Engineering and Economic Issues in Managing the Paradigm Shift in Computing Environments

Elias Drakopoulos

Matthias J. Merges

Corporate computing is shifting from traditional mainframe proprietary systems to distributed computing that is based on heterogeneous, networked, open-system architectures. The shift is a strategic business initiative supporting financial and operational needs. This paper discusses the various strategic and technological forces that are driving the shift to distributed architectures. In this context, the paper combines financial, business, design, analysis, engineering, and project management techniques, in order to analyze business reengineering requirements and issues affecting the planning, design, engineering, and implementation of information system architectures.

Introduction

In the traditional corporate computing environment, mainframe hardware and software systems were dominant. That is changing. Today, corporations are shifting to distributed computing environments. Mainframe computing, also called *host-centered* computing, is based on proprietary and closed hardware and software, whereas distributed computing, also called *network-centered* computing, is based on open hardware and software.

New network-centered products and systems, unlike their host-centered counterparts, enjoy the technological and financial benefits of an active, global competitive market. In addition, the new distributed computing environments are better able to support current business needs, such as globalization and decentralized decision-making.

By design, distributed computing systems are more flexible and can be modified with minimum, localized impact, making them more responsive than host-centered systems to changing business processes and markets. On the other hand, the successful implementation of distributed systems poses new engineering and operational challenges because they are, by nature, a collection of independent resources using finite bandwidth networks to exchange information to collectively solve complex, data-intensive problems.

Implementing distributed computing requires careful financial management, as the invested costs, or *portfolios*, of host-centered hardware systems must be converted into portfolios of distributed computing assets that have different market life cycles. These portfolios represent significant capital investments or long-term cash-flow obligations and, if mismanaged, can adversely affect the market value of the corporation.

Engineering issues encountered in the design and deployment of distributed computing go beyond the traditional technical considerations to include data network design, operational design, and financial management. This paper combines financial, business, design, analysis, engineering, and project management techniques, to analyze business reengineering requirements and issues affecting the planning, design, engineering, and implementation of information system architectures.

The paper is organized as follows:

- "Business Requirements" presents a brief examination of business requirements in a global competitive environment and the implications for the information systems of a global corporation;
- "Information Systems Strategic Direction"

combines business requirements with the financial, engineering, technology, human factors, organizational trends, and requirements, in order to develop open and distributed computing systems;
- "Distributed Computing Architecture" discusses the overall architecture of a distributed computing system;
- "Design Methodology" describes the methodology used to design and engineer distributed systems;
- "Case Study" illustrates the technical design process of a distributed computing environment.
- "Information System Management" addresses organizational and human interface issues; and finally,
- "Financial Planning" discusses several critical financial planning issues.

Panel 1. Abbreviations, Acronyms, and Terms

API — application programming interface
CPU — central processing unit
DBMS — database management system
FDDI — fiber distributed data interface
ISDN — integrated services digital network
IT — information technology
MIPS — million instructions per second
SQL — structured query language
TCP/IP — transmission control protocol/Internet protocol

Business Requirements

The sole purpose of corporate *information systems* is to support the business processes that are required to carry out the corporation's mission. Corporate *information technology (IT)* infrastructures provide the information required to support and sustain these processes. This dictates a symbiotic relationship between IT and the other functions of the corporation, since IT's traditional administrative role has expanded to the point that it is pervasive in all aspects of managing the modern global enterprise.

Making Timely Decisions. One of the critical factors for a modern corporation's success is its ability to make and execute timely decisions, which, in turn, requires making widely available to all decision-makers timely and accurate information—such as market and competitive analysis, and product and service development and delivery.

Reengineered Processes. In addition, as global markets evolve and businesses react with new corporate strategies and customer solutions, IT must be able to quickly change to provide the new information required to support the reengineered processes.

Thus, the survival of the modern corporation depends on the ability of its IT architecture to respond to the constantly evolving information needs of its business processes. Clearly, business processes must not be driven by the capabilities of IT. Rather, IT must be ever responsive to the needs of the enterprise.

Corporate Trends. Current corporate trends include globalization, commodity product markets, shrinking market windows for the introduction of new products, a decrease in inventory cycles, and an orientation toward customer solutions with many heterogeneous products and services. More and more, decisions need to be made closer to the customer and in real time. It also is widely expected that management's span of control will greatly increase and that there will be a marked increase in the delegation of authority.

Corporate IT Architectures. We purport that, accordingly, modern corporate IT architectures must be:
- Responsive: real-time information must be provided to support distributed decision-making.
- Pervasive: all business processes need to access and share the information necessary to support corporate strategies and tactics.
- Distributed: distributed and mobile organizations will demand computing systems that are capable of being distributed and, thus, easily accessed locally.
- Easily changed: future IT systems must be designed to evolve along with business processes.

