A New Architecture for Corporate Desktop Computing

By
Mark E. Cummins

B.S. Electrical Engineering 1982, Lawrence Technological University
B.S. Mathematics and Computer Science 1990, Lawrence Technological University

Submitted to the System Design and Management Program in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Engineering and Management.

At the
Massachusetts Institute of Technology
January, 2000

Signature of the Author ....
Mark E. Cummins
System Design and Management Program
January, 2000

Certified by ....
Charles W. Boppe
Aeronautics and Astronautics Department
Senior Lecturer

Accepted by ....
Thomas A. Kochan
LFM/SDM Co-Director
George M. Bunker Professor of Management

Accepted by ....
Paul A. Lagace
LFM/SDM Co-Director
Professor of Aeronautics & Astronautics and Engineering Systems

A New Architecture for Corporate Desktop Computing

By

Mark E. Cummins

Submitted to the System Design and Management Program in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Engineering and Management.

Abstract

This paper investigates the corporate computing problem being caused by the drastic relative increase in the service costs of desktop computers as compared to the rapidly decreasing acquisition costs of these computers and proposes a new architecture to address this problem. First, the problem is discussed in substantive detail. Then, the prior art and existing proposed solutions are described to outline the current solution space. Next, the needs of the major stakeholders are identified and analyzed. The needs are then used to develop several alternative architectural concepts that are possible solutions to the corporate computing problem. These concepts are then evaluated by several means including Trade Studies. Next, a preferred architecture is selected and discussed. Finally, recommendations and opportunities for future investigations are provided.

Author:	Mark E. Cummins
Advisor:	Charles W. Boppe, Aeronautics and Astronautics Dept., Senior Lecturer
Title:	A New Architecture for Corporate Desktop Computing
Date:	January 2000

Acknowledgements

In grateful acknowledgement of the their support and assistance throughout the last two years, the author recognizes the following people:

My advisor, Mr. Charles Boppe, whose direction, wisdom, and concern proved critical in allowing me to complete this thesis in a timely and effective manner;

My SDM colleagues Michael Shashlo, James Goran, Joseph Wickenheiser, Michael Pepin, and Ed Esker for logistical assistance and encouragement;

My managers at Ford Motor Company, Ms. Sue Takai for nominating me for the SDM program, Mr. Robert Treharne for critical support during the final stages of the program, and Mr. Robert Dickson for steadfast support during the entire two year program;

My colleagues at Ford, Mr. Richard Minnick II for his wisdom and experience with the MIT processes, Mr. John VanStipdonk, Mr. Larry Foltz, and Mr. Alan Ratzow for carrying more than their share of workload for the last two years so that I could manage the SDM workload, and Mr. Steven Kardel, whose immeasurable logistical assistance, moral support, and occasional proofreading proved, as always, to be invaluable;

My Mother and Father for instilling in me the belief that working hard and being kind will bring rewards beyond your wildest imagination;

Finally, my most heartfelt thanks goes to my wife, Katherine, for her patience throughout the simultaneously overwhelming responsibilities of my SDM participation, my position at Ford, her position as an Elementary School Principal of two charter schools, and the joyful birth of our twins, Connor and Victoria, right in the middle of the SDM program! It was an amazingly complex undertaking and we are better for the effort. Everyone should be so blessed.

Thank you all so very, very much

Table of Contents

List of Figures

Chapter 1: Introduction

General business computing is at a crossroads. The cost to acquire personal computer (PC) hardware is decreasing rapidly while the costs to install, configure, maintain, and decommission that hardware (henceforth to be known as the service costs) are increasing at general economic inflationary rates or greater. The changing ratio of service costs to acquisition costs is causing substantial pressure on corporate information technology (IT) organizations to find a way to reduce the service costs associated with desktop computing.

Constraining IT options for reducing service costs is the strong demand from users for retention of the current model of decentralized control of their computing environment, namely PCs. This demand for user-managed control appears to stem from deep-seated negative perceptions and/or experiences of users derived from trying to accomplish their required computer usage in the centrally controlled architectures of the past.

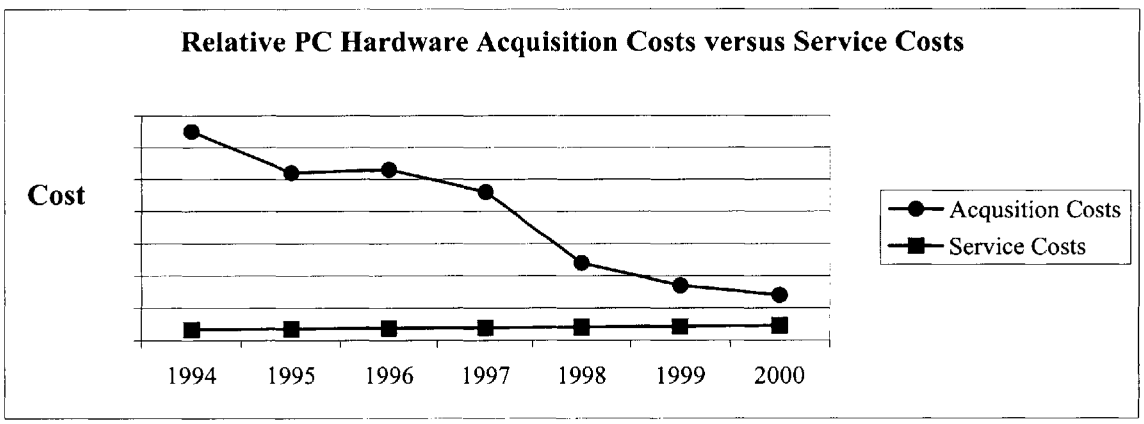
The struggle between IT's desire for centralized control of the corporate computing environment and the perceived need of users for relatively complete control over their computing destiny has not gone unnoticed by the computing industry. Indeed, there have been a wide variety of architectures proffered as solutions to the corporate desktop computing morass. These architectures span the spectrum from returning to "dumb" terminals with mainframes to completely mobile computing with wireless connections for everyone. Given the maturity of the problem domain and the solution space, the next dominant design will likely fall somewhere between these extremes.

This paper discusses the corporate desktop computing problem being caused by the drastically increasing ratio of the system service costs to the system acquisition costs and proposes a new architecture to address this problem.

Chapter 2: Problem Discussion

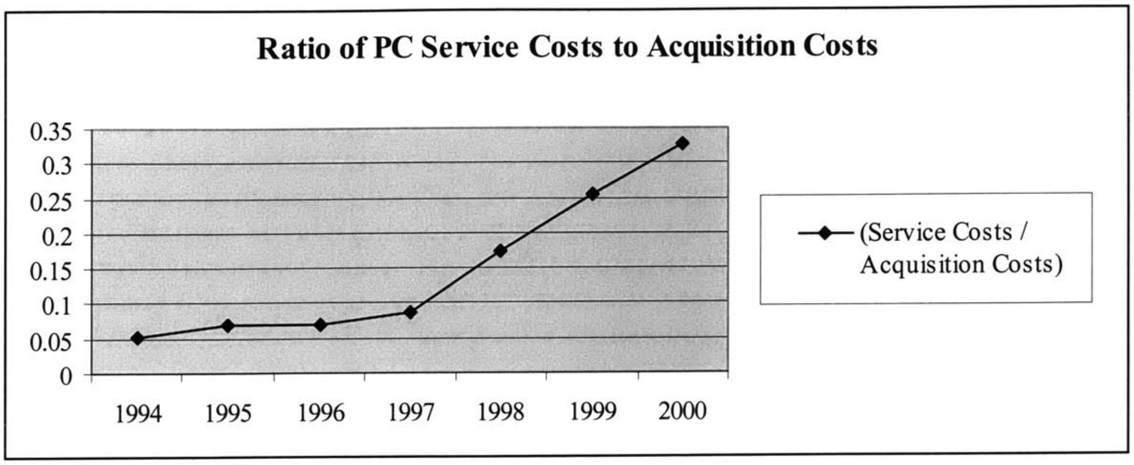
Figure 2-1 indicates the relative acquisition and service costs for PCs for the last six years plus a projection for year 2000. Figure 2-2 indicates the ratio of service costs to acquisition costs over the same seven years. As can be seen, while service costs are not increasing dramatically, they are becoming a substantially larger part of the Total Cost of Ownership (TCO) of the corporate PC-based computing environment. It is this increasing share that has attracted significant interest from IT managers and thus has become a fertile area for new solution development.

Figure 2-1



(Source: Ford Motor Company)

Figure 2-2

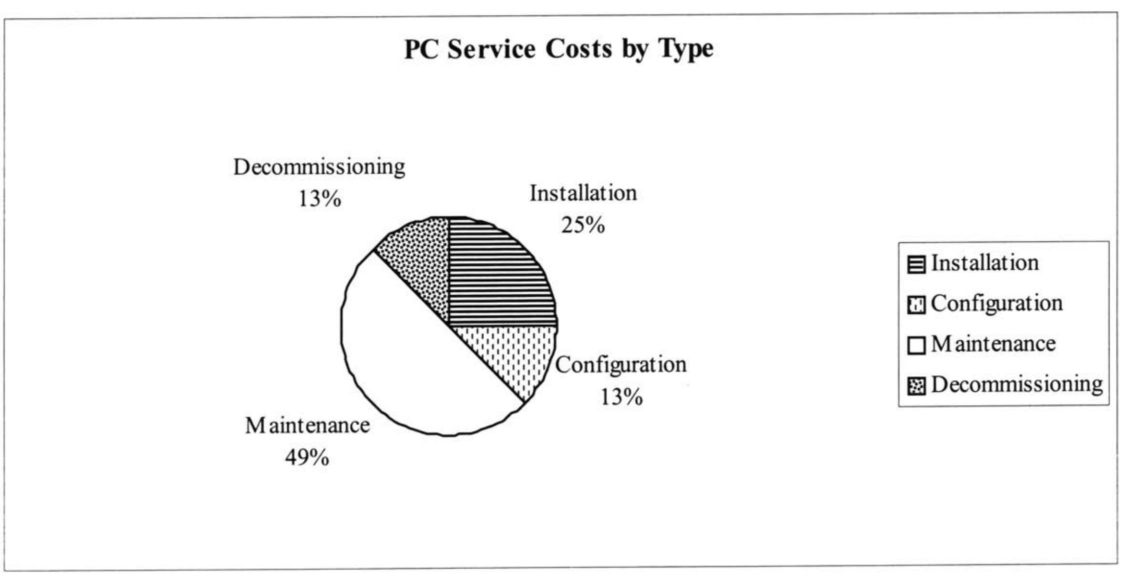


(Source: Ford Motor Company)

2.1 Service Cost

The service cost component of PC TCO is comprised of installation, configuration, maintenance, and decommissioning costs. The proportion of each cost relative to the entire service cost is shown in Figure 2-3.

Figure 2-3



(Source: Ford Motor Company)

2.2 Installation Cost

Since users are being provided with some type of equipment, regardless of its relative ability to be customized or configured, there is always some installation cost, even in situations where users pick up their equipment from a depot. Of course, equipment that is smaller, lighter, and less complex is easier to transport/install and thus is less costly to install as well.

2.3 Configuration Cost

Once installed, configurable devices (some auto-configuring or non-configurable devices can be envisioned) vary in the time and skill levels required to configure these devices to perform the tasks required by the user. For configurable devices, the configuration task may entail entering a few networking parameters, the creation or application of passwords, or the running of installation processes for one or more applications. In addition to the cost of basic task time for each configuration task, the support of a wide variety of user configurations can also significantly impact configuration times, required skills, and cost.

2.4 Maintenance Cost

Maintenance encompasses both the hardware and software aspects of the PC. Once installed and configured, hardware can malfunction or break, requiring one or more service visits to a user's location. Additionally, software may malfunction or damage the system's operability so that service visits are required. In either case, a number of factors affect the maintenance portion of the PC's TCO. These factors can include: the travel time for service personnel, the quality of the diagnostic tools available, the skill level of the service personnel, the variety of PC models and software supported, and the spare parts acquisition/software reload process timing.

2.4.1 Travel Time

The cost of travel increases with time and distance. In the ideal, a service person would not have to travel to provide maintenance for a user's desktop computer.

2.4.2 Diagnostics

The cost of maintenance is certainly dependent upon the speed and accuracy of the available diagnostic tools. In the ideal, the tools would be very fast and accurate and the same tool would be used for all systems.

2.4.3 Skill Levels

The required skill level for service personnel correlates strongly with the complexity of the system to be serviced. Once a system becomes significantly complex, non-linearities in the cost of acquiring and retaining sufficiently skillful personnel can become a significant component of desktop computing TCO.

2.4.4 Variety

The variety of PC models and the various software applications supported has an impact on maintenance costs. The impact can be minor to significant, again depending on the breadth of the variety.

2.4.5 Sparing and Reload Process

The time and capacity for obtaining spare parts and/or software reloads impacts desktop computing TCO substantially since PC downtime generally translates into unproductive time for users and that cost can become significant very quickly.

2.5 Decommissioning Cost

At the end of service life, all equipment must be decommissioned and that cost must be considered in system TCO. There is travel time for personnel, downtime for users, disposal costs, etc. that must be considered in the decommissioning cost factor.

To be accepted, any proposed changes to the current networked desktop PC architecture for general corporate computing will need to provide features which reduce the most significant components of service cost. But the same proposed changes must also support the most critical needs of both corporate IT and users. These needs are discussed in greater detail beginning in Chapter 6.

Chapter 3: Current Architectures

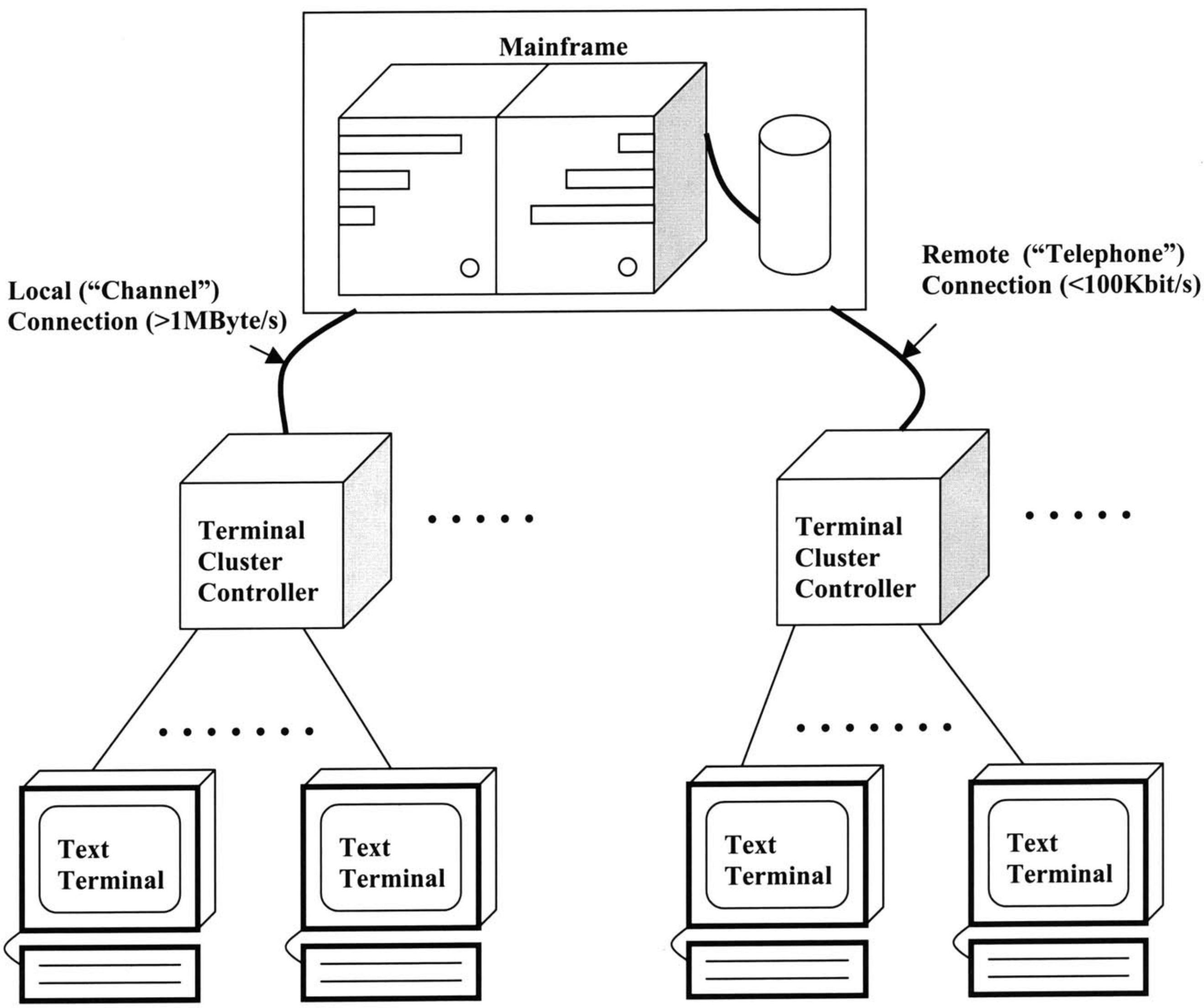
This chapter discusses the currently deployed architectures for corporate desktop computing. While each architecture is described individually in a section, an overall comparative perspective may prove useful to the reader while reading each section. This comparative view is provided in Figure 3-9 at the end of this chapter. The architecture-specific sections of this chapter are to be used in conjunction with Figure 3-9 to develop an adequate understanding of the features of each architecture.

3.1 Mainframe and Terminal

The first architecture that supported access to significant computational resources via a remote desktop device was the timesharing system architecture developed in the 1960s. This architecture is still in use today for applications whose usage requires only data entry and retrieval at the user desktop. This architecture is depicted in Figure 3-1.

Figure 3-1

Mainframe/Terminal Architecture



Key: ". . . . ." indicates potential plurality

The mainframe-based systems use either hardcopy terminals (all data is printed as entered and received) or via terminals employing Cathode Ray Tubes (CRTs), Liquid Crystal Displays (LCDs), or other dynamic display technologies. The terminals themselves have only data entry, transmission, reception, and display capabilities – no data manipulation or computational capabilities are included. In effect, these remote terminals are simply extended input/output ports of the mainframe computer system.

In this architecture, the physical connection between the desktop terminals and the mainframe computer are either dedicated wiring (for intra-campus connections) or telephone dial-up or leased line connections for truly remote terminals. In some cases, sophisticated intermediary elements such as terminal cluster controllers are included in the architecture to enable the mainframe computer to support large numbers of concurrent users.

Four developments in this architectural genre deserve mention here.

First, mini-computers, such as the seminal Digital Equipment Corporation VAX series, adopted the mainframe/terminal architecture during the mini's rise of the late 1970s and 1980s. The mini's typically use less costly (and lower speed) terminals which are directly connected to the mini (no cluster controllers or concentrators), causing mini-computer processing overhead for terminal support to be higher and thus reducing available computational power for applications.

Second, mainframe/terminal (and mini/terminal) architectures have acknowledged the pervasiveness of Local Area Networks (LANs) in corporations. As a response, the dedicated wiring for intra-building terminal connections is being replaced with LAN connections for these terminals. This migration is occurring as existing terminals are being retired or as LANs are being installed or upgraded.

Third, the deployment of desktop PCs with excellent data entry and display capabilities, plus substantial data manipulation and computational capabilities, has given rise to terminal emulation applications which run as one task on the multi-tasking operating systems of today's PCs. These terminal emulators allow users with PCs to eliminate the need for a dedicated terminal. This architectural change has effectively eliminated dedicated terminals for access to central computing applications in all but the most basic and cost sensitive environments.

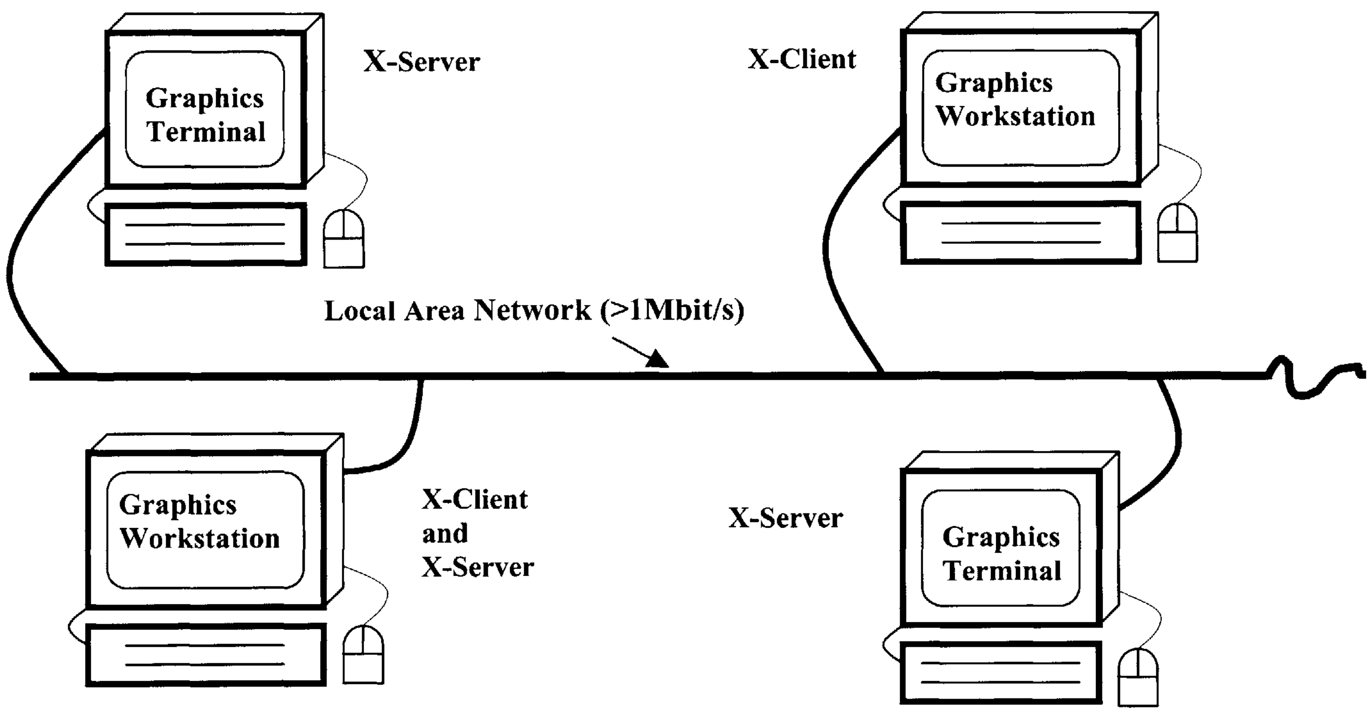
Finally, the issue of graphics versus text capabilities is an issue. The dedicated terminal market delivered the first "business graphics" capability (color displays, pixel-addressable displays, light pen or joystick graphical input devices) in the late 1970s. The cost was prohibitive and the resolution was moderate by today's standards. By the time the developers responded, the PC market had grown substantially and was delivering graphics superior to most of the dedicated terminals. This in turn discouraged many developers of mainframe or mini-computer software applications from developing applications for the dedicated graphics terminals and thus the market for these devices never really materialized.

