Payoff

Client/server technology is a complex and inherently difficult technology to implement successfully. Nonetheless, today's business problems demand that IS address and contain the risks and move forward in delivering client/server solutions. This article discusses technical issues in implementing client/server business solutions and how to organize the many parallel work efforts that need to be undertaken.

Problems Addressed

During the past 25 years, no computing technology has emerged as quickly and grown to such a dominant position in providing business solutions as client/server computing. This is evident in the emergence of the new leading vendors in the computing field and in the decline in the influence of older vendors. Systems integrators and consultants are in the first wave of new client/server applications; many report that a growing percentage of their work involves client/server solutions. For example, in 1993, 40% of the work of Andersen Consulting was based on client/server computing. Client/server technology is a development that IS professionals must follow, understand, and develop skills in. This article discusses the structure for client/server computing and presents steps for implementing client/server business solutions.

Structure for Client/Server Computing

Client/server computing is a style of computing that involves multiple processors—one of which is usually a workstation—over which complete business transactions are processed. The typical structure for client/server computing is discussed from three perspectives:

- System software, hardware, and network.
- Data.
- Processing.

System Software, Hardware, and Network-Evolving Client/Server Environments

Exhibit 1 shows the typical hardware and network configuration in a client/server solution. According to the definition, workstations provide the business user with access to the computing capability. The workstation is connected with other workstations on a local area network (LAN). The workstation is also usually connected on the LAN to a work group server that provides data base server capabilities to the workstations. This work group server or another server provides support in communicating to other work groups and to the enterprise machine.
Typical System Software, Hardware, and Network Configuration in a Client/Server Environment

An intrinsic part of the client/server solution is the enterprise computer, which is usually the mainframe of a minicomputer. Typically the connection to the enterprise machine occurs through the use of a Wide Area Network.

There are, of course, many variations on this theme. One is that the workstation will change in form during the next few years. For example, personal digital assistants, such as Apple's Newton, may be used as mobile workstations in the future. Another variable is the intermediate work group server, which is often a widely debated element in client/server computing. (The next sections on data and processing allocation explain the need for an intermediate work group server in client/server business solutions.) The form of the enterprise computer is also likely to change as traditional mainframe computing technology is replaced by more powerful but much less expensive processors that use Reduced Instruction Set Computing. The network may change as well, with client/server being a driving force for mobile computing. Many of the traditional wiring approaches for networks could be replaced by wireless technology, for both local and wide area network (WAN).

Client/server hardware and network computing technology will change in form and behavior as solutions become more pervasive. However, the roles of these processors and networks will remain stable.

Data Perspective

Exhibit 2 shows a typical data allocation in a client/server business solution. If the workstation has a disk, the application usually involves stable, frequently referenced information. For example, in an auto insurance underwriting application, there are rating tables and a list of the automobile types that are covered. Both represent data that would be frequently referenced in the process of underwriting an insurance application. Placing the data on the workstation contributes to the overall processing speed by allowing fast access to the data. It may also contribute to the stability of the application by not requiring access of the data over one or more networks that may be subject to variable and unpredictable loads.

Typical Data Allocation in a Client/Server Environment

On the intermediate processor, multiple types of data reside. This includes what might be termed work-in-process data, or data that is built up in the process of doing a business transaction until the transaction is complete. Nontraditional data, such as image and voice, is also included.

Finally, the enterprise machine contains the data that must be shared across the enterprise. The auto insurance policy is an example of such data. The claim is made at some site other than that at which the policy was written. If the insurance policy resides on the enterprise machine, whoever in the enterprise needs the information has access to it.
Processing Perspective

Exhibit 3 portrays a typical processing allocation in a client/server environment. On the workstation is the processing that controls the flow of the application. To use the insurance example again, if the agent decides to move from descriptions of autos to be covered to specific types of coverage by automobile, the processing logic to do this would reside on the workstation.

