DOCUMENT RESUME

ED 401 862 HE 029 715

TITLE	Information Technology Architectures. New Opportunities for Partnering, CAUSE94. Track VI.
INSTITUTION	CAUSE, Boulder, Colo.
PUB DATE	95
NOTE	91p.; In: New Opportunities for Partnering. Proceedings of the 1994 CAUSE Annual Conference (Orlando, Florida, November 29-December 2, 1994); see HE 029 709.
AVAILABLE FROM	CAUSE Information Resources Library, 4840 Pearl East Circle, Suite 302E, Boulder, CO 80303 (Individual papers available to CAUSE members at cost of reproduction).
PUB TYPE	Reports - Descriptive (141) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC04 Plus Postage.
DESCRIPTORS	Colleges; *Computer Networks; Computers; Cooperation; Educational Planning; Higher Education; *Information Management; *Information Networks; Information Systems; *Information Technology; Models; *Partnerships in Education; Retrenchment; Technological Advancement; Universities
IDENTIFIERS	*Campus Wide Information Systems; *CAUSE National Conference; Thomas More College KY; University of California Davis

ABSTRACT

Eight papers are presented from the 1994 CAUSE conference track on information technology architectures as applied to higher education institutions. The papers include: (1) "Reshaping the Enterprise: Building the Next Generation of Information Systems Through Information Architecture and Processing Reengineering," which notes developments at the University of Pittsburgh (Nicholas C. Laudato and Dennis J. DeSantis); (2) "A Distributed Computing Architecture and Transition Strategy," (Joan Gargano); (3) "Getting the Right Fit: Institutional Downsizing Without Capsizing," which discusses the replacement of mainframe computers with workstations at Lehigh University (Pennsylvania) (Timothy J. Foley); (4) "Rightsizing--Changing the Culture," which reviews the transition from a mainframe to client/server environment at Syracuse University (New York) (Sue Borel and Natalie Vincent); (5) "A Data Warehouse--The Best Buy for the Money," which discusses the experiences of Catholic University (District of Columbia) experiences with Data Warehouse technology (Leonard J. Mignerey); (6) "A Data Warehouse: Two Years Later...Lessons Learned," which reviews developments at Arizona State University (John D. Porter and John J. Rome); (7) "Administrative Systems Vendors Talk Turkey on Client/Server Computing" (John Stuckey); and (8) "Migrating from Host-Based Computing to Client/Server" (Jay Blum and Richard Reno). (Some papers contain references.) (MDM)

New Opportunities for Partnering

CAUSE 94

TRACK VI
INFORMATION TECHNOLOGY ARCHITECTURES

Coordinator: Jacobus T. Meij

Proceedings of the 1994 CAUSE Annual Conference

November 29–December 2, 1994
Walt Disney World Dolphin
Orlando, Florida

Reshaping the Enterprise: Building the Next Generation of Information Systems through Information Architecture and Process Reengineering

Nicholas C. Laudato
Dennis J. DeSantis

University of Pittsburgh
Pittsburgh
Pennsylvania

A team of faculty and staff at the University of Pittsburgh has completed the design of an enterprise-wide information architecture and a framework for engaging the University community in business process reengineering. The architecture provides the blueprint for developing an integrated set of information services, processes, and technologies, enabling significant efficiencies in business and service processes, and facilitating informed decisions concerning information technology expenditures and acquisitions. Revolutionary in design, the architecture supports evolutionary implementation and intelligent use of legacy systems. The implementation does not adhere to a traditional master plan, but rather adapts principles taken from the Oregon Experiment, to grow the envisioned information system from the ground up. This paper reviews a unique approach to creating the architecture and initiating its implementation. The approach included building consensus on a general philosophy for information systems, utilizing pattern-based abstraction techniques, applying data modeling and application prototyping, and tightly coupling the information architecture with efforts to reengineer the workplace.

Reshaping the Enterprise: Building the Next Generation of Information Systems through Information Architecture and Process Reengineering

Nicholas C. Laudato and Dennis J. DeSantis
University of Pittsburgh

A team of faculty and staff at the University of Pittsburgh has completed the design of an enterprise-wide information architecture and a framework for engaging the University community in business process reengineering. The architecture provides the blueprint for developing an integrated set of information services, processes, and technologies, enabling significant efficiencies in business and service processes, and facilitating informed decisions concerning information technology expenditures and acquisitions.

This paper reviews a unique approach to creating the architecture and initiating its implementation. The approach included building consensus on a general philosophy for information systems, utilizing pattern-based abstraction techniques, applying data modeling and application prototyping, and tightly coupling the information architecture with efforts to reengineer the workplace.

Background

The University of Pittsburgh is an independent, nonsectarian, coeducational, public research institution. Founded in 1787, it is a state-related member of the Commonwealth of Pennsylvania System of Higher Education. In addition to the main campus in Pittsburgh, the University operates four regional campuses. The Johnstown, Greensburg, and Bradford campuses offer four-year baccalaureate programs, and the Titusville campus offers lower-division programs and two-year degrees. Among its five campuses, the University offers over 400 degree programs, and for fiscal year 1994, conferred 7,079 undergraduate, graduate, and professional degrees. The University enrolled 33,756 students (headcount) at its five campuses during the Fall Term, 1993.

The University of Pittsburgh has a central-site information system configuration. Like most systems, it was not specifically architected as it is currently configured, but rather has evolved over the years to meet specialized functional needs. This central site system relies primarily on a mainframe environment using an IBM 3090-500E computer system dedicated to administrative computing applications.

The Administrative Information Systems (AIS) group within the Computing and Information Services responsibility center is charged with supporting the administrative computing needs of the University. AIS is staffed by approximately 75 personnel skilled in creating and supporting batch and character-based interactive systems. This environment encompasses the MVS (Multiprogramming Virtual Storage) operating system, the CICS (Customer Information Control System) telecommunications monitor, file systems, database management systems (Cincom's Supra and Supra II), and the COBOL and MANTIS (4GL) programming languages.

Most of the University's financial, student, library, and personnel systems run in this environment. Some exceptions include the central purchasing system, running on a departmental minicomputer, and an institutional reporting system, running on a VAX/VMS system. For the latter system, data is extracted from the MVS mainframe, and loaded into an Oracle database.

4

Most of the current administrative systems rely upon batch processing and paper reporting on a scheduled basis, but ISIS (Integrated Student Information System) also provides on-line data entry and inquiry. The majority of mainframe applications utilize proprietary files and databases, making access to the data extremely difficult, requiring significant manual intervention.

As in most other large institutions, there are islands of automation throughout the University in the form of thousands of microcomputers and hundreds of LAN's. The desktop devices are 67% PC's, 30% Macintoshes and 3% Unix workstations. Some of the LAN's are quite large with over 100 desktop devices connected. These types of information processing platforms are found in business units, administrative offices, and most of the schools and departments throughout the University. In addition, there are stand-alone PC's and Macintoshes used for personal productivity applications. These LAN's and stand-alone units are considered by the owning units to be an integral part of information services provided to end-users. Many of them support business applications that duplicate or complement some of the functionality of the central systems.

Some of the data used by these local applications are duplicates of the data maintained on the AIS mainframe with some local enhancements. This data is either entered from the same forms that are sent to the central business units for entry, entered from reports generated by the central system, or downloaded from the mainframe system for use by the local applications. This duplication is quite costly in terms of personnel, hardware and software. But a more critical issue is the timeliness and accuracy of the information on these local systems as compared to the central site systems, and the difficulty of integrating data from multiple systems and platforms. Since there are inconsistencies between multiple sources of data, a major effort involves reconciliation between the data on the local systems and the data on the mainframe.

Project Mission and Goals

Like many institutions of its type, the University finds itself in an economic, social, and political climate that demands the ability to respond to local, regional, national, and international changes in a timely and relevant manner. To accomplish this, University leaders must be able to access and utilize information about all aspects of the enterprise and must change the way its people plan, make decisions, and perform work. In short, the University must transform itself into a modern organization where information is viewed as an asset and used to strategic advantage.

As an initial step in this transformation, the Senior Vice Chancellor for Business and Finance conceived an approach in August, 1992, and initiated the Information Architecture and Process Innovation Project in February, 1993. The project was headed by a senior faculty member and staffed by an advanced graduate student and three staff members taken from their normal responsibilities for the duration of the project. The team distributed its final report to the University community in June, 1994. The project staff defined the following mission:

- Design an architecture for the University Information System (UIS) that will provide a framework for making decisions about information systems and for improving the UIS in the future;
- Establish a methodology for business process reengineering using the UIS; and
- Develop a plan for migrating from the current systems to the envisioned UIS.

The architecture will provide an overall, high level design for the UIS, identifying scope, direction, components, relationships, and behaviors. Understanding and intelligently deploying information technology in compliance with the architecture will, in turn, play a crucial role in successfully reengineering the University's business processes.

Key Elements of the Project's Approach

The Information Architecture and Process Innovation Project employed a methodology that combined information engineering with business process reengineering. These two components have a symbiotic relationship - the information processing technology empowers users and customers to reengineer business processes, and the reengineered processes determine the need for the information technology.

The project began with the articulation of a philosophy and set of architectural principles. The creation of the University Information Systems Philosophy Statement directly involved over 100 faculty and staff. The statement was debated in three formal focus groups that were specifically configured to represent all constituencies in the University. It was also published in the *University Times* and in several electronic bulletin boards. Through this process, the philosophy statement was refined to reflect the desired goals and directions of the entire University community.

Because of the broad scope of the envisioned University Information System, it became clear that its implementation would have to be phased in over several years. Consequently, when choosing an implementation strategy, the project team eschewed the traditional master plan in favor of a pattern-based approach to building the information architecture. This methodology was inspired by the Oregon Experiment[1], a highly successful approach used over the past 20 years in designing the University of Oregon campus. The methodology recognized many parallels between the architecture of towns and buildings and that of information systems.

In a pattern-based approach, the architecture is documented in a set of patterns, or information processing principles. Decisions about developing, modifying, or acquiring components of the architecture are made by evaluating proposals based on their adherence to the specified patterns. The patterns are subject to on-going review and refinement to ensure that they incorporate advancing technology and continue to meet the needs for which they were designed. The information architecture will evolve as more and more projects are implemented that comply with its specifications.

The patterns must be communally designed and adopted, and will guide the design of everything in the University Information System. Patterns can be both very large and general, as well as very small and specific. Some general patterns deal with the behavior of computer interfaces, some with the distribution of data, some with hardware configurations, some with network protocols, and others with data access methods. More specific patterns deal with report formats, application-specific functions, ordering of data on displays, etc.

The use of a pattern-based approach prevented the project staff from being overwhelmed by the volume of small details necessary to implement a specific task for a specific function for a specific application. Such details are better addressed using prototyping techniques at design time, not at the architectural phase of an information system. The project staff therefore developed a set of common information processing tasks based upon a series of interviews and an analysis of user requirements. The architecture is a response to these patterns of information use across all University activities and related processes which are found in every application.

The architecture project staff preferred to recommend guidelines that could be implemented using state-of-the-practice technology and reasonably cost efficient methods. For this reason, many of the principles espoused in the architecture statement were illustrated through a set of prototype applications that would serve as "proof of concept" and validate the premises put forth in the philosophy statement.

The architecture staff completed four major prototypes during the life of the project. The prototypes illustrated several information processing tasks (patterns) that had been articulated in the architecture statement. For example, a ***finder***, is used to identify an object in the database that the user wishes to view. A finder prompts the user for information that could either uniquely identify the desired object, or identify a list of objects. If the search results in more than one object, the prototype would generate a ***browser***. A browser provides a list of objects, with enough information to allow the user to select the exact object to be viewed. Finally, a ***viewer***, displays the object. The viewer is typically segmented into pages or scrolling sections to allow all attributes associated with the object to be viewed without invoking additional transactions. Viewers also provide "hot button" links to other associated viewers and functions.

One of the premises of the architecture is that these three patterns, among many others, would repeat over and over again in different applications, with only the specific data elements changing from application to application. For example, a student finder would prompt for an ID number, but also allow a search on name; a purchase order finder would prompt for PO number, but allow a search by account number, user, vendor, and commodity; and a course section finder would prompt for term and course reference number, but allow a search by subject, number, and campus. If all of the University's business applications were constructed from such recurring patterns, it would be easier for users to master the interface and seamlessly move from one application to another.

In addition to creating prototype applications to illustrate architectural principles, the project staff completed a pilot business process reengineering effort to test the process innovation methodology they had developed. After identifying all business processes in the enterprise, the project staff selected the procurement process. A reengineering team, composed of a representative cross section of faculty, staff, and others, redesigned the procurement process in six months.

Information Architecture Philosophy

The philosophy and related principles provide a framework for the information architecture by articulating the objectives and quality characteristics that the architecture should follow. These, in turn, are intended to guide the analysis, design, and decisions made relative to all aspects of information systems and processes at the University. They determine the technological approach taken in defining components of the architecture and how they must operate, and are meant to provide a set of patterns by which information system design decisions can be made. Some of the key components of the UIS philosophy include:

- Regard and manage information, information technology, and infrastructure as University assets.
- Capture data one time, at its source.
- Enable organizational units and individuals to share information by making data and documents visible via seamless interconnections and adherence to database standards.

- Assure the quality of information (timeliness, reliability, and accuracy) via a centralized data and document administration function with established data ownership and stewardship policies.
- Reduce the manual effort and paper required to perform information processing activities.
- Facilitate flexibility and ease of adapting to changes in policy, to incremental improvements in processes, to specific needs of local units, and to advances in technology.
- Guarantee choice via a systems environment that is open technologically, operationally, and commercially.
- Utilize the client/server model as the basic paradigm for applications in the UIS.
- Implement a common graphical user interface (GUI) for all business applications. The common GUI will provide a consistent look and feel across all applications, be easy to learn and use, be intuitive and consistent with the standards relative to its particular platform, and enable easy transferability of skill from one application to the next, facilitating substitutability of personnel across applications.
- Ensure effective use of information technology via education and training.

Architectural View

To its users, the UIS will appear as a single set of applications automating the information processing activities the user performs. All activities will involve a familiar set of information processing tasks, each with a standard interface. The system will create the illusion that all data is stored and processed at the user's location.

The UIS architecture will be distributed and layered. Eventually, all applications will be constructed and integrated using foundation software, including a data management system, common utilities, a user interface library, and network services. Each will conform with emerging industry standards for distributed information systems. Such standards facilitate the use of common tools such as spreadsheets and statistical packages, facilitate electronic data interchange with organizations outside the University, and promote independence from individual vendors.

Architectural Layers

<table>
<tr><td rowspan="5">Philosophy and Principles</td><td>Desktop/Client Interface Layer</td><td rowspan="5">Organizational Structure and Responsibility Model</td></tr>
<tr><td>Application Layer</td></tr>
<tr><td>Data and Document Management Layer</td></tr>
<tr><td>System and Network Management</td></tr>
<tr><td>Platform Layer
Hardware, System Software and Networks</td></tr>
</table>

The ***desktop/client interface layer*** addresses the end user's window to the world of information and information technology. An interface library will allow an application to interact with a variety of interface devices. Applications will obtain interactive input and present interactive output through routines in the user-interface library. Consistent use of library routines will standardize the look and feel across UIS applications.

The ***application layer*** addresses the software used to support processes within the University. A set of common utilities will provide applications with consistent services common to most UIS applications. Use of the common utilities by all UIS applications will help standardize the way users perceive and perform activities, and will reduce the effort required to create and integrate applications.

The ***data and document management layer*** will standardize the description, storage, and retrieval of all UIS data. All applications will access data through services provided by the UIS data management system. Furthermore, applications will use the data management system to determine whether a user has privileges to perform activities.

The ***system and network management layer*** will facilitate the management of the configuration of computer processors, networks, software, access devices, data storage devices, and other devices, like any other assets within the University. This layer addresses the functions of monitoring, scheduling, controlling, configuring, licensing, upgrading, problem solving, and recovering from abnormal conditions or failure.

The ***platform layer*** addresses the hardware, system software, and networking components of the architecture that support applications and user access to information system resources. This view of the information architecture provides an overall framework for the infrastructure necessary to accomplish the objectives of the other architectural components and provides a basis for determining hardware and system software acquisitions.

Until the architecture is fully implemented, existing systems and commercial packages will be evaluated on their ability to meet functional needs, their compatibility with UIS data management and network standards, and the ease of integrating them with the UIS interface library and common utilities.

Process View

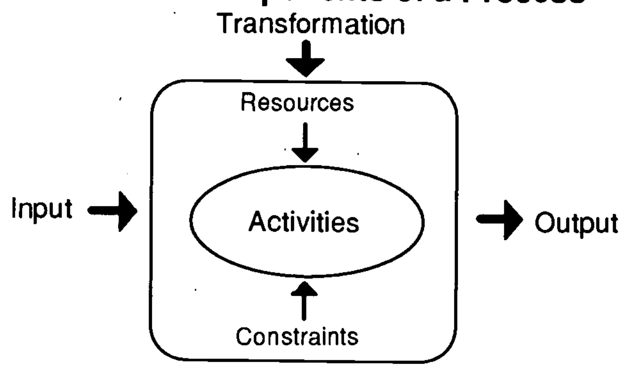
The University's work activities are currently organized around functional units, and the organization can be viewed as a series of vertical organizational structures. All activities are based upon this set of vertical compartments. The current information systems are also organized in this manner, as is all the information technology used to support the work of the University. This traditional organizational structure is not unlike organizational structures found in other industries. Work activities organized around such functional organizational structures are commonly characterized by inflexibility, unresponsiveness, the absence of customer focus, an obsession with activity rather than result, bureaucratic paralysis, lack of innovation, and high overhead.

Current processes, such as the procurement process, are long, convoluted assembly lines that are plagued by inefficiencies, delays, excessive paper, multiple levels of authorizations, errors, lack of access to information and customer dissatisfaction. Personnel are specialized, lack adequate access to electronic information and spend too much of their time on work flow and paper flow issues. Processes are badly in need of significant reductions to the costs of delivering services and radical improvements to the quality of the services delivered.

One of the ways to begin addressing these characteristics is by viewing the organization as a set of processes instead of individual functions. Once natural processes are identified, we can then focus on how well all activities in the process support the process outcome and how well the process outcome helps the University to achieve its goals and objectives. Morris and Brandon[2] state that process can be viewed as the essence of business. Not only is most work accomplished through processes, but a great deal of what differentiates organizations from each other is inherent in their individual work processes. This seems perfectly reasonable, since the same raw materials and human capital are available to every organization. Process is therefore one of the most important factors contributing to competitive advantage. However, despite the importance of process, it seems to have been largely ignored by management theorists and managers themselves.

A process is a logical and finite set of observable, interrelated (or hierarchical), work activities utilizing *input*, that when performed in a pre-defined series, produces *output(s)*. Processes have internal and external customers, and are independent of an organization's functional boundaries. Output is generated by a *transformation* of the input(s). As displayed in the figure below, activities are limited by the resources available to work activities, and the constraints imposed by mandates (policy, laws, and regulations).

General Components of a Process



The Information Architecture and Process Innovation Project identified four general clusters of processes (shown below) and defined the processes and components related to each cluster. The flow through a process represents the data and documents that enter into and exit from the activities for a process. Each of the processes have a set of sub-processes which act as threads of inter-related activities. These process clusters represent the workflow of the University and the services provided by administrative systems to support the mission of the University. The focal point of these processes is the set of customers that the process is intended to support. The data and document processing required to provide service to customers must be supported by the information architecture.

Process Clusters

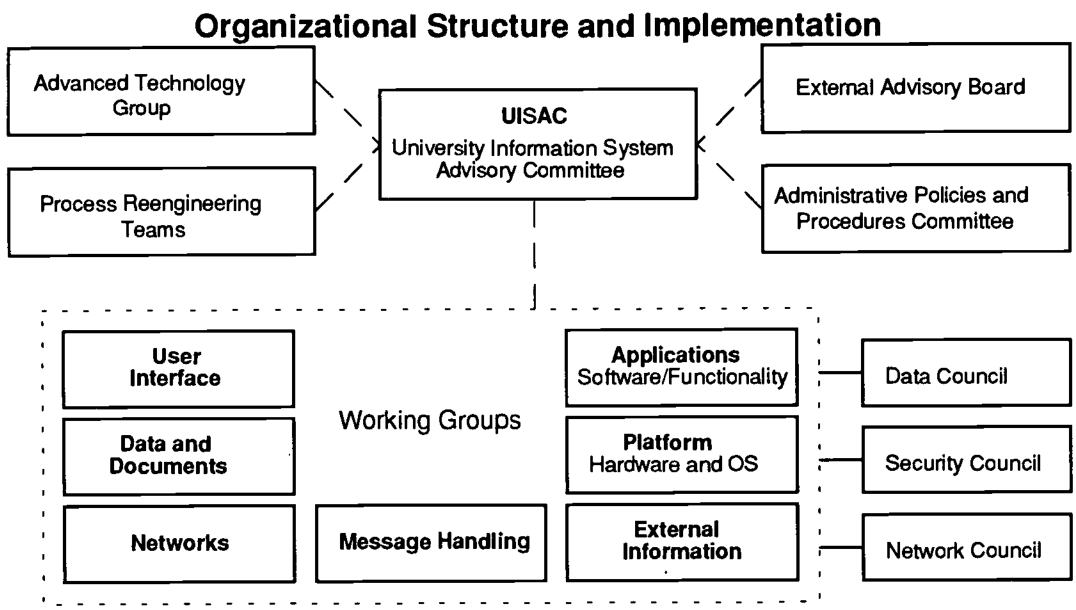
<table>
<tr><th>Educational Support</th><th>Research Support</th><th>Patron Support</th><th>Administrative Support</th></tr>
<tr><td colspan="4">Processes Flows Events Activities Outcomes
Customers
Constraints Resources Transactions Triggers</td></tr>
</table>

In their seminal work, Hammer and Champy[3] define reengineering as "the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service and speed." In order to make dramatic and meaningful improvements, an organization must identify and take a fresh look at natural "beginning-to-end" processes. Reengineering means starting over and asking why we do what we do. The purpose of process reengineering is to make the processes as streamlined as possible and provide a high level of service to customers. Part of the streamlining requires the use of information technology to permit sharing of data, parallel activities, increased responsiveness and improved quality.

Transforming the University

Three organizational units will play a prominent role in the implementation of the proposed architecture and process innovation initiatives. The first, an advisory committee, will be formed to provide overall guidance, direction, and priority setting. The second, an Advanced Technology Group, will be formed to investigate and implement emerging technologies, as well as to develop the technical capabilities for staff in AIS. Finally, AIS will assume the ongoing responsibilities of the Information Architecture and Process Innovation Project, ensuring that the architecture evolves and grows with changing technology and that the process reengineering efforts are related and refined.

The basic organizational structure proposed for policy formation and implementation of the information architecture centers around the creation of the University Information System Advisory Committee (UISAC). The UISAC will be composed of representatives from the University community, including academic units, administrative units, regional campuses, AIS, the Board of Trustees, and one outsider. The UISAC will report to the Senior Vice Chancellor for Business and Finance.



The UISAC will be given responsibility for creating an enterprise-wide business and information system strategy, and for making policy and funding recommendations for information system and reengineering projects proposed by academic and administrative unit design teams and by AIS. The rationale behind the formation of the UISAC is the strongly felt need for a consistent and coordinated approach to the University's administrative information systems and information technology infrastructure, and the policies, tools and techniques required for their development as well as the coordinated implementation of the information architecture and business process reengineering initiatives. The focus must be on technology supporting business and its customers. The UISAC will be a major agent of change and, as such, needs to create an environment of trust, and demonstrate effective planning and committed leadership.

