PART THREE

UPPER LAYERS

6 OPEN SYSTEMS APPLICATIONS

An architectural characteristic of open system networks (as opposed to proprietary networks) is the assumption of a set of generic, or generally available, applications that serve as building blocks or "tools" for constructing more complex distributed system applications. These are generally regarded as "applications" in the TCP/IP world; in OSI, they are called *distributed application services*.

Distributed application services, whether OSI- or TCP/IP-based, share some characteristics in common. For example, irrespective of whether one describes a file transfer application that uses TCP/IP's File Transfer Protocol (FTP) or OSI's File Transfer, Access, and Management (FTAM), the file transfer application will have at least the following characteristics:

- An end-user interface that provides a human or another application with the means to enter commands that direct the application to send files to and receive files from a remote host, list or change directories, rename or delete files, move files from one directory to another, etc. (There will also be a means for the application to inform the end user of the results of the actions, successful or failed.)
- The means of performing input to and output from mass storage device(s) (disk-tape).
- The means of transferring the files and file-related information between hosts.

Thus, for both OSI and TCP/IP, there are local and communications components to every distributed or end-user application. The *local* *component* of an application consists of the end-user and/or programmatic interfaces to the application, functions that access local input/output resources such as disk, and access to computing resources of a host machine. The *communications component* consists of the entity that provides distributed communications capabilities to the distributed application; OSI calls the communications piece an *application entity* (AE) and the sum of the parts that comprise a distributed application an *application process* (AP). (See Figure 6.1.)

Although the notion of an application process is common to both TCP/IP and OSI, their approaches to constructing application entities is different. In TCP/IP, each application entity is composed of whatever set of functions it needs beyond end-to-end transport to support a distributed communications service—e.g., the exchange of mail, remote file access, or file transfer—into the protocol(s) of that particular application; in other words, each application process builds in its own, often unique, set of tools, commands, and exchange mechanisms. File Transfer Protocol, for example, has an entirely different set and way of exchanging commands and replies than "Internet mail" (sometimes called "SMTP/822" mail, referring to RFC 822, which describes mail message contents, and RFC 821, Simple Mail Transfer Protocol [SMTP]; see Chapter 8). There is no common notion of establishing application "connections" among Internet application services,[1] and no common reliable

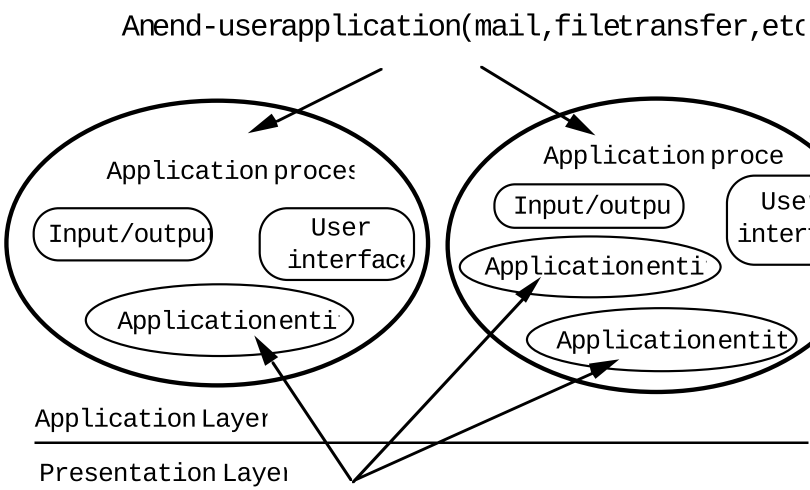

FIGURE 6.1 Application Process Structure

1. In the implementation of the original set of application or *host protocols*, the *initial*

transfer and dialogue-control service, nor is there a true, common network-programming language or a common remote procedure call mechanism. (The *External Data Representation* [XDR; RFC 1014] and *Remote Procedure Call* [RPC; RFC 1059], developed in conjunction with *Network File System* [NFS; RFC 1094; Sandburg 1988], are, in a sense, application service elements but are not formally used as such outside of NFS.) This is not intended as a criticism but merely to illustrate that, by and large, each Internet application process builds in what it needs and assumes only that an underlying transport mechanism (datagram or connection) will be provided.