Processing Allocation in a Client/Server Environment

The intermediate processor does the processing that directs the flow of work through the office. Thus, if a receptionist in the insurance office takes initial information on customers and determines that they are interested in buying life insurance, the logic to move this data to the appropriate underwriter would reside on the local office server.

The enterprise processor in this description serves as a database server to the enterprise, supplying data where and when it is needed across the enterprise. In addition, the processing of large batch applications such as cycle billing seem to fit best on this processor.

Issues in Implementing Client/Server Computing

It is probably safe to say that in the move to client/server computing, the issues associated with successful implementation are not always thought through carefully. Client/server computing involves distributed data and processing; therefore, client/server applications represent a more difficult problem to implement than traditional computing.

There are six key areas of concern for organizations that are implementing client/server applications, and each covers a broad set of issues. The key areas are:

- Change management.
- Technology architecture decisions.
- Delivery of infrastructure.
- Systems management.
- Network delivery.
- Site implementation.

Change Management

The cost and difficulty of client/server computing can be justified only if the organization is, in some fundamental way, changing the business process. Business process reinvention (also known as business process reengineering) reflects changes in roles, responsibilities, and rules. For any organization attempting the implementation of client/server computing, it is an essential consideration.

To define the reinvented business process, the first thing needed is an understanding of the business, the information technology, and what the technology can do for the business.
The organization must analyze the business process and each of its steps, asking why and how the steps and the overall process contribute to the business, before a reinvented process emerges.

The most difficult part of reinventing business processes is getting the user to accept the reinvented business process. Inherent in Business Process Redesign is that the business users must change their behavior to work with the reinvented system. Therefore, an extensive change management effort must be undertaken.

There are various life cycles for implementing a change management effort. Exhibit 4 gives a conceptual overview of such an effort. The business user moves through several stages over time to acceptance of the reinvented business process. Change management begins well in advance of user training.

The Change Management Curve

User Awareness
The first stage is to build awareness of the reinvented business process. This means helping the business user understand what the system is about, why it must be changed, and at a superficial level, how the reinvented business process will work. This stage is often accomplished by communication through a series of presentations on the process, demonstrations of prototypes, and extensive written and verbal communication on the new business process. These early communication efforts must be led by a business user who has the respect of the business organization. Without this, the typical IS person has a credibility problem in advising business users on the need to do things differently.

User Knowledge
Awareness building is followed by an effort to build knowledge of the proposed reinvented business process. This process is a more intensive effort to ensure that the user understands how the system behaves and how the user will behave with it. A model office is useful at this stage. A model office includes a complete simulation of the proposed work environment. The business user can go through significant parts of the workday and deal with specific parts of the new business process.

Trial and Evaluation
Business users need to test the reinvented business process against their knowledge of the business and what the business needs to be effective. It is imperative that the change management and systems-building teams listen to the business users' concerns about the reinvented business process and understand the problems. The teams must think carefully about whether the issues that users raise are a reflection of the current view of how work is done or are in fact a real shortfall of the reinvented process.

If the problem relates to users' reluctance to change, there must be more effort to explain why the new way is the answer for the future. However, if the systems builders determine that there is a need to change the planned reinvented process, changes must be undertaken. The business users who wanted the change must review and evaluate it to ensure that it addresses their concerns. This is a way for users to see that they are a part of the systems-building process and can have an impact when they have real concerns. This perspective may be invaluable in enhancing the credibility of the systems-building team.

Although strategies vary depending on the size of the user population, the change management effort should attempt to move as many business users as possible through
this process. Furthermore, the business users should have credibility as businesspeople and they must be asked to communicate their experiences and their perspectives of the system to the rest of the business community.

User Training
As this testing and revision process begins to stabilize, in terms of additional changes, the training effort can be built. However, training should not be started too early, because if done before the user has built a knowledge and understanding of the system, it can give users the impression that their input will not be used.