One of the critical elements for any information systems organization in this age of rapid technological development is to develop and retain a staff trained in the use of new and productive technologies and techniques. The recommended approach to this issue is to form an

Advanced Technology Group whose function is to develop applications using the newest technologies and techniques available on a prototype scale. This group could attract faculty and advanced students to work with AIS personnel on projects that are developmental in nature but have a potential payoff for the University. Such a group could also begin to attract external funding as well as become a beta site for hardware and software vendors.

The information architecture will be implemented through a project approach. Projects will be proposed by project design teams that are formed within the administrative and educational units of the University. The design teams for projects may be reengineering teams, or they may be smaller incremental improvement project teams. The teams will propose projects in accordance with detailed guidelines that ensure they will be aligned with the information architecture. This project approach is preferred over a master plan approach in order to avoid the problem of plan obsolescence typically associated with large master plan implementations.

The project design teams will present their project proposals to the UISAC, which will review the proposed projects and recommend revisions as necessary. Project proposals submitted for funding will be described using a pattern language and will contain an environment section, a functional section, a performance section and a budgetary section. The decision to fund projects should be based largely on their adherence to the architectural patterns.

This approach is similar to that taken at the University of Oregon and found to be quite successful in designing and building the campus over the last 20 years[4]. The following set of principles is modeled after the Oregon Experiment:

- Organic order: The planning and implementation of the information architecture will be guided by a process that allows the whole to emerge gradually from local implementations, guided by the proposed information system philosophy and structure.
- Participation: All decisions about what will be built, and how it should behave will be in the hands of the users at various levels. This is based on the assumption that users help shape the environment, know their needs best, and can define the qualities of the information system required to satisfy their needs.
- Piecemeal growth: Piecemeal growth hinges on dynamic and continuous growth. Therefore, the UISAC will distribute funds for small, intermediate and large projects equally. Funds must be made available without an overwhelming amount of specific, low level details, since resources consumed attempting to determine low level details could be better spent on implementation.
- Patterns: All design and implementation will be guided by a collection of communally adopted design principles, called information processing patterns, that will guide the design of everything.
- Diagnosis/Evaluation: The well being of the architecture and the envisioned information system will be protected by an annual diagnosis/evaluation system that will explain, in detail, which information system activities are alive and which are dead, at any given moment in the history of the system.
- Coordination: The slow emergence of organic order in the whole will be assured by a funding process that regulates the stream of individual projects put forth by users. The use of a standard template to fund projects, describe projects, describe patterns of information processing, perform diagnosis and estimate costs will aid in prioritizing projects.

The design team concept takes advantage of the expertise available across the University and permits multiple views of the information system project, consistent with the notion that partnerships produce a better design in a more cost effective manner than if any one of the team components attempted to implement the project alone. It also leverages the knowledge of the unit's needs, the specific knowledge of the local unit's information system and the knowledge that AIS personnel has of the University's central systems.

There is a need to define the roles and responsibilities of AIS personnel, Information Systems (IS) owners and IS coordinators, local unit technical personnel, end-users and management. One scenario for this definition of roles and responsibilities is:

- Managers have the responsibility to assemble design teams for projects and to provide release time for design team members to work on the information system projects being proposed.
- End users have the responsibility to identify requirements for information system services, products and features known to the design team in a timely manner and in a form and format that is understandable to them.
- Local unit technical personnel, IS owners, and IS coordinators act as information system design team members, local system implementors, local system managers, local application developers, and local end user consultants and trainers.
- AIS personnel act as technical consultants and design team members for implementing information system projects. AIS may also act as application developers for both the client and server sides, as well as act as IS suppliers and IS operators.

All proposals must indicate what University and other standards are being utilized as part of the project. If proprietary products are being used which do not adhere to an open systems architecture as proposed by the IA, then a rationale must be provided as to why a closed system product or approach is being used.

Summary

The Information Architecture and Process Innovation Project determined that the current University administrative process environment can benefit from drastic improvements in quality and efficiency by employing the methods available through process reengineering. The project also determined that modern information processing technologies and systems are required to support the flexibility, rapid response time and information access requirements needed by end users to perform their work, deliver quality services and make informed decisions.

The implementation strategy is driven by business process reengineering projects, but, at the same time, these new system implementation projects must be balanced with projects to improve access to information using the current systems. The implementation strategy is based upon process owners, system owners and end user initiatives for projects that follow the architectural principles and the natural relationships between activities of a process and the inter-relationships between and among processes.

---

[1] Christopher Alexander, The Oregon Experiment (New York: Oxford University Press, 1975

[2] Daniel Morris and Joel Brandon, Re-engineering Your Business (New York: McGraw-Hill, Inc., 1993) p. 38

[3] Michael Hammer and James Champy, Reengineering the Corporation: A Manifesto for Business Revolution (New York: Harper Business, 1993) p. 32

[4] Christopher Alexander, The Oregon Experiment (New York: Oxford University Press, 1975)

A Distributed Computing Architecture and Transition Strategy

Joan Gargano
University of California, Davis
Davis
California

The UC Davis Information Technology Strategic Plan acknowledges the value of a distributed client/server computing environment to the campus and the need to provide a supporting infrastructure. However, the decentralization of computing also requires a complimentary level of campuswide standards, centralized services and support infrastructure to ensure a reliable, coordinated and interoperable campuswide computing environment in which institutional information is readily accessible as well as secure and well managed.

The UC Davis Information Technology Division demonstrated its commitment to open distributed systems by creating a new department to serve as a campus resource and to provide the staff to develop new infrastructure services. As its first order of business, the Distributed Computing Analysis & Support Department spearheaded efforts to transition existing IT systems to open systems in the Summer of 1993. An open invitation was sent to the campus to attend a two day seminar on Project Athena given by Jeff Schiller from MIT. This conference provided a practical overview of the issues of distributed computing in an academic setting and provided a forum for discussing the key issues associated with implementation, operations and maintenance of systems in this environment. It also served as a way to coalesce a broad based group of technical staff from many departments to work on a campuswide statement of the distributed computing architecture.

This paper provides an account of the U.C. Davis experiences in defining the distributed computing architecture at U.C. Davis, developing a transition plan for existing systems and the results of the first year of implementing new infrastructure systems as part of the transition.

The University of California, Davis (UCD) has established a new department and distributed computing architecture statement to begin creating the infrastructure and support needed for a fully distributed computing environment. The new department, Distributed Computing Analysis and Support (DCAS), is part of Information Technology and is the result of a campuswide strategic planning process and reorganization to meet the changing needs of the campus community. This article describes the role of the new department, the distributed computing architecture that has emerged and the first year of experience implementing new systems within this framework.

Strategic Planning and Reorganization

The U. C. Davis strategic plan, completed in December 1992, "Creating a New Information Technology Reality: Strategic Directions for the Campus," defines new directions for computing which we expected to affect our strategy for the deployment of technology, but which also included several tasks related to funding models that, as we found out later, had a significant impact on the way we would approach the development of our distributed computing architecture. The components of the plan that affect our decisions the most are:

- Establish a dynamically-responsive structure to support a distributed information technology environment, blending centralization and decentralization in an effective manner. Strengthen the Information Technology (IT) organization's role in centralized coordination of decentralized resources. Provide centralized leadership and support for campuswide activities which are common to multiple units or constituencies.
- Assist units in the development of technology plans, and develop campuswide plans to complement and enhance unit plans.
- Advance the use of collaborative information technologies, including electronic mail, telecommunications, bulletin boards, conferencing systems, and public information networks.
- Provide a base level of service, funded centrally by a campus allocation "off the top," at no visible cost to the individual user or department, including support for student instruction and access to administrative systems and databases required to perform one's assigned duties, e.g., student, financial, payroll, personnel systems.

The tasks listed in a strategic plan should not dictate any particular organizational structure and this was the case at UC Davis. Initially we envisioned these tasks being handled within existing organizational units charged with academic or administrative support. However as tactical plans were developed, it was decided that a dedicated unit was needed to provide the level of service and attention this new infrastructure requires. A small unit was created to focus on these issues.

Distributed Computing Analysis and Support

The new group, Distributed Computing Analysis and Support, is staffed by experienced programmers and analysts from many areas of Information Technology. This group can be described as a combination of two categories of advanced technology groups described by Gartner Group's Bill Caffery and summarized by Megan Santosus in a 1994 CIO article[1], "The Guerrillas" and The Navigators". Their role is to identify technologies that can be leveraged throughout the University, but they focus on serving departments with solutions that work immediately. Initially, the staffing of DCAS was based upon the interest of the staff members to work in the area of distributed computing infrastructure. When creating the group, we placed the highest value on experience in creating campuswide solutions and less on the specific area of expertise, such as academic or administrative information systems. In retrospect, we've found that the range of experience found in the DCAS team is important to creating these services and their rapid deployment.

The mission of DCAS includes the following:

The Distributed Computing Analysis and Support group surveys technologies relevant to mission critical information systems in the context of the Information Technology Strategic Plan, designs new systems of widespread campus use and assists in their transition to fully operational support systems. Major areas of activity include:

- Coordinate the implementation and interoperability of campuswide distributed computing infrastructure and support systems.
- Identify the critical components of networked systems and their relationships and develop reporting, management and problem resolution systems to ensure an optimum balance between reliability, performance and cost.
- Design, plan and support networked systems architectures and applications.
- Research and report on advances in technology that will have a substantial effect on campus computing services in the next three to five years.
- Research key computing developments and chart strategic directions for network systems architectures.
- Design, test and implement new systems in established and newly identified strategic areas as appropriate.
- Coordinate the technical strategic planning process for the Campus with full participation by broad-based campus constituencies.

It is important to note that the DCAS mission specifically refers to implementation , not just design and planning. From the beginning, members of the team have taken responsibility for the smooth operation of production systems through first hand

[1]Megan Santosus, "Down to Earth," CIO, April 15, 1994, p 54.

experience, no matter which department might ultimately manage the services. The group feels that this experience is important to the creation of robust services that are easy to manage.

DCAS also strives to obtain full participation by broad-based campus constituencies in the technical strategic planning process to ensure technical directions identified through this process are relevant to the computing needs of the campus and adopted campuswide. This was the approach that was taken when DCAS initiated the strategic and tactical planning for the campuswide distributed computing architecture.

A New Direction

The strategic value of open systems to Universities has been clearly demonstrated in the use of open internetworking protocols and this concept now generally extends to all aspects of computing. The commitment to open systems within a University is not a difficult decision to make. Deciding how to make the transition is much less clear.

As its first order of business, the Distributed Computing Analysis & Support department spearheaded efforts to transition existing IT systems to open systems in the Summer of 1993. An invitation was sent to the campus to attend a two day seminar on Project Athena given by Jeff Schiller from MIT. This conference provided a practical overview of the issues of distributed computing in an academic setting and provided a forum for discussing the key issues associated with implementation, operations and maintenance of systems in this environment. It also served as a way to coalesce a broad based group of technical staff from many departments to work on a campuswide statement of the distributed computing architecture. This resulted in a working document, "The UC Davis Distributed Computing Environment 1994 - 1995" which provides an overview of the existing computing environment and charts the course for then next two to three years.

The architecture chosen by UC Davis is very similar to that reported by Arizona State University in the Summer 1994, CAUSE/EFFECT.[2] The architecture is summarized in the following table.

[2]Neil Armann, L. Dean Conrad, Darrel Huish, and Bruce Millard, "Developing a Distributed Computing Architecture at Arizona State University," CAUSE/EFFECT, Summer 1994, p 12.

Architectural Component	Standards	Initial Products	Future Products
Distributed Logic	OSF/DCE RPC	Use proprietary RPC from DB manager or TP monitor until 1995	OSF/DCE RPC
Name Service	DNS - Domain nameservices, CDS	DNS generally available through hardware and software vendors	DNS, CDS
Print Service	OSF/DCE Print Service	TCP/IP, LPR/LPD	DME Print Service
File Services	OSF/DCE Distributed File System	Transarc Andrew File System (AFS)	OSF/DCE Distributed File System
Time Service	OSF/DCE Time Service	Internet Network Time Protocol	OSF/DCE Time Service
Security	OSF/DCE Security Service, Enigma Logic one time passwords, RSA public key	Version 5 Kerberos server, ftp Software, Inc. DOS/MS-Windows clients. Enigma Logic gold card.	OSF/DCE Security Services, Enigma Logic gold card, RSA public key.
Operating System/Graphical User Interface	DOS/MS Windows, Mac/System 7 UNIX/Motif	DOS/MS Windows, Mac/System 7 UNIX/Motif	
Database Management	SQL	Oracle, Sybase	Oracle, Sybase
Electronic Mail	Simple Mail Transfer Protocol (SMTP) with MIME extensions	SMTP	SMTP with MIME extensions
Campuswide Information Systems	Gopher and World Wide Web	Gopher and World Wide Web	World Wide Web
Calendaring	None	Departmental use of Oracle Office, Microsoft Office and Meeting Maker	None

Working within the new framework, we then reviewed the OSF/DCE products for distributed computing and the Project Athena environment to begin the development of a tactical plan for the deployment of services. It was immediately clear that the OSF/DCE products were incomplete and too immature for deployment in a production environment. Our alternative environment, Project Athena, provided the most cohesive interim solution while OSF/DCE products were under development, but the tight integration of services did not accommodate a staged campuswide deployment. Given the lack of a comprehensive suite of products, we then decided to develop a tactical plan that would provide solutions to immediate problems but position us well in our transition to standard products. Our analysis of the standards and systems in place showed that in all of the distributed infrastructure development, databases of information on networked entities, people, devices and services, were central to the interoperability of infrastructure services. We concluded that efforts which began to assemble this data, create a uniform name space and automated its creation would be the best way to prepare for the development of new services.

We then began evaluating our highest priority for service development under this new strategy. Our most pressing problem during this time was a side effect of one of the strategic plan tasks:

- Provide a base level of service, funded centrally by a campus allocation "off the top," at no visible cost to the individual user or department, including support for student instruction and access to administrative systems and databases required to perform one's assigned duties, e.g., student, financial, payroll, personnel systems.

This task required the accommodation of every student, faculty and staff member on centralized machines in accessing a base level of service including electronic mail, the campuswide information system and administrative services. Access to many of these services will also require a higher level of security such as that provided by Kerberos. The type of data and functionality needed to support these services had already been refined in one environment through Project Athena. The Moira database system provided us with a good example of how this type of service might be deployed. Using this model we began the development of our local system with the following design goals:

- All University affiliates will have access to computing resources and a base level of services as soon as administrative records reflect their association with the University.

- The system must be platform independent and support the creation of accounts on any computer system.

- The system must be as close to paperless as possible.

- The database must be automatically maintained through normal administrative processes.

- The system must be designed in a highly modular fashion to accommodate the transition to DCE products as they become available.

We chose Oracle as the relational database management system for the new directory service because, as an SQL database it conforms to our stated architecture, UCD has a campuswide site license for the product and there is a great deal of local expertise to support it. However, a conscious decision was made to use it as a database engine only and not to use any vendor specific features that might restrict porting the application to another platform in the future.

Work began on August 12, 1993 with two full time programmers on the project and the need to have a baseline system operational by September 14, 1993. The use of Project Athena's Moira as a model for database design and functional requirements helped us minimize the early phases of planning and design and move directly into development. A new database was created and populated with data from the Student Information System and the Payroll/Personnel System. The new database was designed by Larry Johnson, a programmer with extensive experience working with our administrative systems. Working closely with systems administrators for the U. C. Davis Unisys A11, the U. C. Davis Sequent S2000 and the U. C. Office of the President IBM 3090, Larry coordinated the creation of data extracts and file transfers between legacy systems and the Oracle database. In parallel, Dan Dorough, a programmer with extensive experience working with Unix based networked systems, began working with systems administrators for the academic systems to automate account creation by extracting data from the Oracle database and linking it to the account creation mechanisms on Unix and VMS machines. The two systems were then merged together and on September 14, 100 medical students registered their accounts online using the new system. By October, the new system was managing account creation on all of the Information Technology and Electrical and Computer Engineering central systems and only special requests for accounts required a written request and manual data entry.

The Year End Report

Goal: Complete the tasks outlined in the strategic plan

Last year, during the first week of classes, we automatically registered 200 accounts and continued to add a total of 12,000 accounts during the year. This year, we automatically registered 1,780 accounts during the first week of classes and we are continuing to see a steady rate of account creation at about 300 accounts per week. All of these accounts are accessible within 24 hours of activation and provide our customers with online access to electronic mail through Pine or exchanges using the Post Office Protocol (POP), Gopher, the World Wide Web and all other standard Internet utilities. The account creation system will soon be tied to our Student Information System and is planned for use by the Financial Information System under development. The new system is a success, providing a mechanism for accomplishing the strategic plan task of providing a base level of service to all students faculty and staff.

Goal: Create an infrastructure service aligned with developing standards

The decision to create directory services that support the creation of infrastructure services has also proved successful. The structure of the database system is supporting a proprietary method of account creation, but it also allows us to extend its support to other infrastructure services such as Kerberos and eventually DCE Security Services, mail routing through aliasing and online directory services through Whois, Gopher and the World Wide Web. Several processes that were previously manual have been automated providing cost savings and improved customer service.

During the past year we have used the same model to create a database of network devices that automatically generates configuration files for the domain name system. It is expected that this database may eventually be extended to support version control services for client applications.

Conclusion

The creation of a separate unit within Information Technology to support the distributed computing infrastructure has worked well at UCD and provides us with the resources to respond quickly to the needs of the campus. Our tactical plan of creating infrastructure services with the tools available today, but aligned with emerging standards is allowing us to move forward in a staged deployment, but without wasted effort. However, the most significant contribution to our ability to respond quickly was the mix of experience found in the DCAS team. The team working on this first directory service project combined the talents of a staff member with extensive experience supporting centralized networked systems and a staff member with extensive experience supporting administrative applications on mainframe systems. In retrospect, it was this combination of talents that ensured the success of the project. While neither programmer had previously worked with Oracle, Project Athena or OSF/DCE products, based on their years of experience in providing central services, they were able to quickly understand the technical details of those systems and assess their role within a developing distributed computing environment. Even more important was their understanding of existing computing systems and their ability to work with systems administrators to link the old systems with the new. Finding staff with this combination of experience, flexibility and the ability to bridge the established and emerging technologies is the critical success factor to the smooth transition between centralized and distributed computing environments.

GETTING THE RIGHT FIT:
INSTITUTIONAL DOWNSIZING WITHOUT CAPSIZING

Timothy J. Foley
Lehigh University
Bethlehem, PA 18015
E-mail: tjf0@lehigh.edu

ABSTRACT

Downsizing and rightsizing are buzzwords that have gained much acceptance in current computing literature. Can mainframes be replaced with high performance clusters of workstations? That is a question many computing center directors are asking, as high-end workstations eclipse the capabilities of traditional mainframes. At Lehigh, we have found that the answer, at least for us, is "Yes".

Lehigh University has undergone a dramatic change in its computing environment over the last three years. Starting in 1991, Lehigh eliminated its three academic mainframes, introduced more than 150 workstations, installed compute and file servers, and introduced software to make this combination act as a unified computing system. The result has been much greater computing power, all financed from existing funds.

LEHIGH OVERVIEW

Lehigh University is an independent, academically selective, comprehensive university (4500 undergraduates and 2000 graduate students) which has been described by Lehigh's president as "small enough to be personal, yet large enough to be powerful". Lehigh has four colleges: Arts and Science, Engineering and Applied Science, Business and Economics, Education, and also 35 research centers and institutes. In 1990, Lehigh's primary computing environment consisted of a CDC Cyber Model 180 purchased in 1985, a DEC Vax 8530 purchased in 1987, and two IBM 4381's purchased in 1986 and 1988. The IBM mainframes provided administrative support and support for Lehigh's Campus-Wide Information System (CWIS). Lehigh had also implemented a small workstation site with Sun workstations and provided over 300 microcomputers in public sites. By 1994, this environment had been transformed into a mainly Unix environment supporting IBM RS/6000s running AIX with three of the existing mainframes removed and over 150 RS/6000s placed in public sites. Lehigh has installed three distinct server clusters to serve the requirements of the user community. A cluster of RS/6000s has been designated as compute servers and consists of two IBM RS/6000s, models 990 and 580. Another cluster is designated as network servers which support the needs of our CWIS and consist of IBM RS/6000s models 990, 980, TC10, and 560. Three RS/6000s have also been designated as AFS file servers to provide file services for the 150 distributed workstations. Along with this increased computing power, Lehigh's microcomputer support requirements have continued to grow with over 350 public machines and thousands of other microcomputers appearing on campus. It should also be noted that some administrative applications have been moved to RS/6000s though most of them are still running on the remaining IBM 4381.

Lehigh's high-speed networking activities have also been an integral part of our downsizing plans. In 1985, Lehigh installed a totally digital, non-blocking PBX providing voice and low-speed data connections (19.2 Kps). Data connections were provided everywhere including classrooms and student living units. Lehigh has since expanded its networking capabilities to include a high-speed fiber optic backbone supporting FDDI data rates of 100 Mbps. All of the Computing Center's sites are connected to the high-speed backbone along with most of the campus buildings. Most of the subnets connecting to the backbone are 10 Base-T ethernets with the exception of public sites. These were the first sites networked and were wired with thin ethernet. Connections to the residence halls were started in the fall of 1993 with plans for completion by fall, 1995.

CATALYST FOR CHANGE

The major catalyst for changing Lehigh's computing environment was the development of a five-year strategic plan for computing and communications. This plan was developed in the spring of 1990 and finalized in 1991. The plan called for the removal of all academic mainframes. It also specified the transition to a Unix environment. At that time, the Computing Center was supporting the following operating systems: CDC NOS/VE, DEC Vax VMS, IBM VM/VSE, IBM VM/MUSIC, and the Apple Macintosh and MS/DOS Windows environments. The plan recommended a phased approach with the removal of the academic mainframes and an upgrade to the Administrative IBM 4381. Administrative applications were to be migrated to the workstation environment at a slower pace than were the academic applications. The Vax 8530 and CDC Cyber 180 were to be replaced by a compute server. RISC, vector, and parallel machines were investigated as possible replacements. Another key aspect of the plan was the requirement to implement a common database throughout the campus. For this, the Oracle database was chosen and was used as the basis for replacing the CWIS running under the MUSIC operating system on the IBM 4381 mainframe. Other key components of the plan were the funding of distributed servers for academic departments (with research departments providing their own funding) and the financing of all these changes with existing funds.