3.2 X-Windows

X-Windows is a client/server system developed by a consortium of corporations and universities during the 1980s. X-Windows is the architecture, as well as the protocols, for the request, transfer, and management of graphical information in a windowed environment over a network. This architecture is depicted in Figure 3-2.

Figure 3-2

X-Windows Architecture



The concepts of server and client are reversed in X-Windows terminology versus most other architectures. Specifically, an X-Windows server is an application that runs on a client device, typically an engineering workstation, dedicated X-Terminal, or a PC, which provides X-Windows display management capabilities to any requesting X-Windows client application. The client application is any application, typically executed on a high-performance computer (most architectures would call this high-performance computer a "compute server"), which will require input or graphical output from a user via a keyboard, mouse, and graphical display interface device.

The X-Windows architecture is quite flexible, allowing the client and server applications to execute on the same computer or on separate computers. Additionally, the X-Windows server provides a standard, network-accessible interface and window manager so that many X-Windows client applications can concurrently share the mouse, keyboard, and graphical display of the user.

The X-Windows architecture is primarily utilized in engineering environments where access to compute servers is a frequent operation. The popularity of the architecture peaked in the late 1980s and is now in decline due to competitive pressures from PCs and lower-cost engineering workstations, among others. To extend the reach of the architecture, lower cost user devices, X-Windows "terminals," were developed. These X-Terminals were quite expensive and did not really offer much advantage over similarly priced PCs running an X-Terminal emulator application (an X "server"), so the X-Terminals did not achieve significant market adoption and are now almost gone from the market.

The most significant disadvantages cited for the X-Windows architecture are those of poor performance and substantial network traffic generation (bandwidth consumption), relative to other available protocols and architectures. Both issues are related to the use of very low-level protocols for the transmission of user input and output information. To achieve good performance, expensive computing and networking hardware is required, further diminishing the luster of the very flexible and widely supported (at least, in engineering circles) architecture.

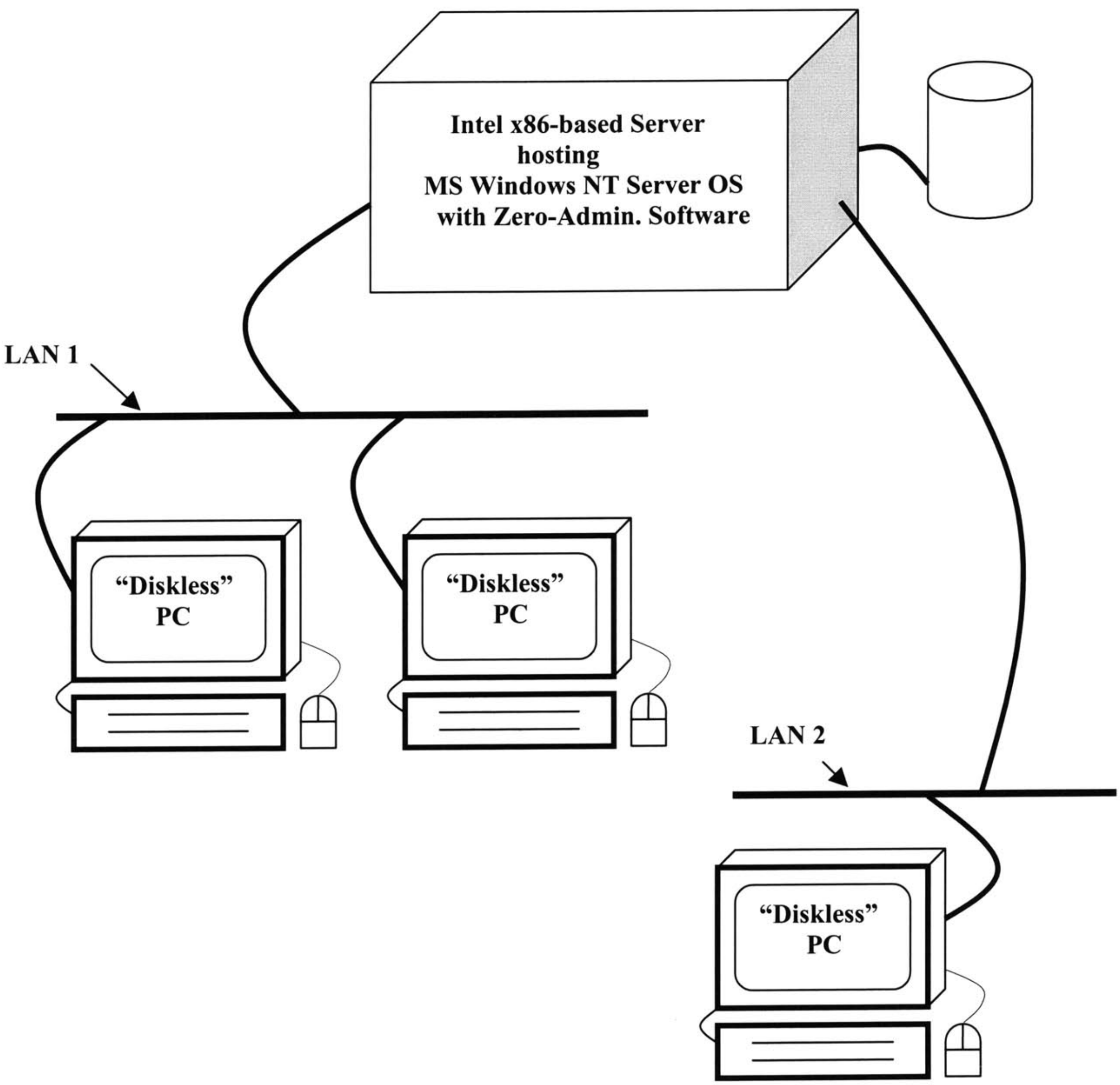
X-Windows has some interesting architectural features to consider in this investigation, but is clearly a declining architecture.

3.3 NetPC (Network PC)

The NetPC architecture was designed with client devices (NetPCs) that were intended to only run a special version of Microsoft Windows that would provide access to Microsoft-Intel compatible applications hosted on Intel-based servers running the Microsoft Windows NT Server operating system. This architecture is depicted in Figure 3-3.

Figure 3-3

NetPC Architecture



The NetPCs were "real" PCs in that they included all of the major features of a standard desktop PC, but the case was sealed to prevent expansion of the system's capabilities by end users.

The NetPC proponents, including most of the top-tier PC manufacturers, barely shipped any product before the movement ended. The NetPC initiative appears to have been a competitive stalling response from Microsoft and Intel to the Network Computer (NC) architecture being promoted by Sun and IBM. As such, it is not surprising that actual viable products were never well supported or aggressively sponsored.

The NetPC architecture experience provides a hint as to what users want in application and operating system compatibility and client hardware capabilities. However, the architecture failed to succeed in reducing service costs and complexity, two of the primary reasons cited for development of the architecture at the time. This too provides valuable insights that may prove useful in this investigation.

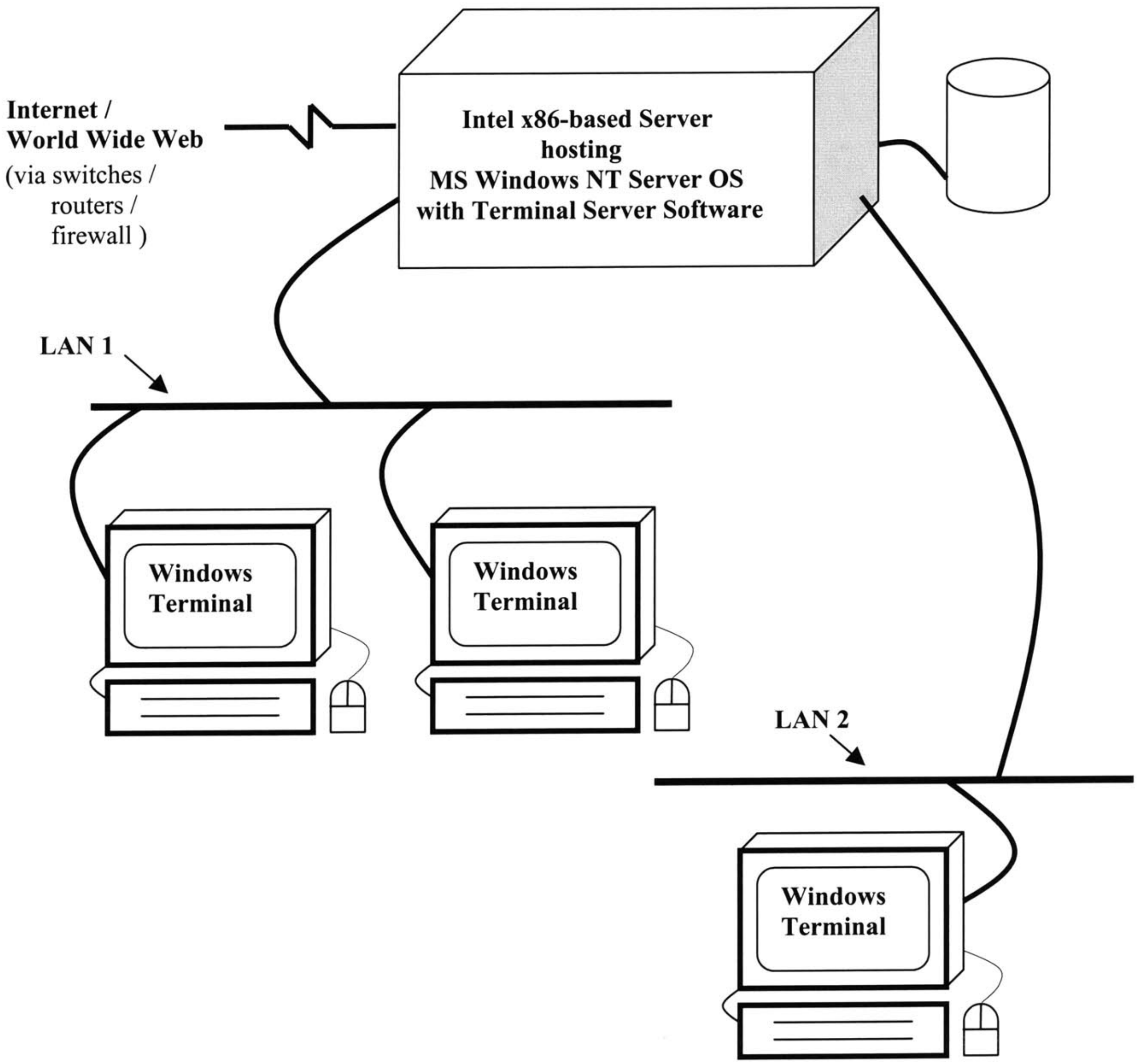
The NetPC architecture is nearly extinct – almost no manufacturers are producing these products.

3.4 Windows Terminals

While NCs and NetPCs were fighting for corporate Information Technology (IT) mind share, Citrix Corporation introduced a "thinner" client architecture based upon the Microsoft Windows NT Server operating system. Using a protocol named ICA (Independent Computing Architecture) that it developed, Citrix allowed Microsoft Windows compatible applications to be executed on the Windows NT Server in a multi-user manner (Microsoft NT Server was not designed as a multi-user application serving environment). ICA is a fairly low-level protocol which maps the console input/output from the applications across the network to clients running the ICA client software portion of the protocol. This Windows Terminal architecture is depicted in Figure 3-4.

Figure 3-4

Windows Terminal Architecture



With ICA, Citrix enabled the use of lower cost devices, such as special ICA-enabled terminals, UNIX workstations, and older PCs to be used to access Microsoft-Intel compatible applications with moderate network requirements.

While the ICA approach sounds very promising, the costs for adequate server performance are substantial, as are the support costs. Further, multi-media functions are not well supported, performance is significantly below that of desktop PCs, and many applications that will run on client versions of Microsoft Windows will not execute correctly (or require substantial repackaging effort before installation) in the Citrix/NT environment. Thus, ICA is primarily useful for mainstream MS Windows applications, such as MS Word, Excel, etc.

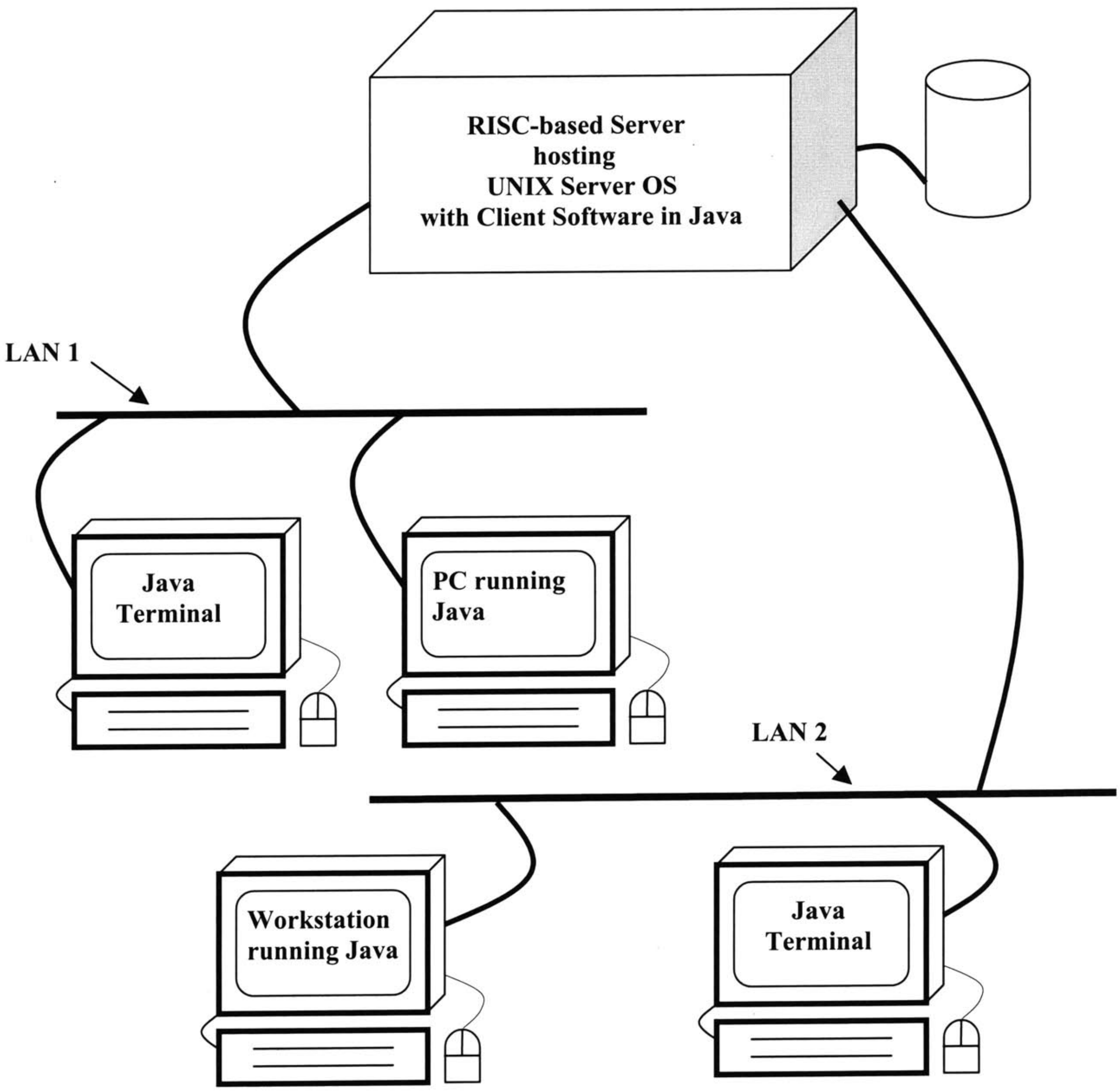
Microsoft recognized the value of the ICA solution as a tool for extending Microsoft products to non-PC users and thus acquired access to the ICA solution for use in its Windows NT Server – Windows Terminal Server Edition. Thus, this type of "thinner" client solution is now available in a mainstream manner.

3.5 JavaStation (a.k.a, Thin Client or NC)

The Network Computer (NC) architecture was developed to break the Microsoft-Intel hegemony in the desktop computing space. The architecture consisted of somewhat simplified client devices, known as Network Computers (NCs), Thin Clients (although this moniker is used for other architectures as well), or JavaStations, connected via a network to servers. The NCs were computers using non-Intel processors and lower cost electronics (compared to legacy-burdened PC electronics) and no hard disk. These computers powered up into a session manager running on top of an interpreted Java Virtual Machine and all applications had to be written in the Java programming language and downloaded to the NC from a server each time an application was needed. Some architectural improvement techniques were implemented by some proponents, including the caching of frequently utilized application code in the client device and Just-In-Time compilation of the Java code into a native executable format for the processor in the client device. Each of these techniques provided some performance improvements for the users, but the resultant performance was still substantially slower than PCs running applications locally. Additionally, the techniques moved the architecture further from its processor-independent underpinnings, thus weakening the architecture's overall appeal. This architecture is depicted in Figure 3-5.

Figure 3-5

Network Computer Architecture
(a.k.a. JavaStation or Thin-Client)



The NC architecture touted the grail of "write once, run anywhere," but poor run-time performance and a lack of business applications caused this architecture to fail to attain necessary adoption rates.

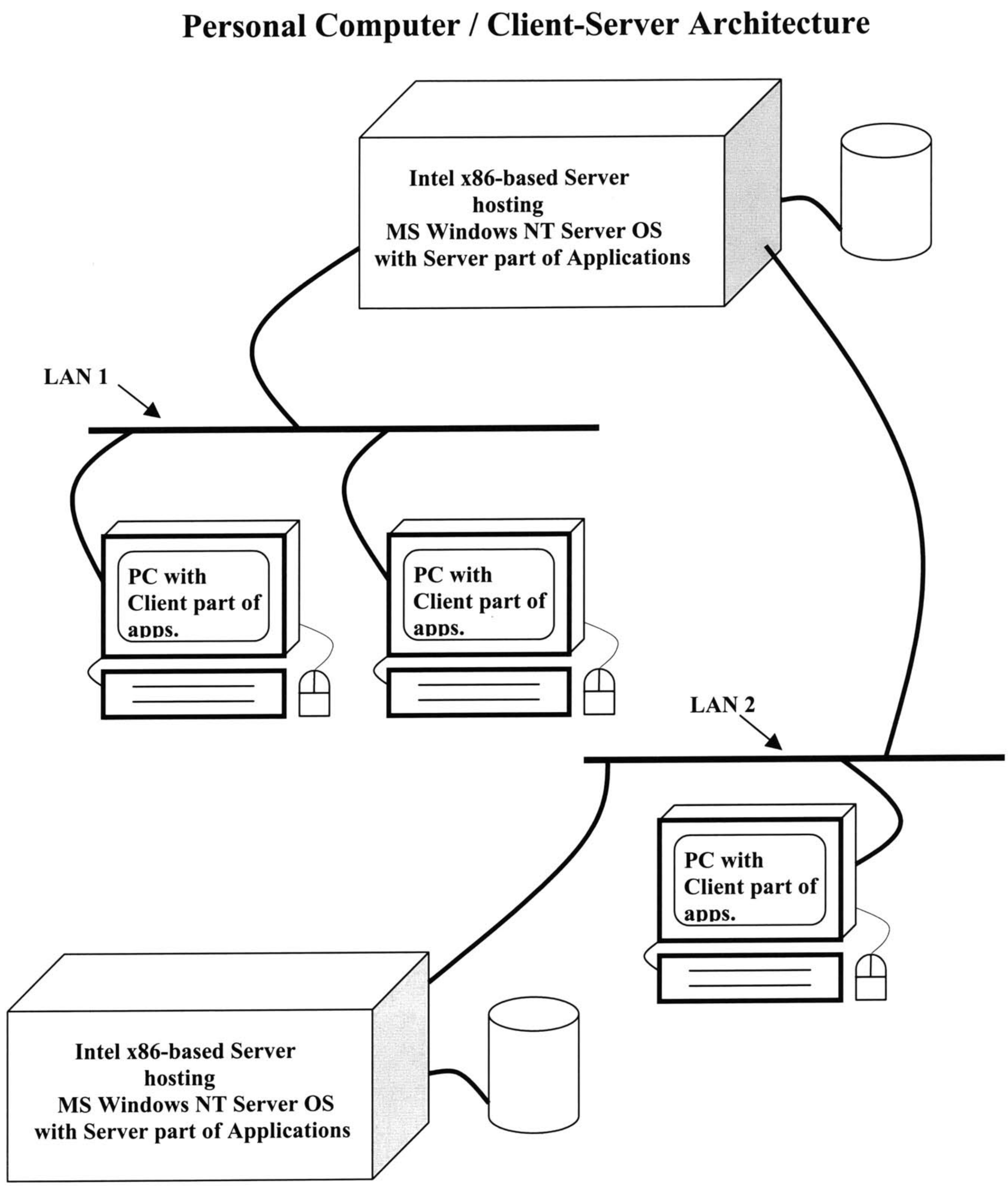
The NC architecture is effectively dead in the desktop computing space. Some revitalization of the concept is occurring in the Internet appliance domain, but the success of that foray is still highly uncertain.

