

Demystifying Object Options

Whether you're a shop that prefers to build your own components or not, the good news is distributed computing is getting easier. Today, the field has narrowed to two competing architectures: J2EE and Windows 2000 DNA.



A year ago, you lost sleep over competing object models. Should you buy into Microsoft Corp.'s Component Object Model (COM) or standardize on more open models such as Sun Microsystems Inc.'s Enterprise JavaBeans (EJBs) or Object Management Group's Common Object Request Broker Architecture (CORBA) that pretty much everyone else was embracing? Today, the field has narrowed and it's more a question of competing architectures: the Microsoft Windows 2000 Distributed Internet Architecture, or DNA (see Figure 1), on one hand, and the IBM Corp. and Sun-Netscape Alliance Java 2 Enterprise Edition, or J2EE (see Figure 2), on the other. Either one of them can serve as the foundation of an organization's distributing computing environment—and both want to be the market leader.

Unfortunately, both of these architectures are still works in progress, and although the future seems to hold the

by Karen Watterson

STEVE ALEXANDER

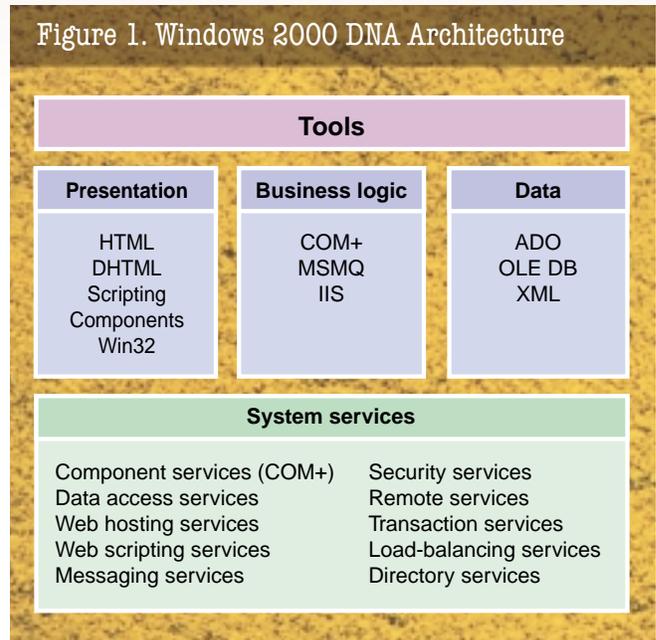
promise of an object-agnostic, service-centric world greased by technologies such as eXtensible Markup Language (XML) and Simple Object Access Protocol (SOAP), we're not there yet.

Components for Sale

We have made progress, though. Today, when you build distributed applications, it's becoming a lot easier to adopt a "best-of-breed" approach—buying some components, building others and accessing yet others embedded in other organizations' applications. Perhaps the world of "software IC chips" that Brad J. Cox outlined in his seminal *Object-Oriented Programming: An Evolutionary Approach* (published by Addison-Wesley Publishing Co., 1991, ISBN 0201548348) is finally becoming a reality. Michael Blechar, vice president and research director of application development tools and technology for Stamford, CT-based market research firm Gartner Group Inc., thinks so. "By 2003, at least 70% of the total number of new applications will be built primarily from 'building blocks' such as software components and application frameworks, increasing both products speed to market and enterprises ability to cope with change."

There's evidence to support Blechar's claim. Consider, for example, the emergence of component brokers such as ComponentSource, Marietta, GA, and Flashline.com Inc., Cleveland, OH. ComponentSource, which was founded in the United Kingdom in 1995, has built the world's largest repository (more than 2000) of open-market software components for all leading platforms. Although COM components currently represent the lion's share, ComponentSource has a growing inventory of Java components. Flashline, probably the second largest component vendor, offers some 400 discrete COM and Java components. It's interesting to see how the two companies categorize their offerings. ComponentSource categorizes components a variety of ways (see Table 1), not only by component type but by tool type, for example, applet, servlet, add-in or wizard; by source code language; by compatible containers, for example, Visual Basic 6, Delphi 5, or VisualAge C++; and by component function, for example, financial or file-handling components.

