Adopting DCE Technology for Developing Client/Server Applications

HP's information technology community has adopted DCE as the infrastructure for developing client/server information technology applications. The team developing the DCE service has discovered that putting an infrastructure like DCE in place in a legacy environment is more than just technology and architecture.

by Paul Lloyd and Samuel D. Horowitz

Many companies are navigating the path to open systems. Vendors, including Hewlett-Packard, claim that companies can receive significant benefits by adopting open system client/server approaches for implementing information technology solutions. While the benefits may be attractive, the array of architecture and technology choices is bewildering.

Hewlett-Packard's information technology group has adopted the Open Software Foundation's Distributed Computing Environment (OSF DCE) as a recommended technology for the implementation of client/server applications within HP. The adoption of a technology, or even an architecture, is not sufficient to ensure that the benefits of the client/server model are realized. In fact, once the architecture and technology are chosen, the real journey is just beginning. This paper discusses the issues that led HP to shift toward open systems for information technology client/server applications, the rationale for choosing DCE as a key technology, and the elements of a new infrastructure built to provide the necessary services required to realize the benefits of open systems.

HP's Legacy Environment

Until very recently, HP's legacy environment included multiple mainframes and over 1000 HP 3000 computers operating in more than 75 data centers located around the world.

Business transactions were processed at places called sites, which were major HP installations including manufacturing, sales, and administrative centers. Each site had a local HP 3000 computer system. Most applications were written in COBOL and made extensive use of HP's TurboImage database management system and VPlus/3000 routines for terminal screen management. These tools were used because they made the most effective use of the HP 3000 computer system. At periodic intervals, batch jobs on the HP 3000 systems would create transaction files for transmission to the mainframes. Other batch jobs processed files received from the mainframe. A proprietary store-and-forward system provided the link between the interactive HP 3000s and the batch-oriented mainframes. Fig. 1 illustrates this legacy architecture.

This architecture gave each site access to its own data, but only its own data. Once a transaction was generated and sent to the mainframe, interaction with other production systems meant that response was indeterminate. For example, system users would have to check repeatedly to determine if a purchase order that had been entered was accepted by the factory and scheduled for production. This acknowledgment could take from hours to days depending on the complexity of the order and the number of HP divisions supplying content. In addition, processing and data communication delays anywhere in the company could impact the response time for the transaction, but the user had no way of finding the bottleneck. Further problems

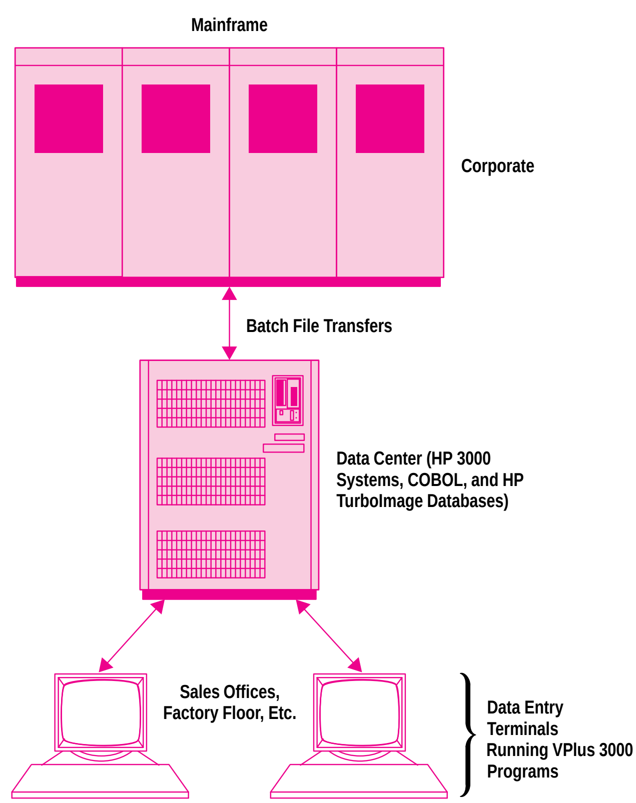

Fig. 1 HP's legacy environment for information technology.

were found during massive data center consolidations that have taken place over the past few years.