3.6 PCs and Client-Server

PCs have been the dominant design for corporate desktop computing for at least a decade. The late 1980s and early 1990s saw the rise of the network as a key enabler in extracting more benefit from the installed base of isolated PCs. This network information flow was primarily from clients (PCs) to mainframes (via terminal emulation applications) and eventually from PCs to a new breed of computer called "servers."

The initial architecture built upon PCs and servers was called "Client-Server." This architecture is depicted in Figure 3-6. The functioning of the architecture required that each application, e.g. a system to record actual hours worked, be built in two pieces, a client piece and a server piece. Each application had its own client piece and server piece. Thus, as the number of Client-Server applications used on a particular PC increased, the PC's installed software became large and complex. Conflicts between client pieces of differing applications became common and the value of the Client-Server architecture reached its peak.

Figure 3-6

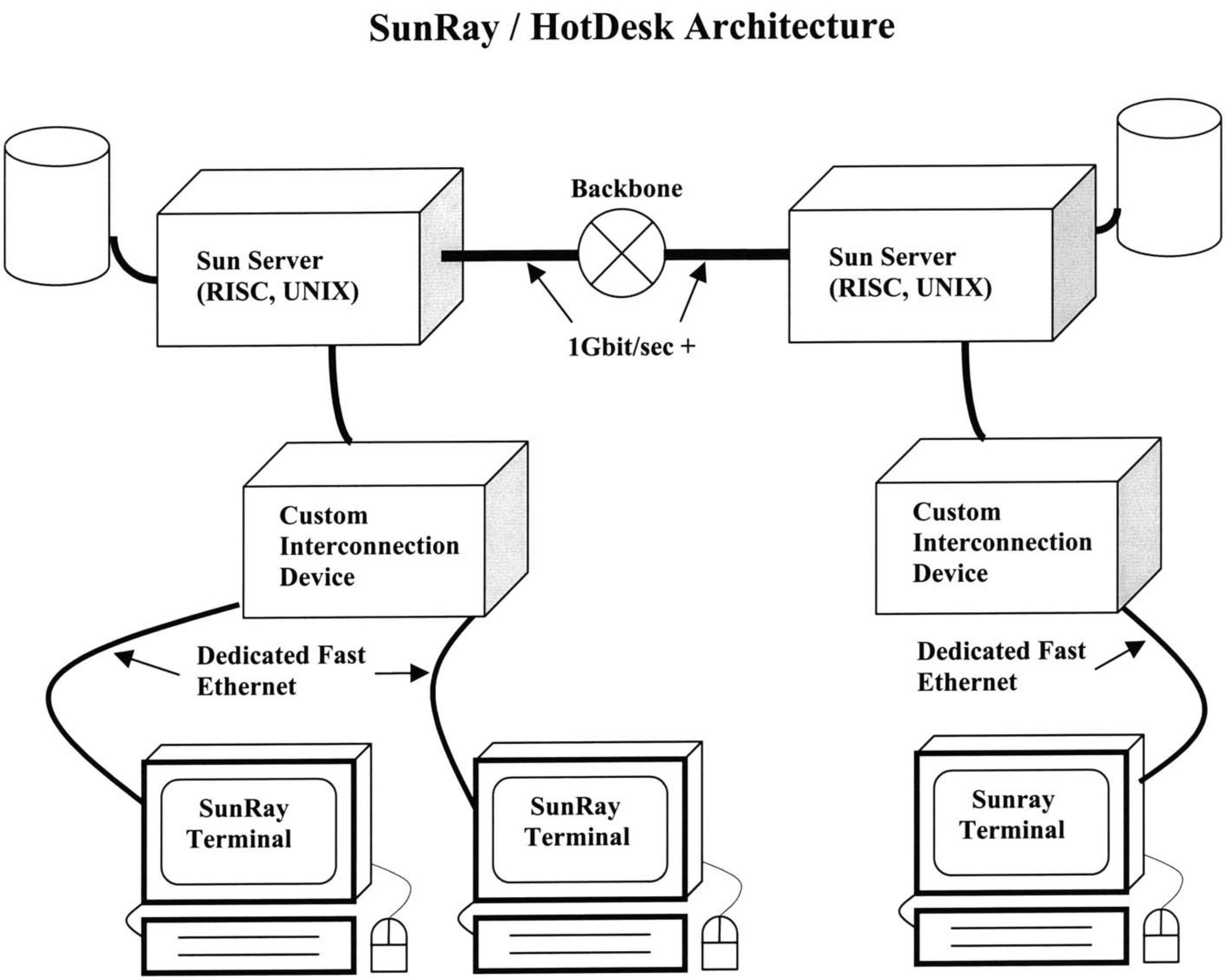

Personal Computer / Client-Server Architecture
Intel x86-based Server
hosting
MS Windows NT Server OS
with Server part of Applications
LAN 1
PC with
Client part of
apps.
PC with
Client part of
apps.
LAN 2
PC with
Client part of
apps.
Intel x86-based Server
hosting
MS Windows NT Server OS
with Server part of Applications

Client-Server as an architecture has matured somewhat since the chaotic days of the early 1990s. Standards have been adopted to eliminate the need for some of the unique client pieces, although many still exist. Client-Server is still a popular architecture and many corporate applications are being built upon it. However, the move to a World Wide Web-based ("Web") architecture for corporate applications is beginning to overtake traditional Client-Server development. The Web architecture is still Client-Server, except that the client piece is primarily just a Web browser, but possibly with some custom add-ins for unique capability requirements.

3.7 SunRay / HotDesk

Perhaps the most interesting architectural development in the corporate desktop computing space of recent years is the October, 1999 announcement by Sun Microsystems of their HotDesk technology and the first implementation of that technology using the SunRay client device and associated infrastructure. This architecture is depicted in Figure 3-7.

Figure 3-7


SunRay / HotDesk Architecture
Backbone
Sun Server
(RISC, UNIX)
Sun Server
(RISC, UNIX)
1Gbit/sec +
Custom
Interconnection
Device
Custom
Interconnection
Device
Dedicated Fast
Ethernet
Dedicated Fast
Ethernet
SunRay
Terminal
SunRay
Terminal
Sunray
Terminal

In HotDesk, Sun has attempted to revive the mantra that the "Network is the Computer" (a Sun phrase). The HotDesk architecture places all standard context for an end user in a Sun server. Thus, the user can theoretically move from client device to client device, anywhere that the network can provide connectivity between the client device and the server holding the user's context, and the user will see exactly the same computing environment. This is a highly desirable computing paradigm for most corporate users, particularly given the increasing mobility of the typical corporate workforce. However, network performance and routing realities make this "same environment, same performance" goal difficult, if not impossible, to achieve.

The SunRay client device is an inspired piece of architectural refinement. The device is consists of a display (Cathode Ray Tube or Liquid Crystal Display technology) and a keyboard and mouse interface. Additional features such as speakers and microphone may be included in certain models. Extensibility is supported via two version 1 (1Mbyte/s) Universal Serial Bus (USB) ports. The USB ports allow for the addition of moderate speed peripherals to the configuration, but at the risk of allowing context to be stored at the user's desktop, rather than only in the Sun server.

The SunRay device utilizes a triangle-based protocol for accepting display information from the Sun server over a Fast Ethernet (100Mbit/s) link, which is basically dedicated to the display task as the bandwidth requirements can be substantial, depending upon the change rate, resolution, and color depth of the display information. The use of triangles to reduce network traffic (as opposed to the lower level primitive of pixels) has both potential advantages and disadvantages that are discussed later in this thesis.

The architecture includes a specialized interconnect device between the Sun server and the SunRay client devices. The device handles the translation of server commands into network-compatible triangle information, along with other lower bandwidth communication information.

The architecture was introduced with a specific claim that the Interconnection and Client Device elements of the architecture could be used with any Central Equipment (server) architecture, not just the Sun server variants. However, to date, no other Central Equipment manufacturers have produced equipment which interoperates with the Interconnection and Client Device elements of the HotDesk / SunRay architecture.

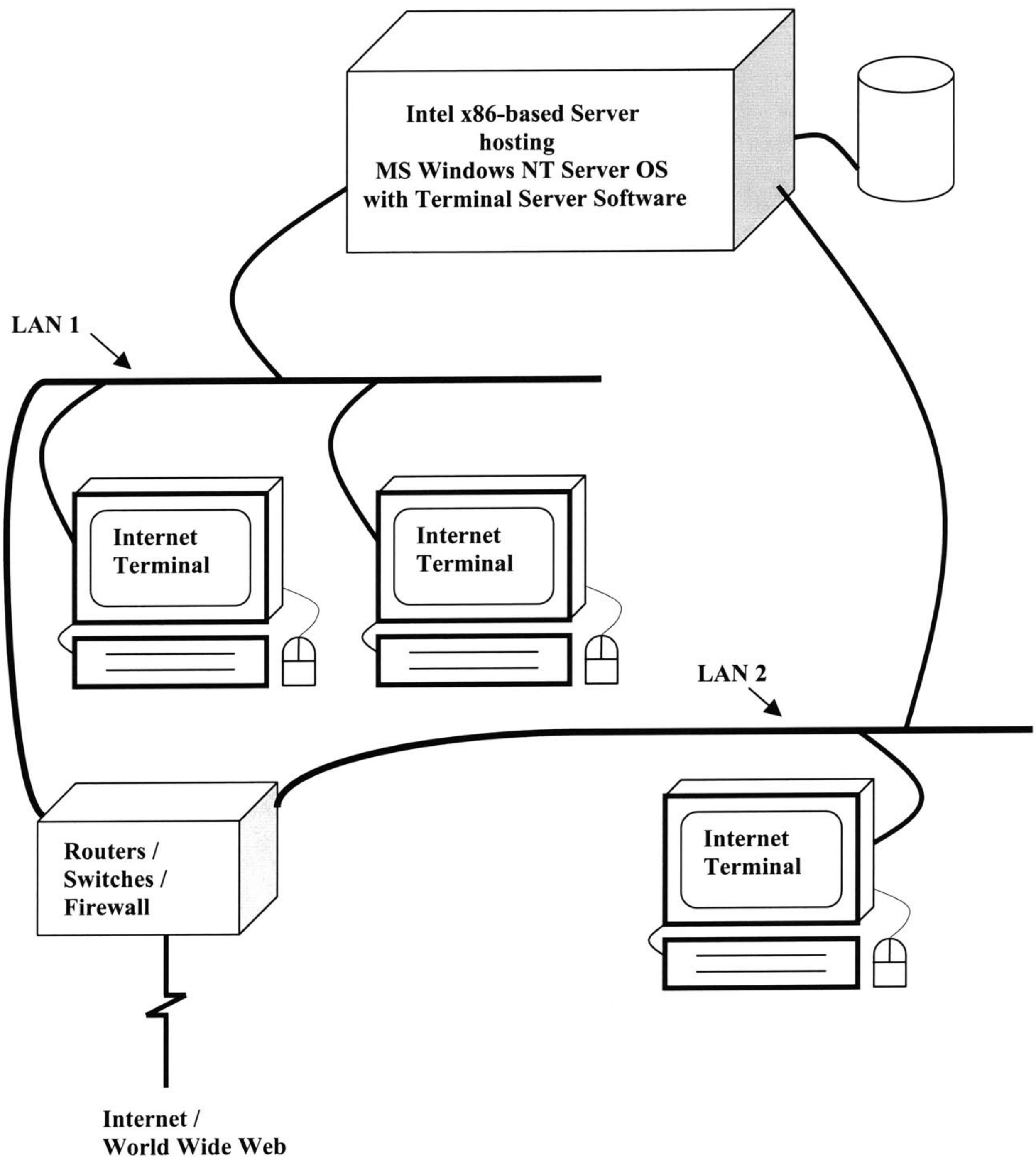
The fate of this architecture is still being determined. It has some very nice features, including a greatly reduced complexity profile for the client device and the use of new industry-standard technology (USB) for supporting extensibility at the end user's work area, but also a proprietary flavor and a strong server bias that together may prevent its widespread adoption.

3.8 Internet Terminals

Internet Terminals are a recently proposed architectural variant of Windows Terminals. The Internet Terminals would possess all basic Windows Terminal capabilities for accessing MS Windows-compatible applications running on a networked server that was running Microsoft Windows NT – Windows Terminal Server Edition. Additionally, Internet Terminals would contain an integrated "browser" application for accessing the World Wide Web directly. The direct access to the Web is intended to improve the performance of the device by bypassing the server. This architecture is depicted in Figure 3-8.

Figure 3-8

Internet Terminal Architecture



Internet Terminals are not readily available yet. One sticking point for this architectural extension is that it places persistent storage into the client device, thus increasing administration costs in keeping the browsers at a fixed level throughout the corporation. Additionally, because the Web is still in its infancy, often different browsers from different companies are required to access different Web pages – the Internet Terminal proposal does not directly address this issue.

The Internet Terminal architecture offers an intriguing bit of information about how architects have chosen to address sluggish network access times when servers are included in the architecture. This may be of significant benefit in this investigation.

Figure 3-9

Feature Comparison of Various Corporate Desktop Computing Architectures

	Mainframe and Terminals	**X-Windows**	**NetPC**	**Windows Terminals**	**NC**	**Client - Server (PCs)**	**SunRay / HotDesk**	**Internet Terminals**
Client Operating System	Not Appl.	UNIX or Windows	Windows 9x/NT	Not Appl.	Java VM	Windows 9x/NT		Not Appl.
Host / Server Operating System	Proprietary	UNIX	Windows NT	Windows NT - Terminal Server Edition	Any that can serve Java Apps	Windows NT (others possible, but not popular)	Not Appl.	Windows NT - Terminal Server Edition
Applications	Limited	Scientific / Engrg.	Repackaged MS Windows Apps.	Repackaged MS Windows NT 4 Apps.	Star Office, Java Apps	All MS Win. Apps. Incl. Legacy Apps.	All Sun Solaris Apps.	Repackaged MS Windows NT 4 Apps.
App. Storage Location	Mainframe	Host ("X-Client")	Intel Server	Intel Server	Server	Client	Sun Server	Intel Server
App. Execution Location	Mainframe	Host ("X-Client")	Client NetPC	Intel Server	Client	Client	Sun Server	Intel Server
Mobile Computing Support	No	No	Depends	No	No	Yes	No	No
Disconnected Comp. Support	No	No	No	No	No	Yes	No	No
Roving Computing Support	Yes	Yes	Some	Yes	Yes (wired)	Wireless - Yes; Wired - Y2000;	Yes (wired)	Some
Peripherals Support	Printers	Marginal	Standard PC Peripherals	Insignificant	Insignificant	Substantial	Insignificant	No
Performance for User	Varies Widely	Moderate	App. Start slow Due to Network; App. Run Time Same as PC	Moderate	Slow (Java is Interpreted)	Excellent	Bandwidth / Responsiveness is TBD	Moderate
Connection to Host / Server	LAN or Dedicated	LAN	LAN	LAN	LAN	LAN	Dedicated Wired Link	LAN
Hardware or Software Capacity Limitations	Terminals Not Upgradable	Terminals Not Upgradable	No Local Storage -- Server Storage always in short supply	Typically Not Upgradable	Yes	None Significant	Bandwidth and Server Performance May Limit Scalability	Typically Not Upgradable
Form Factor	Monitor + Keyboard	Monitor + Keyboard + Mouse	Small Form Factor PC	Small Desktop Case or Integrated in Monitor	Small Desktop Case (Next to or Under Monitor)	Wide Variety	Small Desktop Case (Next to or Under Monitor)	Small Desktop Case or Integrated in Monitor
Network Overhead	Minimal to Moderate	Moderate to Heavy	Moderate to Heavy	Moderate to Heavy (User Interface is Managed over the LAN)	All Data and Apps. On Server; Load Will Vary Significantly	Minimal	Not on LAN -- Dedicated Link per user	Moderate to Heavy (User Interface is Managed over the LAN)
Availability	Yes	Yes, but declining	One or two suppliers remaining	Several Hardware and Software Suppliers	Very Few Suppliers	Many Suppliers	Only from Sun	Soon from Several Hardware and Software Suppliers
Accessibility for Users with Disabilities	Some	Some	Yes	TBD	TBD	Yes, Well Established	TBD	TBD
Security Issues	Minimal	Minimal	Same as PCs Without Local Storage Concerns	Minimal	TBD	Yes, Unencrypted Local File Storage and Viruses	TBD	Minimal

Chapter 4: Analysis Methodology Plan

4.1 Existing Concept Search

For many design efforts, the engineer has envisioned a new concept or combination of features for a product or architecture based upon his/her subject matter knowledge and experience. Before the formal customer requirements gathering process is undertaken, a search of existing literature and patents is often undertaken to prevent fruitless analysis and design efforts. This is usually a highly valuable effort.

4.2 Customer Requirements Identification

The most important initial step in the analysis and design process is the identification and prioritization of customer needs or wants, which are often referred to as requirements. This is somewhat misleading terminology as "requirement" is typically intended to imply that a feature is mandatory whereas the customer requirements processing in this investigation will prioritize the requirements, implying an optional attribute for these "requirements." Subsequent to the prioritization, synthesized concepts and designs may not completely achieve implementation of features to fully address one or more of these customer needs and yet the selected architecture may provide the best balance in addressing the needs as a whole. To recognize the not-quite-mandatory nature of the customer input to this investigation, the term "customer needs" is henceforth employed, rather than "customer requirements," except in a few instances where the use of "customer requirements" is more appropriate.

Many methods exist for obtaining customer needs, including questionnaires, concept demonstrations/prototypes, observation of customer usage of similar existing equipment, interviews, and focus groups, to name a few of the most popular. The method for this investigation is questionnaires.

Before the questionnaire can be developed and distributed, the customer must first be identified in order to ensure relevant responses to the questionnaire are obtained. The traditional interpretation of "customer" for a computer system or architecture is that of an "end user," or someone who uses a computer system to accomplish a task in support of a business process. During the initial stages of this investigation it became clear that typical end users would not provide useful input for the needs gathering process activity of this investigation for the following two reasons:

1) The focus of this investigation is at an architectural level rather than at an implementation details level and typical end users are most often very detail- and implementation-oriented and are ineffective when considering conceptual and architectural level information, and
2) This investigation deals with the entire architecture life cycle, including many aspects that are not usually encountered by end users, but rather are handled by Information Technology staff members.

For these reasons, the customer surrogates for this investigation were Information Technology specialists at several major computer system manufacturers and computer system "user" companies.

4.2 QFD Requirements Matrix Development

The methodology chosen to assist in the transformation of customer needs information into product design/feature information in this investigation is the Quality Function Deployment (QFD) methodology[1]. Specifically, this investigation uses the techniques of the QFD Requirements Matrix and Product (Design) Matrix.