Flashline, launched in late 1998, has established four major categories: Java products, COM products, developer tools and beta beans. Java components are further categorized as EJBs, Internet/Web components, user interface (UI) components, information management components, network components, or training components. COM objects are simi-



larly categorized by function: UI, information management, Internet, or training.

Why should you be interested? Well, we're fundamentally talking about the age-old buy vs. build argument. In today's Internet time, the argument seems to be in favor of buying market-tested tools; and quality assurance is essentially a service offered by the component brokers. The brokers are attracting tool vendors you might not expect, such as EDS Corp., Plano, TX, a supplier of e-business management tools and information solutions. According to Gary Barnett, analyst with London, U.K.-based research and IT consulting company Ovum Inc., "This kind of initiative is exactly what is needed to boost the open market for components." Big Five consultancy PricewaterhouseCoopers LLP, New York, NY, meanwhile, predicts the open market for software components will be worth \$1 billion by 2002.

What Exactly are Components?

According to Neil Ward-Dutton, Ovum Principal Consultant and co-author of the 1998 report, *Componentware: Building it, Buying it, Selling it*, a software component is a unit of software that "implements some known function and hides the implementation of that function behind one or more unambiguous 'interfaces' that it exposes to its environment." In other words, components know something about themselves and can be interrogated.

Ward-Dutton thinks the component market can be usefully divided into only two major categories: technical services and business components. Technical services components are associated with tasks like

Table 1. Component Types According to ComponentSource

ActiveX (OCX)	VisualAge C++ Class Library
ActiveX Designer	Static Link Library
Dynamic Link Library (DLL)	Windows Foundation Class (WFC)
Visual Basic Extension (VBX)	COM Object/ActiveX DLL/In-Process Server
Visual Component Language (VCL)	COM Object/ActiveX EXE/Out-of-Process Server
Visual Basic Class Library	COM Add-in for Office 2000

Object Development

database connectivity, security, interprocess communications (IPC), or handling UI functions. Business components, on the other hand, typically handle middle-tier business logic and encapsulate business rules.

For Microsoft, components are basically COM objects. If today's COM objects are third-generation components (earlier generations were called Visual Basic Extensions, or VBXs, and ActiveX Controls), fourth-generation COM objects, which will be part of Microsoft's Next Generation Windows Services, are far more likely to be associated with Web services. For example, COM+, which is part of Microsoft's Windows DNA 2000 framework (<http://www.microsoft.com/dna>), doesn't refer so much to an object model or components as to Windows' role in delivering component services. Ovum's Ward-Dutton says the value proposition of components is morphing from one of providing services to individual enterprises to a broader one of providing services for Internet communities.

Sun, of course, sees components a bit differently. In the glossary on its <http://java.sun.com/developer> site, it defines the J2EE platform as an environment for developing and deploying enterprise applications, which consists of a set of services, APIs and protocols that provide the functionality for developing multitiered, Web-based applications. The J2EE platform defines four types of components: enterprise beans, Web components, applets and application clients.

An enterprise bean is a component that implements a business task or business entity, and can be either an entity bean or a session bean. An entity bean is an enterprise bean that represents persistent data maintained in a database and can either manage its own persistence or delegate this function to its container. Entity beans are always identified by a primary key. Session beans, as you might expect, usually exist only for the duration of a single client/server session. They're enterprise beans that have been created by a client to perform operations such as calculations or database access. They can be stateless or maintain state via their EJB container.