The architecture for HP legacy applications yielded a large number of applications, each of which required large maintenance and support staffs. Many applications were customized to address the peculiarities of various sites. This contributed to the support problem. As processing power and network bandwidth grew, the customized versions of standard applications made consolidation difficult. In some cases, support costs actually grew.

New applications were both expensive and took a long time to develop, integrate, and deploy. They made little use of previously written code, nor did they share data or other resources effectively. This resulted in a great deal of replicated data within the company. As the environment continued to evolve and new applications came online to address business needs, users found themselves having to manage a multitude of passwords for a large number of systems. From a security standpoint, each password was transmitted multiple times daily over the network, and host-based login services provided the foundation for all data security.

Movement toward Change

Several years ago, HP realized the benefits that could be achieved through open systems client/server architectures. The single biggest driver for the change was a desire to reduce business implementation time, which is the time from when a business need is identified to the time when a production system is in place to address the need. Other drivers included the need for greater cost-effectiveness of the information technology (IT) organization, and the need to reduce operational and administrative costs. An information technology steering group determined that the widespread use of a client/server architecture would enable a reduction in business implementation time and provide increased organizational effectiveness and reduced costs.

A group of IT leaders representing multiple HP organizations formed a task force to develop a client/server architecture for use within HP. Some of the factors the task force considered when choosing the best client/server technology to adopt for our environment included:

- Training optimization and the experience of the current staff
- Concurrent processing in a distributed environment
- Enhancing security for confidential and critical data
- Moving application servers with minimal impact on clients
- Providing interoperability with existing client/server applications and tools
- Enhancing the portability of applications across architectures
- Operating across the HP internet on an enterprise-wide basis.

The evaluation of these factors by the task force resulted in a recommendation that the Open Software Foundation's Distributed Computing Environment be adopted as a standard technology for the implementation of client/server applications within HP.

DCE and the Evaluation Factors

DCE excels in the area of optimizing the training and experience of the current staff. One problem faced by early adopters of client/server computing was that no one could agree on the definition of what was a client and what was a server. This led to a plethora of home-grown and purchased solutions that did little to leverage the nature of the HP computing environment.

The definitions of client and server within DCE are consistent and tangible. A client refers to a program that calls a remote procedure. A server refers to a program that executes the procedure. There is no confusion with hardware or clients and servers on the same system or even a single program being both a client and server. The definitions are entirely consistent. In addition, these definitions fit perfectly within the context of HP's client/server application model.

DCE uses the remote procedure call (RPC) mechanism for client/server communication. This too is beneficial for programmers because it is an extension of a concept that every programmer knows and understands: how to write and execute procedures (or subroutines). In DCE, RPCs behave the same as local procedures. They are still distinct, modular collections of functionality with well-defined parameters that behave in a "black-box" fashion; send them the required parameters, and they reply in a predefined and predictable manner. Further, RPC is unobtrusive in that it hides the complexity of the distributed environment.

With RPCs, application programmers do not need to learn the intricacies of data networking or the particulars of a variety of application programming interfaces (APIs) to implement distributed applications effectively. Unlike other technologies, RPCs ensure that the operational considerations of network programming are both hidden and automatic. Lastly, the DCE APIs required to establish the client/server environment can be easily abstracted to hide even more from the application developer with the further benefit of contributing to consistency in the environment.

Using these concepts, and the tools described later, several of our application teams have experienced reduced implementation times in spite of the need for training in new technologies.

New issues and opportunities arise with the movement to client/server architectures. One of these opportunities is the ability to gain more effective use of computing resources on the network. Through the implementation of a threads facility, DCE gives application developers the ability to have a client call multiple servers simultaneously. In this way, an individual user executing a client program can invoke the parallel processing power of many servers. On the other end, the threads technology also allows servers to respond to multiple clients by processing each request within its own thread. This entails significantly less overhead than the creation and destruction process employed by many alternative technologies that require a unique server process per client. DCE threads are briefly described in the article on page 6.

DCE also incorporates a time service API to provide a consistent network-wide view of time. This service addresses the issues created when applications require time stamps to be reconciled across geographic boundaries or between systems.