Host-centered systems supporting off-site processes and heterogeneous information structures cannot be quickly or easily modified to support emerging flexible organizations. It has been persuasively argued[1] that traditional host-centered IT architectures are not optimal to support rapid change, distributed access, and data conversion—making them unsuitable for the emerging global corporations of the 21st century. Thus, modern IT architectures must be, in contrast, network-centered based.

Corporate IT Infrastructures. The emerging new requirements for corporate IT infrastructures have implications for the IT organization. In particular, IT organizations need to be:

- Accessible: if the IT organization remains isolated in its "glass house"— the traditional host-centered computing environment—it will not be able to understand or react to changing business conditions in a timely manner, thus jeopardizing the corporation's future worth.
- Service oriented: the sole reason of the IT organization's existence is to provide service to support critical business processes.
- Customer focused: the IT organization must supply quality service as defined by its customers.

Cost of the IT Infrastructure. Another business requirement to consider is the cost of the IT infrastructure. Today's emerging market conditions and consequent management trends clearly indicate that the demand for information processing will increase at accelerating rates into the next century. On the other hand, pervasive corporate cost management dictates a decrease in the amount of money available for supporting this processing.

As discussed in the next section, this dilemma can be resolved by moving the IT infrastructure away from host-centered computing and toward distributed systems that can take advantage of the cost effectiveness of new technology. As corporate chief information officers come under increasing pressure to better manage their expenses, distributed computing infrastructures will provide the basis for the solution.

Corporate Portfolio of IT Assets. A final financial business requirement deals with the corporate portfolio of IT assets. To develop the flexibility required by modern IT infrastructures, portfolios of computing assets must be sufficiently liquid to respond to both internal business processes and evolving IT technology in external markets. This issue is addressed in the last section of the paper.

Information Systems Strategic Direction

The cost-value tradeoffs of computing technologies follow traditional modern technology S-curves, as depicted in Figure 1.[2] The S-curve describes the relationship between the performance/value of a computing environment and the effort/investment needed to achieve that performance/value.

The S-curve in Figure 1 demonstrates the limits of a computing environment and the points of *discontinuity*, where limitations of one computing environment

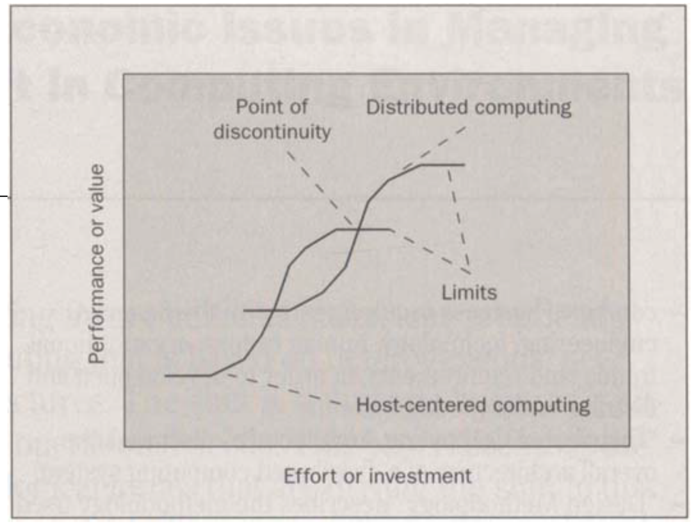

Figure 1. The S-curve describes the relationship between the performance/value of two computing environments—host-centered and network-centered—and the effort/investment needed to achieve that performance/value. It is estimated that host-centered computing has peaked, and that distributed computing is just beginning its climb up the performance/value curve.

reduce its cost effectiveness and cause alternate environments to be developed, based on new structures. We are approaching the top of the S-curve—discontinuity—for host-centered system computing, where the marginal cost exceeds marginal benefits. The marginal cost includes all the costs associated with a change to the computing environment. When adding or changing capacity to host-centered computing infrastructures, many applications are affected because of the nature of the central architecture. Even though the cost of system hardware technology is decreasing, total marginal costs are excessive.

Business demands for more computing at less cost dictate a shift to distributed computing architectures that are attaining their S-curve values.

In addition, the investments required to advance the state of the art of IT technology are rapidly moving away from host-centered systems and toward distributed computing. This trend is indicated by the projected market for future computing products and solutions. For example, it was estimated that by year end 1994 only 10 percent of disk storage shipments were for host-centered system applications.[3]

The development investment trends, along with the inherent capabilities of the competing technologies, also are accelerating the movement of the IT industry to the flat part of the host-centered system S-curve—discontinuity—and to the maximally sloped segment of the S-curve for distributed computing. This discussion of technology, it should be noted, applies to hardware, software, and system technologies.