In OSI, each distributed application service selects functions from a large common "toolbox" of *application service elements* (ASEs) and complements these with application service elements that perform functions specific to a given end-user service—e.g., mail (message handling) or file transfer (see Figure 6.2). Conceptually, application entities in TCP/IP have a single service element, whereas application entities in OSI may have many.

An application entity that supports OSI's File Transfer, Access, and

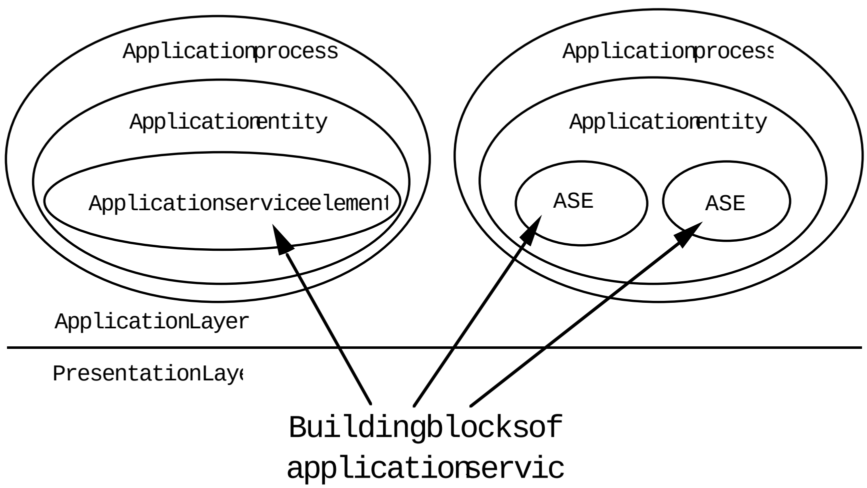

FIGURE 6.2 Composition of an Application Entity

connection protocol (ICP; RFC 123) conceptually served this purpose, which we shall see in Chapter 10 is provided by the association control service element in OSI. While the initial connection protocol was implemented separately in a few implementations, most folks implemented the functions used from this conceptual "inner protocol" directly in FTP, Remote Job Entry, and TELNET rather than using it as a layer or independent module on top of the connection-oriented, host-to-host protocol, which later evolved into TCP.

Management thus has at least one application service element (FTAM) that is different from OSI's Message Handling System (MHS) but both use the same "tool" for establishing communications (the association control service element discussed in Chapter 10). Similarly, the OSI Directory service and common management information service elements use the association-control "tool" as well as a common remote procedure call "tool" (the remote operations service element, also discussed in Chapter 10). In such configurations, one application service element is a *user element* of other ASEs (see Figure 6.3).

Extending the "tool" analogy a bit further, to tighten a screw or bolt on an automobile, a mechanic (the end user) uses tools like an electric screwdriver and a ratchet wrench (ASEs); both of these require another tool (another ASE) to complete the task: the electric screwdriver requires a bit, and the ratchet wrench requires a socket. The screwdriver and wrench are thus user elements of the bit and socket ASEs, respectively. Absent the bit and socket, these tools may be useful, perhaps, but only as hammers.

Note that in OSI, application service elements that provide intuitively obvious "end-user" application services—message handling, directories, file transfer—can themselves be regarded as tools of multifunction distributed applications; a messaging application may, for example, make use of both message handling service and directory service elements, the latter invoked for the purpose of obtaining inter–mail application routing