Change Management and Training for IS
A change management process for the IS person is as valid a concern as for the business user. The objective of the change management effort in the case of the IS staff member is to complete the learning curve in the application of client/server computing. This effort tends to be overlooked or treated in an ad hoc fashion. This is unfortunate because client/server technology can succeed only if systems personnel are well informed.

One approach to developing training for IS personnel is dubbed the “rules, schools, and tools” method. The first step to building new skills for IS personnel is to define the overall methodology used for development. This methodology defines how systems are to be developed in a client/server environment to the point of tasks, steps, and deliverables. It defines the roles of designer, programmer, architect, and project manager.

With the rules in place, it is possible to define the training required by role and then go through a process of selection, purchase, or building of required training to execution of the actual training. In addition, as the rules are established, it becomes more logical to define criteria for selection and opportunities for building the tools to be used as part of the systems-building effort for future client/server environments. If the rules are not well defined, the selection process can lead to a set of discrete tools that are difficult to integrate and share across the entire development process.

The amount of training depends on the individual’s background and the rules selected. It is reasonable to expect programmers to need two to four weeks of training, designers one to three weeks, and architects one to four weeks. Project managers may need one to two weeks of training. Consequently, training can run from $1,000 to $4,000 per person. IS management should plan on one to two weeks of skills upgrading when going forward with this technology.

Training Costs.
Another issue is how to account for training costs. Companies use several strategies. One is to ignore the costs and hope in some way they go unnoticed. Although this seems unwise, many sites try to do just that. Inevitably, what happens is that training is not done or is done inadequately, with subsequent loss of effectiveness of the personnel.

A second approach is to have a formal training budget that treats skill upgrades as part of doing business in a technology area. The advantage to this method is that it allows IS to focus on and control the cost of skill upgrades. It allows the organization to implement a program for the IS department, which can ensure a consistent and high-quality training program. Such an effort also creates a sense of the IS department working as a team to meet the shift to a new technology. The problem with this approach is that when a large technology shift such as that to client/server is undertaken, the costs to build skills in the IS department can be so high that they may be questioned and perhaps denied.
A third strategy is to make the cost of learning a part of the business case for the application that is prompting the move to client/server. Because these costs are typically only for the team members committed to the project, they are often smaller and more acceptable as an overall amount. In addition, the costs can be made a part of the business case for the application and thus can have a direct payback. The issues associated with such an approach are that it may not be fair to ask one project to absorb all of the costs associated with training people when that knowledge can be used across subsequent projects, or the training received may be project-specific and may not address the full range of training required for client/server competence. The fact that only team members are trained can also create an “us versus them” situation in which some people seem favored and others do not.

Decisions on how to pay for training depend on site specifics. The key point is that significant training is required for the move to client/server technology. The enterprise committing to this technology must plan on the investment in developing personnel if it is to succeed at the effort.

Technology Architecture Decisions

One of the most common questions companies have when moving to client/server computing centers on what technology to select, and it is also one of the most difficult issues to resolve. Exhibit 5 provides a framework in which to evaluate the technology decisions. This exhibit portrays, at a high level, the concept of a layered technical architecture. Fundamental technology decisions are needed to define a technology architecture and to address the following:

- Platform and networking.
- Operating system software.
- Networking software.
- Data base management systems.
- Graphical user interface.

The Technology Architecture Framework

After these decisions have been made, the next step is a set of decisions on development tools and systems management approaches. These decisions in turn lead to a development architecture and a systems management architecture.

These decisions are difficult to make because in an open systems environment, there are four to five hardware vendors, an equal number of networking vendors, two to three operating system strategies, four to five data base management system (DBMS) vendors, and two to three Graphical User Interface vendors from which to choose. The number of possible combinations of solutions in this case could exceed 1,000. At a minimum, IS should focus on those components that affect the enterprise's ability to interoperate, or share data across departments. If possible, IS should define, at the enterprise level, those components that allow departments to share information.
For example, mixed hardware such as Reduced Instruction Set Computing processors and CISC processors can present ongoing problems in sharing data because of basic dissimilarities in bit patterns. Different networks in different departments present ongoing problems when those departments want to share data. Different data base management systems present basic problems when there is a desire to access one department’s information from another department. In each case, a means can be found to circumvent these problems, and systems integrators are widely involved in solving them. However, if IS sets basic guidelines on what constitutes a consistent technical architecture, it does not need to find workarounds.