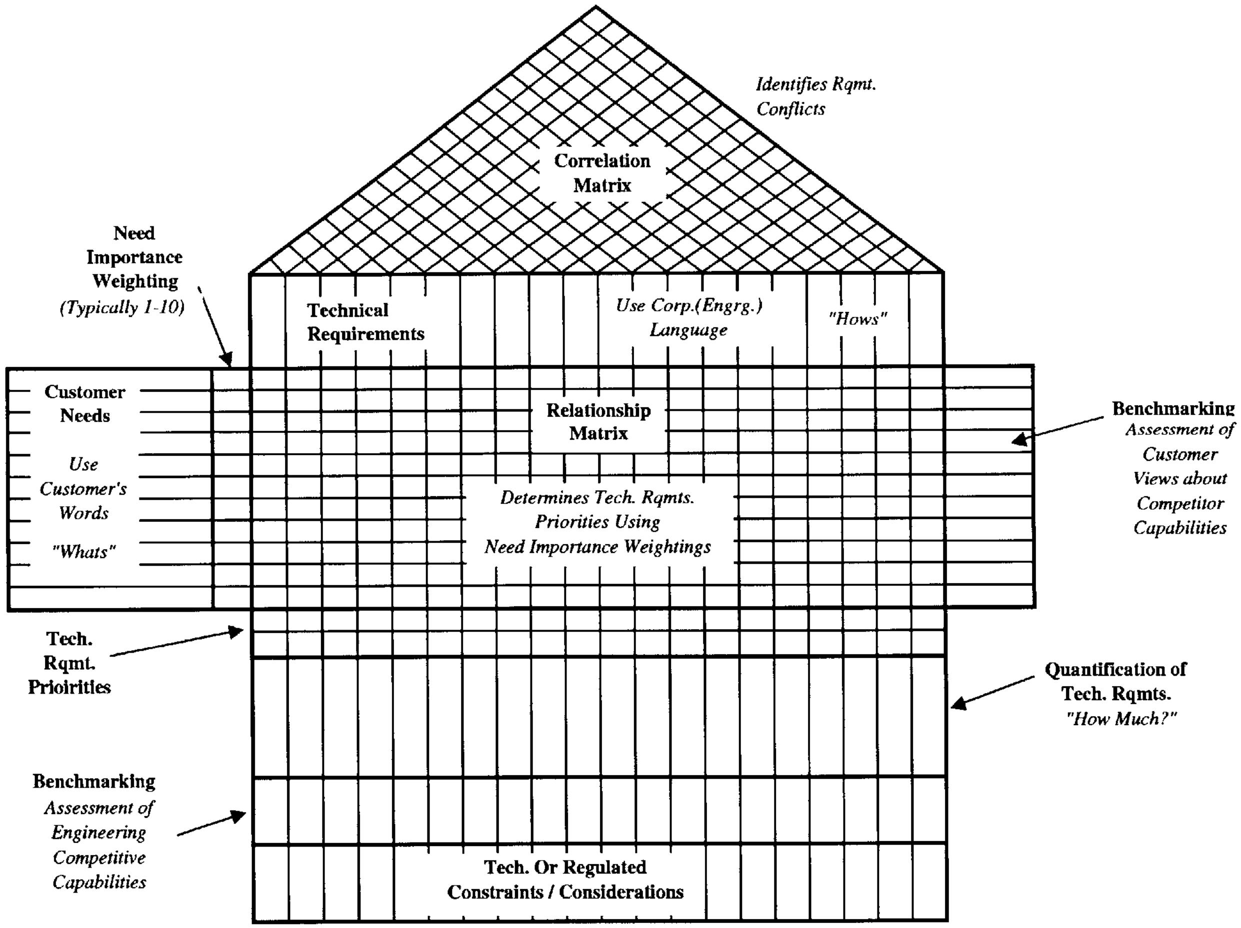
The QFD methodology is a product-process attribute development process that provides support for the following important analysis and design activities:

1) Translation of customer needs
2) Minimization of human biases
3) Prioritization of technical requirements
4) Provision of requirements traceability
5) Provision of a common framework and communications mechanism
6) Linking of competitive benchmarking information into the design process

The development of the QFD Requirements Matrix (see Figure 4-1) begins with the listing of customer needs in the left side of the matrix, one requirement per line. (For specificity, references to the contents of the customer needs section of the QFD Requirements Matrix will henceforth be capitalized as Customer Need(s).) Often there are many Customer Needs and thus only the highest priority items are included to begin the process. This investigation employs ten Customer Needs at this stage.

Figure 4-1

QFD Requirements Matrix Block Definitions



The customer importance weight for each need is then included in the column to the right of each Customer Need. This weighting is usually on a scale of 1 to 10 with 10 being the most important from the customers' perspective.

In some cases, groups or themes of needs are apparent. These categories are included by listing them to the left of the Customer Needs. This categorization can assist in discussions of intent later in the QFD process.

Once the Customer Needs have been entered into the Requirements Matrix, a list of Technical Requirements is generated. These Technical Requirements are intended to be implementation neutral but are to provide an intermediate step necessary to synthesize an architecture or product to effectively address the Customer Needs. The Technical Requirements are listed in the Requirements Matrix as one per column across the top row of the matrix.

Once the Customer Needs and Technical Requirements are entered, the Relationship Matrix portion of the QFD Requirements Matrix is completed. The Relationship Matrix entries define the strength of the relationship between each Technical Requirement and each Customer Need. A common practice is to use graphical symbols for the strength values entries. This investigation uses three strength value symbols, Strong, Moderate, and Weak, with corresponding numeric strength values of 9, 3, and 1 (on a 1 to 10 scale).

Each Relationship Matrix entry relates a Technical Requirement to a Customer Need by forming a sentence of the form "Technical Requirement X will have a [Strong, Moderate, or Weak] effect on helping the system satisfy Customer Need Y."

Once the Relationship Matrix is completed and reviewed, most often by a team, the Importance Weight and subsequent Relative Weight of each Technical Requirement is computed. The Importance Weight for a Technical Requirement is the sum of the values of each cell in the Technical Requirement's column multiplied by corresponding Customer Importance Weights for each Customer Need's row. For instance, if an entry in the Relationship Matrix for Technical Requirement X's effect on Customer Need Y indicates a Moderate (value = 3) effect and the Customer Importance Weight for Customer Need Y is 7, then the Importance Weight for that cell is 3x7=21. The Importance Weights for all cells in the column of each Technical Requirement are thus calculated and summed to provide the overall Importance Weight for each Technical Requirement. The Importance Weights are entered on the first row of the QFD Requirements Matrix below the Relationship Matrix.

The Relative Weight for a Technical Requirement is the rank of its Importance Weight relative to the Importance Weights of all of the other Technical Requirements. These Relative Weights are easily calculated and entered into the QFD Requirements Matrix on the row directly below the Importance Weights of the Technical Requirements.

The next step in completing the QFD Requirements Matrix is the preparation of the Correlation Matrix. The Correlation Matrix documents potential interactions between Technical Requirements. In particular, this matrix is valuable in identifying possible negative interactions or conflicts between Technical Requirements. The value in this process step is the identification and communication of these possible conflicts very early in the design process so that the conflicts can be addressed before too many decisions have been made which would be costly to change.

Additional fields are present in some implementations of QFD Requirements Matrices. These fields include:

A) A Benchmarking section which assesses customer views on competitor's capabilities for each Customer Need;
B) A Quantification of Technical Requirements section, which addresses the "how much" question for each Technical Requirement;

C) A Benchmarking section which assesses the engineering capabilities for each Technical Requirement; and
D) A Technical Constraints and Regulations section.

From the above list, field A is omitted for this investigation as each of the Customer Needs is addressed effectively by at least one existing architecture while simultaneously addressed ineffectively by at least one other existing architecture. Thus, the bar graph in this field for each Customer Need would be a solid bar from 1 to 5 and would provide no useful information.

Field B's value is dependent upon the specific Technical Requirements derived. As such, it is an optional element in this investigation.

Field C is deemed unnecessary since an actual implementation by a specific organization is beyond the scope of this investigation.

It is not clear if Field D would be valuable, so it is left as optional in the process with the intent that it be completed only if it appeared that one or more Technical Requirements might be substantially affected by Technical Constraints or Regulations at the architectural level.

4.4 Conceptual Level Trade Study

The method for comparing concepts during the analysis phase of this investigation is to formulate several concepts which address the QFD Technical Requirements with the highest Relative Weights, then rate the concepts against the current dominant design architecture (Intel-based Personal Computers connected via Ethernet LANs to Intel-based File and Print Servers). This rating process employs a Pugh[2]-type "comparative" (non-numeric) scale with a goal of identifying the concepts which best address the most important QFD Technical Requirements. The result of this analysis step is one or two "best" concepts that are carried into the synthesis phase of this investigation.

4.5 Summary of a Preferred Conceptual Level Architecture

The Pugh-type analysis presents a view of some good potential architecture concepts very early in the design process, focused only on the highest priority QFD Technical Requirements. An optional summary section may contain a view of the preferred concept or concepts relative to *all* of the Technical Requirements if deemed appropriate upon completion of the previous process steps.

Chapter 5: Synthesis Methodology Plan

5.1 QFD Product Matrix Development

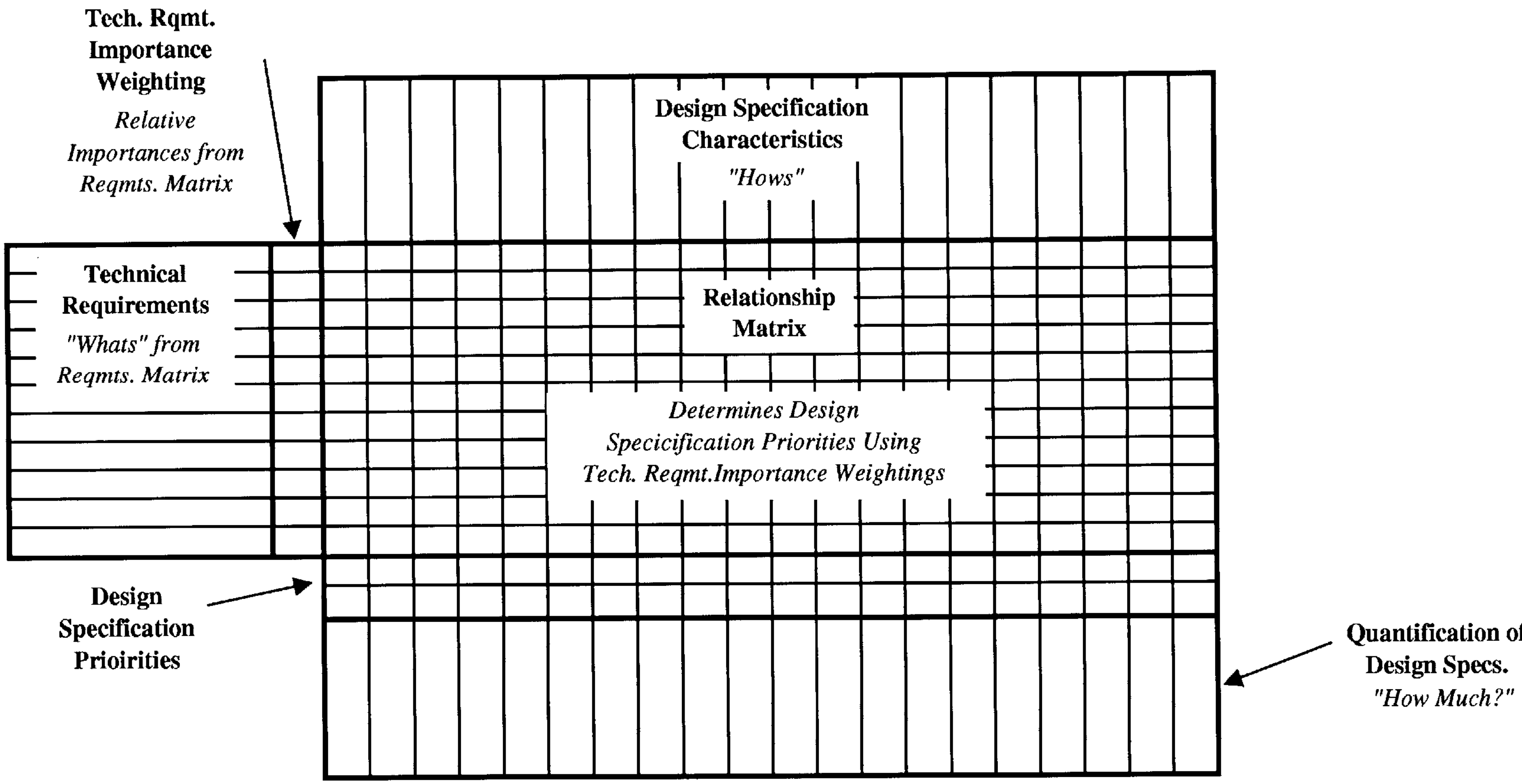
The next step in the formal QFD process, after the completion of the Requirements Matrix, is the development of a QFD Product Matrix (see Figure 5-1). The Product Matrix defines the relationships between the Technical Requirements (the "whats") from the Requirements Matrix and newly created Design Specification (Subsystem) Characteristics (the "hows"). Moving from the Requirements Matrix into the Product Matrix takes the analysis and design process from the Requirements Analysis phase into Product Design phase.

To develop the Product Matrix, the Technical Requirements from the Requirements Matrix are listed on the left side of the Product Matrix, one per line, with the Relative Weight placed to the right of each Technical Requirement. The list of Technical Requirements can be either sorted or unsorted by Relative Weight when transferred into the Product Matrix, depending upon individual or team preference.

With a significant number of Technical Requirements (for example, more than ten), the Product Matrix has the potential to be significantly larger than the Requirements Matrix. One approach to managing this growth in the sizes of subsequent matrices as the QFD process progresses toward more specific product and process definition levels is to drop the less important requirements with each step. This investigation carries all Technical Requirements through from the Requirements Matrix into the Product Matrix, where practical.

Figure 5-1

QFD Product (Design) Matrix Block Definitions



Once the Technical Requirements are entered into the Product Matrix, with their associated Relative Weights, the process of defining Design Specification (Subsystem) Characteristic(s) begins. These Subsystem Characteristic(s) are intended to be implementation options that can be employed to provide the functions and features defined by the Technical Requirements. Each Subsystem Characteristic may address the needs of one or more Technical Requirements. Similarly, one or more Subsystem Characteristic(s) may address each Technical Requirement. The Subsystem Characteristic(s) are listed, one per column, in the top row of the Product Matrix.

The relationship between each Subsystem Characteristic and each Technical Requirement is captured in the corresponding column-row cell of the Relationship Matrix portion of the Product Matrix, in a manner similar to the Relationship Matrix portion of the Requirements Matrix. That is, symbols for Strong, Moderate, and Weak relationships are entered in each column-row cell to complete a sentence of the form "Subsystem Characteristic W will have a [Strong, Moderate, or Weak] effect on helping the system satisfy Technical Requirement X."

Once the Relationship Matrix portion of the Product Matrix is completed, then the calculation of the Design (Subsystem) Characteristic priorities, both absolute and relative, is undertaken. The calculation method is directly analogous to that used to calculate the Importance Weights and Relative Weights for the Technical Requirements in the Requirements Matrix, namely the multiplication of each Relationship Matrix symbol entry's value by the Relative Weight of each Technical Requirement, followed by the summation of the products on a column basis for each Subsystem Characteristic. Then these Subsystem Characteristic Importance Weights are ranked relative to each other and a prioritized Relative Weight entry for each Subsystem Characteristic is entered below the corresponding Importance Weight in the Product Matrix.

An optional section of the Product Matrix is the row of cells below the Subsystem Characteristic Relative Weights row. This row is provided to allow for the quantification of Subsystem Characteristic(s) by defining "how much" of the Subsystem Characteristic (the "how") is required. This section is optional for this investigation.

5.2 Design/Implementation Level Trade Study

Once the Product Matrix is completed, a second Trade Study is conducted. This trade study utilizes the preferred concept(s) from the Pugh-type Trade Study conducted earlier as the basis for more detailed architectural options to be considered here. Several architectures are synthesized using the Relationship Matrix information of the Product Matrix such that these architectures, fitting under the umbrella of the preferred concept(s), address the highest priority Technical Requirements using various viable mixes of Subsystem Characteristics.

Once these candidate architectures are synthesized, a Pugh-type comparative scoring matrix is developed to contrast these architectures against the baseline architecture and each other. The objective of this exercise is to provide information for the synthesis of a final architecture.

5.3 Synthesis of a Preferred Implementation Level Architecture

Using the comparative information from the Product Matrix-based architecture comparison, the synthesis of an optimized Implementation-Level architecture is completed by adopting the best Subsystem Characteristic(s), either individually or in subsystem themes or groups from each candidate Implementation Level architecture feature into one preferred architecture.

Chapter 6: Application of Methodologies

Up to now, this document has discussed the business problem driving this investigation, as well as currently deployed solutions. Additionally, the plan for the analysis and synthesis process that is key to this investigation has been defined. This chapter documents the actual process as executed. Deviations from the planned ideal process are referred to as "tailored process" versions of the planned process steps.

6.1 Cost Versus Usage

In developing this thesis, cost was the fundamental driver. However, cost numbers for IT activities are either considered highly confidential by corporations or are of questionable accuracy when available from industry consultants. For this reason, the cost issue has been discussed, but the analysis and synthesis efforts of this investigation have been undertaken with usage (supported by functions and features) perspective, rather than a detailed cost perspective. It is expected that an architecture that better addresses the primary customer usage requirements, which include Minimizing Cost, would also be a cost-effective architecture if Total Cost of Ownership is employed as the measure of effectiveness.

6.2 Existing Concept Search

The initial step taken was a review of several computer architecture textbooks and documents found on the Internet at the web sites of various universities and computer equipment companies. This search revealed no concept that appeared to address the business problem better than the existing architectures discussed in Chapter 3.

The next step was a search of the United States patent database. A list of 800 patents with possible relevance to the architectural subject was identified by keyword and category searching at a United States Patent Library service center. This list was then used to retrieve and review the abstracts of the most relevant 200 patents using a publicly accessible Internet web site provided by the International Business Machines Corporation (http://www.patents.ibm.com/ibm.html as of this writing). This extensive effort revealed no patents covering the architectural problem being considered.

6.3 Customer Requirements Gathering

Customer Needs (Requirements) were obtained by sending a very detailed questionnaire (26 pages, 170 questions) to Information Technology specialists at several major corporations, including both computer system manufacturers as well as computer system "consumer" companies. This questionnaire is provided in Appendix B.

The "questions" in the survey were actually statements which posed a condition regarding an aspect of any new desktop computing architecture and then asked for a rating from the respondent on a scale of 1 to 10 where 10 indicated that the proposal would be a mandatory feature of any new desktop computing architecture and 1 would be an absolutely not acceptable feature. Where necessary, the condition was posed as being relative to a widely recognized architecture's features or capabilities or to the generally deployed set of architectures.

The responses were tabulated per question and the mean, median, mode, and standard deviation were calculated. At that point, a subjective final value was developed using the raw data and the calculated measures. This final subjective filter was used to reasonably accommodate non-answers (respondents were allowed to respond as "not applicable") as well as any apparently incorrect interpretations of statements. The subjective filter did very little to the mean in most cases, but a few statements did have their final rating modified somewhat due to very large standard deviations or other anomalies via this filtering process.

Once the ratings were finalized for each statement, the ratings of statement groups were examined to determine the overall customer position on the group's theme. The themes then became the first set of customer needs resulting from the customer survey process. At this point, the set contained 47 items, down considerably from the 170 statements, but still too large of a group to manage using the QFD process.

The next step was to filter the set of 47 customer needs by importance rating. The filter applied was to drop all needs that had a customer importance rating of six or lower. This left 28 items, another good reduction, but still too many customer needs to deal with effectively in the QFD Requirements Matrix process step to follow. The next step was a careful review of the remaining 28 items and the creation of 10 aggregated Customer Needs, such as Easy To Use, Easy To Administer, etc. Each of the 28 items was mapped to one of the 10 Customer Needs, the ratings combined arithmetically, the 7-10 span was scaled to the 1-10 range. Figure 6-1 shows the filtering and aggregation results while Figure 6-2 lists the final 10 Customer Needs.

Figure 6-1

Customer Requirements Affinity Processing Worksheet

Ref #	Want	Importance to Customer	Index	Aggregation Category	"Super Category"	Aggregation Category (Index) 1	2	3	4	5	6	7	8	9	10
12	Fast, easy, cheap data creation	9	1	Easy To Use	Primary Mission	9									
4	Non-expert app install/maintain	8	2	Easy To Administer/Maintain	Support Mission		8								
5	Access to core and non-core apps	8	3	Maximize Flexibility	Primary Mission			8							
9	Low acquisition cost	8	4	Minimize Costs	Asset Management				8						
10	Fast, easy, cheap data access	8	1	Easy To Use	Primary Mission	8									
11	Fast, easy, cheap data sharing	8	1	Easy To Use	Primary Mission	8									
15	Individual storage on mobile devices	8	3	Maximize Flexibility	Primary Mission			8							
27	Graphics and text interface	8	1	Easy To Use	Primary Mission	8									
1	Access MS Windows Apps (either form)	7	3	Maximize Flexibility	Primary Mission			7							
2	Access non-Windows Apps	7	3	Maximize Flexibility	Primary Mission			7							
6	Rapid app deployment	7	2	Easy To Administer/Maintain	Support Mission		7								
7	Multiple hardware sources	7	5	Maximize Choice	Asset Management					7					
8	Hardware is/will be a dominant design	7	5	Maximize Choice	Asset Management					7					
14	Individual storage both cental and distributed	7	3	Maximize Flexibility	Primary Mission			7							
16	Hosting of data entry/display software distributed or both	7	3	Maximize Flexibility	Primary Mission			7							
17	Hosting on mobile devices	7	3	Maximize Flexibility	Primary Mission			7							
21	Reliability	7	6	High Availability	Support Mission						7				
22	System monitoring/diagnostics	7	6	High Availability	Support Mission						7				
24	Industry standards (wire type, distance, link types, protocols)	7	7	Maximize Interoperability	Asset Management							7			
33	Low cost for upgrades	7	4	Minimize Costs	Asset Management				7						
34	Automatic backup and restore	7	2	Easy To Administer/Maintain	Support Mission		7								
35	Allow user-perfomred backup/restore	7	2	Easy To Use	Support Mission		7								
37	Removable media at client	7	3	Maximize Flexibility	Primary Mission			7							
38	Extensibility at client (current/TBD devices)	7	8	Maximize Extensibility	Support Mission								7		
39	Theft deterrence at client/system	7	4	Minimize Costs	Asset Management				7						
44	Effective support for all mobility classes	7	3	Maximize Flexibility	Primary Mission			7							
45	Consistent HW/SW on global basis	7	1	Easy To Use	Primary Mission	7									
47	Support for users with disabilities	7	3	Maximize Flexibility	Primary Mission			7							
3	Significant developer support now/future	6	5	Maximize Choice	Asset Management					6					
13	Shared storage both central and distributed	6	3	Maximize Flexibility	Primary Mission			6							
18	Hosting of data access/processing software on "both"	6	3	Maximize Flexibility	Primary Mission			6							
19	Hosting on mobile devices	6	3	Maximize Flexibility	Primary Mission			6							
20	Performance level	6	9	Provide Moderate Performance	Primary Mission									6	
25	Display res, colors, frame rate: moderate to high	6	9	Provide Moderate Performance	Primary Mission									6	
29	Simpler mobile devices	6	1	Easy To Use	Primary Mission	6									
30	Keep service ops in central area	6	1	Easy To Use	Primary Mission	6									
32	Upgradable capabilities on per-user basis	6	4	Minimize Costs	Asset Management				6						
40	Info security in central elements	6	10	Protect Information (Security)	Primary Mission										6
23	Connection type (wired vs. wireless)	5	3	Maximize Flexibility	Primary Mission			5							
26	Shared media (vs. private)	5	4	Minimize Costs	Asset Management				5						
31	Walk-up window for service	5	4	Minimize Costs	Asset Management				5						
36	Low Cost for backup/restore	5	4	Minimize Costs	Asset Management				5						
42	Info security in client devices	5	10	Protect Information (Security)	Primary Mission										5
43	Info security in interfaces	5	10	Protect Information (Security)	Primary Mission										5
46	Multi-language/nation support	5	1	Easy To Use	Primary Mission	5									
28	Complexity allocation	4	2	Easy To Administer/Maintain	Support Mission		4								
41	Info secuirity in commun. Elements	4	10	Protect Information (Security)	Primary Mission										4
					Raw Sum	57	33	95	43	20	14	7	7	12	20
					Instances	8	5	14	7	3	2	1	1	2	4
					Avg.	7.1	6.6	6.8	6.1	6.7	7.0	7.0	7.0	6.0	5.0
					Normalized (1 to 10)	###	7.6	8.5	5.4	7.9	9.5	9.5	9.5	4.8	0.0

Figure 6-2

Category	Customer Need	Customer Importance Weight
Primary Mission	Easy To Use	10
	Maximize Flexibility	8
	Provide Moderate Performance	5
	Protect Information (Security)	1
Support Mission	Easy To Administer/Maintain	8
	High Availability	5
	Maximize Extensibility	8
Asset Management	Minimize Costs	9
	Maximize Choice	9
	Maximize Interoperability	9

Once the Customer Needs were established, implementation neutral Technical Requirements were identified. The Technical Requirements are aspects of the proposed architecture, from a technical perspective rather a customer perspective, each of which addresses one or more of the Customer Needs. The process of developing the Technical Requirements included several iterations. A final list of 29 Technical Requirements was developed.

