Author: Angela Hey at INPUT
Date: 11/9/95 3:26 PM
Priority: Normal
TO: Mail List - #Sales
CC: Mail List - #Research
Subject: Table Of Contents for Component Software Report
---------- Message Contents ----------

Attached is a flyer with table of contents for the forthcoming Component Software: ORBs, OLE and OpenDoc report. It is stored in F:\COMMON\SALES\TOCS\CLT4TOC.DOC

Component Software Battles: ORBs, OLE and OpenDoc

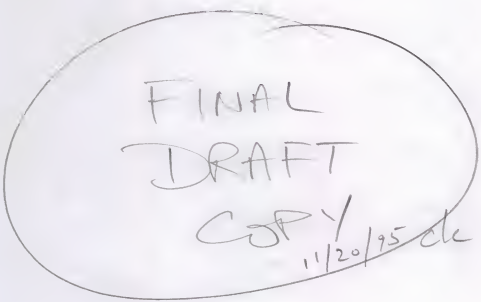
This report describes two software architecture battles. The first is for component integration, between Microsoft's OLE (Object Linking and Embedding) and Component Integration Labs' OpenDoc standard. The second is for component interoperability between Microsoft's COM and the Object Management Group's CORBA standards. Surrounding these standards are complex alliances of vendors. The strategies, strengths and weaknesses of 24 vendors are analyzed. The report positions major players, like IBM, Sun and Microsoft, as well emerging technology vendors like Iona Technologies, PostModern Computing and Expersoft.

Component Software Battles: ORBs, OLE and OpenDoc provides market forecasts to help decision makers understand the impact of technologies. It also discusses trends, issues and future technology directions.

Table Of Contents

NOVEMBER 1995

Component Software Battles: ORBs, OLE and OpenDoc



INPUT®

Frankfurt • London • New York • Paris • San Francisco • Tokyo • Washington D.C.

??THIS PAGE SHOULD BE REPLACED
WITH ABOUT INPUT PAGE??

Published by
INPUT
1881 Landings Drive
Mountain View, CA 94043-0848
United States of America

Client/Server Software Program

Component Software Battles: ORBs, OLE and OpenDoc

CLT4 • 670 • 1995

Abstract

This report describes two software architecture battles. The first is for component integration, between Microsoft's OLE (Object Linking and Embedding) and Component Integration Labs' OpenDoc standard. The second is for component interoperability between Microsoft's COM and the Object Management Group's CORBA standards. Surrounding these standards are complex alliances of vendors. The strategies, strengths and weaknesses of 24 vendors are analyzed. The report positions major players, like IBM, Sun and Microsoft, as well emerging technology vendors like Iona Technologies, PostModern Computing and Expersoft.

Component Software Battles: ORBs, OLE and OpenDoc provides market forecasts to help decision makers understand the impact of these technologies. It also discusses trends, issues and future technology directions.

The report contains 98 pages and 13 exhibits.

(Blank)

1

Introduction

This chapter describes the purpose and scope of this report and lists related reports published by INPUT.

A
Purpose of the Report

Two software architecture battles are raging in distributed object computing. The first battle is over component integration. Component integration architectures determine how software building blocks are assembled to make documents, games and applications. The second battle is over component interoperability. Component interoperability architectures determine how objects interact with each other across heterogeneous networks.

The component integration battle is being waged between Microsoft's OLE (Object Linking and Embedding) architecture and Component Integration (CI) Labs' OpenDoc architecture. The component interoperability battle is being fought between Microsoft's COM (Component Object Model) and the Object Management Group's CORBA (Common Object Request Broker Architecture) and related standards. This report shows that these two battles are intertwined and that winning the component integration battle is the key to owning the network.

Component Software Battles: ORBs, OLE and OpenDoc analyzes the battlefield and its players. It anticipates major changes in the client/server software market over the next five years based on:

- accelerating investment in component software
- increasing maturity of application development tools and run-time software needed to build and manage those components

This report gives enterprise system planners, component software marketing management, and investors insights into market trends, issues, and emerging products. For systems integrators, systems suppliers, and independent software vendors, the report describes potential partners. For all audiences, it views two intersecting aspects of component software from both a technical and a business perspective. The two aspects are:

- multivendor component software interoperability
- management of distributed components

Rather than discussing abstract computer science terminology to obtain these perspectives, this report analyzes the outlook, strategy, strengths and weaknesses of 24 systems and software vendors. The report illustrates by example how components interact when used in diverse systems.

In addition, this report examines major efforts to facilitate the process of assembling components into meaningful applications. These efforts are being undertaken by many individual software vendors, including Microsoft and Oracle, as well as an organization called the ComponentWare Consortium (CWC) and various special-interest task forces within the Object Management Group (OMG).

B
Scope

The report answers the questions:

- What are the main categories of software components?
- What is an object request broker (ORB) and what is its relationship to component software?
- What is a compound document architecture (CDA) and what is its relationship to component software?
- Who are the leading vendors of component software and tools to build and integrate such software?
- What are these suppliers' strengths and weaknesses?
- How is the component software market evolving?

- How big is the market for software components?

This report focuses primarily on independent software vendors, but also includes system suppliers that are developing component software technology. It provides both U.S. and worldwide forecasts.

The report does not cover the use of component software for graphical user interfaces, operating environments or databases. For the latter two areas, see INPUT's report, *Object-oriented Platforms for Client/Server Computing*. It also does not cover message-oriented middleware or transaction monitors, which are covered in an earlier INPUT report, *Middleware: Is DCE The Answer?*.

Sun's Java language and Internet browsers represent opportunities for Microsoft's competitors to seize control of the desktop. This report only briefly touches on these technologies that may threaten some of the compound document approaches of OpenDoc and OLE, particularly in publishing, on-line information services and information retrieval applications.

The COM and CORBA releases that are discussed in this report are COM 2 and CORBA 2. The current release numbers are not usually specified in the text to save repetition.

The forecasts cover spending by programmers on components that they can integrate into applications. They do not cover spending by users on compiled components which are expected to be a significant opportunity, but currently are difficult to quantify. The forecasts also predict expenditures on development tools, software packages and enterprise software.

C
Methodology

The research relies on interviews with vendors and software demonstrations. Reviews of published material from vendors, on-line networks and case studies were also used to compile this report. Twenty-four vendors are reviewed in detail. To create the market forecasts, the revenues and product strategies of over 600 software vendors were analyzed, using software catalogs, vendor publications and on-line databases.

D

Related Reports

Caps Caps

Related reports in INPUT's Client/Server Software Program include:

- *Middleware: Is DCE the Answer?*
- *Object-Oriented Platforms for Client/Server Systems*
- *U.S. Client/Server Market Analysis, 1993-1998*
- *Client/Server Service Opportunities — Europe, 1993-1998*
- *Client/Server Impact on Major Project Contracting—Europe, 1993-1998*
- *Client/Server Trends in the Federal IT Market: 1994*
- *Client/Server Applications Trends—Banking and Finance*
- *Client/Server Applications Trends—Insurance*
- *Client/Server Applications Trends—Discrete Manufacturing*
- *Client/Server Applications Trends—Process Manufacturing*
- *Client/Server Applications Trends—Health Services*
- *Client/Server Applications Trends—Telecommunications*
- *Client/Server Applications Trends—State and Local Government*
- *Client/Server Applications Trends—Retail Trade*
- *Client/Server Applications Trends—Utilities*

In addition, INPUT reviews vendor strategies in its Vendor Analysis Program and in its Client/Server Vendor Profiles.

Client/Server Systems Management Software

Internet Security: The Impact of Firewalls on Client/Server Applications

3 MOST RECENT STUDIES MISSING FROM ORIGINAL AS SUBMITTED. (CAUGHT BY MARK D. AFTER BOOK WAS PRINTED — CORRECTED PAGE HAD TO BE HAND INSERTED.

Client/Server Explosion - How Users Choose Platforms

D
Related Reports

Related reports in INPUT's Client/Server Software Program include:

- *Middleware: Is DCE the Answer?*
- *Object-oriented Platforms for Client/Server Systems*
- *U.S. Client/Server Market Analysis, 1993-1998*
- *Client/Server Service opportunities — Europe, 1993-1998*
- *Client/Server Impact on Major Project Contracting—Europe, 1993-1998*
- *Client/Server Trends in the Federal IT Market: 1994*
- *Client/Server Applications Trends—Banking and Finance*
- *Client/Server Applications Trends—Insurance*
- *Client/Server Applications Trends—Discrete Manufacturing*
- *Client/Server Applications Trends—Process Manufacturing*
- *Client/Server Applications Trends—Health Services*
- *Client/Server Applications Trends—Telecommunications*
- *Client/Server Applications Trends—State and Local Government*
- *Client/Server Applications Trends—Retail Trade*
- *Client/Server Applications Trends—Utilities*

In addition, INPUT reviews vendor strategies in its Vendor Analysis Program and in its Client/Server Vendor Profiles.

II

Executive Overview

This chapter summarizes the key market and technology trends related to component software. It contrasts the fairly distinct polarity of approaches between systems suppliers and independent software vendors in the battle for the network.

A
Issues and Trends

Software components are the foundation of modern computing applications and systems. This report discusses how software components are integrated and how they interoperate. In both of these areas a battle rages. At its simplest, it is Microsoft and its allies versus major computer industry players. Microsoft is working from the desktop to conquer the network; server vendors like HP, Sun and IBM are trying to dominate the network in the hope that they can then influence the architecture for assembling software components on all platforms, including the desktop.

Component integration can be achieved using Microsoft's OLE technology or Component Integration Labs' OpenDoc architecture, which also includes support for OLE. Component interoperability comes from using Microsoft's COM or the Object Management Group's CORBA standards. Another competitor to OLE and OpenDoc is the Internet, where Sun's Java technology may prevail for creating compound documents from networked elements.

Trends and key issues related to component software are summarized in Exhibit II-1.

Exhibit II-1

Component Software Issues and Trends

Issue	Explanation	Trend
Slow standards efforts inhibit market growth	Component software efforts depend on widespread acceptance of a standard.	Few software vendors or corporate IS departments are willing to make major investments in component software until standards emerge. INPUT expects this to happen in the 1996-1998 timeframe.
Users fear getting locked into a vendor	Microsoft is the obvious candidate here, although others like IBM, Sun or HP may emerge.	Microsoft has not committed to OpenDoc. Developers cannot proceed with confidence that OLE objects will be able to be managed by any underlying OS. Microsoft threatens to lock developers into its own Windows environments despite its stated intention to run OLE across other platforms.
Pure object environments do not exist	Most companies cannot neglect legacy code and start with easily integrated object frameworks.	Many corporations will adopt hybrid strategies. By wrapping legacy code in new software it can be made to appear a component of a modern object-oriented architecture. Some companies will find it easier to develop new applications that are object-oriented.
Software components do not interoperate easily	Each development environment supports its own interoperability mechanisms.	Slowly maturing standards efforts are aimed at making objects readily interchangeable on a plug-and-play basis. OpenDoc promises this type of plug-and-play support, but software based on it is not yet available.
Programming is moving to higher levels on the network	Distributed object environments mask the programmer from underlying network complexity.	With component software, distinctions among underlying chip architectures, operating systems, local- and wide-area networking standards and interoperability protocols, etc. can be eliminated.
Systems suppliers lose control of their infrastructure to Microsoft	Losing the infrastructure means that resellers, systems integrators, software developers and ultimately users desert a platform.	Systems suppliers realize that if they do not maintain control of the infrastructure, their products will become even lower margin commodities than they have already become over the last 10 years. Microsoft is equally aware of the infrastructure's importance and points to distributed object management as the next area of growth. Microsoft controls the desktop, but foundation objects for enterprise systems are still an open battleground and are likely to continue to be so for the next five years.
Corporate power structures are threatened.	Component software issues affect people. Modular software will favor small, agile development teams.	IS departments and the systems they control are likely to be broken up. IS department power is threatened. Information utilities may become available to commercial users as well as consumers.
Rapid development, code re-use and deployment	Systems are being built in days or weeks, instead of months, using object-oriented tools.	Managing reuse is critical to success. Large objects are candidates for reuse; smaller objects can be recoded in less time than the time.

Source: INPUT

In summary, component software enables a new class of applications to emerge with dynamic user interfaces, complex network interactions and unprecedented access to information. Standards are crucial to the widespread acceptance of component software and the battles are likely to be set over the next few years. The emergence of a strong software component market threatens the infrastructure of IS departments and fosters the emergence of information utilities, akin to power or telephone companies, that can sell and support software components.

B
Market Forecasts

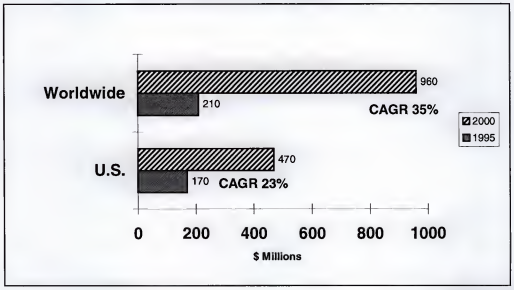
In this report, INPUT has divided the forecasts for software that uses OLE, OpenDoc or CORBA standards into three areas:

- components
- application packages
- enterprise software

These forecasts are for software licenses and upgrades only; they exclude services. This chapter summarizes how much programmers spend on software components that can be used to create applications. The other forecasts appear in Chapter V. Exhibit II-2 shows the spending by programmers on software components.

Exhibit II-2

Spending by Programmers on Software Components Worldwide and U.S. 1995-2000



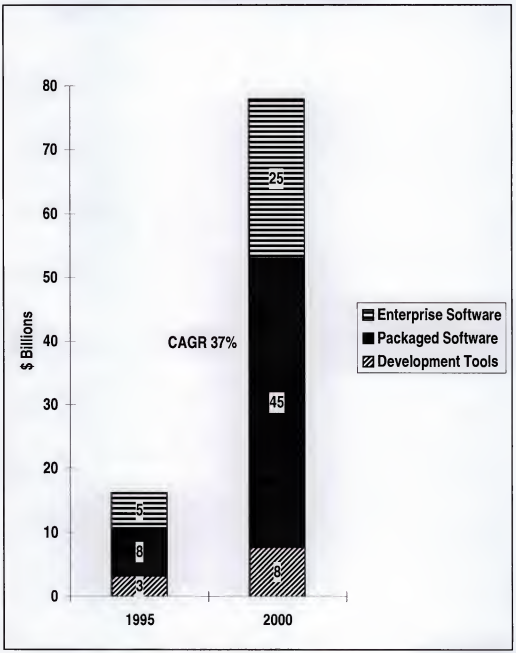
Source: INPUT

In the U.S. market, expenditures by programmers for software components are expected to grow from $170 million to $470 million at a 23% (CAGR) from 1995 to 2000. The worldwide market will grow faster—at 35%—from $210 million in 1995 to $960 million in 2000. Worldwide, users will spend $16 billion in 1995 on software that uses or can interface to some component software standard. This market

will grow at 37% to $78 billion in the year 2000 as shown in Exhibit II-3.

Exhibit II-3

Spending by Users on Software that Supports Components Worldwide and U.S., 1995-2000



Source: INPUT

The current market consists of:

- development tools—for newer OLE features OpenDoc and distributed enterprise systems based on CORBA
- packaged software—where Microsoft's OLE is dominant
- enterprise software—where CORBA and OLE interfaces are already incorporated

The forecasts include software that supports earlier OLE 1 and CORBA 1 standards. The forecasts show that there is already considerable use of components in leading-edge systems and packages. Systems integrators and major corporations with large IS infrastructures, in particular, have already made substantial investments in component software.

C
Vendor Summary

Chapter IV includes the detailed vendor analysis on which this section is based. It analyzes each of the suppliers listed here from two perspectives:

- Business-plan dependence on acceptance of component integration standards, OpenDoc or OLE
- Technical bias toward component interoperability, CORBA or COM

Delays in setting standards and the inability to mix and match the technologies mean that corporate IS departments and software vendors are often reluctant to commit to component software except in limited pilot projects. This process has frozen the market to Microsoft's benefit, providing Microsoft time to deploy its complete offering. Microsoft rules the desktop. The open question is whether it can execute its announced plans to control the distributed object platform (see the Microsoft analysis in Chapter IV) and the network as well. In short, how widely will OLE be distributed on non-Microsoft platforms?

Other vendors' positions relative to this market freeze vary. Application development tool suppliers tend to support a range of standards. They design their technology to selectively manage components or broker object requests at run-time and adapt to whatever standards are in place. For example, ParcPlace-Digitalk (particularly the former ParcPlace organization) typifies most object-

oriented language suppliers, showing little bias to any distributed-object-management standard and little dependence on the acceptance of a standard.

Object-oriented database management system (OODBMS) suppliers follow suit. Although Object Database Management Group (ODMG) members, such as Objectivity and Object Design, actively promote their CORBA compliance, they more aggressively promote the ODMG-93 extension to CORBA (thereby going beyond compliance with standards). RDBMS vendors—typified by Oracle—have indicated they that will incorporate object technology into their relational products rather than introduce revolutionary new (and self-competing) product sets.

The systems suppliers that have already begun to market ORBs — Digital, IBM and ICL—are less dependent on standards acceptance because their robust products can be moved to the tool/component-supplier category easily if CORBA is not widely accepted. AT&T GIS has already repositioned COOPERATION, one of the original ORB products, as an application development framework.

HP's and Sun's prospects are less certain because their distributed object management products are still in flux. Little is left of their joint 1990-1992 Distributed Application Architecture (DAA) research project into this technology—except the IDL, the heart of the CORBA standard itself. Sun may encroach on the OpenDoc market with Java in applications that run over internal intranets externally over the Internet. Indeed, HP's failure to support DAA with application development tools was one of the key reasons that Microsoft was able to encroach on its market with OLE.

Apple is in a special position because of its dependence on OpenDoc's success. Even Apple is hedging its bets with OpenDoc's OLE interoperability software. Apple's Claris division is one of the few organizations that could make OpenDoc a real success. Indeed, the MacOS is one of the first non-Microsoft operating environments that supports OLE-enabled applications from Microsoft. Apple and the rest of the systems suppliers are primarily interested in selling platforms and that overriding objective will ultimately determine all of their strategies.

NeXT is the wild card of the component software industry. It has quietly and consistently built a track record that indicates that it knows what it is doing with the technology—and knows what the user

community wants. It has accomplished this without getting immersed in industry politics. The technology leader in object-oriented operating environments on the market today is NeXT's OpenStep. HP was an early reseller of NEXTSTEP, OpenStep's predecessor, on its RISC workstations. Sun is working with NeXT to develop OpenStep to run on Solaris. OpenStep for Solaris is already shipping, and with Sun's NEO strategy, the UNIX platform has a viable alternative to OLE and OpenDoc for some distributed computing applications. A more important strategy for Sun is Java.

Finally, three independent ORB suppliers are analyzed—PostModern, Iona and Expersoft. All are dependent on acceptance of CORBA standards; that is what they sell.

D
Recommendations

1. Recommendations for Enterprise IS Development Groups

- Foster object-technology pilot tests and invest in component libraries.
- Recognize that development cycles will change with object-oriented systems. They will quickly provide benefit, but may need much fine tuning to make them scaleable and industrial strength. Don't think a system is nearly finished because it demos well.
- Allow adequate resources for systems integration and interoperability testing after the initial system is deployed.
- Prepare to decide on enterprise-wide standards, primarily between OpenDoc and OLE; alternatives are OpenStep, Java and Smalltalk environments.
- In RFPs, consider specifying standards, performance criteria and inter-enterprise interfaces for distributed object technology.
- Encourage development of standard components for users like spreadsheet templates, clip art, and reports.
- Encourage users to submit their own scripts and programs to a group that can deploy them to others in the organization to accelerate business processes.

- Organize a user-driven consortium, an Underwriters' Lab for objects. It may balance the vendor-dominated "standards" consortia. (Note: since this went to press, INPUT has learned of OAG, a user group for objects).

2. Recommendations for Applications Development Tool and Other Systems Software Vendors

- Become more involved in the standards development process.
- Create cross-platform tools, but don't be afraid to add innovative features for a specific environment—for example, an OLE component for a Windows environment that integrates with a common Windows application.
- Make families of software tools that can be used by both professional programmers and users.
- Make components that users can integrate with their applications with minimal help from IS departments.
- Encourage third parties to provide services around your products, such as custom programming, training and integration.
- Don't neglect the installation process for components; this may be trickier than anticipated.
- Don't be tempted to underprice components; giving them away in the expectation of future revenues from more fully featured software may set customer expectations of unrealistically low prices. It is better announce a price for components and give them away for a limited period if a promotion is required to sell the product.
- Add one or two leading-edge features for each technology to make the components attractive to the market.
- Widely distribute and promote components. If components are useful, then they need to be widely distributed quickly to establish a market presence.
- Make decisions based on Microsoft owning the desktop. Assume that it will have significant market share in LAN networks. Enterprise networks still require multiple standards to be supported.