Security is another area that raises both issues and opportunities. HP has traditionally used host password security and the security features inherent in the operating system to protect data and applications. If a user gained access to an application, then the user was presumed to have authority to execute any transaction performed by the application. In recent years, some application teams have supplemented host security with features provided by relational database management systems (RDBMS), but these too are usually limited in their flexibility. For example, a user that may have the ability to change a record when executing an authorized transaction should be prohibited from doing so with a database maintenance utility. Such discrimination is beyond the capability of most relational database management systems and requires added attention to system administration.

DCE extends the concept of security to the application itself. The principles of DCE security are authentication and authorization. DCE provides three services to enable the ultimate authorization of actions within a server. The registry service is a database used to keep information about users, groups, systems, and other principals* that can have an identity within the DCE security framework. The authentication service, based on the widely respected Kerberos technology from the Massachusetts Institute of Technology, is used to allow principals to authenticate themselves. The privilege service supplies the privilege attributes for a principal. These attributes are used by an access control list (ACL) manager within the body of a server to make authorization decisions. Using an ACL manager, server authorization decisions can be as granular as business needs dictate. Back doors, such as maintenance utilities or rogue programs, are not possible because users have no access to the systems on which critical data is stored. This makes the properly authenticated and authorized transaction the only vehicle by which a user can affect the database. Security and ACLs are also described in the articles on pages 41 and 49, respectively.

In addition to authentication and authorization, DCE provides features to protect both the integrity and privacy of data transmitted over a network. These features can be invoked by clients or servers when the sensitivity of business data dictates that such precautions are prudent.

Another challenge of the environment is change. Data centers are consolidated and moved, and hardware within the centers is replaced on a regular basis.

DCE provides a flexible, scalable directory service that can be used to apply human readable names to objects such as servers. Servers record their binding information at startup. Clients then locate servers wherever they may be. Multiple directory types permit great flexibility for the application developer. For example, the corporate telephone directory may have replicated instances of the server at many locations. Should one server fail, a client can automatically bind to another. In the case of an online transaction processing system, the one and only server can be found reliably by a client even if it has been moved temporarily after a disaster. Both of these cases can be accomplished with no changes to the client or the user's system configuration. DCE's global directory services are described in the article on page 23.

* A principal can be either a human user or an active object (machine, file, process, etc.).

Hewlett-Packard was a significant contributor to the technology suite that makes up the OSF DCE definition. One of the most important contributions was the RPC mechanism.

DCE's RPC is a compatible superset of the Network Computing System (NCS) from what was once Apollo Computer. The principles upon which the two solutions are based remain the same. They include platform independence and platform unawareness.

DCE platform independence comes from the fact that it runs on all computing platforms in common use within HP's IT environment: HP 3000 computers, HP 9000 workstations, Intel-based Windows® 3.1, and Windows NT. Platform unawareness comes from the fact that application programmers only need to concern themselves with the platform they are working on. Thus, when a developer codes a client, there is no need for the developer to be concerned with what platform the server will run on. Conversely, the server developer does not need to know what platform the client is using. Thus, an application client running on a desktop PC can send a byte string or pointer to a server running on a PA-RISC platform even though the data representations on the two systems are different. Fig. 2 shows a typical configuration of some of the components in HP's DCE client/server environment.

RPC provides the added benefit of interoperating with clients and servers already implemented using NCS. This provides a transition for applications to the more robust world of DCE.

HP operates one of the world's largest private Internet Protocol (IP) networks. The final criterion used by the task force was that the client/server technology chosen must operate