At this point in the process, the primary inputs for the QFD Requirements Matrix were in hand and the process proceeded to the formal development of the QFD Requirements Matrix.

6.3.1 Customer Requirements Gathering – Tailored Process

It is noteworthy that the filtering, aggregation, and scaling operations implemented were not anticipated during the planning of the Customer Needs (Requirements) gathering process. While some accumulation and averaging was anticipated, the specific process steps employed and the number of iterations were tailored for this implementation. In retrospect, this was a very valuable part of the process of "internalizing" the core Customer Needs.

6.4 QFD Requirements Matrix Development

Placing the 10 Customer Needs, with associated category and Importance Weights on the left side of the QFD Requirements Matrix and the 29 Technical Requirements along the top of the matrix, the next step involved completing the Relationship Matrix entries. The common practice of using three graphical symbols and three strength values (strong=9, moderate=3, weak=1) was employed and the Relationship Matrix was completed after several iterations. The QFD Requirements Matrix is shown in Figure 6-3.

Figure 6-3

QFD Requirements Matrix

Correlations

- ◉ Strong Positive
- ○ Weak Positive
- × Negative

Relationships

- ● Strong - 9
- ○ Moderate - 3
- △ Weak - 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
2																												
3																												
4																												
5																												
6				×																								
7				×																								
8			×	×	×																							
9			×	×	×																							
10			×		×		×																					
11							×			×																		
12							×	×		×																		
13			×		×	×	×	×			×	×																
14							×		×																			
15																												
16					×		×	×	×	×			×															
17																												
18					×		×			×	×	×																
19			×		×						×																	
20											×																	
21					×						×	×																
22					×					×	×	×			×	×			×									
23										×						×		×	×		×	×						
24																												
25			×		×			×		×			×			×		×			×	×		×				
26			×	×	×				×	×	×	×	×					×	×		×		×		×			
27			×		×		×	×	×	×			×		×	×	×	×		×	×	×	×	×		×		
28							×	×		×			×	×		×		×	×		×	×				×	×	
29							×			×			×	×		×		×		×	×					×		

Category	Customer Requirement	Customer Importance Weight	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	Competitors' Capabilities (0 – 5)
Primary Mission	Easy To Use	10	●	●	●	●	●	●	○	○	○						○		○		△			△	○				○		○	
	Maximize Flexibility	8						○	○	○	●	△	△		△									○	○	○		○	○			
	Provide Moderate Performance	5				△						△	●	○			△								△			○				
	Protect Information (Security)	1							△						●	△																
Support Mission	Easy To Administer/Maintain	8	○	○			○	○				○	●	○			●	●	●	○	○	●	●		○				●			
	High Availability	5																○		●	●	●	△					○	△			
	Maximize Extensibility	8						△	△			△	○	○										△	●	△	△	○		△		
Asset Management	Minimize Costs	9	●	○	○		△		○			●	●	△			△	○			○				○	△	●	△	●			
	Maximize Choice	9											●	●			○										●		△	●	●	
	Maximize Interoperability	9					○	●	△	○	○		△	○			○							○	○	○		△		○	●	
		Importance Weight	195	141	117	95	150	227	99	81	129	126	320	180	17	1	170	114	102	69	106	117	77	69	209	68	170	96	221	116	192	
		Relative Weight	5	11	14	22	10	2	20	23	12	13	1	7	28	29	8	17	19	25	18	14	24	25	4	27	8	21	3	16	6	

No.	Technical Requirement
1	Familiar Human-Equipment Interfaces
2	Familiar Metaphors
3	Familiar Applications
4	Consistent Performance
5	Familiar Operating Sys./User Environment
6	Network Access Capability
7	Reconfigurable Storage Location
8	Support Multiple Application "Variants"
9	Support Mobile Clients
10	Capacity Aggregation Capability
11	Off-the-shelf Components
12	Readily Available Technologies
13	Electronic Security Features
14	Physical Security Features
15	Use Common Media
16	Allocate Complexity to Central Equipment
17	Integrated Help Capability
18	Self-Healing Capability
19	High Reliability
20	System Monitor Capability
21	Sophisticated Diagnostics
22	Newest Equipment-Equipment Interfaces
23	Established Equipment-Equipment Interfaces
24	Upgradable Equipment
25	Multiple Suppliers
26	Dynamic Hardware/Software Function Allocation
27	Low Complexity Equipment
28	Dominant Design Status
29	Use Industry Standards

During the completion of the Relationship Matrix, it became clear that a "flat" matrix development tool could be quite insufficient for most QFD efforts due to loss of information. Specifically, the matrix developer's thoughts, both context and intent, are reduced to a single symbolic relationship entry. Upon review, it was sometimes difficult to recall context or intent when considering particular Relationship Matrix entries, and this could easily make development of team consensus difficult. This issue identified the need for tool capable of expanding from the symbolic entry into a textual description of the entry's rationale and context. The development of such a tool is discussed in Chapter 7.

As planned, once the Relationship Matrix entries were completed, the Importance Weight and Relative Weight for each Technical Requirement was calculated and entered.

Finally, the Correlation Matrix (the "roof" on the QFD "house") was completed. Although Figure 4-1 shows the traditional Isosceles triangle-shaped Correlation Matrix, software limitations prevented the entering of relationship symbols in the "tilted" squares cells of that Correlation Matrix format. As an alternative representation of the same information, a modified stepped triangle version was utilized. As planned, the common approach of entering only negative (potentially problematic) correlations was deemed sufficient and was employed and after several iterations, the Correlation Matrix was completed.

6.5 Conceptual Level Trade Study – Tailored Process

The planned Pugh-type comparative Trade Study of Concepts was conducted, but with a focus differing from the plan described in Chapters 4 and 5. It was determined that existing architectures had significantly addressed the solution space defined by the Technical Requirements and as such, the generation of Concepts by choosing which

Technical Requirements to address and how/how much would likely result in only one "best" Concept. However, that Concept might be infeasible due to its inclusion of mutually exclusive or interfering features addressing one or more Technical Requirements (hence, the value of the QFD Correlation Matrix was considered).

As a response to this realization, the Trade Study was modified to be a learning experience followed by a synthesis step. The learning phase consisted of rating each of the key existing architectures (Windows Terminals, JavaStations, and SunRay/HotDesk) against the baseline architecture (Intel-based PC attached via Ethernet networking to Intel-based File and Print Server) using the '+', '=', '-', or '?' symbols for "better than baseline," "equivalent to baseline," "worse than baseline," or "not sure/unrated" (see Appendix C). Working through this rating process, it quickly became clear that the maximum value would be obtained if each rating, cell by cell, were documented with a few words as to the considerations applied. Thus, the comparative rating matrix grew substantially and really became unmanageable when the considerations for each architectural concept were placed next to the symbolic rating of the architecture for each Technical Requirement.

This issue identified the potential need for a three-dimensional matrix development tool where each symbolic rating cell could be selected for expansion into the considerations text. This improved capability would both better document the thoughts of the matrix developer as well as reduce the symbolic ratings matrix to a manageable format. This opportunity is discussed further in Chapter 7.

Upon completion of the comparative rating of the key existing architectures against the baseline architecture using the Technical Requirements as evaluation criteria, a significant amount of insight into what features have worked and where architects are still exploring alternatives was achieved. Using this information, the opportunity areas of the Desktop Computing Architecture space were considered and a new Concept architecture was created as the last column of the comparative matrix. This new Concept was then allocated comparative ratings for each Technical Requirement relative to the baseline architecture. Context and significant feature expectations/projections were documented in the considerations entries for each symbolic rating entry, just as had been done for the existing architectures.

6.6 Summary of a Preferred Conceptual Level Architecture

The comparative rating of existing architectures and the subsequent synthesis of a new Concept were well documented in the previous process step, so no reiterative summary section was deemed necessary.

6.7 QFD Product Matrix Development – Tailored Process

The Product Matrix effort was undertaken as planned but the value of the effort in identifying a new architecture was quickly called into question. Specifically, the listing of Subsystem Characteristics resulted in about 45 items after the first attempt. Most of these items were common implementation features and functions, such as keyboards, Ethernet networking, etc., and as such they were recognized as not being effective at providing the discriminating factors required for the ensuing Implementation Level Trade Study step. Given this realization, the completion of the Relationship Matrix portion of the QFD Product Matrix was deferred until suitable values for the effort could be identified.

To address this lack of available discimination in Architecture Characteristics, another Systems Engineering approach was employed – Morphologic Problem Solving[3].

6.8 Morphologic Problem Solving

To this point, the following Systems Engineering Principles and Methods were employed in this investigation to develop a New Architecture for Corporate Desktop Computing:

1) Literature Search;
2) Capturing the "Voice of the Customer" via a detailed questionnaire;
3) Aggregation of customer input into categories via Affinity Diagramming;
4) Needs Translation via QFD Requirements Matrix;
5) Requirements Translation via QFD Product Matrix – partially completed;
6) Trade Study Techniques using weighted matrices to compare alternatives.

These methods provided substantial structured analysis and learning for the problem space as well as existing solutions, but they did not appear to be offering a path to an Design Level architecture that was new, valuable, and viable. Referring to the SDM Systems Engineering course notes, the concept of reducing the potential architectural solutions to a few key elements, then applying Morphological Problem Solving appeared to offer a potential path through this quandary and thus this method was implemented.

Morphological Problem Solving is effectively the synthesis of all possible combinations of features or factors and the subsequent processing of all of the combinations to try to find one or more solutions in these combinations. The method is most effective in situations where, such as in this case, a designer "can not see the forest for the trees" due to overwhelming number of details being considered. By re-abstracting the problem to a simpler set of higher-level elements, then expanding into all possible combinations, often new insights into the truly important architectural features will be reached. It has further value in simply ensuring that all possible solutions have been considered and that personal or corporate culture biases are not constraining the designers to an available solution.

In this investigation, the architectural elements were re-abstracted to three elements, with each having several potential values as listed below.

Element 1: Client Device
Value A: Intel-based Personal Computer (see Note 1)
Value B: MS-Windows Terminal ("WinTerm")
Value C: JavaStation
Value D: Terminal/Communication Device

Element 2: Interconnection
Value E: Point-to-Point (dedicated) Ethernet
Value F: Shared (bused) Ethernet
Value G: Point-to-Point (dedicated) Non-Ethernet
Value H: Shared (bused) Non-Ethernet

Element 3: Central Equipment
Value I: File and Print Server
Value J: Application Server
Value K: Specialized Communications Device
Value L: Centralized PC (see Note 2)

Note 1: The exclusion of client Personal Computers or Workstations not based upon Intel-x86 architecture processors and the general PC platform chipset, firmware, and peripheral set was not undertaken lightly. However, it was clear from the Voice of the Customer that native compatibility with existing Microsoft Windows/Intel-based Operating Systems and Applications was a key feature of any device designed to provide Desktop (client) computing services to Corporate users. This is a verbose way of saying non-PC computers would not survive further process steps in this investigation and were thus not included here.

Note 2: The "Centralized PC" is an innovation of the thesis author and is key contributor to the New Architecture for Corporate Desktop Computing being developed herein. The Centralized PC is a fundamentally a standard PC whose functions and capabilities have been reallocated to separate end user interface functions from processing and storage functions. The Central PC's chassis is placed in a central area of a corporate facility. End user access to the PC's processing and storage capabilities is achieved through a desktop communication device / terminal. The communication between the Central PC and the desktop device is via a high-speed (100Mbps or higher) communication link. Reallocation of the functions and capabilities of the system provides opportunities for reducing service costs for the system – the primary impetus for this thesis. More details on the function and capability reallocation are provided throughout the remainder of this thesis.

Once the three major elements and their potential values were available, an exhaustive list of all 64 combinations of these three elements and respective values was developed. This list was then reviewed and each combination was labeled as being either a version of an existing architecture, a non-valuable or non-viable combination of features, or a potential candidate for the New Architecture for Desktop Computing. The result of this process was the identification of four combinations as potential candidates for further consideration. All other combinations were deemed to be either versions of existing architectures or non-valuable or non-viable combinations of the three major elements. Figure 6-4 shows the exhaustive list of configurations with attribute information.

Figure 6-4

Morphologic Problem Solving -- Architectural Evaluation Worksheet

	Clients	
A	PC	Desktop or laptop Wintel box
B	WinTerm	"Dumb Terminal" programmed only to interface to Windows NT App Server; New generation have browsers to access world wide web directly.
C	JavaStation	Hard-disk-less computer: can load programs from network devices (servers) for execution locally
Y	Terminal/Comm Device	Essentially hardware-only device: operation is fixed, though possibly upgradable
	Interconnect	
D	P2P Ethernet	Point-to-point Ethernet (10Mb, 100Mb, or 1Gb)
E	Shared Ethernet	Multiple (bused) Ethernet (10Mb, 100Mb, or 1Gb)
F	P2P Non-Ethernet	Any point-to-point comm link other than Ethernet variants
G	Shared Non-Ethernet	Any multiple-device (bused) comm network other than Ethernet variants
	Central Equipment	
H	F/P Server	Server provides network-accessible file storage and printing queues; May serve minor applications (download and run); can provide server portion of C/S e-mail, other apps)
I	App Server	Server runs applications -- send/get user interface information (various levels of abstraction) to/from clients via comm link
J	Special Comm device	Handles interface between Central Equipment (may be Mainframe, Mini, or Server) and client over comm links
K	Central PC	Repackaged Wintel-compatible desktop PC in rack or other form factor for central location convenience

1	A	D	H	Baseline	PC	P2P Eth	F/P Server
2	A	D	I	Baseline + Hydra	PC	P2P Eth	App Server
3	A	D	J	PC as Terminal	PC	P2P Eth	Special Comm
4	A	D	K	Peer-to-Peer (No value?)	PC	P2P Eth	Central PC
5	A	E	H	Baseline	PC	Shared Eth	F/P Server
6	A	E	I	Baseline + Hydra	PC	Shared Eth	App Server
7	A	E	J	PC as Terminal	PC	Shared Eth	Special Comm
8	A	E	K	Peer-to-Peer (No value?)	PC	Shared Eth	Central PC
9	A	F	H	Baseline	PC	P2P Non	F/P Server
10	A	F	I	Baseline + Hydra	PC	P2P Non	App Server
11	A	F	J	PC as Terminal	PC	P2P Non	Special Comm
12	A	F	K	Peer-to-Peer (No value?)	PC	P2P Non	Central PC
13	A	G	H	Baseline	PC	Shared Non	F/P Server
14	A	G	I	Baseline + Hydra	PC	Shared Non	App Server
15	A	G	J	PC as Terminal	PC	Shared Non	Special Comm
16	A	G	K	Peer-to-Peer (No value?)	PC	Shared Non	Central PC
17	B	D	H	"Internet" Terminals	WinTerm	P2P Eth	F/P Server
18	B	D	I	WinTerms	WinTerm	P2P Eth	App Server
19	B	D	J	NA	WinTerm	P2P Eth	Special Comm
20	B	D	K	NA	WinTerm	P2P Eth	Central PC
21	B	E	H	"Internet" Terminals	WinTerm	Shared Eth	F/P Server
22	B	E	I	WinTerms	WinTerm	Shared Eth	App Server
23	B	E	J	NA	WinTerm	Shared Eth	Special Comm
24	B	E	K	NA	WinTerm	Shared Eth	Central PC
25	B	F	H	"Internet" Terminals	WinTerm	P2P Non	F/P Server
26	B	F	I	WinTerms	WinTerm	P2P Non	App Server
27	B	F	J	NA	WinTerm	P2P Non	Special Comm
28	B	F	K	NA	WinTerm	P2P Non	Central PC
29	B	G	H	"Internet" Terminals	WinTerm	Shared Non	F/P Server
30	B	G	I	WinTerms	WinTerm	Shared Non	App Server
31	B	G	J	NA	WinTerm	Shared Non	Special Comm
32	B	G	K	NA	WinTerm	Shared Non	Central PC
33	C	D	H	Network Computer	JavaStation	P2P Eth	F/P Server
34	C	D	I	Network Computer	JavaStation	P2P Eth	App Server
35	C	D	J	NA	JavaStation	P2P Eth	Special Comm
36	C	D	K	NA	JavaStation	P2P Eth	Central PC
37	C	E	H	Network Computer	JavaStation	Shared Eth	F/P Server
38	C	E	I	Network Computer	JavaStation	Shared Eth	App Server
39	C	E	J	NA	JavaStation	Shared Eth	Special Comm
40	C	E	K	NA	JavaStation	Shared Eth	Central PC
41	C	F	H	Network Computer	JavaStation	P2P Non	F/P Server
42	C	F	I	Network Computer	JavaStation	P2P Non	App Server
43	C	F	J	NA	JavaStation	P2P Non	Special Comm
44	C	F	K	NA	JavaStation	P2P Non	Central PC
45	C	G	H	Network Computer	JavaStation	Shared Non	F/P Server
46	C	G	I	Network Computer	JavaStation	Shared Non	App Server
47	C	G	J	NA	JavaStation	Shared Non	Special Comm
48	C	G	K	NA	JavaStation	Shared Non	Central PC
49	Y	D	H	NA	Term/Comm	P2P Eth	F/P Server
50	Y	D	I	Dumb Terminal Arch	Term/Comm	P2P Eth	App Server
51	Y	D	J	SunRay	Term/Comm	P2P Eth	Special Comm
52	Y	D	K	**New Candidate 1**	Term/Comm	P2P Eth	Central PC
53	Y	E	H	NA	Term/Comm	Shared Eth	F/P Server
54	Y	E	I	Dumb Terminal Arch	Term/Comm	Shared Eth	App Server
55	Y	E	J	Dumb Terminal Arch	Term/Comm	Shared Eth	Special Comm
56	Y	E	K	**New Candidate 2**	Term/Comm	Shared Eth	Central PC
57	Y	F	H	NA	Term/Comm	P2P Non	F/P Server
58	Y	F	I	Dumb Terminal Arch	Term/Comm	P2P Non	App Server
59	Y	F	J	Dumb Terminal Arch	Term/Comm	P2P Non	Special Comm
60	Y	F	K	**New Candidate 3**	Term/Comm	P2P Non	Central PC
61	Y	G	H	NA	Term/Comm	Shared Non	F/P Server
62	Y	G	I	Dumb Terminal Arch	Term/Comm	Shared Non	App Server
63	Y	G	J	NA	Term/Comm	Shared Non	Special Comm
64	Y	G	K	**New Candidate 4**	Term/Comm	Shared Non	Central PC

The four combinations identified as potential candidates were:

1) Terminal/Communication Device, Point-to-Point Ethernet, Centralized PC
2) Terminal/Communication Device, Shared Ethernet, Centralized PC
3) Terminal/Communication Device, Point-to-Point Non-Ethernet, Centralized PC
4) Terminal/Communication Device, Shared Non-Ethernet, Centralized PC

These four candidates all share two common "major element" values – the Terminal/Communications Device and the Centralized PC. The variation between the four candidates is found in the Interconnection method selected.

The use of Morphologic Problem Solving thus yielded architectural configurations that the other Systems Engineering methods had not specifically provided and thus this method's value was realized.

With the identification of these candidate architectures, the development of a Conceptual Level New Architecture for Corporate Desktop Computing was completed. The findings, better defined here, are consistent with those identified in Section 6.4.

6.9 Design/Implementation Level Trade Study – Tailored Process

To further develop the New Architecture for Corporate Desktop Computing, the use of Pugh's Divergent – Convergent Development methodology was identified as a valuable method for the following reasons:

1) Interconnection is a well-understood feature, not worthy of further investigation;
2) Terminals are somewhat well-understood, but variants can still be developed;
3) The Centralized PC is a new feature, well suited to concept generation and downselect at the Design Level.

While some advantages and disadvantages exist for each of the four variants of the Interconnection element of the architectural candidates, those variants were not investigated further herein due to the widely understood nature of the subject.