3. Recommendations for Systems Integrators and Applications Software Vendors

- Already, major systems integrators like Andersen, EDS and SHL Systemhouse are building significant systems using objects. They are already building inventories of objects that may be reused. Smaller systems integrators, like BSG, are already starting to invest in object-oriented tools like those from Forté. Smaller systems integrators also need to invest in component software that serves their areas of applications expertise in order to compete.

- Applications software vendors need to commit to standards now in order to have product in the 1997 timeframe. They are choosing OLE COM on the basis that they will at least have a market on the desktop. Applications software vendors that can afford to wait are doing so.

4. Opportunities for New Services

- Testing labs—a service to certify the interoperability and manageability of foreign objects is needed by both corporate IS groups and software vendors

- On-line component software delivery—companies like CompuServe enable software to be downloaded. With better user interfaces and improved billing systems, there are opportunities for new on-line vendors of specialized components.

- Software maintenance—updating and managing component software across a network will be a significant business. Already, outsourcing vendors with networks have the infrastructure to manage and maintain systems. They are in a strong position to leverage their network support and move into the component software maintenance market.

- Public software repositories—while enterprises are building software repositories for internal reuse and management of code, there is also an opportunity for public software repositories that can be used to store software components, deliver them as needed and manage standards.

- Information utilities—the major on-line service providers, like America On-line, as well as telephone companies and software vendors like Microsoft will compete to become seamlessly integrated vendors of software and other IT services over networks. This is an area where well-financed enterprises like media publishers can also compete with established outsourcing vendors like EDS. Startups focused solely on being information utilities may also emerge, but they will need strong partners and access to significant resources, of capital and expertise.

- On-line software component stores for programmers—the forecasts show that, compared with the packaged software market, this is not such a large opportunity. Whereas there are many programmers, on-line services vendors that expect to generate large revenues from selling to programmers may be disappointed unless they have a special niche. They must provide additional value to their customers, such as technical support.

E
Conclusions

Network-based component software can provide quantum improvements in enterprise-wide software maintenance and re-usability. It will dominate the future of software services and applications development. Network-based component software will have to be implemented using object technology because current procedural methods and tools cannot support the complexity of the operating system and network management needed to control the infrastructure.

The U.S. market for component software is ahead of the world market in maturity, in part because the business climate encourages small entrepreneurial firms that can build components. Software development is still a strength of American industry and there is a strong home market for the cottage industry of component creation.

The OpenDoc standard promises much, but unlike OLE 2, it generally lacks the commitment of major developers, systems integrators and user organizations. OpenDoc's success will depend on the few major developers it can attract such as IBM and Apple's Claris division. The OMG has many more corporate participants than does CI Labs. It is most likely to be used for innovative Macintosh-based applications initially. OLE 2 is firmly established and can only continue to propel

Microsoft into networked applications, particularly those suitable for workgroups.

Object technology requires either dejure or defacto standards development and acceptance to achieve the promised benefits. The systems suppliers and their allies have already delivered functional interoperable object brokering software, i.e., ORBs—although it probably will not interoperate seamlessly until 1997. They now need to concentrate on delivering a standard compound document architecture solution.

Microsoft has not yet delivered an interoperable object brokering technology, and probably will not do so until 1997. It has a functional compound document architecture today based on OLE with millions of users. Microsoft controls desktop object management on its own operating systems and is dedicated—at top corporate levels—to making its approach to network-based object management a standard as well. The rest of the industry will wait to determine if Microsoft can execute its plan, almost freezing widespread use of distributable component software.

The systems suppliers' strategy to counteract Microsoft's plans for dominance of the network are in the same slow standards process as the overall client/server standards process (e.g., UNIX, CDE (Common Desktop Environment), etc.). This is reminiscent of the UNIX versus Windows wars, where Motif standards were hotly promoted by UNIX leaders against Sun's NEWS for the desktop, yet Windows grabbed the lion's share of the market.

For less complex applications of compound documents, like information retrieval and document publishing, Internet technologies will compete with OpenDoc and OLE, albeit at a higher level of integration. OpenDoc is most likely to succeed in the OS/2 and MacOS environments. However, the tools are very immature and the software components it produces are at a low level. IBM and Apple are the two most likely companies to make OpenDoc succeed, but it faces considerable competition from OLE and, possibly, from Sun and NeXT with standards based around OpenStep. The lack of a strong systems infrastructure around OpenDoc makes it unlikely to be a major component software standard for the next five years. After a five year struggle, if IBM and Apple market aggressively and if components can be created at higher levels for users to customize their software, it may have greater market acceptance.

Blank.

III

Component Software Directions

This chapter describes software components available or under development to support enterprises, independent software vendors, and systems software suppliers. It also describes how distributed components will be managed using ORBs, including CORBA-compliant ORBs and Microsoft COM. The chapter illustrates how these technologies differ. Finally, it discusses the place of object-oriented compound documents.

A
Component Software Segmentation

1. What Is Component Software?

Many software engineers believe that the age of large monolithic software packages is over. These packages are hard to maintain, inflexible and difficult to integrate with existing systems. Packaged software applications are increasingly being built from components. A market in software components will be created via the use of object-based technology. At issue is how big will this market be and who will control it.

"Component" refers to a piece of software with documented interfaces that a programmer can use to build applications. The terms "component software" and "objects" are used in this report interchangeably for convenience, although this is not technically precise. Technically, a component may include an object and the methods (actions) that operate on it, but to differentiate between objects with and without related methods would be tedious and add nothing to the discussion of trends and future directions.

The basic goal of object technology is to take advantage of the conceptual orientation of merging action and information to foster

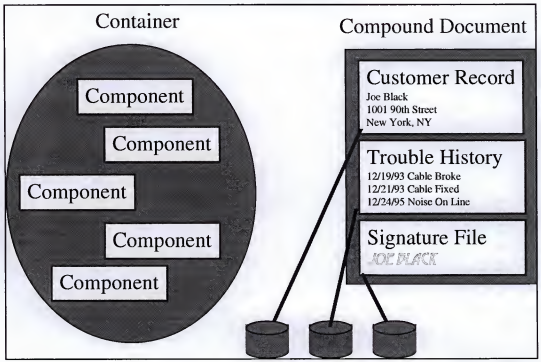
reuse and minimize maintenance. Data and actions performed on the data can be encapsulated in an object. This is diametrically opposed to the conventional procedural approach to development , which separates action and information. Current procedural software development methods and dependencies cannot support the complexity of leading-edge component software environments.

Corporate computing systems and packaged software applications will increasingly be built from components. For agility, code reuse and design simplification, corporations are designing modular systems. Components range from Smalltalk objects to Microsoft Excel spreadsheets to software that represents entire business processes. Systems are becoming so complex that they cannot be redesigned from scratch. They are increasingly being upgraded, extended, maintained and modified with software components.

The technologies described here attack the problem of component integration, depicted in Exhibit III-1, and component interoperability, depicted in Exhibit III-2.

Exhibit III-1

Component Integration



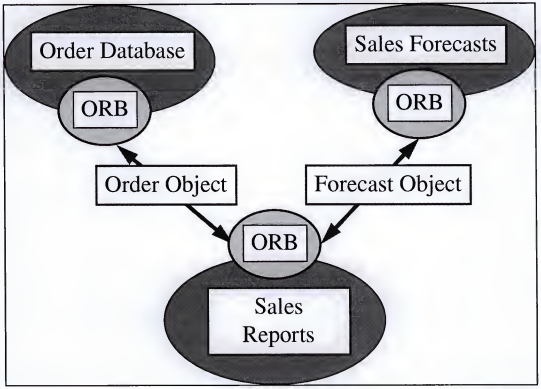
Source: INPUT

The left side of Exhibit III-1 shows an abstract entity, a container, that integrates the components. On the right, side it is translated into

a compound document that contains three components: a customer record, a trouble history and a signature, all pulled from separate databases. OLE and OpenDoc are concerned with making not just documents, but also applications, games and multimedia scenes from components. There are also embedded components that can be used in PBXs and printers to enhance their functionality.

Exhibit III-2

Component Interoperability



Source: INPUT

Exhibit III-2 shows a networked system with objects moving between applications. COM and CORBA standards deal with how objects and software components interoperate between applications. In the above example, a sales report may be created by combining data from an order application and a sales forecasting application. Interoperability is achieved via ORBs that may support OLE and/or CORBA standards. Data warehousing and information retrieval are just some of the applications that can benefit from object-oriented technology, particularly when the network is complex.

2. Software Component Segments by Technology, User and Provider

The main users of software components today are programmers in user organizations, independent software vendors (ISVs), systems vendors, peripheral vendors and systems integrators. From a technical perspective, component software includes:

- Fine-grained components used by programmers. Examples are sliders, window frames and scrolling list boxes.
- Content components that consist of text, data, image, sound and other possible files comprising electronic documents. These are usually created by users after the software has been deployed. They may also include images, sound and video clips to make an application more fun to use.
- Application components that are larger than the fine grained components. They can usually be added to a package after it has been deployed and include drivers for printers, video displays, communications networks and other peripherals. They may also include gateways, middleware and application frameworks that support enterprise applications.
- System components—entire application packages or databases, especially parts of legacy systems. Legacy application fragments may be wrapped in object-oriented code to make them compatible with newer architectures.

Exhibit III-3 shows these component types in terms of characteristics, users and providers.

Users and developers may buy an initial package with basic features or they may purchase the entire product as components. To obtain components that either enhance the original package or comprise the entire product, development teams—and even individual users—will select software from network-based, CD-ROM or similar-media-based libraries, information utilities or information factories. From the library, they will assemble applications as needed to meet specific organizational or personal productivity needs.

Exhibit III-3

Component Software by Technical Characteristics, Users and Providers

Types	Description	Users	Providers
Fine grained components	Small pieces of an application, such as a scroll bar, menu or list	ISVs, corporate IS departments	Systems software suppliers (including systems suppliers); all types of ISVs
Content components	Portions of electronic documents, such as spreadsheets, word processor files, templates of any type	Individuals, systems integrators	Horizontal and vertical ISVs, corporate IS departments
Application components	Everything from .DLLs and C++ code to entire run-time subsystems such as a transaction manager or messaging backbone. Also includes apples such as Microsoft Graph.	ISVs and corporate developers	Middleware, application development framework and language-subsystem ISVs
System components	Applications such as a word processor, browser, etc. and legacy code	ISVs, corporate developers, systems integrators	Systems suppliers, technically knowledgeable users and corporate IS departments

Source: INPUT

Application, tool and utility ISVs will do the same; the resulting ISV product will itself be a component, further marketed through the same libraries, and likely to become a component of a larger whole. Software vendors will also supply components to these libraries as well as build their own products from components, liberally buying from or trading with other component software suppliers. As an example, consider the latest version of Microsoft's Word for Windows95. Add-on utilities, such as those used to create HTML documents for the Internet, can be downloaded from Microsoft's Internet Web page or shipped to the purchaser for a nominal fee on a floppy disk.

3. Software Component Segments by Application

Component software can be classified by application segment, as shown in Exhibit III-4:

- Services, such as object-request brokering, naming and security
- Common-facility components, including the document-architecture functions of automation, control and document management described later in this chapter

- Productivity tools, including run-time components of language products, for both developers (e.g., application development and design tools) and users (e.g., word processors)

- Cross-industry application components, such as financial-management or engineering software

- Vertical application components specific to particular industries, such as health services, insurance, telecommunications, banking and finance, etc.

Service components are needed to manage systems. As software standards like OLE, OpenDoc and CORBA, mature users need directories, security mechanisms and licensing services. These services enable PC networks to replace enterprise computers for some applications. No longer will lack of support tools for smaller client/server applications be an issue if third party software vendors develop products based on service components.

The common components can be shared by applications, and include spelling checkers and the foundations of messaging and printing. Scheduling and time management are areas where new component-based groupware applications are emerging. Productivity tools are for both programmers and users—software vendors do not differentiate between a word processor that can be programmed and scripted to create an application and an application development tool. One effect of component software is to turn everyone into a programmer. This has already happened with spreadsheets, where many users write scripts. Now publishers and writers have the opportunity to program significant applications to improve their productivity.

Cross-industry and vertical components are particularly attractive for systems integrators, as they enable integrators to customize applications more rapidly and improve their profit margins. Specialization in industry processes will be key to success in vertical markets. In cross-industry markets, vendors will compete on the ability to integrate with existing systems, price/performance and ease of maintenance.

Exhibit III-4 relates these application categories to terminology used by the OMG and Microsoft to describe their components in these areas.

Exhibit III-4

Component Software By Application

Application of Component Software	OMG Terminology	Microsoft Terminology
Service components	Object Services -Directory -Security -Authentication -Licensing -Transaction -Query -Externalization	OLE Controls - Property Pages - Property Change Notification - Events OLE Enterprise Development Interfaces Network OLE
Common components	Common Facilities -User I/F -Systems Manager -Workflow automation -Document Architecture -Speller -Mailer -Time Manager	OLE Drag and Drop OLE Document Management - Linking - Embedding - Inplace Activation OLE Automation OLE Networking OLE Custom controls (.OCX)
Productivity tools	Application Objects - Word Processor - Document Processor - Spreadsheet - CASE tools	
Cross-industry application components	Horizontal User Facilities (e.g., application development facilities, financial management components)	
Vertical application components	Vertical User Facilities (e.g., mapping, retail, computer-integrated manufacturing, oil/gas, telecommunications)	

Source: INPUT

4. The Role of Standards In Component Software Market Growth

Theoretically, component software development means that:

- software components can be built in an information factory
- standard software components can plug and play like physical components
- developers will have the luxury of developing applications from an enterprise, rather than a technical, perspective
- users can purchase applications that are independent of hardware, networks and operating systems

A significant problem with the above scenario today is the lack of accepted standards according to which these components should interoperate and how they should be managed. Developers need to know how:

- separately developed components will interoperate
- components operating on disparate platforms will interoperate

Object technology being the key to the component software market makes the object standards-setting process of more than just academic interest. Corporate standards are being defined by enterprise developers storing objects in repositories, but these are often hard to access. It is difficult to make developers aware of what has already been programmed. It is even harder to train developers to reuse software components. Standards are also being created by developers of software libraries and frameworks. Unfortunately, some of the best libraries—from companies like NeXT—do not have widespread portability and acceptance.

As with dynamos and generators used in the power-utility industry, component software should not have to be built by the same supplier, or work in the same way. As long as nothing in a component's design or assembly causes other components to malfunction or damages the network, the suppliers of component software (including operating systems software suppliers) are free to design and build their components in any way that achieves market penetration and dominance.

Protection against power-utility equipment causing malfunctions, damage or harm is achieved via standards monitored by inspection organizations. In the fairly mature power industry, those standards are defined by the American National Standards Institute (ANSI) in the U.S. ANSI works with similar national level groups around the world under the umbrella of the International Standards Organization (ISO). Adherence to the standards is measured in various ways by Underwriters' Laboratory (UL) in the U.S. and similar organizations around the world.

The less mature information systems industry also works with ANSI and ISO. But ANSI and ISO activity in IS simply tends to ratify standards achieved via supplier agreements (dejure) or market acceptance (defacto). Two such standards battles are under way in the area of component software. The first is an attempt to standardize how various pieces of component software interoperate. The current contenders are:

- OMG and its Common Object Request Broker Architecture (CORBA) specification

- Microsoft and its Component Object Model (COM), an integral part of Microsoft's better known Object Linking and Embedding (OLE) architecture

A second standardization activity is an attempt to specify how interoperating component software is managed. The two main distributed object management architectures being proposed for market acceptance are:

- Component Integration Labs' (CI Labs) OpenDoc, which includes technology for Document Services, Component (Management) Services, Automation Services and Interoperability Services, all of which operate with a CORBA-compliant ORB called the System Object Model (SOM)

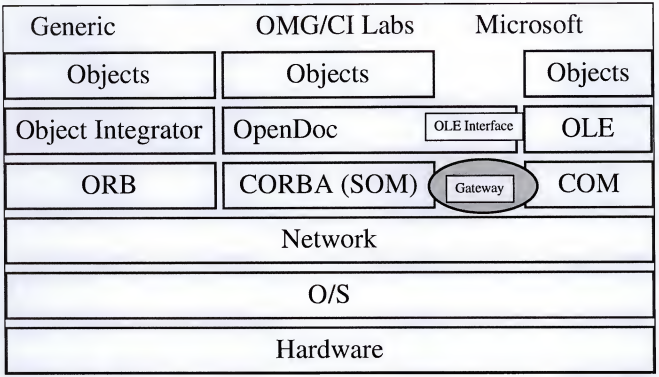
- Microsoft's OLE Control, OLE Automation, and OLE Documentation, all of which operate with COM

Note that the distributed object management architectures are also termed Compound Document Architectures (CDA), a term that both CI Labs and Microsoft contend is misleading because more than traditional document-dependent applications can be built to their specifications.

Exhibit III-5 illustrates the interrelationship of these four proposed standards. It shows a layered architecture, with a generic model at the left, the OMG position in the center and Microsoft's position to the right. In each position, the lowest level represents the operating system (O/S) and a communications layer, such as Novell's NetWare (COMM). Above this in the generic model is an Object Request Broker (ORB) managed by an object manager, and above this are objects that may be incorporated in diverse applications. The OMG model has CORBA (the Common Object Request Broker Architecture) as its standard with CI Labs' OpenDoc as the object manager. IBM has contributed its SOM (System Object Model) as a standard for this environment. The Microsoft model assumes that some version of Windows is the operating system (Windows 3.1, 95 or NT and follow-on products). Its OLE (Object Linking and Embedding) technology is the foundation for linking documents and other application components and its COM is the object interoperability architecture.

Exhibit III-5

ORB, OLE and OpenDoc Interrelationships



Source: INPUT

5. The Effect Of Delayed Software Standards

A major disjunction in the component software market is that CORBA products exist, but OpenDoc objects are still two years away; similarly, first-generation OLE objects exist but COM is not yet available.

As Exhibit III-5 shows, the OpenDoc architecture depends on CORBA and the three related OLE functions depend on COM. These mutually exclusive dependencies frame the major issue delaying the growth of the component software industry. If the two could simply square off in the marketplace and let developers decide the issue, the use of component software would be more likely to grow rapidly. Unfortunately, there are two a major obstacles:

- CORBA, the interoperability scheme underlying OpenDoc, exists. IS and ISV developers are actually paying for and using the technology today. COM, on the other hand, is only in beta test for distributed object interoperability and is too immature to use in major development efforts.

- As component integration architecture OLE, the CDA design operating on top of COM, exists. IS and ISV developers are actually paying for and using the technology today. OpenDoc, on the other hand, is only in beta test and is too immature to use in major development efforts.

Neither approach can be totally deployed and tested. Therefore, the market cannot make a decision until 1996 or beyond. As an aside, even when both approaches are totally available, no independent inspection facilities—such as UL in the power industry—exist in the IT market. This vacuum provides an opportunity for a company or an organization funded by IT users. This is not unlike the open systems operating systems wars where UNIX vendors formed OSF to market X-Windows. This was an attempt to challenge Microsoft Windows.

B
ORBs

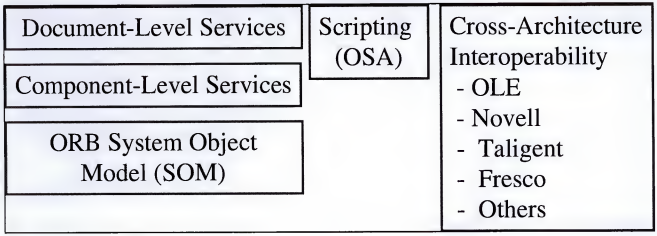
1. Description

"Object Request Broker" (ORB) is accepted industry terminology for the communications interface between objects. An ORB provides the mechanism by which objects transparently make requests of, and receive responses from, other objects. Simplistically, ORBs are to distributed object environments what PBXs are to phone networks. Just as a PBX helps users connect to phone networks, an ORB helps objects connect to applications.

In a complex heterogeneous environment, such as that found in networked systems management and trading applications, ORB interoperability simplifies system architecture and maintenance. Exhibits III-6 and III-7 illustrate the two most commonly known ORB architectures—CORBA and COM. The illustrations use an abstraction of the OpenDoc and OLE architectures, respectively, to illustrate the placement of the ORBs in distributed object management.