Host-centered systems have been managed in

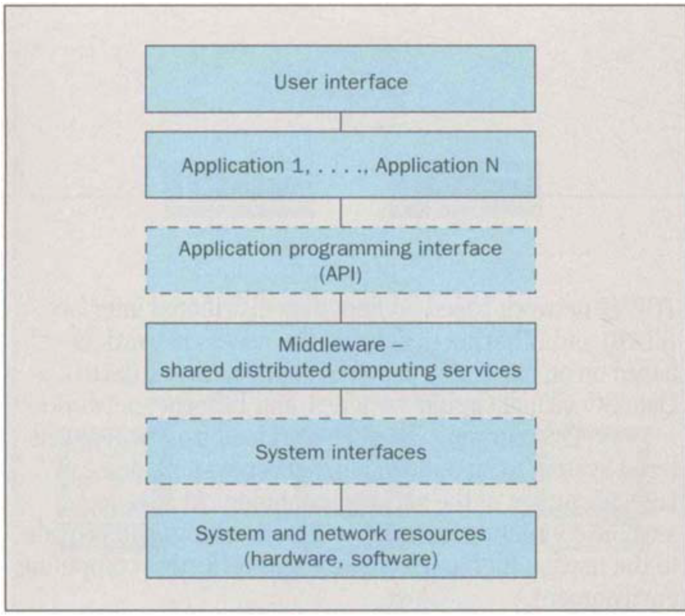

Figure 2. The main components of the distributed computing architecture are shown.

cartel markets at best and usually in near monopolistic markets. The market for these systems is relatively small, the market entry cost substantial, and the software generally proprietary.

On the other hand, with few exceptions, the technology of distributed systems is commodity based because of large market sizes, a proliferation of open software, and established market channels that are not vendor specific. The result is an entrepreneurial market with normal profits and minimum prices.

It is expected that these financial trends will continue as global markets continue to expand for the new distributed computing systems.[4] Managers responsible for corporate IT are expected to react to these market realities by moving their IT infrastructures away from host-based computing to maximize the use of their corporate IT investment.

The previous section argued that the requirements of business information management dictate a corporate IT architecture that is based on network-centered computing. As discussed in the "Case Study," this need is being met at AT&T Bell Laboratories by a research and development IT architecture that is open, networked, and standards-based. This architecture has lead to the design of computing environments that are not only cost effective today, but capable of evolving in a non-disruptive manner. This new architecture takes advantage of new, more cost-effective IT technologies as they become available, enabling the computing environment to continue to meet the business and financial requirements of its customers.

There are, of course, distributed computing systems that exist today in most corporations, although they are recognized, engineered, managed, and connected in varying degrees. They are usually not considered to be part of the official corporate IT infrastructure but, nevertheless, are generally a critical component of the corporation's processes.

But the discussion of moving to a fully distributed IT environment often gives rise to concerns about the costs of converting host-centered system applications. If there are static business processes and applications that require no maintenance or enhancements, it is argued, conversion costs could potentially negate the benefits of moving from host-based systems to distributed computing.

However, based on Bell Laboratories experience in developing its distributed computing environment and understanding the pervasive changes occuring in business processes, we can claim that there are few situations that satisfy these conditions.

Distributed Computing Architecture

The main components of the distributed computing architecture, shown in Figure 2, are:

- User interface,
- Applications,
- Application programming interfaces (APIs),
- Middleware (shared distributed computing services),
- System interfaces, and
- System and network infrastructure.

The basic principle of the architecture is the provisioning of as many computing *shared services* as possible in the form of *shared servers*, achieved through the *middleware* (shared distributed computing services) layer. Examples of shared services offered by this layer are directories, security, remote call procedures, distributed time services, distributed file services, and data management and access. The architecture enables the logical integration of all system resources; makes resource location, access, and program execution transparent to the users; and supports mechanisms that enable efficient resource and system management.

The system and network infrastructure includes computing and storage resources interconnected via a network structure that may consist of a mixture of multiple access networks, switches, hubs, and internetworking devices.

Design Methodology

This section presents a methodology to systematically address the various design issues in distributed computing. The design process consists of the following steps:

- System requirements specification,
- Conceptual design,
- System design and analysis, and
- Implementation.

In the first phase of the design process, the *system requirements* are specified in terms of user-level functions, hardware and software configuration, performance, reliability, availability, serviceability, and cost.

After the requirements specification phase, the *conceptual design* phase addresses a number of design issues related to the system architecture, such as functional allocation, hardware selection, and communication network and software selection.

The *design and analysis* phase that follows analyzes access requirements and workload characteristics and uses analytical and simulation models to study system performance under various load conditions and evaluate design alternatives. The result is a system architecture that is then validated for its functional correctness and its ability to meet the design requirements. This step is iterative and may require several modifications until the design meets the objectives.

The final step is the *implementation* of the design.