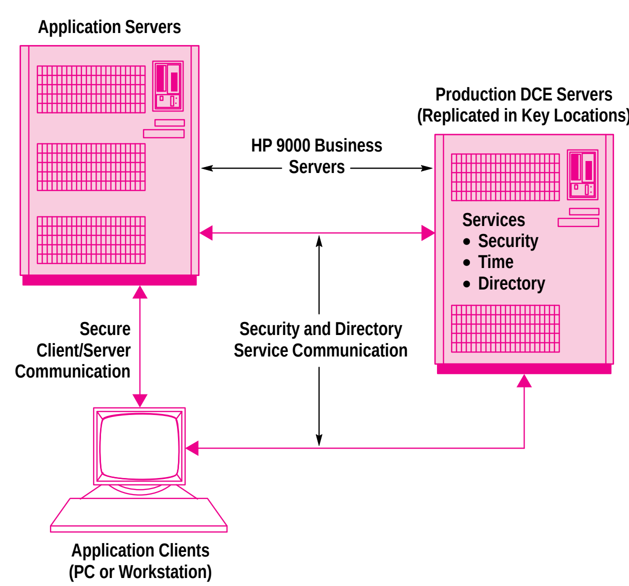

Fig. 2 The new client/server environment for information technology operations.

on HP's internet in an enterprise-wide fashion. DCE was designed to operate in the IP environment in a scale well above the size required for HP's enterprise.

The task force concluded that the adoption of DCE as a standard technology would enable some significant benefits including:

- Replacement of batch store-and-forward applications with OLTP
- Encapsulation of legacy code and data into servers that can be accessed by GUI clients
- Implementation of client/server applications with minimum training
- Abstraction of much, if not all, of the infrastructure so that application teams can concentrate on the business aspects of applications
- Implementation of enterprise-wide robust security
- Movement of servers between systems without impacting the client or the configuration of the client's host system.

Building HP's DCE Infrastructure

Because of the scope of DCE and the scale of problems that DCE addresses, careful planning is required when deciding how to deploy DCE. We found that the best approach is to start with the customers. As a group responsible for delivering DCE to HP's information technology community, we defined several categories of customers:

- End users. These are the people who interactively use applications.
- Application development teams. These are the people responsible for designing and constructing applications in response to some stated business need.
- Application administrators. These are the people who administer and support business applications in production.

Our group, which is the client/server tools group for HP's corporate network services department, has a long history of providing application data transfer solutions to each of these types of customers. The lessons gained from this experience are simple, even intuitive. End users want technology to be as absolutely invisible as possible, application development teams want to focus on their specific business problems, and application administrators want tools and support. In other words, whenever users must rely on technology to provide a solution, they want to be consumers of the technology and like all consumers, they demand certain things from a technology supplier such as:

- A higher level of abstraction than is usually provided by the technology
- The ability to make necessary, simplifying assumptions
- A consistent level of service in all cases.

Given these requirements, HP has chosen to deploy DCE in the form of an infrastructure. This infrastructure is known as the HP DCE service. In corporate network services parlance a service is an infrastructure with some specific properties. The prime reason for the term service is that the entire effort is focused upon meeting the needs of our customers, the people of Hewlett-Packard. The term service has connotations of careful planning, standardization, published guidelines, and customer satisfaction. It does not have connotations of central control, corporate mandate, or arbitrariness.

Finally, a service is a process as much as it is a tangible solution, and it recognizes that as the sophistication of its customers and their business problems grow with time, so too will their expectations.

Associated with every service is a value proposition. The value proposition formally defines the customers, the benefits provided to the customers, and the cost of these benefits.

Our experience designing and constructing a DCE infrastructure can be summarized very succinctly:

- The infrastructure must accommodate diversity.
- The infrastructure must provide consistency.
- The infrastructure must grow experience.

Accommodating Diversity

Identifying diversity is a crucial aspect of customer awareness. The customers of a distributed computing solution come from all parts of the organization with distinct requirements. We identified four categories that the DCE service must accommodate.

The first category is network performance. Customers of the DCE service are, by definition, users of HP's IP network infrastructure. Because of differences in networking technology, customers of the HP internet can realize a difference in both bandwidth and transit delay exceeding two orders of magnitude. Given the request/reply nature of the RPC protocols, this difference will also be reflected in every DCE operation. The lesson of this category is that the infrastructure must not make assumptions regarding the motion of packets.

The second category is application scope. Many business applications are truly enterprise-wide in scope in that there are tens of thousands of clients and massive, dynamic replication of the application servers. Many business applications are only deployed to a single group or department where there are perhaps a few dozen clients and only a single application server is required.