Exhibit III-6

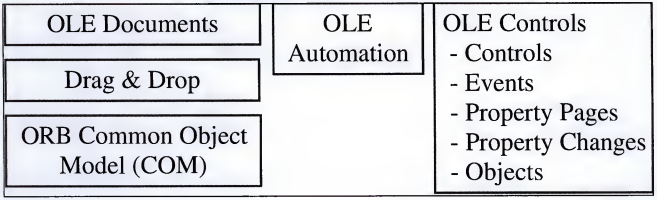
Typical CORBA-oriented Architecture



Source: INPUT with OMG and Component Integration Labs

Exhibit III-7

Typical COM-oriented Architecture



Source: INPUT with Microsoft

2. Pre-CORBA ORBs

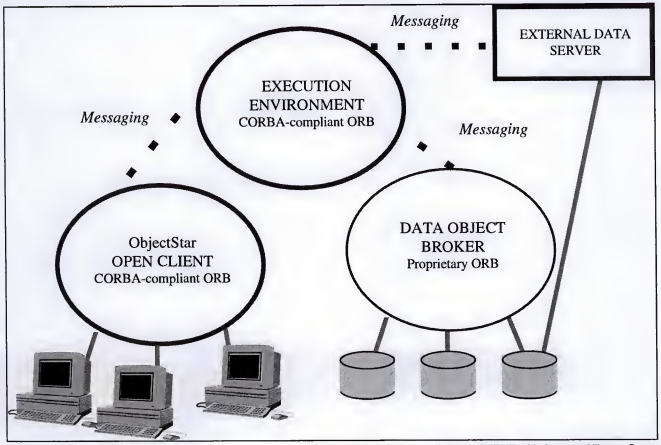
There was wide-ranging object technology activity prior to the proposal of CORBA standards. For example, Amdahl developed an object environment called Huron in the late 1980s. Huron is now marketed by Antares Alliance Group (a joint venture between Amdahl and EDS—see Chapter IV). Huron includes its own proprietary object brokering capability.

Exhibit III-8 illustrates the Antares product, now called ObjectStar, to show the relationship of the proprietary ORB and how the product now incorporates a gateway to systems using another ORB via a

CORBA-compliant ORB. Other organizations that which have their own ORB include NeXT and the European Community's Portable Common Tool Environment (PCTE).

Exhibit III-8

Example of Proprietary and CORBA-compliant ORB Integration, Antares Huron ObjectStar



Source: INPUT with Antares Alliance Group

3. Differences in ORB Standards

Exhibit III-9 provides some of the differences in ORB standards, comparing the COM model with the CORBA model.

Exhibit III-9

Comparison of ORB Standards

Issues	CORBA	COM
Standards development status	By design; only as precise as need be to ensure portability and interoperability; as a dejure specification, many feel it is too slow to change (but the process is much faster than true international dejure standards, such as those maintained by ISO or CCITT)	Rigorous specification, but as a defacto, Microsoft-driven specification, it is always subject to change
Object orientation	Uses a full object-oriented design at the standards level, but that does not necessarily mean that all standards-compliant ORBs incorporate object technology	By design, COM does not support inheritance or polymorphism because Microsoft believes unrestrained inheritance for binary code is bad practice
Platforms supported	UNIX, Windows, OS/2, MVS, OpenVMS, Macintosh and others	Windows and Macintosh; UNIX support was planned as part of a joint development effort with Digital, but that effort has been abandoned
Underlying communications protocol	TCP/IP base required to be termed CORBA 2-compliant; however, SPX/IPX and DCE/RPC extensions are likely	DCE-like RPC, said by Microsoft to be functionally equivalent to OSF's DCE standards; COM cannot generate any other wire format
Distributed nature	Has supported homogenous distributed object interoperability (i.e., same brand ORB on multiple platforms) for over three years and promises widely supported heterogeneous interoperability in 1996	Local RPC mechanism only at this time; whether COM can expand its functionality to the infrastructure is often questioned, but experts on both sides of the issue say there is no technical reason that it cannot
Object definition	Incorporates IDL-based definition and dynamic invocation methods; no pointers	Proprietary IDL with pointers to implementations
Bridges and gateways planned	RFP for an OLE gateway standard in process; some OLE interoperability product is already shipping in anticipation of the standard	Digital plans a CORBA 2 gateway to COM
Status of generally available product	Heterogeneous distributed object interoperability promised for 1996	Homogeneous distributed object interoperability promised for 1996, based on Microsoft platforms

Source: INPUT

4. Vendor Commitments

Competing ORBs include:

- Products from systems suppliers who are already marketing CORBA 1-compliant ORBs—such as IBM (SOM), ICL (DAIS), Digital (ObjectBroker)—or who are promising CORBA 2-compliant products—such as Sun (NEO) and HP (Orb-Plus) in the near term
- Initial CORBA 2-compliant products from software vendors including IONA (Orbix), PostModern (ORBeline), Expersoft (PowerBroker) and Teknekron Software Systems (ObjectBus); most of these suppliers also marketed a CORBA 1-compliant ORB

- Object-based products incorporating nonstandard ORB or ORB-like technology, including NeXT (PDO), Antares Alliance (ObjectStar), Microsoft (current-generation OLE) and others (most of these suppliers have promised incorporation of both emerging standards). NeXT has already announced CORBA 2 compliance.
- COM from Microsoft, currently in beta test and promised for general availability in 1996

5. Issues

Until now, distributed component software has been developed assuming one or more of the following:

- CORBA orientation
- COM orientation
- a proprietary ORB
- an ORB-neutral position

Linking systems together using these different orientations potentially presents problems. Gateways and bridges that enable interoperability between the above architectural approaches will be used to overcome this incompatibility. If these mechanisms prove to cause performance problems, which is likely because bandwidth is still a major wide-area network concern, the market may choose only one of the above approaches as a defacto standard. Meanwhile there will be a market window of opportunity for gateway providers. Long term, gateway technology will be absorbed into leading products.

CORBA achieves its heterogeneous interoperability through the use of gateways. The standard requires compliant ORBs to communicate directly via TCP/IP, as a minimum. Major CORBA supporters—Digital and HP—prefer to interoperate via a DCE-based RPC mechanism that is independent of the underlying network protocol, and are likely to promote that method heavily. This technicality divides the CORBA community at a time when it can least afford to confuse the development community, given that Microsoft is planning its push to establish OLE COM as the standard for early in 1996.

OLE COM also uses an RPC-like mechanism, functionally equivalent to DCE. This is Microsoft's strategy for interacting with enterprise suppliers like Digital, IBM and HP that use DCE. It has the

advantage of being network independent, but the downside is that there may be performance problems. This may become an issue in standards acceptance. Alternatives based on transaction monitors and message-oriented middleware are likely to evolve.

6. Future Directions

The battle for the network will commence, based on ORB standard acceptance, in 1996. The sides have drawn clear distinctions between their approaches. The systems suppliers appear to have a better technical approach, but they are somewhat divided over implementation. Microsoft has market power and a clear focus.

Most of these functions will be incorporated into future operating systems, and many ISVs and corporate IS departments are likely to wait until standards are widely disseminated.

C
OLE And OpenDoc

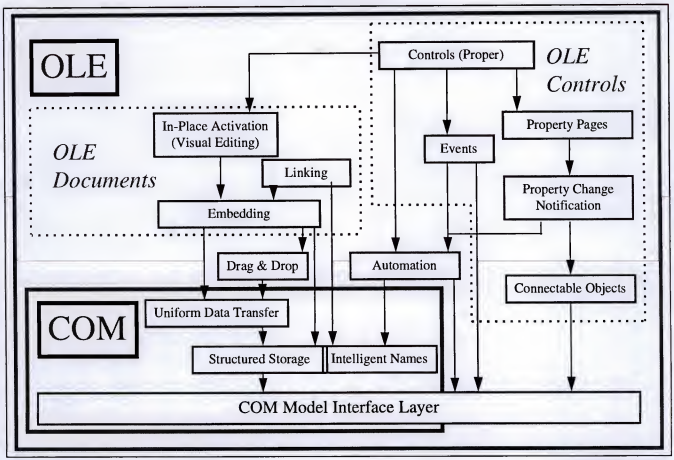
1. Description

To understand the need for component integration, a further analogy to electric power is useful. Components are like electric appliances; they must work anywhere in the same way that a GE light bulb works with a Philips lamp—and vice versa. Just as utility equipment suppliers, owners and managers agree on voltages and current measurements and —even across large geographic areas—the size and shape of plugs and receptacles so as to make electric appliances useful the suppliers and users of component software have to agree on how components will be managed together.

Just as ORBs define how "foreign" objects interoperate, OLE and OpenDoc are intended to define how they will be managed and integrated. OLE's architecture at a technical level is illustrated in Exhibit III-10.

Exhibit III-10

OLE Architecture



Source: Microsoft

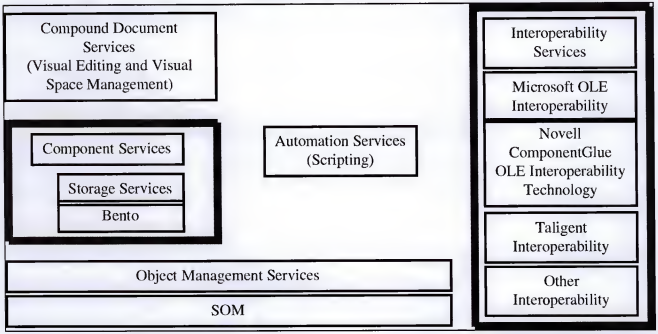
The area labeled "OLE Documents" has the components familiar to users of Microsoft Office that enable spreadsheets and graphics to be linked or embedded in Microsoft Word documents. The area labeled "OLE Controls" is most closely linked to the user interface and goes beyond "Controls" (proper), which refers to the control (.OCX and .VBX) files used by programmers to support such features as slide bars, menus and window scrolling. It includes events such as mouse clicks, arrival of a mail message or window opening. Furthermore, it includes "Connectable Objects" that include windows. These two main areas are linked to the "COM" area that provides storage, naming and data transfer over a common interface layer.

OpenDoc technology is illustrated in Exhibit III-11. It is similar to the OLE architecture, but with some notable differences. The storage system, based on Bento, is designed from the outset for component software storage. Lotus has used Bento for some time. Developers generally believe Bento provides superior storage for objects than

Microsoft's architecture. OpenDoc developers—in particular Novell with ComponentGlue, have gone to considerable lengths to provide OLE interoperability.

Exhibit III-11

OpenDoc Architecture



Source: INPUT, Novell, CI Labs

Exhibit III-12 compares OLE and OpenDoc standards. Both are modular, but OpenDoc places greater emphasis on interoperability with other architectures, like that from Taligent. The storage structures differ in that Microsoft supports atomicity (i.e., storage of subcomponents as separate elements), whereas OpenDoc does not.

OpenDoc is Mac-oriented when it comes to its scripting language, leveraging the investment in AppleScript that is already widely deployed. For the OLE environment a number of nonstandard scripting tools are available from third parties and Microsoft. This offers alternatives, but creates new standards. Both architectures support a range of object granularity; Microsoft supports its existing programming development environments of Visual Basic and Visual C++. Indeed, with the OpenDoc development kit for Windows, Visual C++ may be used as an OpenDoc development tool. OpenDoc is an entirely new architecture, which will require new cross-platform programming environments if it is to succeed. Taligent could provide that environment, as could Apple. IBM is tightly connected to OpenDoc via the underlying object architecture. The battle could

become one between IBM and Microsoft, should IBM control and promote the development environment for OpenDoc.

Exhibit III-12

Comparison of OLE and OpenDoc Standards

Issue	OLE	OpenDoc
Modularity	Tightly integrated, supporting installed Windows applications like Word and Excel	Components more easily interchangeable, later design with cleaner interfaces than OLE
Storage structure	Based on Microsoft docFile product, not a container-based design; supports atomicity	Uses a code-container design, but does not support atomicity, uses Bento file system that more readily supports components than traditional Windows file systems
Scripting language	Includes a procedural API that can be supported by any scripting language; no standard for event suites. Users can choose from many scripting languages.	Includes a procedural API that is supported by AppleScript, Frontier and QuickKeys; uses proprietary AppleEvent event-suite approach
Multimedia component support	No special handling	Separate object registry and management using Bento technology
Minimum object granularity	.OCX objects, similar to .VBX objects but fully distributable	OpenDoc provides a fairly granular object facility based on its 'parts' architecture
Cross-platform compound document architecture (CDA) standard support	No support planned for OpenDoc components. OLE runs on the MacOS in Microsoft applications but may not be prevasive there.	Support for OLE component interoperability is promised, as well as interoperability with Taligent's Commonpoint and X-Consortium's Fresco components
Underlying technology	Microsoft Windows-based Component Object Model	IBM OS/2-based Systems Object Model (SOM); total plug-and-play CORBA-compliant ORB support is promised for the future
Status	Version 1 SDK in use since 1991 with thousands of components already in use; Version 2 SDK promised in 1996	Version 1 SDK available for the first time i7/95; resulting components promised for 1996

Source: INPUT

2. Vendor Commitments

- Microsoft OLE—Only its drag and drop facilities are related to the original OLE released in 1991. Many of the illustrated interfaces and services are in extended beta test.

- CI Labs OpenDoc—Based on the Apple Scripting Architecture and Bento storage system along with the System Object Model ORB from IBM. Only an SDK has been released as of Fall 1995; actual components (called Parts in OpenDoc terminology) will not appear until 1996 or beyond.

3. Issues

- Defacto standards acceptance—unlike the CORBA vs. COM standards process, OpenDoc and OLE technologies are vying for defacto standard status based on market acceptance. CI Labs contends that its specification is fully documented and managed as a consortium, protecting the investment of developers that write to the specification. Microsoft replies that over 5,000 developers are involved in setting the OLE standard despite the fact that it—rather than a committee—controls any changes. Both claim to have shipped over 70,000 developer kits.

- Plug and play—technically, the OpenDoc approach is much more modular, in that various brand ORBs could be substituted for SOM and various brands scripting systems could be substituted for the Apple Scripting Architecture. That flexibility has yet to be proven in the market.

4. Future Directions

- The decision by the marketplace on a compound document architecture standard is critical to nearly every supplier's business plans. Unfortunately, this will not be settled until 1997 or later, when OpenDoc parts reach the market to compete with OLE components and COM functionality reaches the market—separately or as part of next-generation Windows operating software—to compete with CORBA-compliant ORBs.

- Microsoft has promised that existing OLE-enabled objects—and OLE-enabled objects developed today—will work within this emerging structure. It has also promised a category of additional services called OLE Interfaces for Enterprise Development for transaction monitoring and similar functions; the latter is completely a paper layer of OLE at this time.

- The compound document applications for both OLE and OpenDoc may use Internet-based technologies like HTML as it is extended to support more object types. For example, VRML will support CAD and multimedia. Sun's Java may prove to be the cross-platform language that enables compound documents in heterogeneous networks. On the other hand, for games, where the software components represent characters, weapons, places and other such items, OpenDoc and OLE may prove ideal for integrating objects.

- The PBX area is one area where software components are used like hardware, OLE technology is successful and a significant component software is emerging.

D
Typical Customers

Technically knowledgeable programmers form the market for CI Labs' OpenDoc software development kit and for application development tools that support OLE. They are likely to be C++ or Visual Basic programmers. Symantec is the leading C++ compiler vendor for the Apple Macintosh platform and sees OpenDoc as an opportunity for Apple programmers to create innovative applications.

Systems integration companies are particularly interested in gaining a competitive advantage using objects. Many already have significant investments in object-oriented technology to model business processes and create value in their organizations. Pioneers like Chrysler Financial, Swiss Bank, MCI, and McCaw Cellular have been able to gain competitive advantage through flexible systems that use object technology.

NeXT's OpenDoc developers are often Macintosh developers and include many small, entrepreneurial component vendors. Compiler developers like Micro Focus and Symantec are among developers critical to the success of OpenDoc. As of May 1995, CI Labs estimates over 70,000 developer kits for OpenDoc had been shipped..

OLE developers include those that use the Microsoft Visual Basic and Visual C++ developer kits. OLE can also be used from higher level applications like Microsoft Excel, which includes Visual Basic code.

Developers use ORBs to link major applications and include the systems integration arms of major hardware companies like ICL, IBM and HP. ORB developers also include enterprise programmers and application development tool developers.

Packaged software users can expect to buy a basic application that they can customize by adding component software modules. Microsoft has already started this approach with Microsoft Word, where Internet and file conversion features are available on the Internet as add-on products.

E
Customer Motivation and Issues

An incentive for customers moving to OpenDoc and OLE is to provide more flexible user interfaces. Just as Windows and the MacOS had a profound impact on the look of applications, OpenDoc and OLE promise new user interfaces. For example, one can display a compound document that shows embedded buttons and a moving ticker tape of stock prices using OLE or OpenDoc. Animated applications and games will be beneficiaries of component software on the desktop.

The motivation to move to ORBs is to provide interoperability across a network. Traditional systems architectures are too complex. They do not mask network complexities from the programmer. Some observers estimate that 80% of IS budgets is for maintenance. Companies need to be able to swap components in and out of a system to support new hardware, fix bugs and make business changes.

Users need applications to be built fast. By investing in objects and disseminating them throughout an organization, IS managers expect to reduce programming time. Another IS imperative is to connect remote branches, mobile employees and external organizations into mainstream corporate environments.

Applications such as document image processing that require retrieval of images, data and documents from different storage areas can benefit from the application of ORB technology. Another area that is likely to benefit is customer service, where systems must interface to databases, document retrieval and telecommunications systems.

IV

Vendors

This chapter reviews vendors' corporate positioning, product marketing strategies, strengths and weaknesses and outlook for the future.

A
Vendor Strategies

1. Major Suppliers' Positions on Component-Software Standards

In the area of distributed-object-management standards, the major systems manufacturers and operating systems suppliers — other than Microsoft — follow a two-pronged approach:

- continue to promote conformance to the Object Management Group's OMA object model (of which CORBA is a part and on which OpenDoc rests)
- simultaneously develop and deliver OLE COM-compliant product capability

Digital is forthright and aggressive about this two-pronged approach, hoping to marry the market acceptance of OLE-enabled component software with the proven technology of its CORBA product, ObjectBroker, via an OMG-compliant CORBA-OLE gateway. Digital to date has refrained from promoting its position on OpenDoc. This positions Digital strongly as a supplier of network-based products and support services. It was very successful in the early 1980s connecting to IBM systems. In the same way, it can leverage its expertise in both Microsoft and OMG environments to become a product leader.

Apple, Novell and IBM are the drivers behind OpenDoc. Apple could be judged least aggressive about its two-pronged approach, positioning

itself in competition with OLE in its OpenDoc literature, but leaving the door open about OpenDoc's OLE interoperability technology.

IBM, HP and Sun intend to follow Digital's lead, depending on whether the market accepts the COM or CORBA interconnection technology. Their positions vary on OpenDoc, with IBM—(as an investor and technology contributor)—and HP providing a broad range of distributed object solutions via OpenStep, Distributed Smalltalk and Taligent's frameworks.

Novell's strategy cannot fairly be termed two pronged. After a false start in 1993 with a CORBA-based approach, Novell apparently will build a completely ORB-neutral interconnection scheme based on NetWare Directory Services and the TUXEDO transaction monitor. (Note: the term "apparently" is used because just as this report was being sent to print, the Novell product was announced, it reorganized its management to address this area, and it announced it was selling off it its office applications software division.)

Microsoft is very clear about its strategy. It is starting from a position of strength: two hundred million OLE COM platforms on enterprise desktops. Microsoft owns the desktop. In reality, Microsoft is primarily aiming for management of other platforms, rather than following Sun's strategy of managing the network, through a program called Windows Interface Source Environment (WISE). WISE provides both SDKs and emulators that enable UNIX and Macintosh developers write to the Windows 32-bit APIs as well as objects (OLE), providing compatibility and portability. To the extent that WISE succeeds, interoperability and standard compound document architecture is less of an issue. Microsoft concedes that interoperability will always be required in some form, so it is also working toward control of the network via DCE.

2. Other Market Participants' Position on ORB Standards

Other independent software vendors—most importantly, Oracle—and second-tier operating environment suppliers such as NeXT walk a fine line between Microsoft and the systems suppliers.

Toolkit suppliers, such as C++ library vendor RogueWave (not otherwise covered in this analysis), and object-oriented development environment providers (such as Antares Alliance Group and Texas Instruments) are less involved with the standards adherence and acceptance issues.

Object-oriented toolkit suppliers are realizing real revenues today—and building businesses—based on providing object technology by selling their own object models. The development environment suppliers believe that the debate over distributed-object-management standards is a platform issue. They fall into two camps: those with cross-platform solutions, like ParcPlace-Digitalk, and those with a single-platform focus, like Borland, whose C++ development tools support the PC (albeit on DOS as well as 16- and 32-bit Windows). The cross-platform vendors will support whatever platforms emerge, migrating their own object libraries to support the underlying operating environment. PC vendors like Borland will race to keep ahead of Microsoft.

