 2006 was 60% (Google Inc. 2007b). The GSA does not improve Google's margins but provides entry into the enterprise software market.
Less vulnerability to software piracy	The GSA shields Google from uncontrollable copying of its software. In general, Google is very concerned about secrecy regarding its algorithms and its business strategy.
Maintenance fees	The base price for the GSA includes a 2-year support plan that includes software upgrades (Perez 2006). Google doesn't disclose pricing for maintenance fees after that.
Predictability of total cost of ownership	Due to a fixed price for a prescribed maximum number of documents total cost of the GSA becomes more predictable.

IT personnel costs	Google markets the GSA as easily installable, configurable and maintainable to save IT personnel time.

4.2.4 Caching Appliances

When many Internet users want to access identical content at the same time, this causes network congestion and a slow-down for all users. For example, if a news item released by a broadcasting company regarding an airline crash or an upset in a sports championship, potentially millions of users would try to connect to the news website to get information. The resulting congestion of users would not only result in long wait times for the queries, but could slow down the local or regional Internet. One solution for companies or institutions that have a large volume of simultaneous traffic on their Internet sites is to use a Content Delivery Network (Göldi et al. 2006a).

According to the Computer Desktop Encyclopedia, a Content Delivery Network (CDN) is (Freedman and Morrison 2007):

> *"A system of distributed content on a large intranet or the public Internet in which copies of content are replicated and cached throughout the network. When content is replicated throughout the country, or throughout the world, users have quicker access to it than if it resides on one Web site."*

By placing multiple copies of the content at the network "edge" of an enterprise or internet service provider, a CDN distributes the content so that deadlock can not occur in any one critical point. A CDN can be established by installing CDN software on the web servers that contain the content and on the web servers where the content is to be cached (i.e. stored). Technically, this does not require dedicated hardware.

4.2.4.1 The Market for Content Delivery Networks

The CDN market was estimated at $600 million dollars in 2006 (Palumbo 2006). This includes direct bandwidth accounts that do not go through a CDN specialist for distribution (i.e. Yahoo, AOL and Real Networks).

Some of the companies active in this segment are:

- Akamai Inc.
- Limelight Networks
- Mirror Image
- VitalStream

Some of the players are concentrated on subsegments such as streaming audio or video, software, media or entertainment downloads.

The clearly dominating company in this segment is Akamai Inc. who pioneered the technology and early on chose to use the appliance model to distribute its software. The following case study on Akamai shows how the appliance model as a distribution method for a service-based business model enabled a start-up software company to dominate its industry niche.

4.2.4.2 Case Study – Akamai Inc.

Akamai Inc. was founded in 1998 out of the Massachusetts Institute of Technology (MIT) by applied mathematics professor Tom Leighton and graduate students Daniel Lewin, Jonathan Seelig and Preetish Nijhawan in Cambridge, Massachusetts. Akamai is publicly traded on NASDAQ with a current market capitalization of $7.1 billion. In 2006, Akamai had revenues of $429 million with a gross margin of 76% (Akamai Inc. 2007a).

Service Business Model

The founders of Akamai developed innovative algorithms for caching content on the Internet. By caching and syndicating content through a Content Distribution Network Akamai delivered a solution to the impeding problems of network congestion described above. To deliver this caching and traffic management ability, Akamai chose a service-based business model. Customers pay based on their utilization of bandwidth and expect the service to allow their users faster access to the company's website. Akamai was then faced with the decision of how best to build a large service network without incurring very high upfront investment costs.

Appliance Model to Roll Out Network

Akamai chose the appliance model as a method to distribute its software to Internet Service Providers (ISPs) and to its corporate customers. Akamai developed and shipped its entire solution as a hardware appliance, the EdgePlatform. But Akamai doesn't sell the EdgePlatform appliance, it only sells a service.

Because it saved ISPs money by reducing bandwidth costs and because the appliances needed only rack space but little installation, Akamai was able to co-locate its servers at ISPs for free. In its marketing literature, Akamai describes the ease of integration (Akamai Inc. 2007b):

> *"Because these technologies are deployed as managed services, you can integrate with the Akamai EdgePlatform in just days, for instant global reach and capacity with no assembly required."*

Within two years of launch, Akamai was able to build a network of over 4,000 EdgePlatform web servers worldwide. Akamai was thus able to speed the adoption of the core software technology by enabling users to install a prepackaged appliance.

Figure 27: Akamai EdgePlatform servers

To date Akamai has shipped over 20,000 servers across 71 countries and now has its own secure service platform. Today, Akamai delivers between 10-20% of all Web traffic. It claims that the EdgePlatform is the world's largest distributed computing platform. Akamai monitors the operation of the platform continuously from its Network Operations Command Center (Akamai Inc. 2007b).

Summary

Akamai is the least typical but an intriguing example of the appliance model. Without actually selling appliances, Akamai has made use of most drivers of the appliance model to build a proprietary secure platform for content distribution around the world.

Economic driver	**Akamai EdgePlatform**
Hardware commoditization	The EdgePlatfrom is a rack-mountable industry-standard server with an Akamai front logo.
Open source software	Akamai doesn't publish if it uses open source software on its Akamai appliances.
Margins gained by vertical integration	Akamai's gross margins of more than 75% are enabled by their choice of delivery model.
Less vulnerability to software piracy	The EdgePlatform servers belong to Akamai and are shielded from piracy because they allow no outside access to the software.
Maintenance fees	Not applicable because the appliances belong to Akamai.
Predictability of total cost of ownership	Customers only pay for the managed service according to their usage.
IT personnel costs	ISPs and corporate customers have little personnel costs due to the EdgePlatform appliance.

5 Getting There: The Transition from Selling Software to Offering an Appliance

Depending on which variant of the appliance model a company chooses, managing the transition from selling software to offering an appliance can be a daunting task for the management of a company. This section attempts to give some perspectives on the key aspects of such a transition.

Once the decision for the appliance model has been made (using e.g. the guidelines of the strategic decision funnel presented in Section 3.4) the software company has to look at a number of issues. The most important dimensions are:

- Changing the mental model
- Building company competencies
- Product design and marketing
- Assembling the value chain

5.1 Changing the Mental Model

An essential building block of becoming an appliance company involves changing the mental model that software companies have of themselves. While a software company thinks about products from the "inside out" i.e. starting from the application software outwards to the physical layer, hardware equipment manufacturers and consumer electronics companies usually think about products "outside in", i.e. about their physical characteristics first (cf. Figure 28).

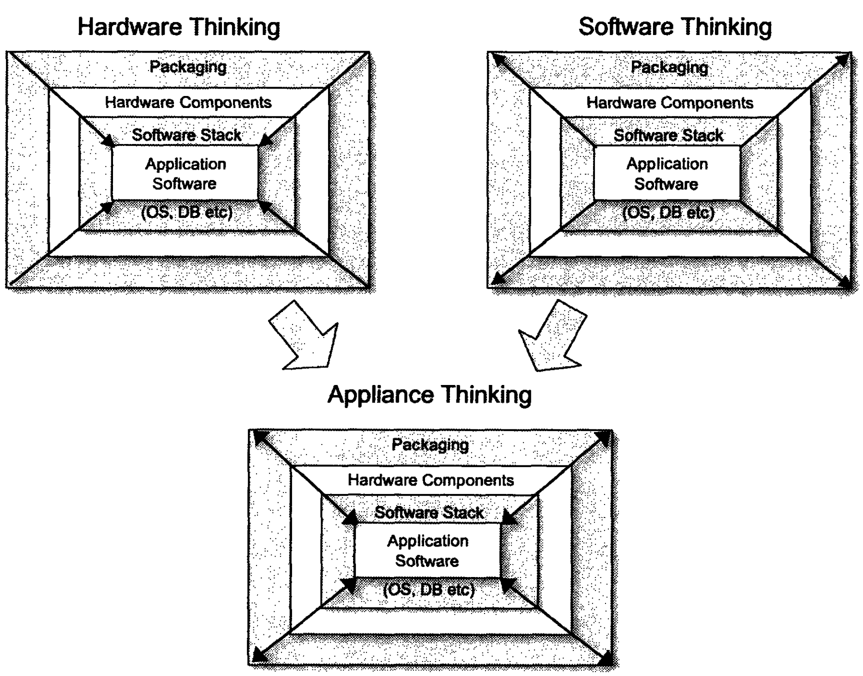

Figure 28: Integration of "outside-in" and "inside-out" thinking

To fully exploit the competitive advantage that the appliance model brings, companies have to learn to think holistically and iteratively about their product development processes. Only an integrated view of the whole product will allow the company to develop a truly useful appliance.

5.2 Building Competencies

As mentioned in the strategic decision funnel in section 3.4, most software companies will have to build additional competencies in order to become appliance companies. Described below are the transition process and well as the hiring and organizational aspects of competence building.

5.2.1 Transition Process

Companies that want to adopt the appliance model have to derive a transition plan from being a software company to being an appliance company. This plan involves a step-by-step process that may take a decade (as in the case of TomTom described in section 4.1.2.4) or may be parallelized when starting an appliance company from the outset (as in the case of the movie appliance start-up Vudu mentioned in section 4.1.3).

For a **consumer software** company, the transition from software to an appliance can look like the process shown in Figure 29:

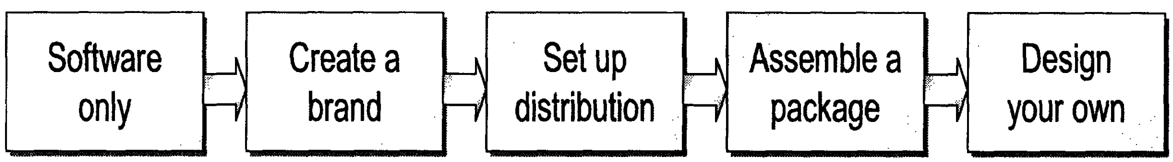

Figure 29: Transition process for consumer software

Starting from a pure software product, companies entering the appliance space can first work on creating a consumer brand for their software. If a loyal following can be generated e.g. via Internet sales, it will be easier to convince other distribution channels to pick up the shrink-wrapped software product. If this works, the next step could be to assemble a bundled package from commodity components available on the market (e.g. a PDA, GPS, other accessories). Designing your own hardware is the final step in becoming an appliance company. For all of these steps, the company has to find the right staff with fitting knowledge.

The process shown above is a risk-minimizing approach that may not be feasible in a fast-moving market or for a limited-time market opportunity. In those cases, parallelization or leapfrogging of steps (such as the shrink-wrapped software distribution) can be a better strategy.

For an **enterprise software company**, the transition from software to an appliance can look like the process shown in Figure 30:

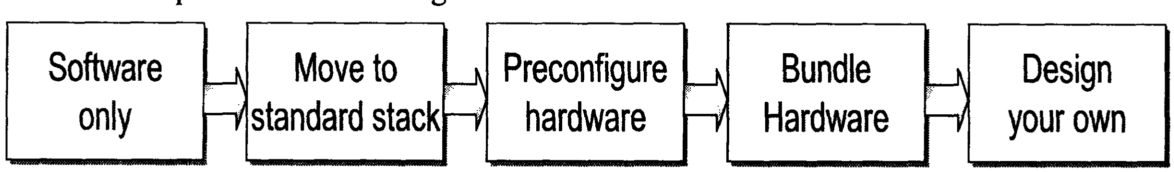

Figure 30: Transition process for enterprise software

The process differs slightly because enterprise software companies have less flexibility in creating the entire package. Most enterprise appliances will have to use commodity hardware that can be integrated into the IT infrastructure of customer enterprise easily. This makes the gap in competence easier to bridge and also, as described below in the description of the value chain, easier to outsource.

5.2.2 Hiring and Organizational Design

When building an appliance company, management can't content itself in hiring additional experts on hardware design or retail distribution to fill the competence gaps, the staff required will need overlapping competencies in order to be able to work together. Ideally, members of the management team have competencies in several fields (like Todd Mozer of Sensory Inc. described in section 4.1.4.2).

