Client/Server Computing: Making the Transition with Macintosh

Contents

Introduction

Today, many companies cannot build new information systems fast enough to keep pace with their changing business needs. And growing cost pressures only compound the problem.

No wonder client/server computing is emerging as the principal model for building information systems in the ’90s. Flexible and scalable, client/server computing systems can be readily adapted to changing business requirements. They combine the strengths of host systems, network servers, and personal computers to deliver applications that, until recently, have simply not been possible. Client/server computing can also reduce costs, both by exploiting the growing performance of personal computers and workgroup servers and by using host systems more efficiently.

But the transition to client/server computing also poses real challenges. Multivendor networks are growing in size and complexity. Installing, supporting, and managing these networks taxes IS resources at a growing rate. In most firms, older host-based systems must remain in place and provide value to the organization for some time to come. Yet the valuable data on those host systems is, in practical terms, often inaccessible to the average user. And for software designers, the client/server model represents a significant departure from host-based development techniques and tools.

Apple® Macintosh® products and technologies can ease the transition to client/server computing. Macintosh lets companies move gradually toward a client/server computing model, while leveraging the information systems that are in place today. The key to this transition is the notion of a “virtually integrated” information system. A virtually integrated information system is one that can deliver new value to users—without forcing them to abandon current computing investments.

A virtually integrated system has two key advantages. First, virtual integration lets the information systems builder continually improve existing systems, without seriously disrupting operations. Second, to the end user, the information system appears to be an integrated whole—even when it is actually composed of incompatible applications and data, residing on a variety of platforms and networks.

With virtual integration, the information systems builder can deliver systems that better exploit existing systems, applications, data, and networks; minimize incremental costs (in system development and deployment time, in training time for IS staff and users, and in maintenance and support); and increase users’ overall effectiveness.

In the following pages, we’ll explore Macintosh technologies and products that deliver virtual integration today. We’ll see how Macintosh can do the following:

- Easily connect with diverse systems and networks, such as large IBM mainframes and LANs—and offer extensive compatibility with existing MS-DOS and MS Windows applications and data.
- Enhance the usability and value of existing host-based applications with sophisticated development tools for front-ending.
- Allow off-the-shelf applications to access existing corporate databases—both relational and nonrelational—on a wide variety of platforms.
- Provide a full suite of development tools for building client/server applications, and, in particular, for virtually integrating existing host-centered applications with personal computer applications.

Finally, we’ll discuss VITAL, Apple’s approach to information systems design. VITAL provides a blueprint for better integrating personal computers with existing enterprise systems. VITAL is a framework for moving today’s information systems toward a client/server computing approach. VITAL advances information systems while maximizing the value of existing technology investments.

Network Integration

As information systems builders move toward a client/server computing architecture, they encounter a basic problem: integrating networks and systems from different vendors into a coherent enterprise network. That's why Apple designed the Macintosh to promote seamless connectivity, and has continually enhanced the connectivity and communications options available. The remarkable ease of connectivity of Macintosh makes it easier to build and maintain the networking foundation for client/server computing.

Apple's overall strategy is to fully integrate Macintosh into four major enterprisewide network types: IBM's SNA/SAA, Digital's DECnet™ and NAS (Network Application Support), TCP/IP (which has emerged from the UNIX world to become a popular multivendor networking standard), and the International Standards Organization's OSI. Macintosh connects to all major host systems—such as IBM mainframes, AS/400 systems, Digital VAX™ systems and UNIX systems—via popular cabling systems such as Ethernet and Token-Ring.

IBM SNA/SAA Networks

Apple's SNA • ps™ Gateway and a wide range of third-party products link Macintosh systems to SNA networks via 3270, 5250, and APPC/LU 6.2 protocols. Apple and independent hardware and software vendors provide capabilities such as terminal emulation, file transfer, graphical front-ending to host applications, and host database access. For example, Apple's SNA • ps 3270, Avatar's MacMainframe, and DCA's MacIRMA all provide 3270 connectivity for Macintosh.