The first divergence – convergence of concepts (Pugh calls the steps "Concept Generation" and "Controlled Convergence") in the process has already occurred via the Morphologic Problem Solving step followed by the selection of the candidate architectures. The Design Level trade study formed the second divergence – convergence instance of the Pugh development methodology.

The Design Level trade study was conducted for the system formed by the Client Device (Terminal/Communication Device) and the Central Equipment (Centralized PC) elements of the candidate conceptual-level architecture while assuming that the Interconnection element would be available and of adequate capability to support any Design Level concept developed.

6.9.1 Design Level Trade Study – Concept Generation

A QFD Product Matrix was chosen as the first Systems Engineering method to attempt to generate Design Level variants of the selected conceptual architecture. This Product Matrix utilized the original system-level QFD Technical Requirements as evaluation criteria. The objective in preparing the Product Matrix was to generate enough implementation choices to allow the synthesis of several viable alternative configurations of features. Once the alternative Design Level "concepts" were identified, the process moved to the Controlled Convergence or actual weighted comparison and selection phase.

In building this Product Matrix, it quickly became apparent that there are some Technical Requirements that would be affected only by the Client Device feature choices, a few Technical Requirements that would be affected by both the Client Device and Central Equipment feature choices, and some that would be affected only by the feature choices for the Central Equipment. The decision was made to focus first only on those Technical Requirements that would be affected by Central Equipment feature choices. Thus, the Product Matrix was developed specifically for the Centralized PC architectural element.

The Product Matrix is shown in Figure 6-5. The first attempt to complete the matrix at the level of specific implementation choices, such as "Rack" and "Shelf" under the "Mounting" Subsystem Characteristic was quickly determined to be little value as the Relationship Matrix entries for each set of specific choices just echoed that of the more general Subsystem Characteristic. As such, the Product Matrix was completed with categories, rather than specific choices, serving as the Critical Subsystem Characteristics entries. Importance Weights and Relative Importance Weights were calculated for the Product Matrix categories in the usual manner, resulting in the completed Product Matrix.

Figure 6-5

QFD Product Matrix for Centralized PC Architectural Element

Relationships

● Strong - 9
○ Moderate - 3
△ Weak - 1

Technical Requirement	Relative Priorities from Rqmts Matrix	Importance Weight from Rqmts Matrix	Subsystem Characteristics	1 Mounting	2 Packaging	3 Hard Disk Configuration	4 Client Direct to LAN	5 Video to Client	6 Admin. Access	7 Removable Storage Configuration
Off-the-shelf Components	1	320		●	●	●	●	●	●	●
Network Access Capability	2	227		△	△		●			
Low Complexity Equipment	3	221		○	△	●	△	●	△	●
Established Equipment-Equipment Interfaces	4	209		●	●	●	○	●	△	●
Familiar Human-Equipment Interfaces	5	195		●	●	○			△	○
Use Industry Standards	6	192		●	○	●	○		○	●
Readily Available Technologies	7	180		△	●	●	○		△	○
Use Common Media	8	170			△	△				△
Multiple Suppliers	8	170		△	●	●	△	●		●
Familiar Operating Sys./User Environment	10	150				△	△		○	○
Familiar Metaphors	11	141								△
Support Mobile Clients	12	129				△	●	●		○
Capacity Aggregation Capability	13	126			●	●		○		○
Familiar Applications	14	117					△	○		△
System Monitor Capability	14	117				△				△
Dominant Design Status	16	116		△	●	●	○			△
Allocate Complexity to Central Equipment	17	114					○	●	●	
High Reliability	18	106			○	○	○		○	
Integrated Help Capability	19	102								
Reconfigurable Storage Location	20	99			△	△				
Dynamic Hardware/Software Function Allocation	21	96								
Consistent Performance	22	95			△	●	○			●
Support Multiple Application "Variants"	23	81				△		○	△	●
Sophisticated Diagnostics	24	77			△	△	△	△	△	●
Self-Healing Capability	25	69			●	●			△	
Newest Equipment-Equipment Interfaces	25	69			●	△		△		●
Upgradable Equipment	27	68		△	●	●	△	●		●
Electronic Security Features	28	17			●	●	△	●	●	●
Physical Security Features	29	1		●	●	●			○	●
				9677	15643	17851	9940	12350	6438	16681
				6	3	1	5	4	7	2

The completed Product Matrix allowed for the expansion of the most critical categories into specific choices to support a Design Level Trade Study. In this case, the following expansions were made:

Mounting
a) Rack
b) Shelf

Packaging
a) Board in Backplane
b) Board in Case

Hard Disk Configuration
a) Dedicated
b) Shared – LAN
c) Shared – non-LAN ("clustered")

Client Direct To LAN
a) Hardware Switch
b) Route Through Central PC via software

Video To Client
a) Pixels
b) Triangles

Administration Access
a) Primary LAN
b) Dedicated [administration] LAN

Removable Storage Configuration
a) Dedicated
b) Shared – LAN
c) Shared – non-LAN
d) Direct – On-Demand

To test these implementation choices relative to the Voice of the Customer, a weighted analysis was completed (see Figure 6-6) using the ten original QFD Customer Needs.

Figure 6-6

Susbystem Characteristics - Evaluation of Alternatives On Individual Basis

Category	Implementation Options	Customer Requirements	Easy To Use	Maximize Flexibility	Provide Moderate Performance	Protect Information (Security)	Easy To Administer/Maintain	High Availability	Maximize Extensibility	Minimize Costs	Maximize Choice	Maximize Interoperability	Computed Score
	Importance Rating		10	8	5	1	8	5	8	9	9	9	
Mounting	Rack		8	8	5	9	9	9	7	3	6	8	504
	Shelf		8	8	5	1	6	9	9	7	9	8	551
Packaging	Board in Backplane		9	6	5	9	8	9	8	8	4	8	525
	Board in Case		6	9	5	1	7	9	7	6	9	8	522
Hard Disk Configuration	Dedicated		9	5	9	9	3	3	7	5	9	7	468
	Shared - LAN		7	8	4	7	5	6	8	5	7	7	466
	Shared - non-LAN		8	7	6	9	5	9	9	7	2	5	458
Client Direct to LAN	Hardware Switch		9	8	9	9	9	9	8	5	5	8	551
	Route Through Centralized PC via Software		8	8	7	5	8	8	8	9	7	8	568
Video to Client	Pixels		5	9	9	1	8	9	8	6	7	3	485
	Triangles		5	3	9	3	9	9	8	6	8	4	465
Administration Access	Primary LAN		5	9	5	3	9	7	9	7	9	9	554
	Dedicated LAN		9	1	9	9	9	9	7	9	6	4	496
Removable Storage Configuration	Dedicated		1	3	9	9	3	8	5	1	7	5	309
	Shared - LAN		9	8	4	1	4	7	7	7	8	8	505
	Shared - non-LAN		1	7	6	5	6	9	6	4	6	5	377
	Direct - on-Demand		3	3	9	9	7	4	9	6	9	1	400

The results of this analysis show that most of the choices, in the isolation of their category, do not address the primary Customer Needs in a significantly better manner than the alternative choices. However, the analysis did also demonstrate that a few specific choices did appear to be superior to the alternatives. The goal of this analysis was to establish a baseline against which the ensuing Trade Study, which attempts to weigh the value of the entire feature set of each candidate configuration, could be compared. This analysis was also completed simply to revitalize the importance of the original Customer Needs since the Trade Study was to be completed using the QFD Technical Requirements, which are one level translations of the Customer Needs.

The final step in this Concept Generation step was the actual generation of a manageable and effective list of configurations that would be evaluated and compared in the Controlled Convergence step to follow. From the list of implementation choice categories (Critical Subsystem Characteristics) and specific choices shown above, the possible number of configurations was calculated as 2 x 2 x 3 x 2 x 2 x 2 x 4 = 384. Given that each configuration would be evaluated for each Technical Requirement, 384 was deemed to be too many.

To reduce the number of configurations to be evaluated, the Product Matrix Importance Weights and Relative Importance Weights for the choice categories were reviewed. The spread between the Importance Weights of the choice categories ranked first, second, and third by the Relative Importance Weights and those choice categories ranked fourth and lower was found to be substantial. This substantial spread indicates that the categories ranked lower were expected to have significantly less impact on a configuration's overall suitability than those categories with higher Relative Importance Weights. Thus, the decision was made to only use the choice categories with the three highest Relative Importance Weights for Concept Generation.

The Design Level Trade Study concept list was generated by forming all combinations of the choices for the choice categories of Packaging, Hard Disk Configuration, and Removable Storage. This formed a total of 24 configurations for the Trade Study.

6.9.2 Design Level Trade Study – Controlled Convergence

The Design Level Trade Study involved assessing the 24 configurations generated in the previous concept generation step using the QFD Technical Requirements as the evaluation criteria. Given the use of only the top three choice categories, a review of those categories' Relationship Matrix entries in the Product Matrix to the Technical Requirements was conducted. This review indicated that the three choice categories only were expected to have a strong or moderate effect on 13 of the 29 Technical Requirements and thus the other 16 Technical Requirements need not be included in the evaluation effort.

To further reduce the complexity of the Trade Study, it was observed that after the Technical Requirement with the Relative Importance Weight (priority) of 13, the next Technical Requirement that was expected to be affected in a strong or moderate way was ranked 23rd of the 29 Technical Requirements. The difference between the Importance Weights of rank 13 and rank 23 are significant (almost a 2:1 ratio). This observation led to the conclusion that no Technical Requirement ranked below 14th should be considered in the Design Level Trade Study.

With these complexity reduction methods completed, the Trade Study matrix was built (see Figure 6-7).

Figure 6-7

Design Level Trade Study Matrix

Configuration Index	Mounting*	Packaging	Hard Disk Configuration	Client Direct to LAN*	Video to Client*	Administration Access*	Removable Storage Configuration	Technical Requirements	Off-the-shelf Components	Network Access Capability	Low Complexity Equipment	Established Equipment-Equipment Interfaces	Familiar Human-Equipment Interfaces	Use Industry Standards	Readily Available Technologies	Use Common Media	Multiple Suppliers	Familiar Operating Sys./User Environment	Familiar Metaphors	Support Mobile Clients	Capacity Aggregation Capability	Familiar Applications	System Monitor Capability	Dominant Design Status	Allocate Complexity to Central Equipment	High Reliability	Integrated Help Capability	Reconfigurable Storage Location	Dynamic Hardware/Software Function Allocati	Consistent Performance	Support Multiple Application "Variants"	Sophisticated Diagnostics	Self-Healing Capability	Newest Equipment-Equipment Interfaces	Upgradable Equipment	Electronic Security Features	Physical Security Features	Raw Score	Ranked Score	Relative %
								Scaled Importance Weight -- Top 13 TRs	10	5	5	4	4	4	3	3	3	2	1	1	1																			
								Relative Priorities from Rqmts Matrix	1	2	3	4	5	6	7	8	8	10	11	12	13	14	14	16	17	18	19	20	21	22	23	24	25	25	27	28	29			
								Importance Weight from Rqmts Matrix	320	227	221	209	195	192	180	170	170	150	141	129	126	117	117	116	114	106	102	99	96	95	81	77	69	69	68	17	1			
1		Board in Backplane	Dedicated				Dedicated		7		7	9		8	9		8	9			6																	248	4	93%
2		Board in Backplane	Dedicated				LAN		6		6	9		7	9		7	9			7																	227	8	85%
3		Board in Backplane	Dedicated				Clustered		5		5	8		7	8		7	8			8																	204	16	76%
4		Board in Backplane	Dedicated				On-Demand		6		7	9		7	9		8	9			8																	236	5	88%
5		Board in Backplane	LAN				Dedicated		6		6	8		6	9		7	8			7																	217	11	81%
6		Board in Backplane	LAN				LAN		6		5	8		5	9		6	8			8																	206	15	77%
7		Board in Backplane	LAN				Clustered		5		4	7		5	8		6	7			8																	182	24	68%
8		Board in Backplane	LAN				On-Demand		6		6	8		5	9		7	8			8																	214	13	80%
9		Board in Backplane	Clustered				Dedicated		5		5	8		7	8		6	8			8																	201	18	75%
10		Board in Backplane	Clustered				LAN		5		4	7		6	8		5	8			8																	185	23	69%
11		Board in Backplane	Clustered				Clustered		5		4	8		7	7		5	7			9																	189	21	71%
12		Board in Backplane	Clustered				On-Demand		5		5	8		6	8		6	8			9																	198	19	74%
13		Board in Case	Dedicated				Dedicated		8		8	9		9	9		9	9			3																	267	1	100%
14		Board in Case	Dedicated				LAN		8		7	9		8	9		7	9			4																	253	3	95%
15		Board in Case	Dedicated				Clustered		6		6	8		7	8		7	8			5																	216	12	81%
16		Board in Case	Dedicated				On-Demand		8		8	9		8	9		9	9			5																	265	2	99%
17		Board in Case	LAN				Dedicated		7		7	8		7	9		7	8			4																	233	6	87%
18		Board in Case	LAN				LAN		7		6	8		6	9		6	8			5																	222	9	83%
19		Board in Case	LAN				Clustered		6		5	7		6	8		6	7			5																	198	19	74%
20		Board in Case	LAN				On-Demand		7		7	8		6	9		7	8			5																	230	7	86%
21		Board in Case	Clustered				Dedicated		6		6	8		8	8		7	8			5																	220	10	82%
22		Board in Case	Clustered				LAN		6		5	7		7	8		6	8			5																	204	16	76%
23		Board in Case	Clustered				Clustered		5		4	8		7	7		6	7			6																	189	21	71%
24		Board in Case	Clustered				On-Demand		6		6	8		7	8		6	8			6																	214	13	80%

To build the Trade Study matrix, the Technical Requirements weighting was first normalized. This was accomplished by forcing the Importance Weights (from the QFD Requirements Matrix) into a one to ten range shown as the "Scaled Importance Weight" row at the top of the matrix. Next, the Raw Score, Ranked Score, and Relative % columns were added at the right. The configurations were entered, as were the Technical Requirements, then each configuration was evaluated as to its ability to satisfy each Technical Requirement, on a one to nine scale. Finally, each cell's rating value was multiplied by the "Scaled Importance Weight" of the column and the row's total was calculated in the Raw Score column.

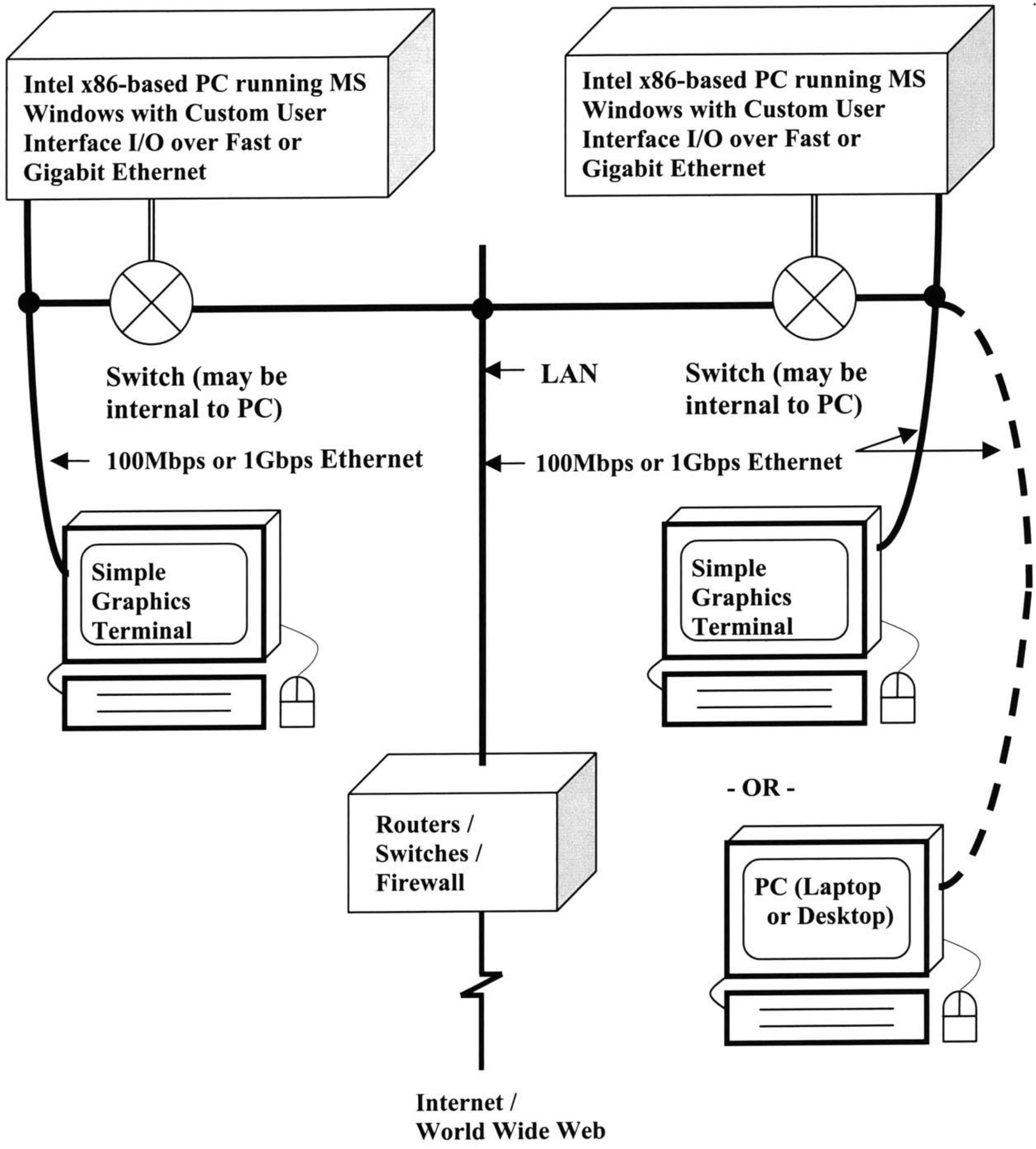
The Ranked Score column of the Design Level Trade Study matrix indicates each configuration's preference rating, with one being the most preferred, relative to the other configurations. The Relative % column provides an indication of each configuration's Raw Score relative to that of the most preferred configuration.

6.10 Summary of a Preferred Design/Implementation Level Architecture

From the Design Level Trade Study matrix, the most preferred Design Level configuration for the Centralized PC element of the candidate architecture is characterized by Packaging in a standard desktop PC case, Hard Disk Configuration using a hard disk internal to the standard case, and a Removable Storage Configuration of removable storage devices internal to the standard case. Thus, the preferred Design Level architecture is a standard desktop PC form in which the user interface is provided by a terminal device on the user's desk, the Interconnect is a high-speed network, and the new "Centralized PC" is placed in a centralized computer room for ease of administration. This architecture is depicted in Figure 6-8.

Figure 6-8

Preferred Design Level Architecture



6.10.1 Sensitivity Analysis

From the Design Level Trade Study matrix, it can be seen that while 13 (reference is to configuration index, column 1 of the matrix) achieved the highest score. A few other configurations were found to be very close in value to the most preferred configuration.

For configurations 14 and 16, which scored within 5% of configuration 13, the only difference in configuration is the Removable Media Configuration choice category value. This suggests that the Removable Media Configuration choice could be varied among these three specific values without substantially impacting the alignment of the architecture to the Customer Needs. To confirm this relative equality of choice values, the weighted analysis of choice options versus Customer Needs (see Figure 6-6) was consulted. This consultation did not result in a confirmation of the relative equality of the three choices, but rather indicated that a Removable Storage Configuration choice of Shared – LAN was strongly preferred over the other choices, when evaluated in isolation. However, when the Design Level Trade Study was conducted and the implementation choices were evaluated as a completed configuration, other factors of the configurations reduced the differentiating value of the Removable Storage Configuration category choices.

Configuration 1 ranked 4th in the Design Level Trade Study scoring, achieving 93% of the effectiveness of the preferred candidate. The difference between these configurations is the Packaging, with the preferred configuration including a standard desktop PC case and the other including a "Board in Backplane," more like a utility's backroom switching system with interchangeable boards and flexible signal routing. Referring to the weighted analysis of choice options versus Customer Needs (see Figure 6-6), in this case the evaluation of the implementation options in isolation versus the more system-level context of the Trade Study resulted in similar results, namely that the Packaging option was not a significant differentiator between otherwise similar configurations of the Centralized PC element.

It is worth noting that had the factors of central facility space and wiring complexity been identified as significant customer concerns, the Board in Backplane option would most likely have been preferred over the Board in Case option. However, none of the surveyed customers identified these factors as significantly important. Perhaps a larger and more varied survey audience would result in slightly different customer needs and/or importance ratings. Of additional interest, if accurate cost numbers could be generated, Board in Backplane systems would likely be found to be substantially less expensive to acquire and service than systems based on Board in Case packaging.

6.10.2 Client Device Element Issues

The Design Level concept generation and analysis steps were completed with a primary focus on the Central PC version of the Centralized Equipment architectural element. As such, assumptions regarding the features and capabilities of the other architectural elements were necessarily made. These assumptions included Interconnection with adequate capability and performance as well as Client Devices of the Terminal/Communication Device type with appropriate features and performance. The selection of the preferred architectural form for the Central PC Design Level choice subscribed the Client Device to certain features that are discussed briefly below.

The Client Device was selected as a Terminal/Communications Device type. This implies a minimum of machine intelligence, little or no in-client persistent storage, and limited expandability. This baseline view of a client device would work well with the developed Centralized PC form. However, a couple of significant feature/function allocation issues need to be explored in this Client vs. Central allocation exercise.

One feature that needs to be allocated is Removable Media. There are at least three "users" of removable media in a corporate desktop computing environment: 1) End-users, 2) Installers, and 3) Administrators.

