

# Object Middleware Starts to Take Shape

Cooperation and interoperability make CORBA, OLE/COM and OpenDoc viable, but the adoption cycle is just beginning.

By Philip J. Gill

**N**ot long ago it appeared that object middleware would turn into yet another battleground for competing vendor camps intent on pushing their own technologies and agendas as "industry standards." The contenders were the Object Management Group (OMG), with its Common Object Request Broker Architecture (CORBA) specification; Microsoft, with its Windows-only Object Linking and Embedding (OLE) desktop object component architecture; and OpenDoc, a multiplatform client technology pitched as an OLE alternative and sponsored through Component Integration Laboratories (CI Labs), a multivendor consortium.

Today, rather than an unqualified victory for any one side, there's peace of sorts in the object middleware marketplace, brought about by the convergence of all three parties on a simple truth: No single object component model can address all the needs of all users, so all three must coexist and interoperate.

That is the direction in which CORBA, OpenDoc and OLE are headed today, though each object component model stands at a different stage of completion, availability and market relevance. CORBA and OLE have clear dominance on the Unix server and Windows client, respectively, but OpenDoc's relevance isn't so certain—it has no dominant niche from which to build a broad base of support, and some observers question its viability. (For more, see "OpenDoc: Dead or Alive?")

It is important to remember, however, that although the different camps have declared peace and are working toward seamless coexistence, object componentry, by and large, is not a settled technology. "This stuff is still more in the future than the present," says Steve McClure, director of object technologies research at International Data Corp. (IDC) in Framingham, MA. "We're at the begin-

ning of a long adoption cycle. There has been limited success thus far."

CORBA is the most "real" of the three technologies, but there are only a handful of working CORBA-based user installations. And those who have working systems are reluctant to talk, believing the technology is giving them a competitive advantage.

McClure qualifies even those successes. He says the few users who have been successful at building and deploying significant CORBA-based commercial business applications have done so using a single vendor's object middleware technology. That is a significant achievement but far short of the original goal—interoperable, multivendor object component technology—that the industry set out to achieve.

## Serving Up the ORB

Object middleware is concentrated into two camps. On the traditional open systems side is the OMG, a vendor-neutral standards organization based in Framingham, MA, whose membership includes all the leading hardware and software companies, such as Digital Equipment Corp., Hewlett-Packard, IBM, Sun Microsystems and many others. Its contribution to object middleware, the CORBA specification, is the basis for a dozen or so object request broker (ORB) products.

Although CORBA-compliant ORBs can be implemented on clients, CORBA is essentially a server-oriented object component technology that has roots in the Unix operating system and C and C++ programming languages. To complete the client/server paradigm, OMG recently agreed to adopt the OpenDoc client application programming interfaces (APIs).

Chris Stone, president and CEO of OMG, says the adoption of the OpenDoc APIs provides a complete middleware environment for developing fully interoperable object components. "OpenDoc can wrap OLE Automation when you build parts," Stone says. "Those now become distributed via CORBA. Additionally, the OLE/COM-to-CORBA spec ensures OLE-only applications a bridge."

Despite best efforts, early commercial versions of the CORBA-compliant ORB suffered the fate of other industry standards, such as structured query lan-

guage (SQL). That is, everyone agreed to adopt the standard specification, then set about building their own, often incompatible implementations. Such incompatibilities eventually lead to the formation of the SQL Access Group (SAG) to devise methods for interoperability.

IDC's McClure says OMG has cleaned up its act with later versions of the CORBA specification and is working to make sure that different vendors' implementations can interoperate with each other. That, after all, is why the industry, through the auspices of the OMG, created this technology in the first place.

Annrai O'Toole, president of Iona Technologies, a Dublin, Ireland-based leader in ORB implementations, admits there have been interoperability problems between different vendors' ORBs, but he insists that Iona and other vendors have addressed those issues. "That should be behind us by the end of the year," he says.