IMPLEMENTATION STRATEGIES

A typical university implementation strategy was taken to plan for the removal of Lehigh's mainframes (i.e., form a committee with an "interesting" acronym). The acronym chosen was CINC (Computer Intensive Needs Committee), pronounced "sink", to enable Lehigh to downsize without capsizing. Often during this process we have had "CINCing" feelings so the acronym seemed very appropriate. The committee was composed of three members from the four colleges and the Computing Center. The first task of the committee was to survey our existing users. Users were queried on software applications, current satisfaction with our systems, possible conversion problems, and the characteristics they felt were needed in a new system. The survey indicated that 84% of the mainframe usage was research

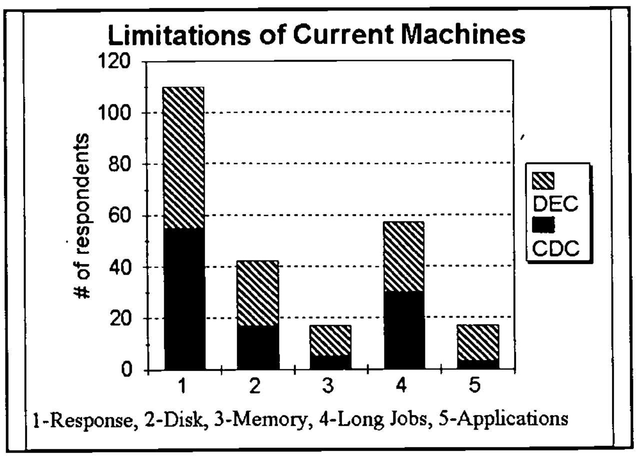


Figure 1

related. Figure 1 illustrates what users perceived as the major limitations of the current machines with response time being the largest problem. Figure 2 shows that users wanted at least 10 times the power of the CDC Cyber with computing speed and better graphics rated as very important features.

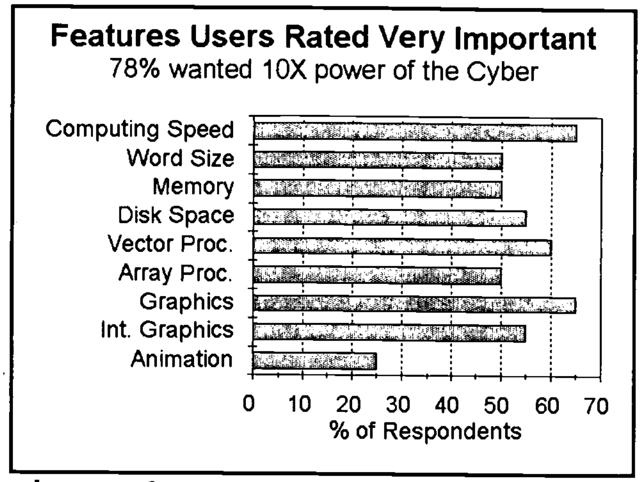


Figure 2

Besides surveying the users of the system, the CINC committee also held an open faculty meeting (with 44 computing intensive users attending) to discuss the five-year plan, the current hardware options, and users' current needs. The result of these activities was a report prepared for the Computing Center Advisory Committee (CCAC). The report stated that: the current mainframes were saturated; researchers had already begun moving to a workstation environment; and high-speed network expansion was critical to the needs of computing intensive users. The final recommendation was that since about 90% of Lehigh's computing power was being consumed by 10% of the users, Lehigh should attempt to provide a computing solution to satisfy this 10%.

The development of a request for proposals (RFP) was the next step in the process of determining an appropriate replacement system. Eleven vendors were originally contacted representing RISC, massively parallel, and vector architectures. Ranking of vendors was based on system capacity, application software, system management, end-user software, and three year costs. Software availability was a major requirement and eliminated three of the vendors. Of the remaining vendors, seven responded to the RFP and were asked to run a set of representative benchmarks. The benchmarks were developed from representative user jobs and ported by the vendors to run on their systems.

The choice of vendors was narrowed to three - HP, CDC, and IBM - based on both the benchmark results and our rankings. During this time, negotiations were also occurring concerning a large acquisition of workstations. An opportunity to work on a development project with IBM was the key factor in Lehigh's final decision to go with IBM and their RISC platform for our replacement solution. The key issues were that the benchmarks were inconclusive when comparing the top three vendors and that IBM offered the best deal financially.

MAJOR TRANSITION ISSUES

CWIS Migration

One of the first issues in replacing Lehigh's mainframes was the migration of Lehigh's current campus-wide information system which had initially been developed on an IBM 4381 running the MUSIC operating system. In 1989, it was decided to move to another platform and work was begun to port the software. A major challenge was designing the system to remain consistent with our existing interface while planning on moving to a client/server platform and an eventual graphical hypertext interface. The IBM 4381, which utilized flat ASCII files that were accessed by a control file, was replaced with a distributed database model using the Oracle database management system on a cluster of RS/6000s.

Lehigh's CWIS provides the following applications to a user base of over 7000 active accounts: electronic mail, bulletin board and conferencing facilities, access to national and international networks, on-line forms and survey processing, fax delivery and retrieval, and an access point for library services. This system is widely used by the campus with 95% of the community using the system on a regular basis. Lehigh's goal of decoupling the user interface from the actual database was accomplished to some degree and recent developments have allowed the client portion of the interface to run using Mosaic and World-Wide Web.

Another important design feature is the concept of "distributed services" encompassing both the management and the location of databases and applications. Data management activities are the responsibility of individual topic coordinators. Topics on the CWIS are fed and nurtured by, and the responsibility of, the information providers. Another aspect of distributed services is the ability to access designated hosts or data directories. Distributed applications are also being explored to allow selected applications to transparently run on another host. The overall goal is to move from a tightly coupled cluster supporting all available services to a more diffuse system. This will allow data and processing power to be distributed to the most appropriate location.

Software Identification and Licensing

Software identification was first done by surveying users and contacting vendors on availability of software. Committees were also formed to identify software which would increase the functionality of the workstations in the distributed environments. Reports were prepared on: desktop environments, graphics/CAD software, text processing, database and spreadsheets, mathematical software, scientific libraries, and statistical software. These reports resulted in obtaining a number of attractive site and floating licenses. The major site licenses obtained were Maple, NAG, and BMDP. Floating licenses were obtained for Matlab, Island Graphics (Write, Paint, and Draw), AutoCAD, WordPerfect, and Lotus. Another major agreement that made the IBM RS/6000 platform very cost effective was IBM's HESC (Higher Education Software Consortium) program. This program dramatically reduced Lehigh's operating systems cost. Lehigh's overall software costs for the CDC Cyber and DEC Vax was $150,000/year, while it is only $130,000/year for the RS/6000s. This is in contrast to the first estimates that had the software costs in the distributed environment at over $250,000/year. Figure 3 illustrates that Lehigh's major savings have been in the area of operating system costs while application costs have been very similar.

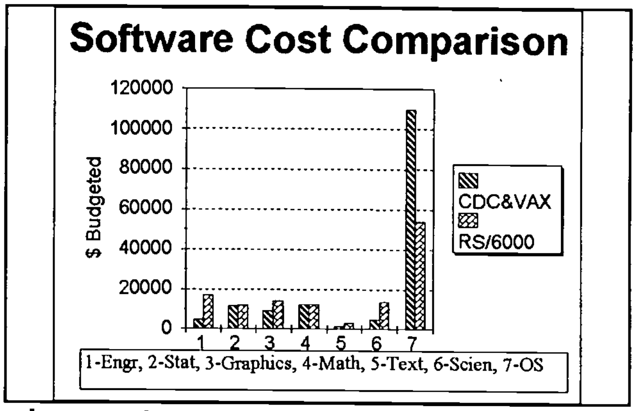


Figure 3

User Training and Documentation

The overall transition to the new environment required a great deal of additional staff and user training and a revamping of practically all the existing documentation. User training was accomplished in a variety of ways with special seminars developed to deal with users' transition problems. These were hands-on sessions which dealt with the new operating environment, and also special conversion problems that needed to be addressed in moving from the NOS/VE and Vax operating systems to Unix. The move from MUSIC to Unix was easier since we kept the same interface in moving from the IBM 4381 to the RS/6000s. Figure 4 illustrates the growth of our Unix-related seminars over this timeframe.

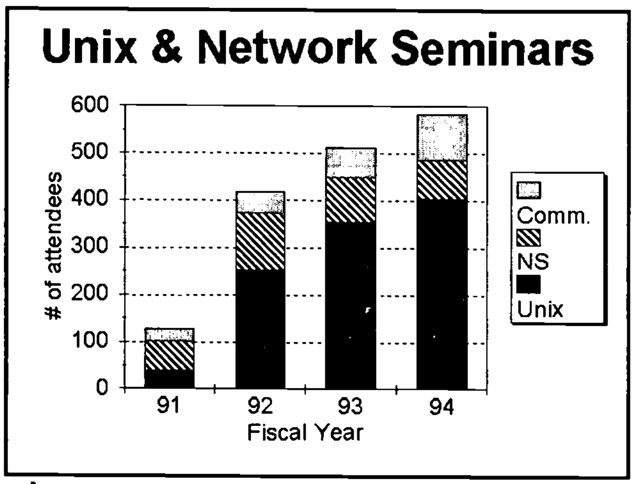


Figure 4

User documentation was and still is a big issue that needs to be addressed in the workstation environment. Essentially, all existing documentation had to be revised or rewritten during this transition period. The Computing Center investigated a number of tools such as IBM's Infoexplorer, utilizing Unix man pages, and possibly creating a searchable WAIS server to provide on-line documentation. Initially, a simple text help file was placed on-line which listed all available commands and where to go for help on running them. The Computing Center has also started to provide all its documentation on WWW; User's Guides, Technical Bulletins, and seminar handouts are currently being converted into HTML documents.

Tape, Program, and File Conversion

The conversion of tapes, programs, and files presented another interesting problem for the Computing Center. Hundreds of tapes resided in the machine room and many of them had been in the tape library for years. Each tape user was sent a list of his or her tapes and also was contacted by students hired to work over the summer to assist with tape conversions. This process went better than expected and many users determined that the data that they had stored for years was really not worth saving.

Program conversion was handled by providing hands-on conversion training sessions and by individual consultation. Back-up sites were arranged to assist with tape and program conversion for users who had problems getting the conversion done. Files were migrated automatically for CWIS users while Cyber and Vax users issued a command that transferred files to individual Cyber and Vax directories on the compute server cluster.

Hardware Maintenance

A major cost issue associated with distributing hundreds of workstations throughout the campus is how to maintain these devices. The cost to provide reasonable maintenance for all of these devices was double our existing budget. Vendors were contacted and proposals were received from each. After analyzing the costs, it was decided to provide self maintenance through our Computer Store with a parts contract from a parts supplier. Critical machines such as the compute servers, AFS file servers, and CWIS servers were kept on vendor maintenance. Figure 5 shows that the overall

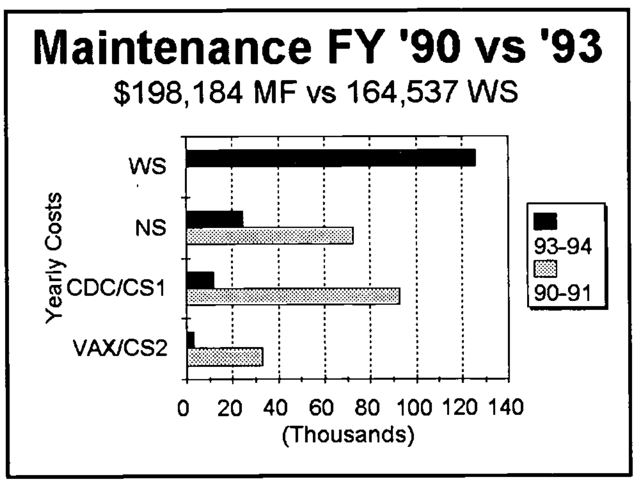


Figure 5

hardware maintenance costs have been reduced by over $30,000/year.

Distributed Support Issues

Once the university entered the agreement to receive the 150 RS/6000 workstations, space needed to be found to house them. Departments made proposals outlining how they would use the workstations, the space they had to house these workstations, and their software requirements. This resulted in the creation of 12 semi-public sites which were to be available to the public when not in use by the departments. Procedures were established for providing support for these sites with each site having a department contact and Computing Center support personnel from Systems Programming, User Services, and Operations. Meetings were initially held to establish the guidelines for support of these sites, and software requests were directed to Lehigh's existing software committee. A minimal set of documentation was provided for each site with the emphasis placed on using on-line documentation for most tasks.

SUCCESSES AND PROBLEMS

The CWIS migration turned out to be a success and problem at the same time. The transition to the new environment went very smoothly with usage growing to new levels. The problem has been that the continual growth has put added strain on the system which has been expanded to three servers and exceeding 10,000 logins a day (see Figure 6). The redesign of the system utilizing Oracle did make the transition to gopher and WWW much easier. Our current implementation allows the CWIS to be accessed through WWW with authentication, but most of the campus is still accessing the system in text-mode using TCP/IP or serial protocols. Microcomputer pop mail programs and workstation mail are encouraged as users become connected to the backbone network. It is hoped that some of this, along with the transition to Mosaic/WWW, will reduce the load on the CWIS servers.

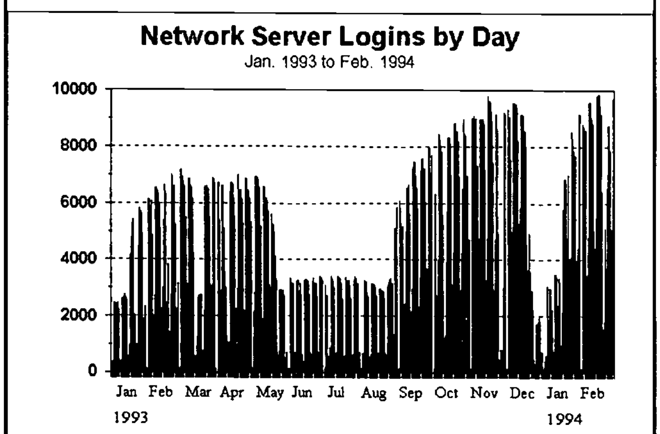


Figure 6

The implementation of the compute servers and the workstations throughout campus has dramatically increased computing usage. CPU usage was compared from the month of May 1991 to the month of May 1993; it had increased from 63,000 CPU minutes to over 1,800,000 CPU minutes. Many researchers have been able to complete tasks that were not possible in the past using existing mainframes. Another part of the equation, however, is that the 90/10 rule for usage which was referred to previously had changed to a 95/5 rule with 5% of the users using 95% of the CPU (see Figure 7). Some reasons for this will be discussed in the next section which shows that the nature of the work being carried out on the workstations has changed from what was previously done on the mainframes. Another success and also a problem has been the use of the AFS file system, which has provided a common userid and password for all of our systems along with support for distributed files and security features. Some serious AFS reliability problems were encountered during the last few years which have been very frustrating to our users.

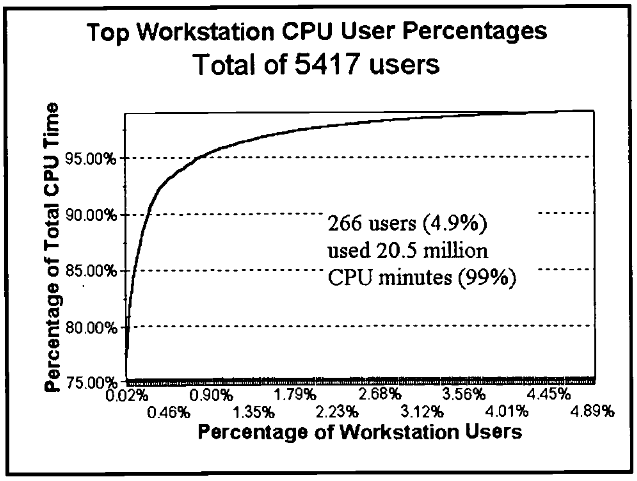


Figure 7

The implementation of the distributed environment has also dramatically increased our disk and tape management problems. As workstation use has grown, disk requirements have increased at a much higher rate than in our previous mainframe environment.
System backups are taking an inordinate amount of time and a project is currently underway to examine hierarchical backup systems. During the transition, our overall disk capacity went from 35 gigabytes to over 100 gigabytes without any sign of this demand stabilizing.

User Survey Summary

To determine the extent of usage and also to find how users felt about the new environment, a survey was distributed to all workstation users. The survey results indicated that 54% of the respondents were from the Engineering College, 31% from the Arts and Science College and the remainder from the Business and Education colleges. The survey results showed that most users were satisfied with what they could do on the workstations. Users were queried on their satisfaction in 16 areas relating to the new environment; in all areas there was a surprising level of satisfaction with the workstation environment (see Figure 8).

The lowest-rated factor was response time, with 68% of the users satisfied or very satisfied with response time. The survey also found that there was a major shift in how the new machines were being used. In the past the major usage of our mainframes was for research. In the distributed environment only 33% used the systems primarily for research while 54% used them for communication purposes and 10% for course usage.

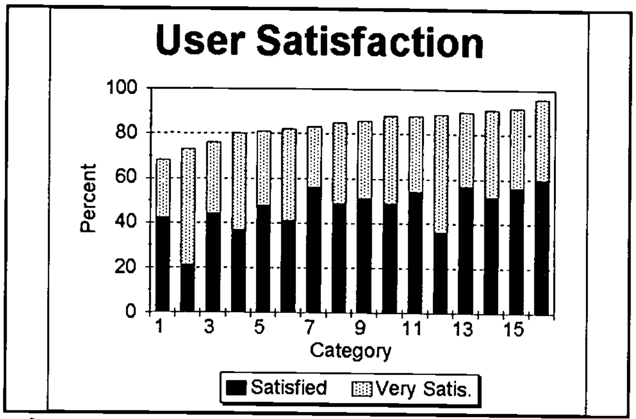


Figure 8

KEY FACTORS AND SUMMARY

User involvement was a major factor in accomplishing the transition to a distributed environment. It was crucial to keep the user community informed and involved in the decision making process. Lehigh was also under strict financial constraints so everything had to be done within the context of existing budgets and the reallocation of funding from the former mainframe budgets. The creation of a RFP and user benchmarks also helped to clarify computing needs. The benchmarks helped in eliminating some vendors and in determining that there was little distinction between the performance of the three top companies. Another key factor was the establishment of very aggressive timelines and goals. Often these seemed unreasonable, but they were being driven by our financial constraints. Finally, a willingness to compromise in coming to a decision to take part in a development project allowed this transition to be made within our existing budgets.

In summary, Lehigh was able to remove three mainframes within a nine-month timeframe, deploy 150 workstations in 18 months, and increase its overall computing power by a factor of 500. The result has been much greater computing power, all financed from existing funds. Users have access to more computing power, better interfaces and more advanced software, thanks in large part to the savings realized from eliminating the overhead and maintenance on the former mainframes. Administrative Computing is still running most of its applications on the IBM 4381. Admissions and Development software are running on RS/6000s but in general the administrative transition is going to take considerably longer than the academic transition. Microcomputers are also still a very major factor in Lehigh's computing structure. They have not been reduced and replaced by workstations (which was suggested in our five-year plan). A new five-year plan is currently under construction which will stress the enhancement of the new computing environment along with goals to further incorporate this technology into the educational environment of Lehigh's students.

Rightsizing - Changing the Culture

Sue Borel and Natalie Vincent

Syracuse University
Syracuse, New York

Syracuse University has embarked on a project to move all of its administrative computing applications from a mainframe to a client/server environment. Early in this project, it became apparent that many of the challenges would not come from the technological transition but from the changes required in the way both we and our clients work. This paper details some of the changes that our computing organization has made in the first eighteen months of this transition, the changes we have asked our clients to make. Initiatives such as restructuring the Information Systems organization, retraining existing technical staff, training clients and finding new ways to do business with our client base will be examined and evaluated.

Syracuse University is a private research institution located in central New York state. It has 12 schools and colleges with an enrollment of around 15,000 grads and undergrads. In March, 1993, the Vice-President for Research and Computing published a document which was a vision of computing at the University in the next five years. The following is an excerpt from that document:

> "Information technology will have an increasing influence on University life in the future. Communication, even more than computation, will be the essence of the revolution. The volume of available information will continue to increase at a staggering rate (currently doubling every four years), with effects that are both daunting and tantalizing. The challenge will be to access selectively the information we need and to use that information to develop knowledge, understanding, wisdom. A key objective will be to empower each member of our community with the appropriate technology and facilitated access to all of the information to which he/she is reasonably entitled."[1]

The effect of this document on our computing organization and our clients was dramatic. It led us to look for organization models, training methods and support models which would move both our staff and our clients into the future.

We, like many other institutions, are a mainframe shop. Information Systems (IS) has a programming and technical staff of 35 FTEs to support administrative applications. Our client base is approximately 2600; we have a portfolio of sixty application systems, a well satisfied customer base and have done a good job at providing systems and reporting needed to support the administrative operations of the University. We have not, however, been equally successful in distributing information to our schools and colleges for management and decision making.

A client/server project team was formed to develop a detailed plan for meeting the objectives stated in the computing plan:

- To migrate from primarily mainframe platforms to a primarily client/server, distributed environment.
- To make information about students, courses, financial accounts easily accessible to stakeholders.
- To provide our end users with an easy to access reporting environment with appropriate software.

One of the assumptions of our plan was that we would use existing staff rather than outsourcing the majority of the work. At the time we began this project we had a very stable, experienced staff. We felt we needed people who knew and understood our business and we would retrain them. The first phase of our migration involved setting up a client/server evaluation platform - a sort of laboratory. As we evaluated technologies, we realized that perhaps the most difficult transition would be changing the way the programming staff and our clients work. This change would be effected at the same time the University was undergoing restructuring and budget cuts. We looked first at our own organization and staff.

[1]Benjamin R. Ware, 93 Forward Directions for Computing at Syracuse University, March, 1993, p.1

Reorganizing

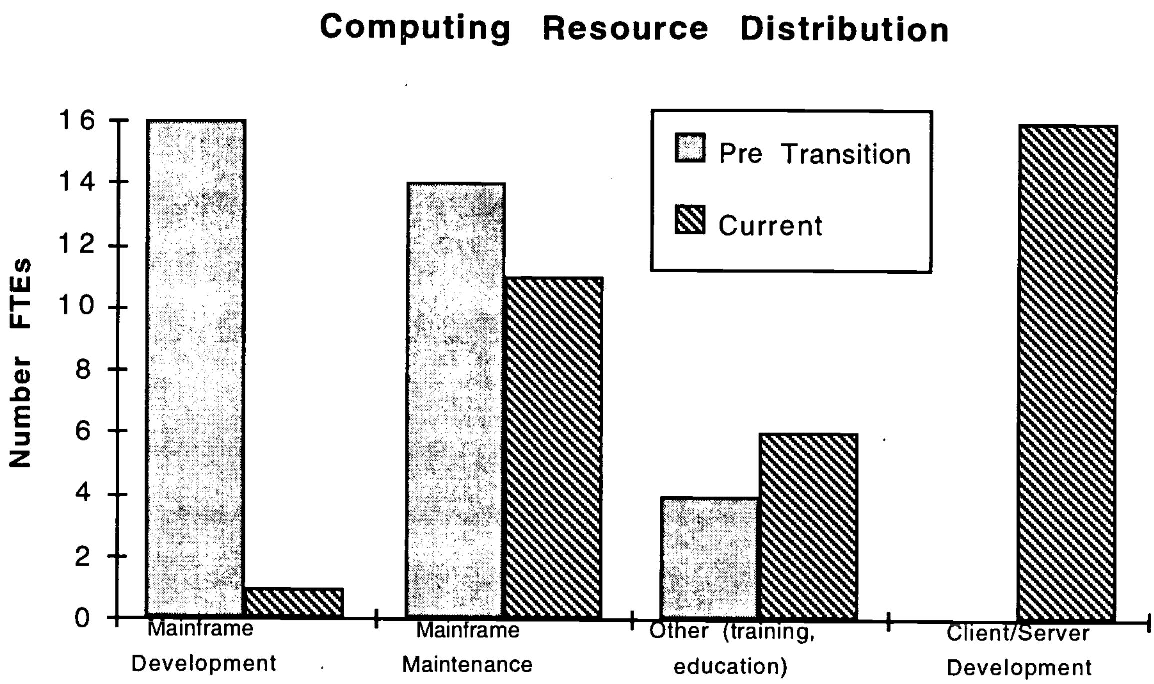
Our own structure was a hindrance to the way we needed to work. The programming staff reported to three application managers. Each manager was responsible for a specific suite of applications and clients. Resource was managed within each group with occasional transfers from one group to another. One advantage of this model was that our clients had a stable point of contact with programmers who were very familiar with their systems. Clients often would go directly to a programmer with questions and problems. Requests for system enhancements went to the appropriate application manager who would set schedules and priorities using the resource in their own group. Clearly this kind of model would not serve us well in our transition. We felt we needed to do three things:

- Establish an internal organizational structure which would be flexible and responsive to ever changing resource needs for both the mainframe and client/server environments.
- Ensure that our legacy systems were maintained but enhanced only when absolutely necessary.
- Change the way our clients communicated with our office to allow our staff time to retrain and move ahead with new technologies.