In the Design Level Trade Study, four choices for this Removable Media Configuration were considered: Dedicated, Shared via LAN, Shared via Cluster, or Direct connection only On-demand. Each of these options has implications for each of the three user groups, as shown in Figure 6-9.

Figure 6-9: Central PC Removable Media Options and Effects

Configuration	User Group	Advantages	Disadvantages
Dedicated	End-Users	+ Known location + Supported natively by OS and applications + Highest performance	- User must request action by administrator (Central PC is in controlled access area) - May prevent access to latest technology
	Installers	+ Available at exact installation site + Supported natively by OS and applications	- Many physical drives – chance for problems increased
	Administrators	+ Known location + Minimum configuration concerns	- May require visit(s) to Central PC to handle media - Many physical drives – higher number of failures
Shared via LAN	End-Users	+ May allow access to more capable devices + Relocatable	- Devices may already in use when needed - Devices may be missing when needed
	Installers	+ May reuse same device/media with multiple installation instances	- Devices may be far from exact installation site - Devices may be missing or in use when needed
	Administrators	+ May substitute "known good" devices easily during troubleshooting + May substitute local device for remote device during troubleshooting	- Devices may be missing or in use when needed - Access control is difficult/burdensome - Device access may not be natively supported by OS and/or applications
Shared via Cluster	End-Users	+ High-performance + May allow access to more capable devices	- User must request action by administrator - Devices may be missing or in use when needed - Device access may not be natively supported by OS and/or applications
	Installers	+ Known location + May quickly substitute known good device	- Devices may be far from exact installation site - Devices may be missing or in use when needed - Device access may not be natively supported by OS and/or applications
	Administrators	+ Easy to reconfigure virtual attachments via software	- Device access may not be natively supported by OS and/or applications - Devices may be missing or in use when needed
Direct only On-Demand	End-Users	+ Reduces unauthorized access to Central PC + May enable access to latest technology	- User must request action by administrator - Devices may be missing or in use when needed
	Installers	+ Can carry installation "kit" with them (eliminates searching)	- "One more" physical device to fail - Risk of incompatible drives (install vs. use)
	Administrators	+ Controls access to Central PC	- Have to visit Central PC, possibly many times

Some corporate IT staffs view all software installation as their responsibility and do not provide for end-user access to any removable media (although users often find ways around this policy, such as network-mounting of computers or peripherals with the required capabilities). As such, these corporations would favor no removable storage capability at the client desktop and the least cost (including acquisition cost, service cost, and administration time cost) configuration in the Central Equipment element of the architecture. These corporations also do not usually require high-performance removable storage, making the option of LAN-based peripherals for Installers and Administrators and a viable option for them.

Other corporations still allow end-users to directly access information and install programs provided on CD-ROM, floppy diskette, and other media. These corporations would likely require that client devices have direct support for removable media devices, including high-performance video and audio delivery capabilities. These corporations would find none of the four choices already considered for the Removable Media Configuration to be acceptable for supporting end-user needs.

For the "Installer" group of users, the Removable Media Configuration is a minor issue, given the proximity of all of the equipment in the new Central PC closet as defined in the new architecture. Installers might actually prefer a single, networked instance of a removable device such that installation-supporting media would only have to be physically loaded once, then simply referenced via network mounting commands as each new Central PC was configured during the installation process. However, it is likely that installers would be able to work with any of the four choices for Removable Media Configuration with minor cost and time implications.

The third user group to be considered is that of the [system] Administrators. The role of these administrators in most corporations is to install programs, monitor user license compliance, publish software upgrades to client systems, and provide technical assistance when technical difficulties occur. Service and Help Desk personnel should also be considered under the umbrella of this rather broad description of administrator since each might need to access the user's system at some time during the life cycle. With the allocation of complexity to the Central Equipment element of the architecture afforded by this new architecture, administrators will rarely need to actually visit a user's desktop. As such, administrators might favor a LAN-based Removable Media Configuration so that they may configure a peripheral on-demand. However, if performance requirements cannot be met via LAN–attachment, then a direct (inside of the Central PC's case or directly attached to it) attachment method might be preferred.

As can be observed from the above discussion, end-users and other users might view the need for Removable Storage Configuration much differently. This is the type of duality of requirements that drives system complexity. As a result, it is likely that the only architecture that can support all of the varied user groups' needs is the allowance for removable media at both the client device as well as in the Central Equipment element (in one or more configurations).

Another feature that requires allocation is extensibility. Closely related to upgradability and expandability, extensibility is a measure of the system's ability to support additions, sometimes unpredictable in form at the time of the system's architectural development, that enable new uses of the system. In the case of the Client Device element of this corporate desktop computing architecture, typical drivers for extensibility might include new types of equipment interfaces, such as IEEE-1394 serial bus connections, video capture devices, etc. The allocation of this extensibility throughout the architecture is always a significant consideration as the provisions or hooks for extensibility are often costly and have no immediate usage purpose at system launch and thus they are almost always closely scrutinized by current system development management.

In the case of the architecture under study, the consideration of extensibility in the client device is quite interesting. The architecture has been developed to dramatically reduce the complexity (and associated management and maintenance costs) in the client device. As such, the in-service lifetime for the client devices would be expected to be significantly extended (further reducing costs). As the in-service lifetime is extended, the technological capabilities of the device will be further behind the mainstream level than a more frequently replaced client device. This "relative degradation" is a classic example of a need which is often best supported by including extensibility support into a device so that the device's capabilities may be "modernized" to extend the working life of the base device and minimize both lifetime costs and mission risks.

However, with computing platforms extensibility sometimes comes at a very high price. For instance, one of the best ways to allow for the support of future equipment-to-equipment interfaces is the inclusion of a high-speed, parallel bus, such as the PCI bus. Inclusion of such an expansion bus in the client device would allow for the future installation of PCI adapter cards, which are readily available, to allow the client device to access new types of storage, serial interfaces, video sources, or as-yet undefined devices. But the inclusion of PCI bus capability violates the architecture's goal of eliminating client device variation (which simplifies service issues) as well as typically requiring more expensive power supply circuitry, a larger and more complex case design, and possibly a more expensive computing capability. Thus, the answer is not obvious and may be highly situation dependent.

As an example, Sun's SunRay client device has chosen to forgo any parallel bus expandability by including only two relatively slow (1MByte/sec) USB ports to support extensibility at the client device. While simplifying administration and providing limited extensibility, the low performance of the current USB interface does not provide for significant asset-protection, nor for the attachment of even moderate-speed peripherals at their rated speed.

The allocation of the Removable Media Configuration and Extensibility are two of the most significant issues in the allocation of function between the Client Device and the Central Equipment elements of the new architecture developed herein. The allocation of many of the other features also requires a Systems Engineering view of the system in order to develop a highly useable system that effectively addresses all of the primary Customer Needs.

6.11 Reflections on Methodology

This experience, and in particular the process tailoring instances, demonstrate the value of planning and using Systems Engineering methods in the real world – the methods are not so rigid (as are many discipline-specific engineering methods) that they preclude changes in the design process "on the fly" but yet they provide enough of a directional framework to prevent the "paralysis by analysis" syndrome.

Chapter 7: Recommendations for Further Investigation

7.1 Simulation

The Interconnection element of the architecture was assumed to be of sufficient capability and performance to support any architecture defined by the other elements. This simplification may not adequately portray reality, particularly when cost is given heavy weighting in system development considerations. Of particular concern are the bandwidth requirements for the Interconnection element.

To support pixel-based video transmission from the Central Equipment element to the Client Device element can be a staggering feat of electrical engineering. For instance, to support a display of 24 frames per second (motion picture quality frame rate), with a color depth of 24-bits (16M colors, a.k.a., "true color") with a resolution of 1600x1200 yields an uncompressed bit rate of 24 x 24 x 1600 x 1200 = 1,105,920,000 bits/second! Note that this resolution of 1600 x 1200 is not exactly an "extensibility"-supporting capability value, but might work for a few more years. More likely, something on the order of 2000x1333 at 24-bits color depth and 30 frames per second (1.9Gbits/second) will be required for architectures that will support a 10-year working life.

The above analysis indicates that pixel-based transmission of the display information requires multiple channels of gigabit Ethernet, some compression techniques, or both. The lure of pixel-based transmission is its simplicity – it allocates most of the complexity to the Central Equipment element of the architecture.

An alternative to pixel-based transmission is triangle-based transmission. This is the method employed by Sun in the Interconnection supporting their SunRay client devices. The use of triangles reduces the amount of information to be transmitted, possibly by a substantial factor, thus potentially allowing the use of lower cost Interconnection technology. Reduction in bandwidth requirements might also serve to support longer Interconnect distances. In Sun's case, the Interconnection is specified as Fast (100Mbit/s) Ethernet, but performance when supporting the display specifications used in the analysis for pixel-based transmission may not be predictable. A potential key disadvantage for the use of triangle-based protocols (or similar construct-based protocols) for the display information transmission is that the introduction of graphics constructs at any level higher than pixels may cause a significant increase in the complexity of both the Central Equipment and the Client Device. Selection of a construct-based protocol may also lock-in an architectural factor that would otherwise be supplanted by a newer construct in a few years whereas using a pixel-based protocol is about as basic and timeless as possible.

Simulation of the selected information transmission method and Interconnection technology would be a prudent step before the construction of an actual working model of the architecture.

7.2 Prototyping

Unless radically new technology is employed in one or more elements of the architecture, prototyping would need to be performed only at the system level to ensure that all the benefits and drawbacks of the architecture were properly considered during the analysis and design process. One aspect of the system that would benefit from prototyping is that of administration and serviceability.

To evaluate administration and serviceability, a prototype of several different configurations for the Central PC element might be effective. These prototypes would portray various characteristic configurations such as Mounting (Rack vs. Shelf), Packaging (Board in Backplane vs. Board in Standard PC Case), and Storage configurations. The prototypes should be subjected to actual usage by administrators and service personnel to determine if the proper balance of acquisition costs versus administration and service costs were achieved before the forms of the architectural elements are finalized.

7.3 Actual Cost Study

The thesis work began with good intentions of building a cost-based case for the new architecture. However, after significant effort and the review of publicly available cost information, it became apparent that the available costs values, for service and acquisition costs, varied widely (3 to 1 or more) for the same types of operations, depending upon the source of the information. This is typical in the Information Technology industry as corporations vary substantially in their abilities to control non-production costs such as those of computing infrastructure. As a result of this wide disparity in the available information, it was decided that this thesis would not include a cost study activity.

If reliable information could be obtained and disclosed, it might be quite instructional to build an actual cost model of the construction, acquisition, operation, and retirement phases of the architecture's lifecycle.

7.4 Tool Development

As noted throughout this document, several shortcomings in the toolset were identified during this investigation. Each is described below with the hope that a future investigation may find useful the development of these capabilities into an existing tool (if such capabilities do not already exist) or the development of new tools.

7.4.1 *Relationship Matrix Text Enhancement*

During the completion of the relationship matrix, it became clear that a "flat" matrix development tool could be quite insufficient for most QFD efforts due to loss of information. Specifically, the matrix developer's thoughts, both context and intent, are reduced to a single symbolic relationship entry. Upon later review, it was sometimes difficult to recall context or intent when considering particular relationship matrix entries, and this could easily make the development of team consensus difficult. This issue identified the need for tool capable of expanding from the symbolic entry into a textual description of the entry's rationale and context.

7.4.2 *Comparative Matrix Text Enhancement*

During the Conceptual Level Trade Study effort, the need to document the considerations for each architectural concept relative to each Technical Requirement became evident. Upon completion of this documentation step, the resulting matrix was very large in size (not number of elements) and was unwieldy to work with. This issue identified the potential need for a three-dimensional matrix development tool where each symbolic rating cell could be selected for expansion into the considerations text. This improved capability would both better document the thoughts of the matrix developer as well as reduce the symbolic ratings matrix to a manageable format.

7.4.3 *General Comments*

Throughout the analysis process, the need to document the thoughts of the process participants became apparent. The effects of time, context change, and team membership changes call out the need to document what people are thinking as they complete each item in each process step. The understanding of simple phrases such as a QFD Technical Requirement is a simple example – the team or individual may think they know what is meant by a phrase when it is created only to encounter disagreement discussing the item only a few minutes, hours, or days later.

This need for documenting common understanding is important, but it must be balanced with time and information manageability realities. Thus, the development of tools to facilitate the information capture, access, management, and archival processes is a real opportunity.

References

[1]Hauser, J.R., and Clausing, D.; "The House of Quality," *Harvard Business Review*, May-June 1988.

[2]Pugh, S.: "Total Design: Integrated Methods for Successful Product Engineering," *Addison-Wesley, NY*, 1991.

[3]Zwicky, F.; "The Morphological Method of Analysis and Construction," Courant, Anniversary Volume, pp. 461-470. *Interscience Publishers*, NY, 1948.

Appendix A – Acronyms and Abbreviations

CD-ROM	Compact Disk – Read Only Memory (removable optical storage)
CRT	Cathode Ray Tube
F/P	File and Print
IBM	International Business Machines Corporation
ICA	Independent Computing Architecture
IEEE-1394	A High Speed (>100Mbit/sec) Serial bus
I/O	Input and Output
IT	Information Technology (same as Information Systems)
LAN	Local Area Network
LCD	Liquid Crystal Display
MS	Microsoft Corporation
NC	Network Computer
OS	Operating System
P2P	Point to Point
PC	Personal Computer
PCI	Peripheral Component Interconnect (a parallel bus)
QFD	Quality Function Deployment
RISC	Reduced Instruction Set Computing
TCO	Total Cost of Ownership
USB	Universal Serial Bus
Windows NT	Microsoft Corporation's Operating System for Corporate Usage (NT originally represented "New Technology", but is now just a label)
x86	Intel 8086 instruction architecture-based processors

Appendix B – Customer Survey

Survey

on

Corporate Desktop Computing Architecture

by

Mark Cummins

Graduate Student in the
MIT System Design and Management Program

Introduction

This document is a survey intended to capture "needs" and "wants" for Corporate Desktop Computing Architectures. The responses to this survey will be used as input to a Quality Function Deployment (QFD)-driven System Engineering effort for a graduate thesis by Mark Cummins which will (hopefully) lead to a MS degree from the Massachusetts Institute of Technology.

Information Processing

The information provided by the survey respondent will be aggregated with that of the other respondents into a unified statement of needs and wants. Individual responses will not be published and respondents will remain anonymous The unified statement will include frequency of response and importance ratings for each need and want and is the key input for the QFD process mentioned earlier.

Your timely and complete response to this survey is sincerely appreciated.

Acronyms
The acronym ACDC represents "Architecture for Corporate Desktop Computing."
The acronym NACDC represents "New Architecture for Corporate Desktop Computing."

Rating
Please indicate your rating by marking the appropriate box for each statement using the following scale:

0 = Totally unacceptable
1 = Would not expect it, but not totally unacceptable
2 = Highly Undesirable
3 = Substantially undesirable
4 = Somewhat undesirable
5 = Neutral (not a positive or negative factor in choosing an ACDC)
6 = Somewhat desirable
7 = Substantially desirable
8 = Highly desirable
9 = Expected, but not quite Mandatory
10 = Mandatory

Important Note on Context:

Please complete this survey from the perspective of a corporation that already has desktop computing infrastructure in place. Thus, attributes such as "compatibility with existing infrastructure" should be factored into the direct costs attribute rating rather than being considered separately.

This section addresses the needs and wants for software availability for any ACDC.

Software: current and future availability

Ref	*Statement*	Rating
1	The ACDC will provide access to only "native" (unmodified) versions of all Microsoft Windows-compatible Client applications.	
2	The ACDC will provide access to only "non-native" (modified for execution on Servers) versions of Microsoft Windows-compatible Client applications	
3	The ACDC will provide access to both native and non-native versions of Microsoft Windows-compatible Client applications	
4	The ACDC will provide no access to native versions of Microsoft Windows-compatible Client applications.	
5	The ACDC will provide no access to non-native versions of Microsoft Windows-compatible Client applications.	
6	The ACDC will provide no access to both native and non-native versions of Microsoft Windows-compatible Client applications.	
7	The ACDC will provide access to only native" versions of "other" (non-MS Windows-compatible) applications (ex: applications running on MAC OS, IBM OS/2, Sun Solaris, or other UNIX variants, MS Windows CE).	
8	The ACDC will provide access to only non-native versions of "other" applications.	
9	The ACDC will provide access to both native and non-native versions of "other" applications.	
10	The ACDC will provide no access to native versions of "other" applications.	
11	The ACDC will provide no access to non-native versions of "other" applications.	
12	The ACDC will provide no access to both native and non-native versions of "other" applications.	
13	The ACDC is currently "well supported" by the commercial software development community.	
14	The ACDC is likely to be "well supported" by the commercial software development community.	
15	The ACDC is likely to be only "marginally" supported by the commercial software development community.	

Software: expertise required for installation, maintenance and administration.

Ref	*Statement*	Rating
16	The ACDC requires a trained expert to install and maintain applications.	
17	The ACDC does not require a trained expert to install and maintain applications.	

Software: deployment capabilities

(Contextual note: Most architectures support manual delivery and installation of software. The following statements are probing the wants and needs relative to other software delivery and installation attributes of the architecture)

Ref	*Statement*	Rating
18	The ACDC supports software delivery/availability mechanisms to allow each user to access "core" (high-user-count and/or frequently accessed) applications.	
19	The ACDC supports software delivery/availability mechanisms to allow each user to access "non-core" (low-user-count or infrequently accessed) applications.	
20	The ACDC does not support software delivery/availability mechanisms to allow each user to access "core" applications.	
21	The ACDC does not support software delivery/availability mechanisms to allow each user to access "non-core" applications.	
22	The ACDC does not support software delivery/availability mechanisms to allow each user to access either "core" or "non-core" applications.	
23	The ACDC supports the rapid deployment of new applications or versions to many/all users.	
24	The ACDC does not support the rapid deployment of new applications or versions to many/all users.	

This section addresses the needs and wants for hardware availability for any ACDC.

Ref	*Statement*	Rating
1	The ACDC utilizes hardware that is available from multiple manufacturers.	
2	The ACDC utilizes hardware that is available from only one manufacturer.	
3	The ACDC utilizes hardware that is a "dominant" design in the market.	
4	The ACDC utilizes hardware that is likely to remain a dominant design in the market.	
5	The ACDC utilizes hardware that is likely to become a dominant design in the market.	
6	The ACDC utilizes hardware that is unlikely to become a dominant design in the market.	

This section addresses the needs and wants for the direct costs for any ACDC.

Ref	*Statement*	Rating
1	The acquisition (including installation) cost for the ACDC is significantly higher than that of alternative ACDCs.	
2	The acquisition (including installation) cost for the ACDC is somewhat higher than that of alternative ACDCs.	
3	The acquisition (including installation) cost for the ACDC is approximately equal to that of alternative ACDCs.	
4	The acquisition (including installation) cost for the ACDC is somewhat lower than that of alternative ACDCs.	
5	The acquisition (including installation) cost for the ACDC is significantly lower than that of alternative ACDCs.	
6	The ACDC makes data access easy, fast, and inexpensive for typical users.	
7	The ACDC makes data sharing easy, fast, and inexpensive for typical users.	
8	The ACDC does not make data access either easy, fast, or inexpensive for typical users.	
9	The ACDC does not make data sharing either easy, fast, or inexpensive for typical users.	
10	The ACDC makes data creation and processing easy, fast, and inexpensive for typical users.	
11	The ACDC does not make data creation and processing either easy, fast, or inexpensive for typical users.	

This section addresses the needs and wants for the physical location of hardware for any ACDC.

Ref	*Statement*	Rating
1	The ACDC supports shared storage only in a centralized location within a facility.	
2	The ACDC supports shared storage only in distributed locations (desktop clients) within a facility.	
3	The ACDC supports shared storage either in a centralized location or in distributed locations (or both) within a facility.	
4	The ACDC supports individual user storage only in a centralized location within a facility.	
5	The ACDC supports individual user storage only in distributed locations (desktop clients) within a facility.	
6	The ACDC supports individual user storage either in a centralized location or in distributed locations (or both) within a facility.	
7	The ACDC supports individual user storage in a mobile client computing device.	
8	The ACDC does not support individual user storage in a mobile client computing device.	
9	The ACDC supports locating the hardware that executes data entry and display software only in a centralized location within a facility.	
10	The ACDC supports locating the hardware that executes data entry and display software only in distributed locations (desktop clients) within a facility.	
11	The ACDC supports locating the hardware that executes data entry and display software either in a centralized location or in distributed locations (or both) within a facility.	
12	The ACDC supports locating the hardware that executes data entry and display software in mobile client computing devices.	
13	The ACDC does not support locating the hardware that executes data entry and display software in mobile client computing devices.	
14	The ACDC supports locating the hardware that executes data access and processing software only in a centralized location within a facility.	
15	The ACDC supports locating the hardware that executes data access and processing software only in distributed locations (desktop clients) within a facility.	
16	The ACDC supports locating the hardware that executes data access and processing software either in a centralized location or in distributed locations (or both) within a facility.	
17	The ACDC supports locating the hardware that executes data access and processing software in mobile client computing devices.	
18	The ACDC does not support locating the hardware that executes data access and processing software in mobile client computing devices.	

This section addresses the needs and wants for the performance attribute for any ACDC.

Ref	*Statement*	Rating
1	The ACDC supports application execution performance that is faster than that of any other currently available ACDC.	
2	The ACDC supports application execution performance that is approximately equal to that of the best currently available ACDC.	
3	The ACDC supports application execution performance that is approximately equal to that of the average currently available ACDC.	
4	The ACDC supports application execution performance that is somewhat slower than that of the average currently available ACDC.	
5	The ACDC supports application execution performance that is very much slower than that of the average currently available ACDC.	