Most enterprises end up with incompatible technical architectures. Many factors contribute to this situation, including legacy decisions. IS personnel contribute to the problem when they take too long to come to a decision on technical architecture and, as a result, the end-user community goes ahead without IS involvement. Therefore, the major reason to focus on interoperability as a criterion for technology decisions is to define a minimal subset of all the decisions so that they can be made more quickly.

Delivery of Infrastructure

When making decisions on the technical architecture, IS must begin thinking in terms of what reusable components the enterprise must build to make the technical architecture usable by developers and business users. These enterprise-built reusable components are what is meant by infrastructure.

In a traditional IS shop, guidelines for designers and programmers are often referred to as the programming standards. An equivalent is needed for client/server computing. Indeed, the equivalent may be even more critical because often the designers and programmers do not have the experience in client/server to fill in the gaps.

Some of these standards are addressed by the methodology decision, but many are not. For example, standards for the input/output design of applications using a Graphical User Interface may be needed. Programming and design standards for workstation environments may be required. Time and effort can be saved if a consistent layered architecture, such as shown in Exhibit 6, is used. Such an architecture must be established, defined, and explained to developers as a consistent framework for all developers to use. As this infrastructure is defined, there follows a need to select or build tools to provide the components and to develop designer and programmer guidelines and training in how to use the infrastructure.

Infrastructure Architecture for Client/Server Environment

Systems Management

Systems management addresses the ongoing operation of the client/server application when in production. Systems management should start with and be driven by the definition of Service Level Agreement. These agreements are, effectively, commitments to meet certain levels of overall system performance, reliability, and recovery from problems. The starting point is framing these service-level agreements. With the agreements defined, those parts of the systems management environment that specifically contribute to meeting and confirming conformance to service-level agreements receive the first attention in delivering systems management.
The typical activities that are associated with systems management in a client/server environment include, but are not limited to, the following:

- Configuration management, which involves managing components found in the application and the status and version of these components.

- Activation and execution of components of the client/server application. This activity can include bringing up online and batch components and the activation and execution of service components such as the data base management system.

- Determination of fault status, assessment of reasons for failure, and the initiation of recovery from failure.

- Help desk facilities to answer inquiries from the user and system communities on problems, questions, and steps to recover from problems.

- Determination of performance reliability and, potentially, assessment relative to service management contracts.

- Vendor contract management and associated contacts and payments.

Although this list is not complete, it is representative of the range of issues to be addressed when starting with a service-level orientation. Among client/server solutions, there are currently few, and mostly limited, prepackaged software solutions available to the systems builder. In comparison, traditional computing offers rather extensive solutions often as a part of the system software.

The cost of preparing solutions for systems management in a client/server environment is high. Experience suggests that 10% to 40% of the development budget for initial client/server solutions should be devoted to the delivery of the system management facilities.

Help Desks

On the low end of this range, a strong help desk facility can be built. In traditional environments, the help desk is used to collect and route problems with some sort of minimal problem analysis/resolution role. In a client/server environment, a relatively complete analysis of the problem may be required as the problem occurs. Immediate analysis is needed for at least three reasons. First, the problem is frequently due to events on the client computer. There are few if any tools to capture status and playback of events on this machine. Once the client has rebooted, all traces of actions and events are often lost. Thus, the detailed analysis must be done while the situation exists.