One result of the technology alliance between Apple and IBM is that Macintosh will become even more tightly integrated into IBM enterprise networks. Apple will extend its SNA • ps Gateway to support APPN, IBM's peer-to-peer networking standard for SNA. IBM will also integrate Apple's SNA • ps Gateway into its network management products. This will mean that IBM Netview network management consoles will be able to "see" Macintosh systems on an AppleTalk® network. And IBM's LAN Network Manager will embrace AppleTalk network nodes, as well.

To connect Macintosh to AS/400 systems, Apple will build a 5250 terminal emulator, similar to the SNA • ps 3270 product. And AS/400 services for DOS and OS/2 machines will become fully available to Macintosh users.

To guide customers who are integrating Macintosh into IBM's enterprise networks, IBM has published a network installation and configuration guide, The IBM/Apple Enterprise Networking Guide for SNA Products. This guide is available from any IBM branch office (document number Z325-6027-0).

Digital's DECnet Networks and NAS

Since 1988, the Apple/Digital alliance has provided a stable foundation that has enhanced an array of Macintosh-to-VAX connectivity products. In particular, the alliance spawned Digital's PATHWORKS™ for Macintosh, which provides 13 different network services in a single, integrated product. PATHWORKS offers file and print services, database access, DECwindows™ support, an AppleTalk/DECnet gateway, and LAN interconnections for Macintosh systems, over DECnet wide-area networks.

Macintosh users can also access Digital's electronic-mail systems. PATHWORKS includes Mail for Macintosh, which provides access to VMS Mail. And Macintosh client software for accessing ALL-IN-1 Mail is also available from Digital.

Another Digital product, Desktop ACMS Client for Macintosh, enables Macintosh users to work with Digital's ACMS on-line transaction processing system. Macintosh clients are also available for Digital's conferencing application (Notes) and videotext application (VTX).

Apple and independent software vendors provide VT100/200/300 terminal emulation and file transfer capabilities using LAT, DECnet, or AppleTalk protocols.

TCP/IP and OSI Networks

Apple supports the full range of low-level OSI-compliant network links, including 802.3 Ethernet, 802.5 Token-Ring, X.25, and ISDN. Apple also provides ODA document conversion capabilities, X.400 electronic-mail gateways, and a complete set of OSI protocols for the Macintosh.

With Apple's MacTCP® software, Macintosh becomes a full TCP/IP network node. In addition, Macintosh products from Apple type, type-layout, and printing capabilities. Display and printing of Adobe Type 1 outline fonts will be supported directly. The new Line Layout Manager and enhancements to TrueType™ font technology will provide typographic-quality text layout to all applications software. A new architecture for printing will simplify the user interface and allow output-device vendors to develop high-quality device drivers (for printers, plotters, videotape recorders, and so on) that integrate seamlessly with the rest of the system.

Later, when three-dimensional imaging capabilities are incorporated into the Macintosh imaging system, applications will deliver even more realistic graphic content. In support of this vision, we are taking several steps now. First, we are developing data standards to facilitate interchange of three-dimensional models between documents. Second, we are working on new human interface guidelines to ease the creation, editing, and manipulation of three-dimensional objects. Third, we are developing an open architecture to support third-party rendering software such as Doré and RenderMan.

Application Integration

The interapplication communications (IAC) architecture provides a means for individual applications to work together on one computer and over networks. This integration of applications will help people to work together more closely, provide a means to easily construct custom software solutions, and give users a greater ability to choose the right application tool for the job. Applications supporting IAC get network access by default. In addition, Apple events provides a standard communications language, which enables applications developed by different vendors to work together without special arrangements. IAC allows the following:

Linked/Embedded Documents

Using the Edition Manager (publish and subscribe), applications can share document content. For example, a financial analyst can “publish” spreadsheet-based cash flow data from Lotus 1-2-3 while a manager can “subscribe” to the information over the network to use in a word processing document with automatic updates. If the manager wants to edit the original embedded cash flow spreadsheet (and has been granted security access), he or she can retrieve it directly over the network.