The third category is the different types of application users. Some users use data entry applications in which a small number of transactions are constantly performed to add or modify data. Other users use data query applications to perform a modest number transactions to read data. Other applications provide decision-support services, which typically allow users to perform ad hoc transactions that read data. Finally, some applications serve noninteractive clients that typically invoke large numbers of transactions that read, add, and modify data.

The fourth category is the geographic dispersal of the enterprise. HP does business in several countries all over the world. What this means is that nighttime or weekend service outages cannot be tolerated.

We learned from this kind of analysis that enterprise-wide is a one-dimensional term, and real enterprises are not one-dimensional. We have added the terms enterprise-deep and enterprise-broad. Enterprise-deep addresses the diversity of application users because it acknowledges that a successful infrastructure will accommodate every type of user. Enterprise-broad addresses the diversity of network performance and application scope by acknowledging that all business processing must be accommodated. In addition, today's business processes often cross company boundaries. In these situations, it also necessary to be enterprise-independent.

HP's DCE service accommodates diversity in several key ways. The policies and guidelines for the assignment of registry and namespace (DCE cell directory service) objects support massive replication of application servers. This allows DCE to be used as a foundation for truly enterprise-wide computing. The support model spans all organizations and all time zones, and no customers are ever treated as second class. The subscription model allows all customers to gain access quickly and easily. A subscription-based service provides convenient focal points for satisfying service requests. Finally, policies and guidelines delegate control to the appropriate level. Thus, since DCE is a distributed computing solution, its administration must also be distributed.

Consistency

Providing consistency is a crucial aspect of customer satisfaction. Despite their diversity, all customers are consumers of the same technology. They all demand a complete and comprehensive solution. Furthermore, the biggest return on IT investment comes from building on a consistent foundation that encourages resource sharing and leverages off other infrastructures. As in the case of diversity, the best place to start is with the customers.

End users have specific requirements regarding consistency. Since they must sit in front of and interact with the applications, end users are best served when all applications based on DCE offer a consistent interface with respect to DCE. A consistent interface offers equivalent dialog boxes for performing the standard tasks of DCE login and credential refresh. A consistent interface also offers an equivalent mechanism of dialog boxes or configuration files for server binding and server rebinding. Since DCE is an enabling technology, end users are best served when they can access a variety of DCE applications using their unique, enterprise-wide identity. Gaining credentials should be a side effect of being an employee of the organization, not a side effect of being a user of application X. Finally, there are standard tasks, such as password administration, that all end users must perform and are best served when they all have access to a standard set of tools.

Application development teams have specific requirements regarding consistency. Since application development teams are responsible for incorporating DCE functionality into applications as part of a business solution, they are best served when they can make the necessary simplifying assumptions. Application teams do not want the burden of acquiring and administering the core servers that provide the DCE security service and the DCE cell directory service. Removing this burden is especially important for teams who develop applications that must scale to serve the entire enterprise.

Application development teams also benefit from the ability to use tools that abstract the native DCE APIs. These tools dramatically reduce implementation time, and if they are standard and consistent, training is leveraged across application team boundaries as well as across applications.

Finally, application development teams benefit from the ability to leverage from established best practices and established experts. Despite the common misconception that best practices and experts are an attempt to constrain teams, experience has shown their advantages. Code reuse and resource sharing improve because similarity can be leveraged. Also, business implementation time is less because the need for retraining is reduced, and application quality increases because teams refine, improve, and reuse their skills.

Application administrators have specific requirements regarding consistency. As is the case with application development teams, administrators are best served when they can make the necessary simplifying assumptions. If an end user approaches the administrator to gain access, the administrator should be able to ask the end user's principal name and then perform the appropriate application-specific ACL administration and group management. The nonapplication-specific tasks of requesting an end user principal and machine principal and obtaining properly licensed copies of the DCE software should be left to the end user. The benefit to the application administrator is vastly reduced workload because the administrator only deals with the application. In addition, registry objects such as groups are leveraged across application boundaries.

Application administrators also benefit from the ability to leverage the best practices and standard tools. If DCE applications use DCE objects such as registry groups and namespace entries in a consistent fashion, retraining is minimized and a large cause of administrative errors is reduced.

