EIS InfoBook

Multi-Architecture Solutions

EIS International Corporation employs a multi-architecture approach to information management. This approach has evolved over several years as we recognized that no single anchitecture is entirely appropriate for all of the clients who use our family of InfoBook software. Our clients include those who use single computers on emergency response vehicles making life or death decisions, so we need an architecture that can produce an extraordinarily rapid response time. Yet, we have clients who require a large number of users with multiple access to a single shared database from widely distributed remote sites, necessitating a client/server architecture for effective data transfer and sharing. We also require applications that have real-time sensor input for chemical detection sensors or weather sensors used in complex models run by a very small number of people, but with the results shared with many. All of this combines to demand what we call multi-architecture solutions.

Multi-architecture consists of six different types of computing.

- Standalone and Local Area Network (LAN)
- Distributed
- Two-Tier Client/Server
- Data Warehouse
- Multi-Tier Client/Server
- Internet/Intranet Multi-Architecture

Each of these is described in the sections below.

Standalone and LAN Architecture

At one end of the spectrum of the EIS multi-architecture is a standalone or LAN (Local Area Network) system in which each computer contains (a) the GUI that presents the data to the user, (b) the application that runs the information processes requested by the user, and (c) the database that stores the data that will be analyzed for the user. This

particular architecture requires that the standalone or LAN computers be completely self-contained. All the information required for analysis is located on an accessible hard disk and all executable programs to carry out analyses are located there as well.

For many user requirements, this is a sufficient and excellent configuration. However, in those applications that require the ability to communicate and distribute information beyond a single computer or LAN, this architecture is inadequate. Therefore, to the standalone and LAN architecture, EIS International added the ability to carry out very robust communication with a product called ECOMM. The use of the ECOMM software is the beginning of the second type of architecture, "distributed."

Distributed Architecture

Distributed architecture as depicted in Figure 1 (below), involves the use of a communications server to manage the exchange of data among various computers distant from each other. The communications server is the EIS ECOMM product installed as a workstation on a LAN. As shown in the illustration, any computer on the LAN site can send and receive ECOMM messages which contain not only the narrative message similar to an E-mail but also the attached data records from the EIS databases for exchange among sites. In the illustration, the lightning bolt-like lines indicate a remote communications capability. This remote communications can consist of regular telephone lines, cellular telephone, packet radio, satellite, and Internet E-mail. Remote sites can be either single computers or local area networks.

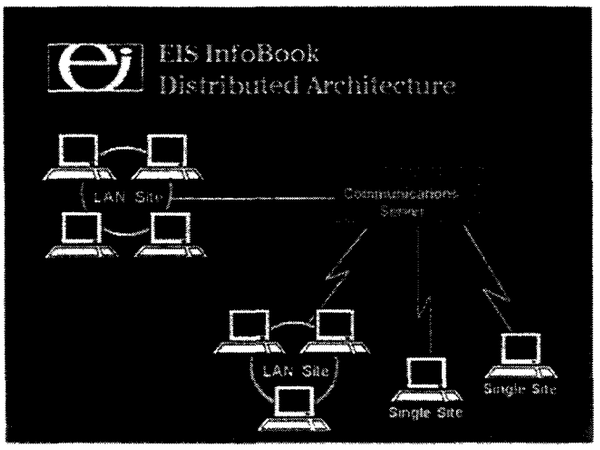

Figure 1

The distributed architecture allows a single computer, a LAN, and any other combination of single and LAN sites to communicate with each other in both an E-mail format as well as to exchange data among databases. EIS software has the ability to do unattended, timed queries and extracts which allows a remote site to identify changed data that is

automatically extracted and made available for an upload to a communications server to update another site's resident database. This effectively forms a replication capability within this distributed network. The distributed network's principle reason for being is to support low bandwidth communication of specific data inserts and updates among a series of sites. This provides a reasonably robust form of data exchange at specific intervals of modest frequency. It is satisfactory for a system that lacks the communications connectivity necessary for the next type of architecture, client/server.

Two-Tier Client/Server Architecture

We define client/server computing as involving two or more computers that share tasks as part of a complete application. The design of the system optimally allows each computer to perform tasks that are appropriate to its capabilities and specific function. This suggests that computer resources and data storage resources are positioned where they will do the most good in fulfilling the application being performed.