Second, client/server applications have a rich set of interactions and it may be essential to find out at the time the problem occurs what steps the business user went through while they are fresh in the user's mind. Finally, a client/server application is probably deeply embedded in the business flow; to get the user working again, it may be essential that the help desk go through a complete assessment and recovery while the user is on the phone seeking assistance. The help desk staff should be required to have a thorough understanding of the application and how it is applied in the business process.

In this proactive role, the help desk should be viewed as a problem manager having the authority to contact IS and operations personnel and the ability to direct their work to correcting the problem. This approach raises many questions about who has authority for
the direction of personnel. The issues must be resolved in favor of what best supports the business user in need of assistance.

High-end solutions include sophisticated software for the detection and isolation of problems. Although only limited software solutions are available at this time, there is reason to believe that much more complete solutions will become available from vendors during the next few years. Thus, to avoid investing in systems software solutions that are overtaken by vendor products, IS should focus on the lower end of the scale and, specifically, the delivery of a highly effective help desk facility and tools to upgrade the applications and overall configuration.

Network Delivery

In the client/server architecture in Exhibit 1, the question is: Where is the computer in this architecture? The answer is that it is all the computers. The network is the platform. Client/server computing is as much a communications solution as it is a computing solution, so there must be as much focus on the network side as on the computing solution.

The difficulty is twofold. First, the typical information processing person is not strong in networking knowledge and design. For many years, the proprietary networks associated with traditional technology were largely the domain of the network specialist; the IS person simply provided the network specialist with requirements while the specialist did what was required to meet the needs.

At the same time, the network specialist is confronted with a new set of terms and concepts for modern networking technology. In addition, the network has such a profound impact on the capabilities of the client/server solution that network specialists must be brought into the design process much earlier. Often they are the ones to say yes or no to a functional requirement on the basis of the networking capabilities.

The actual delivery of the network may require greater lead time to assemble and integrate the required components as compared with earlier proprietary solutions, in which the network components usually arrived from the vendor with assurance that the vendor would integrate them after assembly. In a client/server environment, usually components from different vendors are being integrated to support the network and resolution of problems falls to the integrator. There may also be a need to upgrade the overall network, which can greatly lengthen delivery time frames. Even as the components are assembled and tested, it is common to encounter transient problems that take extended time to resolve. When the network goes into production, there still may be a need for ongoing network problem solving to make the network stable.

Site Preparation and Installation

Site-related work refers to:

- The process of reviewing sites, assessing what is required to prepare them for installation of the client/server solution.

- The process of readying and installing the sites.

- The process of testing an installed site to ensure that it works as expected.
Ongoing maintenance of an inventory of the site and its status for running the application.

These efforts may not be directly related to client/server computing, yet most sites involved in the first-time application of client/server technology to business problems encounter significant and difficult problems in this stage. If hundreds of sites require significant upgrades, site installation and upgrade may be the single largest line item in the development budget and may require the longest lead time.

If the enterprise has already installed client/server applications, site-related work may be less of an issue. But for first-time applications, it is necessary to assess the site's suitability to run a client/server application. This assessment should address all aspects of site preparedness, including power, air conditioning, and possibly the installation of physical wiring. Some potential problems may seem obscure. For example, at one manufacturing site, a stamping process created a heavy vibration that required additional protection for disk drives. Another site was next to a radio station that posed a problem with electromagnetic fields.

These problems must be recognized and resolved before the purchase and installation of hardware and software. In addition, sites should be tracked because in the process of building the system, the initial survey/assessment may become outdated.

Arrangements must be made to have carpenters, electricians, and air conditioning specialists come to each site to make any necessary changes. Often, complex negotiations and ongoing contract management are needed to find parties qualified to do this work. A building contractor may be retained to see the work that is to be done. When many sites are involved, a management challenge arises for which the organization might have little competence. It is worthwhile to consider outsourcing the site upgrades to companies that specialize in this effort.

Organizing for Successful Client/Server Engagements

*Exhibit 7* is a planning chart in which each box suggests a work segment to be done to address the previously discussed issues in implementing a client/server solution. The arrows and lines suggest a relative sequence and interdependency of the efforts. The completion of a segment usually means that a major work deliverable has been completed.