In regards to organizational design, it is important that interdisciplinary teams be formed. Only teams with mixed competencies will be able to design an appliance that is more than the sum of the individual components.

5.3 Product Design and Marketing

The strategic design of an appliance product and the marketing approach are one of the key determinants of success or failure. Some of these decisions may seem very technical in nature but they have strong effects on market acceptance and profitability.

5.3.1 Technical Configuration

An appliance company will need to determine many technical details of the appliance, for example:

- Open source components (e.g. Linux) or commercial standard software (Microsoft etc)? This decision determines licensing requirements and cost.
- Which parts of the software stack are proprietary? Proprietary elements enhance differentiation but also add costs.
- Which hardware components need to be sourced? Will the company design its own hardware? Again, differentiation and cost need to be traded off.

More exotic technical configurations will require different competence levels and quality of staffing than commodity-based solutions.

5.3.2 Differentiation and Competition

How will the appliance differentiate itself from other players in the market? The management team will need to decide on this issue before proceeding. Will the product compete head-on with an established software product (or another appliance) or will the product enter into an underdeveloped niche? For example, Apple decided not to include a DVD player or a TV tuner in its new Apple TV product so that it wouldn't be put into DVD or DVR product category by retailers and consumers (Mac Zone 2007). The appliance's feature set will determine its potential buyers but also its cost structure.

5.3.3 Product Line Differentiation

The appliance model allows companies to tangibly differentiate between customer segments by choosing different hardware packaging or feature sets for each segment. For example, the Apple iPod nano costs $200 for the 4GB version but $250 for the 8GB version (Amazon Inc. 2007c). The current wholesale price for the additional 4 GB of flash memory is less than $5 (Economic Policy Department 2007). Apple thus has more than 1000% margin on this feature because music aficionados are willing to pay to be able to store more music on the otherwise identical device. Similarly in the enterprise segment, section 4.2.1.2 describes the price differentiation through different packaging in Cisco routers.

These examples show how crucial it is for appliance companies to differentiate their product line by customer segments. Higher margins on top models may allow for more aggressive pricing and better market penetration in more price-sensitive customer segments.

5.3.4 Integration

Another crucial market decision is how to structure and design the integration with standard interfaces. Despite their relative self-reliance appliances don't exist in a vacuum. Choosing the right interfaces or format can be very important for market acceptance.

For example, RIM's Blackberry has to interact with all mobile operators' networks; TomTom's newer PNDs have to support traffic data formats; Google's Search Appliance has to be able to search not only Adobe PDFs but also more exotic formats; the iPod had to choose which formats of audio compression to support. In the enterprise sector, integration of an appliance in the management tools of the data center can be a significant cost factor (Enck, Dawson, and Phelps 2006).

All of these examples underscore the need for careful evaluation of the appliance's ecosystem before proceeding.

5.3.5 Usability And Design

Usability and design deserve extra mention in this section on product design because often the usability and for consumer devices the physical design win customers for an appliance. Software companies transitioning into the appliance sector will need to build some in-house expertise, but investments in external industrial designers and usability testers can be well worth the investment. For example, section 3.3.3.3.5 mentions how much the design of the Apple iPod influences the purchase decision. When companies hire an industrial designer or outsource other development functions it is important to develop top-level in-house knowledge to be able to integrate all functions (cf. interview with TomTom's Peter-Frans Pauwels in the Appendix).

5.4 Assembly of the Value Chain

The assembly of the entire value chain of an appliance can seem a like a very challenging task to a software company that is neither used to physical distribution nor manufacturing or inventory management.

The following value chain components can be relevant for a company entering into the appliance business:

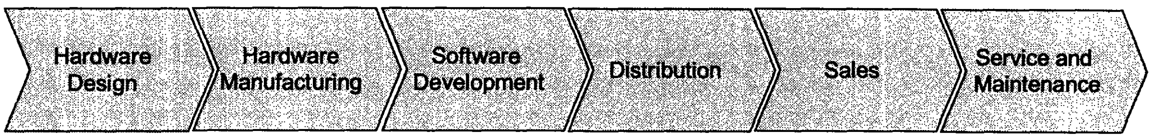

Figure 31: Appliance value chain

While software development, distribution, sales and maintenance are elements that traditional packaged software companies also are confronted with, their characteristics may change significantly during the transition to the appliance model. Hardware design and manufacturing as well as after-sales hardware service are completely new elements that the software company has to adapt to.

5.4.1 Degree of Vertical Integration

Vertical integration is described as an economic driver but also one an economic barrier in chapter three. While integration increases the share a company gets of the value created by a product it also increases the cost for the company. An appliance company therefore has to decide carefully which elements of the value chain provide so much value or are so critical to success that it has to build its own competency and capacity, and which areas can be outsourced without compromising the quality of the total customer experience.

As mentioned in section 3.3.4.1.2 the expansion of contract manufacturers to more and more services like design, transport or after-sales service have made it easier for software companies to concentrate on their core competencies.

5.4.2 Maintenance and Service

Even if an appliance company can outsource the after-sales service (which will not always be feasible in e.g. critical enterprise software solutions) the company management still has to create processes around service and maintenance that feed back into the research and development departments. For example, what will happen with company data stored on the appliances at the end of their lifecycle? How is the migration process to larger versions of the appliance or new product generations structured? Appliance companies have to find solutions for these issues because customers often base purchase decisions on these aspects in order to protect their investments.

5.4.3 Distribution

The relationship building, logistics and financial issues associated with physical distribution are foreign to most software companies. An appliance company has to become acquainted with all of the issues related to becoming listed with retailers, delivering physical products as well as managing and financing inventory. These are very cost-intensive aspects of the value chain that require much more planning and foresight than most software companies are accustomed to.

6 Beyond the Appliance Model: Virtual Appliances

While the hardware-based appliance model may have many economic advantages over the traditional packaged software model, the need for physical hardware brings a number of disadvantages (discussed in the section on economic barriers) that lower the margins for software vendors. These factors are, for example, distribution logistics and hardware maintenance.

For enterprise software applications, the virtual appliance model (introduced in section 3.2.3 on technical concepts) offers to alleviate some of those disadvantages while keeping many economic advantages. In the consumer software segment, the often-customized mobile devices make the hardware-based appliance model more resilient. But for the enterprise segment, is the hardware-based appliance model only an intermediate step before the pendulum swings back to a software-only solution?

This chapter discusses this recent trend and its implications for enterprise software companies.

6.1 Enabling Virtualization

As described in section 3.2.3 the virtual appliance model implies that vendors deliver their software as a virtual machine, packaged in a digital file. The virtual machine runs in a virtualization layer that simulates a consistent set of hardware on top of most operating systems and x86 processors. Instead of having to physically travel to the data center, mount a hardware appliance on a machine rack, and connect the necessary cables, customers can deploy virtual appliances by simply opening the virtual machine file. Because the software on hardware appliances uses commodity stacks that run on x86 machines, virtual appliances can simply be run on existing hardware. Just like hardware appliance vendors, virtual appliance vendors can modify the underlying operating system and device drivers to optimize performance and improve security.

The following table shows some of the benefits of the virtual appliance model compared with traditional packaged software and hardware appliances (Göldi et al. 2006a):

Benefits	**Traditional Software**	**Hardware Appliance**	**Virtual Appliance**
Customers			
Compatibility with existing x86 hardware	✓	✗	✓
Plug and play installation	✗	✓	✓
Easy maintenance – Simple backup/restore; non-disruptive upgrade/co-existence	✗	✗	✓
Demand-driven scalability	✗	✗	✓
Vendors			
Low cost production and distribution	✓	✗	✓
Controlled platform for building and testing	✗	✓	✓

Tighter integration to underlying OS	×	✓	✓
Optimized hardware for applications	×	✓	×

Source: Modified from http://www.vmware.com/appliances/benefits.html

For virtualization to become a mainstream phenomenon, a certain technical infrastructure and ecosystem has to be built. In the following section, two components of the virtual appliance model are described that aim to enable virtualization (Göldi et al. 2006a).

6.1.1 OS Componentization

Virtual appliances share hardware resources with other appliances. In order to save memory and CPU resources it is therefore key to include only those components of the operating system that are essential for the operation of the appliance. Only then can the virtual machine image file be minimized. This process of minimization is called OS componentization.

One of the leading new companies providing componentization support is rPath Inc.. Founded in 2005, rPath's main product allows software vendors to deploy a Linux-based application as a hardware, software or virtual appliance. Given the application, rBuilder automatically determines the minimal set of components required for the appliance and builds an image of the virtual appliance. This not only reduces the size of the image, but also improves security by reducing the number of potential vulnerabilities. Customers can deploy updates to existing appliances, merge the updates into the default image or rollback to previous images through a web-based configuration interface (rPath Inc. 2007).

6.1.2 x86 Virtualization

In order for software application vendors to construct virtual appliances, they need software that provides hardware virtual machines. The most popular virtual machines are those that provide virtualization of industry standard computers with x86 processors running unmodified PC operating systems. According to analysts at the Yankee Group, VMware, founded in 1998 and now a subsidiary of EMC, is currently the market leader in virtual infrastructure software with a 55% market share (Marshall 2006b). Competitive products are Microsoft's Virtual Server (29% market share) and XenSource's open source Xen (1 % market share).

Because of the intense competition in this area to establish an industry standard, all of the companies mentioned above now distribute their entry-level products for free. The inclusion of virtualization technology in Linux distributions and Microsoft Vista are signs that virtualization software could become a mainstream commodity product (Macehiter 2006). This is an indication that a growing technology ecosystem supports development of virtual appliances.

6.2 Economic Drivers and Barriers for Virtual Appliances

By examining the benefits and drawbacks of virtual appliances, the strength of the **economic drivers** of this new business model can be evaluated in comparison to hardware appliances and traditional packaged software:

- **Hardware commoditization:** By unbundling from the hardware, virtual appliances allow companies to procure standard x86 hardware in bulk and host multiple low utilization virtual appliances on a single physical server. Hardware costs can therefore be reduced. For vendors, though, this marks the return of the argument that software is infinitely replicable and therefore, free. By having several virtual appliances on the same hardware again, virtual appliances cannot leverage customized hardware to optimize performance. Virtualization also creates overhead and decreases the overall application performance.
- **Open source software:** Just as hardware appliances, the virtual machines also bundle open source stack components.
- **Margins gained by vertical integration:** Because of the early stage of the virtual appliances market, it is hard to assess whether these vendors can ask for the same prices as hardware appliance vendors. While they are creating economic efficiencies, it remains to be seen whether the companies can appropriate the economic rents of the value they create.
- **Less vulnerability to software piracy:** A virtual appliance is more guarded against software piracy than traditional packaged software but more vulnerable than a hardware appliance because the virtual appliance's disk image is easily copied and distributed.
- **Maintenance fees:** Because virtual appliances don't run on vendor-supplied hardware and because updates can be delegated back to the enterprise customer, the incentive to create efficient software applications that need less maintenance is reduced. Less economic efficiency is therefore to be expected in this regard from virtual appliances. On the positive side, virtual appliances simplify maintenance compared to physical hardware by enabling easy backup and restore as well as less disruptive updates or rollovers.
- **Predictability of total cost of ownership:** Just as hardware appliances, virtual appliances offer the advantage of easy configuration and no unpredictable third-party licenses. They require somewhat more administration than hardware appliances in regards to the coordination of hardware resources and also need to be installed by the customer. On the other hand, virtual appliances can be distributed over the Internet, saving time and cost for the enterprise customer.
- **IT personnel costs:** The same savings in configuration time that apply to hardware appliances apply also to virtual appliances. As mentioned above, more administration may be required for coordination of hardware resources and installation. Less time and cost is involved when a virtual appliance fails compared to a hardware appliance. Overall, it is not obvious whether hardware or virtual appliances generate lower personnel costs.