Database suppliers, including object-oriented DBMS suppliers, are less concerned about the development of such standards. The database community has defined its own standards for interoperability, such as ODBC. For systems that go beyond simple database access and retrieval, a distributed object infrastructure will be required. Acceptance of and commitment to a single distributed object model would simplify their internal development, as did agreement on SQL.

3. Vendor Classification

Consistent with the above analysis, Exhibit IV-1 classifies the systems and software suppliers covered in this report into one of four categories:

- Microsoft and its allies: in addition to Microsoft, there are component software suppliers with products or planned products completely oriented to OLE as a compound document architecture and to COM as its underlying object interoperability mechanism
- CORBA developers and their allies: primarily the major systems suppliers, each of whom—with more or less commitment—is attempting to win the battle of the server by making their distributed object interoperability mechanism the defacto standard
- Other systems suppliers and object-oriented software vendors that dependent on the evolution of distributed-object-management and compound-document computing standards

- Object-oriented software vendors that are not dependent on the evolution of distributed object standards

Exhibit IV-1

Vendor Classification, Critical Success Factors and Risks

Classification	Members of group	Success Factors	Risks
Microsoft and Allies	Digital[1] Micro Focus Microsoft Texas Instruments	Dependent on success of Windows as a server operating system	The open question is whether Windows NT and successor Microsoft operating system software is capable of handling enterprise-wide, mission-critical applications; n the short term, UNIX and MVS will be the leading high-end OSs
CORBA developers and allies	Apple Digital[1, 2] IBM Hewlett-Packard[2] SunSoft[2]	Dependent on continued need for heterogeneous computing standards	CORBA 2 gateways and bridges, designed to mask transports and provide interoperability with the desktop, may have performance implications that inhibit truly standard CORBA acceptance
Other suppliers whose success depends on standards evolution	Expersoft ICL Iona PostModern	Acceptance of either a distributed-object-management or compound-document standard is key to their plans	OpenDoc depends on CORBA, a loose but widely implemented server specification for server-based object management; OLE automation and related components depend on COM, a rigorous but as yet unimplemented standard
Suppliers whose business plan success is independent of standards evolution and acceptance	Antares AT&T GIS[2] Centerline Hitachi I-Kinetics NetLinks NeXT Novell Oracle ParcPlace-Digitalk Teknekron Visual Edge	These companies are basing product on both emerging applications standards, component software products are not dependent on standards	Very little dependence on the acceptance of either a distributed-object-management or compound-document standard

(1) Note that Digital is placed in both the Microsoft group and as a developer of CORBA.

(2) Along with HyperDesk, which no longer markets a distributed-object-management system, and Object Design, these are the developers of CORBA.

Source: INPUT

The original CORBA was a combination of a joint HP/SunSoft static-invocation-based proposal and a Digital/HyperDesk dynamic-invocation architecture. AT&T GIS and Object Design actually contributed more moral support than technology.

In Exhibit IV-2, these suppliers are also divided among those that sell ORBs, those that sell components that interface with ORBs, and those that sell other related object technology.

Exhibit IV-2

Types of Object Technology Supported

ORB Suppliers/Product	ORB-oriented Component Products	Related Software
Digital ObjectBroker[1]	Antares Huron ObjectStar[1]	Centerline Object Center[1]
Expersoft PowerBroker 4.0[1]	AT&T GIS Cooperative Frameworks[1]	NetLinks[1]
HP ORBplus	Apple OpenDoc 'parts'	Novell Object Services
IBM's (Distributed) Systems Object Model[1]	Hitachi transaction product	Oracle Power Objects[1]
ICL's DAIS[1]	I-Kinetics RDBMS and Framework Components[1]	ParcPlace-Digitalk Smalltalk[1]
Iona Technologies' Orbix 2.0[1]	Micro Focus Object COBOL[1]	Visual Edge[1]
Microsoft's Component Object Model (COM)	Microsoft COM-compliant applications	
NeXT Portable Distributed Objects (PDO)[1]	Teknekron	
PostModern Orbeline 2.0[1]	Texas Instruments next-generation repository product	
Sun's Project NEO (was DOE)		

(1) Currently available

Source: INPUT

B
Vendor Strengths and Weaknesses

1. Antares Alliance Group

Strategy

Antares Alliance is an independent software vendor originally funded by Amdahl and EDS. The motivation was to combine Amdahl's technology with EDS' consulting expertise. It has no direct sales channel but markets through channel partners, the largest being Amdahl.

The company is concentrating on integrating legacy applications and data into a distributed object environment. Originally named Huron by Amdahl and in the process of being renamed ObjectStar, this development suite predates both CORBA and OLE COM. Thus, the basic product includes a proprietary object broker but also interacts with CORBA-compliant ORB and OLE-compliant objects. ObjectStar uses its proprietary ORB, Data Object Broker, to manage the

execution of objects stored in both an ObjectStar-managed "MetaSTOR" and external data sources, typically RDBMSs and traditional MVS databases. It runs on MVS, AIX, Solaris, Windows/NT and HP-UX servers interacting with TTY, 3270, Motif and Windows clients. Once an object is passed to the distributed execution environment by the proprietary ORB, the environment manages data, methods and presentation. To access a non-Motif client, including ODBC drivers, OLE Automation components, C APIs and TTY connections—ObjectStar uses IONA Technologies' Orbix ORB.

Strengths

- Enterprise experience with large systems
- Early to market with understanding of distributed object infrastructures in a corporation
- Able to migrate legacy code and established applications into a modern distributed object environment
- Transport independent
- Repository scheme links objects dynamically at run time

Weaknesses

- Proprietary ORB provides no direct interoperability with other object systems
- Not yet totally object-oriented; Antares Alliance plans to use object philosophy in a future version of the non standard MetaSTOR repository

Outlook

As Exhibit III-8 illustrates, Antares Alliance is CORBA-oriented; not nly did it choose Orbix to interact with the outside world but its proprietary ORB behaves more like CORBA than COM. Antares is least dependent on the acceptance of any standard. It has its own ORB and could just as easily interact with clients via OLE COM. The ObjectStar approach to protecting an enterprise's investment in legacy applications and recently acquired client/server infrastructure makes sense to long-time CICS and TSO users accessing IMS and same-generation databases. Adding distributed object infrastructure to

these systems can help reduce application backlogs and the cost of maintenance.

As a systems integrators' development environment, ObjectStar has much to offer, but Antares Alliance's EDS connection may limit the group's ability to market ObjectStar to other systems integrators. However, it already has relationships with HP's Consulting Services group and Keane Associates.

Antares is in a strong position to compete for enterprise business. However, it needs to increase its marketing visibility among corporate users who are selecting a development environment. It may also consider investing in more LAN-based and client software to make its environment more scaleable at the low end.

2. Apple

Strategy

Apple's primary goal is to sell Macintosh computers based on the PowerPC processor and on System 7 and follow-on operating environments. It is also making some headway with turnkey servers for the Internet. It is moving more toward consumers and mobile professionals with its emphasis on Newton, notebook computers and digital cameras.

From an Apple corporate perspective, software components have an important tactical role that helps it catch up with Microsoft. Apple has traditionally been ahead of Microsoft in its operating system development, but Microsoft has taken shortcuts to a component world with OLE, using incremental improvements to its platform.

Unlike Microsoft, Apple is making no pretense to any segment of the enterprise computing market other than workgroup productivity applications on the desktop (some of which support workgroup computing). Apple markets an OpenDoc Software Developers' Kit for the MacOS and IBM markets one for OS/2. IBM has recently taken over the development kit for OpenDoc for Windows from Novell.

Strengths

- Apple has years of object experience. Apple understands that its early adoption of windowing technology gave it a tremendous competitive advantage. It believes that in the multimedia era, OpenDoc gives it an opportunity to move desktop applications to another level with

animation, integration of multiple data sources and unprecedented ease of use.

- Apple is a founding member of the OpenDoc Consortium and funds Component Integration Labs, together with IBM and Novell. This gives it early insight into OpenDoc trends.

- Apple is seeing strong demand for its PCs and workstations, its strength being integration of the OS with the hardware. In the OpenDoc environment, it can ensure that the software is integrated well into its platform.

- Apple has a strong presence in the educational market, one in which OpenDoc can really add value to computer-assisted training applications by drawing on data from many sources. Apple needs to leverage this advantage. If it can do this successfully, students trained on Apple Macintoshes may take them with them into the corporate environment and thereby strengthen Apple's enterprise market position, at least on the desktop.

- Apple's strong public relations means that OpenDoc toolkits have been widely disseminated.

- Apple, together with software providers, is seeing success with its Web servers for the Internet. It is less likely to be blind-sided in the corporate document arena by the emergence of compound document architectures from the Internet than are some other PC vendors.

Weaknesses

- OpenDoc is currently a development toolkit and a set of definitions on paper. Apple has not been quick about contributing to the deployment of OpenDoc.

- Apple has poor distribution channels for its MPW development tools through APDA. Fortunately, Symantec is increasing its retail presence with its C++ compilers for Apple's platforms.

- Apple has not yet discovered "killer applications" using OpenDoc that can give it a competitive advantage over computer vendors such as Compaq that deploy Microsoft's technology.

- Apple development is slow, with its next generation Copeland and Gershwin operating systems still under development.

- Apple perceives itself as being on a competitive battlefield with Microsoft. In fact, its core competencies are very different, and it has failed to leverage its advantage of vertical integration, i.e., integrating the OS with the hardware with a few basic applications. It is this failure to realize its true advantages that has weakened Apple's market position.

- The compound-document paradigm restricts applicability to enterprise requirements such as transaction processing and real-time data acquisition and control. Technically, there is no reason why OpenDoc is not an excellent environment for showing real-time data in a window of a compound document. Apple needs to market OpenDoc using other paradigms to go beyond the publishing area that it knows well.

- Apple does not understand how to sell solutions; it must to position OpenDoc as a solution rather than a technology. It also needs to promote innovative applications and their vendors.

Outlook

INPUT believes that Apple, wanting to differentiate itself from the Microsoft environment, is strongly relying on OpenDoc and, by implication, Distributed Object Management (DOM), which leads to CORBA standards for enterprise connectivity (see Exhibit IV-3). However, it has abdicated ownership of key technologies, transferring:

- Bento, OSA and related CDA technology to CI Labs

- Pink technology to Taligent

Believing its proprietary OS to be a marketing disadvantage, it has made efforts to license its technologies to clone vendors like Radius. In an attempt to be open, Apple is licensing object technologies via CI Labs and Taligent. It clearly believed that as good as the technology was, it was never going to succeed if it was marketed within Apple, because it depended on a platform with less than 20% of the market. Apple perceived the competitive arena as "Macintosh versus Windows" and acted decisively to try to build a broader market demand for its inventions.

Microsoft has component software on the market today. Apple needs to accelerate the marketing and support of OpenDoc components, which are lagging behind the Microsoft component market. Apple

needs to attract and be seen to be partnering with major professional services firms and systems integrators. It should also create a strong VAR program and resell VAR components.

Apple is hindered in taking advantage of OpenDoc as long as it sees itself competing with Microsoft, rather than competing for the customer and the third party software developer. Over the past two years, Apple let its third-party software efforts diminish. It also lost focus on its role as an enterprise client platform vendor by concentrating on home and education markets as well as on its Newton PDA. Apple needs to be more proactive in the component software arena, enabling its developers to create the best component-based applications and supporting a distribution channel for third-party software components.

3. AT&T GIS

Strategy

AT&T GIS (formerly—and possibly again to be—NCR), like other systems suppliers, wants to sell platforms and service. NCR was one of the first vendors to recognize the value of object-oriented software for the desktop, with its COOPERATION concept, but with its acquisition by AT&T and competition from Microsoft, it lost its lead.

As a result of its early experience and long history of software engineering expertise, AT&T GIS is well-positioned to take advantage of enterprise software built from components on its platforms. With strong networking expertise, it recognizes the importance of standards and will also ensure compatibility with the emerging component software standards.

Based on NCR's CORBA contribution, AT&T GIS markets a line of developer component software under the banner of the Cooperative Framework Libraries. It is a set of object technology services that were originally part of NCR's COOPERATION concept. The Framework Libraries include directory services, distributed processing management, customization, systems management, security, logging and exception-handling services (all of which are analogous to, but not compliant with, the OMG Common Object Services), and a human-interface framework. AT&T has announced the intention to make these object services OMG COSS and CORBA compliant. These services may eventually be combined with the AT&T TOP END transaction-monitor technology, also acquired in the NCR purchase, in

the same way that Novell has combined the TUXEDO transaction monitor with its NetWare object-services offering.

Although it is listed as a CORBA technology contributor, in the market it will probably concentrate on OLE compliance to satisfy its PC users and enhance its position in vertical LAN markets like retailing and banking.

Strengths

- A wide variety of languages, operating systems, and communications techniques are supported
- Experience in large distributed transaction processing networks
- Retail, telecommunications and banking expertise
- Understanding of networking and how to integrate large systems (e.g., a major retailing customer has 10,000 computers that appear as one using AT&T technology)

Weaknesses

- Neither CORBA nor OLE compliant; slow to move
- Not yet tied into other strategic AT&T GIS efforts such as TOP END or Teradata follow-on products

Outlook

AT&T GIS is not going to be a strong supplier of component software, except it will sell its Framework Libraries to its sizable installed base. It has the potential to build up a significant component library for its own systems integration business, with emphasis on components for distributed transaction support and imaging. However, its current component software libraries are not receiving the R&D emphasis from AT&T that object technology-dependent ISVs are placing on equivalent products, thus AT&T will lose its early lead as the supplier of a viable and useful development framework.

The uncertainty surrounding the company and the possible sale of some divisions mean that AT&T GIS can still choose whether to be an enterprise software vendor, using component technology. In the wake of the AT&T break-up, the company has the opportunity to develop a strong corporate identity so that customers can understand its

strengths. AT&T GIS has the potential to be a leader in transaction processing using OO technology, if it moves fast. It is likely that it will use its components to benefit its own systems integration activities.

4. CenterLine

Strategy

CenterLine, a privately held Cambridge, MA application development tools vendor, develops and markets object technology development environments and quality-assurance software. Its main thrust is its Object Center C/C++ development environment for the UNIX market. It positions itself as one of the few tool suppliers in the object technology space that includes a compatible quality assurance (QA) component. In short, CenterLine brings to C++ the support for objects and components that has long been a part of Smalltalk environments.

The Interactive Workspace component of Object Center lets programmers test code iteratively and without having to write a complete program or module. After completion, the program or module is then compiled with CenterLine's C or C++ compiler for better run-time performance.

Object Center's Data Browser lets programmers inspect complex objects and modify data structures; objects can be manipulated directly from within the browser. In addition to letting developers step through code line by line (including C++ templates), Object Center is used to set breakpoints and watchpoints. Class and library information can be viewed with The Class Examiner and Inheritance Browser, which graphically depicts all levels of the hierarchy and includes virtual classes. The Class Examiner displays member functions or data. Filters let programmers separately explore inherited, public, protected or private interfaces and selectively display static, virtual, constructor or other member types.

Strengths

- Early supporter of C- and C++-based on components and rapid application development (RAD)
- Independent tools that incorporate unique QA suites
- Recognized position as one of the leading language suppliers in the UNIX workstation market

Weaknesses

- Lacks team-development abilities
- Higher level components are missing. The company needs to partner with companies that can supply vertical market and cross-industry components based on its technology.

Outlook

CenterLine typifies the object technology supplier that is neutral on the DOM standards available and relatively unconcerned about whether one or more particular standards is adopted (see Exhibit IV-3). Like other C++ and Smalltalk product suppliers CenterLine emphasizes ease of program development rather than management of the resulting components. CenterLine's QA differentiate it from other tool providers and the suppliers that emphasize integrated suites. If CenterLine adds higher level components to support enterprise developers, it should be able to expand its business into more enterprise development sites, which commonly find C++ difficult to teach to COBOL programmers.

CenterLine needs to partner with vertical market and cross-industry developers and integrators to identify components that it can either reference sell or resell from other developers. It also should to consider developing components for vertical markets to strengthen its thrust into the object management arena.

5. Digital Equipment

Strategy

Digital, like other systems suppliers, is interested in software components to protect or expand its systems business on both Intel and Alpha processors. Digital's Object Broker (formerly called ACA Services) is based on one of the foundation CORBA technologies. It is currently being further developed by incorporating a gateway to Microsoft COM that Digital hopes will be accepted as an OMG standard.

Digital's Object Broker provides server-registration, method-invocation and data-exchange functionality across HP-UX, IBM AIX, SunOS, Apple's System 7, Windows 3.1, and Windows NT, as well as Digital's own OpenVMS AXP, OpenVMS VAX, ULTRIX RISC, and Digital UNIX operating software.

Strengths

- Proven experience with object technology and distributed object management
- Strong messaging experience, which enables Digital to cover both the CORBA standards and the message-oriented middleware standards for less time-critical applications
- Strong research capabilities in networked systems; co-inventor (with HyperDesk) of the dynamic invocation interface, which should have increasing application in enterprise computing applications
- Close work with Microsoft on several projects, including attempting to bridge the COM and CORBA environments, has given Digital considerable experience in determining what types of interfaces and gateways work and which ones are tricky to implement.
- Digital is making a major thrust with Windows NT. This should position it as a major player and platform provider for systems built around the OLE COM environment. With its Digital UNIX offerings and enterprise connectivity, it should see success in the CORBA market.
- As a services provider for Windows NT and Windows 95, Digital gains access to Microsoft's customer base. It needs to supply more products to this market using component technology.
- Digital appears to have overcome earlier financial problems as the market accepts its RISC-based computers using its Alpha processor. Digital also has a strong PC business which positions it closely to Microsoft.

Weaknesses

- ObjectBroker currently depends on DCE's RPC mechanism. This has the advantage that it works with the Microsoft environment and masks it from underlying network protocols, but has the disadvantage that the requirement for DCE RPC limits its applicability. Limitations may include poor network performance and inability to support complex transaction routing in a multi vendor network.
- The COM-to-CORBA bridging project failed because it required an object model fundamentally different from the CORBA object model. Digital will now develop a gateway product.

- Digital lacks a good desktop presence with its own proprietary software. It has strong groupware, messaging and systems management tools, but has failed to capitalize on these in the way that HP has with OpenView.

- Instability in its systems integration and services businesses.

Outlook

Digital appears to be well-positioned to take advantage of the emerging component software market, no matter which technology orientation—if either—predominates. As Exhibit IV-3 illustrates, INPUT believes Digital is ORB-neutral. Digital is the least dependent on standards acceptance of all the mainline systems-supplier competitors. However, it is not as independent of standards acceptance as tool and database suppliers, or object technology suppliers, such as Antares and Texas Instruments, that are building product on top of a distributed object management structure . Digital is in both the UNIX camp and the Windows NT camp for products and services, as well as in its own OpenVMS markets. This provides it with a range of enterprise solutions and it is well-equipped to provide cross-platform solutions using component software. In addition to the development agreement with Microsoft, it is working with Expersoft, Object Design and others to supply its customer base with advanced development and end-user component software.

Digital needs to create major platforms with its technologies, as it did with All-in-One for office automation in the early 1980s. Here is an opportunity for Digital to use its connectivity and object-oriented expertise either alone or with outside software developers to create major systems for competitive advantage. Messaging, integrating the Microsoft MAPI, Internet and Digital technologies, is a market in which Digital could become a leader.

6. Expersoft

Strategy

Expersoft's PowerBroker suite of products (formerly XShell) claims to integrate objects written in C++, Smalltalk, IDL and Visual Basic—via CORBA and/or COM—on a variety of platforms, including Windows, Windows NT, Sun OS, Solaris, HP-UX, AIX, IRIX and Digital UNIX. Expersoft has integrated PowerBroker with ODI's ObjectStore database, Rogue Wave's Tools.h++ class libraries, Persistence's tool for

object-oriented access to relational data, and GUI builders from Visual Edge, Imperial Software, VISIX, and OpenWare. Expersoft also has strategic OEM relationships with many companies, including Digital.

As XShell, the product suite predates CORBA. Expersoft's Extended C++ product contains a preprocessor, class libraries, and management tools. The class libraries provide developers with service classes and enhancements to support advanced C++ functions. PowerBroker CORBA C++ is a CORBA 2.0-compliant IDL compiler used in conjunction with Extended C++. PowerBroker bridges to COM-compliant objects via a wrappering technique.

Strengths

- Experienced provider of object management products with strong visibility in the component software arena
- Agile, emerging vendor
- PowerBroker 4.0 product line (formerly XShell) promises both OLE and CORBA 2.0 support
- Strong OEM relationships and ability to leverage technology.