Client/Server Processing

Using the Apple Event Manager, applications can dispatch processing to more powerful computers over the network. For example, using Apple’s MPW® development environment and ToolServer, a programmer on a Macintosh LC can delegate a lengthy compile/link to a more powerful Macintosh Quadra™ series computer.

Combined Application Capabilities

Using the Apple Event Manager, applications can work together flexibly. For example, rather than building forms processing capabilities into its Accountant Inc. package, SoftSync/Bloc uses IAC to treat Shana’s InFormed forms processing software as an easy-to-use front end. This approach allows departments from production to order processing to use the forms they are familiar with in conjunction with the main accounting system.

Customization and Scripting

Building on top of the IAC and Apple event foundation, we have created the Open Scripting Architecture. The open architecture supports a wide variety of scripting system choices, from Command-key assignment utilities (such as CE Software’s QuickKeys) to programming environments (such as Simple Software’s ControlTower, UserLand’s Frontier, and Apple’s future AppleScript™).

With scripts, applications can be used as building blocks to automate work flow, perform unattended maintenance or processing, and facilitate standard network administration. For example, a script could be created that would automatically aggregate on a monthly basis all open purchase orders for departments within a division. The script could create and distribute management summaries and send electronic-mail messages to all managers whose purchase

Graphical Front Ends to Host Applications

"Front-ending" is the technique of building graphical user interfaces that replace the character-oriented screens of applications deployed on mainframes and minicomputers. It is an efficient and effective way to add value to existing production systems by exploiting the graphical user interface of a personal computer to simplify the user's interaction with the host. This simplifies user training and makes users more productive.

There are many situations in which information systems users have to interact with more than one production system—systems that may be on multiple hosts, with different screens and access methods. Front-ending with Macintosh can be particularly valuable in such instances. Macintosh front-ending tools make it easy to construct front ends that present a single interface for user interaction—regardless of the number of different host sessions that may be concealed behind that single interface. The significance of such a system lies in the fact that it gives users a single, unified way to work with disparate systems, which is the essence of virtual integration.

In front-ending, the graphical front-end software runs on a desktop system and uses the communications protocols and wiring already in place. The front-ending software tool preserves all the functions of the host application. It can also be used to extend the functionality of the existing application by exploiting the capabilities of the personal computer on which the front-end software runs.

There are two types of front-ending products available for the Macintosh: development tools for creating custom front ends to in-house applications; and off-the-shelf packages that provide ready-made front ends to widely used host applications, for example, to an electronic-mail system such as PROFS.

Off-the-Shelf Front Ends

MitemMail from Mitem Corporation and MacPROFF from Mariette Systems International provide graphical interfaces to the electronic-mail system. These products insulate users from the native PROFS terminal screen, delivering all of the scheduling, calendaring, and mail functions of PROFS in a far more intuitive and accessible way.

Front-End Development Tools

Front-end development tools for the Macintosh fall into two categories. One type interprets (or remaps) the host-to-terminal data stream, graphically presenting command menus and information to the user. The Both product from Connectivité exemplifies this category. The other type directly accesses the application programming interface (API) built into a terminal emulator. MitemView from Mitem Corporation falls into this category.

Using a sophisticated set of algorithms, the Both product identifies each 3270 screen as it arrives at the Macintosh through a terminal emulator. The developer simply navigates through the sequence of screens that the user will encounter. Both stores the screens locally, automatically naming them sequentially.

The next step is to work with the stored sequence of screens. Using pre-fabricated user-interface "objects," the developer builds the desired graphical user interface for each stored 3270 screen—in a fraction of the time required to work with a 4GL.

MitemView takes a different approach. It provides a pattern-matching engine that (using a terminal emulator) recognizes predefined host states. The developer highlights unique elements in each screen of the host session, and identifies every screen with a name. For selected screens, the developer defines "events" in HyperTalk® (the English-like programming language in HyperCard® from Claris, an Apple subsidiary). These HyperTalk "events" can control functions in other applications; for example, data from a host session can automatically update a desktop spreadsheet.

Macintosh front-ending tools have the advantage of being object oriented. Unlike tools based on 4GLs, they offer the programmer sophistication as well as point-and-click simplicity.

