



OPEN SYSTEM INTERCONNECTION MODEL

UNDERSTANDING INTEGRATION

ABSTRACT

This paper introduces the basic models of communication such as publish-and-subscribe and synchronous versus asynchronous. It reviews the Open System Interconnection (OSI) model and uses that model to explain the success of the Internet and Internet-based integration.

OPEN SYSTEM INTERCONNECTION MODEL

UNDERSTANDING INTEGRATION

BASIC COMMUNICATIONS

The communications used by legacy electronic commerce are really quite simple. They should not intimidate you. The communications come in certain types, which we will categorize.

PRODUCERS AND CONSUMERS

Communication is sharing information. With communication, there is a producer of information and a consumer of information. The same resource, application or person can be both a producer and a consumer. With any particular message or packet of information, however, that resource is either producing the message or consuming the message.

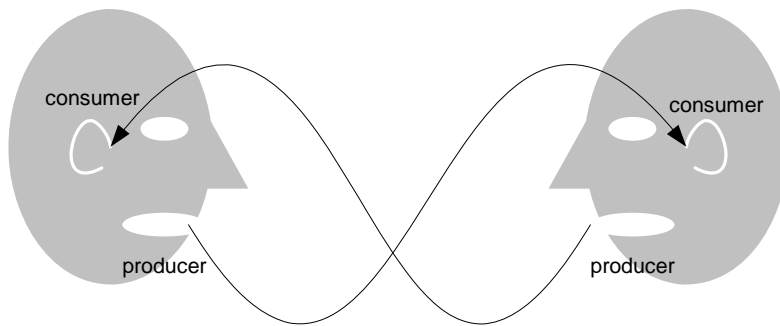


Figure 1 Communication involves producers and consumers of information

SYNCHRONOUS OR ASYNCHRONOUS

The communication is either persistent or not persistent. Either the channel is maintained like a telephone conversation or it is series of messages like mail. A proper term for persistent is synchronous, and asynchronous for not persistent.

In a conversation between two people, there is little distinction between producer and consumer. A synchronous arrangement makes sense for a conversation. In a classroom, there is a distinction between the producer or lecturer and the consumer or student. An asynchronous arrangement makes sense for a classroom.

In some situations, the client and the server need to constantly communicate. In such situations, a synchronous connection can be used. In most situations, however, the client and the server only need to know about changes. Asynchronous communications are usually much easier to implement and usually less taxing on network resources. TELNET is an example of a synchronous communication protocol. HTTP is an asynchronous protocol.

PUSH OR PULL

If the communication is asynchronous, it is either a push or pull arrangement. Either it is the responsibility of the producers to push their data to the consumer or it is the responsibility of the consumers to pull data from their producers.

For instance, suppose we wanted to send orders to our supplier and all we had was FTP. We are the producer; we are creating the orders. The supplier is the consumer; he is accepting the orders. We would have to decide who was transferring files to whom. In a push arrangement, the supplier would allow us FTP access to his server. We would log on to his FTP server and push (or PUT in FTP parlance) the orders to him. In a pull arrangement, we would give the supplier FTP access to our server and he would pull (or GET in FTP parlance) the orders from us.

As you can see, the push or pull arrangement is a trust and security issue. Most organizations are reluctant to give access to their systems, but are willing to gain access to another's systems.

PEER-TO-PEER OR HUB-AND-SPOKE

In a peer-to-peer relationship, every resource, application or person has direct access or connection to every other resource, application or person. As networks grow larger, this becomes very complicated, exponentially complicated. We discuss just how complicated later on in the chapter entitled "The Theory of Integration."

The alternative is to create a hub-and-spoke arrangement. Each resource, application or person (the spokes) goes through a single entity (the hub) to reach any other resource, application or person. The post office is an example of a hub-and-spoke arrangement.