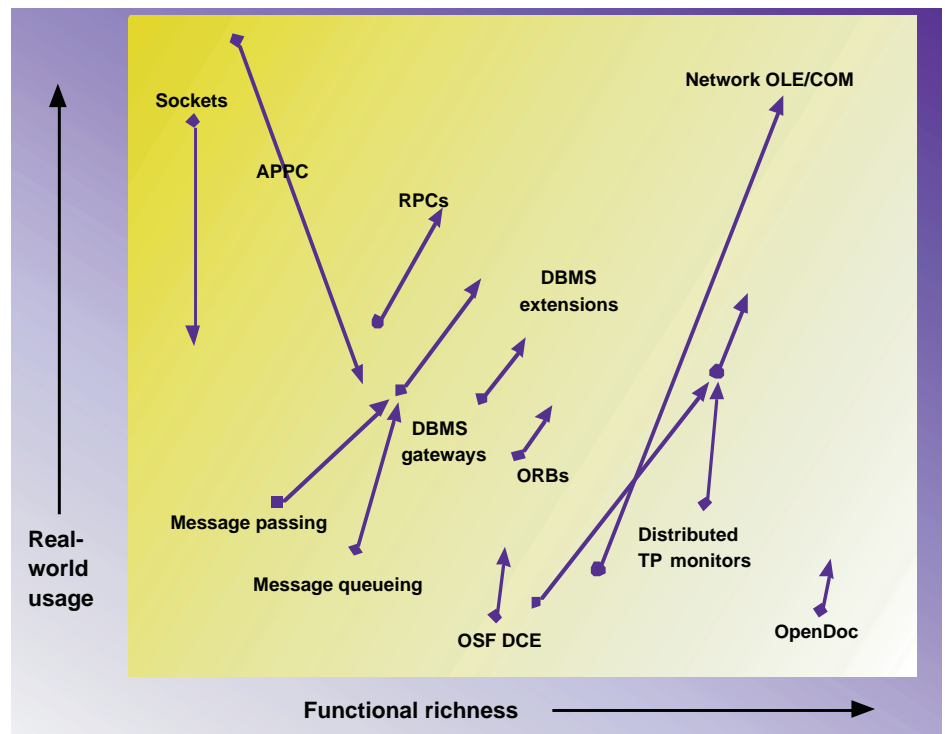
OMG's Stone and others argue that many of the incompatibility issues in the CORBA 1.0 specification have been addressed in version 2.0, although there remains no formal certification process, such as the X/Open testing and branding of products that conform to an agreed-upon Unix specification.

### Moving CORBA Ahead

Stone says the OMG is forging ahead, particularly in regard to making its CORBA specification relevant to today's hottest programming environment, the Internet. "CORBA 2 is complete, and vendors are beginning to ship implementations using IIOP [Internet Interface Object Protocol]," he says. IIOP permits objects to interoperate over the Internet.

In addition, adds Stone, "We are working on revisions to include changes that make the CORBA IIOP more acceptable to the IETF [Internet Engineering Task Force] for adoption. We will be adding asynchronous messaging, three-byte language support, Java mapping, Cobol mapping and a secure protocol."

Iona's O'Toole adds that CORBA's backers are looking beyond full, two-way OLE-to-CORBA interoperability, in two new directions. At the high end, OMG and ORB vendors are working to integrate the technology with message queuing systems, including transaction processing (TP) monitors and the Open Software Foundation's Distributed Com-



Gartner Group predicts strong demand for OLE/COM in the next three years, although OpenDoc has greater functionality. Source: Gartner Group

puting Environment (OSF DCE).

At the low end, like everyone else, CORBA proponents are looking at interfacing with Java applets. (Java, of course, is an object-based programming language from Sun Microsystems that was designed specifically for the Internet and the World Wide Web). Some such connectivity measure is needed. Alexis de Planque, senior analyst with the advanced information management (AIM) service of the Meta Group in Westport, CT, points out that Java applets have no capability to talk to one another. "ORBs will enable Java applets to communicate with one another," she says.