First, we changed the organization of the office. For maximum flexibility, all incoming work and programming resource should be managed from one point. The three groups of application programmers were consolidated to report to two application managers. Although each manager has a group of clients for whom they are the primary contact and coordinator, they jointly oversee project schedules and manage programming resource. Programmers were told to refer all calls from clients to one of these application managers. This moved our clients' "stable" point of contact to the managers making fewer interruptions for the programming staff and allowing the managers to be the master schedulers for all work. It also provided our clients with their first visible effect of the technology transition. For our staff, this model emphasizes work in teams which are organized for the life of a project.

We have three groups of technical support in the computing organization, two in the academic area and one for administrative. We evaluated combining these groups last year. Although it makes sense that they will be combined at some point in the future, there is not yet enough commonalty to make the change feasible.

Having settled our internal organization we looked at how we would interact with our clients during the transition. Our vice-president sent a letter to the administrative vice-presidents stating that the majority of our programming resource would be directed to client/server projects and mainframe applications would be enhanced only to meet changing legal requirements. We waited for the roof to cave in - it didn't! About two months later, reality set in and we did receive some letters of protest. Clients who were displeased with our decisions have been encouraged to talk to our vice-president. He plays a vital role in facilitating the politics and funding for this project. In addition, our project managers have been extremely diplomatic and sympathetic but firm with our clients. We made a smooth organizational transition with the aid of their facilitation, listening and negotiating skills. We have found, however, that we will need to spend some resource in enhancing those systems which will not be replaced until after 1997. We are working with our clients on a case by case basis to decide what will need to be done for these applications. The change in the distribution of work effort is reflected in the following graph:



Rethinking

As a result of our experience with client/server pilot projects, we realized that we needed to change the way we developed systems. Members of the client/server project team made the following recommendations for application development:

Use object oriented analysis techniques
All of the application development tools that we evaluated were using some object orientation principles. We felt that the staff would need to understand these principles and they should be used in our system development methodology.

Use a three tiered architecture for building our applications
One of the productivity gains we wanted was from reusable code. From reading, conversations with consultants and others experienced in client/server architecture, the three tiered approach to programming seemed the way to go. This would divide an application program into three parts - presentation, process logic and database services.

Define specialists for some areas
Given the long list of new technologies, we would need to designate specialty areas where we would concentrate training on a small number of people who would then be available as a support resource to project teams. We refer to these people as mentors to reflect their role as an educator and guide.

Foster success
With so many things going on in parallel we would need to look for combinations of software, human resource management techniques and organizational structures to provide the best possible environment for success.

We looked first at the role that mentors would play. We would not be able to train everyone in all of the new technologies; it was just too much change in a short period of time. The mentors would be trained and skilled in a particular area. They would join a project team as needed and work with them. We identified mentors for project management, object orientation, human interface design, data modeling, database design, networking, desktop hardware, server hardware and printing. In most cases, the mentor role is part time and we have at least two people who are knowledgeable in each area.

Led by the object orientation mentors who created a system development methodology outline, the mentor groups worked together to evolve the outline into a detailed document. The methodology is packaged as a project notebook which contains:

- A sample project management chart with each task being a methodology step
- An explanation of each step in the project management chart
- Samples of deliverables
- Guidelines from mentor areas
- Expectations and procedures for each mentor area
- A list of the mentors for each area

A notebook is provided to the team leader of each new project. Built into the methodology are "check in" points where the project team meets with some or all of the mentors to review models and plans. These meetings ensure that communication is taking place between all of the mentors and the project teams. The methodology has provided some structure to projects and gives us a common ground on which to discuss progress. It is not a finished product. To encourage feedback on the methodology and project management in general, the application managers convene a weekly meeting of team leaders to talk about issues, problems and even good things that are going on - a forum to find out what works and what doesn't. This is a productive group and contributes to the evolution of the methodology. Our experience is that the majority of the problems in implementing new applications have to do with the management of the project and not with the technology. In this environment, there are many more pieces to manage. The applications staff has to coordinate the installation of network connections, desktop equipment and server hookups. We continue to look for ways to improve our skills in this area.

Parallel to the development of applications, we have several teams involved in searches for application packages - namely, student systems, space and facilities management, classroom scheduling, and alumni development. By the time we were beginning to feel comfortable with the work we had done in systems development, we found that chances were good that we would be able to purchase software for our major application systems. We have backed off our plans to delve into three-tiered program architecture because it is not clear that we will be developing any major applications ourselves. Does this mean that all of our systems development methodology work was in vain? No indeed, the principles we have learned have helped our reeducation and reorientation process. We are now developing a package search methodology and find we still need to interact with mentor groups to do technical evaluations of software. We're still learning ways to work together effectively. We're learning to be better communicators. The days when we could develop applications in our own cubicles is over. We have asked for a lot of client interaction in the past and now we are asking for even more. Everyone has a stake in the project and in improving the process.

Retraining IS

As far as technologies were concerned, it appeared that there was very little that would remain of our former lives once this transition is complete. The list of new things to learn was daunting: operating systems, networks, CASE software, object orientation, application development tools, end user tools, networks, databases. In the first six months of the project we brought in a new database manager, a new CASE tool, several query products, two application development tools, new server hardware and workstations for staff who were still using dumb terminals. We used a variety of ways to reskill ourselves including reading, free seminars, vendor demonstrations, conferences, consultants, professional training, internal workshops, departmental work groups, and other University resources.

While the client/server team was evaluating hardware and software, the rest of the staff was told to spend a minimum of 20% of their time reading about client/server or enhancing their microcomputer skills. Our goal is to replace the majority of our applications by the end of 1997 so we looked for ways to come up to speed as rapidly as possible.

Many of the mentors attended professional training but we have also used conferences, user groups and reading in the education process. We found a graduate student with experience in client/server technology at one of the Fortune 500 companies and hired him to consult with some of the mentors.

There were some skills needed by a larger audience - and we have managed to provide them in a variety of ways. We felt everyone on our staff needed to understand object orientation. Our object orientation mentors formed a work group of people who were interested learning about object oriented analysis - this turned out to be most of the office! Led by the mentors, the group selected a specific methodology, purchased books and actually worked through the text, lesson by lesson, in weekly meetings. One of the things that made this work was the formation of small teams within the group. Each week one of the teams was responsible for leading the lesson discussion and exercises. This was a very successful model. It started our staff thinking differently, it reinforced team concepts and helped transition staff into the client/server project even before there was actual work.

For our applications development tool, we brought an instructor to our site to provide a week of training. Some months later, when several projects were underway, we hired a consultant for a few hours a week to help refine the way we were using the tool. The programmers also formed an informal group of their own with weekly meetings to share experiences, tips and techniques using the new software.

Finally, we worked with the Human Resources Staff Development office to provide a special session of Facilitation Training for people in our office who were interested. These are skills that we feel will be vital to our staff in the future - one thing that is evident is that a large part of our job will be to bring things together - people, processes, software and hardware to provide a solid computing environment. We are already putting these skills to good use.

Current Issues

We have learned that in our new environment we have to try things before we have all the answers. We have to be willing to implement short term solutions with a vision of the long term. We have to be willing to try new things and look for creative solutions. We have to be willing to redo. These are concepts that are very different from the carefully-studied, tried and true solutions we have implemented on the mainframe. This is an ongoing process and we continue to evolve as professionals and as an organization. Some of the issues which we are currently attempting to address are:

Changing job descriptions
Although we have changed the organization of our office and changed the way we work, our job descriptions and titles remain the same. The more we look at them, the more we realize that they do not fit the way we work today and how we need to work in the future.

We don't yet know exactly what our office will look like in four years, but we believe we need to evolve to another model and have the flexibility to move people where they are most effective.

Employee Recognition
This has been an issue for some time, it has just become larger. The University has no employee recognition program and neither does our department. At a time when we are asking more of people and are unable to reward them monetarily, it becomes critical to have ways to let people know they are doing a good job.

Performance Reviews
In the past, performance reviews have been done by the managers of each area. With our current work model, we need to look at a different way of approaching reviews so that work done in the various project teams is taken into account and recognized.

This transition is not without pain, all of the changes that we have been making began at the same time that the University was restructuring. Once we started changing, we had to begin to reeducate our clients.

Dismantling the old

Before we began our move to client-server, as we met with staff from many other universities as well as contacts in the business world, we had been congratulating ourselves on the excellent job we had done providing data access and reporting avenues to our administrative clients. Now, looking at our expanded clientele, two things became clear:

(1) There existed a group of over 145 staff members who were using our mainframe reporting processes as a vehicle to obtain information. Many of these individuals had been trained in the use of what are referred to as 4th generation query tools (in reality about a 3.2 generation!) and had over the past few years built substantial self-generated reporting structures.
(2) There was another group, an even larger group, of faculty and staff who were using many sources to gather information, often re-entering data on their desktop machines to create reports they were unable to generate from the mainframe administrative processes.

In addition, some individuals in each group had invested time and resource creating departmental systems -- desktop database systems, usually populated and refreshed from mainframe extract files. Information that was not available on the mainframe was also stored in these databases -- 'value added' is a terminology we used to describe this data that is particular to the business or interests of a school, college or department.

Each group offered unique challenges, but had one thing in common: they were faced with moving from their current technology, re-investing both time and dollars into a new way of retrieving and reporting on data. Prior to 1992, Information Systems had provided quick turnaround, ad hoc reporting services for a nominal fee. In 1992 we moved to a process which incorporated the use of a query tool with a user generated request to combine query results and standardized outputs such as reports, labels or files available for download. This process worked well for our administrative clients who could access and understand mainframe, on-line systems which provided these services. However, few faculty or staff in academic offices needed or wanted access to these systems. Client server will change both the need and the source of the data available for reporting. Mainframe clients will be using PCs or Macintosh based retrieval and reporting tools. Those already familiar with the world of DOS will find themselves moving into the world of Windows and probably to an upgraded if not entirely new set of software tools. The benefit to our clients is that these tools provide reporting and database services through a GUI, point-and-click, easy to use presentation, making desktop reporting and record keeping a viable alternative for our current and our new clients.

While the cost of standard desktop machines has moderated over the past few years, the level of equipment that is required to handle multiple software products accessing large reporting databases has become an issue. We have changed our recommended level of equipment several times during the past two years and know that we will continue to issue new guidelines based on advances in technology. But what we have come to understand and accept has caused many of our clients to suffer from 'sticker shock'. A natural reaction to this ongoing, rapid change is for the client to say, 'I'll just wait a little longer to get just the right equipment, just the right version of the software, just the right training'.

Some are enthralled with technology and want to set up the office of the future immediately; others are overwhelmed by the complexity of the new environment. Our job has been to work with all areas to develop hardware, software and training guidelines based on the information needs of each client and to encourage them to start down the information super-highway sooner rather than later. The operative word here is start. Our evaluation may show that an office should start using the data warehouse. That the best way to start using the data warehouse is to purchase one or two high-level machines and start learning Windows or the Macintosh environment. The staff members who will be accessing the data warehouse can then start to learn the software products that will allow them to turn data into information. Because we have a timeline for the movement of our major mainframe systems to the client-server environment, we are able to help offices plan for future purchases of equipment, software and training. If the on-line computing services which they are currently using are not scheduled for migration until 1996, then a workable computing environment can be phased in, giving a department more time to manage the impact of this change upon their resources, staff as well as financial.

Training

Very early in our client-server initiative we realized that while re-training our Information Systems staff was a major project, re-training our client community involved educating large numbers of individuals about a new computing environment, client software and applications, and for those using the data warehouse, providing information about information! A decision was made to appoint a full-time training coordinator. We were fortunate to have an IS staff member with a strong background in analysis and design who also had experience preparing clients for new system implementation. Her degree path had originally been directed toward teaching and she was prepared to step up to the challenge of addressing training from a University-wide perspective, encompassing staff, faculty and students.

By combining our strong network of volunteer training staff with professional consulting and training organizations, we have put in place a wide range of training opportunities. We have opened one training facility, away from phones and every-day interruptions and will be opening another in the Spring. Class schedules are mailed to staff and faculty, posted via electronic mail and published in our campus newspapers. Where training in the mainframe environment consisted of 'How to Use the Student Records System', our sessions now cover a variety of topics such as Beginning WORD, Using the Internet, Windows at three levels: Beginning, Intermediate and Advanced. Courses are available throughout the year, in convenient time slots. Some are directed toward faculty and students while others are designed to address the needs of the administrative or academic department staff. We have negotiated with professional training organizations to provide reasonably priced, on-campus training to University personnel and continue to survey the staff to plan for future offerings. We believe in 'just in time' training; coordinating new software, new releases of currently used software or implementation of new systems by scheduling sessions that have a client leaving the training course, returning to his/her desktop machine to begin putting new skills to use immediately.

Initiatives

As our client-server vision began to develop, we felt that one of the most important steps we could take was to let the University community know what we were trying to accomplish, how we planned to make this move, and how our time frame would effect various areas within the University. We developed 'THE ROADSHOW', a presentation which gave a brief overview of the current mainframe environment, our vision of the client-server environment, the benefits to the University of this move and a time-line projecting movement of major systems. We made this presentation to our 'old' clients, to our new audience, to faculty, to staff, to students, to anyone who wanted to learn more about our plan. All of our sessions were inter-active, providing an overview of the project, encouraging client input, offering to speak to focus groups, and just generally trying to put this change into perspective. Like us, 'THE ROADSHOW' kept changing; as we gained new insights and understandings about the client-server environment, we passed that information on to our audience. In addition to the presentations, we provided multiple avenues for clients to contact us with questions, suggestions and yes, even criticism. We set up an electronic mail ID, we published phone numbers where we could be reached and encouraged clients to let us know what issues needed further explanation. We stressed open communications because we felt, and continue to believe, that fear of the unknown can undermine a project, that information can allay fear and build understanding. Understanding the impact that this change will have on our client community has enabled us to begin working together toward a successful implementation of our computing plan.

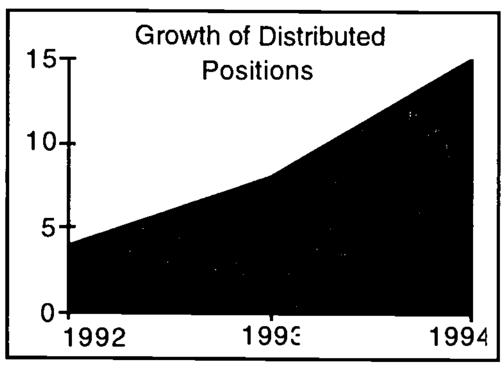
One of the portions of 'THE ROADSHOW' that was met with interest and enthusiasm was the prospect of a data warehouse, an easily accessed repository of University information which could be used to meet the day-to-day reporting needs of a wide range of staff: administrative and academic. The warehouse categorizes data into subject areas such as Student Information or Financials and will include current, historical and summary data in each of the areas. Human Resources information was our first subject area and has been available since Spring 1994. This fall we are providing information on currently registered students and will have Admissions, Financial Aid and historical student information available during the Spring 1995 semester.

The availability of this information in the warehouse has created a new audience for us -- faculty and staff in academic offices. Our direction at Syracuse University is to encourage greater involvement of our schools and colleges in the recruitment, admission and retention of students and access to information is an absolute necessity if these areas are to successfully discharge their new responsibilities. In selecting desktop query tools, we had to take into consideration the needs of this audience as well as those of our mainframe query audience; the tools needed to provide 'push-button' access as well as powerful selection and reporting capabilities. Other important requirements for a tool, based on our installed base of PCs and Macintosh hardware, was that the software run on both platforms, that it be similar enough in presentation and functionality to allow training and support to be addressed from a single viewpoint. After reviewing a wide-range of products, we selected two which we feel meet our current needs. We have recently completed 'train the trainer' sessions which included Computing & Media Services staff members as well as key individuals from the administrative areas supporting the data warehouse such as the Registrar's office, Human Resources, Admissions and Financial

Aid. Our Training Coordinator is developing courseware at a variety of levels, from 'fill in the blanks' to the 'point and click - create a query' class to 'so you really want to learn SQL!' level. Because understanding the software is only half of the learning experience and creating reports is only productive when you understand the data you are using, classes will also be available for staff to help them understand all the information that is available from the data warehouse. We have also created an on-line meta data database -- information about data: what do these code values mean, what operational system supports this piece of data, are there special considerations related to this information?

Two other initiatives are also playing a part in helping our clients deal with change . Last year several of our administrative clients banded together to form a user group, New User Technology Support Services. This group has sponsored several forums to discuss various aspects of managing departmental computing services. Many of them have experienced the 'pain' as well as the 'gain' associated with computing and would like to share their experiences with others, providing support and guidance to staff in areas just beginning to investigate how access to information can work for them. While Information Systems participates in this group, and offers support when requested, the driving force behind the discussions and presentations is from a client perspective.

Second, a program was developed to enable university offices to have some computing expertise in their department by subsidizing the hiring of staff with computing experience. We call these distributed positions - the department funds 2/3 of the person's salary and our computing organization funds 1/3. The distributed staff work in the client office but have an informal organization coordinated by a member of our organization. They have monthly meetings and are included in all of our departmental mailings and events. The first wave of distributed positions was in college offices; we are now beginning to see a move into administrative offices. These people are extremely effective because they are in the client offices every day, serving as liaisons between their 'home' department and Computing & Media Services. We all gain through this program since we are able to focus on some long term goals while we continue to support each other in our day-to-day operations.



Areas which have hired a distributed staff member are usually supporting a LAN. Since many University departments do not want to absorb the financial or personnel resources required to maintain a network, Syracuse University has developed a model to make local area network technology available to these departments. Services offered through this model include file sharing, backup services, access to supported software products, printing services and Internet access. A department can subscribe to one or more of the available services and the monthly fee is determined by the level of subscription. This model supports the Macintosh and DOS/Windows environments and will be one of the vehicles used for deployment of client-server applications, including access to the data warehouse.

All of these initiatives have one thing in common: they represent our attempt to provide support to our clients and to our staff. We are continually evaluating the impact that all of these changes have on us as information providers and on our clients, as the accessibility of information becomes an integral part of their job responsibility. We say that there are two words to describe our mission for the next few years, but we say them over and over again. Training, Training, Training followed by Support, Support, Support. Two issues related to training and support are currently underway:

Help desk software.

We have convened a committee to look at help desk software. Our help line handles a wide variety of inquiries dealing with all aspects of computing services at Syracuse University and we anticipate increased demands as more applications become available in the client-server environment. While this new environment often raises more questions than it answers, we know problems such as getting your password reset, accessing the server or getting help with a software error message will probably be repeat inquiries. We know that we will be able to develop standard responses and resolutions for many of these problems. This type of software will help us manage the administrative overhead associated with providing support to our clients.

Office Technology Support Group.

We have recently formed a group to provide both day-to-day support and long-range planning to our user community. Administrative and academic clients can call a single phone number, inquire about network or hardware problems, request an evaluation of their office needs or find out when their network wiring request is scheduled. Some of the support problems which this group is attempting to address require an on-site visit, not always an easy task since some units within Syracuse University, including Information Systems, are located a mile or more away from the main campus. So while the core of the Office Technology organization is small, they operate as a virtual organization. They can bring together skilled staff from a variety of units to address a request or they can call on a staff member in a remote location to make that all important personal contact with a client who needs support and assistance.

We know that we do not have all the answers. We think we have put into place a framework that will help us identify the questions and develop solutions. We have come to understand that as we plan for the future, we can no longer apply the words *long term* to our computing solutions, that our plans must remain flexible. Since many of our solutions rely on technology which remains in the future, we have added *interim solution* n to our vocabulary. But we have developed a plan; we have set goals and are moving forward. A positive attitude and technical training will be equal partners in our ongoing staff development as we begin to use new skills to provide better service and support to our clients. Our advice: Look forward and never believe that you and your organization don't have what it takes to succeed.

A Data Warehouse--The Best Buy For The Money

Leonard J. Mignerey

The Catholic University Of America

Washington

D.C.

ABSTRACT

Most, if not all IT professionals in university environments are operating in the extremely stressful reality of shrinking resources and expanding demands for service. Coupled with this dynamic is an increasingly rapid technology cycle. If demand could be held level this factor alone would be exerting extreme pressure on the IT environment.

The beginning of the paper briefly describes why Data Warehouse technology is a smart investment, in both resources and technology, and why it provides added value to the user community. The Catholic University of America's (CUA) experience in successfully implementing a pilot Data Warehouse project is also described.

A DATA WAREHOUSE--THE BEST BUY FOR THE MONEY

CUA's MIS department is continuously trying to answer the "Where do we go from here?" question. To accomplish this, current technology is constantly analyzed to separate the fact from the hype. Once the facts are established, a determination is made as to where our very limited development dollars should be spent to provide the greatest possible service improvement to the user community. Our goal can be stated as rule: Provide the greatest short term Information Technology (IT) improvement to the user community within the long term context of building an IT architecture that will be capable of evolving to the technologies of the future.

Data Warehouse technology fits this metric. This technology is based on the premise that there are two fundamental types of data existing within any enterprise. The first and most widely understood is termed operational data. Operational data is the data that directly supports the business functions and for which the majority of applications have been written since business programming became a practice. The second type of data is informational data. This is the data that supports the decision-making process of an organization and as a specific form of data, it is not as well understood as operational data. Many organizations have not as yet made the distinction between the two data forms.

> W.H. Inmon (1993), in his landmark work, *Building the Data Warehouse* (QED Press, Wellesley, Massachusetts), offers the following definition of a data warehouse: "A data warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision making process."...The driving force behind the evolution to the data warehouse is the need to gain *informational access* as opposed to *operational access* to corporate data. Operational access means access to the current state of specific instances of data....Informational access, by contrast, implies access to large volumes of data for higher level assessment, planning and strategic decision-support activities. (Ferrara and Naecker, 1993, pp. 26-28)

Differentiating operational data from informational data dictates a fundamentally different design criteria for the operational database versus the data-warehouse database. An operational relational database is (theoretically) built according to the rules of the first three normal forms. In brief, data is stored in its elemental form, there is no redundant storage of data, and any required data that does not represent an elemental data element is derived from an amalgamation of elementary-data elements. Data can be both extracted from and stored into the database. In general, the database is optimized for the update process not the extraction process.

The source of the operational data is generally from interactive on-lines and the operational database is designed with great care. The functional processes of the enterprise are supported by the implemented structure, which is built according to the rules of the standard Software Development Life Cycle (SDLC). On-going revisions to the basic data structure are not part of the plan.