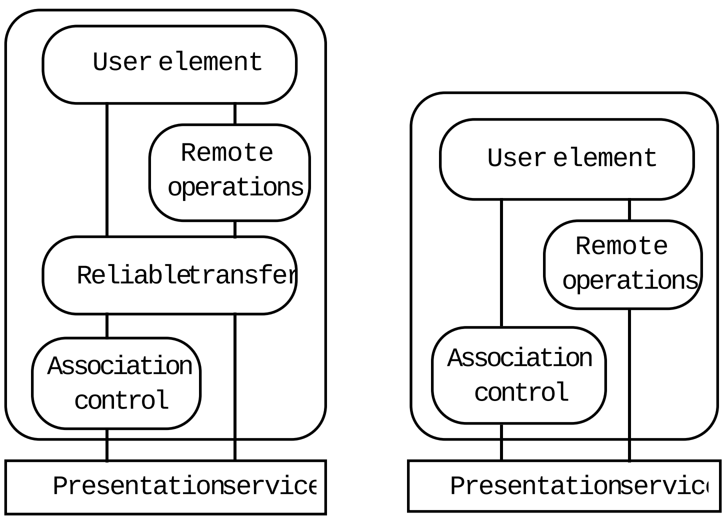

FIGURE 6.3 ASE Use of Other ASEs

information or mail addresser. Moreover, an application service element such as a message handling service element in one instance provides an "end-user" application—mail handling—but also acts as a "tool" for other application services, such as electronic data interchange and office document interchange. Finally, an end-user application is not restricted to using only those application service elements that are made standards by CCITT and ISO/IEC: "home-brew" user service elements can be written in ASN.1, and these, too, may use standard application service elements.

Given the contrast in styles, the TCP/IP approach to building applications has sometimes been called a vertical one: each application was developed independently, "top" (i.e., end-user application service) to "bottom" (i.e., transport). The OSI approach, consistent with the pervasive notion of layering, has been called a horizontal approach (end-user applications developed using a common application-development infrastructure; see Figure 6.4).

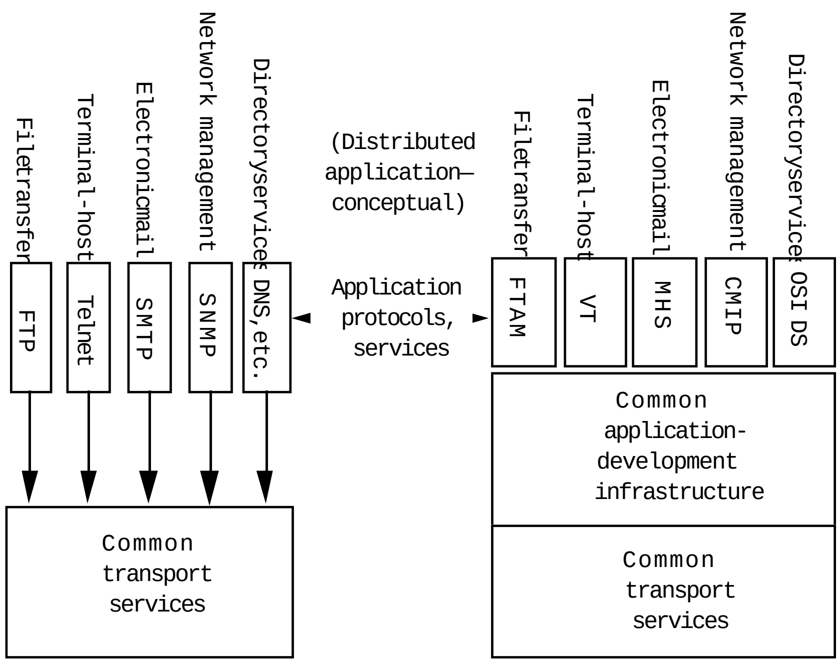

FIGURE 6.4 Comparison of Internet and OSI Application Structure

Distributed Application Services

OSI is clearly not the exclusive distributed applications environment for all forms of distributed applications services. Applications developed over TCP/IP and other protocol suites tending toward "openness" satisfy many essential end-user needs and are likely to continue to satisfy these needs along with OSI. This diversity is not a truly bad thing (see Chapter 17), since, historically, diversity and competition have often led to improvements and landmark developments in computer communications. It is also important to note that the applications-development infrastructure of OSI—the upper layers—may easily be ported over to existing transport service infrastructures; this is especially true for TCP/IP, and the success of enterprises of this sort is discussed in subsequent chapters.