Even in the unlikely event that Microsoft were to shut out CORBA technology from its Windows NT server stronghold, it will still have an important role to play in facilitating communication between Java applets, de Planque says, adding that nothing that Microsoft has proposed thus far could perform that function.

### Extending OLE's Reach

On the other side of the object middleware controversy are Microsoft Corp. and its primary ally, Digital, which has feet firmly in both camps. Besides being an aggressive reseller of Microsoft's NT operating system, DEC has developed the

Component Object Model (COM) to help bridge OLE into the CORBA world. Microsoft has also agreed to cooperate with OMG to ensure interoperability, though the degree of its cooperation is not clearly defined.

The original OLE holds limited appeal to the heterogeneous, distributed client/server network environment common in the enterprise today. Microsoft's dominance in desktop operating systems and applications makes OLE a sort of standard, but OLE is a Windows-only, desktop-only technology. Furthermore, it was developed for linking components within different Windows-based applications, not for linking client components and server components across a departmental local-area network (LAN) or enterprise wide-area network (WAN). Another limit is that original OLE is essentially a compound document technology and does not address enterprise-class applications, such as transaction processing and database activities.

Microsoft intends to extend OLE's reach beyond the desktop, however, into a full-scale enterprise computing object middleware architecture through a new version of the technology. Originally called Network OLE, it now goes by the name DCOM, perhaps a reflection of its close reliance on the COM technology

## OpenDoc: Dead or Alive?

**All three object component architectures are real. Are they all relevant?**

**C**ORBA occupies the high end, in several senses. For one thing, even though there are few working applications, CORBA now dominates the Unix server market for object middleware and will for some time to come. On the desktop, the popularity of Microsoft's operating systems and applications has made its OLE technology a de facto standard. Who, then, needs OpenDoc?

Observers and analysts think the answer is not many people, despite the efforts of its original backers—a consortium including Apple Computer, IBM, Lotus Development and Novell—and superior technology. "OpenDoc is an option at the desktop that is probably more functional, but ISVs will probably be more interested in OLE," says Steve McClure of IDC, because of Microsoft's dominance on the desktop.

Alexis de Planque of Meta Group is more blunt. "OpenDoc is dead," she says.

Frank Mara, vice president of sales and marketing for CI Labs, the Sunnyvale, CA-based organization that is developing and licensing OpenDoc source code to the industry, believes that its technology is misunderstood by analysts and the press. "OpenDoc and CORBA together are a client/server model," he says. "OLE/COM is not."

CI Labs has some points in its favor. For one thing, despite the problems Apple and other original backers have had, the consortium today has over 300 members around the world. It also has more than one million lines of source code for the various OpenDoc components, which anyone can purchase for \$350. OLE continues to be a Windows-only technology, but OpenDoc supports multiple desktop platforms. Finally, going with

OpenDoc doesn't exclude OLE; the technology interoperates with OLE using the OLE Automation mechanism.

"For a distributed component model to win, it has to be platform-independent," Mara says. "One company can't control how this evolves. You have to give licensees room to innovate and add value."

OMG's recent endorsement also adds credibility. Chris Stone, OMG president, contends OpenDoc is "very relevant. People want to build distributed components, either intranet- or Internet-based, that they can register across multiple platforms."

OMG is adopting the OpenDoc APIs to insure CORBA interoperability with the desktop but also (and perhaps more importantly) to ensure access to distributed services, such as security, naming, events and others OpenDoc provides. "This is distributed computing with components," says Stone.

Those arguments, analysts say, don't reflect the reality of the object middleware marketplace today. For instance, the Meta Group's AIM service provides IT advisory services to large user organizations. As part of that service, it regularly polls its customers and other user organizations on their object middleware plans. "We don't see any of our end-user customers talking about using OpenDoc," says de Planque.