One key advantage of virtual appliances not mentioned above is the ability to dynamically scale the number of virtual servers based on utilization. This creates significant flexibility for the enterprise user that may e.g. need more financial accounting resources at the beginning of the quarter or more Internet transaction servers at Christmas time. The current trend towards utility and grid computing strengthens this advantage.

Overall, it seems that virtual appliances offer more flexibility and may optimize hardware cost, but the economic drivers are not significantly improved compared to the hardware appliance model.

A look at the **economic barriers** paints a different picture:

- **Company capabilities:** While companies do still have to build more knowledge about the stack they will use, they do not have to build physical distribution and hardware design or maintenance know-how.
- **Distribution costs:** Virtual appliances can be distributed as traditional packaged software or over the Internet.
- **Need to evolve:** While virtual appliances are also pre-packaged their delivery model could enable more flexibility to adapt software to the individual customer. Also, they can be migrated to more powerful hardware and scaled more easily if requirements grow over time.
- **Up-front investment:** No upfront investment is needed.
- **IT security:** Corporate officers may perceive IT security as slightly higher because the virtual appliances run on company-procured hardware.

In summary, for the enterprise market, virtual appliances economically benefit software vendors strongly because they remove the investment and capability barriers. This could be beneficial for small, innovative companies that want to enter the enterprise software market but would not be able to finance an appliance rollout. Apart from the gain in flexibility and scalability, software customers profit less from this new development. Because of this gap, new outsourcing companies like Network Engines (Network Engines Inc. 2007) are evolving that take virtual appliances and turn them into hardware appliances and service them for a fee.

6.3 Industry Examples of Virtual Appliances

Because literally all application software can be turned in to virtual appliances, there already are hundreds if not thousands of virtual appliances available. VMware's Virtual Appliance Marketplace alone has more than 500 virtual appliances available on website, ranging from spam filtering email gateway to network traffic analyzers.

Judging from VMware's Virtual Appliance Marketplace the most popular categories for virtual appliances are currently:

- open source OS distributions
- application/web servers
- IT administration

- networking
- security
- content management and collaboration
- databases

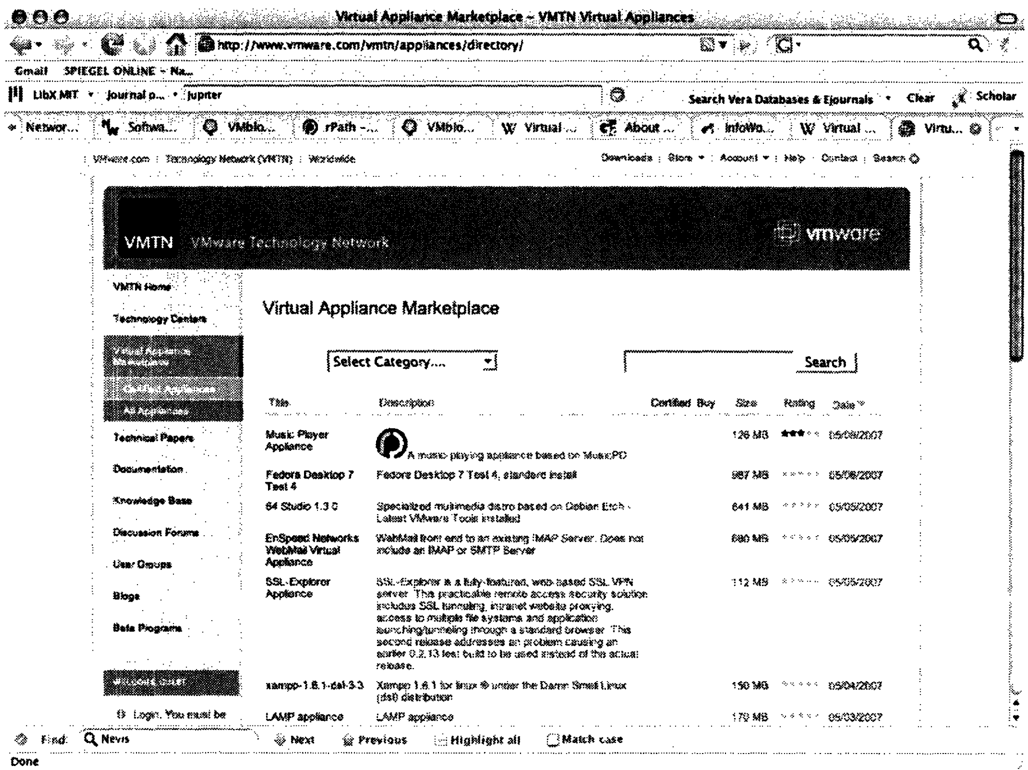

Figure 32: VMware's Virtual Appliance Marketplace

Only about 20 of the 500 appliances listed, though, are certified and available for purchase. It remains to be seen, therefore, whether virtual appliances can develop into a profitable industry segment.

7 Conclusion

"If you are serious about software, you should make your own hardware."

Steve Jobs at Mac World 2007

7.1 Summary

As outlined in the research question, this study attempted to analyze the anecdotal trend in the software industry towards an appliance business model of packaging software with dedicated hardware.

An analysis of the history of software business models showed that originally, software and hardware were always packaged together. Today, the complexity of software, the trend to open source software and the commoditization of hardware have led to economic incentives to revitalize this business model.

By looking at the economic drivers and barriers of the appliance model the study demonstrated that while software buyers, both in the consumer and enterprise software segments, are set to profit from this new trend, the equation is less clear for software companies. For software vendors, the appliance model brings opportunities for capturing additional value and increasing competitive differentiation but it also increases the costs incurred and the company capabilities needed.

The study therefore suggests a number of decision criteria for companies considering the appliance model. These are grouped into a strategic decision funnel in order to narrow down relevant questions.

To illustrate the different aspects of the appliance model, four industry examples for each the consumer and the enterprise software segment were examined by describing the market environment and the relevant economic drivers that made these appliances successful. The examples confirm that the appliance model can be applied to a number of vertical markets and there can be significant variation in implementation.

Building on the lessons learned from the industry examples, the relevant factors for transitioning or implementing the appliance were explored.

Finally, virtual appliances as a potential next step in the development of the appliance model are evaluated. For enterprise software vendors, they offer to alleviate the burden of having to deal with hardware but also offer less creative scope to design competitive advantage into the total product. It therefore remains to be seen whether virtual appliances will replace hardware appliances.

7.2 Implications

The findings of this study have implications for

- **Management of software companies**: Although this study has described some of the barriers and disadvantages of the appliance model, the current market environment in consumer and, especially, enterprise software warrants that management of software companies carefully consider new business models.

Because of the added hardware dimension, the appliance model offers additional potential to demonstrate and create value for customers by giving more control over the end user experience to the software vendor. Companies that exploit this potential can do extraordinarily well with the appliance model. Management, though, has to be aware of the increased risk associated with this new business model with regards to building company competencies and the level of investment capital needed.

- **Software customers**: This study has shown that customers can expect time and cost savings from appliances. But when making investment decisions, customers have to carefully evaluate their individual situation and compare other new business models offered. For example, for companies that have large capacity fluctuations, it may be better suited to go one step further and employ virtual appliances. Also, claims of better usability and personnel cost savings should be closely examined before making extensive capital investments.
- **Further research**: This study has collected and analyzed available public information to create a framework for closer examination of the appliance business model. It did not attempt to quantitatively measure to what extent the appliance model has been adopted by the software industry. Further specific research could focus on measuring the adoption rate of the appliance model in comparison to other new business models such as software-as-a-service or the advertising-based model. Also, as the number of companies employing the model increases, empirical research could be conducted on the financial success of appliance companies. In a broader perspective, this study raised questions about the future of our daily interaction with computing power. Will computing power diverge to many devices or will it converge in a single device or computing grid? How much of the functionality will be hidden? Will the appliance model stifle innovation by hiding generic computing power and making it inaccessible to the majority of users? Is the question of convergence or divergence irrelevant because the next step in technological development will link all of the appliances together in a platform approach? These are all questions that future research can consider.

8 List of Figures

9 Bibliography

3TERA Inc. 2007. *Company website.* Retrieved March 27, 2007, from http://www.3tera.com/

8e6 Technologies. 2005. *Web filtering appliances heat up the hardware vs. software debate (white paper).* Retrieved April 3, 2007, from http://8e6.com/docs/white_papers/hardware_vs_software.pdf

ABI Research. 2006. *Global navigation satellite positioning systems and devices.* Oyster Bay, New York: ABI Research Inc. Retrieved March 29, 2007, from http://www.abiresearch.com/product/market_research/Satellite_Positioning_Systems_and_Devices/AI

Akamai Inc. 2007a. *Company website - financial information.* Retrieved May 1, 2007, from http://www.akamai.com/html/investor/financial_info.html

Akamai Inc. 2007b. *Company website - the EdgePlatform.* Retrieved May 1, 2007, from http://www.akamai.com/html/technology/edgeplatform.html

Amazon Inc. 2007a. *Amazon web services.* Retrieved March 28, 2007 from http://aws.amazon.com

Amazon Inc. 2007b. *BlackBerry 7290 phone (cingular).* Retrieved March 31, 2007, from http://www.amazon.com/BlackBerry-7290-Phone-Cingular/dp/B0006852GS

Amazon Inc. 2007c. *iPod nanos.* Retrieved March 31, 2007, from http://www.amazon.com/iPod-Nanos-Computers/b/ref=amb_link_1/102-5020034-0952923?%5Fencoding=UTF8&node=15761351&pf_rd_m=ATVPDKIKX0DER&pf_rd_s=left-1&pf_rd_r=1XXBRYGV0MVFF9YM5T3P&pf_rd_t=101&pf_rd_p=236993801&pf_rd_i=13660271

Anderson, Chris. 2006a. *The long tail.* New York: Hyperion.

Anderson, Chris. 2006b. *The rise of "freeconomics".* Retrieved May 6, 2007, from http://www.longtail.com/the_long_tail/2006/11/the_rise_of_fre.html

Andrews, Marc. 2006a. *The future of enterprise search.* Retrieved May 1, 2007, from http://marcandrews.typepad.com/marc_andrews/2006/06/the_future_of_e.html

Andrews, Whit. 2006b. *Google for enterprise search is simple and popular, yet limited.* Stamford: Gartner Inc.

Apple Inc. 2007a. *100 million iPods sold.* Retrieved April 14, 2007, from http://www.apple.com/pr/library/2007/04/09ipod.html

Apple Inc. 2007b. *Company homepage.* Retrieved April 14, 2007, from http://www.apple.com/iTunes

Barton, Jim. 2006. *TiVo-lution.* ACM Queue, 4(3)

Best, Mark, and Andrea Wood. 2006. *Stand-alone digital video recorders: Is there a market?* JupiterResearch.

Blight, Joanne. 2006. *Presentation at the SVOX forum, November 17, 2006.* Unpublished manuscript.

Borzo, Jeanette. 2007. *Now you're talking.* Business 2.0 magazine. Retrieved April 16, 2007, from http://money.cnn.com/magazines/business2/business2_archive/2007/02/01/8398978/index.htm

Brown, Bruce. 2001. *PDA upgrades rival new models.* Retrieved March 31, 2007, from http://www.pcmag.com/article2/0,1895,79365,00.asp

Brown, Matthew. 2006. *The Forrester wave: Enterprise search platforms, Q2 2006.* Cambridge, MA: Forrester Research Inc. from http://www.forrester.com/Research/Document/Excerpt/0,7211,38355,00.html

Bulkeley, William M. 2004. *IBM strikes deal with rival Lenovo.* Wall Street Journal, December 8, 2004.

Cain, Matthew W. 2006. *MarketScope for E-mail systems* No. G00136445). Stamford: Gartner, Inc..

Campbell-Kelly, Martin. 2003. *From airline reservations to Sonic the Hedgehog: A history of the software industry.* Cambridge, Massachusetts: The MIT Press.