AHA *The generality and flexibility of the OSI applications-development infrastructure is not without cost, and detractors continue to berate it, criticizing it for its complexity (in both specification and implementation) as well as the excessive processing and memory overhead that is associated with operating distributed applications in this environment. Some of this criticism must be tempered by the fact that many of the existing implementations are "first-generation," and it is to be expected that further field and implementation experience will yield leaner, faster software. Some of the criticism is well-founded, and one can only hope that those who have implemented OSI-based applications will work to see that the standards are revised to correct major shortcomings. Despite the criticism, OSI continues to be a catalyst for the development of protocol and service frameworks for interesting distributed services.*

OSI's applications-development infrastructure provides a convenient and multipurpose framework for the development of a wide range of distributed applications; a sampling of these is described briefly in the following subsections. Some are variations on a familiar "theme"—electronic mail, file transfer, directories—whereas others are more forward-looking.

Electronic Mail and Message Handling System (MHS) "E-mail" is the ability to send and receive the electronic equivalent of written correspondence typically delivered through a postal agency. In addition to simple textual mail messages, OSI message-handling services facilitate the electronic exchange of documents that, in principal, can be comprised of facsimile, graphics, or even speech or video. OSI's Message Handling System is also likely to provide a distribution platform for office and electronic document exchange.

Electronic Data Interchange (EDI) The ability to exchange business documents—e.g., standard "forms," such as invoices, purchase orders, payment orders, and customs declarations/reports—is an international concern, especially for the European Economic Community. OSI standards for these forms are consistent with the United Nations/EDIFACT Standard, and the forms can be transferred using OSI's Message Handling System (see Chapter 8).

Office Document Architecture/Interchange (ODA/ODIF) The ability to exchange documents containing text and graphics—spreadsheets, page layouts from desktop-publishing applications, papers produced using word-processing applications—between like and dissimilar applications (e.g., from brand X word processor to brand Y) without losing any of the document's contents is accommodated within OSI's office document architecture, also known as the CCITT T.400-series Recommendations for document Transfer, Access, and Manipulation (DTAM). The office document architecture specifies document structures, interchange formats, character content architectures, and content architectures for raster, tile raster, and geometric graphics (ISO/IEC 8613: 1989), in many parts).

Directory Services Like the operator-assisted directory services offered through the telephone network, this service includes the ability to match names with addressing information. In addition, OSI offers a comprehensive registration and identification infrastructure that helps individuals, applications, and organizations acquire information ("attributes") that provides a more detailed characterization of things ("objects") that are named. The OSI Directory is expected to serve as a repository for information that characterizes people, applications, mail systems, management systems—virtually any information that one wishes to register and make publicly available.

Distributed File Systems The ability to access and manage file systems mounted on remote computers is an integral part of distributed processing today. OSI offers new tools in this area and also an equivalent environment upon which to run existing and eminently popular tools such as Network File System.

Network Management Network management provides the ability to monitor the status and use of resources of a distributed processing environment—hosts, bridges, routers, the transmission facilities that interconnect them, and software (application as well as protocol processes) resident on these machines—as well as the ability to detect, isolate, and circumvent problems that might arise in any of these network components. OSI has a comprehensive "common management" applications infrastructure that provides monitoring, analysis, accounting, and diagnostic services and more.

Remote Database Access OSI provides a generic remote database architecture—protocols and services—for client/server interactions (dialogue, transaction, and data-resource management) and also provides for a set of "specializations" that allow one to further define the parameters of remote database access operations to accommodate a specific remote database language—e.g., the standard query language SQL (ISO/IEC 9579: 1992).

These are but a few of the areas of distributed processing and information technology that can use OSI. Of course, many of the capabilities OSI offers have easily identifiable TCP/IP counterparts. Applications based on the Simple Mail Transfer Protocol (see Chapter 8) offer textual electronic-mail correspondence similar to that offered via OSI's Message Handling System and, with recent extensions that are today experimental, may enable multimedia messaging as well. The Domain Name System offers a host name– to–IP address service similar to an application-name–to–presentation-address capability offered by the OSI Directory; other Internet applications—such as WHOIS, FINGER, Archie, the Wide Area Information Service, and the WorldWideWeb—offer a variety of resource locators and information services (see Chapter 7).