Case Study

This section presents a case study of migrating from a host-centered to a distributed computing environment. The study illustrates the application of the design process to a complex client/server design problem. The client/server model of distributed processing is based on a client/server relationship among processes, where shared processors, or servers, respond to requests made by the client processes. The case study is based on an open systems client/server computing environment that provides computing services to more than 3,500 users located at various sites in the United States and Europe.

The system architecture, shown in Figure 3, consists of a cluster of servers interconnected by a redundant TCP/IP network based on fiber data distributed interface (FDDI) and Ethernet networks. The access network is based on an integrated services digital network (ISDN), Datakit® virtual circuits switches, and Ethernet networks.

By migrating the user workload from a host-centered system to an open client/server system, one can take advantage of the various technological advances, such as computing, storage, and networking, and provide to the users a high-performance, cost-effective computing environment.[5]

Conceptual Design. A combination of the dedicated-function and load-sharing is used for functional allocation. In the *dedicated-function* approach, the basic system functions are identified and assigned across multiple processors. This approach leads to simple and low-cost system architectures, but it may also lead to an unbalanced system, due either to an allocation imbalance or workload fluctuations.

In the *load-sharing* function approach, the workload is divided equally among a number of processors which, in aggregate, are capable of supporting the entire workload. The workload assignment process can be either static or dynamic. The use of both approaches permits the advantages of one to be used to offset the disadvantages of the other.

The major functions that need to be considered in the allocation process are:

- Electronic mail,
- File service,
- Database service,
- Computing service,
- Print service,
- Communication services,
- Data management services,
- Software distribution and management services,
- Distributed system management services, and
- Applications processing.

To reduce the complexity of interacting with the system, the software architecture must enable logical integration of the hardware devices, and make resource allocation, access, and program execution transparent to the users.

System Design and Analysis. To design the access network, data collected on the existing host-centered mainframes were used to estimate the number of simultaneously active user sessions that would be migrated to the new client/server system. Figure 4 shows the historical

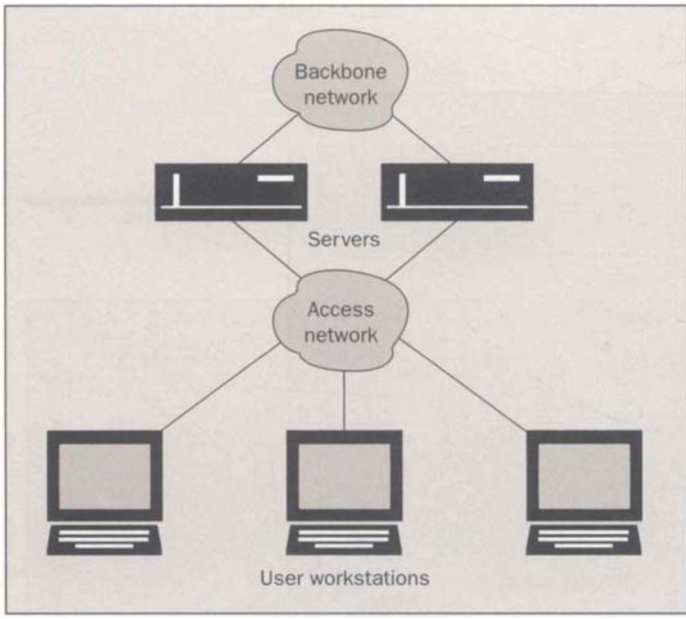

Figure 3. The client/server system architecture consists of a cluster of servers interconnected by a redundant TCP/IP network, based on fiber data distributed interface (FDDI) and Ethernet networks. The access network is based on integrated services digital network (ISDN), Datakit® virtual circuits switches, and Ethernet networks.

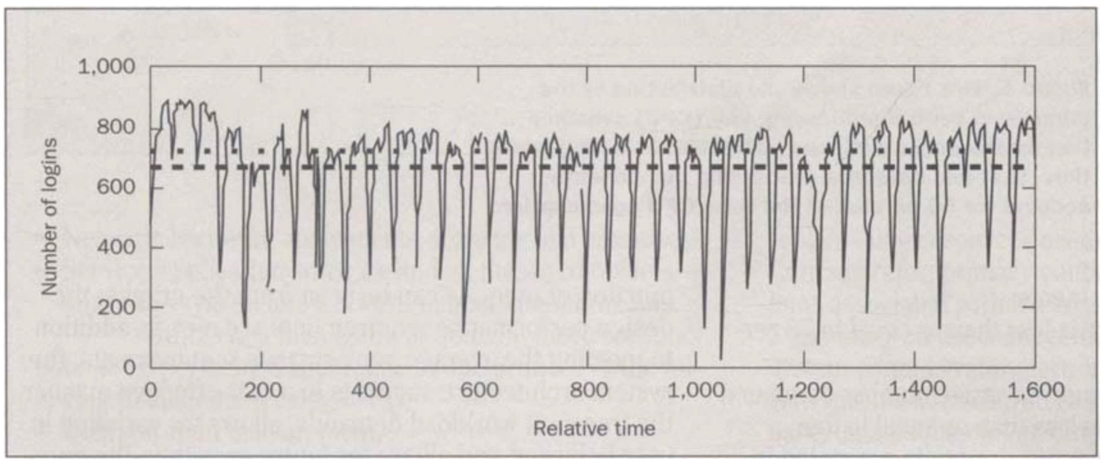

Figure 4. This chart shows the historical login statistics for one of the mainframes in a host-centered network. Data collected on all mainframes that served the user workload indicated an average of 1,400 simultaneously active user sessions. Each data point represents the number of logins over a period of 15 minutes. The dashed line is the median and the dotted line is the mean.