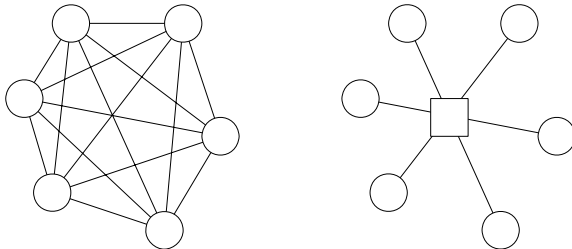


Figure 2 Peer-to-peer versus hub-and-spoke

Consider a large office. In a peer-to-peer arrangement, each person is responsible for getting documents to its consumer in the office. The producer and consumer can come to an arrangement. Either the consumer will go and push it under the consumer's door or the consumer will come and pull it off the producer's desk.

A postal service alleviates a lot of traffic. However, the postal service must resolve the push or pull arrangements with the spokes. In a post office scenario, the hub puts information it produces and pulls information it consumes. The office worker walks down to the post room and puts items in a big bin for delivery and pulls items out of his labeled mailbox. In a post delivery scenario, the postman comes to the worker's office, pulls items out of the worker's outbox on the desk, and pushes items into the worker's inbox.

It becomes a matter of security arrangement. Either the spokes have access to the hub's system (post office arrangement) or the hub has access to the spoke's system (post delivery arrangement).

VAN EXAMPLE

Electronic data interchange (EDI) began as peer-to-peer. In the early days, General Electric dialed directly into General Motors to send or receive files. As EDI expanded, participants moved to a hub-and-spoke arrangement. EDI has traditionally worked over private networks called value added networks (VAN). (*Value added network* is one of those names like *life insurance*. You just have to applaud whoever came up with such a brilliant misnomer.)

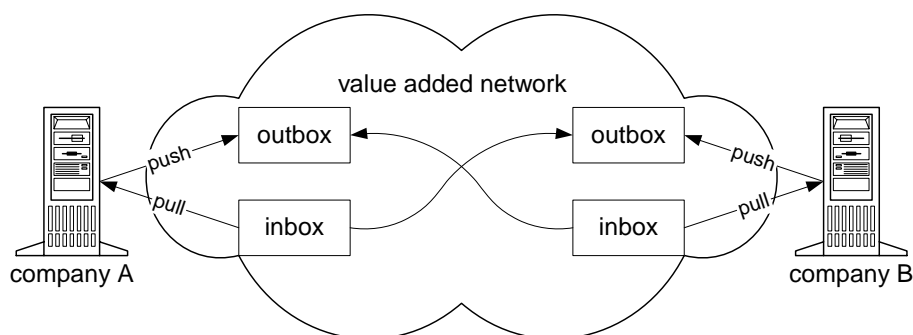


Figure 3 Value-added network

The companies are the spokes and the VAN is the hub. A VAN works in a post office arrangement. A company dials into the VAN and posts or pushes all the outgoing documents it has. The company looks in its inbox to see what documents it has received and pulls them down. The company then disconnects. It is the VAN's responsibility to take documents from company outboxes and route them to the appropriate company inboxes.

DIRECT ROUTING OR PUBLISH-AND-SUBSCRIBE

VANs rely on a process of direct routing. That means that the producer knows the receiver and specifically addresses the message to that receiver. We will address specific e-commerce addressing schemes in a later chapter. We use direct routing in the post office and normal e-mail. That is, we address the message directly to the intended receiver. Direct routing is well suited for when there are only a few consumers.

The alternative to direct routing is publish-and-subscribe. In a publish-and-subscribe model, there are usually multiple consumers. The producer does not need to know who the consumers are. The producer simply notifies publicly the types of information that it produces. It is the responsibility of the consumers to subscribe to the producer. The infrastructure should handle the rest. Newsletters and list servers are good examples of a publish-and-subscribe model.

QUEUING

Queue servers play an important role in enterprise communication infrastructures. Message queues provide guaranteed delivery and capacity management. For guaranteed delivery, the queue maintains a persistent form of the message until it gets confirmation that the consumer received the message intact. That way, the queue has the ability to recover and try again should the power fail or the network connection break sometime during transmission. Having such capability as guaranteed delivery as part of the infrastructure alleviates the need to build it in every piece of software.