The diagram is complex because there are a complex set of interrelated problems that must be dealt with as a whole. At the same time, the exhibit is still at a conceptual level, because it does not show all the interrelationships or the actual work tasks to be done.

To provide a frame of reference, the work segments labeled Design Applications and Implement Applications represent the basic process of designing and building the application systems. Full life cycle methodologies for this effort in a traditional environment may span 700 to 1,000 pages—and that is just one component of the total systems effort.
Making Change Happen When It Is Needed

The boxes across the top of Exhibit 7 pertain to organization roles and user training; these work segments address the change management issues of getting the business user to work with the new client/server system. The first major segment of work is to define the new business organization. The next work segment is to design the specific work roles in terms of daily tasks and deliverables. Finally, training is defined that ensures that the business user can take on and fulfill the new roles.

Change management includes ensuring user awareness, knowledge, and trials and evaluation of the reinvented work process, and it is one of the most challenging and creative jobs to be found in the process of building systems today. The real difficulty is defining where the responsibilities for change management end and where the role of the systems designers begins. There can be an overlap between what the change management personnel and the systems-building personnel define. In essence, both groups are making commitments on what the system will do. It is important that these work efforts be coordinated. The key to avoiding the potential overlap is to have the change management and systems people sit down and resolve what each group will deliver.

Technical Architecture Design

The Design Architecture work segment refers to the many decisions on hardware, systems software, and networking. The decisions made in the work segment Implement Architecture form the basis for building the infrastructure of tools, standards, and methodology that the systems builders need.

Included in the technical architecture decisions are:

- Going-in positions on hardware platforms to be allocated by site.
- Associated decisions of operating systems, Graphical User Interface, and network strategy.
- Decisions related to the DBMS and development tools and languages.
- Positions on the intent to provide support for the developer and user. For example, will the design environment be highly integrated with the implementation environment? Will a repository strategy be used? What facilities and capabilities are to be provided with regard to testing? What capabilities are to be provided with regard to training and help facilities?

Without these initial decisions, progress on all other application and business issues is difficult. At the same time, once these decisions have been made, they must be reevaluated as experience and time passes.

Getting the Systems Management In Place

The Design Operations Architecture and Implement Operations Architecture work segments (see Exhibit 7) address the steps needed to get the systems management approach in place.

Design Operations Architecture should begin after the technical architecture decisions have been made and as the functional capabilities and requirements of the application are
beginning to be defined. Information on functional capabilities is needed to help determine service-level agreements. The service-level agreement is key to defining the overall systems management architecture.

When the decisions on the overall systems management architecture have been made, implementation work begins. This is often a matter of purchasing individual tools and integrating them. Included in this effort is the definition of help desk features and the beginning of training and actual implementation of the help desk.

**Client/Server Is Communications**

The tasks of network design and implementation also must be performed in order to deliver a client/server business solution. The Network Design and Network Implementation work segments should begin once the functional capabilities of the application are known.

Client/server computing demands a much tighter integration of network and systems design and building compared with traditional development. Careful attention should be paid to the evolving network design and how extensive an upgrade is needed from current capabilities.

**Having Someplace to Put the Client/Server Application**

The tasks at the bottom of Exhibit 7—site surveys, preparation, and implementation—include steps for preparing the physical premises for a review of the client/server application. Specifics vary, but plans should be made to survey every site to assess its suitability to run a client/server application. Change management efforts and the site upgrade work must also be coordinated.

**Make Sure It Works**

A final work segment worth noting is Release Testing. This work is needed in client/server environments to ensure that all the components previously described come together as a system and work. By definition, client/server systems run in the business environment, perhaps thousands of miles away from the systems developers. Furthermore, if an organization bases a new business process on the system, the business could then become critically dependent on the system. Releasing a less than completely tested system would be damaging.