The data warehouse is not built to support the functional process of the enterprise. It is built to facilitate the use of information. The source of data for the data warehouse is the operational database, which is optimized for the extraction process. In fact, the data warehouse can only be updated by the operational database; it is a read-only resource. Unlike the operational database, the normal-form rules do not apply and any de-normalization in the design that will facilitate the information-gathering process is acceptable. Therefore, fields containing summarized and other forms of derived data are perfectly acceptable. "Most access of the warehouse is at the higher levels of summarization. These levels contain less data than the lower levels do and are indexed on many fields. The lower levels of data are indexed on only a few fields" (Inmon and Kelley, 1993, p. 38). Furthermore, the design is iterative in nature. Since a warehouse does not support a suite of update applications it is not dependent on a pre-defined data structure; and, because the warehouse assumes the predominance of ad-hoc usage, design changes can be made as the need becomes apparent. Therefore, there is minimal impact resulting from design-change requests because only the interface between the two databases is affected.

The first step in data warehousing is to simply create a specialized, replicated database that is optimized for the "what-if" informational needs. The only additional technology needed for this step is a method to perform the extraction of data from the operational database into the data warehouse including the appropriate machinations for aggregation. Although it is certainly possible to develop this interface, there are a number of commercially-available solutions from the major database vendors.

The data warehouse is ideal as a centrally-maintained, distributed resource. The user community can help design it and is then free to use Rapid Application Development (RAD) technology to build its own applications for access, with the support and encouragement of the IT staff. This is a significant role reversal for the IT and user communities--IT is doing the data entry and the user community is building the tools to use the data!

At CUA we decided to build a prototype data warehouse. The first step was to identify a pilot group. A pilot group should have certain characteristics, the primary one being an active interest in the concept of a data warehouse. Furthermore, the members of this group must be willing to set aside time in their schedules to participate in the process. The choice of members for this group is very important because a successful pilot project sets the precedent for enhancements and expansions that will follow the project into its production phase. The success or failure of the pilot project will influence resource allocation for additional data-warehouses around the campus.

At CUA, three individuals, the Registrar, the Enrollment Management Analyst for Admissions & Financial Aid, and the Assistant Director for Financial Aid were asked if they would be interested in participating in a data-warehouse pilot project; all three accepted. These individuals do not have comparable positions on the university's organization chart, but they each had an active interest in more efficient data access for reporting purposes. Two additional individuals, the Director of Financial Aid and the Assistant Registrar lent support and suggestions to the group out of general interest for the project.
The first phase of the warehouse project focused on the immediate improvement of the reporting capabilities available to the pilot group. With the exception of this pilot project, CUA's operational database serves as the sole source of ad-hoc reporting on administrative data. This database contains 100 tables and 921 data domains; it was designed to support the functional processes of the university rather than the decision/planning processes. Although the functional structure provides the ability to perform ad-hoc reporting, it is not the ideal structure for report generation.

In a normalized database, data stored as elementary data elements, serves the on-line update applications very well. However, the query process is complex for even simple types of extractions, such as extracting a translation of a code along with, or instead of, the code itself. Extracting code translations is one of CUA's biggest problems with ad-hoc queries and provides a prime motivation for building a data-warehouse. For multiple translation retrievals, one must make the database treat the INDIVIDUAL_CODE table as if it were a series of separate tables, each containing the values for a specific CODE_TYPE. Each reference to the INDIVIDUAL_CODE table must have a unique name or alias. For some "real-life" queries, this can quickly reach a level of complexity that is too intricate for the user, an ad-hoc query tool, and eventually the system itself. Retrieval times shoot up into hours rather than minutes; some retrievals have run for more than a day in a test environment before having to be terminated.

An additional problem with using CUA's operational database as a query resource relates to the large amount of data that it stores. In private industry the operational database is a relatively small entity designed to control daily functioning. For example, once the widget is manufactured, sold and paid for, the operational database does not need to track it. In this environment the data-warehouse is the larger of the two databases. It is designed to provide a resource for historical data, and management uses it for analysis and planning.

A university environment is the exact opposite of most industries and is also opposite the general concept of data-warehousing. In a university environment, data stays active on a student for many years; so the historical database is the operational database. On-line programs quite often access, and sometimes update student data from prior semesters. Also, the operational database contains preparatory information for future semesters. Management, however, does most of its analysis on the current and future academic years, and is only rarely interested in the full historical database. The historical data often present an un-necessary level of complexity for managements queries.

Discussion of Project

General Discussion of VAX Data Distributor

Because CUA participates in Digital Equipment Corporation's (DEC) Campuswide Software License Grant (CSLG) program, and because CUA is already using DEC's Rdb for our operational database; DEC was the prime candidate as a source to provide a tool to implement the data-warehouse project. Included in this program, which allows CUA to use much of DEC's software for one low fee, is a product titled VAX Data Distributor (VDD). The following description from the VDD documentation provides a general overview of the product:

> Data distributor makes data available to users and applications at multiple sites in a network. From a source database, Data Distributor enables you to perform the following tasks:
>
> Transfer an entire source database or a subset of that database. The target of the transfer can remain on the same processor or can be on a remote processor.
>
> Create a target database that maintains a relationship with the source database. By maintaining this relationship, Data Distributor can periodically update the target database to reflect any changes made to the source database.
>
> Transfer data from multiple source databases into a single target database.
>
> Schedule transfers for future, automatic execution. (DEC, 1993, p. 10)

Conceptually, VDD does not do anything that could not be done manually by an experienced database administrator. The strength of the product lies in its ability to automatically generate all the database code that is necessary to create and maintain a target database, the contents of which are based on the contents of a source database. It can be thought of as a 4-gl for database administration. Based on a set of user-supplied requirement definitions, it generates complex database code.

The VDD process of creating and/or maintaining a target database from a source database is called a Transfer. There are two fundamental types of Transfers:

extraction and replication. Both types of Transfers can be done on demand, or they can be based on a defined schedule. Extraction Transfers create a complete new target database each time that the Transfer executes. A replication Transfer only transfers those data items that have changed (insert/update/delete) since the last Transfer process executed.

The replication Transfer was initially considered to be the superior choice because the total transfer time should theoretically be shorter than an extraction Transfer on a database that does not generally experience heavy updating, . Except for some pre-defined periods, CUA's database fits this criteria. However, further research into the replication Transfer revealed enough negative characteristics that, at least for the pilot project, the extraction method was chosen. The two major drawbacks were: a) the performance impact on the operational database, and b) the replicated tables had to match exactly the source tables, eliminating the possibility of moving the translation values from the INDIVIDUAL_CODE table to the same level as the coded values (denormalizing).

Creating a Transfer

Once the extraction method was chosen, the process of creating a workable Transfer began. The first step was to create the necessary VIEWS on the source database that would create the TABLES on the target database. The VIEWS needed to incorporate three criteria (a) they needed to contain matching translation fields for the requested coded fields, (b) they needed to contain only those student records from and including the first semester of the 1993-1994 academic year, and (c) they needed to exclude any records that had been marked as deleted.

The relationship between three key tables; CORE_DATA, ACADEMIC_CORE, and PROSPECT, presented a problem for the Transfer process. These are the three parent tables to all the other tables for the Admissions system, the Registration system and the Financial Aid system. There is a row in the CORE_DATA table for every student who is represented anywhere in the database. It is the basic table that contains fields like NAME, TITLE, etc. The ACADEMIC_CORE table contains basic academic data like SCHOOL, MAJOR, CUMULATIVE_AVERAGE, etc. The PROSPECT table contains basic Admissions data like SAT_SCORES, HIGH_SCHOOL, etc. Many of the VIEWS required logic that would include a data row if the student was represented in either the ACADEMIC_CORE table OR the PROSPECT table. When this limiting logic was combined with other criteria in the WHERE clauses, some selection processes took over an hour to start returning data.

It was necessary to create a special-purpose "driver" table in the source database to solve the problem. All records in the database for a particular student are related by a special-purpose field that contains a unique number, ID_SYNTHETIC. To solve this processing problem a table (WH_DRIVER_AD_FA_RG) was created that contained a single field, ID_SYNTHETIC for those records that met the limiting "OR" condition. Because this statement contains only one field and because there are no other conditions added to the WHERE clause, this selection starts returning rows almost immediately. The source database can actually load the 59,801 qualifying rows in 00:05:49. The remainder of the selections now include only an equality match to this *driver* table. The previously complex WHERE conditions are now simplified to:

```
WHERE
     some_table.ID_SYNTHETIC = WH_DRIVER.ID_SYNTHETIC  AND
     {any other conditions specific to the table}        ;
```

With this method, the selections written for the 20 tables requested by the user group all started returning rows in 00:02:00 or less.

To use this method, the driver table must be re-created before every Transfer. Fortunately VDD provides for user-controlled Transfer pre-processing and Transfer post-processing. VDD permits the definition of a Prologue command procedure and an Epilogue command procedure. These command procedures can contain any valid commands that can normally be executed in either Digital Command Language (DCL) or interactive SQL. In this situation, a prologue command procedure was created to perform the following steps:

I. Drop the existing special driver table (WH_DRIVER_AD_FA_RG)
II. Create a new driver table
III. Load the new driver TABLE based on the above explanation
IV. Create a unique index based on the sole field, ID_SYNTHETIC

With this process in place, the environment was established to create an actual test Transfer using the tables and fields requested by The Group. VIEWS for each of the twenty tables were created on the source database and a simple epilogue procedure was written to place indexes on the target tables after the Transfer was complete. The most translations requested by the Group for any single table were six on the ACADEMIC_CORE table. Once all syntactical problems were corrected, the Transfer was initiated. The total process took approximately 00:01:45 to complete.

Refining the Transfer

After the initial prototype proved to be functional, The Group was reconvened. Each member was handed a packet identifying the tables and fields that were contained in the new data-warehouse and a list of questions designed to further refine the design. They were asked to review the material and return all suggestions within a week. When the materials were returned, the following refinements had been requested:

I. A number of additional translations had been requested, in particular five additional translations had been added to the ACADEMIC_CORE table, raising the total number of translations on this table to eleven.

II. A number of tables had been further refined to contain fewer fields.

III. Two tables, the ADDRESS table and the FA_ALLOCATION table, were asked to be de-normalized. The ADDRESS table as designed in the operational database contains seven possible address types for each individual. To retrieve a particular address the user has to, (a) cross the ADDRESS table with the ADDRESS_DATA table, (b) specify the correct ADDRESS_TYPE code, and (c) specify a linking field, ADDRESS_NUMBER.

 The Group was only interested in two of these address types for the data-warehouse. Therefore, two new tables were created: CURRENT_ADDRESS and PERMANENT_ADDRESS. Addresses can be retrieved directly from them without any crossing.

 The FA_ALLOCATION table contains the dollar figures indicating how much money a student is to receive in aid per semester. If the user wants to retrieve all allocations for a particular academic year the FA_ALLOCATION table had to be crossed over itself three times to retrieve the data. The data-warehouse FA_ALLOCATION was de-normalized to contain parallel fields for FALL, SPRING, and SUMMER all in the same row.

These changes were added to the VIEW definitions that had been initially defined and the Transfer was re-run. Eight hours later the Transfer was manually terminated without having run to completion. When the log files of previous runs were compared to this run, the transfer time on the ACADEMIC_CORE table had increased from 00:12:35 to 02:31:59. When the transfer time for the table was divided by the number of records transferred it was evident that the data-record transfer rate had dropped from 46 records/second to 3.04 records/second. Obviously, the additional translations that now had the ACADEMIC_CORE table crossing the INDIVIDUAL_CODE table over itself 11 times, had reached some critical mass.

CUA's systems manager was consulted and the machine resource Input/Output statistics were reviewed for a test Transfer that included only the ACADEMIC_CORE table. From reviewing system performance statistics it was evident that all the activity was absorbed with the database's attempt to resolve the translations, while very little actual data was being accessed.

The multiple crosses of the INDIVIDUAL_CODE table were the obvious source of the problem. The first attempt at a correction was to create, on the source database, individual tables for each of the required translations. These individual translation tables contained only those code values that matched the code type of a specific translation. The idea was based on the assumption that Rdb would have an easier time loading values from many small tables than it would with loading values from one large table that had many virtual copies of itself. Since the ACADEMIC_CORE table contained the most translated fields, it was chosen as a bench-mark test table. A special Transfer process was defined to create a target database containing just ACADEMIC_CORE. The WHERE condition in the SQL code replaced the multiple crosses of INDIVIDUAL_CODE with 11 equality conditions, each satisfying a single translation value. This extraction method produced a transfer time on the ACADEMIC_CORE table of 04:33:28 and data record transfer rate of 1.6 records/second, which was surprisingly worse performance than the use of the single INDIVIDUAL_CODE table.

It was next theorized that the performance problem may be related not only to the large number of crosses or tables in the select statement, but to characteristics of VIEWs that may not be encountered if native TABLEs were used as the source for the Transfer. Although VIEWS appear to the user as a TABLE, they do not actually exist until a selection request is made against them. However, another VDD limitation was encountered at this point. In the Transfer definition statement, a SELECT statement can be used on a TABLE to filter the data that is actually transferred to the target TABLE. However, a SELECT statement used in this manner is restricted; only the TABLE being transferred can be named in the SELECT statement for that TABLE. Since the WH_DRIVER_AD_FA_RG table was necessary to make the selections start returning rows in a reasonable time-frame, this had a major impact on the project. If the driver table was to be used, VDD syntax dictated that VIEWS must also be used.

The next idea was to remove all processing involving the code translations from the source database VIEWS. The INDIVIDUAL_CODE table would be added to the list of tables in the Transfer and re-created on the target database. Once the Transfer process had completed, the epilogue command procedure would then create individual code_translations on the target database. The transferred data tables would then be ALTERED on the target database to contain fields for the required translations. The tables would then be UPDATED on the target database using these individual code tables as the source for the translated values.

Another test Transfer was defined for ACADEMIC_CORE. The select statement only contained a cross with the driver table and a few fields to force the use of an existing index. This time the Transfer of ACADEMIC_CORE took only 00:04:16 and had a record transfer rate of 127 records/second and the system performance statistics now showed a much more balanced I/O picture. This method was then extended to all 20 tables that were to be part of the pilot data-warehouse. The full Transfer ran with a total elapsed time of 02:58:24. The elapsed times for the individual components of the Transfer were:

I. Elapsed time for the prologue command procedure--00:08:16
This included the time to drop, create, and reload the special driver table.

II. Elapsed time for the actual Transfer procedure--01:27:04
This included the time to create and load the 20 tables in the target database that were specified in the source database.

III. Elapsed time for the epilogue command procedure--01:23:04
This included the time to:

A. create indexes on all the new tables,
B. create and load the individual translation tables,
C. alter the data tables to contain short and long translation fields,
D. update the data tables with the actual translations,
E. create and load the VALID_FIELD_VALUE_LOOKUP table. This is the table that gives the users an on-line dictionary resource for all the valid coded values and their translations.

A Transfer schedule was then created and the process was scheduled to run every day at 22:00:00. The Transfer log was examined each morning for any reported errors until all syntactical errors had been removed from the Transfer. A review session was then scheduled with The Group for their first hands-on experience with the data-warehouse.

Results

Response to the product was very good. The Group felt that the data-warehouse demonstrated all the requirements they had asked for in the design stage of the project. They were particularly pleased with the ability to look up code values and as expected, the existence of the translation at the same level as the coded fields was very well received. The user group has now been given access to the warehouse and they are in the process of evaluating it. They have been reminded that this is an iterative process of building successively better prototypes and that they should feel free to be critical of the product. The project will continue past this initial delivery and it should evolve into a production system within a fairly short time period.

Conclusions

The successful completion of the data-warehouse pilot project and the pilot group's enthusiastic response to it has demonstrated that it is a needed resource at CUA. In a recent interview for Forbes magazine, Michael Hammer makes an interesting comment about the nature of work. "Work is the way in which we create value for customers, how we design, invent and make products, how we sell them, how we serve customers" (Karlgaard, 1993, p. 70). It is an extremely important concept. It is increasingly easy for managers of technology to lose touch with the "added value to the customer" component of the job. It is exceptionally easy for those managers to justify technological change from a technological perspective, and it is often difficult for them not to. The rate of technological change is so great that significant amounts of time are spent figuring out how to maintain functioning systems as technology continously changes out from under them.

The data-warehouse adds value to the CUA user community. It provides users a way to perform a portion of their work more quickly and easily. The data-warehouse is also in line with current technology trends.

As the distributed computing, client-server paradigm evolves, the issue of information retrieval must be totally re-evaluated. In many respects the industry is still attempting to do flat-file reporting against relational databases. In the future we will need to develop technology that can abandon the process of examining retrieved data for information, and instead will be intelligent enough to automatically provide the end product (information) to the appropriate clients, be they silicon or carbon-based.

Reference List

Digital Equipment Corpopration, DEC data distributor handbook. Marlboro: Digital Equipment Corporation, 1993.

Ferrara, R., & Naecker, P. A., The data warehouse: A giant step forward. DEC Professional, 1993, 12(11), 26-39.

Inmon, W. H., & Kelley, C., Rdb/vms: Developing the data warehouse. Boston: QED Publishing Group, 1993.

Karlgaard, R., Interview: Mike Hammer. Forbes ASAP, September,1993, pp. 69-75.

11/10/94 LJM

The Data Warehouse: 2 Years Later...Lessons Learned

John Porter, Data Administrator and Director of Institutional Analysis
Arizona State University
Office of Institutional Analysis and Data
Box 871203
Tempe, AZ 85287-1203
(602) 965-5959
Internet: john.porter@ASU.EDU

John Rome, Data Analyst Principal
Arizona State University
Office of Institutional Analysis and Data
Box 871203
Tempe, AZ 85287-1203
(602) 965-5959
Internet: john.rome@ASU.EDU

Abstract

A data warehouse is often the first client/server application that institutions attempt. Such was the case at Arizona State University (ASU). Two years ago, ASU initiated a project that brought together student, financial and human resources data in an integrated data warehouse. In this presentation, we plan to share the lessons learned after two years of work. Some of those lessons include: learning new technologies, understanding warehousing concepts, integrating data, designing the warehouse, marketing the idea, finding resources, establishing "officialness" of data, evaluating impact on Data Administration, defining data, making a production system, prioritizing information going into the warehouse, data access tools, training and security. We plan to share our success stories as well as our challenges. There will be a discussion of how the ASU Data Warehouse fits in the University's information architecture, future plans and a live demonstration showcasing the ease of use and power of this innovative information resource.

The Data Warehouse: 2 Years Later...Lessons Learned

It's amazing that the words data warehouse have become such a glamorous, *sexy* expression.[1]

Introduction

To remain competitive in today's business climate, an organization needs a solid foundation of quality data. Institutions of higher education need this capability as much as Fortune 500 companies. To give colleges and universities the "edge," many are turning to data warehousing. These institutions see the data warehouse as the *de facto source* of quality data for tactical and strategic decision making. Some critics believe data warehousing adds to an organization's information problem by adding yet another data source. However, the success organizations are experiencing with the data warehouse is evident. Data warehousing is a solid business strategy for the 1990s.

In this paper, we share the lessons learned from Arizona State University's (ASU's) data warehousing experiences. Building a data warehouse is extremely complex and takes commitment from both the information technology department and the business analysts of the institution. It takes planning, hard work, dedication and time to create a relational database management system (RDBMS) that delivers the right data to the right user. A data warehouse excites, but also disappoints. ASU's data warehouse is not a panacea for all the university's data woes, but a darn good start.

Data Warehousing Popular...But Not New

Data warehousing is not new. Data warehousing reminds us of an old mainframe concept from the mid-1970s: take data out of production databases, clean it up a bit, and load the data into an end-user database. International Business Machines Corporation (IBM) was first to coin the phrase "information warehouse" in late 1991. IBM's original concept met with skepticism because accessing non-relational data stores (such as IDMS®, IMS® or VSAM®) was too complex and degraded operational system performance. Based on these experiences, experts now agree that a warehouse needs to be a separate data store built with an RDBMS. While names such as "information factory" or "information refinery" surfaced and went, "data warehouse" is now the generally accepted term.

Definition

The most widely recognized definition of a warehouse is a subject-oriented, integrated, time variant, non-volatile collection of data in support of management's decision making process.[2] ***Subject-oriented*** means the data warehouse focuses on the high-level *entities* of the business; in higher education's case, subjects such as students, courses, accounts, and employees. This is in contrast to operational systems, which deal with *processes* such as student registration or rolling up financial accounts. ***Integrated*** means the data is stored in a consistent format (i.e., naming conventions, domain constraints, physical attributes and measurements). For example, ASU's production systems have four unique coding schemes for ethnicity. In the data warehouse, there is only one coding scheme. ***Time variant*** means the data associates with a point in time (i.e., semester, fiscal year, and pay period). Lastly, ***non-volatile*** means the data doesn't change once it gets into the warehouse. At ASU, even if a person's gender was unreported in a previous semester, the warehouse won't go back in history to correct that.

Use Increasing

In higher education, glimpses of data warehousing exist in the file extracts which institutional research departments receive or the end-user reporting databases that information technology provides. Consequently, data warehousing is nothing new; it is an old concept with a new name and better technology. The data warehouse is likely to become the cornerstone of client/server activity in 1995.[3] So popular is the notion, that a recent META Group report indicates 90% of their clients are undertaking warehouse initiatives, up from less than 10% just a year

[1] Inmon, W. H. (1992). [Conversation with paper's author.]
[2] Inmon, W. H. (1992). What is a data warehouse? (Tech Topic Vol. 1., No. 1) Prism Solutions, Inc.
[3] White, C. (1994, December). Client/server obsession. Database Programming and Design: Special Supplement, 7.

ago.[4] Similar trends are occurring in higher education, judging from the number of inquiries about ASU's data warehousing efforts. Some of the universities developing warehouse capabilities include Stanford and the University of Michigan. In the business market, analysts estimate the industry will grow to $2.1 billion by 1998, almost three times the 1993 total of $753 million.[5] Some of the major players vying for this money include IBM, Hewlett-Packard Co., Oracle Corp., Sybase, Inc., AT&T GIS and SAS Institute, Inc.; as well as Prism Solutions, Inc. and Red Brick Systems, companies already established in the warehouse market.

ASU's Warehouse Development

History

Development of ASU's data warehouse started in the summer of 1992 as an effort championed by the Department of Data Administration. Negotiations with an RDBMS vendor and a UNIX workstation vendor resulted in a one-year "lease" of their products for the cost of the annual maintenance contract (approximately $8000). While getting the warehouse server in place, over 20 companies agreed to provide complimentary copies of their data access tools. Although many of the access tools reviewed were in their adolescence, accessing data was much easier with these graphical user interface (GUI) tools than with the fourth generation tools in use. After successfully connecting to the warehouse server through the network middleware, Data Administration started compiling minimum hardware and software requirements for Macintosh® and PC/Windows™ machines.

A warehouse team of twelve individuals from Data Administration and Information Technology formed a development team to "prove" the warehouse concept. The team selected a representative group of ASU staff to serve as pilot users to test the data warehouse and access software. During the next few months, the team designed a student warehouse model based on over 200 questions, which the pilot users considered difficult to answer using current information resources. By extracting data from the operational IDMS database and loading that data into a Sybase® SQLServer™ database, the first client/server application was in place at ASU.