Capacity management is a bit different. Think of the lobby of a large office building and the constant traffic of people moving through it. Sometimes lines form in the lobby for the restaurant or

the elevators. If it just started raining outside, lines will form at the front door going out. Network queues work much the same way as the lobby. Applications can become temporarily bogged down with a particularly large processing job or there may be an unusually large peak in requests for service. As a result, message requests back up in the queue. Without the queue, the application would take the messages it could handle at the particular time and lose the rest. The queue enables the application to overcome these peaks and valleys. However, if the message is consistently larger than the capacity of the application, there is little a queue can do.

OSI MODEL

DEFINED

The International Organization of Standardization (ISO) introduced the Open System Interconnection (OSI) reference model in 1984.

7	application
6	presentation
5	session
4	transport
3	internetworking
2	data link
1	physical

Figure 4 OSI model of networks

The OSI reference model is used to explain and compare network applications. It consists of seven layers: application, presentation, session, transport, internetworking, data link and physical. We will use this model to explain the evolutionary process we are witnessing. The process is simple: application development has been gradually assuming more and more of the stack.

To appreciate the impact of standardizing to the Internet, Table 1 lists just some of the more popular networking protocol hierarchies available from the major companies.

OSI Layer	Apple Computer	Banyan Systems	DEC DECnet	IBM SNA	Microsoft Networking	Novell NetWare	TCP/IP Internet	Xerox XNS	OSI Protocols
Application Layer 7	Application Programs and Protocols for file transfer, electronic mail, etc.								
Presentation Layer 6	AppleTalk Filing Protocol (AFP)	Remote Procedural Calls (Net RPC)	Network Management Network Application	Transaction Services Presentation Services	Server Message Block (SMB)	NetWare Core Protocols (NCP)	Application Specific Protocols	Control and Process Interaction	ISO 8823
Session Layer 5	AppleTalk Session Protocol (ASP)		Session	Data Flow Control	Network Basic Input/Output System (NetBIOS)	Network Basic Input/Output System (NetBIOS)	(Telnet, FTP, SNMP, SMTP, ICMP, etc.)		ISO 8327

Transport Layer 4	AppleTalk Transaction Protocol (ATP)	VINES InterProcess Communications (VIPIC)	End Communications	Transmission Control	Network Basic Extended User Interface (NetBEUI)	Sequence d Packet Exchange (SPX)	Transmissio n Control Protocol (TCP)	Sequence d Packet Protocol (SPP)	ISO 8073 TP0-4
Network Layer 3	Datagram Delivery Protocol (DDP)	VINES Internet Protocol (VIP)	Routing	Path Control		Internet Packet Exchange (IPX)	Internet Protocol (IP)	Internet Datagram Protocol (IDP)	ISO 8473 (CLNP)
Data Link Layer 2	Network Interface Cards: Ethernet, Token-Ring, ARCNET, StarLAN, LocalTalk, FDDI, ATM, etc. NIC Drivers: Open Datalink Interface (ODI), Network Independent Interface Specification (NDIS)								
Physical Layer 1	Transmission Media: Twisted Pair, Coax, Fiber Optic, Wireless Media, etc.								

Table 1 Various network protocols mapped to the OSI model. Source: Atlantic Technology Group, Inc.

Even in the 1990s, companies faced wide area network (WAN) challenges of integrating two or more of these network stacks. For instance, two companies merge. One company had implemented a DECnet network while the other implement IBM SNA. With e-commerce, one does not need to merge or acquire another company in order to face the challenge of integrating with their network. To continue with so many network standards, e-commerce would have to be built point-to-point. As we shall see later in our discussion of the theory of integration, a global point-to-point network would be unsustainable. Only by standardizing to one protocol is global e-commerce even possible. TCP/IP is that universal protocol.

A PROCESS OF EVOLUTION

XML and HTML are part of a larger evolutionary process of software development. That software development is evolving should surprise no one. Software is, after all, a very new art, and even older sciences such as physics and mathematics continue to evolve.

