

Client/Server Software: What Is It All About?

Written by Emily P. McAnally, Bonner & Moore Associates, Inc.
Tuesday, 01 July 1997 15:09

Modern program architecture benefits provide flexibility, reliability, and responsiveness for larger applications.

Client/server methodology and architecture offer a versatile approach to the development and deployment of software for major computer applications critical to strategic planning and enterprise operations. Maintenance information systems and computerized maintenance management systems (CMMS) for larger plants fall into this critical software category.

The client/server approach benefits both the software developer and the user. It is especially important for larger applications if the system is to provide the reliability and responsiveness associated with classic mainframe systems.

As defined by Microsoft Corp., client/server architecture generally applies to a software architecture in which functions are split into independent tiers or collections of "services" and "requesters" on a single computer or several computers. But the potential of client/server technology is broader. It encompasses more than a personal computer accessing a server on a network. Its greatest potential is in creating and managing distributed tasks, business logic, and shared, reusable software components so the end product is efficient, reliable, and less costly.

The key to maximizing this strategy is the application of a three-tiered architectural approach to client/server solutions that is generally considered to be the leading model for application development. This approach separates the various components of a client/server system into three logical tiers of services that come together to create an application:

1. User services, which provides for presentation of information, functionality, and navigation
2. Business services, which provides for shared business policies and rules, and generation of business information from accumulated data
3. Data services, which provides for definition of data, and storage and retrieval of persistent data.

User services

The user services tier is responsible for supporting users as they interact with the application. It presents information to and gathers data from the users. This tier's key characteristics include

Client/Server Software: What Is It All About?

Written by Emily P. McAnally, Bonner & Moore Associates, Inc.
Tuesday, 01 July 1997 15:09

- Giving users control as opposed to the software package taking control
- Supporting the activities users perform while using the package
- Providing the visual interface for presenting information and gathering data
- Linking together a collection of business services that deliver business capabilities.

Working functions of this tier include

- Relate with other user services and the business services tier
- Use business services to obtain information and make requests
- Format content, present it to the user, and interpret users' actions.

The user services tier presents information to users and generates a form that is most appropriate to the current user. User services is a secondary data source for the business services. Most of the business data received will come from the data services. Users are not aware of the data services consumed by the business services tier.

Business services

The business services tier links related tasks with the relevant business rules. Business tasks are defined as the steps needed to accomplish a business decision. The key characteristics of the business services tier include

- Linking related business tasks with the relevant business rules
- Responding to requests from the user services to execute a related business task
- Converting data received from business and user services into information
- Securing the data services needed to accomplish the business task with the relevant business rules.

Working functions of this tier include

- Support the user services by defining available business tasks, information needed by the business task, and information that is returned by the task
 - Process information sent by a user-initiated action
 - Interact with the data services by defining the rules needed to manipulate, structure, and use the data.

Client/Server Software: What Is It All About?

Written by Emily P. McAnally, Bonner & Moore Associates, Inc.
Tuesday, 01 July 1997 15:09

The business services tier is the only service in an application that has a relationship with both the user and data services tiers. The relationship of business services with data services is quite a bit different. Business services defines the rules or some of the rules needed to manipulate data. The business services tier enforces the rules the package uses to manipulate data.

Data services

The data services tier is responsible for maintaining the availability and integrity of data as a corporate asset. This tier supports the lowest level used in manipulating data within an application.

Key characteristics of the data services tier include

- Defining, maintaining, accessing, and updating all data in the database
- Managing and satisfying business services requests.

Working functions of this tier include

- Has no relationship with user services
- Uses the business services to provide information and determine if data are valid.

Data services is primarily responsible for accessing the raw data for the business services tier. Business services then takes the data and applies the business rules to generate the information a user has requested. User services takes the information and delivers it in a form that is meaningful to the user.

Assembling a solution

Once properly designed, components are then assembled and applied to the appropriate logical

Client/Server Software: What Is It All About?

Written by Emily P. McAnally, Bonner & Moore Associates, Inc.
Tuesday, 01 July 1997 15:09

tier. Many factors should be considered in making component distribution decisions, including

- Locality of reference--putting a component where it is best used, minimizing overall network traffic in the physical environment
- Data distribution--taking network load into account
- User interfaces--providing standard user interaction
- Correctness--serving competing applications on the network, including such issues as the possibility of deadlock, fair allocation of resources, and security
- Reliability--avoiding excessive demands on the network that may make it unstable
- Network reliability--serving competing applications.

Once the software components are applied to the appropriate logical tier, it is easier to distribute them in the physical environment. Ideally, physical location decisions are driven by business and operational needs. In reality, there are technological limits, many of which come from the software package itself.

For example, if a CMMS is designed around a "fat client" or two-tier architecture, performance is limited and the CMMS cannot support large numbers of concurrent users. A fat-client architecture places most services on the desktop personal computer and the data on a server. Smaller organizations with a few users will not find this method to be a problem, but larger, multiplant, multidepartmental organizations accustomed to mainframe-level performance will be frustrated by increased response times, slow performance, and increased network traffic.

Distributed CMMS applications must take network usage into account. An application that places excessive demand on the network may be unstable. Thus, these applications must be designed to use the network realistically.

For multiuser systems, the total concurrent user population is only one factor in the measurement of response time. Components should be deployed to minimize the number of network "calls" and the amount of data transferred across the network. A developer can do this by locating components as close as possible to the resources they will consume most heavily. For example, user services (tier one) should generally be located on the client workstation. Business services that interact heavily with other components should be physically located on an application server (tier two). Business components that access data stores heavily are best located on a database server (tier three).

Client/Server Software: What Is It All About?

Written by Emily P. McAnally, Bonner & Moore Associates, Inc.
Tuesday, 01 July 1997 15:09

Client/server benefits

Microsoft's software development experience as well as the experience of Bonner & Moore has shown that a tiered, component-based solution approach to client/server development has many advantages over traditional application architectures.

From a development team organizational standpoint, the component-based approach splits teams into two groups: one group develops core components (custom controls, stored procedures, object linking and embedding--OLE--servers, and so on) that can be useful to many applications. The second group builds business solutions by integrating the services provided by these components. This approach has many advantages over traditional application development.

- Developers can specialize in what they are best suited for, which lowers training costs. For example, database-oriented programmers write stored procedures while visual programmers design user interfaces. This method translates into lower software licensing costs to the user.
- Reusable components can be written in a variety of programming languages such as Visual BASIC, Microsoft Visual C++, or SQL, providing a great deal of language independence and the opportunity to develop in the language that best suits the specific task. This advantage translates into more efficient software code and a more responsive system.
- Components can be deployed on the network to maximize efficiency, performance, security, and ease of maintenance. Users get better response times and better use of current computer technology. The package also can grow with the user community.

These and other advantages make the three-tiered component solution approach a sound method of application development for mission critical systems. It also provides reliable, mainframe-level performance in a distributed client/server environment. **MT**

Emily P. McAnally is manager, software development and support, Bonner & Moore Associates, Inc., 2727 Allen Pkwy., Houston, TX 77019; (713) 522-6800.