Hewlett-Packard's DCE service provides consistency in many ways. Cell boundary decisions are weighted in favor of larger cells to promote genuine enterprise-wide computing. Tasks associated with DCE cell administration have been abstracted into high-level tools based upon the service's subscription model. These tools automate and hide specific, low-level DCE tasks. For example, the task that corresponds to an application subscription creates principals, groups, and accounts for the application's servers, creates namespace entries for the application, and modifies all associated access control lists. The benefit of this abstraction is the consistency that it ensures because the actual registry and namespace objects are generated and administered in a standard, documented manner.

Growing Experience

Growing experience, which means making both application developers and application users successful, is a crucial aspect of realizing the business benefits of DCE. Clearly, an infrastructure that is not used is useless. Growing experience is a two-step process that never ends. The first step is to identify barriers, and the second step is to remove these barriers by any means necessary. Such means include, but are not limited to, the development and deployment of custom tools, the abstraction of DCE tasks to better suit existing business practices, and exploitation of the fact that DCE is already one of its own best customers. The need for custom tools is by no means a negative reflection on DCE, but simply an acceptance of the fact that no single solution can do everything for everybody. Abstraction is simply a way to make DCE fit the business rather than forcing the business to fit DCE. Taking advantage of DCE means understanding that everything in DCE is basically a DCE object accessed by a client through an interface and protected by an ACL.

Application developers face a variety of barriers. The most traumatic barrier stems from the large number of new technologies directed towards development teams. Keep in mind that in most organizations, new technology really means new to the organization. In Hewlett-Packard, most IT application teams are new to writing distributed applications using DCE's client/server split or RPC paradigm. Our distributed applications have traditionally been based on file transfers or message passing. The learning curve for all of the technologies associated with DCE is nontrivial, especially when the development tools associated with the technologies are still evolving. The consequence to application developers is that creating the first DCE application with out-of-the-box DCE, even an evaluation application, is usually a difficult task. The risk is that IT application teams will not consider using DCE.

Application developers also face barriers when testing or deploying applications. The richness of DCE offers the developers an often bewildering variety of choices such as different ways to take advantage of the namespace or different methods of allocating registry objects to take advantage of DCE security. Without guidelines, established practices, and assistance some teams will simply try anything and then fail. Reports of these failures usually travel faster and wider than reports of teams that succeed.

HP's DCE service removes these barriers in three key ways. First, the service provides a DCE development library that abstracts the native DCE APIs into two very high-level API routines that include one call for the client and one call for the server. Second, the service offers a custom version of the OSF DCE programmer's class, which focuses on HP's IT environment. Third, the service offers consultants who can help other entities start DCE projects. A typical consulting venture involves the creation of Interface Definition Language (IDL)† files, a skeleton server that takes full advantage of security and the namespace, and a skeleton client that can bind to the server. After this is all done the application team only has to add the application code between the curly braces. The best part of DCE is that it allows one to distribute an application without worrying about how to do it.

Application administrators face barriers because for DCE applications, there will be DCE related tasks that they must perform either directly or indirectly in a production environment. Although production-quality DCE applications do not require much attention, there are still issues that can arise. For example, there is the occasional administration of endpoints and namespace entries in server failure cases, the occasional administration of server keytab files, and most of all, the administration of the application's ACLs used to control authorization. The out-of-the-box DCE tools are cumbersome and error-prone, and worst of all, they are fairly low-level and require a fairly detailed knowledge of DCE concepts. The risk is that DCE applications can acquire an undeserved reputation as being costly and difficult to support in production.

HP's DCE service removes these barriers by providing custom OSF/Motif tools to ease these DCE related tasks. Also, the published guidelines and best practices that bring consistency to DCE applications can help to grow experience by reducing the need to retrain.

System administrators face barriers because of the complexity of one of the most common tasks in a growing, maturing DCE cell. Since DCE regards each physical machine as a principal with its own authenticated identity, configuration must be done on each machine when adding it to the cell. The out-of-the-box tools have two significant problems. First, they require coordination by the system administrator for root access. Second, if configuration is done across the network, both the root password and the DCE cell_admin password are exposed. These are unacceptable security holes especially for a system that is intended to serve as a foundation for secure distributed applications. In addition, many machines already run NCS applications, and these applications must not be impacted by the migration to DCE. As a result, installation and configuration are tedious and potentially insecure. The risk is that deploying DCE throughout the enterprise will be viewed as slow and expensive.