Canalys.com. 2006. *Mobile GPS navigation market doubles year-on-year in EMEA.* Retrieved April 26, 2007, from http://www.canalys.com/pr/2006/r2006081.htm

Canalys.com. 2007. *US GPS navigation market picks up speed.* Retrieved May 1, 2007, from http://www.canalys.com/pr/2007/r2007031.htm

Charette, Robert N. 2005. *Why software fails.* Spectrum Online, September 2005, Retrieved March 31, 2007.

Christensen, Clayton M. 1997. *The innovator's dilemma.* New York: HarperBusiness.

Cisco Systems Inc. 2007. *Company website.* Retrieved April 17, 2007, from www.cisco.com

Claburn, Thomas. 2006. Google extends search to enterprise apps, upgrades search hardware. *InformationWeek,* April 19, 2006.

Commission of the European Communities. 2006. *Communication From the Commission to the European Parliament and the Council: Taking stock of the GALILEO programme.* Retrieved April 14, 2007, from http://ec.europa.eu/dgs/energy_transport/galileo/documents/doc/comm_pdf_com_2006_0272_en.pdf

Consumer Electronics Association. 2004. *04Q1 - forecast shipments of MP3 players, U.S.* Retrieved April 15, 2007, from http://www.infotechtrends.com.libproxy.mit.edu/cgi-bin2/cif/sub_read.pl?ux=1197&quar=04Q1&04138017.htm=on

Consumer Electronics Association. 2006. *Average U.S. household now owns 26 consumer electronics products, finds CEA's annual ownership study.* Retrieved April 15, 2007, from http://www.ce.org/Press/CurrentNews/press_release_detail.asp?id=10995

Consumer Electronics Association. 2006. *Household penetration of CE products soars in 2005.* Retrieved April 15, 2007, from http://www.ce.org/Press/CurrentNews/press_release_detail.asp?id=10753

Consumer Electronics Association. 2007a. *CEA finds American adults spend $1,200 annually on consumer electronics products.* Retrieved April 30, 2007, from http://www.ce.org/Press/CurrentNews/press_release_detail.asp?id=11274

Consumer Electronics Association. 2007b. *Digital America: History of digital audio.* Retrieved April 15, 2007, from http://www.ce.org/Press/CEA_Pubs/932.asp

Cooperstock, Jeremy R. 1997. *From the flashing 12:00 to a usable machine: Applying UbiComp to the VCR.* Paper presented at the CHI 97, Atlanta. from http://acm.org.libproxy.mit.edu/sigchi/chi97/proceedings/short-talk/jrc.htm#REF94770

Crowley, Tom 2006. *Aha! what your refrigerator can teach you about selling software.* Retrieved April 8, 2007, from http://opensource.sys-con.com/read/244363_2.htm

Cummings, Joanne. 2005. *Software as appliance.* Retrieved April 7, 2007, from http://www.networkworld.com/buzz/2005/092605-softapp.html

Cusumano, Michael. A. 2004. *The Business of Software.* New York: Free Press.

Dembo, Ron. 2004. *From plug and pray to plug and play.* Risk, December 2004.

Dix, John. 2006. *Router man.* Retrieved April 17, 2007, from http://www.networkworld.com/supp/2006/anniversary/032706-routerman.html?t5

Dougherty, Dale. 2001. *LAMP: The open source web platform.* Retrieved March 29, 2007, from http://www.onlamp.com/pub/a/onlamp/2001/01/25/lamp.html

Economic Policy Department. 2007. *Macroeconomic review April 2007.* Volume IV, Issue I. Singapore: Monetary Authority of Singapore. Retrieved April 29, 2007, from www.mas.gov.sg/resource/publications/macro_review/MRApr07_secured_upload.pdf

Enck, John, Philip Dawson, and John R. Phelps. 2006. *Predicts 2007: Brace yourself for the next wave of server technology.* Stamford: Gartner, Inc.

Feldman, Susan. 2006. *Worldwide content access tools (search and retrieval) 2006-2010 forecast.* Framingham, MA: IDC Inc. Retrieved April, 2, 2007, from http://www.idc.com/getdoc.jsp?containerId=201032

Feldman, Susan. 2007. *Worldwide search and discovery software 2007–2011 forecast.* Framingham, MA: IDC Inc. Retrieved April, 2, 2007, from http://www.idc.com/getdoc.jsp?containerId=206148

Flextronics Inc. 2007. *Company website.* Retrieved April 7, 2007, from http://www.flextronics.com/en/AboutUs/Backgrounder/tabid/62/Default.aspx

Fraunhofer Institut. 2007. *History of MP3.* Retrieved April 15, 2007, from http://www.iis.fraunhofer.de/fhg/iis/EN/bf/amm/mp3history/mp3history01.jsp

Freedman, Alan, and Irma Morrison (Eds.). 2007. *Computer desktop encyclopedia.* Computer Language Company Inc.

Fried, Ian. 2001. *Apple's iPod spurs mixed reactions.* Retrieved April 14, 2007, from http://news.com.com/Apples+iPod+spurs+mixed+reactions/2100-1040_3-274821.html

Fried, Ian. 2007. *Dell stops selling Axim handheld.* Retrieved April 14, 2007, from http://news.com.com/2100-1041_3-6174546.html

Frost & Sullivan, Inc. 2006. *North American enterprise e-mail and instant messaging solution markets*. San Antonio: Frost & Sullivan, Inc.

Gallaugher, John. M. 2002. *E-commerce and the undulating distribution channel.* Communications of the ACM, 45(7), 89-95.

Garmin Inc. 2007. *2006 annual report.* Retrieved May 1, 2007, from http://www8.garmin.com/aboutGarmin/invRelations/finReports.html

Gartenberg, Michael, and Andrea Wood. 2007. *US portable media players forecast, 2006 to 2011*. New York: Jupiter Research. Retrieved March 30, 2007, from http://www.jupiterresearch.com/bin/stream/US_Portable_Media_Players_F.pdf?report=98951&file=184121&TYPE=PDF&SID=

Gawer, Annabelle, and Michael A. Cusumano. 2002. *Platform leadership: How Intel, Microsoft and Cisco drive innovation*. Boston: Harvard Business School Press.

Gilfix, Michael. 2007. *The appliance model and general purpose software stacks.* January 20, 2007. Retrieved February 25, 2007 from http://www.michaelgilfix.com/ techblog/2007/01/20/appliance-model-and-software

Gilliam, Dale. 2005. *Understanding the dominance of iPod - A consumer perspective.* Dallas: The Diffusion Group.

Göldi, Andreas. 2007. *The emerging market for web-based enterprise software.* Master's Thesis, MIT Sloan School of Management.

Göldi, Andreas, Bettina Hein, Paul Hsu, Caleb Hug, and Tim Jones. 2006a. *Appliance model for software.* Unpublished manuscript.

Göldi, Andreas, Krishna Boppana Bettina Hein, Paul Hsu, and Tim Jones. 2006b. *Challenges from new software business models.* Unpublished manuscript.

Google Inc. 2006. *Google announces third quarter 2006 results.* Retrieved April 26, 2007, from http://investor.google.com.libproxy.mit.edu/releases/2006Q3.html

Google Inc. 2007a. *Company website - corporate information.* Retrieved May 1, 2007, from http://www.google.com.libproxy.mit.edu/corporate/

Google Inc. 2007b. *Company website - financial tables.* Retrieved May 1, 2007, from http://investor.google.com.libproxy.mit.edu/fin_data.html

Google Inc. 2007c. *Docs & spreadsheets.* Retrieved May 1, 2007, from http://www.google.com.libproxy.mit.edu/google-d-s/intl/en/tour1.html

Google Inc. 2007d. *Google desktop for enterprises.* Retrieved May 1, 2007, from http://desktop.google.com.libproxy.mit.edu/en/enterprise

Google Inc. 2007e. *The Google Search Appliance.* Retrieved April 26, 2007, from http://www.google.com.libproxy.mit.edu/enterprise/gsa/index.html

Hickey, Andrew. R. 2007. *Network hardware sales up.* Retrieved April 17, 2007, from http://searchnetworking.techtarget.com/originalContent/0,289142,sid7_gci1244816,00.html

Hochmuth, Phil. 2007. *Cisco looking to counter used-gear competitors.* Retrieved April 17, 2007, from http://www.networkworld.com/news/2007/040207-cisco-channel-partners.html?page=5

Hong, David. 2006. *An introductory guide to speech recognition solutions (white paper).* New York: Datamonitor Inc. Retrieved April 27, 2007, from http://www.rightnow.com/pdf/whitepapers/Voice_Datamonitor.pdf

IDC. 2006. *Third annual BSA and IDC global software piracy study.* Business Software Alliance. Retrieved April 17, 2007, from http://www.bsa.org/globalstudy/upload/2005%20Piracy%20Study%20-%20Official%20Version.pdf

Kessler, Scott H. 2006. *Standard & Poor's industry surveys computers: Consumer services & the Internet* Volume 174, No. 38, Sec. 2. New York: Standard & Poor's.

Kohler, Chris. 2006. *TiVo tastes some sweet revenge.* Retrieved April 26, 2007, from http://www.wired.com/science/discoveries/news/2006/08/71686

Landauer, Thomas. K. 1996. *The trouble with computers: Usefulness, usability, and productivity.* Cambridge: The MIT Press.

Levy, Stephen. 2006. *The perfect thing: How the iPod shuffles commerce, culture, coolness.* New York: Simon & Schuster Inc.

Lüthje, Boy. 2002. *Electronics contract manufacturing: Global production and the international division of labor in the age of the internet.* Industry and Innovation, 9(3), 227-247.

Mac Zone. 2007. *Apple TV's real potential.* Retrieved May 6, 2007, from http://mac-zone.org/article.php?page_id=41

MacDonald, Neil, Thomas J. Bittman, Martin Reynolds, and Brian Gammage. 2006. *Findings: Not all "appliances" are appliances.* Stamford: Gartner, Inc..

Macehiter, Neil. 2006. *VMware throws next punch in virtualisation battle.* Retrieved May 5, 2007, from http://www.theregister.co.uk/2006/04/11/vmware_opens_file_format/

Marshall, Billy. 2006. *The software appliance.* August 1, 2006. Retrieved on March 28, 2007 from http://www.sandhill.com/opinion/editorial.php?id=93&page=2

Marshall, David. 2006. *Survey suggests server virtualization catching on.* Retrieved May 5, 2007, from http://weblog.infoworld.com/virtualization/archives/2006/07/survey_suggests.html

Meisel, Bill. 2005. *Will Furby be the breakthrough in speech recognition for toys?* Retrieved April 16, 2007, from http://www.tmaa.com/recent_articles/furby.htm

Mirapoint Inc. 2007. *Company website.* Retrieved April 17, 2007, from http://www.mirapoint.com/

Moore, Geoffrey A. 1991. *Crossing the chasm.* New York: HarperBusiness.

Mossberg, Walter M. 2006. The HDTV dilemma: *Pay for TiVo's recorder or settle for cable's?* Wall Street Journal, December 28, 2006

Nakahara, Tsuneo 2004. *Speech at 29th IRU world congress.* Retrieved April 26, 2007, from http://www.iru.org/index/en_events_2004_yokohama_speeches_nakahara

National Space-Based Positioning, Navigation, and Timing Coordination Office. 2007. *The global positioning system.* Retrieved April 14, 2007, from http://www.gps.gov/systems/gps/index.html

Nayak, Shivashis. 2006. *Pricing and licensing of software products and services: A study on industry trends.* Master's thesis, Massachusetts Institute of Technology, 1-116.