Tests are needed to ensure that the application works as a whole, that the infrastructure and technical architecture to be released is the one that applications have been running on, and that the installation procedures are usable by someone at the local site. Any emergency procedures or error recovery procedures should be tested to determine whether they are understandable and workable by people at the local site. End-to-end performance must be addressed at this time to ensure that the service-level agreements can be met. In addition, operations and help desk procedures should be tested to ensure that they do what is expected. Release testing can make a significant contribution to the ongoing reliability and usability of the system; at the same time, it is not a trivial effort. IS can expect to plan a two-to-three-month test cycle in release testing.
Just In Time Adds Tension

A key concept in the framework is the just-in-time delivery of components of the parallel work segments. Ideally, people like to do work in a sequential fashion; in today’s business environment this is not practical. Furthermore, it is more efficient to have results shared between the work segments as the work proceeds. As a result, all the work segments shown in Exhibit 7 should be performed in parallel.

This just-in-time approach, in which several teams must be kept tied together and working as a whole, is a major challenge. To manage the overall effort, a program management strategy must evolve that ensures that individual efforts are proceeding according to an overall plan and that the deliveries from each team meet expectations in terms of quality and timeliness.

The Value of Moving to the Right Answer

There is a return arrow at the top of Exhibit 7 that signifies that the adoption of client/server computing requires an iterative approach. It is essential in this environment to get the application delivered to the user. Inherent in adopting a new business process is the unanticipated impact it will have on the business user. Also inherent in the use of client/server computing is the rapid evolution of technology. Iterations provide the opportunity to rethink architectures to determine whether previous decisions and approaches have aged and are no longer valid.

Iterations must be done quickly. According to one view, iterations should be done in six months or less; however, it is difficult to get all of these parallel efforts organized and drawn to a conclusion in six months. A more practical approach is a 12- to 18-month cycle for each iteration, considering the number of parallel work efforts involved.

Recommended Course of Action

For IS, the course of action depends on where the organization is in the application of client/server technology. For the site just beginning the application of client/server technology, the following suggestions are made; IS managers should:

- Choose applications that would benefit from the distinct technology features of client/server computing, such as its ability to put processing where it is needed and the communications capability that is an inherent part of client/server applications. Publicize that the application will be a first delivery of processing capability in a client/server environment. Ensure real and lasting commitment from the business community for the application.

- Build a team with skills in client/server technology. This can be done through the use of training for in-house personnel and by retaining outside expertise. Ensure that users with credibility in the business world are committed to the effort.

- Organize an effort according to the planning chart shown in Exhibit 7. Time and resources are needed to ensure the following:

  - The change management team defines the reinvented business process and begins the work activities shown in Exhibit 4.
- The technical team defines a technical architecture addressing the execution components of Exhibit 6.

- The design of the application to support the reinvented business process is begun and coordinated with the change management effort.

- As business worker requirements for performance and reliability under the reinvented business process begin to appear, a systems management effort should be initiated to build the systems management capability that can determine whether these needs are being met over time.

- A network design effort is started that brings online the network required to support the effort.

- As required, the site evaluation, installation, and delivery effort should be planned so the sites are ready to run the client/server application as it is implemented.

- A program management effort can be instituted to ensure that all these strands of work deliver the components needed on a timely basis and at a sufficient level of quality to make for an acceptable application.

Client/server computing appears to provide the next level of technology that organizations need to solve the problems of reinventing business processes. At the same time, it is a more complex and inherently more difficult technology than traditional computing to successfully implement. It is safe to say that business problems demand that IS address and contain these risks to advance the value of business computing.

**Author Biographies**

Hugh W. Ryan

Hugh W. Ryan is a partner in Andersen Consulting's TSI-Worldwide Group in Chicago, where his work focuses on the identification and application of leading-edge information technology to business opportunities. Previously, he headed Andersen's new age systems group of cooperative processing and the enterprise information architecture group.