login statistics for one of the mainframes. Data collected on all mainframes that served the user workload indicated an average of 1,400 simultaneously active user sessions.

The major workload components are text processing, database processing, general UNIX* commands, electronic mail, communications, and print. Figure 5 shows the distribution of the prime-time central processing unit (CPU) consumption for the major workload components. The figure shows that during prime time, text and database processing components account for 50 percent of the total CPU consumption.

After analyzing the workload processing requirements across all mainframes used, it was estimated that a total of 110 System/370 instruction set MIPS[6] are required. In terms of storage capacity, it was estimated that the workload requires at least 100 gigabytes of disk storage. To determine the input/output requirements, the activity on all the file systems used by the workload was analyzed, and it was found that the most active file systems are related to mail and database functions.

The database activity of the workload can be divided in two categories: activity associated with structured files and activity associated with flat files. The majority of flat-file databases, as shown in Figure 5, are accessed via a distributed file system, while the structured databases are controlled via a relational database management system and is accessed using structured query language (SQL).

A queuing network model of the system[7,8] was used to estimate the number of servers needed to support the user workload, specify their configuration, study the performance of the system under various load conditions, identify system bottlenecks, and compare design alternatives. The results from the workload analysis, and the target system specifications, constitute the input to the model.

The model was used to derive a system architecture that can support the workload and meet the follow-

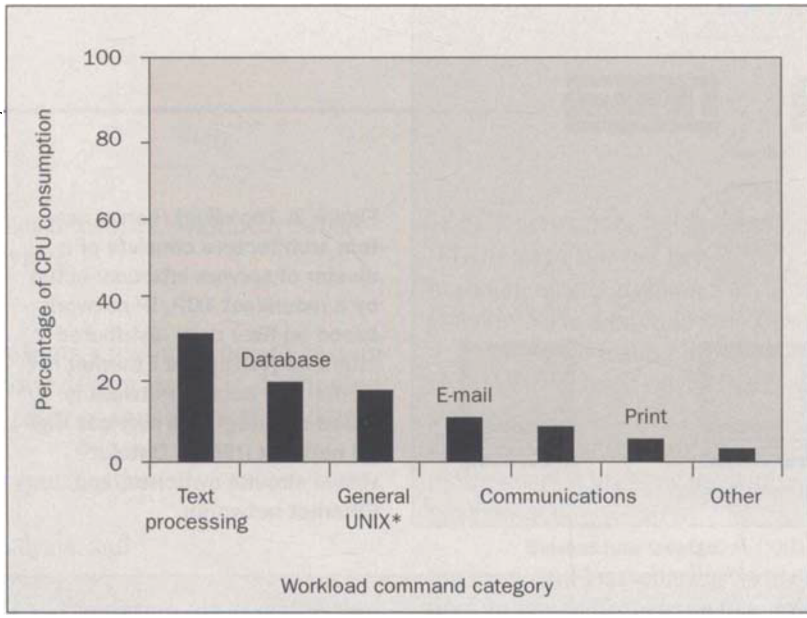

Figure 5. This figure shows the distribution of the prime-time central processing unit (CPU) consumption for the major workload components. During prime time, text and database processing components account for 50 percent of the total CPU consumption.

ing performance requirements:

- Average processor use is less than or equal to 75 percent, and
- Average run queue length (average number of queued runnable processes) is less than or equal to four.

The performance requirements are stated in terms of average performance measures and are chosen to account for variability in the workload. The design must ensure that enough system resources are available to allow for variation in the behavior of the users, and to handle the transient demand that may be placed on the system.

The architecture of the servers considered in the computing environment is not designed to survive hardware or software failures. To meet reliability and serviceability requirements, a combination of server redundancy, disk mirroring, and on-line backup mechanisms were employed. A number of servers are configured as secondary servers for certain functions, such as print service and file service. In the case of a failure of the primary dedicated server, a switchover mechanism, which in most cases is automatic, switches service provision to the secondary server.

Figure 6 shows performance characteristics for a typical server in the environment. Specifically, the processor use and the run queue length are shown as a function of time. As can be seen from the graphs, the design performance requirements are met. In addition to meeting the average performance requirements, the system architecture supports in a cost-effective manner the transient workload demands, allows for variation in user behavior, and allows for future growth in the number of users and their computing workload, storage, and networking requirements.