Network Database Access from the Desktop

As users demand better access to corporate data, particularly for analysis and reporting, the information systems community faces a serious challenge. Data often resides in different places within different databases, located on a variety of computing platforms. And to access most data requires considerable expertise. Users must find and connect to the right systems, navigate through long sequences of terminal screens in a variety of mainframe environments, and know SQL to query relational databases.

Virtual integration—and the Macintosh computer—can be the answer to this problem. Macintosh offers advanced tools for providing users with access to a variety of corporate databases—both relational and nonrelational. Using applications written with these tools, Macintosh users can seamlessly access the information stored in such databases, without being aware of where the data resides and without having to learn the complexities of query languages such as SQL.

DAL: Virtual Integration and Relational Data

Data Access Language (DAL), is a software product from Apple that links stand-alone personal computer applications such as spreadsheets and desktop databases to relational databases on many different platforms. By exploiting DAL, information systems builders can deliver instant data access to users, with a simplified development effort. Little user training is required, and support and maintenance resources are minimized. With DAL, users can apply familiar personal computer applications such as spreadsheets or other analytical tools to their ad hoc reporting and data analysis needs, and become self-sufficient data users.

DAL contains two key components: client software and server software. On the client side, DAL provides mechanisms to build SQL query and update statements and to transport the requests across the network to the appropriate host or server system. On the server side, DAL interprets the request for the specific database, assembles the data, and transports the results back across the network to the client application.

A distinguishing feature of Data Access Language is the single programming interface that it provides on the client side, through which multiple relational databases on different platforms can be accessed on the server side. In other words, with a single programming effort on the client side, a user application can be created to provide access to information on a wide range of host and server systems. In addition, because DAL support is available for both Macintosh and MS Windows, a single programming effort using DAL produces a cross-platform data access solution.

Apple has broadly licensed DAL technology for all the leading client and server platforms, including IBM mainframes, minicomputers, such as VAX/VMS and AS/400, and UNIX systems. DAL supports most of the major SQL-based data management systems, including DB2, ORACLE, Sybase, Informix, Ingres, and Rdb, and will soon support major production data management systems, including CA-IDMS and CA-Datacom.

Data Access Language gives software developers the ability to provide user access to information on a wide range of host and server systems, through a single programming interface. With DAL, corporate application developers can quickly create powerful client/server applications in cross-platform environments.

With DAL's single programming interface, corporate application developers can easily target information on multiple types of databases and hardware. In addition, "DAL-aware" personal computer applications are available from key software companies, including Lotus, Microsoft, and Claris.

Accessing Nonrelational Databases

With EDA/SQL from Information Builders, Macintosh users can transparently access nonrelational data residing within 45 different databases and file structures, on 35 hardware platforms. EDA/SQL also lets Macintosh participate in IBM's Information Warehouse framework.

EDA/SQL provides an ANSI-standard SQL API for the Macintosh. In addition, HyperCard developers can easily integrate EDA/SQL with their applications through a set of XCMDs, using the EDA/SQL HyperCard Extender.

EDA/SQL exploits a database access API that is built into the Macintosh operating system. An application based on this API can transparently access virtually any database management system—relational databases via DAL and nonrelational databases via EDA/SQL. Such applications enable Macintosh users to perform ad hoc data queries quickly and easily.

Query Tools: Off-the-Shelf Data Access

Query tools are database access utilities that manage data connections, query remote databases, and manipulate data from multiple sources. Macintosh-based query tools currently available from independent software vendors include DataPrism from Brio Technology and GQL from Andyne Computing.

Query tools make it easy to incorporate database access capability into existing applications and gain the efficiencies of client/ server computing. For example, by combining a Macintosh spreadsheet with an off-the-shelf query tool, a spreadsheet user can query a database on a remote server. The results of the query return to the spreadsheet,

in whatever location the user specifies.

With query tools, developers can quickly incorporate data access functionality into custom business applications, providing simultaneous connectivity to multiple relational databases with minimal effort.