Network Engines Inc. 2007. *Company website.* Retrieved May 5, 2007, from http://www.networkengines.com

Niquette, Paul. 2006. *Softword: Provenance for the word 'software'.* Retrieved March 11, 2007, from http://www.niquette.com/books/softword/tocsoft.html

Novell Inc. 2007. *Definitions & glossary.* Retrieved April 2, 2007, from http://www.novell.com/partners/oem/definitions.html

Nuance Communications Inc. 2006. *Annual report 2006,* 10-K/A SEC filing

Palumbo, Paul. 2006. *CDN growth and market share shifts: 2002 – 2006.* Accustream Research iMedia Research. Retrieved April 24, 2007, from http://www.accustreamresearch.com/products/cdngrowth-02-06.html

Parker, Sybil P. (Ed.). 2004. *McGraw-hill encyclopedia of science and technology (5th Ed).* New York: The McGraw-Hill Companies, Inc.

Pash, Adam. 2006. *Hack attack: Turn your $60 router into a $600 router.* Retrieved April 17, 2007, from http://www.lifehacker.com/software/router/hack-attack-turn-your-60-router-into-a-600-router-178132.php

Perez, Juan C. 2006. *Google upgrades enterprise search device.* Retrieved April 26, 2007, from http://www.infoworld.com/article/06/09/19/HNgoogleappliance_1.html?CONTENT%20MANAGEMENT

Polifke, Arndt. 2006. *Portable navigation devices - the revolution on the dashboard.* Unpublished presentation. Gesellschaft fuer Konsumforschung (GfK).

Porter, Michael E. (1980). *Competitive strategy: Techniques for analyzing industries and competitors.* New York: The Free Press.

Prentice, Stephen, Simon Hayward, Brian Gammage, et al.. 2006. *Consumerization of IT: The Gartner analyst keynote, Spring 2006.* Stamford: Gartner, Inc..

Rabiner, Lawrence, and Biing-Hwang Juang. 1993. *Fundamentals of speech recognition.* Englewood Cliffs, New Jersey: PTR Prentice Hall.

Radicati Group. 2004. *Mirapoint message server total cost of ownership.* Palo Alto: Radicati Group, Inc.

Radicati Group. 2006. *Microsoft exchange market share statistics, 2006.* Palo Alto: Radicati Group, Inc.

Rappa, Michael. 2007. *Managing the digital enterprise.* Retrieved March 15, 2007, from http://digitalenterprise.org/models/models.html#Advertising

Raymond, Eric. S. 1998. *Goodbye, "free software"; hello, "open source".* Retrieved March 30, 2007, from http://www.catb.org/~esr/open-source.html

ReplayTV. 2005. *ReplayTV announces new strategic direction with DVR for the PC* Press release. Retrieved April 23, 2007, from http://www.digitalnetworksna.com/About/Replaytv/press.asp?ID=613

Robinson, Jon. 2007. *Why network appliances suck and what to do about it.* Retrieved April 7, 2007, from http://jonsnetwork.com/2007/03/why-network-appliances-suck-and-what-to-do-about-it/

rPath Inc. 2007. *Company website.* Retrieved May 5, 2007, from http://www.rpath.com/corp/

SBD. 2006. *Navigation (embedded & portable) report.* Northampton: SBD.

Schmidt, Ulrich. 2007. *Professionelle Videotechnik: Analoge und digitale Grundlagen, Filmtechnik, Fernsehtechnik, HDTV, Kameras, Displays, Videorecorder, Produktion und Studiotechnik* (4th Edition). Heidelberg: Springer.

Sensory Inc. 2007. *Company website.* Retrieved April 16, 2007, from http://www.sensoryinc.com/

Snyder, William R. 2006. *The software value gap is driving a wedge between vendors and customers.* Stamford: Gartner, Inc..

Soanes, Catherine, and Angus Stevenson (Eds.). 2003. *Oxford dictionary of English* (2nd rev Ed.). Oxford: Oxford University Press.

Sony Corporation. 2006. *Annual report 2006.* Retrieved April 7, 2007, from http://www.sony.net/SonyInfo/IR/financial/ar/2006/qfhh7c00000akt4k-att/qfhh7c00000akt5x.pdf

Sony Corporation. 2006. *Sony history: Walkman chapter 17/18.* Retrieved April 7, 2007, from http://www.sony.net/Fun/SH/1-18/h4.html

Stone, Brad. 2007. *Vudu casts its spell on Hollywood.* New York Times, April 29, 2007

Strategy Analytics. 2005. *Navigation market status report.* Boston: Strategy Analytics.

SVOX AG and Bettina Hein. 2001. *Company material.* Unpublished manuscript.

The iPodObserver. 2005. *iSuppli says new 30GB iPod costs $151 to manufacture.* Retrieved April 14, 2007, from http://www.ipodobserver.com/story/24024

The NPD Group Inc. 2007. *Spending on consumer technology products increased in 2006 but at a slower rate.* Retrieved April 8, 2007, from http://www.npd.com/press/releases/press_070222.html

The Standish Group Inc. 2005. *The CHAOS report.* The Standish Group Inc.

TiVo Inc. 2005. *2005 annual report.* Retrieved April 26, 2007, from http://investor.tivo.com/

TiVo Inc. 2006. *10-Q report for the quarterly period ended July 31, 2006.* Retrieved April 26, 2007, from http://investor.tivo.com/

TomTom NV. 2005. *2005 annual report.* Retrieved April 26, 2007, from http://investors.tomtom.com/tomtom/reports/

TomTom NV. 2006. *2006 Q3 report.* Retrieved April 26, 2007, from http://investors.tomtom.com/tomtom/reports/rep_2006/2006-10-26/2006-10-26pr.pdf

TomTom NV. 2007. *2006 annual report.* Retrieved April 26, 2007, from http://investors.tomtom.com/tomtom/reports/

TomTom NV. 2007a. *Company profile, corporate history.* Retrieved April 26, 2007, from http://www.tomtom.com/about/about.php?ID=1&Language=1#history

Traudt, Erin and Amy Konary. 2005. *2005 software as a service taxonomy and research guide.* Framingham, MA: IDC.

Turner, M., D. Budgen, and P. Brereton. 2003. *Turning software into a service.* Computer, 36 (10), 38-44.

U.S. Census Bureau. 2007. *Quarterly E-commerce retail sales.* Retrieved April 8, 2007, from http://www.census.gov/mrts/www/data/html/06Q4.html

U.S. Department of Labor Bureau of Labor Statistics. 2007. *Employment, hours, and earnings from the current employment statistics survey (national).* Retrieved March 31, 2007, from http://data.bls.gov/PDQ/servlet/SurveyOutputServlet?&series_id=CEU5000000006&series_id=CEU0500000006

Urlocker, Zack. 2007. *Disruption in the software industry.* Retrieved March 30, 2007, from http://www.theopenforce.com/2007/01/disruption_in_t.html

Vasquez, Raphael. 2006. *Dataquest Insight: Separating notebook end-user wants and needs.* Stamford: Gartner, Inc..

Vincent, Philippe, Tim Jellison, Bart Hughes, and Philipp Stauffer. 2005. *The future of enterprise software: How software companies can achieve high performance in an era of disruptive change and uncertainty.* San Francisco: Accenture.

VMware Inc. 2007. *Company website - virtual appliance marketplace.* Retrieved May 5, 2007, from http://www.vmware.com/vmtn/appliances/index.html

Wikipedia. 2007. *Software appliance.* Retrieved March 30, 2007, from http://en.wikipedia.org/wiki/Software_appliance

Williams, Sampodyne. 2002. *Free as in freedom: Richard Stallman's crusade for free software.* Sebatopol, California: O'Reilly Media.

Wood, Andrea, and Belis Aksoy. 2007. *GPS: Focusing the core feature to drive sale*s. New York: JupiterResearch.

Young, Jeffrey S., and William L. Simon. 2005. *iCon: Steve jobs, the greatest second act in the history of business.* Hoboken, New Jersey: John Wiley & Sons Inc.

10 Appendix

10.1 Interview Guideline TomTom

Company: **TomTom NV**

Interviewee: **Peter-Frans Pauwels**

Title: **CTO, Founder**

Date: **January 5, 2007, 12.30 pm**

Place: **Amsterdam**

Questions

1. Introduction (ca. 5 minutes)
    a. Explanation of topic
    b. Timing
    c. Process (transcript approval etc.)
2. Business Idea (ca. 10-15 minutes)
    a. When you started the company, did you want to produce only software?
    b. How did the idea change over the years?
    c. When did you realize that you need to bundle your software with hardware?
3. Technical Realization (ca. 10 minutes)
    a. What did technical process of making a bundle look like?
    b. What were the technical challenges you faced (and continue to face)?
4. Company Competencies (ca. 30 minutes)
    a. What differentiates your company from a pure software company?
    b. What differentiates your company from other consumer electronics companies?
    c. How did you build the competencies to build an integrated product?
    d. Apart from growing the organization, what were the challenges you faced in the transition from a software company to manufacturer of an integrated product?
    e. How did you build a team that was able to handle hardware design plus software fitted for that particular hardware?
5. Assembly of Value Chain (ca. 15 minutes)
    a. How is your value chain organized? How hard was it to assemble?
    b. How do you manage your margins?
    c. What challenges do you face regarding distribution channels?
6. Future outlook (ca. 15 minutes)
    a. How do you see the future of the industry? – Will you ever go back to pure software?
    b. What is your advice for entrepreneurs that want to follow your footsteps?
    c. Is there anything you might want to add?

10.2 Interview TomTom, Peter-Frans Pauwels

Peter-Frans Pauwels is a co-founder and current CTO of TomTom NV based in the Netherlands. TomTom makes personal navigation devices. The interview was conducted on January 5, 2007 in Amsterdam.

Hein: Thank you for seeing me today. My first question is: when you started the company did you want to produce only software?

Pauwels: Yes. When we started back in 1991 the idea was to create a software company. The opportunity that we had was that Harold Goddijn, who is now CEO of TomTom, at the time was running a different company importing handheld computers into the Netherlands. When I was looking for job I ran into him. He said, "I'm importing these devices but they don't actually do anything. They need to be programmed, and they can be great for bar code scanning in supermarkets, logistics and other types of applications. Why don't you come and join my company and create software for that?"

Upon which I thought, "Well, if I do that and its becomes very successful how am I going reap the benefits from this particular decision?" So I went back and said, "I need a structure to get my share if this is a success." Whereupon he said, "Well let's start a joint venture." And that's when we started a small company aimed at creating software for handheld computers. That was one of the best decisions of my life.

So originally we started out as a company writing bespoke software for these handheld terminals. Basically we went out to customers with a warehouse that imported something and redistributed it and our handhelds scanned the goods at the gates.

We tried to make that into standard software but we failed miserably. During the first five years this was what we did. We were a six to ten person company. At the time I myself was actually in sales.

Hein: You were in sales?

Pauwels: Yes. If you're a small company you do what's needed. We had some very challenging and interesting projects. We worked on probably the first couple of real wireless data projects in Europe using a data technology called Mobitext. This was all prior to GSM. As we grew as a company doing these bespoke projects we found out that we were not really right guys to run projects within the concept of business-to-business. We were always ending up with deals where we had to commit to a price beforehand and then, of course, that we spent much more time than we hoped on it. That was hard and we weren't really growing.

Hein: So when did you enter into the consumer market?

Pauwels: What happened in the mid 1990s was that Psion (the handheld platform we were making the applications for) started enter into the consumer market with their organizers or PDAs.

Hein: Yes. I remember that.

Pauwels: They had the clamshell device with little keyboards. We realized that with our skills we could also make applications for those devices. So we originally developed a scientific calculator program and a full-scale double bookkeeping program for those PDAs.

But what we lacked, of course, was distribution, brand and all those kind of things. So we went to Psion based in London in the UK. We said, "We have some nice accessories for the Psion. Can you help us sell those?" And they said, "Ok, we'll put the application in a box for you. We'll put our name on it and we'll distribute it. And every there months we'll send you a royalty statement."