Weaknesses

- No current DCE strategy
- Expersoft's long-term strategy risks competing with systems suppliers for developer component software revenue. This could alienate Expersoft from systems suppliers.

Outlook

Expersoft is the least dependent on CORBA of all the independent ORB suppliers (see Exhibit IV-3). And unlike Iona and PostModern, it is not even very dependent on the acceptance of DOM standards. The underlying standards become less critical because Expersoft offers a complete product offering of language systems, class libraries, and object services in addition to its ORB. A new agreement with Digital demonstrates the potential success—and risk—of Expersoft's strategy of providing a full array of object technology products instead of a simple middleware product. As Expersoft expands, it more directly competes with IBM, Sun and Hewlett Packard as well as Microsoft.

Expersoft may consider partnering with a middleware provider to serve the OEM market jointly. It must continually innovate and expand its class libraries. It may also consider acquiring class libraries from others that it can publish. At some point, Expersoft will need to expand its distribution to VARs and systems integrators. To do this, it will need more powerful application development tools. Expersoft risks being acquired by a systems supplier or major software company in the long term.

7. Hewlett-Packard

Strategy

HP was the first of the CORBA founders and systems suppliers to publicly recognize that an ORB is just a piece of middleware. On the desktop, HP was one of the first vendors to link desktop applications with agent technology using NewWave.

Originally planning to develop its own complete environment, HP subsequently decided to give its customers a wide choice from among these and other leading object software suppliers. HP markets its own and jointly developed object technology tools, including its Softbench development suite for C++ programmers, Distributed Smalltalk (which adds CORBA standards to ParcPlace's Smalltalk) and object-oriented DCE, an object-oriented layer on top of DCE that works with an HP transaction-management product based on Encina. HP is also collaborating with NeXT and has taken an equity position in Taligent. It sees Taligent as a solution for IBM customers, NeXT for distributed computing customers and Smalltalk for those wanting to have cross-platform solutions and access to legacy systems.

HP's Fusion implementation methodology enables it to provide complete solutions. HP views OO technology as only one piece of a solution, believing that process reengineering is another important service element when selling an OO solution. HP recently started an OO Solutions Center as a professional service that draws on personnel throughout the HP organization.

HP has an initiative with Intel to design the next generation of desktop microprocessors. It will be crucial that HP and Intel interest third parties in providing a complete infrastructure, including component software tools and packages, for this environment. HP's long-term success depends on its being able to evolve more quickly and cost effectively than its competitors into being a supplier of advanced

PC platforms. If it can do this and successfully bridge its PC and enterprise hardware lines, it will be in a strong position to own the network.

Customers of HP include the Bank of Switzerland which developed a distributed object trading system, and Services Canada which tracks ice flows with OO DCE. HP has seen programming time reduced by a factor of five using Distributed Smalltalk over conventional COBOL or C++. OO technology typically enables a company to build a prototype fast, but deployment of a large OO application may take much longer because of integration and personnel training issues.

Strengths

- Long corporate experience with object technology, including in its instrumentation business. HP recently set up an object group to support its object-oriented products.
- A wide range of object technology tools supports HP's corporate strategy of selling systems and services.
- HP sells both PCs and UNIX systems.
- HP is strong in systems management over networks with OpenView, which gives it an advantage in trying to control network software and network-based applications.
- HP has many distribution channels: retailers for PCs, distributors, VARs, dealers, systems integrators and direct. It can use these to sell software components for different applications.

Weaknesses

- ORB philosophy—and overall distributed-computing philosophy—is based on DCE, which has received minor support to date from corporate IT groups and negligible support from independent software vendors; HP plans a non-DCE-based ORB implementation.
- Dependent on third parties for COM interoperability (in fact, may depend on third parties for all distributed object management)

Outlook

HP has a strong track record of understanding where it can and cannot succeed and moving on. HP is dependent on DOM standards

acceptance. It has, to date, chosen not to try to dominate object-based distributed computing like Sun and IBM have done. Instead it will offer a broad range of solutions and follow standards as they emerge.

HP's New Wave product ultimately failed, because after initial success at linking desktop applications, it wanted to leverage the component integration technology into a grand architectural scheme, DAA (Distributed Application Architecture). DAA was a proprietary standard that was overtaken by OLE, which had broader support for tools vendors, resellers and integrators.

HP's recognition of an ORB's importance—as middleware that can be connected to many different object technologies—is unique. However, as Exhibit IV-3 illustrates, HP, at least in its enterprise systems products, is still very CORBA-biased. INPUT believes that that bias is historical, with no underlying technical restrictions. However, HP is emerging as a primary Windows NT services vendor and is also strong in PC sales. This potentially puts it as a strong player in the COM camp, but to date this has not been articulated clearly. HP can be expected to follow Microsoft's standards long term, as well as a leader in the CORBA world.

8. Hitachi

Strategy

Hitachi Information Systems has an object technology strategy similar to that of HP and Sun: It will keep up the research investment; lead a loyal customer base into a new model of information technology with clear answers and workable solutions; and build on existing strengths. In Hitachi's case, that involves exploiting its strength in transaction processing. Following this incremental strategy, Hitachi has recently announced that it is combining the object-oriented tool set it has been marketing since 1993 (Object IQ and C++ class-library framework, object-oriented COBOL compiler and object-oriented database management system (OODBMS)) with the PostModern CORBA-compliant ORB to build an object-oriented transaction processing system. Hitachi also markets an OLE-compliant GUI-based development tool called APPGALLERY.

Strengths

- ObjectReuser for C++ code reuse and object-oriented COBOL compiler

- Transaction management experience

Weaknesses

- Current OSF DCE/RPC dependence for most of its tools
- No position on bridging the gap between its OLE- and CORBA-compliant tools

Outlook

Hitachi has had a long-term strategy of converting its IS technologies into separate businesses; object technology is no exception. Its position as "not too dependent" on the acceptance of a DOM standard and "not too biased" toward either CORBA or OLE COM (see Exhibit IV-3), as appropriate for an $80-billion-plus technology supplier. Although such a position provides its worldwide IT customer base with assurance that its primary supplier is investing in this next-generation technology, it makes it highly unlikely that Hitachi will break out of the pack of tools suppliers, ORB suppliers, or systems suppliers to claim any sort of marketing leadership in object technology. Hitachi will be a strong follower, and that may be just what its customer base expects.

9. IBM

Strategy

IBM quietly led the adoption of object technology in the information systems business while rarely using the terms object technology or componentware. Both OS/400 and SystemView exhibit a strong object orientation. IBM's System Object Model (SOM) is a language-neutral object model designed initially for developing and reusing class libraries. It offers Replication and Persistence object services.

Although IBM was not an early provider of CORBA technology and not involved in the initial specification, it had two competing technologies in the CORBA 2.0 RFP process. IBM has promised to make SOM CORBA 2.0-compliant this year. SOM provides the underpinnings of Component Integration Laboratories' OpenDoc architecture.

IBM is also marketing VisualAge Smalltalk, VisualAge C++, and the wide range of other development tools and utilities users would expect of IBM. Visual Age is IBM's main focus for development tools that

enable the component market. It is essential that IBM promote tools that support OpenDoc if the standard is to compete successfully against Microsoft in the distributed object infrastructure battles.

IBM has another development strategy in Taligent, jointly set up by Apple and IBM. Taligent may offer too little, too but late on the other hand, it may be the enterprise development platform that can make IBM successful in network-centric computing. IBM, under Gerstner, has moved its focus from OS/2 to distributed object infrastructures, as the Lotus acquisition shows. It also has a strong networking thrust with the IBM Global Network and Internet products and services.

IBM sees OpenDoc as its opportunity to create components for users long term. Workgroup automation, mail, and calendaring are all applications in which components from IBM or its third-party software vendors will be found. Brokerage and Internet applications may benefit in the long run from IBM's commitment to OpenDoc, but IBM has to move fast if it is not to be outrun by Sun with Java. IBM sees OpenDoc as maturing for developers in the 1998 timeframe and as a technology for users around the turn of the century. It sees OpenDoc as an opportunity to move out from its corporate software base into creative software for the masses, games being an area with potential.

Strengths

- IBM marketing power
- Wide range of platforms and computing environments supported: OS/390, AIX, OS/2, Windows
- Can potentially support both the IIOP and GIOP CORBA standards with native implementations
- IBM has a strong vision for object technology

Weaknesses

- Highly tied to the OpenDoc initiative but currently less active in providing distributed object technology into enterprise-wide applications that play to IBM's traditional strengths and customer needs. Its systems management thrust with SystemView and its AS/400 initiatives may be its strongest enterprise OO offerings.

Outlook

IBM is a leader in distributed object management at all levels: systems-management applications, tools, class libraries, databases and middleware. It is funding consortia-based activity at Component Integration Labs and Taligent that may or may not further that leadership. As Exhibit IV-3 indicates, it is both CORBA-oriented and more dependent on standards acceptance than most of its system-supplier and ISV rivals. IBM object technology personnel intend to set and promote that standard with SOM.

IBM needs to provide a full suite of object-oriented enterprise frameworks, application packages and components. It needs to leapfrog the competition and promote innovative solutions through its systems integration business. IBM is ideally positioned to be the leader in winning global contracts for enterprise OO systems. It then has to continually update and modify them using components.

IBM historically has had many strategies, among its different organizations. It needs to develop a coherent strategy that goes beyond its client/server Open Blueprint to real products. Until it has strengthened its retail and mail order channel, which it can do with Lotus and the success of its ThinkPad notebook computer, it cannot succeed in component-based applications that go beyond OS/2 and Lotus products. IBM can use the retail channel to provide solutions to smaller organizations that need to interact with major enterprises. It can also support smaller organizations through networks.

Opening up the mainframe environment with OS/390 to make it more UNIX-like while maintaining MVS's historical strengths positions IBM well to leverage its SOM technology from the data center to other corporate departments. IBM needs to strengthen its midrange enterprise solutions by creating applications based on SOM. Data warehousing is an area that provides this opportunity.

10. ICL

Strategy

ICL offers a complete range of object services within an object framework called Distributed Applications Integration System (DAIS) Revision 2.1. The services include Alert, Recovery, Factory and Trader (a derivation of the ISO X.900 work), working with a CORBA 1.1 complaint ORB that supports OS/2, OpenVMS, 16-bit Windows,

VME and an assortment of UNIX operating systems on RISC and CISC chips.

ICL also markets the DAIS Information System 2.1, a distributed multi-database system built on top of the DAIS tool set. IDMSX, Oracle, Sybase, Ingres, SQLBase and Informix databases can be mixed and matched on a DAIS Information System, which would be considered a Common Facility in OMG terminology. Originally aimed at ICL's large-enterprise customer base, the company has recently started to market the concept to VARs and applications software vendors.

ICL has recently invested $30 million in a new Silicon Valley enterprise focusing on DAIS and distributed computing. DAIS is a particularly strong product, with over 100 enterprise installations in major systems. This positions ICL as a leader in CORBA implementation. It is particularly strong in the retail and image processing markets. ICL also markets other software, such as X.500 directories, through its Reston, VA office.

Strengths

- Illustrates clear understanding of applicability to ICL's traditional customer base in its team development and trading-based services and heterogeneous database facilities
- Portable to DOS/Windows, UNIX and proprietary ICL environments with Macintosh and NetWare support promised

Weaknesses

- Will find it hard to keep up with the efforts of larger systems suppliers, although it has recently announced another 300-plus developer years of DAIS development effort
- No COM position or plan; must depend on third-party gateways

Outlook

ICL is as biased to CORBA as any supplier (see Exhibit IV-3), but is less dependent on acceptance of a DOM standard than either the "pure" ORB suppliers or IBM, HP and Sun. DAIS is a well-conceived and-executed object system that will provide ICL customers reasonable value irrespective of the industry controversy over ORBs. ICL has promised more tools, services and facilities to complement

the environment and is ahead of its traditional competitors in bringing an object-technology system to market.

ICL in Reston could provide an ideal distribution channel for enterprise components to systems integrators.

11. I-Kinetics

Strategy

I-Kinetics' strategy is to connect different databases and ORB implementations. I-Kinetics markets a product set called ComponentWare, the first pieces of which are called the Framework Component and the RDBMS Component. The Framework Component acts as a bridge connecting among various ORB implementations.

The RDBMS Component offers install-and-run, object-oriented access to Oracle and Sybase relational databases and is compatible with Oracle Objects for OLE and Microsoft's Data Access Objects. The Framework Component layers on top of any ORB chosen by the developer; RDBMS Component has been demonstrated with Orbix, ObjectBroker, and COM. The product speeds development of ORB-based applications and allows development by programmers not trained in underlying object techniques because it is independent of development languages, tools and underlying object request brokers. ComponentWare is, in turn, part of what I-Kinetics calls the Virtual Application Warehouse, any organization's collection of commercially available component software, plus the organization's own wrappered legacy applications and data. These components can be fine-grained objects such as OLE Custom Controls or entire legacy applications.

I-Kinetics is the lead contractor in support of the U.S. Navy's NAVSEA/NUWC logistics planning and workflow integration project. I-Kinetics is also the leader of the ComponentWare Consortium (CWC). As such, its strategy is built around partnerships (including relationships with BB&N, Digital, Iona and others).

Strengths

- Virtual application warehouse concept built on ComponentWare has wide enterprise applicability
- Neutral position relative to CORBA and COM; presumably can support a proprietary ORB as well

Weaknesses

- Concept is universal; I-Kinetics is dependent on partners over which it has little leverage
- Data warehousing vendors will offer their own gateways and bridges, making I-Kinetics less valuable. The company has to keep ahead with innovative technology and rapidly leverage its customers to gain market share if it expects to become a leader.
- I-Kinetics risks being acquired by one of the data warehousing or application development tool vendors, which could make its products less universal.

Outlook

As Exhibit IV-3 illustrates, I-Kinetics is in the enviable position of being ORB-neutral and not too dependent on acceptance of a DOM standard. Its success as leader of the group working on the multi-million-dollar U.S. Navy contract indicates that it could successfully lead the ComponentWare Consortium to become a truly neutral arbiter in the OpenDoc and OLE controversy; it provides components for both. To be fair, CWC argues that its purpose is to promote vertical and horizontal applications components on top of OpenDoc and OLE Automation technology. But neither CI Labs nor Microsoft will be fair.

I-Kinetics is in a risky position. On the one hand, it could rise to be a leading tools and connectivity software vendor like Information Builders. Even with standards, there are many opportunities for connecting different object-oriented environments. On the other hand, it risks being acquired by a major developer, which may mean losing its neutral position. I-Kinetics has to race to stay ahead with technologies. The Navy contract is attractive, but it is the kind of contract that could consume many resources and make the company less agile. I-Kinetics needs more commercial customers and strong partnerships with resellers to succeed. I-Kinetics may consider acquiring smaller VARs and building a significant services business to integrate large distributed object environments.

12. Iona Technologies

Strategy

Orbix was the first full implementation of OMG's CORBA 1.1 standard and one of the first examples of a CORBA 2.0 ORB. Marketed by IONA Technologies of Dublin and Marlboro, MA, Orbix is currently available on 20 operating systems, including most versions of UNIX, Windows 95 and Windows NT. The Windows port is significant, as that version of Orbix incorporates both a Windows-based CORBA-compliant ORB and a gateway to COM, implemented by modifications to the Orbix IDL. The Orbix IDL compiler can generate a COM-compliant object in the same way that it generates a CORBA-compliant object.

Versions of Orbix are currently under development for OpenVMS, OS/2, MVS and other desktop and server operating systems. IONA has recently announced plans to integrate Orbix with ODI's ObjectStore database and has completed integration with Stratus' Isis Distributed Systems technology and Novell's TUXEDO transaction processing system. Orbix is an outgrowth of EC research into ORBs at Trinity College, Dublin. Sun holds a minority interest in IONA, which did the COM integration component of Sun's Project NEO.

Strengths

- Although CORBA-biased, Orbix has a good COM gateway capability
- Software is compact, meaning it can run on older and less powerful desktop computers
- Early acceptance by leading-edge users such as Motorola and Boeing means the product deserves consideration in every object technology sales situation
- The company knows how to leverage its development through OEMs and partnerships
- Agile—when the company started with a technology that was not supported by the OMG, it swiftly partnered with Sun to obtain the necessary support to make its software part of the standard

Weaknesses

- Sun's equity position is both a weakness and a strength; Sun's position on distributed object management fluctuates, making IONA's long-term position in that strategy vulnerable

- It is hard to find fault with Orbix's feature set, assuming that one believes that static and dynamic binding are equally important, that a lightweight implementation is preferable to deploying complex middleware throughout the infrastructure, and that CORBA and COM can be bridged without major performance penalty

Outlook

Iona is certainly dependent on CORBA, but not as dependent as are PostModern and most systems suppliers. Based on its Esprit background and smooth embracing of OLE COM, IONA has a business even if CORBA fails. IONA's business model is still based on the acceptance of some standard for distributed object management. It wants to hide network details from the developer the same way a compiler hides assembler language details, but it needs standards to make that plan executable. The existence of a standard is more important to IONA than whether that standard is CORBA, OpenDoc or OLE COM.

IONA can continue to leverage its customer base and supply it with more software to support the CORBA environment. It can also invest in its own components and sell them both to user organizations and to developers long term.

13. Micro Focus

Strategy

In the first releases of its Object COBOL compiler, Micro Focus has clearly chosen OLE COM over CORBA based on where it sees market growth for COBOL—enterprise connectivity from the Windows desktop. Micro Focus's decision is particularly indicative because, unlike many of the other suppliers analyzed in this report, Micro Focus had the luxury of making a decision fairly late in the standards-development process (1993 versus 1990 for most of the participants). The compiler features integration with OLE Automation capabilities and complete support of the Windows 32-bit environment (multi-threading, etc.) as well as a visual development environment.

Strengths

- Priced to quickly permeate the market and compete with Visual BASIC at $499
- Visual Object COBOL programs, as OLE service providers, can be controlled from a Visual BASIC or Visual C++ program
- Leverages an installed base of programmers who have been waiting for OO enhancements to COBOL, rather than moving to Smalltalk.
- Micro Focus is experienced with COBOL programmers' requirements and how to make the visual appearance of its software attractive.

Weaknesses

- No enterprise connectivity position, including connectivity to other Micro Focus compiler-generated programs
- Acceptance of COBOL by the object-oriented community is still unclear. Assuming acceptance, interoperability will be a major issue due to COBOL's dominance of enterprise business computing.
- Micro Focus is late to market with a visually appealing tool; had it been earlier it could have prevented some PowerBuilder, Smalltalk and Visual basic sales. It now has to play catch-up.

Outlook

Micro Focus is included in this review primarily as an example of a language-system supplier that is more dependent on standards acceptance than ParcPlace and CenterLine and one that is currently biased to one of the proposed standards. Micro Focus's success with Visual Object COBOL is highly dependent on the COM standard (see Exhibit IV-3). Should Micro Focus want to move to a neutral position or CORBA bias, it has the development resources. It has indicated that it will market CORBA-oriented products as market demand builds.

Given COM's conceded ownership of the desktop, Micro Focus's success with the product is fairly well assured, assuming market demand. Although Object COBOL is fairly dependent on COM being accepted as a standard, it could exist without COM for many installations, where PC to mainframe connectivity is sufficient. Micro Focus is less dependent on COM than are the object request broker

suppliers, systems suppliers and Microsoft itself, whose whole business model is based on standards acceptance.

Micro Focus must aggressively promote its tools to programmers outside the traditional COBOL base, such as Smalltalk programmers. It also needs to develop relationships with component suppliers to increase the number of interfaces that it can offer to other systems. It needs to expand its toolkit to support networked applications, possibly merging with C++ toolkit suppliers to give it a fuller product line. Micro Focus may also consider merging with a database language supplier or Smalltalk supplier to enable it to sell a wider range of products to corporate developers.

14. Microsoft

Strategy

As illustrated in Exhibit IV-3, Microsoft is both completely biased to one of the DOM approaches (its own Component Object Model) and it is one of the suppliers most dependent on acceptance of a standard. As it has said consistently, Microsoft defines standard acceptance as acceptance by the market, not a committee. Note that Microsoft defines COM as Component Object Model. At one time there was a joint Digital/Microsoft development effort—similar to the Sun/NeXT arrangement—to develop an object management architecture that would bridge the CORBA and COM gap. That group used the term Common Object Model for COM; Microsoft now disavows that use of the term.

Many analyses leave the impression that Microsoft is attempting to manage other platforms and the network simply because it has no more desktop space to conquer. But that assumes a nontechnical motivation that may not be entirely fair: Most object technology experts believe that all of the DOM functionality discussed in this report belongs embedded in the operating system, eventually. And many object technology experts believe that developers cannot achieve the compatibility and portability that WISE seeks, without object technology. From those two perspectives, Microsoft's interest and enthusiasm is justified on more than predatory grounds.