At the least, CI Labs faces a serious selling job, which it is trying to do by making the public aware of its Distributed Component Software Architecture (DCSA), an overall model that includes OpenDoc and CORBA. It is intended, among other things, to enable management of Internet applications by object middleware. "OLE/COM without distributed functionality is a nonstarter on the Net," says Mara. "Java is instant technology, not a whole solution for distributed computing. DCSA does deliver it."

DEC developed. Most observers refer to the technology generically as OLE/COM. Regardless of the name, Network OLE or OLE/COM includes several components: OLE DB, for database activities; OLE Transaction, for enterprise-class transaction processing; and OLE DS, for shared network directory services.

The new features and new name should help, but Microsoft also realizes that to push OLE/COM in the enterprise, it will have to proliferate the technology across multiple platforms, not just the various iterations of Windows. Therefore, it has recruited Software AG of Reston, VA, to port Network OLE/COM to more than two dozen platforms. "Our deal is to take the Network OLE source codes and the so-called Cairo NT reference model, and port that to 19 other operating systems and hardware combinations," says Dave MacSwain, Software AG's vice president of marketing and technology.

The database software and development tools company will port OLE/COM to Caldera Corp.'s Linux, Data General Corp.'s DG/UX, Digital Unix, HP-UX, IBM's AIX and Sun's Solaris, as well as IBM's MVS and OS/400, and DEC's OpenVMS platforms, plus others.

Under the terms of its agreement with Microsoft, Software AG won't be able to deliver any of these non-Microsoft OLE/COM ports until after Microsoft ships them on the Cairo (version 4.0) release of Windows NT. That version is supposed to ship before year's end, but given Microsoft's historic tardiness in delivering major operating system releases, Cairo may not arrive until 1997.

### A Multiprotocol Future

CORBA and OLE can interoperate, and further refinements of both technologies are under way. Yet that begs a larger question. How will users absorb and implement the technology?

Max Dolgicer is a director of International Systems Group, a New York-based consulting and custom software development firm that specializes in enterprise middleware technologies. Dolgicer advises companies that can't wait for the OLE/COM technology to become available across multiple platforms, and who have in-house or can purchase the expertise needed to implement CORBA-based applications, to go with CORBA.

According to Dolgicer, a few users are successfully building and deploying CORBA-based applications that support OLE interoperability, using products such as Iona's Orbix. These user organizations tend to be what he and the Gartner Group of Stamford, CT, classify as "Type A"—technical innovators willing to take risks at the leading edge of computer technology. These users also tend to operate in select industries, such as telecommunications and financial services, and have large IS staffs with in-house expertise in object technology, C, C++ and Unix (and generally large budgets). "Those are about 20 percent of the companies out there," he estimates.

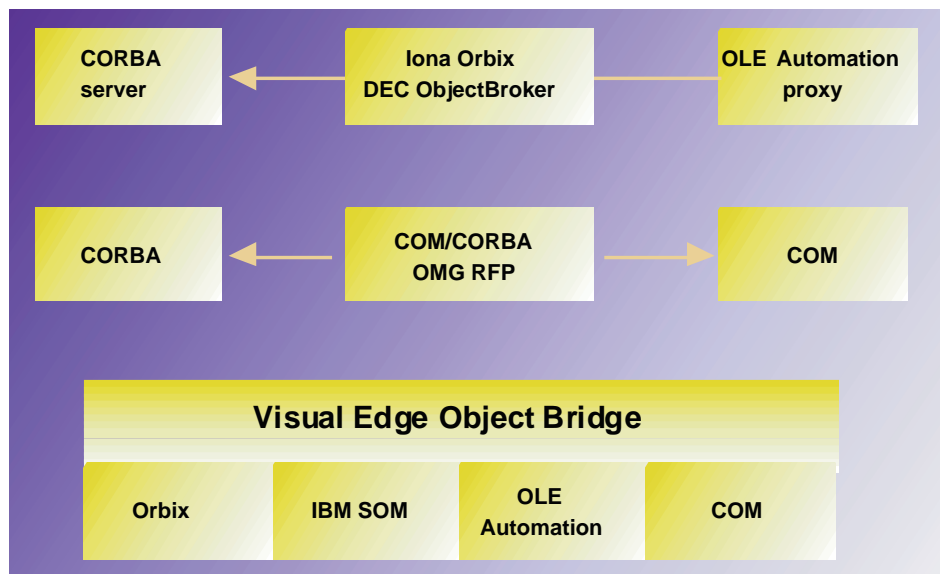
Dolgicer says that OLE-to-CORBA interoperability is important today and will be more so in the future. Products such as Iona's Orbix allow programs built with Microsoft's Visual Basic on the client side to invoke Orbix objects on the server side. However, as of today such interoperability is only one way (from OLE/COM to CORBA). Other CORBA vendors are planning to support such interoperability as well.