The second type of component in Sun's J2EE view of the

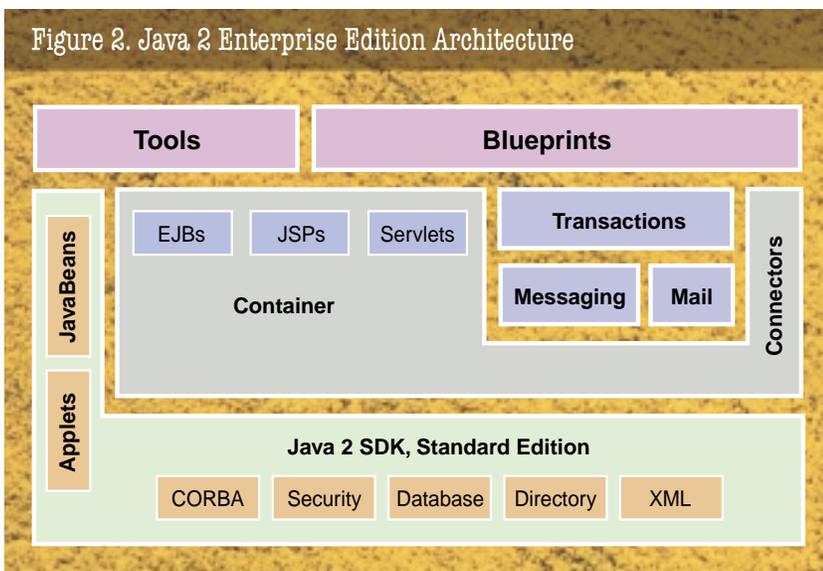
world includes Web components. These come in two flavors: servlets and JavaServer Pages (JSPs). A servlet is a Java program that extends the functionality of a Web server, generating dynamic content and interacting with Web clients using a request-response paradigm, while a JSP is a text-based document using fixed-template data and JSP elements that describe how to process a request to create a response. JSPs are essentially Sun's response to Microsoft's Active Server Pages (ASPs). The goal of JSPs is to make it easy for programmers to script Web sites using a combination of HTML, XML and calls to components. As of early 2000, Sun is working closely with the non-profit Apache Software Foundation (<http://www.apache.org>) to hasten the latter's support for JSPs and XML.

JSPs are essentially Sun's response to Microsoft's ASPs. The goal of JSPs is to make it easy for programmers to script Web sites using a combination of HTML, XML and calls to components.

The other two types of components from Sun's point of view are client components: applets, which are Java components that usually execute in a Web browser, and application clients, which are basically Java programs that execute in a Java Virtual Machine.

So much for a brief survey of the component landscape circa early 2000. As the new millennium dawns, you'll still find passionate debates about object models; see, for example, "EJB vs. COM+," a debate hosted by the Austin Foundation for Object-Oriented Technology (AFOOT) at <http://www.middleware-company.com/debate.html>. In this debate, Roger Sessions, founder of Object-Watch Inc., Austin, TX, and sometimes viewed as Microsoft's poster boy for COM and COM+ (he had been associated with CORBA), weighs in against Ed Roman, chief executive officer of The Middleware Co., Austin, TX. Both have written popular books promoting their respective preferences.

In the debate, Roman points out that "the new idea with components is that you declare the needs that you have on the middleware side of things by setting properties on your components and then the application server fulfills those properties. So, for example, with CORBA, you would use a transaction API to begin and commit a transaction. With EJB, instead of doing that, you can just set a property on your component that says, 'I always need a transaction when my component runs,' and the container will make sure this always happens. This saves you time, because you don't have to program to these APIs anymore." He also points out that there's a "big laun-



Source: Sun Microsystems Inc., Palo Alto, CA

Object Development

dry list” of things that we rely on middleware to do in distributed applications, and that it’s not the kind of stuff you want to write yourself. Although many companies have been “home brewing” this stuff in the past, what you really want to do is let the professionals handle this—professionals like IBM, Sun, BEA Systems Inc. and Oracle Corp. They ship products that handle all of this middleware plumbing for you. Their products are called application servers. Application servers give you these middleware services, allowing you to buy rather than build, Roman says. In EJB terms, an application server is called an EJB container or an EJB server.

Roman stresses that middleware is hard to write and that basic middleware services already exist in the form of application servers. What he doesn’t say is that many application servers support both COM objects and EJBs. In fact, many organizations don’t give—pardon the pun—a bean about which object model an application server uses. They want a product that solves a business task and provides services.



Is Software Dead?

Marc Benioff, chief executive officer of dot-com start-up Salesforce.com, San Francisco, CA, reportedly said at this year’s IDG DEMO 2000, held in February in Indian Wells, CA, that “software is dead.” Perhaps taking a cue from his erstwhile boss Larry Ellison, who two years earlier declared that “client/server is dead,” Benioff was saying that his company provides the service associated with sales force automation software. Forget about buy vs. build arguments. Forget about beans vs. COM objects. Salesforce.com will provide the service for you.