During 1993, many of the original warehouse team shifted back to their regular duties, which left the team with a core of about five people. That core group has remained intact, receiving additional help from ASU's Institutional Research Office and business area improvement groups (BAIGs). The BAIGs, organized to improve ASU's operational and informational data processing capabilities, contribute significantly to ASU's data warehousing project. Meanwhile, Data Administration initiated formal classes to train users on the warehouse. To date, there are over 150 trained warehouse users, with two classes being taught each month. The goal is to train 1,000 warehouse users, approximately 20% of ASU's full-time work force.

Before the production release of the data warehouse to ASU, the team introduced a logo for the warehouse (see Figure 1). Presentations, warehouse brochures, training materials and other documentation display this logo. This simple, yet effective, design helps individuals identify with ASU's data warehouse, establishing the warehouse as a distinct entity.

ASU DATA WAREHOUSE

Copyright © 1993.

Figure 1. ASU's Data Warehouse Logo.

Methodology

---

[4] Hammer, K. (1994, September-October). Will the data warehouse be warehoused? Relational Database Journal, 32.
[5] Cafasso, R. (1994, October 10). Praxis forges data warehouse plan. Computer World, 32.

Methodologies assist in managing a project's development. Formal principles, practices and procedures comprise a methodology. Examples of formal methodologies include Martin, Chen/Bachman, IRM, etc. The best definition of a methodology is Paul Strassmann's. He defines methodology as "a procedure that I understand and like." At ASU, the data warehouse team "understands" and "likes" the concepts of William ("Bill") Inmon. Reading any literature about data warehousing without seeing Inmon's name is rare. He was the first to coin the phrase "data warehouse," and is the evangelical voice on the concepts and benefits of data warehousing. In developing ASU's data warehouse, the warehouse team follows Inmon's ten critical success factors (see Figure 2). ASU's warehouse team still revisits these principles.

Inmon's Ten Critical Success Factors

1. Separation of operational and warehouse data and processing.
   (i.e., different data and processing, different technologies, serve different communities.)
2. Data volume management.
   (i.e., sheer volume defeats purpose, partitioning for performance.)
3. Coexistence with older legacy systems.
   (i.e., warehouse not rewrite of operational system, remember $ invested in legacy systems.)
4. Feedback loop implementation.
   (i.e., not single massive effort but iterative, users initially only give rough estimates of need.)
5. Rigorous and proactive treatment of metadata.
   (i.e., store directory of data, maps data between operational system & warehouse.)
6. Data integration.
   (i.e., warehouse fed from diverse & unintegrated data sources, time consuming & difficult.)
7. Proper user mindset.
   (i.e., users operate in discovery mode, warehouse architects understand and react quickly.)
8. Knowledge of historical versus current-value data.
   (i.e., warehouse not updated, serves management, what-if processing, data driven.)
9. Cost justification.
   (i.e., after warehouse power unleashed, support comes but not based on cost justification.)
10. Knowledge that existing systems aren't perfect.
    (i.e., can't wait until operational systems cleaned up, build independent of reengineering.)

Figure 2. Inmon's Critical Success Factors in Building a Data Warehouse.[6]

Architecture

ASU's data warehouse resides in a client/server environment. [*Client/server* is an emerging computing architecture where processing occurs on both the server and client, radically different from the mature centralized world of the mainframe.] As seen in Figure 3, we extract data from the mainframe and load it into a UNIX server running an RDBMS. ASU's warehouse server is a Sun® Sparc 630™ with 512 megabytes of memory and two processors, running the Sun Solaris 2.3™ operating system. The RDBMS is Sybase SQLServer release 10.x. Users connect through Ethernet to the warehouse over ASU's network backbone via Transmission Control Protocol/Internet Protocol (TCP/IP). [*TCP/IP* is the predominant network protocol used by UNIX systems attached to Ethernet and ASU's "preferred protocol."] The suggested GUI data access tool is DataPrism™ from Brio Technology, Inc., which runs identically in both the Macintosh and Windows environment. Microsoft Access® and Q+E™ from Intersolv, Inc. are some of the other tools used.

GUI tools build the structured query language (SQL) requests and bring the results back to the client machine. [*SQL* is the dominant query language for accessing relational databases and accepted by most non-relational databases in a standardized form.] The retrieved data *exists* on the client machine. This process is much different from the 3270 protocols the users are accustomed to, where the client machine is a "dumb" terminal connected to a mainframe. With client/server architecture, the data existing on the client machine is both

[6]Inmon, W. H. (1992, April). Building the data bridge: The ten critical success factors of building a data warehouse. Database Programming & Design, 69.

empowering and liberating. The users "own" the data, cutting and pasting at will, using their favorite client tools (i.e., spreadsheet, word processor, graphic tools).

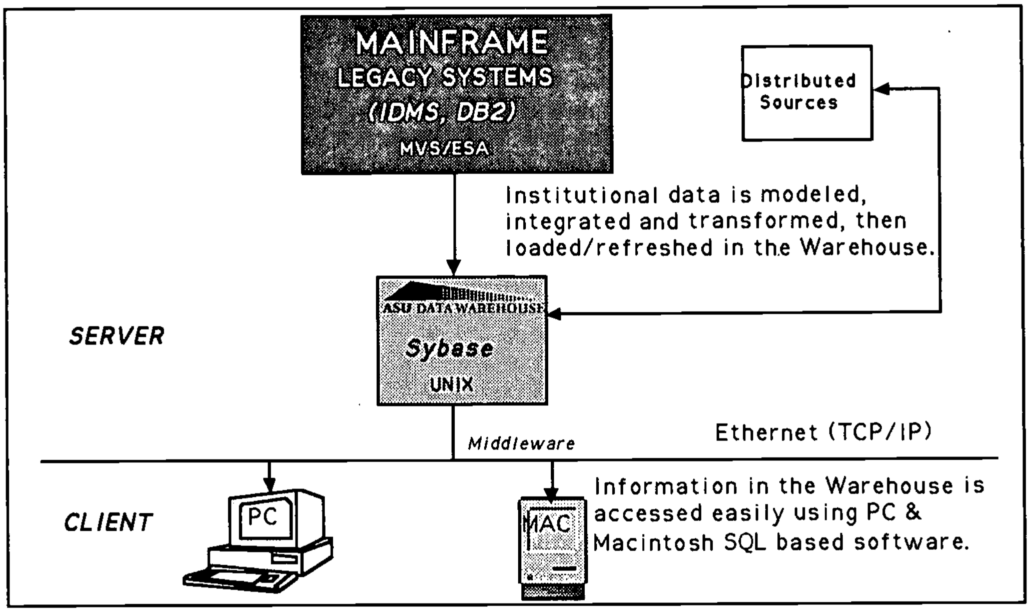


Figure 3. Diagram of ASU's Data Warehouse.

Modeling and Design

As business requirements and database technology become more sophisticated, the need for data modeling and design increases. ASU uses an "upper" computer aided software engineering (CASE) tool to design the warehouse. However, the entity/relationship (E/R) diagramming function and the object repository are the only features of the CASE tool used. [The *E/R diagram* is a pictorial representation of entities, the vital business relationships between the entities, and the attributes or fields used to describe the entities.] The E/R diagramming tool creates a graphical representation of the data in the data warehouse and automates the creation of data definition language (DDL), the technical language used to create the warehouse's tables, views and indexes. The object repository insures consistent definitions and characteristics of fields in the data warehouse. While an upper CASE tool is not imperative in building a data warehouse, it does help automate the development process and the E/R diagrams produce "road maps" to the data.

Designing a data warehouse is an iterative process. Warehouse models change as much as 50% after completion of the design. Designing a data warehouse is different from designing an operational system. First, the data content of the model is different. The warehouse wants data with a high value for executive decision making, whereas the data content of an operational system is more requirements driven. Second, since data is often unavailable, referential integrity in a data warehouse is sometimes inherently wrong. [*Referential integrity* is a feature of database management systems that ensures that each foreign key value has a matching primary key value.] In an operational system, business rules (relationships in an E/R model) dictate that an entity *must* have a relationship with another entity. In a data warehouse, that may or may not be the case. For example, in an operational system, *a student must have an address*. If that address is not available to the operational system, the ability to add that student to the warehouse still must exist.

There are four basic types of tables in ASU's data warehouse: data tables, lookup tables, virtual tables and summarized tables. ***Data tables*** contain raw data, extracted at the unit record level from the operational system. ***Lookup tables*** are code tables, defining the cryptic coding schemes that exist in the operational data. Lookup tables save space, improve flexibility, and allow the description of a code value to change while retaining its meaning. ***Virtual tables*** are views into the warehouse data. Views simplify the user's perception of a data warehouse, presenting data in a different way or restricting access to certain data (i.e., class roster appears as a single table, but the data resides physically on multiple tables). Lastly, ***summarized tables*** contain summarized data. These tables improve response time to frequently queried data and may become the foundation for subsequently developed executive information systems (EIS).

Database design is a creative process. In fact, given the same set of requirements, two designers usually produce different but acceptable solutions. Often, in database design, it is easier to just do it, than explain exactly what you did.[7] ASU's warehouse team follows the design guidelines in Figure 4.

ASU's Warehouse Design Guidelines

√ Identify major subjects or topics (i.e., define as tables on warehouse).
√ Add an element of time to the tables (i.e., semester, fiscal year).
√ Name fields in the tables or views appropriately (i.e., naming standards).
√ Add derived fields when necessary (i.e., calculated age, GPA).
√ Duplicate data if necessary (i.e., to decrease number of tables users need).
√ Exclude extraneous fields found in operational files (i.e., flags).
√ Create logical tables or views for ease of use (i.e., class roster w/ names).
√ Take security into design considerations.
√ Determine if data model answers business questions.

Figure 4. Warehouse Design Guidelines.

ASU's Warehouse Data Issues

What Data to Collect

A data warehouse must deliver the **right data to the right people**. However, the data warehouse may not be able to deliver all the data people want. People are always asking new questions, so predicting what they need is difficult. We started by asking users what data they wanted. Users e-mail or write down their questions, and send them to Data Administration. Another good starting point is to look at the data going to the institutional research department and the data included in official university reports (e.g., data provided to government agencies like EEOC, IPEDS, or NCES). Our experience is that warehouse users quickly let us know what data they want.

Update Frequency

A data warehouse must deliver the right data to the right people, **at the right time**. What is the right time? The answer is, "it depends." In ASU's data warehouse, we enter data yearly, by census date, monthly, bi-weekly, weekly, and daily. By rule, the more often you update a table in the data warehouse, the more operational in nature it is. For example, ASU's data warehouse extracts daily address changes on students. Many warehouse users create labels for student mailings and need current address information. Updates to code tables occur daily too. However, we try to limit the number of data elements loaded on a daily basis, since there is a cost associated with loading the warehouse. [Authors' note: In the future, daily updates to ASU's data warehouse will "replicate" data in operational systems. *Replication* is a popular industry solution of copying data and placing it locally for processing, which appears to users as direct access to data.]

Integration

Integration is the most important characteristic of a data warehouse, and the characteristic lacking in most operational systems. Integration gives a data warehouse credibility, consistency and **real power.** When designing these capabilities into ASU's data warehouse, the team recognized data in need of integration and data that integrates (see Figure 5). Data in need of integration in ASU's data warehouse include fields like ethnicity, gender, and name. A major integration problem at ASU exists with the department code structure. There are three or four recognized sets of department codes in the various operational systems. Because of the number of department codes

[7]Date, C. J. [Discussion on database design.]

in use, we plan to rectify the problem on the operational side first with a single code, before adding the department code to the warehouse.

Data in need of integration

- Name
- Ethnicity
- Marital Status
- Organizational Unit

Data that integrates

- Fiscal Year, Semester
- Account
- Unique Identification ID
- Organizational Unit

Figure 5. Integration examples.

Integration also requires data that integrates. This is the data that spans the high level subject areas of the data warehouse. At ASU, these high level subject areas are students, financial information, human resources, and courses (see ASU data warehouse high-level design in Appendix A). Examples of data that integrate or cross-walk the high level subjects are fiscal year, semester or term, department, course, a person's unique identifying ID, and account number. Data elements that integrate are the very fabric of an operational system. If these elements differ in format or domain between systems, integrating the data in the warehouse is difficult or impossible. [*Domain* is the set of allowable values that a data field can legitimately take, i.e., permitted values, range of numbers, allowable dates.] When data successfully spans high level subject areas, consider the data warehouse completed and "self-actualized."

Officialness

Making **official numbers** available in a data warehouse brings credence and appeases the user getting different numbers with every query. We add official "numbers" to the warehouse to limit how much our users must understand the impact of timing on data. To achieve officialness, institution's select census or "cut-off" dates for measuring data. For example, at ASU, there is the "21st day" (of the semester) census for student enrollment; in the financial system, fiscal year end; in human resources, September 30. With these census dates, there is a distinct period of measurement, making historical trends much easier to compare and allowing integration across systems. For some requests, official numbers are better to use (i.e., historical trends), while at other times the most current data is best (i.e., financial decisions). At ASU, both numbers are available on the data warehouse. To simplify user queries, official and current values appear in separate databases (see high-level design at Appendix A).

Security and Privacy

Security and safeguarding privacy are **major concerns** in a data warehouse. Security in a database means protecting data against unauthorized disclosure, alteration, or destruction. Granting SELECT (authorization to read only) access to tables (or views) achieves security in a warehouse. Although many RDBMSs support column level security, ASU has not implemented this feature, primarily due to the high cost of administering user access. In traditional operating systems, tasks or screens control access. This usually results in access to a single record or instance of data (i.e., verifying admission status of a student, etc.). However, in a data warehouse, employees have access to a table or all tables in a subject area, which means access goes beyond retrieving individual records to retrieving groups of records.

At ASU, read-only access to the data warehouse is at the database level, which means access to a group of tables. This procedure follows an open access policy for employees approved in 1993. For example, the Office of the Registrar is the trustee of the STUDENT database, the Human Resources director trustee of the HUMAN RESOURCES database, etc. In these databases, read-only access excludes access to name and address. To obtain name and address information, our data trustees grant access to the PERSON database. The user's business need determines access to the PERSON database. Additionally, training classes emphasize the Buckley amendment and Family Educational Rights and Privacy Act (FERPA). Also, users receive training on the appropriate use of warehouse data.

Results

The diagram in Appendix A shows ASU's high-level warehouse design in the context of databases. [*Database* is a collection of tables or files and is loosely equivalent to a subject area.] The STUDENT, FINANCIAL, HUMAN RESOURCES, and COURSE databases comprise the foundation of the warehouse. These databases contain the "granular" or detail data and updates occur on a weekly, bi-weekly or monthly basis, depending on the database (i.e., STUDENT = weekly, COURSE = weekly, HUMAN RESOURCE = biweekly, FINANCIAL = monthly, etc.).

Name and address exist in the PERSON database. This database integrates with other databases through a unique identifying ID. Updates to PERSON occur on a daily basis. Individuals creating class lists or labels are the primary users of this database. Although warehouse "purists" may scoff at the idea of daily updates (i.e., reproducing the operational environment), creating labels is a legitimate business need at ASU. For an employee to get access to name and address, they need permission from the trustee (the Registrar for student information and the Human Resources director in the case of employees). We protect names and addresses for security and privacy reasons in PERSON, but the utility of the warehouse for planning and decision making through the other databases is unaffected. The LOOKUP database contains all lookup tables for the entire warehouse, and is updated daily.

The OFFICIAL database contains census values for frequently used data. Instead of hundreds of tables as in the STUDENT database, the OFFICIAL database only contains dozens of tables. The OFFICIAL database helps users understand the concept of officialness and the smaller size makes a good starting point for new warehouse users. The OFFICIAL database is actually a collection of views into the larger, granular databases, such as STUDENT. Besides being easier to use, the OFFICIAL database achieves a summarized flavor, which less sophisticated users can comfortably use. In the OFFICIAL database, properly constructed queries result in answers that match the official reports released by the Department of Institutional Research.

Ten Lessons Learned

During ASU's data warehouse development, we learned many valuable lessons (see Figure 6). Most of these lessons are general in nature, in that any institution starting to build a data warehouse can learn from them. A few lessons are particular to ASU, given our setting and how we decided to use the warehouse. The most significant decision was to make a **management decision making resource** like the data warehouse. Most system developments at ASU support the needs of the operational user, failing to provide management the information they need for decision making. In designing ASU's data warehouse, we decided to focus our resources in addressing the needs of this important, but previously, ignored group of users.

Ten Lessons Learned

1. New technologies have shortcomings.
2. Costs are shifting to the customer.
3. Security and privacy are major issues.
4. Warehouse impacts data administration.
5. Training pays dividends.
6. Support structure needs to be in place.
7. Invest in a warehouse dictionary.
8. Officialness is hard to achieve.
9. Educate on warehouse concepts often.
10. Avoid cost justification if possible.

Figure 6. Lessons Learned

Technology Shortcomings

Client/server technology is still less reliable, secure, and timely than its predecessor. UNIX servers are not as reliable as mainframes and data access tools are just reaching adolescence. Networks add new layers of complexity and monitoring performance and tuning of servers is imperfect. The results are gaps in available technology and software leaving users' needs unmet. One such example is matching a cohort on a desktop machine with the data warehouse. Most query and retrieval tools do not support this type of function (local table join with server table). If the tool allows this function, joining data is slow, making the match process prohibitive for large databases. Allowing users to create tables containing the IDs of records being tracked on the server solves this problem. However, this solution defeats benefits of client/server technology, moving emphasis back to the host machine. The result is user frustration with the warehouse, when the problem is the technology. Also, with new technology, there is always new vocabulary to learn, adding further to the problem: client/server, relational databases, middleware, join, UNIX, decision support, SQL, TCP/IP, Ethernet, Cartesian joins, data administration, E/R models, ODBC, DAL, etc.

Customer Costs

Information technology departments and technology infusion funding traditionally absorb much of the cost of computing at ASU. With the warehouse and client/server computing, the cost of upgrading hardware and buying software for enterprise systems shifts to the individual or department. Employees seeking access to the warehouse need to know the cost of connecting. At ASU, a "connection checklist" is available, detailing all the steps necessary for access. The checklist includes information on these items: data access approval, PC or Macintosh, printer, Ethernet connection, communications software, data access software, software installation, and training. The checklist informs potential users about exactly what they need, how to get what they need, and how much it costs. We find this checklist to be a very helpful document (see Appendix B).

Security and Privacy

Client/server technology will ultimately force society to redefine privacy and organizations to rethink security. A data warehouse brings security to the forefront of this discussion, slowing development and frustrating users. Unfortunately, security and privacy issues may stall or limit development of a data warehouse at many institutions. Even though a data warehouse does not involve update capability, the ability to extract and convert groups of records to usable information is threatening. As a result, one of ASU's early initiatives was to develop a new data access policy that recognizes the value of placing data in the hands of our customers. There will be problems, but training and accountability are the most appropriate ways of dealing with this issue at the present time. At ASU, there is only one case in which we revoked warehouse access due to misuse.

Data Administration

Data administration at ASU has followed the evolution of data administration according to Bill Inmon. Inmon says the data administrator's role has changed dramatically from managing the data dictionary to designing and constructing a data warehouse.[8] ASU's data warehouse put the Office of Data Administration on the map and brings a new awareness of enterprise data. Users do not believe how bad their data is until they see it. For example, one college uses the data warehouse to verify professional program information and correct mistakes on ASU's operational systems. However, the data warehouse is a double-edged sword for Data Administration. Once users start using the warehouse, a "never-ending" list of enhancements quickly appeared, inundating Data Administration. Institutions need to identify permanent resources for warehouse development and support, or other data administration activities begin to suffer.

Training: A Good Investment

The need to train data warehouse users is critical and pays good dividends. In most computing projects, management recognizes the need for training, but does not always fund training. The is true of ASU's data

---

[8] Inmon, W. H. (1992, January). Winds of change: A brief history of data administration's amazing growth and development. Database Programming and Design, 68-69.

warehouse. With every new database there is a need for another training course, complete with reference materials. Every enhancement or change to the warehouse must be documented and communicated to warehouse users. Data Administration assumed responsibility for training and documentation at ASU. While training adds to the critical mass of warehouse development and helps our users, it distracts from development.

Initial training at ASU focuses on the *tool*, the *logic*, and the *data*. While a data warehouse supports hundreds of different access tools, training with one tool reduces a trainee's learning curve. After an extensive review of data access tools, Brio Technologies' DataPrism (an access tool that works in both the Macintosh and Windows environment) is our access tool of choice. Logic training is important also (i.e., SQL operators, Cartesian join, etc.). While this functionality is inherent in most access tools, training on query logic avoids many questions down the road. Lastly, training centers on the data. Data is what users know the least. We spend up to 60% of class time training on data, and hope to increase this percentage as users become more familiar with access tools and query logic.

<u>User Support</u>

While training reduces the number of data warehouse questions, a support infrastructure handles other support needs. At ASU, there is an e-mail address (ware-q@asu.edu) where users can send their questions or problems. Experts on warehouse data, networking, and data access tools receive these messages and respond within 24 hours. We log responses in a searchable database. Also, users can telephone a central help line that will send an e-mail message for them. Second, there is a file transfer protocol (FTP) site available for warehouse users. This site stores postscript copies of all documents associated with the data warehouse and copies of the data models. This is also a site for sharing common queries built by users or the warehouse team. Lastly, there is a Warehouse Users Group (WUG). WUG meets monthly to share findings, educate members about the data warehouse, and provide feedback to the warehouse team (currently there are over 75 WUG members). WUG also gives warehouse users an opportunity to find a "warehouse buddy," so they don't feel alone in ASU's world of data.

<u>Invest in a Warehouse Dictionary</u>

One of the more daunting tasks is to provide users a good data dictionary and source for metadata. [*Metadata* is data about the data, including layout, format, encoding/decoding algorithms, domain constraints, etc.] The problem is that there are thousands of data elements, and populating definitions and metadata is an endless task. Although this process is time consuming, the dividends paid are significant. At ASU, we draw resources from the BAIG to populate definitions (see page 4 for explanation of BAIG). [Authors' note: One of the reasons we recommend users adopt DataPrism as their access tool is the "remarks" feature. This feature functions like a pop-up data dictionary, allowing users to quickly determine the definition and code values of a data element or table in the warehouse.]

<u>Officialness</u>

Providing "official numbers" in the data warehouse greatly improves warehouse credibility. However, delivering "officialness" is not as easy as it sounds. The programs that extract and transform the data from the legacy databases must produce numbers that balance with the official numbers released by ASU. Since different algorithms and extract programs exist, there are often differences between the warehouse and official university reports. The problem multiplies because of ten years of data in the warehouse. Creating and validating ten years of official data is difficult. Going forward in time when building a warehouse is easier than attempting to reconstruct and validate history.

<u>Data Warehouse vs. Administrative Systems</u>

Many users tend to look at the data warehouse as another administrative system. This phenomenon happens since the data warehouse is in relational format. While the warehouse can makeup some of the data shortfalls operational user's experience ("data gaps"), it is not the warehouse's primary role. To help our users understand the difference between the data warehouse and their administrative system, we developed Figure 7. This figure compares a data warehouse to an administrative operational system on a variety of dimensions. Every talk or presentation on the data warehouse includes this slide underscoring the differences between the two. We reiterated these differences frequently, or our users begin to make unreasonable requests of the warehouse.

Data Warehouse
- **data is read-only**
- **serves management**
- **"time fixed" data**
- **"what if" processing**
- **data driven**
- **response.....minutes**

VS.