Our two-tier client/server architecture, shown in Figure 2, is the system of choice when (a) everyone on the network needs to be sharing the same data and (b) the communications channels within or among sites have sufficient bandwidth to carry out an intense and continuous exchange of data. The downside of this architecture is that when communication lines are down, remote sites have no computing capability. In many instances, this two-tier client/server architecture forms the most efficient way to exchange information and has a much lower system-wide maintenance. In the two-tier client/server, the database server contains all the data and many of the data retrieval rules. At the client workstation, the presentation GUI and the application reside in what is called a "thick" client. The GUI and application program must be maintained on each client with the application making requests to the server for data to be delivered to the client for display. Each

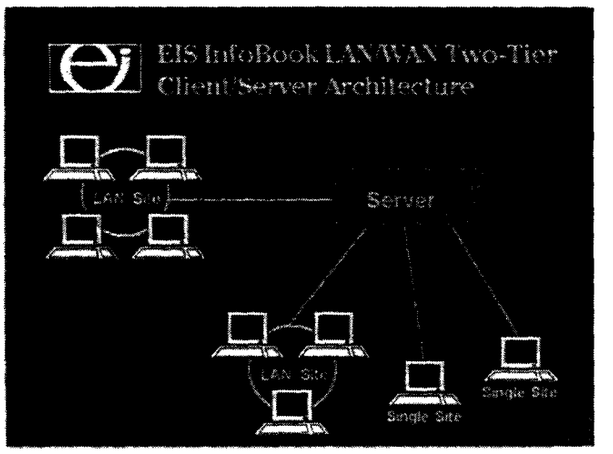

Figure 2

client has direct access to the database and, based on each individual user's permissions, can directly update data on the server. This can lead to two drawbacks: a large number of people accessing a single server can degrade performance and everyone is connected to, processing, and updating the ultimate source data which can result in a risk to the integrity of the data. To counter these two drawbacks, the next type of architecture is available, data warehouse.

Data Warehouse Architecture

A data warehouse is a server, like all other servers in a client/server environment. Its purpose, however, is to be a repository for data that is viewed, not acted upon. As Figure 3 shows, the data warehouse server is directly connected to one or more other servers. This may be in the form of updatable views or standard database replication. The net result is that a client can access either a server that contains a subset of information or the data warehouse server which contains a superset of information. The difference is that, in general, a site will have permission to make maximum use (read, update, etc.) to its own server whereas its access to a data warehouse server does not allow the update of information. The data warehouse represents an integrated view of clean and analyzed data rather than the raw data that would reside in the other servers.

EIS InfoBook's use of data warehouse architecture is ideally suited for those central or headquarters offices that need to understand, but not act upon, information processes taking place at remote sites. This concept of different users with different needs for information processing being served by different information processing capabilities and architectures gives rise to the next type of architecture, multi-tier client/server architecture.

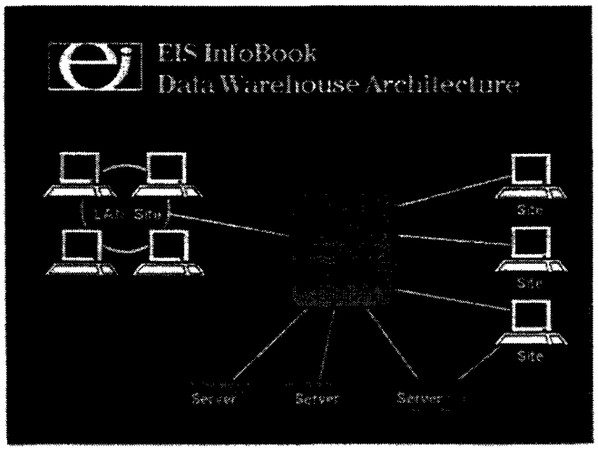

Figure 3

Multi-Tier Client/Server Architecture

Multi-tier client/server architecture consists of both three-tier and two-tier client/server operations. In certain cases, it even includes applications that run solely at the client level, in a sense a no-tier architecture. The motivation behind the multi-tier client/server is to put the most appropriate computing resources in the hands of those who need them. This requires careful assessment of computing requirements and the ability for an information technology department to selectively apply its technologies to solve specific problems.

The left side of Figure 4 shows a traditional three-tier architecture. Here a Web browser client (a very "thin" client) contains the presentation layer which is used to display information to the end user. This presentation layer is connected to an application server over a TCP/IP network (a private Intranet or the public Internet). The application server contains all of the business rules for processing data that is stored on the database server which is connected by a high-speed LAN. Thus, the presentation tier, application tier, and data tier all combine to single-purposely perform the task for which they are most capable. The "thin" client provides a graphical user interface for entry and display of information, the application server runs the rules and processing, and the database server simply responds to queries and returns information.

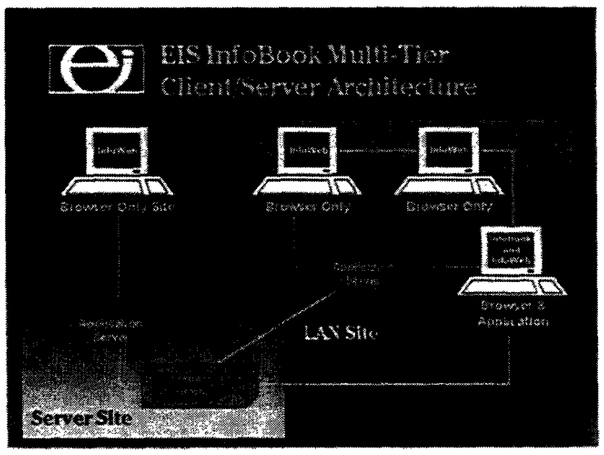

Figure 4

In the center of Figure 4 is another method for utilizing a three-tier architecture. Here a LAN exists remotely. The browser can be used locally on the LAN as the presentation layer at each client. In a case where the application layer is very dominant in the information processing, then the application layer is located on the remote LAN instead of at the site with the data server (as in the above example). This means that the priority relationship is between the presentation client and the application server with

the highest speed connectivity occurring over a local area network, thus improving performance. The public or private TCP/IP connection between the application server and the remote database server provides for the query and retrieval of data. Placing the application server on a LAN (meaning faster data exchange) means that the application can achieve a high level of processing (computer models, for example, of chemical reactions or weather conditions) or contain a subset of cashed data (for example geographic information system data) which it is not desirable to transfer over the network communications lines.