Whether you think of today’s hosted application services as a throwback to the days of timesharing or just another form of outsourcing, the point is they’re often a cost-effective alternative. According to Tony Wind, vice president of research and development and product management for Seagate Software, Scotts Valley, CA, a vendor of decision support and business intelligence software, including Crystal Reports, Crystal Info and Holos, Seagate is considering host-

Interview with David Gee

David Gee, newly hired vice president of marketing for Sun Microsystems Inc.’s Software Products and Platforms (SPP) division, granted *SW Expert* an interview from Cebit 2000, Hannover, Germany.

SWE: Will Sun ever become an enterprise software vendor—more than just a hardware vendor that sells management and operating systems software in order to drive hardware sales?

Gee: I joined Sun about three months ago from IBM in order to build Sun into a world-class software player. With Forté, StarOffice and NetBeans, we’ve made three major software acquisitions in recent months, and we’ll make more acquisitions as needed. I want SPP to provide an end-to-end story: from Solaris and all flavors of Java to iPlanet, StarOffice and Forté Fusion. We’re working on integration now, and we’ve already got an Early Access version of StarPortal available.

SWE: There was some initial confusion in the market about whether you were promoting iPlanet as a portal or portal builder. How are you going to integrate it into the mix?

Gee: iPlanet has its own branding. Although the aim is to bring all the software under the SPP umbrella, iPlanet, for various legal reasons, will have to remain an outlier for the next 12 to 18 months.

SWE: What are you doing to get your message out to the developer community better, like Microsoft does with MSDN and TechNet, and like IBM does with its developerWorks and alphaWorks? Even Oracle’s doing a pretty good job with OTN, the Oracle Technology Network.

Gee: I’m firmly committed to developers. I’m hiring some awesome team members who know how to reach out to developers and I’ve set MSDN and developerWorks up as benchmarks. Do you know the Java Developer Network has 1.7 million members? Well, I want to be a lot more in their face.

SWE: 1.7 million members is impressive, but what about the 7 million Visual Basic coders Microsoft claims, for example?

Gee: Look for our new ad campaign targeting “Uber coders.” Seriously, though, I’m going to take the message worldwide. Do you realize that JavaOne is already the single largest developer event in the world? And we had 25% more presentation submissions this year than last. I want to make it clear that Java is *the* platform. It’s the only one that you can count on to scale from servers to embedded devices.

SWE: What happened to the 100% Pure Java program? Are component vendors like ComponentSource making certification irrelevant?

Gee: No, we’re going to rejuvenate it. Certification will be increasingly important as platforms and form factors proliferate.

SWE: Sun has been quoted as saying JSPs are the way to go, as opposed to servlets. Is this correct?

Gee: Absolutely. We need JSPs to serve pages to the burgeoning mobile device market. We are working very aggressively with the Web and application server vendors to ensure the widest support for JSPs. That’s very high on my agenda right now.

Object Development

ing report servers for its customers. For “commodity” services, hosted applications seem to make economic sense. Of course, there are cultural, political and security issues associated with the decision to go with an application service provider, and each organization needs to work up its own costing model.

Enterprise Application Integrators

The fact is most organizations want their applications (and their components) to communicate intelligently and efficiently with one another. Today, enterprise application integrators (EAI) promise to help glue together your applications. They may add “wrappers” around legacy applications the way Seattle, WA-based host connectivity vendor WRQ Inc.’s new Apptrieve, or Bellevue, WA-based Attachmate Corp.’s eVantage products do. EAI generally provide custom middleware that does the application integration and maintains a repository. They may provide you with either enterprise “portals” or the tools to build your own. According to Framingham, MA-based research and consulting company Hurwitz Group, the major EAI include Active Software Inc., Santa Clara, CA; Neon Systems Inc., Sugar Land, TX; Software Technologies Corp., Monrovia, CA; Tibco Software Inc., Palo Alto, CA; and Vitria Technology Inc., Sunnyvale, CA. According to Bill Roth, product manager for Java enterprise products at Sun, “EAI is a symptom of a problem. It should ultimately be a feature of a larger product”—referring to the possibility of providing the glue via J2EE’s connectors.