Building Client/Server Applications

Front-ending and desktop data access using DAL are steps in a progression toward true client/server applications—applications that are written to split the applications logic between desktop and server, so as to take advantage of the processing power available on a network. And Macintosh provides the tools and technologies needed to continue this progression.

For applications development, Macintosh provides a complete and sophisticated application development environment, as well as tools for every stage of the application life-cycle—design and prototyping; editing, compiling and linking; and testing and debugging. Tools for cross-platform development and object-oriented programming are also available.

In addition, the Macintosh operating system offers a number of advanced services and APIs that simplify the task of building client/server applications. Using these system-level services and APIs, a Macintosh application can exploit a variety of communications links, access various databases on a network, and invoke applications resident on other Macintosh systems on the network.

In effect, the system software capabilities built into the Macintosh enable information system builders to virtually integrate their Macintosh applications with their computing infrastructure—today and tomorrow.

Development Environments and Languages

Macintosh Programmer's Workshop (MPW®) is an integrated development environment that supports many languages, including C, Pascal, and Fortran, plus the object-oriented C++. In addition to compilers, MPW includes many useful tools—a text editor, a source-level debugger, and a source code and project management facility.

Independent vendors also offer programming environments for Macintosh. Products such as Think C and Think Pascal from Symantec offer fast compilers, integrated graphical debuggers and object-oriented programming features. Developers often use these products in conjunction with MPW—using the Think tools for routine development and MPW for source code control and formal releases of applications.

Cross-Platform Development

Macintosh offers a number of tools for cross-platform development. These tools let IS develop applications that will run on multiple desktop platforms with a single development effort.

For example, Information Builders offers two products that support both Macintosh and MS-DOS/Windows: Focus, the popular 4GL, and EDA/SQL, for accessing nonrelational databases. Both Lotus 1-2-3 and Microsoft Excel run on Macintosh and MS-DOS/Windows. And the FoxBase and Omnis 7 databases also support both platforms Apple's Data Access Language is a cross-platform tool. It is supported \on major mainframe and minicomputer systems for all popular relational databases. In addition to Apple's DAL client for Macintosh, Blyth Software provides a DAL client for MS Windows. Blyth's Omnis 7 creates applications that run on both Macintosh and MS Windows. And Neuron Data's Object Nexpert also provides cross-platform development capabilities.

A/UX® is Apple's implementation of industry-standard Unix. It combines the best of the Unix and Macintosh operating systems in a single package. With A/UX tools, developers can build Macintosh, UNIX, and X Window System applications, and all can run simultaneously on the same Macintosh. For users, A/UX provides all the ease of use and consistency of Macintosh. And it gives Unix users access to Macintosh productivity applications.

Object-Oriented Programming

MacApp® is Apple's object-oriented application framework for the Macintosh. A mature class library, MacApp 3.0 (released in 1991) supports both C++ and Object Pascal languages, along with new features of the latest release of the Macintosh operating system: System 7™.

MacApp provides a user-interface framework for quickly building sophisticated applications. MacApp code is modular and reusable—providing enormous savings in code maintenance and boosting programmer productivity.

ProGraph from TGS Systems is an object-oriented, visual programming environment that can greatly speed applications development. It includes a visual programming language, an integrated editor/interpreter, a WYSIWYG interface builder, an integrated database engine, and a compiler. The editor/interpreter makes it possible to build, modify, and debug an entire application while it is running.

Linking Documents and Applications—over a Network

Of all the capabilities offered by the Macintosh operating system, perhaps the most significant is IAC, the interapplication communications architecture of Macintosh. IAC allows dissimilar applications to work together, over a network, by exchanging data and commands.

The interapplication data exchange capability is called publish and subscribe. The publish and subscribe function is provided by the Edition Manager. It creates live data links between applications. For example, a financial analyst using a Macintosh could "publish" information from a Lotus spreadsheet to other Macintosh users on the network. Another person (or many others, for that matter) could then "subscribe" to the spreadsheet data, through a WordPerfect document on another Macintosh.