That's how we started. We realized very quickly was that every three months there would be money (i.e. the royalty statement) coming out of the fax machine. With only one product we could be very successful. We also realized that it was much more fun to be inventors and decide what you want to make. We were able to make our own decisions about what we thought the market would like. We were also not limited in our success by the number of people we employed. Shoot once. If it's successful we get zillions of people who buy it.

Hein: So you went from bespoke project-based software to the classic standard software business model.

Pauwels: Yes. So we gave up all the bespoke stuff. That was hard for a year because that was where the money was coming from. We built our software title portfolio. We added dictionaries together with Harper Collins. We added travel guides together with Michelin and Varta Führer. We added games together with Infogram.

We became more and more successful. We came to a point where we said, "Well we're now only earning a small royalty. What if we set up distribution ourselves? What if we create our own brand? What if we really produce the memory cards or floppy disks ourselves? That would change the business model quite a bit. It would give us more control of our destiny. Give us more margin etc., etc."

So this was in small way already adding value to what we were producing as a company.

Hein: Were you scared of building your own brand?

Pauwels: Yes, we were. It wasn't an overnight thing. But we were lucky because Harold who was still importing the handhelds from the UK into the Netherlands fell in love with the export manager in London who was selling them to him. And she fell in love with him. She resigned, came to live in Holland and worked with a cheese factory for a while. She's French. Selling Dutch cheese into France. And that was extremely boring for her. So I said, "Corinne, can't you help us a bit? Because you were export manager for Psion you know who distributes the handhelds throughout Europe and they know you." Getting the distribution was key. Her credibility and long-standing relationships around Europe gave us leverage.

That's how we managed to start building a brand. Which at the time was called PalmTop.

And fairly quickly after that we started to earn more money. We could bring in some more people. We grew the company to 20 people. We had to have a help desk, customer support, those kind of things. We became Europe's largest software publisher for PDAs. Of course, the whole market was very small. There were something like five companies doing the same thing and we were by far the largest.

Hein: What were your revenues at that point?

Pauwels: They were somewhere between EUR 1 and 2 million. We were minute. But as a software company we were already keeping a bit of stock, i.e. floppies and manuals.

Hein: When did you start with navigation software?

Pauwels: Our portfolio included from 1997 onward also map applications. All you could do was look on the screen, see a map. You could zoom in and out and pan. And calculate routes. And it would give you a description of the route. But this wasn't navigation. This was really an electronic map in your pocket.

We had two types of products. One was a really good map of the whole of Europe but only with the major roads in there. It allowed you to plan a route from Madrid to Moscow. And the other one was a product whereby you would get a – well you would get on a CD all the major cities of Europe in detail. They were disconnected. So you could download onto your PDA let's say the map of Amsterdam or the map of London. You get only the detail but at the time because memory requirements were enormous and capacity was limited. So we actually couldn't fit a whole country onto a handheld.

Those two products were suddenly 60-70 percent of our revenues. So we knew what people wanted. Already back in 1997 we saw that there was something wrong in the automotive industry. Why do navigation systems cost 5,000 to 7,000 Euros? Why is it that expensive? We did quite some research and had a consultant look into in Japanese situation for us.

Pauwels: We asked ourselves, what's wrong there? We know how to build a device like that. What we didn't understand was why it would end up with that price. At the time we were still scared away because of technical complexity and a CD ROM technology at the time was still was flaky and expensive. So we shied away from that. But we started to get a bit of a feeling for the market and also gained the understanding that within the automotive market there were enormous margins to finance expensive distribution etc..

Hein: How did you continue to develop the company?

Pauwels: We concentrated on doing our stuff. We also did some LBS products with Ericsson that failed miserably. That was in March 2001. We were on the verge of going bankrupt. We decided to pull out of that whole thing with Ericsson and LBS. Which obviously never happened. We reacted very, very quickly. We fired a third of our workforce. At the time we were at 35. So we ended up with 20. That was definitely the most horrible day in my working life.

We then concentrated again on what we knew we were good at: doing those accessory applications for PDAs. And that worked fine. But then something happened where I realized, "Hey, this is really technology push. President Clinton on the 6th of May 2000 had opened up GPS for commercial use.

A GPS before that time was still very limited it would only get a position with an accuracy of 30 meters. Which was insufficient to pinpoint you as a GPS user onto a road. Suddenly this made it possible to pinpoint you exactly onto the road.

Pauwels: Another thing that happened was that the cost of memory cards came come down and the capacity went up in orders of magnitude. Of course, this was all driven by digital photography.

Pauwels: The third thing that happened was that the performance with of PDAs had gone up tremendously and at the time another product at the time was the first Compaq iPAQ. They had colors and multimedia and lots of memory. At that point we realized if we take our city mapping product stick it on there and we take the GPS we can actually put the whole country on one card. We could then add turn-by-turn navigation. We actually could make an accessory that turns your iPAQ into a navigation system. And that's exactly what we did.

Hein: And no one else was doing that?

Pauwels: We were not the first ones to do it. There was an American company already doing this. I believe Destinator already also sold navigation software for devices. But only in America and they weren't successful at all. Their background had also not been in selling stuff to consumers but rather B-to-B stuff. Tracking, tracing etc.. So most packages were probably not built as a consumer product. Etcetera. So we were the first ones to roll it out as a real consumer product.

Microsoft was very excited about it. And that helped open up some doors. At the time PDAs were actually becoming less and less popular. This was really when smart phones started to come into the market and PDAs were in decline. And suddenly here was something that could extend the lifetime of the whole product concept. So the PDA manufacturers and Microsoft were all very enthusiastic and got onto that bandwagon.

End of 2001 we had approximately 2 million Euros in revenue. In mid 2002, we introduced a PDA accessory. A box with a GPS receiver, a windshield holder, a cigarette lighter adapter and a CD ROM with the application and maps. You had to buy your own memory card. We introduced that into the Netherlands, Germany, UK and France. In 2002, our turnover more than quadrupled to 8.8 million Euros.

Pauwels: And at the end of the year more than 95 percent of our revenue was from the navigation accessory. The cooking books, the dictionaries etc. was all small, small fry. So we abandoned all of those products. We said, "Now we are navigation company. Let's focus strictly on this. "

Pauwels: In 2003 two important things happened. First of all, we realized very quickly that we were starting to sell to people who didn't own a PDA yet. When we introduced our product we thought, "Our customers have a PDA and they buy the accessory and turn it into navigation system." But through feedback from our distributors, shops and our own help desk we realized that people buying it were now going into the shop to buy a PDA and the box in one go because they didn't want to have PDA. They wanted to have a cost effective navigation system. So for 700 to 800 Euros you could have a fully functional navigation system compared to aftermarket navigation systems at 2,500 Euros and the in-dash navigation system at 3,500 Euro. So that was a bang and sales went up.

The other thing that we realized as well was that the whole user experience of the box plus the PDA was horrible especially because we were starting to sell to non-technical people. These people were having lots of problems. The whole installation procedure and getting a PDA of brand X, Y, or Z and then setting it up with a cradle and a PC and loading from CD ROMs etc.. So we said, we really need to bundle it with a particular PDA. And make sure that everything is optimized in such a way that when you open the box with the least amount of work, or steps to go through you can actually stick that into your car and navigate.

But we didn't want to buy a lot of PDAs. And we also recognized that the PDA manufacturers didn't want to buy a lot of our accessory. At the time, you know we were successful. It was growing. But it wasn't proven in any way. We couldn't ask HP or Palm to buy a zillion of our accessory boxes and then stick it into their boxes. What we came up with then is absolutely an innovation in consumer electronics distribution. Is that we said, "Let's go to a broadline distributor like Ingram Micro or Computer 2000. We'll set up a three party deal. We sell our accessory box to the broadliner. The PDA manufacturer, in this case HP, sells the PDA. On top of that we will deliver a larger box into which both of those products will fit and which has its own article number or SKU. Article numbers or SKUs are holy in distribution. The broadliner was willing to buy those two separate products, put them into that over box and then put it in their catalog as one single product for sale to the chains, to the stores. So what we did here was in a way an innovation. No one had to pay each other or buy extra components or product. And we still made it possible for the shops out to buy one single solution. That worked extremely well.

Hein: But the software was not preinstalled?

Pauwels: It wasn't preinstalled. But we made sure that there was a memory card in the box. The only thing you needed to do was to stick it into the PDA and it would work. And that worked amazingly well. In 2003 our revenues went to nearly 40 million Euros. At the same time we realized, ok, this is still bit of a messy solution. Because you end up with a lot of cables, a multiple cigarette lighter adapter, chargers etc in your car. We thought we knew what people really wanted. They really wanted to have the PDA do just one thing, navigation. Forget about the rest. That's where we came to the conclusion that we had to turn ourselves into a real consumer electronics company. Just do the whole thing. But this was a big bet.

We thought we needed to create a device that go can take straight out of the box at the store and it navigates you home. Up to that moment this wasn't possible. You always had to go home and then do the installation and only then you could navigate.

Hein: So you had to design a hardware product?

Pauwels: Yes. In order to accomplish that we had to do a couple of things. We needed to find someone who understood how to make a consumer electronics device. Because Harold used to work with Psion he knew people in London who knew how to make devices. So we got someone on board who had been making organizers and PDAs for 20 years.

We then needed a manufacturer. Again using contacts from Psion we also found a contract manufacturer in the Far East. We went to IAC, a large Taiwanese company. Luckily, on the basis of earlier relationships they agreed despite the fact that it was very risky for them. We had no track record in this whole area.

Pauwels: Finally we needed financing because we were going to build up inventory of real PDAs that cost hundreds of Euros. Not just these little accessories. The bill of materials cost suddenly went up by a factor of ten. So we went to the bank to get a credit line. But of course that being a Dutch bank they needed some securities. They agreed to extend us a 10 million Euro credit line but wanted the shareholders of the company to be personally liable. Not for the full 10 million but for 1 million. So that was one of those horrible days in my life. I first said, "I hate banks. " But then I did it. We each had to sign up for 250,000 Euros liability. Which at the time, had everything gone wrong, would have put me in great trouble.

Pauwels: The funny thing is that the company was so successful; we generated so much cash that we never needed the credit line.

Hein: And you never got any external investors in the company?

Pauwels: No. Not until the IPO. We invested 18,000 Euros when we founded the company cause that was what the law required. It's always been ours until the IPO.

Hein: That's very impressive. How did you go about launching the first integrated device?

Pauwels: In May 2004 we launched the original TomTom GO. I remember at CeBIT all our competitors were telling us, "You're crazy. Customers want to have an agenda and an address book in there. They want all-in-one functionality." That really scared us. Probably our competitors were secretly happy that we'd gone crazy. But we turned out to be right. Our revenues went from 40 million to 192 million in that year. Now obviously we were selling more expensive products but also the number of units started to grow enormously. In the first six months after introduction of the GO we sold a quarter of a million units. That was stunning. Our contract manufacturer even ran into capacity problems.

The rest is kind of history. The growth went on and exploded. I would say that still we see ourselves as a software company. That is where ultimately the big value is despite that we've become a consumer electronics company. But if you look at where the engineering efforts are going into, definitely 90 percent are going into software.

Hein: So you've really taken a step-by-step approach to becoming a consumer electronics company?

Pauwels: Yes. I see the steps as follows: first we did bespoke software. Then we went into standard software i.e. consumer software on a royalty basis. We only delivered a master copy. That is as pure as you can get in terms of being a software company. Phase three involved doing our own distribution and creating our own brand. In that phase we started putting our software in boxes. That still doesn't involve a lot of money but you need to get distribution right. It's the first step where you really say, "Ok, I start adding value by putting it on something physical." We actually put it on a memory card. Like a Nintendo game pack in a way. That by itself would already add value. It allows you to charge more money than for just a downloadable license.

Pauwels: The fourth step came in 2002 when we wanted to create a full-blown PDA navigation accessory. Not just the memory card but also a GPS receiver, a windshield holder and a cigarette lighter adapter. By putting all that into box again we could say, "Ok, this is worth much more than just software itself." Don't forget maps are also included in bundle. But they're less tangible and people tend to think that we do the maps – so they don't add as much value [even though their license costs are high].