Adminstrative Systems
- **data is updated**
- **serves operational users**
- **"current value" data**
- **processing is repetitive**
- **requirements driven**
- **response.....seconds**

Figure 7. Data Warehouse vs. Administrative System

Cost Justification

If possible, avoid the traditional cost/benefit analysis in justifying a data warehouse. Since a data warehouse benefits the entire organization, ascertaining the benefits from improved decision making is difficult. Fortunately, at ASU, a limited demonstration of the warehouse concept was enough to sell the project. If a more complete cost/benefit analysis were required, the project may never have started. In other words, don't spend too much time justifying a warehouse, just start building one! [Authors' note: A data warehouse may be inevitable, since there is little chance that a technical breakthrough will occur, making access of legacy data easier or cheaper than a data warehouse. The Gartner Group says "Organizations employing a data warehouse architecture will reduce user-driven access to operational data stores by 75%, enhance overall data availability, increase effectiveness and timeliness of business decisions, and decrease resources required by IS to build and maintain reports."[9]]

Conclusion

The future of ASU's data warehouse is just beginning to take shape. Initially, the warehouse served as a resource for accessing information from legacy systems. Now, the warehouse fills a vital role in a client/server environment as a telescope into ASU's distributed data stores. Some of this data will reside in the data warehouse, while other elements will be "viewed" from the RDBMSs where the data resides. We foresee a time when the telescope extends beyond ASU to other institutions with common goals, such as the Maricopa County Community College District. The real power of the warehouse will be actualized in years to come.

The data warehouse also fills an important data administration role in a client/server environment. As distributed application developers move further away from the central computing core, the data elements in the warehouse insure the integrity of the institution's enterprise data. The definitions and coding standards in the warehouse are what distributed developers follow. The warehouse is the "glue" holding enterprise data stores together until a mature repository comes along.

The most important contribution of ASU's data warehouse is the new focus on data integration. While attempting to achieve integration in the warehouse, ASU conceived a new data model which not only integrates the

[9] Gartner Group (1994). [Proceedings from Conference Presentation on Data Warehouse.]

warehouse, but our administrative operational systems as well. By integrating the warehouse, we obtain more powerful data. By integrating our operational systems, we provide strategic new levels of customer service.

The bottom line is that warehousing is here to stay. Data warehousing can give institutions the opportunity to "get their feet wet" in client/server technology, distributed solutions and RDBMS. This is essential for any future mission-critical application. A data warehouse is a low risk, high return investment. The question for corporations and higher education is not simply whether to build a warehouse, but when. Based on predictions by Peter Kastner, an analyst at the Aberdeen Group in Boston, "All companies will build [a data warehouse] in the next five years."[10]

[10]Brandel, M. (1994, October 31). AT&T GIS dives into vertical markets. Computer World, 4.

References

Brandel, M. (1994, October 31). AT&T GIS dives into vertical markets. Computer World, 4.

Cafasso, R. (1994, October 10). Praxis forges data warehouse plan. Computer World, 32.

Gartner Group (1994). [Proceedings from Conference Presentation on Data Warehouse.]

Hammer, K. (1994, September-October). Will the data warehouse be warehoused? Relational Database Journal, 32.

Hackathorn, R. D., & Inmon, W. H. (1994). Using the data warehouse. New York: Wiley-QED Publication.

Inmon, W. H. (1992). Building the data warehouse. Boston: QED Technical Publication Group.

Inmon, W. H. (1992). What is a data warehouse? (Tech Topic Vol. 1., No. 1) Prism Solutions, Inc.

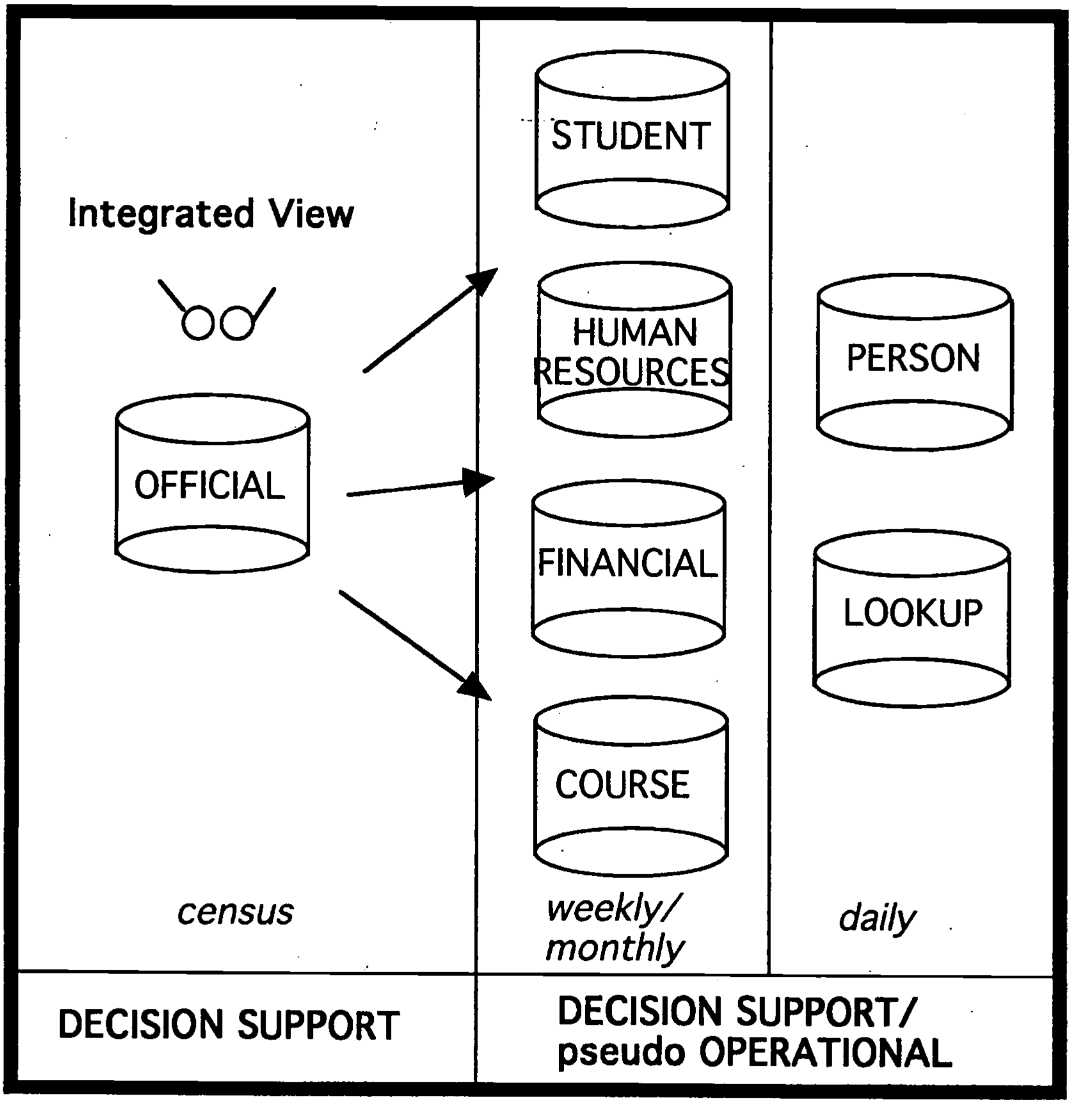
Inmon, W. H. (1992, January). Winds of change: A brief history of data administration's amazing growth and development. Database Programming and Design, 68-69.

Inmon, W. H. (1992, April). Building the data bridge: The ten critical success factors of building a data warehouse. Database Programming & Design, 69.

White, C. (1994, December). Client/server obsession. Database Programming and Design: Special Supplement, 7.

Appendix A

ASU DATA WAREHOUSE



Databases in ASU's Data Warehouse

Appendix B

Arizona State University

ASU Data Warehouse

Connection Checklist

ASU DATA WAREHOUSE

Revised August 1, 1994

♦ Introduction

This checklist is designed to assist you in determining if you have everything necessary to access the ASU Data Warehouse. Some of the information included in this checklist is, by necessity, of a technical nature. We encourage you to call your local or distributed consultant or 965-CNCS if you have any questions related to this document.

The Warehouse requires a number of components that are described in detail in this document.
The checklist includes the following items which must be considered when connecting to the ASU Data Warehouse:

Data Access Approval

- PC or Macintosh
- Printer
- Ethernet Connection
- Communications Software
- Data Access Software
- Software Installation
- Training

After reading this document, please complete one of the attached worksheets for a PC or Macintosh. This will assist you in estimating the costs associated with connecting to the Warehouse. Please keep in mind that the prices listed in this document are estimates only and should be verified based on current pricing structures.

♦ Data Access Approval

To obtain read-only access to the ASU Data Warehouse, you need to do the following three steps:

1. Complete the attached Request for Access to Computing Facilities form, **omitting your signature** in the "Owner Signature" section. A sample form is attached as a guide to completing the request. You will sign the form in training class after the instructor explains security issues.

   For student data: Graduate students requiring access to the Warehouse as part of their ASU employment will require signature from their department sponsor.

   Undergraduate or graduate students not employed by ASU will not be given access to the Warehouse.

2. Obtain supervisor's signature in Sponsor Signature block.

3. Forward the form to the Registrar's Office (Student Data Trustee) for approval (mail stop 0312, SSV B121).

Your userid and password will be established and issued to you by the following process:

Data Trustee will approve the level of access and forward the approved form to the Data Administration Training Instructor.

Data Administration Training Instructor will forward the form along with forms for other students registered for a specific class date to the Computer Accounts Office.

Computer Accounts Office will create your userid and password, and then forward the form back to the Data Administration Training Instructor.

To activate your userid for use with the ASU Data Warehouse, you will need to attend an introductory training class (Accessing ASU Data Warehouse) to become familiar with the Warehouse. Please refer to the section below for training information.

Data Administration Training Instructor will explain security issues in class and obtain your signature on the Request for Access to Computing Facilities form. The instructor will then give you a copy of the form and forward a copy to the Computer Accounts Office.

♦ PC or Macintosh

A PC or Macintosh is required to access ASU's Data Warehouse. Both minimum and recommended configurations for PCs and Macintosh computers are listed here for your reference. These configurations are appropriate for compliance with the ASU Rational Information Technology Environment (ASURITE).

PC Configuration

	Minimum Requirements	**Recommended Configuration**
Processor	386	486
Memory (RAM)	4 MB	8 MB
Disk space	40 MB	230 MB
Monitor	EGA, color	VGA, color or mono
Mouse	Yes	Yes
Keyboard	Yes	Yes
Operating system	DOS 5.0	DOS 6.2
Windows version	3.0	3.1
Ethernet card	Yes	Yes

Macintosh Configuration

	Minimum Requirements	**Recommended Configuration**
Processor	68020	68040
Memory (RAM)	3 MB	8 MB
Disk space	40 MB	120 MB
Monitor	RGB, color	RGB, color
Mouse	Yes	Yes
Keyboard	Yes	Yes
Operating system	System 7.0	System 7.1
Ethernet card	Yes	Yes

♦ Printer

With access to the Warehouse, you are most likely going to be interested in printing your reports and diagrams of the Warehouse databases. In order to accomplish this, you will need access to a laser printer. If you are presently connected to a local area network, such as a Banyan, Novell or AppleTalk network, there may already be a laser printer that exists in your office area that can be used for warehouse printing. If you are not connected directly to a local area network and do not have a printer, we suggest that you consider purchasing a laser printer. In order to print the diagrams, you need a printer that can print postscript files.

♦ Ethernet Connection

An Ethernet connection is a requirement for accessing the ASU Data Warehouse. Ethernet allows you to connect to other computers and printers on campus and provides high speed communications. Ethernet has become a standard for new data communication connections on campus and is rapidly becoming wide spread throughout all university departments. If you already have an Ethernet connection, you may proceed to the next step. If you have a connection such as dial-in, Kermit, Forte, Irma or LocalTalk, you will need to obtain an Ethernet connection in order to access the Warehouse.

To obtain an Ethernet connection, you will need to do the following:

1. Obtain a copy of the Application for Data Network Connection. This form is available from COMPASS at Computing Commons, 2nd Floor or by calling 965-5939.

2. Complete the form. There may be questions on the form which you cannot answer. The Data Communications department, at 965-5911, will assist you in completing the portions of the form that are not clear. Also, they will provide current up-to-date pricing, so that you may complete the necessary PO9 for payment.

> NOTE: Page 2 of the Application for Data Network Connection will ask if you need an Ethernet interface card. If you presently do not have an Ethernet card, we recommend that you request services of the Tech Shop to provide and install the card. When you contact Data Communications, specify that you will need an Ethernet card so that you can complete one PO9 for the connection, card and installation.

> NOTE: Page 2 of the form also asks questions pertaining to the installation of the NCSA/Telnet software. When completing this form, please indicate that you will not be using the NCSA/Telnet software. If you are using a PC, you will need the LAN Workplace software which is described in further detail in the Communications Software portion of this checklist. If you are using a Mac, you will be using MacTCP, which is available through COMPASS.

3. Submit the completed form to Data Communications with a printed screen from the CUFS system showing that you have obtained Level 1 approval for the PO9.

4. A representative from Data Communications will make the necessary arrangements for the Ethernet connection and installation of an Ethernet card, as specified in your application.

♦ Communications Software

In addition to the Ethernet connection described above, communications software is required for your computer. There is a ***minimum*** of two communication programs required for using the ASU Data Warehouse:

1. TCP/IP (also known as Telnet) software is required for you to use your Ethernet connection.

If you are using a Macintosh, this software is included with System 7 and is called MacTCP. If you do not already have a copy of MacTCP, you may obtain this software through COMPASS, at the Computing Commons.

If you are using a PC, we recommend use of the ASUNET program for your TCP/IP software. ASUNET is available through Compass in the Computing Commons, 2nd Floor. If you are currently using ASUNET, you will need to obtain the upgrade to a version dated February, 1994 or more recent. This up-to-date version contains a program called WINSOCK which is required for accessing the Warehouse.

Alternatively, you may use FTP PC/TCP or Novell LAN Workplace on your PC as the TCP/IP package of choice.

2. A program known as Sybase Open Client, Net-Library is also required to access the Warehouse. This program is available through COMPASS for the PC and Macintosh environments. Be sure that your Ethernet connection is installed and working prior to attempting to install this software.

3. Another program, Sybase Open Client, DB-Library may also be required depending on the nature of the data access software that you are using. This Sybase Open Client, DB-Library software is only applicable to PC users and is not required for the Macintosh. This software is required if you choose DataPrism for the PC, but is not required for the Macintosh version of DataPrism. If you select some other product for data access, such as SAS/Access, Microsoft Access, Q+E, Forest and Trees or PowerBuilder, you will need to consult with the vendor to determine if the Open Client DB-Library software is required to run their application program. If the Open Client DB-Library software is required by the vendor, you may purchase this software at COMPASS.

♦ Data Access Software

Several data access software products have been researched by the staff involved in setting up the ASU Data Warehouse. While we have found many to work very well in providing access to the Sybase server supporting the ASU Data Warehouse, DataPrism is the recommended tool. The features of DataPrism are summarized as follows:

- Easy to learn, requiring minimal training to get started
- Useful for generating simple ad hoc queries and reports that may be viewed or printed
- Useful for exporting data to other applications, such as Word, WordPerfect or Excel
- Rapid report generation and execution
- Supported for the Data Warehouse environment

DataPrism is available through COMPASS, at the Computing Commons. Special pricing has been negotiated for this product. Please refer to the attached worksheets for pricing information.

NOTE: There are numerous products available through software vendors that will allow you to access the Data Warehouse. Each of these products offer advantages and characteristics that differentiate them from each other. Several of the products we evaluated for use with the Data Warehouse have problems which effect the usefulness of the product. If you are considering using a tool other than DataPrism, please contact a consultant by calling 965-CNCS or by sending an electronic mail note to WARE-Q to inquire about the product that you are interested in.

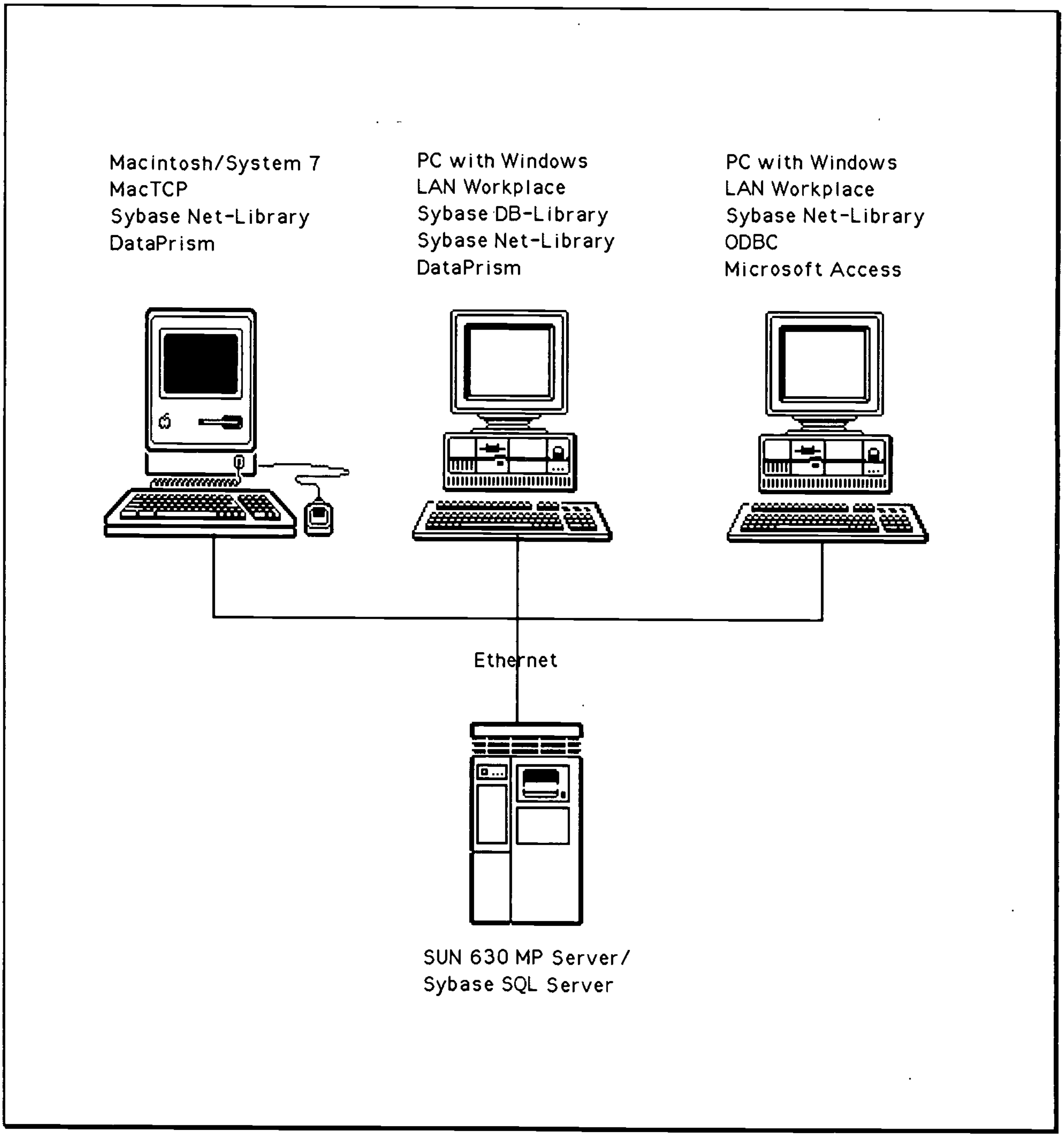
♦ Software Installation

The communications and data access software described in this document is complex and involves configuration options. While there are instructions included with the purchase of these software packages, we encourage you to contract with Facilities Management to perform the installation and configuration of this software. There is a charge of $40 per hour for this service. Depending on the complexity of your configuration and the components required to be installed and configured, this may take from 1 - 4 hours for a Facilities Management technician. To arrange for this service, contact Facilities Management at 965-2826.

♦ Training

After you have all of the pieces in place (data access, PC or Mac, printer, Ethernet connection, communications software, and data access software) you are ready to be trained. Refer to ASU Data Warehouse Training sheet available in COMPASS, Computing Commons 202 on Main campus, or by contacting the IRT Helpline at 543-4357 on West campus.

ASU Data Warehouse Connection Overview



ADMINISTRATIVE SYSTEMS VENDORS TALK TURKEY ON CLIENT/SERVER COMPUTING

John Stuckey

Washington and Lee University

Everybody talks "about" client/server computing, although the careful listener notices that they frequently mean different things by the term. In this panel, product-development managers for three prominent administrative-systems vendors will discuss their products' evolution and their company's strategy. All will refer to the Gartner Group's classic "Five Styles of Client-Server Computing." We drew lots to determine order of presentation.

1. Datatel: Laird Sloan, Director of Product Development

2. SCT: Roy Zatcoff, Vice President, Product Development

3. CARS: Duane Burris, Vice President for Research and Systems Development

After the vendor presentations, Grey Freeman of the Gartner Group will comment.

DATATEL IN THE CLIENT/SERVER ENVIRONMENT

Datatel, Inc., Fairfax, Virginia

Goals

To give our customers and prospective customers an option to choose the client/server style that is compatible with their technological direction and budget.

To protect our customers' investment in computer hardware and infrastructure.

To preserve and enhance the functionality of our existing Colleague and Benefactor software by enabling them to perform in a client/server, character-based or hybrid environment.

To use "open" and "standard" desktop software wherever possible.

To use our CASE tools as the enabling software to move to the client/server environment.

Client/Server Styles

Datatel is currently supporting the Distributed Presentation, Remote Data Management and Distributed Logic styles of client/server, as defined by the CAUSE/Gartner Group white paper on client/server computing.

Distributed Presentation Style

Distributed Presentation is the style where the database management and application logic reside on the server and the user presentation processes are divided between the client and the server. Datatel introduced a graphical user interface (GUI) for this ..style.

The client display technology is based on wIntegrate, an MS Windows-compatible desktop product.

Envision, Datatel"s CASE tool, was enhanced to generate GUI and character-based displays.

All existing application software developed under Envision can be upgraded to GUI through a simple regeneration process. (This includes client-developed applications.)

The enhancement to GUI preserves the use, style and training associated with our current products

The GUI technology supports DDE and Clipboard for interfacing to third-party Windows applications on the client.

The GUI supports multiple terminal emulations using serial ports or TCP/IP; also allows display of report formats (132X66) on a screen.

The GUI includes desktop functions, such as full mouse support, window and dynamic font resizing, icon bars, full color support, dialog boxes, push buttons, list boxes, radio buttons and images.

Multiple server sessions can be supported.

The clients includes extensive script support, executable from the client or the server.

The client supports an online, user-prompted query builder with the ability to import the results into Windows applications

Remote Data Management Style

Remote Data Management is the style where the database management resides on the server and the application logic and user presentation processes are located on the client. At CAUSE in 1993, Datatel introduced the prototype of TopView, our executive information system, formally released on October 10, 1994.

Datatel has created the schema to support executive information functions; the resultant tables (files) can support any SQL-based front end or analytical tool.

The client application is based on a decision support system, Forest & Trees, a product of the Trinzic Corporation.

By generating SQL queries to the Colleague or Benefactor database on the server.

TopView can create application views from multiple servers using SQL and clients using DDE.

A variety of user presentation tools are available on the client, including; tables, charts and graphs; buttons, list boxes and bitmaps; cross-tabulation matrices; printed and on-screen reports.