This section addresses the needs and wants for the reliability, monitoring and diagnostics attribute for any ACDC.

Ref	*Statement*	Rating
1	The ACDC provides system reliability (user up-time) better than that of the best currently available ACDC.	
2	The ACDC provides system reliability (user up-time) that is approximately equal to that of the best currently available ACDC.	
3	The ACDC provides system reliability (user up-time) that is approximately equal to that of the average currently available ACDC.	
4	The ACDC provides system reliability (user up-time) that is somewhat lower than that of the average currently available ACDC.	
5	The ACDC provides system reliability (user up-time) that is very much lower than that of the average currently available ACDC.	
6	The ACDC provides system monitoring and diagnostic features that are best-in-class.	
7	The ACDC provides system monitoring and diagnostic features that are approximately equal to those of the best currently available ACDC.	
8	The ACDC provides system monitoring and diagnostic features that are approximately equal to those of the average currently available ACDC.	
9	The ACDC provides system monitoring and diagnostic features that are somewhat less useful than those of the average currently available ACDC.	
10	The ACDC provides system monitoring and diagnostic features that are significantly less useful than those of the average currently available ACDC.	

This section addresses the needs and wants for the electronic communications portion for any ACDC.

Ref	*Statement*	Rating
1	The ACDC supports client/network connections that utilize wire/cable.	
2	The ACDC supports client/network connections which are wireless.	
3	The ACDC does not support client/network connections that utilize wire/cable.	
4	The ACDC does not support client/network connections which are wireless.	
5	The ACDC supports client/network connections over "industry standard" wiring types.	
6	The ACDC supports client/network connections over "industry standard" wireless links.	
7	The ACDC can be deployed in many existing facilities without significant rewiring of the facility.	
8	The ACDC can not be deployed in many existing facilities without significant rewiring of the facility.	
9	The ACDC supports client/network connections over "industry standard" distances.	
10	The ACDC does not support client/network connections over "industry standard" distances.	
11	The ACDC supports client information display at high resolution, full color, and high frame rates. (ex: 2000x1500, 24bit color, 30 frames/second yields uncompressed bit stream rate of 2.2Gbits/sec).	
12	The ACDC supports client information display at moderate resolution, moderate color, and moderate frame rates. (ex: 1600x1200, 16bit color, 5 frames/second yields uncompressed bit stream rate of 154Mbits/sec).	
13	The ACDC supports client information display at moderate resolution, moderate color, and very low frame rates.	
14	The ACDC support client/network connections using interoperable protocols.	
15	The ACDC supports client/network connections using "private," non-interoperable protocols.	
16	The ACDC supports client/network connections on "shared" media.	
17	The ACDC supports client/network connections only on "dedicated" media.	

This section addresses the needs and wants for the user interface portion for any ACDC.

Ref	***Statement***	**Rating**
1	The ACDC supports text-only user interface technologies.	
2	The ACDC supports only graphics-based user interface technologies.	
3	The ACDC supports both text-only and graphics-based user interface technologies.	

This section addresses the needs and wants for the distribution of complexity for any ACDC.

Ref	***Statement***	**Rating**
1	The ACDC contains complexity only within centralized equipment within the facility.	
2	The ACDC contains complexity only within distributed (client) equipment within the facility.	
3	The ACDC contains complexity in both centralized equipment and distributed equipment within the facility.	
4	The ACDC contains complexity in "connectable" equipment (mobile client computing devices).	

This section addresses the needs and wants for the serviceability for any ACDC.

Ref	*Statement*	Rating
1	Most ACDC service operations will be accomplished within a centralized area within the facility.	
2	Most ACDC service operations will be accomplished at the distributed locations within the facility.	
3	The ACDC service operations will be approximately evenly split between the centralized and distributed locations within the facility.	
4	It is not clear where the service operations will occur most frequently.	
5	Service operations on distributed equipment can be effectively supported via a centralized "depot" ("customer walk-up window") operation within a facility.	

This section addresses the needs and wants for the upgradability for any ACDC.

Ref	*Statement*	Rating
1	The ACDC supports the upgrading of some capabilities on a per-user basis.	
2	The ACDC supports the upgrading of many capabilities on a per-user basis.	
3	The ACDC supports the upgrading of most capabilities on a per-user basis.	
4	The ACDC supports the upgrading of all capabilities on a per-user basis.	
5	The ACDC does not support the upgrading of most capabilities on a per-user basis. (entire system, centralized and distributed components, must be upgraded together)	
6	Upgrading capabilities per-user can be performed quickly and cost effectively.	
7	Upgrading capabilities per-user is either slow, costly, or both.	

This section addresses the needs and wants for information backup and restoration for any ACDC.

Ref	*Statement*	Rating
1	The ACDC supports automatic (from user perspective) user data backup.	
2	The ACDC supports automatic client configuration (programs, settings, etc.) backup.	
3	The ACDC supports rapid, non-"personality driven" user data restoration.	
4	The ACDC supports rapid, non-"personality driven" client configuration restoration onto an identical client device.	
5	The ACDC supports rapid, non-"personality driven" client configuration restoration onto a similar, but not necessarily identical, client device.	
6	The ACDC does not support automatic user data backup.	
7	The ACDC does not support automatic client configuration backup.	
8	The ACDC does not support rapid user data restoration.	
9	The ACDC does not support rapid client configuration restoration onto an identical client device.	
10	The ACDC does not support rapid client configuration restoration onto a similar, but not necessarily identical, client device.	
11	The ACDC supports user-performed backup and/or restoration operations.	
12	The ACDC does not support user-performed backup and/or restoration operations.	
13	Backup operations are expensive (some factors -- direct costs: hardware/software, personnel time; indirect costs: user downtime, managing backup datasets).	
14	Backup operations are inexpensive.	
15	Restoration operations are expensive (directly and indirectly).	
16	Restoration operations are inexpensive.	

This section addresses the needs and wants for extensibility for any ACDC.

Ref	*Statement*	Rating
1	The ACDC supports the installation and use of high-performance removable storage at the client. (ex: disk drives)	
2	The ACDC supports the installation and use of moderate-performance removable storage at the client. (ex: CD-ROM drives)	
3	The ACDC supports the installation and use of low-performance removable storage at the client. (ex: floppy diskette drives)	
4	The ACDC does not support the installation and user of any removable storage at the client.	
5	The ACDC supports the attachment of "new" peripherals that follow "industry standard" interfaces at the client. (ex: IEEE 1394-based digital cameras, USB v1.1-based cameras, IBM PC-based parallel port CD-Rewriters, etc.).	
6	The ACDC does not support the attachment of "new" peripherals at the client.	
7	The ACDC can be extended to support "to be determined" devices and applications at the client (context: traditionally done in PCs via plug-in ISA, PCI, or PCMCIA cards).	

This section addresses the needs and wants for resistance to theft for any ACDC.

Ref	***Statement***	**Rating**
1	The ACDC supports client devices that are designed to be theft deterrent at the system level.	
2	The ACDC supports client devices that are designed to be theft deterrent at the component level.	
3	The ACDC does not include theft deterrence in its client devices.	

This section addresses the needs and wants for information security for any ACDC.
Note: this section does not address physical access/security options, only electronic methods such as passwords, public key/private key technology, authentication and encryption.

Ref	*Statement*	Rating
1	The ACDC supports very strong information security on any centralized computing elements.	
2	The ACDC supports moderately strong information security on any centralized computing elements.	
3	The ACDC supports weak information security on any centralized computing elements	
4	The ACDC does not support any significant information security on any centralized computing elements.	
5	The ACDC supports very strong information security in internal communication elements.	
6	The ACDC supports moderately strong information security in internal communication elements.	
7	The ACDC supports weak information security in internal communication elements.	
8	The ACDC does not support any significant information security in internal communication elements.	
9	The ACDC supports very strong information security in client devices.	
10	The ACDC supports moderately strong information security in client devices.	
11	The ACDC supports weak information security in client devices.	
12	The ACDC does not support any significant information security in client devices.	
13	The ACDC supports very strong information security in communication interfaces to other systems.	
14	The ACDC supports moderately strong information security in communication interfaces to other systems.	
15	The ACDC supports weak information security in communication interfaces to other systems.	
16	The ACDC does not support information security in communication interfaces to other systems.	

This section addresses the needs and wants for mobile user support for any ACDC.

Ref	*Statement*	Rating
1	The ACDC supports intra-office mobility effectively.	
2	The ACDC support intra-campus (multi-building) mobility effectively.	
3	The ACDC supports intra-enterprise (wide area but controlled facilities) mobility effectively.	
4	The ACDC supports "nomadic" (virtually anywhere, anytime) mobility effectively.	
5	The ACDC supports intra-office mobility at only a basic level.	
6	The ACDC support intra-campus (multi-building) mobility at only a basic level.	
7	The ACDC supports intra-enterprise (wide area but controlled facilities) mobility at only a basic level.	
8	The ACDC supports "nomadic" (virtually anywhere, anytime) mobility at only a basic level.	
9	The ACDC does not support intra-office mobility.	
10	The ACDC does not support intra-campus (multi-building) mobility.	
11	The ACDC does not support intra-enterprise (wide area but controlled facilities) mobility.	
12	The ACDC does not support "nomadic" (virtually anywhere, anytime) mobility.	

This section addresses the needs and wants for the global/international attributes for any ACDC.

Ref	*Statement*	Rating
1	The ACDC uses hardware and software which is available in a consistent form on a truly global basis. ("truly global basis" means different things to different businesses, but here it is meant to reflect that the hardware and software needs to be identically the same (except for language and regulation-mandated issues) and available in most anywhere that a typical multi-national or global corporation would chose to do business at this time).	
2	The ACDC uses hardware and software which is not available in a consistent form on a truly global basis.	
3	The ACDC uses hardware and software which is available in derivative forms to support many different languages.	
4	The ACDC user hardware and software which is not available in derivative forms to support many different languages.	

This section addresses the needs and wants for the accessibility attributes for any ACDC.

Ref	*Statement*	Rating
1	The ACDC supports hardware and software that provides access for users with disabilities.	
2	The ACDC does not support hardware and software that provides access for users with disabilities.	

This survey has thus far asked for you views on the following attributes of Architectures for Corporate Desktop Computing:

I. Software availability
II. Hardware availability
III. Direct costs
IV. Physical location of hardware
V. Performance
VI. Reliability, Monitoring, and Diagnostics
VII. Electronic communications portion
VIII. User interface portion
IX. Distribution of complexity
X. Serviceability
XI. Upgradability
XII. Information backup and restoration
XIII. Extensibility
XIV. Resistance to theft
XV. Mobile user support
XVI. Global/international attributes
XVII. Accessibility

If there are other attributes of ACDCs which you feel should be considered, please include that information in the "Your additions" section on the following page (add as many pages as you need). If possible, please include an importance rating with the attribute.

Example

For example, consider an automobile architecture. A survey might have failed to recognize that the distance of the top of the windshield to the driver's head is a concern (perceptual) for many taller drivers. In this case, the suggested addition to the survey might be:

I. The distance from the top of the windshield to the driver's head is less than that of a 1999 Ford Taurus.
II. The distance from the top of the windshield to the driver's head is the same as that of a 1999 Ford Taurus.
III. The distance from the top of the windshield to the driver's head is somewhat greater than that of a 1999 Ford Taurus.
IV. The distance from the top of the windshield to the driver's head is much greater than that of a 1999 Ford Taurus.

Of course, actual values rather than a comparative example might be more appropriate in some cases.

Your additions

Thank you so very much for your participation in this survey. Your input is critical to the completion of the aforementioned survey.

Appendix C – Evaluation of Architectures

Architecture Evaluation Against Baseline (Page 1 of 7)

Line Index	Technical Requirement	Ranking	"Baseline" Architecture
1	Off-the-shelf Components	1	De facto standard
2	Network Access Capability	2	PC directly on network
3	Low Complexity Equipment	3	PCs are complex
4	Established Equipment-Equipment Interfaces	4	De facto standards
5	Familiar Human-Equipment Interfaces	5	De facto standards
6	Use Industry Standards	6	Wintel, Ethernet
7	Readily Available Technologies	7	CRTs, RAM, ROM types, drives, cables
8	Use Common Media	8	1.44MB floppies, CD-ROMs
9	Multiple Suppliers	8	Plenty of suppliers
10	Familiar Operating Sys./User Environment	10	MS Windows Family
11	Familiar Metaphors	11	Overlapping Windows, mousing, keyboarding
12	Support Mobile Clients	12	Laptop connection (wired)
13	Capacity Aggregation Capability	13	Upgrade capability for one user at a time
14	Familiar Applications	14	MS Office, Windows Apps
15	System Monitor Capability	14	PCs are weak on this
16	Dominant Design Status	16	The PC/Server Arch. is THE dominant design
17	Allocate Complexity to Central Equipment	17	PCs are complex
18	High Reliability	18	Most PC software is not highly reliable
19	Integrated Help Capability	19	PCs are weak on this
20	Reconfigurable Storage Location	20	PC/Server Arch supports this
21	Dynamic Hardware/Software Function Allocation	21	Only achievable via extensibility/upgradability
22	Consistent Performance	22	Intra-PC performance is consistent, server/network access is not
23	Support Multiple Application "Variants"	23	Can run apps on PC or on Wintel Server (limited server apps)
24	Sophisticated Diagnostics	24	PCs are weak, Wintel Servers are okay
25	Self-Healing Capability	25	PCs have no support for this; Wintel Servers are okay
26	Newest Equipment-Equipment Interfaces	25	Upgrades available early; mainstream adoption comes later
27	Upgradable Equipment	27	PCs are highly upgradable; so are Wintel servers
28	Electronic Security Features	28	License management in PCs is an issue
29	Physical Security Features	29	Local storage in PCs is an issue; also, "theft" appeal

Architecture Evaluation Against Baseline (Page 2 of 7)

Line Index		WinTerm
1	=	HW standard; SW unique but developed
2	-	Network access through server only
3	+	WinTerms are simpler than PCs
4	=	Uses PC & Networking standards
5	=	Yes
6	=	Use Ethernet; often not Intel CPUs
7	=	Uses Readily Available Technologies
8	-	Typically no removable media supported
9	-	Often supplier-specific solutions
10	=	Generally gives access to MS Windows Apps
11	=	Yes
12	-	Typicallly not easy to attach laptop to network conn.
13	-	Requires server upgrade for single-user improvement
14	-	Generally gives access to MS Windows Apps
15	+	Server-centric -- better than PC/Server baseline
16	-	No
17	+	WinTerms are simpler than PCs
18	=	Not likely much better than baseline (same apps)
19	?	?
20	-	No. All storage is on server
21	-	Client device not upgradable -- no options
22	-	Server/network used for all operations -- inconsistent perf.
23	-	No. Only server-based variants can be accessed
24	?	?
25	+	Server-centric -- better than PC/Server baseline
26	-	Client device not upgradable -- no options
27	-	Client device not upgradable -- no options
28	+	Eliminates license management issues
29	+	No local storage issues; not usable at home - low theft appeal

Architecture Evaluation Against Baseline (Page 3 of 7)

Line Index		JavaStation
1	=	HW standard; JavaVM SW unique but developed
2	=	JS directly on Network
3	+	JSs are simpler than PCs
4	=	Uses networking standards
5	=	Yes
6	-	Use ethernet; Java standards still evolving
7	-	Dependence on non-existant apps coded in Java
8	-	Typically no removable media at client device
9	-	Almost no suppliers
10	=	Depends on apps accessed; can be okay
11	=	Yes
12	-	Ethernet available, but Wintel laptops are completely diff.
13	-	Server upgrades required
14	-	Java apps hard to find; not popular titles
15	?	?
16	-	No
17	-	Actually allocates complexity to distributed servers/apps.
18	-	Java stability unproven
19	?	?
20	-	Yes, but not to client device
21	-	Client device not upgradable -- no options
22	-	Server/network used for all operations -- inconsistent perf.
23	-	Yes, but not client-executed versions
24	?	?
25	-	Not clear what happens when Java App fails
26	-	Client device not upgradable -- no options
27	-	Client device not upgradable -- no options
28	=	Eliminates license management issues
29	+	No local storage issues; not usable at home - low theft appeal

Architecture Evaluation Against Baseline (Page 4 of 7)

Line Index		SunRay/HotDesk
1	+	HW standard but in new configuration
2	-	Network access through server only
3	+	SunRay is low complexity device; special interconnect?
4	-	Uses special interconnection device
5	=	Yes
6	=	Yes
7	-	All except special interconnect device
8	+	Can support extensions at client device
9	-	No, only Sun Servers work with it
10	-	Not supported by MS Windows
11	=	Yes
12	-	No, client interconnect is dedicated
13	-	No, requires server upgrade
14	-	No, not MS Windows applications
15	+	Sun OpSys SW is strong on manageability
16	-	No
17	+	SunRays are simpler than PCs
18	+	Probably, depends on Apps
19	?	?
20	=	Yes, allows storage at client via extensibility
21	-	No, not enough bandwidth for extensions at client
22	-	No, software is executed on shared server
23	-	No, only Sun Server-compatible apps available
24	?	?
25	+	Sun supports fail-over and clustering
26	?	Client is limited in extensibility (no parallel bus)
27	=	Extensible via moderate speed serial buses only
28	=	Eliminates license management issues (if no local storage)
29	+	Local storage issues avoidable; not usable at home - low theft appeal

Architecture Evaluation Against Baseline (Page 5 of 7)

Line Index		New Candidate
1	=	HW standard but in new configuration
2	-	Network access through server only
3	+	Client device very simple; "Central PC" repackaged version of today's PC
4	=	Uses PC standards + Gigabit Ethernet over Cat5 copper wire
5	=	Yes
6	=	Yes
7	=	Central PC is a repackage effort; Gigabit on Cat5 may require work
8	+	Supports removeable media as option at client
9	+	Yes -- Suppliers of Wintel PCs/Servers to Corps. are all potential suppliers
10	=	Yes, runs MS Windows natively on Central PC
11	=	Yes
12	=	Interconnect switch in Central PC will route "remote PCs" directly to network
13	+	Yes -- single Central PC can be upgraded without disturbing servers, others
14	=	Yes
15	=	Same as baseline, maybe better if Central PC manageability features used
16	=	Yes and no. Same software, mostly same hardware components
17	+	Yes. Client device is simpler than PC. PC complexity moved to Central PC
18	=	Same as baseline, maybe better if Central PC manageability features used
19	?	?
20	=	Yes, allows storage at client via extensibility
21	-	Yes, allows storage at client via PCI bus in client device
22	=	Yes, all resources for core apps and storage dedicated
23	=	Same as baseline
24	?	?
25	-	Central PC is same as normal PC
26	=	PCI bus in client allows upgrades
27	=	PCI bus in client allows upgrades; serial buses at client allow upgrades
28	=	Same as baseline
29	+	Local storage issues avoidable; not usable at home - low theft appeal

Architecture Evaluation Against Baseline (Page 6 of 7)

Line Index	Technical Requirements	Ranking	SunRay/HotDesk	New Candidate 1
1	Off-the-shelf Components	1	+	=
2	Network Access Capability	2	-	-
3	Low Complexity Equipment	3	+	+
4	Established Equipment-Equipment Interfaces	4	-	=
5	Familiar Human-Equipment Interfaces	5	=	=
6	Use Industry Standards	6	=	=
7	Readily Available Technologies	7	-	=
8	Use Common Media	8	+	+
9	Multiple Suppliers	8	-	+
10	Familiar Operating Sys./User Environment	10	-	=
11	Familiar Metaphors	11	=	=
12	Support Mobile Clients	12	-	=
13	Capacity Aggregation Capability	13	-	+
14	Familiar Applications	14	-	=
15	System Monitor Capability	14	+	=
16	Dominant Design Status	16	-	=
17	Allocate Complexity to Central Equipment	17	+	+
18	High Reliability	18	+	=
19	Integrated Help Capability	19	?	?
20	Reconfigurable Storage Location	20	=	=
21	Dynamic Hardware/Software Function Allocation	21	-	-
22	Consistent Performance	22	-	=
23	Support Multiple Application "Variants"	23	-	=
24	Sophisticated Diagnostics	24	?	?
25	Self-Healing Capability	25	+	-
26	Newest Equipment-Equipment Interfaces	25	?	=
27	Upgradable Equipment	27	=	=
28	Electronic Security Features	28	=	=
29	Physical Security Features	29	+	+

Architecture Evaluation Against Baseline (Page 7 of 7)

Concepts

1 - Traditional (baseline)
- Server for File Sharing, Print Queues
- Server provides some application execution
- Desktop PC for Program and Data Storage
- Desktop PC executes most programs
- Desktop PC accesses other systems/network directly
- Server is on shared network
- Desktop PC has some portions of C/S app

2 - WinTerm
- Server "is" the computer (storage, execution, network access)
- Desktop Terminal runs User Interface
- Server provides access to other systems/network
- Desktop Terminal only accesses server directly
- Desktop Terminal on shared media (ethernet)

3 - JavaStation
- JS runs Java VM and Web browser
- JS download app code "on demand" from any network device
- Servers can be "code servers"
- Servers can also app servers (executing apps for access via JS)
- Does not support any known mobile clients

4 - SunRay/HotDesk
- Server "is" the computer (storage, execution, network access)
- Desktop Terminal is basically just hardware (no "real" programs)
- Desktop Terminal access only server directly
- Uses dedicated media (Fast Ethernet)
- Does not support any known mobile clients

5 - "New" Candidate
- "Centralized PC" is the computer
- Desktop Terminal is basically just hardware (no "real" programs)
- Desktop Terminal access only Centralized PC directly
- Uses dedicated media (Gbit Ethernet)
- Can support "distributed [traditional] PC" attachment