In the first case (the left hand part of the illustration) the presentation layer is very thin, located at a client site with the public or private TCP/IP network connection going from the client to the remote application server which is co-located (on a local area network) with the data server. In the second case we have just discussed, the presentation client and the application server are located on the same local area network, thus allowing for rapid exchange of large volumes of information, and are linked to the remote data server by the TCP/IP network. (A third model can exist in which a client is remote from the application server which is remote from the database server as well.) All of these implementations can be in use at the same time.

The third part of the multi-tier client/server architecture is illustrated on the far right of the illustration. In this case both a browser and application software can exist on a client workstation on a local area network. This will result in two type of database access taking place, both a three-tier and a two-tier client server. The three-tier uses the browser through the local (or remote) application server to the database server. At the same time, and in a different window, an application can be running with direct access to the data server (thus, a two-tier client/server).

Under what conditions might this be desirable?

Some applications are not suited to a three-tier approach. For example, a computer model that requires significant sensor input. Such a model would have sensor input coming in by RS232 ports into the computer that contains both the browser and the application. The application might very well be processing that sensor data. A second application might be a computer model used only by a handful of engineers that periodically employ the data from the sensors to create model output. If the model

output is desirable for viewing by a large number of people, then it can be stored directly in the data server. It thus becomes available to any browser to look at through the application server. The level of effort and return on investment often is unwarranted to convert the sensor processing program and the analytical model to a three-tier architecture when, and this is the most important point, a very small number of people actually make use of this highly technical application.

Three-tier client/server is most effective in "disconnected" applications. This means that the client user spends a significant amount of time working with or looking at the data received rather than constantly interacting with a succession of newly received data. The requirement to have "real-time" visibility, access, and update for all the data in one's operation at all times really reduces the value of investing in three-tier client/server. The rule to recall is that networks are always slower than computers. Adding communication layers (which one does in three-tier client/server) rarely improves "speed."

Internet/Intranet Multi-Architecture

The composite of all of the different architectures that EIS International supports is shown on Figure 5. In this multi-architecture system, individual computers using low bandwidth communications share information through communication servers with local area networks that have LAN databases and/or traditional two-tier client/server databases. The users may employ, for analytical purposes, a data warehouse. Across the top of the illustration are sites with very "thin" clients using only a browser to access an application server which then proceeds to access any of the other servers. At the far top right is the combined browser and application site (our combined three-tier and two-tier client/server site) which will provide the robust as well as data-sensitive maximization of computing power and network capabilities.

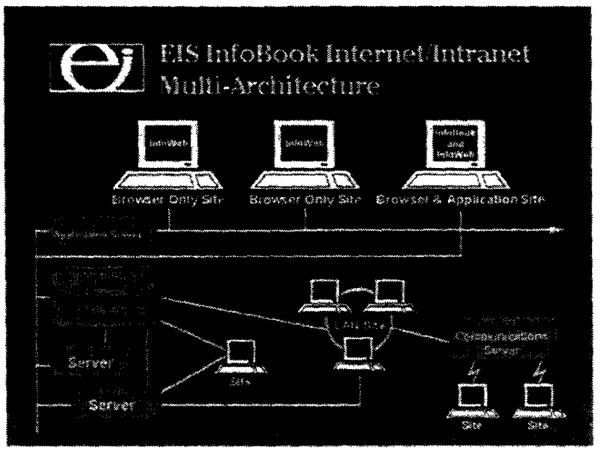

Figure 5

Conclusion

The creation of a computer system in an era of huge technological opportunities is one that EIS International accepts as an exciting challenge. The challenge is to apply the tools to solve exact problems rather than to apply the tools because they are there. The results of the analytical process that EIS International undertakes to match tools to problems results in optimal solutions. We are constantly reminding ourselves that technology is only important when it solves problems. It does not matter whether we settle on two-tier, three-tier, or no-tier. What matters is providing a satisfactory solution. We have been brought to this clear realization by our clients, especially those who demand the latest technology and the best solutions. Marrying technologies, selecting the best mix, and delivering cost-effective performance is our strategic goal. That is what we mean by multi-architecture: blending technologies, current and emerging, into essential information solutions.

For further information, please contact:

EIS International Corporation
Suite 500, 1401 Rockville Pike, Rockville, Maryland USA
Telephone: 301-738-6900
Fax: 301-738-1026
WWW.eisintl.com