The OSI model gives a graphical explanation of why the Internet works. The Internet creates a coupling through TCP/IP between the lower levels (data link and physical) and the upper levels (application, presentation and session). Not too long ago, our developers had to create TCP handlers. Now TCP is assumed infrastructure. An Internet session protocol such as HTTP, FTP, SMTP or POP will run on any network, be it ATM over fiber optic or Ethernet over coaxial cable, so long as it supports TCP/IP.

One of the earliest and most successful Internet applications was electronic mail, or e-mail. Figure 5 shows an e-mail system mapped to the OSI model. Mail messages use a presentation format called MIME. Mail servers exchange MIME-formatted messages with each other through simple mail transfer protocol (SMTP). Mail clients access mail from the mail servers using post office protocol (POP).

application	mail server	mail client
presentation	MIME	
session	SMTP	POP
transport	TCP	
internetworking	IP	
data link	any network that supports	
physical	TCP/IP	

Figure 5 OSI and e-mail

E-mail is an example of special-purpose Internet application integration. POP is a one-to-one integration, that is, it integrates one type of application, an e-mail server, with one type of client, an e-mail client.

The Worldwide Web made the Internet a household word. The Web is the collection of applications on the Internet that use hypertext transfer protocol (HTTP) as a session layer and HTML as a presentation layer. Because of its overwhelming success, however, many people use the words *Web* and *Internet* interchangeably.

The Web moved the OSI stack assumption up two more layers. Whereas e-mail is a one-to-one integration, the Web allows many-to-one application integration. Any application that sends HTML presentation via HTTP session can now work with a web browser. This creates a many-to-one integration: one client to many applications. The web browser is thus called the universal client.

application	server	browser
presentation	HTML	
session	HTTP	
transport	TCP	
internetworking	IP	
data link	any network that supports	
physical	TCP/IP	

Figure 6 OSI and the Web

XML takes this success one step further to allow any-to-any application integration. Because XML makes data both human and machine readable, the application on the client side need not be a web browser.

To fully explain the success of XML, we must expand the OSI model. We will split level 6, presentation into two levels: syntax and semantics. Syntax, in this case, is the structure or rules of the presentation. The semantics are the actual metadata of the presentation. XML is a universal syntax. It has a set of simple rules for expressing data structures. Many different languages are built on top of XML. For instance, XCBL, CXML, EBXML, XEDI, BizTalk are all XML-based languages for e-commerce. Likewise, FIXML, IFX, FPML, XBRL, RIXML, MDDL are all XML-based languages for the financial industry. In this sense, XML is often called a meta-language since it is a language for

creating languages. For our discussions here, we will say that XML is the syntax and all those languages built on top of XML are the semantics.

What XML enables us all to do is standardize one more layer of the OSI stack. It allows programmers to assume the syntax of the data and focus on the semantics. There are unlimited numbers of alternative syntax such as character-delimited formats, fixed width formats, etc. When dealing with systems integration, we need to translate one presentation into another. Each syntax format requires its own special parser. If we standardize on a specific syntax, then we need only one universal parser. That parser handles all the heavy lifting so that programmers can focus on the specifics of translating the semantics. That is exactly what we have achieved with XML.

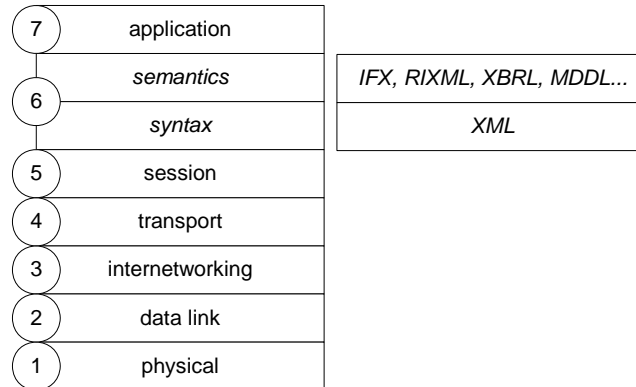


Figure 7 Extended OSI model

Not long ago, developers had to create their own TCP handlers. Now the transport layer and session layer are assumed to exist. With XML, developers can even assume the syntax layer exists and use off-the-shelf open-source XML parsers. Now we focus our debate on which semantics specification will become the universal standard.