Hewlett-Packard's DCE service removes these barriers in three key ways. First, we have developed a scheme that allows a machine to be remotely and securely added to a cell. In particular, this scheme does not expose the operating system or DCE passwords across the network. It also doesn't require any effort on the part of the system administrator other than to install an HP-UX* fileset. Second, we are integrating the DCE client software into a common operating environment for machines that run the HP-UX operating system. Third, we provide a PC-DCE†† license server to ease the distribution of PC-DCE.

End users face a variety of potential barriers. Although it is the job of the application developer to shield DCE from end users, they will still be aware of their DCE principal. Thus, end users should have minimal training on obtaining and refreshing their network credentials as well as managing their principal. Out-of-the-box DCE does not include a standalone password management tool, and users must actually run rgy_edit and modify their account. Also, integrated login solutions in which the operating system and DCE logins are combined are still evolving (see the article on page 34). The risk is that deploying DCE applications to large numbers of end users can be slow, tedious, and expensive, and the end users who are exposed to too much DCE because of poorly constructed applications may assume that all DCE applications are difficult to use.

HP's DCE service removes these barriers in two key ways. First, we provide tools on both the HP-UX operating system and the PC to ease password administration. Second, the service's subscription model provides a simple focal point for requesting and obtaining a DCE principal.

† IDL is a language similar to C that allows developers to specify the interfaces that tie client and server applications together.

†† PC-DCE is an implementation of DCE that runs in an MS Windows environment.

The final barrier to growing experience comes from two groups of people. The first group believes that client/server is really just remote SQL, and the second group believes that client/server just means the motion of bits over the network. Remote SQL is certainly a fine solution for some business problems. However, it is important to remember that vendor lock-in, client-side awareness of database schema, network performance on the WAN, and the usual lack of network security could be problems in dealing with remote SQL. Although DCE does move bits over the network, and other approaches such as message passing using sockets may be an adequate solution for many business problems, the issues of WAN performance, architecture differences, code sharing difficulties, code maintenance difficulties, and the usual lack of network security could be problems in other approaches. When making technical decisions, being dogmatic is usually the first step towards being unsuccessful. The goal is not to dictate or even to impress, but to educate and promote a community in which decisions are made objectively.

Hewlett-Packard's DCE service addresses these barriers in two ways. First, we offer classes on all aspects of DCE and its use. Second, service subscribers can access all published information using Worldwide Web browsers.

Conclusion

Perhaps the best way to get a clear perspective of HP's DCE service is via analogy with other well-known infrastructures. Consider customers of a WAN. Everyone wants access and a consistent service model such as enterprise-wide IP connectivity. Consider the technology that is used to build a WAN. Now consider a successful WAN infrastructure. It is much more than the technology (i.e., routers, bridges, etc.) used to build it, it is a also a distributed creature that requires distributed administration and coordinated planning and guidance. Furthermore, there is no distinction between a test network and a production network. The network is simply an infrastructure that supports all phases of the software lifecycle.

Another valuable analogy is the interstate highway system in the United States. In the 1950s automotive technology boomed and the resulting cost structures allowed many people to own a car. This produced a fundamental change in American society because of the freedom, power, and movement of goods and services the automobile permitted. Perhaps the biggest contributing factor was the interstate highway system. The interstate highway system really wasn't about automotive technology. It was about use and access. Distributed applications are the same. The focus shouldn't be on distributed computing technology but on use and access.

DCE is a powerful and impressive collection of software technology. It offers attractive solutions to the kinds of business problems that most large organizations must address. Our experience has demonstrated the following:

- It is OK to experiment.
- It is important to allow a few key people to become industry-level experts. These are the people who should be responsible for service management.
- DCE should not be managed by regulatory practices.
- It is imperative to abstract anything if the result better fits the business needs and business practices.
- Activities should not be done in secret or kept secret.
- A service such as DCE is as much a continual process as it is a tangible solution.