Pauwels: The fifth and last stage I think is the most important stage and the biggest step. This was the step of designing our own consumer electronics device. This means designing everything down to the level of the printed circuit board. Of course not the CPU but we design everything else such as electronics, mechanics and industrial design. Obviously we used an agency for the industrial design but it was all under our control.

Hein: What were the technical challenges you faced during the fourth and fifth phases?

Pauwels: The challenges are quite different between the fourth and the fifth stages. During the fourth step we were selling the navigation accessory. We wanted it to be compatible with as many different brands of PDAs as possible. We supported and still support more than 50 different PDAs. Even though they are on a platform like Palm, Windows CE or Symbian OS, they are never really standard. Also, when you get into real time operations you really want to deliver a good consumer experience. So a lot of time was wasted on making sure that our navigation worked on each of those devices. Not a very interesting challenge, just an expensive technical challenge.

In our move towards consumer electronics we had to learn some new things that we hadn't done before. You're suddenly creating a closed box where everything in there is yours. You need to control it yourself. Until then we had done application software running on someone else's platform, e.g. on Microsoft Windows CE or Palm OS. For our own device we decided to use Linux as an operating system because of the cost. We also didn't want to tell Bill Gates how many units we were selling. Another reason was the great availability of tested open source software from Linux.

Once you start doing your own device you also need to integrate with the hardware. We had limited understanding of hardware. Not more than your typical software engineer will understand. Suddenly we needed to create boot loaders, device drivers etc.. Certainly when we created the first device, the original TomTom GO, we had huge challenges. It's not like there's no one in the world who has done that before but as an organization you don't know exactly how to do it.

For example, we made a big mistake by outsourcing the integration of the operating system with Linux onto the hardware to a company in India. We had been pointed in that direction by Samsung, the supplier of the core CPU. That turned out to be an extremely costly mistake. These guys were very good at producing all kinds of CMM level three ISO documentation. But the code was a different issue. When they started to overrun the schedule we got really concerned and started to look into the code. We found out that they had put only one fairly junior programmer on the project. So we decided to throw away everything they had done and do it ourselves. This was six weeks before we were supposed to deliver the product. It was tough but we did it. By doing it ourselves we also learned the skills and therefore then had the knowledge in-house.

Another big challenge that we had was that originally our software was written for Windows CE. Now we had to make that run on Linux and at the same time also on Palm OS and Symbian OS. So we decided to throw away everything we had and rebuild the whole navigation system from the ground up. We reused the core navigation algorithms and the core compression algorithms for maps. But we designed from the ground up a software framework that could easily target the different operating systems and different types of devices, for example, those with a touch sensitive screen or with a mobile phone screen interface.

This company has been around now for 15 years. For ten years we've been doing mapping and route planning. By the time we started doing consumer electronics we were ready. We were standing on the shoulders of giants – with us being those giants. There had been so much groundwork done that by the time we went into consumer electronics the challenges were really about doing the consumer electronics.

Hein: How do you differentiate yourself from pure software companies and from regular consumer electronics companies?

Pauwels: Ok. Those are really two questions. How do we differentiate ourselves from a pure software company? We really we create products for consumers that are total products. It's like an iPod. It's a complete experience. What you see is what you get. It's end to end. With pure software products that's never the case. It's always runs on something else. It integrates with something else. It does part of the job. It doesn't do the whole thing.

Once you become a consumer electronics company you really have to think from the consumer's viewpoint. You create software not from a point of view what it can do but you have to limit yourself to what a target customer will be able to understand in a natural way. You have to think coffee machines in terms of user interface.

Now to the second question: What differentiates us from a consumer electronics company like Sony or Phillips is, one, our focus on getting you from A to B. Two, we have a lot of software understanding, certainly in the area of a networked devices, the Internet, GSM, wireless data etc.. We are delivering solutions for consumers based on Internet technology directly relating to the navigation experience. For example, rich traffic information over wireless data networks. We've recently introduced a PC-based application that sits between our rich services and the navigation software. It's a type of conduit. We call it TomTom Home but you can see it's iTunes for navigation. That is not something that a typical consumer electronics company would necessarily think of or have the skills and the mental capabilities to do.

Hein: Is that because they see themselves as the whole and you as a software company are much more used to seeing yourself as a component of some larger system?

Pauwels: Correct. We also have a software technological understanding of what the Web can do for you. Look at Apple, they're very much a software company, too.

Hein: I'd like to hear about the team. Did you have any mental switch problems in your team or how did you build a team that could do this new thing?

Pauwels: We first built a team with hardware engineering capabilities in London. We basically inherited the Psion group. So they were a fully functioning team. Since they were in London and software people were in Amsterdam there was a very logical division of work. Not that much mentality change was needed. Now on the other hand, once you start doing device drivers you need to get people with those skills. I would guess all the other software engineers were really excited about doing something for our own device.

Now what really changed was the need to have a proper product management function within the company. Until then we were a bunch of software guys pushing products out. Suddenly because you're doing consumer electronics and your brand is growing and you need to bring it all together you need to take a bit longer to review. As you're growing you need to have a better understanding of local markets and competition. You therefore need people in the middle who aren't product developers who are there for the whole product creation process until the end of the product life. That was really something different. You need someone who has an understanding of the technology, of the marketing and of the logistics and brings it together. When you are a classical, typical software company a lot of what drives the development are requirements from customers. But there is a lot of room for software engineers to do a lot of stuff. Once you get into the consumer electronics mode and you have product manager, the product manager will take a lot of the nice intellectual or creative decisions him- or herself. That's a mentality change that has taken us a long time and we're still not quite there.

On the sales side in terms of mentality there was no change. The marketing side was a mentality change. Suddenly you're real. People see you as real because you have a physical product.

Suddenly you need a whole logistics department. We're very happy that we have some very good people there. Then at the same time we outsource a lot of it. So we really do the management of logistics. The real stuff is done by another company. But we have to tell them very, very clearly what to do.

Hein: How is your value chain is organized? How do you organize manufacturing and physical distribution?

Pauwels: The world has changed enormously in the last 15 years. It's become relatively simple to do that. You have all these ODMs, companies like Flextronics, IAC and Quanta. You can go to them with a specification, pay some money and they'll manufacturer anything for you and they'll ship it anywhere. We work with IEC and Quanta.

I think the biggest challenge for a software company that was selling via the web is understanding physical distribution. Physical distribution is a people business. Especially in the US because there are not that many companies and chain stores. It's basically CompUSA, Best Buy and a few others. There aren't a lot of them to talk to unless you end up with very small mom and pop stores and that's where you don't want to be.

Understanding the game of selling to a buyer who is responsible for a certain product category for let's say CompUSA is an extreme challenge if you don't know retail electronics. People will laugh at you if you don't understand their game. Luckily we had Corinne.

Also, once you go into retail distribution you cannot permit yourself to make any errors. A product recall is obviously horrible. But also absolutely horrible is delivering thousands of products to Best Buy, then discovering that there is product fault and needing to recall them from distribution. From a brand point of view that's the nicest scenario because you don't need to tell the world that there was a problem and you recalled products. But if a recall happens with one of the big distributors you really have a reputation problem.

Software is built so that it can be easily upgraded. If you have a problem you just upgrade it over the web. But here you can't. People see hardware and software together as the complete product. That is a mentality that a typical software company will need to understand and adapt to. For us it meant investing heavily in a very large testing and QA group. Those were disciplines that we didn't need before.

In this business you also have to show that you're good for what you promise. That means if you promise a product for a certain date you will have to deliver it on that date. If you do all of that right you start having a working relationship. You understand how to assist them with the shelf work and that they want accessories with high margins. If then somewhere in the future something goes wrong once they will help. But getting there is extremely difficult.

In a later phase the logistics of it all gets extremely complicated. You have an enormous number of products. You need to start understanding what inventory is really in stores. What's the sell-through time? What do I have on ships between Asia and America and Asia and Europe?

Hein: Do you manage that yourself or do you outsource?

Pauwels: We do the management of that ourselves. It's a completely a novel discipline to a classical software company. Additionally, the money involved is humongous. For example if you have product in the stores and you introduce the next generation product you need to manage that properly. We didn't know how to manage our new product releases as they related to retailer inventory. When we first started shipping our own products, one quarter we announced a new product prematurely and the retailers still had thousands of units of the old model. They returned the old model and only wanted to sell the new one. Dixon's said to us, "Great, you have a new product. By the way, I still have 10,000 of the old ones. I'm putting them in a container back to you." There's no way that you cannot take those back. You need to credit them for that. We had to write off all of those units. So this is the difficult game of managing an end of life properly. You need to make sure that the stock in the stores goes down while still maintaining sales-through, of course. At a certain point you bring in the new inventory. There are many ways of doing that but it's completely alien to a classical software business.

Hein: How do you see the future of the industry and especially of TomTom? Do you think you'll ever go back to pure software? Is there a cycle for you?

Pauwels: You know, I don't know and I don't care. We want to sell something that people buy. And that's getting you from A to B. Getting you from A to B can be done on the basis of a consumer electronics device. It could also be a wireless service. It could be very well be that one day every car is outfitted with a TomTom software component. We will continue to invest in adding longevity to the brand and the product and the consumer experience. Maybe we will do more and more online services. Not as a replacement but adding value by making it very easy for you to download additional content onto your device or share experiences with other drivers. It's mostly about creating a long term relationship with a consumer. But overall we'll probably be doing consumer electronics for a long time.

Hein: I read that majority of your customers are between 40 and 60 years old. Do you target these groups of customers with specific things? Do you look at technology adoption cycles?

Pauwels: I think we are somewhere in the early, early majority now. I thought our main customers were aged 35 to 55, but I'll have to check with marketing. My silly personal explanation for that would be that our products cost real money so they need to have a proper income. Navigation for many people is not as much getting you from A to B as giving you a sense of security. Once you get older and you have kids that kind of stuff that starts to matter.

Hein: Do you have any advice for entrepreneurs that want to follow your footsteps?

Pauwels: Focus, focus, focus, focus. Certainly in the early stage of company it's so hard to keep focus if you are hungry for money. You see all kinds of different opportunities and before you know it you are working on ten plans at the same time. So choose one. Believe in it and go for it. If it fails, fine. Then you do another one but keep focus.

Hein: Thank you very much for your time, Peter-Frans.

10.3 Interview Guideline Sensory, Todd Mozer

Company: **Sensory Inc.**

Interviewee: **Todd Mozer**

Title: **CEO, Founder**

Date: **March 23, 2007, 3.00 pm**

Place: **Sunnyvale, CA**

Questions

1. Introduction (ca. 5 minutes)
    a. Explanation of topic
    b. Timing
    c. Process (transcript approval etc.)
2. Business Idea (ca. 10 minutes)
    a. When you started the company, did you want to produce only software?
    b. How did the idea change over the years?
    c. When did you realize that you need to bundle your software with hardware?
3. Technical Realization (ca. 10 minutes)
    a. What did technical process of making a bundle look like?
    b. What were the technical challenges you faced (and continue to face)?
4. Company Competencies (ca. 10 minutes)
    a. What differentiates your company from a pure software company?
    b. How did you build the competencies to build an integrated product?
    c. What were the challenges you faced in the transition from a software company to manufacturer of an integrated product?
    d. As a small company how did you build a team that was able to handle hardware design plus software?
5. Assembly of Value Chain (ca. 10 minutes)
    a. How is your value chain organized? How hard was it to assemble?
    b. How do you manage your margins?
    c. What challenges do you face regarding distribution channels?
6. Future outlook (ca. 5 minutes)
    a. How do you see the future of the industry? – Will you ever go back to pure software?
    b. What is your advice for entrepreneurs that want to follow your footsteps?
    c. Is there anything you might want to add?