Information System Management

Now we discuss the management roles in a distributed computing environment. To identify these roles, it is necessary to divide and delegate management functions and demarcate responsibility among multiple managers. There will be a variety of managers fulfilling different functions and operating in different contexts, but having responsibility for the same *managed objects*—defined as an object which represents a resource for the purposes of management. The management structure must be able to model these overlapping responsibilities.

Management domains provide a framework for partitioning management responsibility by grouping together objects that share common management policies. The following management domains were identified:

- System: hardware components, software, processes, and user accounts.

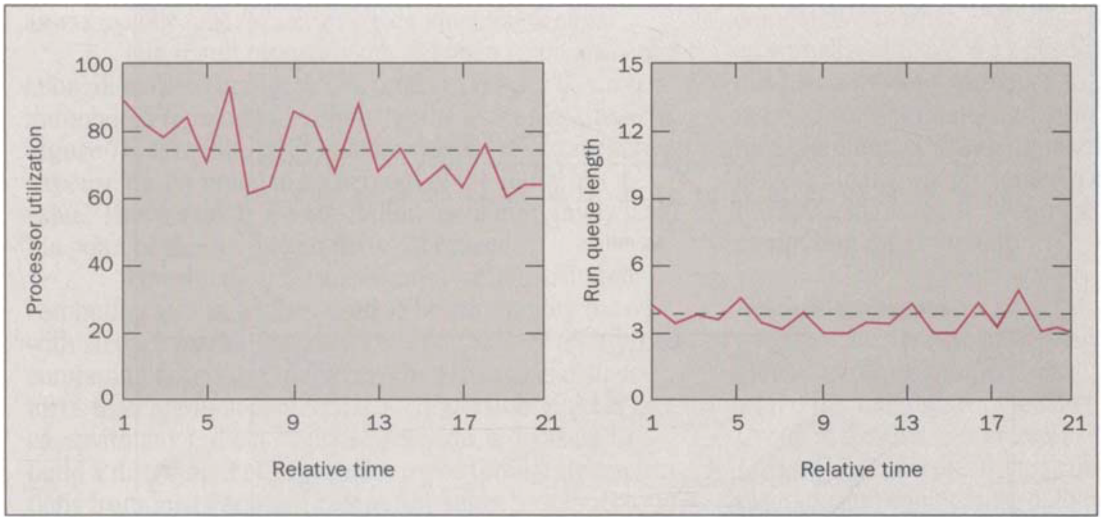

Figure 6. This figure shows the performance characteristics for a typical server in the environment. Specifically, the processor use and the run queue length are shown as a function of time.

- Network: hardware components, software, and processes.
- Service: application software and all managed objects under the system and network management domains.

Within each management domain, there are functions to be performed that relate to one or more of the following management categories:

- Configuration management,
- Performance management,
- Problem management,
- Security management, and
- Asset management.

These categories attempt to create abstractions and compartmentalize the management functions. Nevertheless, these classifications are far from absolute, due to the large number of interdependencies that make the management functions difficult to decompose and quantify.

Based on the above, the following management roles were identified:

- Service manager: responsible for all aspects of service provisioning.
- Service engineer: supports the evaluation of the technical feasibility of new services, long-term technical planning, root cause analysis of problems encountered with the service, technology tracking, and identification of opportunities to improve or initiate services.
- Service delivery engineer: provides technical support for provisioning and completing customer orders for services, resolving problems, and monitoring and replenishing resources needed to provide the service.
- Administrator: primary contact for issues and problems associated with the managed resources. Depending on the management domain, the roles are system administrator, network administrator, and service-specific administrators for such functions as database, data management, output services, security, facilities planning.
- Support administrator: includes operators, input/output support, technical support clerks, and administrative support for the various services provided.

The assignment of management roles to managers is a many-to-many mapping, that is, one or more management roles may be assigned to a manager, and one or more managers may be assigned the same management role.

Management roles are defined in terms of services, which can be classified into the following categories:

- Support services: all the services necessary for the operation, administration, and maintenance of the computing and networking resources.
- Shared services: all the services that manage the resources used by the various application services—such as computing platform, data management, planning and engineering, network application, database, security, software development and support, output, customer support, asset management, and facilities planning services.