Whether you think of today’s hosted application services as a throwback to the days of timesharing or just another form of outsourcing, the point is they’re often a cost-effective alternative.

Unfortunately, the application server/EAI market today is extremely crowded as old guard client/server and business intelligence vendors, along with newer Web server vendors, scurry to reposition themselves as your one-stop shopping source for distributed services. Even database vendors like Oracle, Redwood Shores, CA, and enterprise resource providers (ERPs) like SAP America Inc., Newtown Square, PA, are weighing in with ventures such as Oracle Business OnLine (<http://www.oracle.com/businessonline>) and MySAP.com (<http://www.mysap.com>).

Ovum’s Ward-Dutton estimates there are between 20 and 30 application servers, and lists BEA Systems, Sunnyvale, CA; Iona Technologies Inc., Waltham, MA; Inprise Corp., recently acquired by Corel Corp., Ottawa, Ontario; Progress Software Corp., Bedford, MA; IBM (WebSphere), Armonk, NY; and the Sun-Netscape Alliance (iPlanet), Mountain View, CA; as some of the top players.

Works in Progress

As mentioned earlier, the major competing object models from Sun and Microsoft are both works in progress. Microsoft needs to deliver SQL Server 2000, Commerce Server 2000 and Host Integration Server before the first stage of Windows DNA 2000 is really complete. And it's not expected to deliver the next version of Visual Studio, which will allow program-

BizTalk represents a major departure for Microsoft because it moves Microsoft from a technology focus to a Web services focus, albeit in a nonrevenue-generating mode.

mers to create next-generation COM+ Web components until sometime in 2001. (Visual Basic will also be gaining support for true inheritance in that release of Visual Studio.)

More interesting, perhaps, is how "open" Microsoft's Next Generation Web Services will really be. XML and the SOAP protocol seem to be poised to liberate Microsoft customers from the chains of COM. But one Microsoft independent software

vendor (ISV) who requests anonymity says, "The shift of Microsoft from proprietary technology to supporting a standard such as XML, even at the component level, is a bit curious—and makes me wonder what tricks they have up their sleeves."

He's not alone. Will customers essentially have to buy into Microsoft's version of XML and its BizTalk server in order to do e-commerce with COM-centric partners?

The idea seems to be that anyone is welcome to develop a set of XML schemas that conform to the BizTalk Framework. These schemas are then submitted to the BizTalk.org ("org" has such a nice nonproprietary sound, doesn't it) Web site (<http://www.biztalk.org>) for testing and validation, before being made publicly available. The BizTalk steering committee, which consists of key industry influencers, including the American Petroleum Institute, Baan Co., Boeing, Clarus Corp., Commerce One, Concur Technologies, the Data Interchange Standards Association (DISA), J.D. Edward & Co., Merrill Lynch, Microsoft, Open Applications Group (OAG), PeopleSoft Inc., RosettaNet and SAP, reviews and approves the final BizTalk Framework specification.

Then, individuals or organizations can use published, public XML schemas from BizTalk.org within their applications.

Companies Mentioned in this Article

Active Software Inc.

3333 Octavius Drive
Santa Clara, CA 95054
<http://www.activesoftware.com>
Circle 150

Attachmate Corp.

3617 131st Ave. S.E.
Bellevue, WA 98006
<http://www.attachmate.com>
Circle 151

BEA Systems Inc.

2315 N. First St.
San Jose, CA 95131
<http://www.beasys.com>
Circle 152

ComponentSource

2878 Johnson Ferry Road, Ste. 150
Marietta, GA 30062
<http://www.componentsource.com>
Circle 153

Corel Corp.

1600 Carling Ave.
Ottawa, Ontario
Canada K1Z 8BR
<http://www.corel.com>
Circle 154

Flashline.com Inc.

1300 E. 9th St., Ste. 1310
Cleveland, OH 44114
<http://www.flashline.com>
Circle 155

IBM Corp.

Contact local sales office
<http://www.ibm.com>

Iona Technologies Inc.

200 West St.
Waltham, MA 02451
<http://www.iona.com>
Circle 156

Microsoft Corp.

1 Microsoft Way
Redmond, WA 98052
<http://www.microsoft.com>
Circle 157

Neon Systems Inc.