[Note: Not all of the questions in this guideline were used in the actual interview.]

10.4 Interview Sensory, Todd Mozer

Todd Mozer is the founder and CEO of Sensory, Inc based in Sunnyvale, California. Sensory makes speech recognition chips for consumer electronics and toys. The interview was conducted in Sunnyvale on March 23, 2007.

Hein: Thank you for seeing me today, Todd. My first question is: when you started Sensory 12 years ago, what was your original idea for the company?

Mozer: Our business model really hasn't changed a whole lot of what we're trying to do. The original thinking was that we would do a company that could interact with products in the same way that people interact with each other. We wanted to enable people to interact with products, in that same fashion. And that's where we kind of got the name "Sensory." And originally, I guess I was thinking that I wanted a name that we could grow into over time. We have a lot of different senses and we deploy these different senses in ways of understanding ambiguous information and deciphering it. That's how we communicate with each other.

I had the feeling that the user interfaces of consumer electronics hadn't evolved at the same rate as the other capabilities of the consumer electronics had. Consumer electronics have gone from analog to digital over the last 20 years. But they have so many new features that people can't access. For example 50% of VCRs that are "flashing 12" because nobody knows how to set the time. We've got Moore's Law making more and more powerful micro-controllers and putting more MIPS and more memory, but the user interface hasn't kept up with that to be able to take advantage of it.

So, I wanted to start a company that could improve the user interface by allowing people to interact with products in the same way that we interact with each other, with gestures and with facial expressions and with speech. The first area that we wanted to tackle was speech and we thought the best way to do it was with chips. We had the goal of advancing beyond that once we mastered it. But we haven't quite mastered it yet. So we're still doing speech recognition chips.

Hein: Did you ever think of doing that as software only?

Mozer: No, originally we wanted to package a product in hardware. I've founded and worked in businesses before that did chips, so I have a chip background myself. I've also had embedded software businesses, and it's a lot easier to sell hardware than sell software. From the beginning, I knew that Sensory was going to be a technology company and that a lot of our strength was going to be in software IP. But I always wanted to package it in a chip. We started the company in '94. We had a chip spec in '94. And we had chips to sell customers in '95.

Hein: You always were looking at the proposition of packaging your software with hardware?

Mozer: Yes.

Hein: So how has your idea evolved?

Mozer: 12 years ago when we started I had the vision of adding a lot of other sensory functions to our products. We were also going to do image recognition and other sensory functions, what we see, we smell, we taste, we hear. We got stuck with the speech side of things. Rather than moving into image recognition and some of the other sensory technologies, we've done more enhancement technologies for speech recognition. We've added speech synthesis. We've added music synthesis, and things that our customers have asked for so that we could add more value to our customer base.

Hein: When you set up the company knowing that you would do software and hardware, what were the things that you had to think about in regards to capabilities?

Mozer: The semiconductor industry has evolved over the last 20 years from a place where a chip company used to have its own design, its own fabrication, its own testing, and it's own packaging even. Today it has become a more splintered and specialized function. Part of that is just because of the evolving technology of fabrication. It's gotten extremely expensive to run a fab [chip fabrication manufacturing facility]. So independent fabs got set up in the '80s that spurred the whole fab-less semiconductor business model. So, it's gotten to the point where it's relatively easy to be an IC [integrated circuits] design company without having to have all the back end kind of functions.

If someone in embedded software wanted to take on a chip model like we did, they'd need to hire somebody with some chip design expertise on the engineering side and somebody with chip knowledge on the operations side, but it's pretty easy to subcontract a lot of it out. If you look at the personnel of our company we're more similar to an IP house or an embedded software house than anything else. We've got just a handful of IC focused people. We contract out most of the IC work. We'll architect our chips in-house. We've got two chip designers and a product engineer in-house. But a lot of it we contract out to external consultants - on the design side as well as on the fab side and on the test side. We use an external fab house and an external test house.

Hein: How do you manage the interface between these people e.g. your two chip designers and the software engineers that design your software. How did you do that and has that evolved?

Mozer: Well, we started off with no in-house IC folks, as people always start off with technologists in-house, and we literally contracted everything else out, except the specification and architecture for the chips. So, from the very early days we architected and inspected our chips and then just had an external firm do the development. The tough thing about being a chip company is that in other functions, technology development, you can have your guys focus on technology year around. In a chip company, if you have a large team of chip designers, what happens once the chip goes to market? Well, you're forced to immediately start on a new chip or else you hire a bunch of people and then fire them once the chip comes out. I didn't want to be in that mode. I only wanted to hire enough people that I could really keep onboard on an ongoing basis.

The size of our market doesn't justify doing a whole lot of chips. In our history we've really only done a couple of different chip designs. Regarding the basic architecture of the chip itself, what we're just starting our second generation. Therefore we can't justify the cost of doing chip design all the time and therefore we can't justify the staff to do it. But as we got bigger and we had more resources, we decided that it made sense to hire a couple of people in-house to work on it full-time.

Hein: How many people do you have here right now?

Mozer: We're 35 total including the Portland office. We acquired a company up in Portland back around 2000. And we've got a small office in Vienna and an office in Tokyo and an office in Hong Kong.

Hein: What percentage of your time goes into development on the software side, and what percentage into the chip side?

Mozer: I would say that of our salaries, probably 80% is for technology and software people, and maybe 10% or 5% is for chip developers. Then there's management and marketing and sales. So we're very heavily software technology-focused in terms of personnel. But if you look at our total budget, we're spending close to $2 million on the next chip design and most of that's just being contracted out. That would make it a number that was a lot more balanced in terms of total spending.

Hein: Where do you create most value?

Mozer: Value is created in the technology, really. We don't really need to hire a chip designer at Silicon Valley salaries if you can contract it out to India, China or Portugal. We don't want to spec the chip here. We want to architect the chip here. So the seasonality and cyclicality of needing IC designers that makes me not want to bring it in-house and the other is just the cost of doing here. Silicon Valley is just an expensive place to implement things.

Hein: What functions do you keep in-house?

Mozer: The idea is that you can have, you can hire a company that will do the design for you. They'll do the test for you. They'll do the fab relationships for you. They'll do everything from, you know you hand them a spec or you give them code to design the spec and they'll take it from there and hand you completed chips in whatever package you want. And so they'll take it from almost at the start all the way to the finish for you, and that's a well-established model, too. We don't do that. We just ask a separate design company to do the design for us, and then we handle the fab relationships. We handle the test. We handle the packaging relationships and that sort of thing is in-house. The reason we keep things in-house is cost. It's all driven by cost. We looked at buying ASICS this time around for this new chip. You end up paying a royalty on the silicon in return for having them [the outsourcers] take care of everything. You pay less up front, and you take less risk up front, but you're paying more for the silicon. We decided we needed the lowest cost silicon we could have.

Hein: This is mandated by the business that you operate? In the toy industry do you have to tightly manage the costs of your product to maintain your margins?

Mozer: Yes. I mean. Management of outsourcing decisions depends on what you're selling and who you're selling to and the margins you can get. If we were a board level company and we were selling boards, then the cost of the specific chip, you know if it goes up by 20%, it might not matter because in the grand scheme of the board it's fairly minor. But we're not in that business. We're up against Taiwanese suppliers that are getting low margins and we need reasonable margins because we invest so much in the technology. So we've got to keep our chip costs low so that we can break even and still supply our customers with excellent products at an excellent price.

Hein: How does your business model work?

Mozer: By selling our software with the chip around it, we have a very strong competitive advantage. We set up a fairly high barrier to entry. If we were just selling technology those barriers would be much lower. That's the simple answer.

Hein: Do you compete against people that do only software?

Mozer: Indirectly. We're the only company that designs the speech recognition chip and designs the algorithms that go into it. That gives us a real nice advantage because we understand what drives costs of chips, and so we can architect our technology so that it really will fit into the lowest cost chip. What happens to our competitors is you get these semiconductor guys, speech recognition guys and one approaches the other and says "Hey, why don't you run this on your chips?" So they get a big chip and they get their technology running on it. As a result the technology that's running on it is inferior to what we can offer because they either require more MIPS and more memory to do what we're doing or else they have an inferior kind of performance if they get a comparable sort of cost point. For example, there have been companies like Oki Semiconductor that came out with a speech recognition chip where they licensed the technology for voice-control systems, and GemPlus has a speech recognition chip that they've licensed from a Taiwanese company over in Taiwan. There are usually two separate entities that do it as a joint venture, and it ends up adding a lot of costs. Even if they were smart enough to kind of do the things that we were doing, they'd still have two mouths to feed, so the technology supplier would want a royalty, and the chip company wants to make its margins.

Hein: So by vertically integrating this you create the margins for yourself? Does it help to eliminate the interfaces?

Mozer: It's not just interfacing. It's really architecting our technology designs to work well with a low cost chip, and vice versa. You know, it's not really a one way process where the technology can dictate what the chip should be or the chip should dictate what the technology should be. It's very interactive. So, we understand, you know how to do analyses. We get a quote for a new fab and we can figure out what's the size of our ROM and RAM. Then we figure out based on those different sizes what the cost is if we implement our technology in different ways.

Hein: And software people usually have a totally different mindet?

Mozer: Yes. For them it's all just memory, they need a big hunk of fast memory and there's going to be a system design where they can get it.

Hein: So, this optimization only really works if you build all competencies in your company? You as a founder embody several of those competencies?

Mozer: Yes. I have a background in chips, speech, music and business. I'm a business guy. [Mozer has an MBA from Stanford.]

Hein: How do you deal with the squeeze from, lets say, Taiwanese companies that need less margins?

Mozer: Our strategy is not to compete on price. We won't try to undercut them or even match their cost. What we try to do is offer a better quality product and better tools around it. We have a whole suite of tools so that people can program their chip, C-compilers, assemblers and debuggers. We have a whole integrated development environment for programming your chip. But when you're in the speech business, you need more than that. We developed tools that make it very easy to create a voice recognition set. So we have PC-based tools where you type in words and then you hit a button and it downloads it onto a demo board. Nobody else in the industry can do that. Our competitors tell their customers when they want a vocabulary it's going to take a couple of months. After that they'll come back to you and you can try it out. Then if you don't like it, they'll go away for another couple of months, and then we'll, you know they have to go out and record people saying these things. We've developed a system that can do that on the fly.

Hein: So technology is your big differentiator?

Mozer: Absolutely. Ease of development, accuracy, quality in various forms.

Hein: How big is the competitive pressure in your market?

Mozer: We've been lucky. We've been in our own little niche where competition has not been our biggest issue really. The majority of companies that come to us have decided either they're going to use Sensory or they're not going to use speech recognition at all. So our challenge has been much more along the lines of can we get it cheap enough, can we make it good enough so that people will actually use it, not are we going to lose it to a competitor. We have a market size battle, not market share. We own our market.

Hein: What do customers pay for one of your chips?

Mozer: About $1.50. It comes with the 8-byte microcontroller, it can do speech recognition, it does speech synthesis, compressed speech playback. We can do music. We can do beat detection, all sorts of different technologies and from a single chip. We've got all, the RAM, all the ROM, all the game control and pre-amplification of it and built onto our chip. So, it's very cost effective. Our customers can throw out the chip that's in their system already, so it's not even an incremental $1.50. It is $1.50 minus the cost of their existing microcontroller.

Hein: What kind of gross margins do you have?

Mozer: It depends on the volumes. For low volumes we get above 60%. If it's high volume, we like to get above 40% gross margins. We need to get a little bit higher margins than companies that are doing chips without a lot of intellectual property in it, just because we do invest so heavily in the technology as well as the chip design.

Hein: Will you ever go back to doing pure software?

Mozer: We offer that today. I just don't have any sales people that focus on it. We occasionally get people that just want our software and we have some SDKs available. So we do license software but it's not a big part of our business. Software is very competitive. We like our niche that is less competitive.

Hein: Thank you very much for your time, Todd.