The end goal, however, is truly seamless, two-way interoperability between OLE/COM and CORBA. "Today, there's only one product that provides that: Visual Edge's Object Bridge," says Dolgicer, who adds that many participants in this market, including many CORBA vendors, will have to provide what are called "object adapters" for their products to comply with the Visual Edge specifications.

### Following the Trend

Dolgicer says he believes the other two user sectors, Types B and C, are likely to wait for OLE/COM, if they use object middleware at all. Type B users constitute the IT mainstream, occupying neither the bleeding edge nor lagging behind, while Type C are the followers or laggards. He says both groups will probably be drawn to a single object model, in the form of OLE/COM, which will be used directly or indirectly (most Windows users already use OLE/COM). This might happen for a number of reasons.

First, he says, these users are less likely to have the in-house expertise in C, C++ and object technologies that Type A organizations have. Second, they probably



CORBA and OLE/COM work at various system levels and, within limits, interoperate with each other. Source: Software AG

don't have the budgets to buy that expertise from outside.

Third, the typical Windows programming environment, be it Visual Basic or Sybase's PowerBuilder, are easier to use and more pervasive. Both are already enabled for OLE/COM. Developers can create objects using OLE controls (OCXes), for instance. Fourth, CORBA products and development efforts are expensive, while OLE/COM will ship in every box of operating system and application software Microsoft sells.

Fifth, Microsoft may be weak in servers right now, but it is making a push with its NT Advanced Server (NTAS) platform and gaining ground against Unix. Finally, for mainstream corporations CORBA applications are complicated projects, Dolgicer claims. "Other middleware technologies, such as message queuing and TP monitors, are about three years old, and there are literally hundreds of TopEnd, Tuxedo, Encina and DCE applications running in production today. That's a sure sign that CORBA applications are not easy to build."

### Necessary Trade-Offs

Although Software AG has partnered with Microsoft, MacSwain insists that both CORBA and OLE/COM will continue to coexist and interoperate. Users, he says, will have to choose between a single object model, such as OLE/COM, and a gateway approach that permits OLE/COM and CORBA to interoperate. Both

approaches have pluses and minuses.

For instance, the single object model of OLE/COM everywhere provides a consistent development and management environment, easy integration and assured compatibility. "The degree of difficulty in a gateway approach is higher," says MacSwain, "but there are some benefits in heterogeneity. You have more flexibility in the products and packages you might want to buy."

He compares it to database purchase decisions many large companies frequently make today. "You might want to buy an application for the mainframe that's DB2-based and doesn't work with Oracle, and you might want to front-end it with an application that's Oracle-based and doesn't work with DB2," he says. "You usually bring in databases and network protocols, and presumably object brokers in the future, because you're buying packages of tools for specific problems and those packages and tools drag other technologies with them."

In sheer numbers, OLE/COM may dominate. However, most people who follow the IT industry and the way it evolves expect both OLE/COM and CORBA to be around for a long time to come. In adopting object component technologies, users again will be faced with a juggling act. **IT**

*Philip J. Gill is a free-lance writer and editor based in San Diego. He can be reached at philipgill@aol.com.*