As to its actual object strategy, Microsoft is incorporating object technology in all levels of its OS, networking and applications businesses—and presumably in its on-line services business. The current OLE (sometimes called OLE 2) specification has little

relationship to the original OLE 'drag-and-drop' capabilities released in 1991. Microsoft has promised that existing OLE-enabled objects — and OLE-enabled objects developed today—will work within this emerging structure. Microsoft has also promised a category of additional services called OLE Interfaces for Enterprise Development for transaction monitoring and similar functions; the latter is completely a paper layer of OLE at this time.

Besides its Windows 32-bit (Win32) Software Development Kit (SDK) and server SDKs, Microsoft provides four major desktop development environments to create OLE-enabled applications:

- Visual basic 4.0—in Standard, Professional and Enterprise Editions
- Visual C++ 4.0
- Visual FoxPro 3.0 - a database development tool - in Standard and Professional Editions
- Microsoft Access—a database development tool

In addition, it provides the Microsoft Foundation Class (MFC) Libraries that support over 40,000 lines of classes to support OLE. Approximately 50 independent software vendors are expected to supply OLE components, but many more systems integrators and larger developers will offer components to enhance their installed systems.

Strengths

- Hundreds of real, useful objects sitting in class libraries, application frameworks and elsewhere, just waiting to interoperate
- A very precise, rigorous specification (other industry participants argue that this specification is subject to change by Microsoft at whim and can be used to give unfair advantage to its applications division; INPUT will leave the latter issue to the U.S. Justice Department but thinks the former concern is unlikely, given that 5,000 ISVs and other Microsoft partners are dependent upon it. One supplier, whose product set is truly unbiased as to DOM type, has commented that the "beauty of Microsoft products is that even when they do something bad, they do it consistently." This is a testament to a precise, rigorous specification.

- A very rich infrastructure of tools, third party developers, an on-line network, resellers and integrators supports its technologies. Microsoft understands leverage.

- Microsoft knows how to create awareness of its products in its customers' minds long before anything ships.

- A strong upgrade path from Windows '95 to Windows NT on the desktop using objects and components

- An ability to conquer the component software desktop market by redefining the concept of an object. Microsoft has not let the fact that it lacked a dynamic environment like NeXT's OpenStep stop it from promoting object technology via visions of an object-oriented operating system like Cairo.

Weaknesses

- In object technology purists' estimation, OLE COM is not truly object-oriented; it simply incorporates some object technology, but lacks inheritance. Microsoft contends that purity is the issue; OLE COM architects have purposely avoided inheritance because it has major negative implications in applications software development. They actually consider this a strength: Their object model grew out of the Microsoft application division's need for binary component reuse, rather than out of CORBA's Smalltalk and C++ language heritage.

- Microsoft is not perceived as a high end enterprise software vendor for OS or database technology.

Outlook

INPUT can conceive of no scenario under which OLE Controls and OLE Automation will not be dominant on the desktop until the next major paradigm shift. Because OLE Automation and OLE Controls need COM, COM is here to stay, as well. Every other object model will have to learn to live with it. Whether COM can expand its functionality to the infrastructure is often questioned, but experts say there is no technical restriction. COM's ability to scale well enough to manage distributed objects across wide-area networks and incorporate legacy data and applications is the remaining question; extensive benchmarking beginning in 1996 will begin to answer it. The success of OLE Documents and OLE Enterprise Interfaces depends on that benchmarking.

Microsoft's biggest weakness is its lack of enterprise software customization experience. Merging with or acquiring a major enterprise software company like Oracle would strengthen its enterprise position considerably. Until then, Microsoft will have to be content with the midrange enterprise market. Microsoft also needs to increase its NT presence on non-Intel platforms if it is going to make COM succeed across the enterprise. Digital, Sequent and HP are partners Microsoft must to promote more strongly in this effort.

15. NetLinks

Strategy

NetLinks' GUI-based IDL editor and browser product, called ORBitize, is one of the first tools that lets programmers create new IDL definitions without becoming expert in IDL syntax and the CORBA standard; it also allows developers to browse and edit existing IDL definitions in interface repositories. The edited code assumes existence of a CORBA-compliant ORB; ORBitize Version 1.1, released in September 1994, supports all CORBA 1.2 and CORBA 2.0 implementations including Orbix, SOM and ObjectBroker on Windows, Solaris, SunOS, Windows 95 and Windows NT. An important feature allows the developer to choose from one of two modes of editing: standard or extended. In standard mode, ORBitize enforces OMG IDL conventions; in extended mode, vendor-specific extensions—such as SOM's pointer data types—are permitted. As part of is consulting business, NetLinks has trained hundreds of distributed object developers and specializes in both CORBA and COM.

Strengths

- Netlinks' GUI-based tool simplifies CORBA-compliant programming for C++ programmers who do not wish to become experts in CORBA conventions.
- Wide-ranging experience with distributed object management

Weaknesses

- Limited platform support (e.g., no support for SOM on AIX)
- No current product support for OLE COM

Outlook

Like most tool suppliers (see Exhibit IV-3), NetLinks is not very dependent on acceptance of a DOM standard. Whether or not CORBA is accepted, ORBitize will work with the many products that currently follow the Object Management Architecture. These products will certainly continue to be marketed for some time and will require tools that simplify CORBA-compliant application development. Eventually, some compatibility with the Microsoft interface structure will be required, whether or not CORBA survives as a standard.

16. NeXT

Strategy

NeXT has been a long-time leader in the object-technology sector. In addition, under a joint development effort with SunSoft, NEXTSTEP will become OpenStep. The key to both is support for reuse and easy redefinition in enterprise application development. NeXT has demonstrated CORBA 2 compatibility. It has also demonstrated distributed object capabilities on Windows NT. The transition to more complete and nonproprietary distributed object management makes sense because the key competitor for OpenStep is Microsoft's OLE COM. At the COM level (as opposed to the OLE level), Sun and NeXT are Microsoft's most serious competitors. NeXT's Portable Distributed Object (PDO) environment supports deployment across HP-UX, Sun OS, Windows NT, Solaris, and Digital UNIX servers, as well as NeXT's own MachOS.

NeXT provides excellent database integration. It also has an unusually satisfied customer base, recently joined by Merrill-Lynch. If NeXT is successful with Sun then the entire battlefield will change. Instead of there being a two-sided fight with Microsoft battling IBM long term, Sun will become a third player.

Strengths

- PDO, by itself, provides a high level of portability.
- Enterprise Object Modeler creates a mapping structure that interfaces databases provides database independence from objects and supports of a variety of RDBMSs.
- Integrating object technology with the World Wide Web.

Weaknesses

- OpenStep is not as precise an interface specification as OLE COM (but is more rigorous than generic CORBA); it is an attempt by Sun and NeXT to rise above CORBA's least common denominator status that runs the risk of being perceived as neither fish nor fowl.

- It is hard to determine whether Sun is more dependent on NeXT or whether NeXT is more dependent on Sun, but the dependency is clearly a business-survival issue for NeXT, whereas Sun is simply in the object technology market to sell workstations and servers.

Outlook

NeXT has been vocal in expressing concern about both "committee-ware" standards and monopoly-driven standards. It feels the development community's collective wisdom will rule. NeXT will work hard to help it choose OpenStep.

As Exhibit IV-3 illustrates, NeXT covers both sides of the distributed object management standards movement. It is very dependent on the adoption of a standard to make its business model work. If NeXT's technology became widely accepted as an industry standard its business could expand dramatically. NeXT's tools do not support object technology without its ORB and ORB-based frameworks. NeXT is very dependent on Sun's market penetration with its technology as part of NEO to achieve market prominence. NeXT is in a risky position, but if Sun achieves as much market presence with NEO as it did with NFS, NeXT could fulfill its dream of becoming a major player.

17. Novell

Strategy

Novell is redefining its management and strategic direction in relation to OO technologies. It states that it is a firm supporter of OpenDoc (and an investor in CI Labs), but recognizes and is responding to the market demand for OLE-based components. It recently transferred development and marketing of the OpenDoc for Windows developers' kit to IBM. It believes OpenDoc acceptance awaits the next major paradigm shift in information systems, most likely something based on the Internet.

Novell sees its strengths as providing network services to all sizes of networks, despite the failure of NetWare to penetrate the UNIX market successfully. Object services, together with NetWare Directory Services (NDS) and the Tuxedo transaction monitor, which manage all types of objects, are Novell's future. These include fine-grained objects such as .OCXs or OpenDoc parts, as well as entire applications. They also include transaction processing and document management objects. Network management and directories are other services of interest to Novell.

Novell's NDS architecture will support storage of objects in 1996, just as it supports storage of user names. This will enable system administrators to manage objects with the network management tools they already use. Novell has a tremendous advantage over competitors in managing OCXs and other software components because, according to Novell, its software manages over 80% of Fortune 100 network nodes. Novell is working with SCO and HP to build gateways to DCE services.

Microsoft and now IBM with Lotus are formidable competitors to Novell. Another application area in which Novell could succeed in is groupware, with Collabra's software. However, Novell is now looking to the Internet as a major platform, and this may prove a more profitable route than investing too much in OpenDoc. NetScape's acquisition of Collabra gives Novell the opportunity to partner with the perceived leader in Internet tools and focus on the network.

Novell has succeeded by integrating the network at a lower layer than the object layer. By making networked computers operate as one, Novell simplifies the environment for distributed processing. It has many tools and technologies that are neither OLE-nor CORBA-based. It has traditionally competed with Sun's ONC+ on a distributed computing platform, but Sun has leaped ahead with object technologies to make the NEO environment. Novell, by contrast, has had less of a technology focus and more of a pragmatic services approach, adding network management, directory services, messaging support and transaction processing regardless of an elegant framework. Tuxedo is very different from the NetWare directory technology. However, both Sun and Microsoft can provide networking and object layers. To compete, Novell must either partner with strong distributed object technology vendors or continue to provide point solutions for its customers.

Strengths

- Large NetWare installed base
- Recently decided to refocus on network services
- Significant services and support infrastructure for its NetWare products
- Able to produce pragmatic solutions

Weaknesses

- Not a leader in object technology
- Fails to have a consistent architecture that can span both high and low-end platforms—this is largely an inability to persuade large systems vendors to adopt NetWare
- Unclear commitment to OpenDoc
- TCP/IP is the predominant network infrastructure, Novell offers this, but IPX/SPX is its preferred protocol. Novell has failed to capitalize on the opportunity to upgrade its installed base, in the way that Microsoft is persuading users to move to Windows '95. This has left opportunities for FTP Software and others to provide TCP/IP protocol stacks and thereby Internet connectivity. Novell is now playing catch-up with its own Internet strategy.
- Slow to react to market changes

Outlook

Novell is financially strong, with a large installed base and many useful network services. Its lack of a distributed object infrastructure is compensated for by its being open to different object architectures that it will support with its own network services. Its plan to leverage the NDS architecture to store enterprise objects is brilliant, providing it gets the cooperation of corporate systems managers to adopt its approach to storing objects. Novell will have systems administrators on its side, but to win over programmers who may have used Microsoft tools and have other ideas on how software components should be managed may prove more difficult.

Novell needs to build on its strengths of being able to simplify networking. Whereas it has managed to create a loyal following of

NetWare consultants and integrators, it has not yet managed to leverage this into a distribution channel for software components that can enhance networks.

Novell should consider acquiring some companies with object technology and Internet solutions to enhance its market position. It also needs to increase penetration in high-end systems by partnering more closely with systems suppliers.

18. Oracle

Strategy

Power Objects, Oracle's visual development tool, appears to be an important market entry as a competitor to Visual BASIC and similar tools. In addition, it may contain the seeds of Oracle's strategy progressively to incorporate object technology into its overall product line. Although Power Objects is limited to workgroup applications and the fat-client, two-tier model at present, that does not appear to be a technical restriction. Resulting components are OLE COM compliant and Oracle Power Objects 1.1 is expected to support OpenDoc. In addition, Oracle is working on an object framework, called Sedona, for 1996 or later delivery. It is also supposed to be OpenDoc and CORBA-compliant. Oracle has recently made a major investment in CI Labs. Oracle also has connectivity and systems management projects that can be expected to provide key components of both OLE COM and CORBA environments.

Strengths

- Complete visual development environment
- Optimized to support workgroup-level development
- Macintosh-Windows cross-platform compatibility

Weaknesses

- Not designed for enterprise-level development or processing
- Supports only two-tier database access

Outlook

Like other DBMS suppliers, including object-oriented DBMS suppliers, Oracle is ORB neutral and not at all dependent on whether one distributed object management standard or the other is accepted (see Exhibit IV-3). The company's stated intention is to incorporate object technology into its current systems software product line-up in an evolutionary manner, with the recently announced Power Objects probably serving as a technology indicator. From a marketing perspective, Power Objects was not simply evolutionary; for the first time, Oracle is selling development tools cross-database and is selling through two new channels: Apple dealers and the Internet.

19. ParcPlace-Digitalk Systems

Strategy

ParcPlace-Digitalk's Smalltalk-based VisualWorks is an object technology based OO development environment, comprised of a Database Applications Creator, a GUI builder, run-time database access software, and Smalltalk. The latest revision featured the RAD-oriented Database Application Creator, which in turn includes:

- ObjectLens, a tool to view and manipulate relational data as objects
- Visual Data Modeler, to map relational data to objects
- Reusable Dataforms components, intelligent objects that enhance interaction among relational data without SQL programming

It will be combined with Digitalk's Parts Workbench and team-development tools to provide a very complete application development environment. Digitalk has had a very different philosophy from that of ParcPlace. ParcPlace has chosen to make its software platform independent and portable. Digitalk has chosen to select features of each environment, like OLE support under Windows, and make each port of its software perform optimally for the environment.

ParcPlace-Digitalk's products are marketed through a variety of strategic relationships, primarily with HP, as well as directly and through more traditional distributors. Although the company's products have no particular ORB bias, when placed on top of a structure such as HP's Distributed Smalltalk, ORB relationships come into play.

Strengths

- Experienced object technology supplier of a quality product
- Leader in the Smalltalk development tools market
- Strong library of components for database access
- Experienced in RAD and software engineering

Weaknesses

- Encouraging acceptance of Smalltalk is an uphill battle versus C++ or COBOL
- Revenues slowed in 1995, partly because reseller relationships have not continued to grow at the pace of 1994

Outlook

As Exhibit IV-3 indicates, ParcPlace-Digitalk—like CenterLine from the C++ side—is especially well positioned to endure any prolonged industry indecision over an ORB standard, and—from a standards perspective is as independent of the DOM standards process as CenterLine. In its direction statement, the new company promised products that support both of the ORB standards. However, ParcPlace-Digitalk needs the object technology industry to advance dramatically in order to meet its stated goals, and that advancement may depend on the standards movement. For further assessment, see INPUT's Client/Server Vendor Profile on ParcPlace.

20. PostModern Computing

Strategy

PostModern claims that its ORBeline ORB product is the first available CORBA 2.0-compliant ORB and emphasizes this compliance in environments from Cray supercomputing platforms to embedded systems to Windows NT servers. PostModern is trying to use its technology lead to compete with Iona and Expersoft. ORBeline currently runs on SunOS, Solaris 2.4, SCO UNIX, and both Windows NT and Windows 95. The previous version of ORBeline ran on Cray Unicos, HP-UX, AIX, Digital UNIX and VxWorks, and ORBeline 2.0 may be ported to those platforms as well (but see limitation noted below).

PostModern also developed the NetClasses and NetClasses/Multicast C++ component libraries, which, together with ORBeline, support "publish and subscribe" and "multicast" connectionless broadcasting variations of the messaging interconnection technique. A dynamic directory service included within ORBeline supports these techniques and PostModern positions its products as especially key to fault-tolerant applications. It has recently been selected by Hitachi to provide the ORB underpinnings of a major object-technology-based transaction service. PostModern has also worked with power utilities to integrate software with their electricity distribution networks.

Strengths

- High-node-count WAN connectivity
- Advanced capabilities based on "smart" object binding to optimize choice of transport mechanism at run time
- Agile, entrepreneurial company

Weaknesses

- No direct OLE COM connectivity; will depend on third-party bridges or gateways
- Will need to develop a DCE-based GIOP version if market demand for that type of product grows

Outlook

ORBeline is the most CORBA-oriented of the ORB ISVs (see Exhibit IV-3) and—like Expersoft and Iona—works with a business model that depends on acceptance of DOM standards. Without such standards, it will be difficult for a relatively small ISV to maintain ports to multiple systems suppliers' and larger ISVs' object models. The robustness of PostModern's product offering is a plus. PostModern needs more strategic partners that can leverage its technology. Its ability to scale its system and penetrate new markets will be critical to its success and differentiation. It also needs to attract a VAR and the systems integration community.

21. SunSoft (a Sun Microsystems' Company)

Strategy

SunSoft has been working on its framework of object technology, called Distributed Objects Everywhere (DOE), since 1992. Prior to that time, it had a joint development agreement with HP, called Distributed Applications Architecture, that provided much of the underlying technology for CORBA 1.0, including the Interface Definition Language. Sun had extraordinary market penetration with its NFS technology that linked many competing UNIX systems and PCs. If recently renamed its DOE initiative NEO. NEO is integrating distributed object computing with the Internet via Sun's Java and other technology.

In June 1994, SunSoft announced a marketing/development arrangement with NeXT by which OpenStep would provide a distributed object infrastructure and application objects on Solaris. Sun's motivation for doing this was to provide a robust, innovative, scaleable environment for distributed applications. It provides implementation of CORBA plus OLE COM interconnectivity. Sun believes that this combination—along with IONA Technologies' contribution in providing a method to run on non-Solaris CORBA platforms—gives its customers the object technology version of Sun's open systems distributed networking strengths. Sun has made a significant minority investment in IONA. These strengths have been proven over time with such products as ONC/NFS. Sun is also working with Object Design to provide an OODBMS as a repository for its objects.

A first release of DOE, originally scheduled for the first half of 1995, was delayed. Sun's plan to be first to market with a mainstream distributed object environment failed; apparently, once it was clear that Sun would not be first, it decided to make sure it was right. Now NEO promises not only to enhance distributed computing, but to make Java applications compete with those programmed using OpenDoc standards. NEO enables client software that is independent of the server to access networked applications. For example, Java and NEO enable dynamic graphics and decision support applications to run across the Internet and internal networks. Sun is working with the academic community to get ideas, programmers and a ready customer base. It is intent on providing scaleable solutions for corporations that find PCs too unreliable and need easily administered servers. Sun is focusing NEO, like NeXT, on enterprise developers

who want to create applications for competitive advantage. They are positioning it as one of several development platforms. Java is a more important platform.

Strengths

- Early and ongoing contributions to the CORBA standards effort, including work convincing the OMG community in late 1994 that standard CORBA should be based on TCP/IP rather than on the DCE RPC mechanism
- Leadership in procedurally based distributed computing

Weaknesses

- Paper product
- Planned product is basically the work of a mini-consortium
- Some view OpenStep as proprietary, even though the OpenStep specification is publicly available and HP and Digital support OpenStep.
- Sun has not had widespread success in marketing Solaris independently of its hardware platform.

Outlook

As Exhibit IV-3 indicates, the position of Sun relative to standards bias is more neutral than any systems supplier except Digital. As a systems supplier, Sun's interest is in protecting and increasing systems market share. SunSoft wants its software to go beyond Sun platforms, but to do this it needs greater marketing efforts to Sun's competitors.

In the CORBA environment, Sun is more dependent on standards acceptance than any systems supplier except IBM and Apple because of its dependence on IONA and NeXT. The likelihood of its success is as cloudy as its product delivery schedule; only its previous success in the area of interconnecting disparate, dispersed systems can be considered a plus. Sun will bypass CORBA, OLE or OpenDoc. Instead of taking an architectural approach, it will stimulate application builders to use Java and thereby raise the playing field to a new level.

22. Teknekron Software Systems

Strategy

Teknekron is a software development tool supplier specializing in helping customers build real-time, event-driven applications such as stock market information feeds. It holds patents on publish-and-subscribe methods, an emerging type of middleware especially important to overcoming some of the timing restrictions inherent in messaging. Teknekron's evolution to a tool-set provider—from its position as a systems integrator in trading applications—is based on the strength of its Teknekron Information Bus (TIB), a communications platform that supports a broadcast mode of interconnection. Object technology is especially suited to implementing "publish and subscribe" applications and broadcast techniques. TIB now incorporates the Teknekron CORBA 2.0-compliant ObjectBus as a backbone and supports the more conventional bi-directional client/server mode of communications among objects as well.