Whenever the "published" data gets updated in the Lotus spreadsheet on the "publishing" Macintosh, the WordPerfect document is automatically updated, over the network, on the "subscribing" Macintosh.

The Apple Event Manager is another, more powerful, interapplication communications service. It allows one Macintosh application to execute commands automatically at the request of another Macintosh application, through a standardized set of messages called Apple events. With Apple events, off-the-shelf Macintosh applications can be used like building blocks to deliver a set of capabilities custom-configured for performing a specific set of tasks.

What's more, because IAC can work over a network, a stand-alone application on one Macintosh can be coupled with another application on a different Macintosh—a form of virtual integration that creates a client/server application in a modular fashion from existing applications.

Apple's proposed Open Collaboration Environment (OCE) will extend the interapplication communications architecture of Macintosh even further, by providing store and forward messaging capabilities. Using the store and forward mechanism of OCE, interapplication communications can take place over a network even on a deferred basis.

This means that a local application could invoke services from a remote application, even if the remote application were not running at the time it is invoked. OCE will guarantee message delivery to the other application. Furthermore, OCE will make interapplication communications secure via built-in services for interprocess authentication and message encryption.

VITAL: Virtual Integration for the Enterprise

In addition to providing Macintosh tools and technologies that support virtual integration today, Apple has also developed a framework for designing virtually integrated information systems. This framework is called VITAL (Virtually Integrated Technical Architecture Life-cycle). The VITAL architecture is a technical design blueprint for creating modular, user-centered information systems. VITAL was developed by Apple's own IS organization with cooperation from Digital Equipment Corporation.

VITAL extends the principles of virtual integration to the entire enterprise. It lets IS professionals make the most of current information systems, benefit from client/server computing, and deliver superior value to users.

The VITAL Approach

A rapidly changing business environment requires highly adaptable systems. With VITAL, designers build custom systems from generalized services that can be reused and shared. VITAL identifies these services and specifies how to construct them; how to create, use and manage data; and how to use networked hardware resources effectively.

VITAL lets IS managers construct systems that:

- Provide easy user access to information.
- Are platform-independent.
- Accommodate rapid changes in business needs and available technology.
- Are designed to be cost-effective.
- Support a gradual migration to new systems.
- Integrate installed technology.

VITAL is a modular and open architecture. It provides a platform-independent framework for constructing distributed, multivendor information systems. It accommodates and adds value to popular network, database, and data dictionary/repository environments, whether planned or in place. VITAL complements other major architectural approaches, such as IBM's SAA, Digital's NAS, Open Systems Foundation's OSF1, and DCE.

The VITAL Model

VITAL divides enterprise systems into five logically interrelated environments, linked by a systems infrastructure. The VITAL model describes the standard components of information systems and defines how these components operate with one another. These are the VITAL environments:

- The desktop integration environment provides the human interface, personal application support, navigation, and interconnection to the enterprise computing network.
- The data capture environment supports update-oriented transaction systems.
- The data access environment supports reporting and decision-support applications.
- The repository environment defines and maintains consistency of data definitions, business rules, location of resources and availability of resources on the network.
- The systems infrastructure environment encompasses the utility software, hardware, network, organization, security, standards, and disciplines upon which the other components depend.

Using the VITAL model, IS managers can build system components today that they can easily adapt to future requirements. VITAL allows for continual improvement without significantly disrupting current operations.

Conclusion

Information systems cannot instantly shift to a client/server computing model. Yet users are demanding information systems that are both accessible and usable. They also want the systems to adapt quickly to the changes in the business, even though budgets are shrinking.

Virtual integration is a flexible, incremental response to this challenge. It delivers superior value to the user, while avoiding large-scale disruption of existing systems. It produces results quickly. And it costs less at every level—development, support, training, and maintenance.

With Macintosh technologies and products, information systems builders can create virtually integrated systems—today. Macintosh tools can make the journey to client/server technology easier and more manageable, yet efficient, enabling IS professionals to deliver elegant, high effective solutions to their users.

Apple Computer, Inc.
20525 Mariani Avenue
Cupertino, California 95014