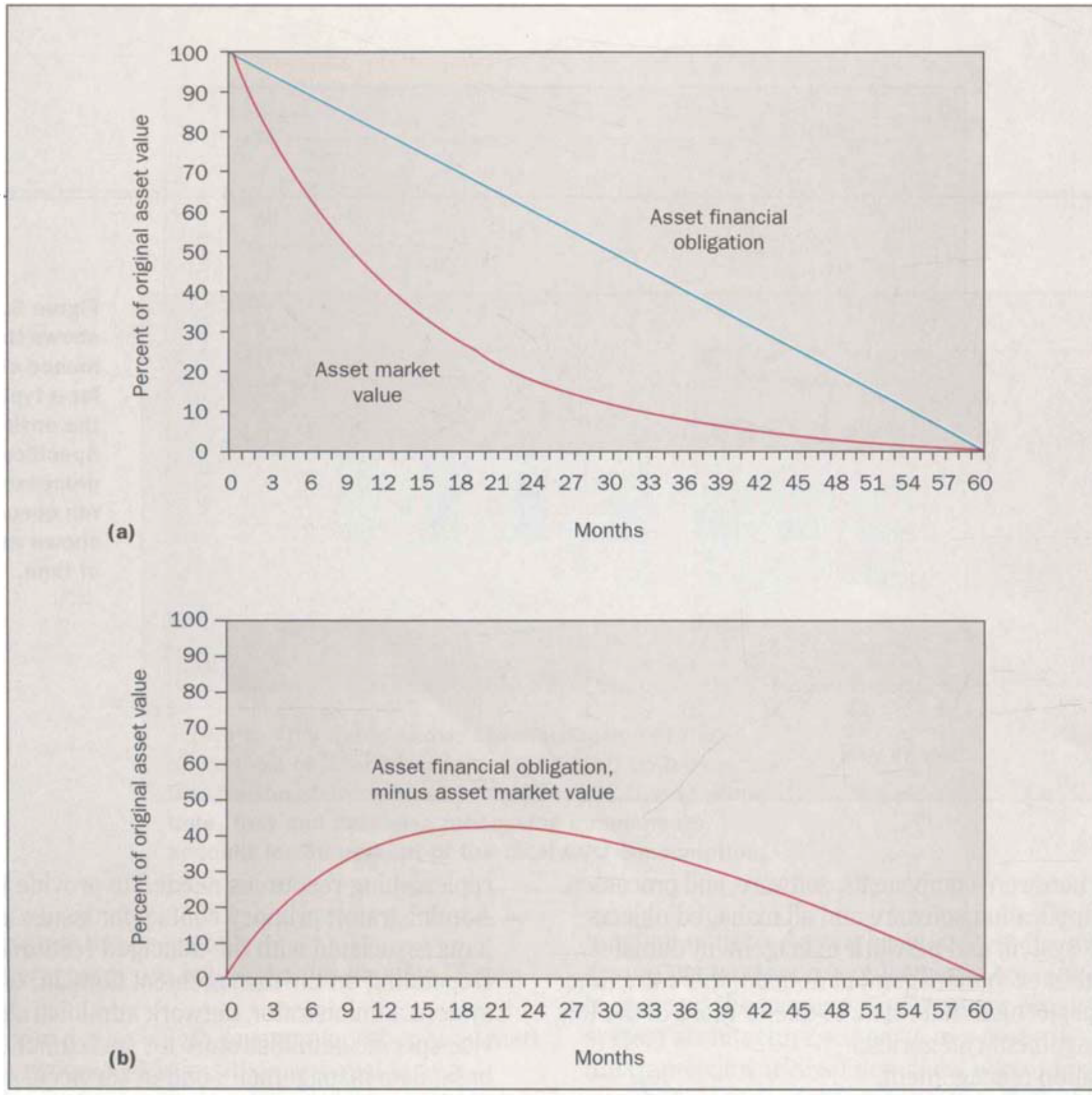

Figure 7. Traditional accounting procedures for the value of computing resources use straight-line depreciation or expense schedules as shown in (A) when, in fact, the market value of the assets usually follows a negative exponential decline. The resulting mismatch between value and obligation, as shown in this illustration, tends to render the assets immobile. For example, as shown in (B) removing the asset at month 24 of its financial life would require an expense outlay equal to 45 percent of the initial asset value.

- Application services: all the applications whose service provisioning is based on the various services in the other two categories.

Financial Planning

Financial obligations under the content of the corporate IT asset portfolio determine the fiscally practical timeframes for change. In particular, changing computing environments from host-centered systems means making significant changes to the portfolio of the IT assets. As with any asset portfolio, changes must take into consideration current market values, as well as accounting and financial obligations. It is not wise to plan to migrate applications from platforms if, for example, the systems are not financially liquid because of mismatches between market and book value, or because lease structures are not in line with market values. The typical five-year straight-line lease with or without early termination penalties can derail the best technical application migration plans.

In addition, any potential change to the capital structure of a corporation, which will usually accompany the shift in computing environments, must be carefully analyzed with respect to all relevant internal and external financial metrics. Traditional accounting procedures for the value of computing resources use straight-line depreciation or expense schedules when, in fact, the market value of the assets usually follows a negative exponential decline.