Strengths

- Provides dynamic "publish and subscribe" information exchange methodology that scales well to large enterprises, particularly in financial trading applications.
- TIB provides a base for an enterprise development toolkit, including the CORBA-compliant Remote Object Framework, a data modeling environment and a method of incorporating object-oriented programs and legacy applications on the same platform.
- Reuters financial ownership

Weaknesses

- Teknekron's recent entry to the market as a tools supplier will be hard to overcome and will hamper distribution of the technology
- Dependent on other suppliers to provide gateways and/or bridges that support OLE-compliant interaction at a performance level to support real-time applications
- Understands applications niches better than horizontal markets.

Outlook

Teknekron's position is one of the "most dependent on a DOM standard" and among the most "CORBA biased" (see Exhibit IV-3). (This is in no way a comment on Teknekron's position and long-term outlook as a systems integrator, only as a provider of distributed object management technology.) It has chosen to compete head on with three companies (Iona, Expersoft and PostModern) that are completely dedicated to selling this technology, unencumbered with running another business. Whether Teknekron's technology lead in the "publish and subscribe" methodology and built-in market of its existing customer base serves to overcome these limitations remains to be seen.

23. Texas Instruments

Strategy

Texas Instruments and Microsoft are jointly developing a new repository technology that can be used by both software developers and technically knowledgeable users. It will store application components like presentations, charts and spreadsheets, as well as fine-grained programmer components.

TI's repository technology is based on components and will be layered for scalability. The result will likely be two separate products from the two companies. TI currently markets the Composer By IEF repository-based application development tool, which evolved from its Information Engineering Facility CASE tools. Composer supports GUIs for event-driven processing, heterogeneous client/server interaction, multiple development approaches, isolation from lower level networking programming, transaction services, and inclusion of third-party development tools. TI also offers a complementary Arranger product line that supports developers in combining repository components. This is particularly useful to application developers wanting to create enterprise solutions from software that has already been deployed and tested.

Strengths

- Will combine the market and development strengths of leaders in their respective areas
- Component software development needs the integrated approach that TI has brought to procedural model of programming for over 10 years

- TI has built its strategy around pragmatic components found on Windows systems, rather than special class libraries that are proprietary. The components may be files, software libraries, documents.
- TI leverages the Windows installed base.

Weaknesses

- Not yet clear that OLE technology can scale to enterprise-wide requirements
- No native OpenDoc/CORBA connectivity, which more applicability to enterprise-level software development

Outlook

In Input's opinion, TI's efforts to build an object-based development environment in association with Microsoft is typical of an object technology supplier that is very biased to a particular DOM standard but relatively unconcerned about whether one or more particular standards is adopted. Like Antares relationship to CORBA and the many C++ and Smalltalk compiler suppliers in the market, TI's emphasis is on ease of program development rather than run-time implications. If OLE were to disappear tomorrow, TI could add to the development effort to include a proprietary ORB mechanism or to adopt CORBA. That flexibility, plus its past leadership in meeting this market need, should bode well for the next generation of IEF/Composer.

In the past, TI has been criticized for developing large inflexible systems, appropriate for Fortune 500 systems architects, but unusable by the average developer. Composer and Arranger are useful tools for the Windows programmer, who may be in a much smaller corporation. TI's challenge is to scale its pricing and make its software modular so that it can offer tools for the professional programmer, the casual programmer and the technically knowledgeable user. It also needs to broaden its distribution with a strong VAR program.

24. Visual Edge

Strategy

Visual Edge of St. Laurent, Quebec, best known for its Motif-based GUI builder development tools, has recently announced Object Bridge

software. Technically, Object Bridge is a class registry that describes the supported object systems (OLE COM, Orbix and SOM, currently) and object system adapters developed by Visual Edge and its licensees. Object Bridge provides bridging technology for CORBA- and OLE COM-compliant objects, and can support other object models such as NeXT's PDO. Functionally, it is similar to the CORBA-COM bridge that Digital plans to provide, but Object Bridge uses a custom conversion engine rather than using a least-common-denominator protocol, wrappering or dynamically invoked converter. Planned for release late in 1995, Object Bridge has already been licensed by Apple, IBM, Iona, Oracle and Taligent.

Strengths

- Chosen as a development tool by leading ORB vendors, including some with competitive products
- Available for OEMs (such as the software suppliers mentioned above) and ISVs, as well as for corporate IS departments

Weaknesses

- Inherently lower performance as a trade-off for transparency
- May be competing against Microsoft's "object middleware" approach, which—although it may not be as technically elegant—will have major marketing power behind it

Outlook

Visual Edge will market Object Bridge using primarily the same business model it used with its GUI builder: indirect sales through systems suppliers and other ISVs. They will choose such a tool because they do not consider bridges and gateway core technology that they have to develop themselves. To be successful with Object Bridge, Visual Edge must convince the market of the superiority of the conversion-engine approach to bridging foreign systems. If successful, it will have little dependence on the acceptance of a DOM standard or need to lean toward either CORBA or COM compliance (see Exhibit IV-3).

C
Vendor Comparison

Exhibit IV-3 positions object technology vendors relative to:

- Their orientation to competing distributed-object interoperability standards, CORBA and OLE COM (horizontal axis). In the CORBA half of the diagram, OpenDoc followers are also included.
- Each supplier's dependence on the acceptance of *any* distributed-object-management standard (vertical axis) for the success of its related product strategy, as described in section IV-B

Note that no supplier is dependent on these orientations or standards acceptance for the success of current products; these positions refer to trends affecting products to be initially released or substantially upgraded beginning in late 1995 through 1997.

1. Least Dependent on Industry Standards

Antares Alliance Group is one of the suppliers least dependent on acceptance of CORBA—or any standard—to establish Huron ObjectStar's market share. Antares' Huron ObjectStar product is based on a CORBA-compliant ORB. If CORBA were to disappear, Huron could be migrated easily to another distributed-object-management approach; whether that approach was an industry standard is irrelevant. Similarly, Texas Instruments (TI) is very dependent on COM technology in the architecture of the CASE repository that it is developing jointly with Microsoft. However, if COM were to fail to gain market recognition (or fail to deliver on market promises), TI could fairly easily move to CORBA or a non-standard method.

2. Most Dependent on Industry Standards

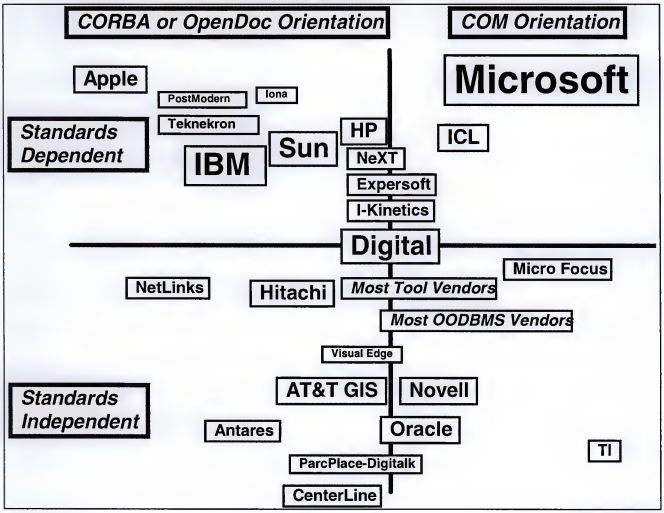
Apple and Microsoft represent the other extreme. The two are completely dependent on distributed-object-management standards acceptance for the success of the component software product plans that they have previewed over the last two years. Apple and Microsoft have indicated that this distributed object technology will be key. Furthermore:

- Microsoft is biased toward (as well as very dependent on) OLE COM.

- Apple's bias is less easy to position; it earns a CORBA bias due to its support of OpenDoc and OpenDoc's dependence on DSOM (however, this assumes IBM's unwavering support of CORBA and continued support of CI Labs).

Exhibit IV-3

Vendor Positioning



Source: INPUT

3. Partially Dependent on Industry Standards

The other vendors fall between these extremes, with Digital holding the enviable (or possibly insupportable) position of having a foot in all camps. Most tool suppliers, with the exception of Micro Focus, typically provide products that support both standards or that are standards-neutral; MicroFocus currently markets only an OLE COM-compliant compiler. Note that AT&T GIS is basically a tool/component supplier in the object technology market; it abdicated

NCR's early leadership in distributed object management, based on a product called COOPERATION.

Language suppliers show little bias to any distributed-object-management standard, and little dependence on the acceptance of a standard. Object-oriented DBMS suppliers follow suit. Oracle Power Objects—introduced in July 1995—typifies that strategy and appears to be a technology that Oracle can migrate to the very top of its DBMS product line at a pace of the company's choosing.

D
Standards Bodies and Associations

The main consortia and associations that cover component software are described below.

1. ComponentWare Consortium (CWC)

The founders are, not coincidentally, the group working under I-Kinetics general management on a large logistics system for the U.S. Navy. The group encourages three levels of actual membership:

- Technology providers that develop underlying infrastructure (e.g., IONA)
- Solution providers that plan and implement component-based solutions (e.g., BBN's Internet Services)
- Strategic partners that use resulting componentware

Associate memberships from academia are also encouraged. CWC promotes an object-oriented systems engineering methodology called the Information Factory methodology and members commit to using the methodology for a two-year period on key projects. The key CWC deliverables are what it calls "ready to run" components. Such components are promised for:

- Data management—FAME, Informix, ObjectStore, Oracle and Sybase
- Mechanical engineering analysis—Finite element analysis and computational fluid dynamics applications
- Logistics—CALS and regional maintenance applications

- Desktop applications—Excel, Lotus 1-2-3, Quatro, Access, AutoCAD and Word
- Development software—Visual Basic and PowerBuilder

2. Component Integration Laboratories (CI Labs)

CI Labs is a non-profit industry consortium run primarily by Apple, IBM and Novell. Oracle and Adobe have made major investments, termed sponsorships, that provide a seat on the Board of Directors, joining the three suppliers mentioned above. Other categories of membership include:

- Full—participate in all aspects of technology evaluation and setting of development priorities
- Associate—participate in relevant workgroups and taskforces; necessary to license OpenDoc source code
- Subscriber—receive information service on the above activity

In addition to the founders that are not sponsors, other full members include the OMG (see below), Lotus, and the X-consortium.

3. Object Database Management Group (ODMG)

The ODMG is modeled after the database industry's SQL standards organization, the SQL Access Group (SAG). Its relationship to OMG is analogous to SAG's subsidiary relationship to X/Open. Voting members of the ODMG include the major object-oriented database companies, i.e., Objectivity, Object Design, Ontos, O2, Poet, Servio and Versant. HP, Intellitic, Itasca, Micram, Persistence, Sybase and TI act as reviewers. Its current standard, ODMG-93, includes:

- a strict superset of the CORBA Interface Definition Language
- an object query language based on SQL SELECT
- C++ and Smalltalk bindings

It is a "standard extension" to the OMG's Persistent Object service. It is an extension in that OMG does not specify APIs as a matter of policy. Basic functionality includes the definition, creation, modification and sharing of objects; plus transaction, relationship and

collection services. Versioning and meta-data access services may be included in future versions of the standard.

4. Object Management Group (OMG)

OMG consists of over 500 systems suppliers, ISVs, and corporate members working with distributed object technology. Its charter is to standardize on and promote detailed object technology-based specifications for distributed computing. It is a very active and dynamic group with strong public relations to promote distributed object computing.

The OMG focuses on promoting:

- a single terminology for object orientation
- a common abstract framework
- a common reference model
- common interfaces and protocols

OMG members have now begun to follow these very flexible specifications—including the CORBA specification—in the object-oriented applications they build and buy. The OMG standards-adoption process is competitive. Rather than sitting around a table designing a specification, companies compete for member approval with specifications that have already been implemented. The OMG technical committees and board of directors review individual specifications using the Object Management Architecture (OMA) as a reference point. The OMA defines the interrelationships of various object technologies as they will ideally exist across heterogeneous, internetworked systems. The first languages to be supported are C, C++ and Smalltalk, followed by COBOL and Ada. Transaction processing and security are other areas of active interest. Interoperability between CORBA and COM will be standardized in two stages. In the fourth quarter of 1995 the OMG expects to standardize OLE 2 to CORBA interfaces and in second quarter of 1996 COM-to-CORBA interfaces will be standardized. Other directions that the OMG is taking (with partners' names in brackets) are:

- User requirements (GPT)
- OODBMS interfaces (Sybase)

- OO data analysis standards (ICL)
- Security (MITRE)
- Telecommunications (Bell Northern Research)
- Manufacturing (Sematech)
- Finance (Stanford Software)
- Health services (Decision Support Group)
- Business objects (Open Engineering)

OMG promotes its agenda through the bicoastal Object World trade events, the Internet, seminars and white papers.

E
Other Software Vendors and Systems Integrators

The packaged software vendor community has long recognized the value of managing modular code. Objects are an extension of that heritage that enables manageable software components to be created. An open question is how software components will be marketed and distributed. Some believe that a cottage industry of small developers will emerge that can create specialized components. This has already happened in the Visual Basic and NeXT communities. In the C++ community, a few vendors, like Rogue Wave Software (Corvallis, OR), have seen success in creating software libraries.

In the CORBA-compliant world, major corporations, systems integrators and leading software vendors will be the main contributors of software components. Vendors of databases, software development tools and applications will also supply system components. User organizations like American Airlines, American Express, ARCO, BASF, Charles Schwab, Chase Manhattan Bank, Deere & Company, Nippon Steel and the Royal Bank of Canada show their interest in CORBA standards by belonging to the OMG. The OMG has also managed to attract the world's leading telephone companies and the major systems integrators. Vendors that can use ORBs in copiers, printers and other devices are also members of the OMG. Although there are smaller vendors in the OMG, most of the members are established enterprises.

In the OLE world, there is more support from packaged software vendors. PeopleSoft already embeds Excel spreadsheets in its human resources applications. SAS Institute has recently announced the SAS System for Application Development, which enables OLE objects to be embedded in SAS applications. Data entry and browsing based on OLE technology can be customized by the user. There will be many small vendors of components based on OLE technology, as well as large ones.

The OpenDoc environment has failed to attract many large software vendors, IBM, Novell and Apple being its main supporters. It has attracted innovative developers, many, like database vendor ACI, already established on the Apple Macintosh platform. Apple's Claris division that markets MacDraw, MacWrite and FileMaker has the best chance of gaining market share with OpenDoc applications. On the OS/2 platform IBM needs to encourage developers to build creative OpenDoc components and packages. Given IBM's struggle with OS/2 it is unclear whether IBM has the agility to drive the OpenDoc applications market. IBM, like Sun, may find Internet tools more widely accepted. OpenDoc vendors can also support OLE. OpenDoc, like the Macintosh, will attract innovators, but the majority of software vendors will bypass it, feeling that OLE is adequate. Vendors that want to support non-Windows platforms are the ones most likely to gravitate to OLE.

(Blank)

Market Forecasts

This chapter provides market forecasts for the U.S. and worldwide. The forecasts are for components, application packages and enterprise systems that take advantage of ORBs, OLE or OpenDoc technology.

A
What Do the Forecasts Show?

A software package may be both a standalone application and an application component. This presents a challenge in forecasting the market for components. Forecasts are made for:

- component software, used by programmers
- development tools, used by programmers or users
- applications packages that are made from or can be connected to interchangeable components
- enterprise software that uses or can be extended by using components

The forecasts are divided into those that represent expenditures by:

- developers on components
- users and developers on application development tools, packages and enterprise software

The forecasts show that the OpenDoc market is very vulnerable to being overtaken by OLE standards. The main platforms for OpenDoc will be OS/2 and the MacOS. It will be IBM, with OS/2, that determines the success or failure of OpenDoc as much as Novell or Apple. Much depends on the success of Microsoft porting OLE

software libraries and development tools to platforms other than the MacOS or Windows. When Windows was first announced it was touted as the cross-platform windowing environment. Even Microsoft Word ran on UNIX in an early incarnation, but the promise of Microsoft Windows and related Microsoft applications packages running across multiple operating systems was never realized. The same could happen with OLE. Microsoft may have the best intentions to make it cross-platform, but in reality, the Windows platform will be the most profitable for Microsoft to invest in and other platforms risk being neglected.

Enterprise systems supporting CORBA will evolve and the server market will offer potential for developers wanting to support shared object environments. The desktop market will largely be an OLE Microsoft market.

Software that does not support the OLE, OpenDoc or CORBA standards includes older packages for DOS, AS/400 and other environments. Software that has its own communications mechanisms, with support for standards like SNA, is not included in the forecasts. Also, specialized software for stand alone applications and those on non standard environments does not support the standards. A plethora of Internet applications may only support OLE and OpenDoc through browser software.

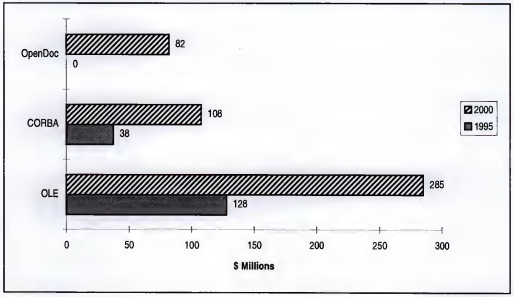
B
Programmer Expenditures on Component Software

The component software forecast will be restricted to components that could not run without being embedded in an application. These include the fine-grained components for developers and application components for application programmers in Exhibit III-1.

In this forecast, the user is usually the programmer, who may make another product to resell with the components. Programmers may be in-house corporate programmers, employees of systems and software vendors or independent consultants. This deviates from INPUT's normal practice of forecasting markets by looking at user expenditures to avoid double counting.

Exhibit V-1

Programmer Expenditures on Component Software—U.S., 1995-2000

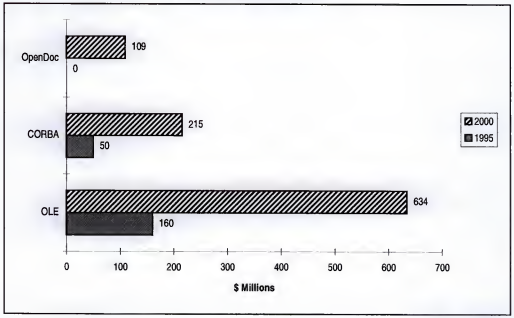


Source: INPUT

The reason for forecasting sales of components to programmers is so that vendors and investors can understand the market potential. The U.S. forecast for component software is shown in Exhibit V-1 and the worldwide forecast in Exhibit V-2.

Exhibit V-2

Programmer Expenditures on Component Software Worldwide, 1995-2000



Source: INPUT

Programmer expenditures form a high proportion of expenditures on software that supports component software as the market is developing. Over time, the ratio of programmer expenditures to user expenditures will fall. The above forecast is only for components that programmers would use, like software libraries. Application and enterprise components are not included here, but in 1996 they are likely to account for over $1 billion in revenues to systems integrators, software companies and value-added resellers.

C
User Expenditures on Software Incorporating Component Standards

There is considerable opportunity for creating tools that enable users to integrate components into systems. Hence, forecasts for application development tools expenditures by either users or developers are given.

For application packages, the forecast estimates the value of shrink-wrapped applications purchased by business users that are created from software components or can easily be integrated into another application because of their support for ORBs, OLE or OpenDoc. A monolithic application programmed as a .EXE file under Microsoft Windows is likely to support OLE, in that it can be integrated into another application, but it may not take advantage of the latest OLE 2 technology. This type of application will not be included in the forecast. Applications that will be included in the forecast are those like Microsoft Word, which enables components to be incorporated readily in documents using OLE. These forecasts are intended to help software library developers and tool vendors gauge the likely acceptance of their technology. The forecasts also help software vendors decide on their priorities and investment levels in a particular technology.

The third forecast estimates the value of enterprise software, like SAP's R/3 client/server integrated finance, manufacturing and human resources application, that can be extended or integrated using CORBA, OLE or OpenDoc standards. This forecast is intended to help systems integrators and enterprise software vendors to determine how much to invest in component software technologies. More details on the forecasts regarding operating systems are provided in Appendix C.

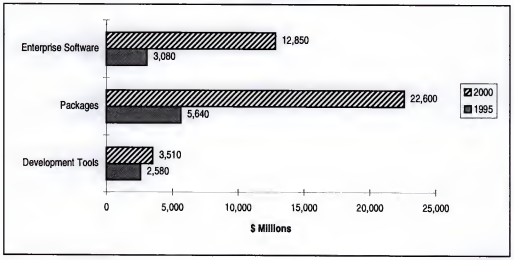
Exhibit V-3 shows U.S. user expenditures on application development tools, packages and enterprise software. Enterprise software will

increasingly be built from smaller components, hence the emphasis on packages, which, by the year 2000, are expected to form some of the building blocks of enterprise systems. The U.S. market is expected to lead the world, but over time will have less of an influence as technology is disseminated. Also sales of development tools outside the U.S. will increase as offshore development gains in popularity.