In certain application areas—messaging and directory services—many believe that OSI adds value to existing TCP/IP applications. In other areas—for example, distributed file services and window-based systems—existing services like NFS are considered superior to application services developed specifically for OSI. In fact, the utility and widespread application of services such as NFS and the X Window System have provided the basis for the development of conventions and eventually standards for operating these applications over OSI stacks.

This is true in the area of network management as well. Although there are those who believe that OSI's "common management" is superior in many ways to the Simple Network Management Protocol, applications based on the SNMP are sufficiently popular that they are now used to manage dual-stack (OSI and TCP/IP) internets (actually, the Simple Network Management Protocol framework provides network management for large-scale internets that are IP- as well as XNS/IPX, AppleTalk-, and OSI-based). And recent extensions to the Simple Network Management Protocol arguably improve on an already useful and proven commodity.

There are also areas where OSI will offer application services that are to date not addressed in TCP/IP: OSI's progress on transaction processing, office document architecture, and electronic data interchange standards is followed with great interest by the Internet community.

Conclusion

This chapter has described how OSI and TCP/IP differ significantly in their approaches to constructing distributed system applications. OSI asserts that distributed applications operate over a strict hierarchy of layers and are constructed from a common tool kit of standardized application service elements; TCP/IP makes no such assertion, insisting only that distributed applications operate over a common end-to-end transport service. Which approach is better? OSI's is general and flexible, and its emphasis on modularity and reuse of common mechanisms comports well with current object-oriented models of application development; but generality costs, and as is the case with many aspects of OSI, the inefficiency of a too-literal implementation may outweigh the theoretical benefits. TCP/IP's "apply to affected area as needed" approach is more application-specific and may lead to the redundant implementation of the same function in many different applications; but in most cases, the greater efficiency and performance of the resulting applications outweigh the potentially greater inefficiency of application software development. It is interesting to observe that the most promising new work on the implementation of the "upper layers" of OSI (see Chapter 11) combines the functions of the application, presentation, and session layers into a common library of modules that are included—or not included—in applications, as each demands. This is yet another example of OSI's learning and borrowing from TCP/IP—an exchange that takes place in both directions much more readily (and frequently) than most people suspect (or are willing to admit).

The remaining chapters in Part Three examine application services and upper layers in a "top-down" fashion. Although this may seem contrary to the customary flow, a top-down approach has the advantage of allowing readers to deal first with easily recognizable and practical examples of services encountered on a daily basis—the use of a telephone, a postal service, a telephone book, or directory assistance—and gradually learn the technical aspects of how equivalent "electronic" services are provided across complex internets. Thus, Part Three continues by providing a comparison of application services offered by both architectures, focusing on function and high-level operation. Chapters 7, 8, and 9 examine three distributed application services—directories, electronic messaging, and network management. These are chosen because they are popular and easily recognized distributed applications and because there is considerable overlap between the OSI and TCP/IP counterparts with respect to services offered. From this sampling, readers are expected to gain an overall understanding of distributed processing in OSI and TCP/IP.

Chapter 10 examines three application service elements—association control, remote operations, and reliable transfer—that provide essential, or "core," services for distributed applications in OSI, and Chapter 11 describes the OSI presentation and session layers. A consequence of the rigid layering in the OSI upper layers is that many of the functions that an application may use or invoke are accessed through application service elements but performed elsewhere in the architecture (e.g., in the session layer). Chapter 10 introduces such functions at a conceptual or metalevel, whereas Chapter 11 describes how these functions are provided.

It should be noted that Chapters 10 and 11 diverge from the side-by-side analysis that is a convention elsewhere in the text, as the functions corresponding to the OSI upper layers are embedded in the Internet applications described in Chapters 7, 8, and 9 on an "as-needed" basis.