The applications on the client can incorporate important user alert tools, such as visual and audible alarms and alarm triggers.

TopView supports real-time, system-generated and user-demand calculation functions for the application views.

The applications can use drill-down functions from the top view to a specific view of the information, providing the option to present the data in both tables and charts at each level.

Datatel has created EIS applications for each major system of Colleague: Alumni & Development, Financial, Human Resources, and Student.

Customers can use TopView to generate their own EIS reports and application views to support the unique needs of their institution.

The system includes a full security system for the client applications.

Distributed Logic Style

At CAUSE in 1994, Datatel is introducing its comprehensive Distributed Logic client/server system.

All Colleague and Benefactor processes can run on the client or the server.

The current version supports UNIX on the server and MS Windows on the client.

The system can simultaneously support client/server computing and a traditional character-based environment.

All Datatel and institution-developed software, created under Envision, can be upgraded to the client/server environment through a simple regeneration process. (Envision, itself, is also generated in client/server mode.)

Envision creates user presentation processes using Microsoft design guidelines and is fully compliant with Windows standards.

The system includes a full online "help" capability using the Microsoft paradigm.

Communications between client and server are handled via ODBC for future compatibility with other systems.

The system supports the Cut-and-Paste, Drag-and-Drop and OLE interfaces between the Datatel and PC applications on the client.

Full automatic client version control to simplify system administration is a feature of the system.

The institution has the option to distribute the application software processes to either client or server to optimize their unique computing environment.

SCT APPROACH TO CLIENT/SERVER PARADIGMS

SCT, Inc., Malvern, Pennsylvania

The SCT BANNER products support all 5 styles of client/servercomputing as defined by the Gartner group. BANNER can be installed such that the server only provides database services or a portion of database services (distributed), database and cooperative logic, database and all of the logic, or database logic and presentation service. The client may also contain one or more of these services depending on the BANNER installation options. More importantly, BANNER supports a concurrent mix of all 5 styles. One user may have presentation and all logic services on their client machine. One user may have presentation and all logic services on their client machine which communicates to the database sertver, while another user may be using cooperative logic and/or cooperative presentation services on both their desktop and the server.

SCT software also supports the current popular GUIs (MAC, Windows, MOTIF, X) as well as character cell devices with one code set, and therefore, can also be deployed in a mixed environment. This allows for a controlled or staggered expenditure for both clients and servers, since they may be introduced over time while everyone still uses the same application and data. The software may also be deployed in a multi-tier client server approach where presentation servers, application servers, and database servers are all used as desired or required. Additional servers for mail, security, imaging and others can work in concert with BANNER client/server applications. The BANNER client/server architecture is extremely robust and flexible, allowing an endless variety of client/server deployment scenarios ranging from very simple architectures to very sophisticated infrastructures. These decisions can be made while enjoying the benefits of a single code set with portability and deploying the portions of the applications to client and servers as is best for an institution.

YOUR PRACTICAL GROWTH PATH TOWARD CLIENT SERVER

CARS Information Systems Corporation, Cincinnati, Ohio

Introduction

One of the most revolutionary advancements in computing today is the emerging of client/server architecture systems. However, the client/server paradigm could also be considered a practical evolution of two other rapidly growing architectures in computing today, namely: a) the widespread use of PC systems on the desktop designed to enhance productivity and b) the rapid growth in capability of minicomputers containing relational database engines and serving as the institution's main data processing system. From this perspective, the development of client/server systems is simply the logical outgrowth of the institution's desire to merge two existing disparate systems which have already been deployed, leading to major enhancements in the operation of each.

I. Technological Advancement

Categorized from the application perspective, the Gartner Group distinguishes five styles of client/server computing models. As mentioned, in the Gartner Group Research Notes August 10th "What Lies beyond the "Five Styles of Client/Server Computing", from an implementation perspective many businesses find the five styles too simplistic. Institutions decide to purchase a system from the very practical perspective. They may have a huge current investment to protect and, at the same time, a limited budget for change. However, because of their current system limitations and pressures from their users, they must change. MIS personnel feel frustrated trying to meet the demands of increased capabilities with limited resources. Dire necessity may prove to be the real mother of invention here in moving to a new system.

II. The Practical Path to Client/Server

In considering the college computing scene, the CARS view is "Your predicament is our challenge". The problems institutions face in moving to C/S represent the primary challenge to CARS as a vendor of C/S systems. Our responsibility is to design and build a client/server administrative system that institutions can not only move toward but also afford. In CARS' view, any movement away from traditional host-based computing toward C/S systems must take cognizance of the resources available to the institution. These resources include the host computer, campus-wide network, existing PC's, software, and, very importantly, the skills of the users. The actual types of systems in place today fall into four distinct categories:

1) Dumb terminals attached to host computers.

2) Dumb terminals attached to terminal servers which in turn attached to a network backbone.

3) PC's acting as dumb terminals attached via serial lines either to a host or to terminal servers.

4) PC's with full network access to the host computer. Of course the actual systems may well be a combination of several of these categories. Institutions in category one have a long, expensive path ahead of them to migrate to a C/S system. On the other hand, clients in category four are nearly there except for the C/S software itself. When constructing a strategic plan for technology addressing its Information System needs, the institution must consider the current status of the campus in its migration.

III. Toward Client/Server

When moving the CARS System toward the Client/Server architecture, we have identified three major tasks within product development for utilizing the PC as the client workstation and the UNIX host as the server. We will first provide the Graphical User Interface (GUI) on the PC as the user interface to the application software within the CARS System running on the host. The PC will be used as a workstation for accessing CARS applications versus being used as a dumb terminal. The GUI platform can be either Microsoft Windows or the Apple System 7 environment. With this GUI front-end, users can access the CARS System utilizing the GUI standard GUI features. The user can use the mouse to select from a set of fields the particular field they wish to update. The user is no longer restricted to a series of up and down arrows to move to the desired field on the screen. With the use of multiple windows within the GUI, the user can click the mouse on the window of the desired application program to be reactivated. The user is no longer required to suspend one application and select through a series of keystrokes the next application to be reactivated. The user is able to use the vertical scroll bar to select an entry in a table that is being displayed in a separate window. The user is no longer required to page through the entries of a table with a series of keystrokes. Furthermore, users prefer the visual appearance of information using the GUI display over the character display.

A "Windowing" PC user appreciates the standard GUI functionality provided within the host-based application processing of the CARS System. This is achieved within the CARS System through the use of the winsock protocol to communicate over the network to a GUI presentation server. The CARS GUI server running on the PC receives the information necessary to present the GUI display from the host. Gartner Group labels this style of client/server computing as "distributed presentation". This approach makes the first step into client/server supportable by both CARS as vendor and you as the client. First, there is only one set of screen definition files to be maintained for both the character and GUI displays. Second, since the same screen definition file is used for both character and GUI displays, the user of the CARS System within even the same office on a campus can be operating the same application software on a combination of terminals and PC's. If the client happens to have a campus-wide system which is predominately in category four above, then the cost of making this step will be primarily the training costs.

The second task is to provide direct access to the database on the server from the productivity software available on the PC. For example, the user of the spreadsheet expects the data elements within the database on the host to be readily accessible from the desktop PC. The user does not want to hear the words "import/export" any more. Since both the productivity software within the office suites and the Informix On-Line Database Engine are supporting the Open Database Connectivity (ODBC) standard, it is finally true that the spreadsheet can issue an SQL (Structured Query Language) statement as a query directly against the database on the host to access data elements for a worksheet. Thus,

the data within the CARS System which is maintained by the On-Line Engine is now directly accessible by the third-party software that is ODBC complaint without "import/export".With the direct access to the database by third-party software, there is the need to maintain the integrity of the database. The "foreign" third-party software does not have the knowledge of the implied structure and constraints of the data elements which were incorporated within the "native" application software that accesses the database (nor the corporate business rules). Thus, the database must be expanded to incorporate this additional information along with the actual data in order to enforce the implied logical operations on the data.

The third task is to code the software, one module at a time, to deliver distributed processing, true C/S applications where the host computer serves as a database server while much of the processing occurs elsewhere. This processing can be done on the workstation or on additional application servers. With hardware costs dropping as rapidly as they are, one could, in fact, envision multiple servers such as a report server which just runs reports on demand. The distribution of the processing load of the administration

applications is a means of addressing the overall cost/performance of the system. When it is both cost effective and feasible, the applications should be running on the client workstations or an application server. It must be recognized that the users are only interested in the perceived performance of the overall system. They do not care about the location of the executing software. The users desire that the functionality is there with a fast response time.

Conclusion

CARS is creating a client/server system that allows our clients to undergo a gradual, cost effective migration using the benefits of their existing systems. During the migration, the users on the client sites will benefit from the inherent attributes of the PC as a workstation. These include the GUI and the productivity software which are familiar tools for the user to use in conjunction with the CARS system.

Migrating from Host-based Computing to Client/Server

Jay Blum
Computer Center Director
Thomas More College
Crestview Hills, Kentucky
and
Richard Reno
CARS Information Systems, Corp.
Cincinnati, Ohio

Abstract

Thomas More College is working to successfully evolve the CARS System from a host-based computing system to client/server architecture. The CARS System provides the institution with the ability to migrate individual offices or office users at a time best suited for them, independently of the migration plan for other offices or users of the administrative system. Technologically, the key benefit is the PC utilized as a workstation on the administrative system versus as a character-based terminal through a terminal emulation package. Thus, the workstation provides for the following:

- Initial balancing of the processing load
- An easy-to-use interface to the host-based applications
- Efficient and effective PC tools that are familiar to the user

This case study of Thomas More College involves the prototyping of the practical path from host-based computing to the client/server architecture of the CARS System. The thinking and the planning for the right architecture at the right time are critical. The implementation and support of this architecture must be feasible for both the institution and its personnel. The appropriate architecture allows the Computer Center to realize the benefits of using the client workstations with the server/host. The CARS System, running on a host computer, acts as the server for the workstations, which are acting as clients. This enhances the users computing resources and CARS as the vendor server enhances the computing of the institution as a client.

Introduction: Partnering the Clients and the Servers within the Campus Information System

One can hardly pick up a computer magazine today without seeing the term "client/server," and many institutions are evaluating the possibility of moving toward that type of system. We hope that our experience at Thomas More College (TMC) will be informative for other institutions considering moving toward client/server architecture.

For our continued discussion, we define "client/server" as follows: The "server" is the computer where the data and maybe the executable program resides. The "client" is the PC or workstation where the executable program is loaded at the time of execution. This client only accesses the server when the program loads or when there is a need to access data.

For Thomas More College, this process began in February of 1991 when the President commissioned the VP of Finance and Administration, to form a task force to develop a long range plan for computing on campus. The group was made up of computer center personnel, faculty, staff, students, administrators, and representatives from IBM. At the time, the TMC administration was running CARS software on an HP 9000/832 connected via serial connections. Academic Computing was using a Micro VAX with twenty-one networked PCs.

As members of the task force, we first assessed the current situation at TMC. We distributed a written needs assessment survey to faculty, staff and students. Based on the responses from that survey, we developed a list of campus needs. From this needs list, we determined the consequences for the campus if these needs were not addressed. We then met with users in small groups to get feedback and begin finding solutions. Armed with this user information, the group then developed a solutions list. With this list, we proceeded to list the benefits. From this work, TMC developed a three-year plan for the installation of a campus-wide network. Total cost of the project was estimated at $1.5 million.

In 1991, TMC was awarded a $750,000 grant to begin the installation of a campus-wide network. Installation began in May of 1992. TMC installed 1400 individual connections using Level 5 UTP, with at least two connections in each room, office or classroom. Some offices and classrooms had as many as 25 connections. Installation included the Computer Center's central concentrator and five (5) other concentrators on the campus connected via a fiber optic backbone.

The grant also allowed us to upgrade the PCs on campus. When examining the direction of computing, we found we needed powerful workstations. Our original plan called for the purchase of forty (40) 386 PCs for users. As an example of how fast technology and pricing can change, in only eight months from the plan's acceptance to completion, we ended up purchasing sixty-seven (67) 486 and thirty (30) 386 PCs for the campus. Forty-six (46) of the 486 PCs went to student classrooms and labs, and the rest to staff and faculty offices. Because of this installation, we were immediately able to consider moving our administrative computer system to a client/server system.

Because of our network, CARS Information System Corporation asked us to be their Prototype Site for the Graphical User Interface (GUI) for the CARS System. CARS has developed a two-step approach to moving to client/server architecture. The first step is to convert the interface for the user from the current character-based interface to the Graphical User Interface (GUI). During this phase, the programs still reside and load on the central computer, whether mini or mainframe. The user interface changes to GUI, allowing the users to learn to use the Windows interface without a major disruption of their work. Step two would be the expanded migration to the client/server architecture.

I. Thinking: The Current Issues of Evolving Computing for a Small Institution

When thinking about the direction in which campus computing should evolve, the following important issues must be considered:

- The available technologies, both in the present and in the immediate future
- The limited resources that the institution can allocate to the project
- Any proposed change should be developed within the framework of a strategic plan for information technology utilization
- The current use of PCs by the staff, since these may be considerable

A. Consideration of the New Technology within Computing

When considering an upgrade for your institutional computing environment, consider the rapid rate of development in technology today. Administrators have a hard time understanding that something that was top of the line only two or three years ago is now literally unusable to carry out current tasks. To ensure that fundamental needs are discovered before implementation, the institution must consider the rapid rate of technological changes in its plans. Because hardware prices are plummeting so fast and the technology is developing so rapidly, an institution benefits from lower prices overall or greater computing power through the implementation cycle.

The most important aspect of considering new technology is to try to foresee the future as much as possible. Institutions must also research new emerging technologies to decide whether they will be important to the institution within the lifetime of the current plans. Good technological examples today are real-time audio and video transmission across the campus networks and presentation of the transmissions to the desktop. This technology promises to be very important to academic institutions. Other technologies of interest include document storage and virtual reality systems.

We believe that the single most important component for any computing model today is the campus backbone network. An institution could get very detailed in terms of anticipated network usage to decide which cable to use. However, if such consideration is based upon current usage plus reasonable growth estimates, and if the network is expected to last ten years, then any such detailed consideration will surely be wrong. These considerations do not consider the future technologies involving the distribution of images, audio, and video across the network. The network usage demanded by these technologies will dwarf the use of the network for "standard" computing operations.

B. Realization of the Limited Resources Allocated within the Institutions Budget

Most academic institutions today are facing shrinking budgets, while simultaneously receiving increasing requests for computing resources from various constituencies. The requests become even greater after a network is set up, and users begin to see what others can do. When the user base encompasses most of the campus, computer center staff can more easily justify purchasing the necessary computing resources. The institution must consider realistic funding estimates when planning. Because Thomas More College experiences the same budgetary constraints as many other colleges, we decided we needed to incorporate our existing resources when moving forward.

C. Recognizing the Need for a Strategic Plan for Information Technology

To change an institution's computing environment, a large segment of the campus community must support it. One way to accomplish this goal is to provide a strategic plan for information technology. In preparing a strategic plan, the computer center must solicit opinions from as many constituents as possible. When users participate in the process, they are more likely to support the plan. If most of the campus supports the IS strategic plan, administrators are more easily able to carry out components of the plan.

D. Using PCs within Administrative Processing

As previously mentioned, TMC started by evaluating the current campus situation and developed a three-year plan. Because of the participation of the whole campus, we had a great deal of support for carrying out our plan. In the plan, we recognized that utilizing PC workstations would be an important component. For budgetary reasons we decided initially that only users using wordprocessing or spreadsheets on a daily or weekly bases would require a PC. The grant allowed us to expand usage with our original purchase, but we still need about seventy (70) PCs to meet the demand.

II. Planning: The Right Architecture for Administrative Computing on the Campus

In planning for an evolutionary change in the computing environment, the architecture of the new system must be considered to find the best fit with the current resources and expertise. Capitalizing on users particular computer expertise and using tools with which they are familiar, the system must effectively tie together the users (clients) with the information sources (servers) for a system to be successful.

A. A Plan for Integrating the Clients and the Servers into Administrative Processing

At TMC, the primary server within the administrative system is the administrative minicomputer. TMC currently has an HP 9000/G30, with HP/UX operating system, running the CARS administrative software package. Before the installation of the network, there were about four users, which was 20% of all users, using a PC workstation with a terminal emulation package, while the rest used character-based terminals. After the installation of the network and the PCs, the percentage of users increased to more than 50%. This increase PC use allowed us to begin looking at client/server.

Because the CARS System features an open system design, TMC could become the prototype site for the CARS System GUI software. This allowed TMC to retain the CARS software with which it was familiar and still move forward as fast as practical with the evolution to a client/server system.

B. Migrating Administrative Systems that are Fully Operational Systems Today

TMC wanted to migrate from the old system to this new technology without disrupting the entire campus. Besides requiring a substantial financial expenditure, such a migration would also entail a major training effort. TMC wanted the migration to take place over an extended period. To meet these concerns, CARS proposed the following two-step approach:

1 Allow users to experience some benefits of a GUI C/S system without requiring a complete changeover. With the graphical user interface, the campus can simultaneously carry out both traditional host-based computing and C/S computing.

2 Expanded migration to the client/server architecture.

C. Allowing the Institution to Evolve Over a Time Frame According to the Needs of the Institution

The primary new requirement for the CARS Graphical User interface is a PC operating as a workstation with a network connection. Because TMC had recently installed the fiber optic backbone on campus, the computing staff simply converted PCs to C/S ready workstations and hooked them to the network. This process can now continue as funds are made available until all CARS users are using C/S style workstations. Users have the flexibility to switch back and forth between terminal mode and GUI according to their individual preference.

III. Implementing: The Practical Path to Client/Server Computing Using CARS Graphical User Interface

Presently, TMC is well along the way to implementing CARS Graphical User Interface across the campus. Those people who are presently using this system, and who had previous PC background, are for the most part happy with the operation of the system. The MIS staff is in the early stages of learning to cope with the problems that will arise in a expanded C/S system. In particular, the staff is learning how to deal with the problems of more users accessing the network, and the increased load on the backbone.

A. A Graphical User Interface for the CARS System

The CARS Graphical User Interface is a CARS application specific tool that resembles an X-windows server. The GUI program server (currently only for Microsoft Windows) acts as a display manager for the software running on the central host. It uses a direct network connection to the host for communications. The CARS GUI allows the user to have multiple sessions active in different windows, move those windows around on the screen, and use a mouse for some selection operations. The most commonly stated benefit of using the GUI is that most users like the uniform operation and the reduced time hitting a return while doing data entry. "They love the point and click."

B. Integrating Familiar User Tools

Allowing the user to interface directly to the system using familiar standard PC tools is one strong argument for moving to the client/server model for the administrative system. The CARS Graphical User Interface allows TMC personnel to get some experience with this now and to prepare for a client/server architecture solution in the future. With the CARS GUI, the user can copy and paste into other GUI products. In addition, the CARS GUI is expected to support file transfer between the host and the user's PC. Users will be able to easily enter scanned images into the database (documents, signatures, photos etc.). Similarly, users will be able to retrieve these stored images and paste them into other applications.

C. Introduce New Software over an Evolutionary Timeframe

TMC hopes to eventually install the link software that will extend the database access to the PCs. One future example might include Q+E software that will link any PC software that allows OLE connectivity to have access to the database. In particular, TMC expects to use spreadsheets and third party report writers. We hope these report writers will relieve some pressure on computer center staff to create special reports. Frequently, the spreadsheet will be the natural interface to the database for the user: for example, applications running projections and statistics. Using a spreadsheet, users who are currently running reports on the system and uploading or hand-entering the data into a spreadsheet, will be able to add the queries in the spreadsheet to get data from the database directly.

IV. Supporting: The Feasible Approach in Computing Architecture for the Campus Personnel

As an institution evolves its computing environment, the computer staff endures great stress to support this change. When institutions try to cut corners in staff training during the change, the success of the change is at risk. It is surprising how many institutions of higher learning do not believe in education for their employees. In the move to the client/server paradigm many, if not most, of the staff may not yet know how to use a GUI interface. Training must be a high priority.

A. Implementing at the Right Time for Each Individual User

At TMC the conversion is now well underway. One distinction of the CARS System Graphical User Interface is the fact that the computer center staff only has to maintain one screen definition file for both the underlying host computer and the CARS Graphical User Interface. The user is still running the same program on the host that they used to run when using a character-based terminal. The application program senses that the PC is running the CARS Graphical User Interface at startup and modifies its behavior correspondingly. Because the programs adapt to the situation on the workstation, TMC staff can on an 'as desired basis' move users to CARS Graphical User Interface. For example, an individual office may be moved, an individual may be moved or an application may be moved. If for some reason users want to suspend using the Graphical User Interface they can turn off the GUI server program and rerun the program; the program will appear in character-based mode.

B. Updating the Administrative Software with Mandatory Changes from the Vendor

Recognizing that institutions are constantly changing, whether to meet regulatory changes or to adapt to institutional policies, CARS realized the Graphical User Interface would need the same flexibility as the rest of the CARS System. When enhancements arrive from CARS, the computer center staff need only install one version. Because of the way that the Graphical User Interface operates, the same screen information a character-based terminal utilizes is also used by the GUI server program. Therefore updates must be made only once.

V. Evaluating: The Benefits of the Practical Path in Enhancing Computing for the Campus

A. How Did the Process Go?

In evaluating this process at Thomas More, we found that the process worked very well. Realistically assessing where we were and where we wanted to go, TMC could map out a plan that incorporated the following:

- Available technology
- Our available budgetary resources
- Our current investments in technology
- Our strategic plan for information technology utilization

Because the MIS plan incorporated the needs of the entire campus, the campus as a whole had a commitment to its success. We found users taking ownership of the process, because they were actively involved. We believe this better equips our campus in preparing for a client/server architecture administrative system.

B. What Did We Learn Along the Way?

A major lesson we learned was that any strategic plan must be flexible, to meet the constantly changing needs of the institution and also the emerging technologies. For TMC, the network provided an excellent example of this needed flexibility. Because TMC installed a fiber optic network, the institution was immediately able to begin testing CARS Graphical User Interface. Through this testing, TMC was also able to see how use of the Graphical User Interface increased network load.

Because the CARS Graphical User Interface allows individual users to migrate on their own time schedule, TMC could carry out an evolutionary migration of the entire campus without disrupting its day-to-day operations.

C. What is the User's Perspective?

Our first user for testing the GUI was a data entry person. This person did not have any PC Workstation experience. After a week of Windows training, the user preferred using the GUI for entering information into the system. In the midst of this process, when the user switched offices, the first question to the Computer Center was, "Will I have the same GUI capabilities in my new office?" From Alumni and Development, another user used the CARS System testing the Graphical User Interface. This user prefers using the GUI, because she finds it easier to use. The data entry users like pointing and clicking on fields versus hitting a return during data entry. "They love the point and click."

D. What is the Computer Center Perspective?

Because both the GUI and the character-based CARS System are maintained by a single screen definition file, the GUI does not require any additional maintenance of the Computer Center staff. When TMC incorporates a new enhancement from CARS, these enhancements will be used in both character mode and GUI mode. Concerning training, we also found that users who were familiar with PC applications, were easily able to switch to using the GUI. For those users for whom the GUI was their first experience in PC capabilities, once they learned how to use it, they were able to easily learn other PC applications. While positioning us to migrate to a full client/server architecture in the future, TMC has found CARS Graphical User Interface to be a cost-effective solution that incorporates current resources while allowing users to enjoy the benefits of a PC workstation.