14100 Southwest Freeway
Ste. 500
Sugar Land, TX 77478
<http://www.neonsys.com>
Circle 158

ObjectWatch Inc.

11414 Pencewood Drive
Austin, TX 78750
<http://www.objectwatch.com>
Circle 159

Oracle Corp.

500 Oracle Pkwy.
Redwood Shores, CA 94065
<http://www.oracle.com>
Circle 160

Progress Software Corp.

14 Oak Park
Bedford, MA 01730
<http://www.progress.com>
Circle 161

Salesforce.com

101 Spear St., Ste. 203
San Francisco, CA 94105
<http://www.salesforce.com>
Circle 162

SAP America Inc.

3999 West Chester Pike
Newtown Square, PA 19073
<http://www.sap.com>
Circle 163

Seagate Software

920 Disc Drive
Scotts Valley, CA 95067
<http://www.seagatesoftware.com>
Circle 164

Software Technologies Corp.

404 E. Huntington Drive
Monrovia, CA 91016
<http://www.stc.com>
Circle 165

Sun Microsystems Inc.

901 San Antonio Road
Palo Alto, CA 94303
<http://www.sun.com>
Circle 166

Sun-Netscape Alliance

501 E. Middlefield Road
Mountain View, CA 94043
<http://www.iplanet.com>
Circle 167

The Middleware Company

12405 Alameda Trace Circle, Ste. 1137
Austin, TX 78727
<http://www.middleware-company.com>
Circle 168

Tibco Software

3165 Porter Drive
Palo Alto, CA 94304
<http://www.tibco.com>
Circle 169

Vitria Technology Inc.

945 Stewart Drive
Sunnyvale, CA 94086
<http://www.vitria.com>
Circle 170

WRQ Inc.

1500 Dexter Ave. N.
Seattle, WA 98109
<http://www.wrq.com>
Circle 171

Object Development

Businesses will also have the option of publishing the schemas on a secure Web site for private use between trading partners. The goal is to get a set of common XML schemas that are tuned to promote the most popular types of e-commerce and business-to-business (B2B) transactions. The Hurwitz Group sees BizTalk as representing a major departure for Microsoft because it moves Microsoft from a technology focus to a Web services focus, albeit in a nonrevenue-generating mode (at this time), and that BizTalk sets the stage for a grander vision that could put Microsoft at the center of an XML e-commerce hub.

Sun needs to beef up the current release of J2EE and EJB 1.1 with EJB 2.0 in order to provide better support for persistence and messaging (partners couldn't agree on a reference implementation for the Java Message Service). And, as far as the J2EE platform goes, some developers are beginning to speak of code bloat, a criticism generally reserved for Microsoft. J2EE consists of nine technologies: EJB 1.1, CORBA, JSPs, the Java Servlet API, the Java Naming and Directory Interface (JNDI), Java Database Connectivity (JDBC), the Java Message Service, Java Mail and the evolving Java XML. It's probably still not as big and complicated as CORBA or Distributed Computing Environment (DCE), but some members of the Java community seem to be beginning to wonder if J2EE isn't beginning to look like a product designed



by committee. Sun's Roth says we can expect a maintenance release for J2EE this spring and EJB 2.0 sometime in 2001.

But whether you're a shop that prefers to build your own components—or at least, some of them—or not, the good news is that distributed computing really is getting easier. XML and the Microsoft-drafted SOAP, a protocol based on XML that promises to handle remote procedure calls across the Internet in a nonproprietary fashion, seem destined to provide the glue that makes distributed computing a “given”—the way object-oriented programming and components are today.

No, we haven't solved all the problems associated with distributed computing. There are still legacy applications (and proprietary repositories) to integrate. We're still developing standards to integrate mobile and wireless devices into our new distributing computing models. Developers are getting more adept at creating modular components. We just need better tools and servers that let us mix and match them. ✍

Karen Watterson is an independent San Diego, CA-based consultant who specializes in database and data warehouse design. She's editor of industry newsletters on Visual Basic and SQL Server and has just completed a book on SQL Server, *10 Projects You Can Do with Microsoft SQL Server*. Email: karen_watterson@email.msn.com.