The resulting mismatch between value and obligation, illustrated in Figure 7A, tends to render the assets immobile. For example, removing the asset illustrated in Figure 7B at month 24 of its financial life would require an expense outlay equal to 45 percent of the initial asset value. This certainly fits the definition of immobility from the point of view of a chief financial officer.

The situation is exacerbated with distributed computing assets, as they tend to be commodity based with steeper market devaluation. Portfolios of distributed computing assets are more sensitive to financial structures than are host-centered system portfolios. As is the case with any critical business decision, a decision to build a distributed IT infrastructure or to migrate applications from host-centered computing must be supported by business case results. Analyzing decisions involving distributed computing systems requires an account of the risk associated with the rapidly evolving technologies and markets. This can be accomplished by developing a likely set of scenarios and evaluating cash flows using expected-value techniques.[9]

As the corporation increases its investment in IT infrastructure, strategic financial planning accounting for the IT assets becomes as important as strategic technology planning. IT plans must meet corporate financial requirements, as well as anticipated user-service requirements. In particular, in light of the need for a flexible response to changing business conditions, IT asset portfolios must be constructed so that individual components are liquid when an IT infrastructure change is required.

This problem can be addressed by constructing lease payments and depreciation schedules that approximate the market's devaluation of the asset, while at the same time not affecting average annual expenses. Thus, one can achieve the asset mobility required to select the most cost-effective computing solutions for business problems, without causing an anomaly to corporate financial results.

Conclusion

This paper treated the various strategic and technological forces that drive the shift from traditional mainframe proprietary computing environments to a distributed environment. We analyzed the ability of the distributed computing environment to take advantage of the technological and financial benefits of a global competitive market, and to support business needs such as globalization and decentralized decision-making. We used a variety of techniques to analyze business reengineering requirements and information systems architecture planning, design, engineering, and implementation issues. Finally, we presented a case study that illustrated the application of the design process to a distributed computing environment.

Acknowledgments

The methods discussed in this paper have been either developed or operationalized by the authors along with other members of the Planning and Engineering Group at the Bell Laboratories QUEST organization at Indian Hill. This consulting group is responsible for the planning and engineering of the Indian Hill computation center. This center was rated best-in-class by the external benchmark consulting firms of Nolan, Norton & Co. and Real Decisions.[10] Currently the center is in the process of migrating its applications from host-centered systems to an open and distributed computing architecture.

(Manuscript approved June 1995)

References

1. B. H. Boar, *The Art of Strategic Planning for Information Technology*, John Wiley and Sons, Inc., New York, 1993
2. S. S. Coleman and R. W. Watson, "The Emerging Paradigm Shift in Storage System Architectures," *Proceedings of the IEEE*, Vol. 81, No. 4, April 1993, pp. 607-619.
3. Gartner Group Storage '93 Conference, Tucson, Arizona, April 1993.
4. *The Gartner Group Monthly Research Review*, March 1993.
5. E. Drakopoulos, "Design Issues in Client/Server Systems," *Proceedings of the 14th IEEE International Phoenix Conference on Computers and Communications*, March 1995, pp. 487-494.
6. J. L. Hennessy and D. A. Patterson, *Computer Architecture, A Quantitative Approach*, Morgan Kauffman Publishers, San Mateo, California, 1990.
7. E. Drakopoulos, "Performance and Traffic Analysis of a Network Based Distributed System," *Computer Communications*, Vol. 16, No. 3, March 1993, pp. 155-167.

8. E. Drakopoulos and M. J. Merges, "Performance Analysis of Client-Server Storage Systems," *IEEE Transactions in Computers*, Vol. 41, No. 11, November 1992, pp. 1442-1452.
9. I. Kleinfield, *Engineering Economics: Analysis for Evaluation of Alternatives*, Van Nostrand Reinhold, New York, 1993.
10. *Bell Labs News*, September 3, 1991, p. 1

Elias Drakopoulos *is a member of technical staff in the Planning and Engineering Group at AT&T Bell Laboratories QUEST organization at Indian Hill, Illinois. He is involved in the design and analysis of communications networks and distributed systems, and performance modeling and evaluation. He joined the company in 1989. He has a B.S.E.E degree from Aristotle University in Thessaloniki, Greece, and M.S.E.E. and a Ph.D. degrees in electrical engineering from Northwestern University in Evanston, Illinois.*

Matthias J. Merges *is a technical manager in the Enterprise Network/Systems Solutions organization in AT&T Bell Laboratories QUEST organization in Indian Hill, Illinois. His organization provides information technology engineering and financial consulting services to internal and external AT&T customers. He joined the company in 1965. He has a B.S. degree in mathematics and physics from King's College, Wilkes Barre, Pennsylvania, an M.S. degree in computer science from Stevens Institute of Techology in Hoboken, New Jersey, an M.B.A. from the University of Chicago, and a Ph.D. in finance from the University of Illinois in Chicago.*