For enterprise technology many large U.S. corporations will choose to purchase software on global contracts in the U.S. and the initial deployment will often reflect U.S. expenditures. The enterprise market has a high percentage of installations in the U.S. initially.

Exhibit V-3

User Expenditures on Software Using Component Technology
U.S., 1995-2000

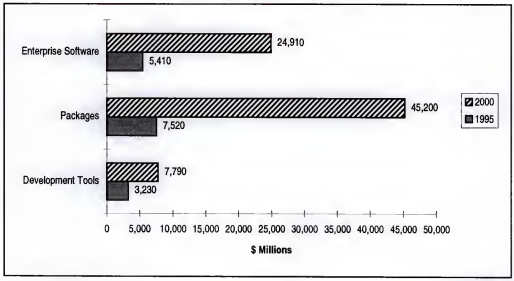


Source: INPUT

Exhibit V-4 shows the corresponding worldwide forecasts.

Exhibit V-4

User Expenditures on Software Using Component Technology Worldwide, 1995-2000



Source: INPUT

As in the U.S. packaged software sales will dominate the market, but there will still be healthy revenues in enterprise software. Of the $170 billion worldwide software market estimated for the year 2000, almost half (46%) is expected to support OLE, OpenDoc or CORBA.

Definitions

This appendix provides definitions of vocabulary used in the report. More definitions can be found in INPUT's *Definition of Terms*.

Agent	An agent is a set of instructions that can carry out tasks automatically. It is usually written in a high-level language script and may run across a network to send messages or find information.
Alert	An error message sent to a host computer or workstation when a problem is detected.
API	Application programming interface—this provides specifications for programmers.
Client	When used in client/server, refers to the computer platform accessed by a user, such as a PC, workstation or PDA.
Client process	A process object initiating requests on another object. At the programming level, clients look at ORBs and object implementations through the perspective of a language mapping.
Client program	Any program initiating requests on another object. At the programming level, clients look at ORBs and object implementations through the perspective of a language mapping.
Component	Component refers to software component, a piece of software with documented interfaces that a programmer can use to build an application.
CORBA	Common Object Request Broker Architecture.

CORBA compliance

In compliance with the CORBA standards as defined by the OMG. To be determined as part of the X/Open branding program and a related OMG program called "Profiling."

CORBA 1 compliance

Objects on different platforms and networks can communicate across various platforms transparently using a single brand of ORB.

CORBA 2 compliance

Intended to enable objects on different platforms and networks to communicate transparently using different brands of ORBs. CORBA 2 accomplishes this by providing for two ORB interoperability schemes based on two different gateway message formats, called the General Inter-ORB Protocol (GIOP) and the Environment-Specific Inter-ORB Protocol (ESIOP).

Development Environment

Set of software used by applications that typically consists of programmers for developing applications that typically consists of compilers, debuggers, visual editors, profilers and performance optimizers.

Development Tools Short for "application development tools."

Distributed System A system that runs across multiple computers.

DLL Dynamic Linked Library, a software component of pre compiled code that can be linked into an application.

Dynamic Invoke Interface

Particular applications will work best if an object invocation is constructed at execution time, supplying information about the operation to be

performed and the types of parameters being passed. The dynamic invoke interface provides this capability.

Environment-Specific Inter-ORB Protocol (ESIOP)

The first ESIOP gateway implementation defined is based on the OSF's Distributed Computing Environment (DCE), and others are likely for COM, Open Network Computing (ONC), etc. They would also use TCP/IP, OSI, IPX/SPX or another transport protocol, whose details would be masked by the higher-level interface.

Framework — A specification or implementation of software that can be used to build an application. It may consist of classes and methods. Motif and the Common Object Request Broker Architecture (CORBA) are examples of frameworks.

Gateway — Software that connects one environment to another. It often translates formats and routes code from one application to another.

General Inter-ORB Protocol (GIOP)

The first GIOP gateway implementation is called the Internet Inter-ORB Protocol (IIOP) and is based on TCP/IP; IIOP is mandatory for CORBA 2, but it can be achieved by a native TCP/IP implementation as well as by a presumably less efficient bridge (i.e., in theory, it is not necessary to deploy TCP/IP stacks on every node). Other GIOPs will probably be developed to run on top of OSI or NetWare.

GUI — Graphical User Interface—a windowing system like Microsoft Windows or X-Windows with Motif that displays graphical objects on a display.

Implementation Repository

Lets the ORB locate and activate object implementations. Also can store additional information associated with implementation, such

	as debugging information, administrative control, resource allocation, security, etc.
Interface Definition Language (IDL)	Defines object types by specifying their interfaces, a set of named operations, and the parameters of those operations.
Interface Repository	Provides persistent objects that represent the IDL information in a form available at run time. Lets a program determine what operations are valid on an object when it encounters an object with an unknown interface, and invoke that object.
Language Mapping	Provides flexibility in defining language-specific data types and procedure interfaces to access objects through the ORB. Different programming languages, both object-oriented and non-object-oriented, may prefer to access CORBA (or CORBA-compliant) objects in different ways, but the standard says that a particular language mapping to CORBA should be the same for all ORB implementations.
MacOS	The operating system for the Apple Macintosh.
Messaging	A general term that describes communication that stores and forwards information. It may also support queues of objects waiting for an event in a network. An example of messaging software is electronic mail or software that supports on-line information services.
Networked ORBs	Provide some set or sets of language mappings and support multinode operation via the IIOP, another GIOP, or an ESIOP. IIOP support based on TCP/IP is mandatory for an ORB to be termed CORBA compliant.

Object Management Architecture (OMA)

OMG architecture that specifies object language interfaces, Common Object Services Specifications (COSS), Common Facilities and a Common Object Request Broker Architecture (CORBA) specifications.

Object Request Broker (ORB)

OMG terminology for the message-based communications interface between objects; an ORB provides the mechanism by which objects transparently make requests of, and receive responses from, other objects; the term has become commonly accepted, but not all products that perform these functions are called ORBs and not all ORBs meet OMG specifications.

OO	Object-oriented
Open systems	In this report, it describes systems that can run on multiple UNIX and/or Windows operating systems, rather than proprietary environments like VMS (even Open VMS) or MVS (even with POSIX compatibility).

Operating environment

Modern term for an operating system plus its application development tools.

ORB Interface	No matter which of the above interfaces makes sense in certain applications, basic operations, such as the operation that returns an object interface type, are common to all objects and are handled directly by the core via the ORB interface. There would probably be few of these common operations, but the standard's flexibility leaves that up to the development process.
OS	Operating system.
Platform	This is the software or hardware that an application program runs on.

POSIX	A standard for operating systems to ensure some level of portability of software code that runs on it. Standards are published by X/Open.
Program	The term is meant to include a wide range of possible constructs, including scripts, loadable modules, etc. in addition to the traditional definition of an application or utility.
Standalone ORBs	Function on single nodes with some set or sets of language mappings and gateway-oriented interoperability, if required.
Suites	Sets of applications or packages. Office suites typically consist of a word processor, a spreadsheet and a database or electronic mail package.
Visual Development Tool	This is the software needed to build an application. It may include a visual editor, a forms designer, a report writer, a compiler, an interpreter, a debugger or a source code control system that enables programmers to share coding tasks.
Windows	Used in this report to refer to Microsoft's Windows if it starts with a capital letter. If it starts with a small letter, it may refer to any software that controls the windows on a computer screen. A window may also be the window seen on a computer screen.

B

Vendor Names and Addresses

This appendix provides names and addresses of vendors and organizations mentioned in the report.

A
Names and Addresses of Vendors

Exhibit B-1

Names and Addresses of Vendors

Company	Notes
Antares Alliance Group 17304 Preston Road, Suite 1200 Dallas, TX 75252 Tel: 214-447-5500 Fax: 214-447-5783	Exhibits good understanding of applying object technology in enterprise-wide applications. Antares Alliance does not provide low-end products that can be easily deployed for pilot testing object technology.
AT&T Global Information Solutions 1700 S. Patterson Boulevard Dayton, OH 45479 Tel: 513-445-5000 Fax: 513-445-1238	AT&T GIS did not choose to emphasize NCR's early lead in object technology after the acquisition. Whether that changes after the pending divestiture remains to be seen.
CenterLine 10 Fawcett Street Cambridge, MA 02138 Tel: 617-498-3000 Fax: 617-868-6655	Leading customers of this eight year-old ISV include Novell, Oracle, Sybase, Apple, Digital, HP and Sun. It uses a direct salesforce in the U.S. and distributors in the rest of the world.
Digital Equipment Corporation 110 Spitbrook Road Nashua, NH 03062 Tel: 603-881-1894 Fax:: 603-881-2790	Digital has a complete line-up of object technology-based products to support both its OpenVMS and UNIX platforms, revolving around its ObjectBroker product, one of the original technologies incorporated in CORBA.
Expersoft 620 Mead Ridge Road San Diego, CA 92121 Tel: 619-546-4100 Fax:: 619-546-4110	Major customers include Andersen Consulting, Chemical Bank, Goldman Sachs, Raytheon, Sprint and U.S. West. Has many strategic relationships with leading systems suppliers.

Exhibit B-1 (cont'd)

Company	Notes
Hewlett-Packard 19310 Pruneridge Avenue Cupertino, CA 95014 Tel: 408-447-4042 Fax: 408-447-5809	HP has a complete line-up of object technology-based products to support its HP-UX platforms, but has not yet announced its ORB technology, called HP ORBplus.
Hitachi Computer Products (America) 437 Madison Ave. 21st Floor New York, NY 10022 Tel: 212-751-6302 Fax: 212-751-6368	This division of a leading technology supplier, Hitachi, has begun to market its OT capability with a concentration on its application in OLTP.
IBM Corporation 1 Old Orchard Rd. Armonk, NY 10504 Tel: 914-765-1900 Fax: 914-765-4190	IBM was an early leader in object technology implemented into the OS/400 operating software. SOM debuted with OS/2 and has been or is being ported to all IBM operating environments.
ICL Enterprises North America 11490 Commerce Park Drive Reston, VA 22091 Tel: 703-648-3300 Fax: 703-648-3380	ICL is a leading European hardware and systems software supplier that has taken an early position in object-based products.
I-Kinetics 1 New England Executive Park Burlington, MA 01803 Tel: 617-270-1336 Fax: 617-270-4979	I-Kinetics is the leader of the ComponentWare Consortium (CWC). As such, its strategy is built around partnerships (including relationships with BB&N, Digital, IONA, and others). I-Kinetics markets gateways and bridges to support the consortium's plans.
IONA Technologies Inc. 55 Fairbanks Blvd. Marlboro, MA 01752 (Corporate Headquarters in Dublin) Tel: 508-460-6868 Fax: 508-481-6099	Major customers for its flagship Orbix product—first introduced in July 1993—include Boeing, Motorola (Iridium Project), and Chemical Bank. IONA uses both a direct sales force and distributors worldwide.
Micro Focus 2465 East Bayshore Rd. Palo Alto, CA 94303 Tel: 415-856-4161 Fax: 415-856-6134	Micro Focus has been the market leader for open-system COBOL compilers for more than a decade. As a natural follow-on, it is among the first to bring an object-oriented COBOL compiler to market to compete with Smalltalk and C++ as a language system for OO development.
Microsoft Corporation 1 Microsoft Way Redmond, WA 98052 Tel: 206-882-8080 Fax: 206-936-7329	Leader of the desktop environment with Windows and the OLE 2 standard for interconnecting applications. Extending its technology into the enterprise server arena is Microsoft's challenge.
NetLinks Technology Inc. 74 Northeastern Blvd. ,Suite 8B Nashua, NH 03062 Tel: 603-594-0525 Fax: 603-594-0527	NetLinks is an independent consulting and training firm that has productized an IDL editor and browser originally developed to support its consulting business.

Exhibit B-1 (cont'd)

Company	Notes
NeXT Computer Inc. 900 Chesapeake Dr. Redwood City, CA 94063 Tel: 415-366-0900 Fax: 415-780-3714	NeXT offers its Portable Distributed Objects (PDO) object model and the Enterprise Objects Framework products under the banner of NEXTSTEP, a cross-platform development and run-time system for three-tier applications.
Novell 122 E. 1700 South Provo, UT 84606 Tel: 801-429-7000 Fax: 801-377-9353	Novell wants to become the leading supplier of services, such as directory and security services, for distributed object environments.
ParcPlace Systems Inc. 999 E. Arques Ave. Sunnyvale, CA 94086 Tel: 408-481-9090 Fax: 408-481-9095	The August 1995 merger of ParcPlace and Digitalk combines two companies with synergistic OT product capabilities on UNIX and Windows systems, respectively.
Oracle Corp. 500 Oracle Parkway Redwood Shores, CA 94065 Tel: 415 506 7000 Fax: 415 506 7200	Oracle has recently taken its first public steps into the OT arena with its cross-database, cross-platform Power Object development tool for workgroup applications.
PostModern Computing 1885 Landings Drive. Mountain View, CA 94043 Tel: 415-967-6169 Fax: 415-967-6212	PostModern was founded in 1991 to focus on communications tools such as ORBs in a distributed object environment. Major customers include MCI, First Pacific, Skytel/Mtel and Fuji Capital.
SunSoft 2550 Garcia Avenue Mountain View, CA 94043-1100 Tel: 415-960-3200 Fax: 415-336-0362	SunSoft has a leading position in distributed UNIX computing that it can leverage.
Teknekron Software Systems 530 Lytton Ave. Suite 301 Palo Alto, CA 94301 Tel: 415-325-1025 Fax: 415-321-3176	A division of Reuters. Over 150 customers, becoming a leading distributed platform for the development of trading applications. Major customers include Chase, Credit Suisse, and Nomura Products; also installed at Bechtel, Chevron Sony and VLSI Technology; working with Sematech to validate the technology for real-time process control. Working with Illustra for ORDBMS.
Texas Instruments 6620 Chase Oaks Blvd. Plano, TX 75023 Tel: 800-838-1843 Fax: 303-294-0930	Texas Instruments, long-time developers and marketers of repository-based team-development environments, announced in 1994 a joint program with Microsoft to build a repository-based application-development environment layered on top of COM and OLE technology.
Visual Edge Software Ltd. 3950 Cote Vertu St-Laurent, Quebec H4R 1V4 Tel: 514-332-6430 Fax: 514-332-5914	Visual Edge markets both its GUI builder and object-middleware technology through indirect channels such as systems suppliers and other ISVs. IBM, Iona, Oracle and Taligent have already licensed the Object Bridge product.

B

Names and Addresses of Organizations

This section lists the leading consortia and standards organizations.

Exhibit B-2

Names and Addresses of Organizations

Company	Notes
Component Integration Laboratories, Inc. P.O. Box 61747 Sunnyvale, CA 94008 Tel: 408-864-0300 Fax: 408-864-0380	CI Labs is a nonprofit organization founded by IBM, Apple, Novell, WordPerfect, SunSoft, Taligent and the XSoft division of Xerox to develop and promote the OpenDoc architecture.
ComponentWare Consortium c/o I-Kinetics 1 New England Executive Park Burlington, MA 01803 Tel: 617-270-1336 Fax: 617-270-4979	CWC is a group founded by I-Kinetics, NetLinks, IONA, Heuristicrats Research, Bolt Beranek and Newman's BBN Systems and Technologies division and Pratt & Whitney to promote the application of component software in mission-critical applications.
The Object Management Group 492 Old Connecticut Path Framingham, MA 01701 Tel: 508-820-4300. http://www.omg.org	The OMG was founded in 1989 by a core group that included Hewlett-Packard, Data General, American Airlines, and 10 other companies specifically interested in promoting object technology standards. It has a close working relationship with X/Open. It has also managed to attract major user organizations, telecommunications vendors and systems integrators.
Object Database Management Group 13504 Clinton Place Burnsville, MN 55337 Tel: 612-953-7250 Fax: 612-397-7203	The ODMG was formed to create one standard interface that all object-oriented database vendors could support. The objective is application portability, as opposed to interoperability.

C

Forecast Details

This appendix provides other INPUT forecasts here for comparison; as the detailed tables are in the report, they are not repeated here.

A
INPUT Software Forecasts

Exhibit C-1 shows INPUT's standard forecasts and growth rates for systems and applications software. As the 1995-2000 forecasts had not been completed as this report went to press, they are based on calculations undertaken in 1994, and the estimates for 2000 have been extrapolated.

Exhibit C-1

Market Forecast for Software—U.S., 1995-2000

Product/Service Sector	1994 ($M)	Growth 94/95	1995 ($M)	1996 ($M)	1997 ($M)	1998 ($M)	1999 ($M)	2000 ($M)	CAGR 95-00 %
Systems SW Products	23,721	7%	25,284	27,135	29,298	31,914	35,046	38,533	9%
Mainframe	9,738	1%	9,870	10,028	10,214	10,426	10,666	10,879	2%
Minicomputer	7,309	3%	7,519	7,741	7,977	8,228	8,493	8,748	3%
Workstation/PC	6,674	18%	7,895	9,366	11,107	13,260	15,887	18,906	19%
Applications SW Products	27,654	15%	31,728	36,421	41,988	48,531	56,129	65,148	15%
Mainframe	5,935	7%	6,330	6,751	7,191	7,635	8,191	8,764	7%
Minicomputer	6,904	9%	7,527	8,172	8,850	9,561	10,384	11,319	9%
Workstation/PC	14,815	21%	17,871	21,498	25,947	31,335	37,554	45,065	20%
TOTAL Software	51,375	11%	57,012	63,556	71,286	80,445	91,175	103,680	13%

Source: INPUT

Exhibit C-2 shows INPUT's standard worldwide forecast for software. As in the preceding chart, the numbers were estimated for 2000 based on 1994 data.

Exhibit C-2

Market Forecast for Software Worldwide—1995-2000

Product/Service Sector	1994 ($M)	Growth 94-95	1995 ($M)	1996 ($M)	1997 ($M)	1998 ($M)	1999 ($M)	2000 ($M)	CAGR 95-00 (%)
Systems SW Products	41,786	6%	44,409	47,443	50,921	55,168	60,341	65,168	8%
Mainframe	17,154	1%	17,333	17,533	17,752	18,023	18,364	19,650	0.02
Minicomputer	12,875	3%	13,205	13,534	13,864	14,223	14,623	15,062	0.03
Workstation/PC	11,757	18%	13,865	16,376	19,304	22,922	27,354	32,551	0.19
Applications SW Products	45,628	13%	51,734	59,095	67,935	78,215	90,319	103,867	15%
Mainframe	9,793	5%	10,321	10,954	11,635	12,305	13,180	13,973	0.07
Minicomputer	11,391	8%	12,273	13,260	14,319	15,409	16,709	18,045	0.09
Workstation/PC	24,444	19%	29,140	34,882	41,981	50,501	60,429	71,848	0.2
TOTAL Software	87,414	10%	96,137	106,538	118,856	133,383	150,660	169,035	12%

Source: INPUT

B
Programmer Expenditures on Component Software

In forecasting the market for components that are used by programmers, estimates of the revenues of companies producing components were made. Microsoft estimates that there will be 50 OLE 2 component suppliers. However, Microsoft itself is also a large component supplier and INPUT projects that many larger software companies are already in or planning to be in the market to upgrade their products.

In making these estimates, the market for components used by programmers is estimated. These may then be integrated into applications and the code resold, possibly on a royalty-free basis, to users. In defining the market the components were separated by the primary technology they supported, recognizing that OpenDoc components can also support OLE. Also as components grow more sophisticated, they may support several standards. However, the primary application of the component is counted. For example, if a component to interface to a CORBA-compliant ORB were marketed as part of a library of OLE components, it would be counted as an OLE component.

The hypothesis is that OLE components will evolve from .VBXs and .DLLs to .OCXs. There will also be free software from universities, as well as organizations that are compensated for service and support. Because the programmer spends nothing on these components, they are not counted. These forecasts do not count the development tools. So although compilers may come with components, the component

portion of the sale is not counted. Only spending on components that are packaged for sale on a separate price list are included.

Exhibit C-3 shows programmer expenditures in the U.S., and Exhibit C-4 shows corresponding figures for overseas.

Exhibit C-3

Programmer Expenditures on Component Software—U.S., 1995-2000

Type of Component	1994 ($M)	Growth '94-95	1995 ($M)	1996 ($M)	1997 ($M)	1998 ($M)	1999 ($M)	2000 ($M)	CAGR 95-00 (%)
OLE	90	42%	128	180	218	243	254	285	17%
CORBA	8	369%	38	46	51	59	83	108	23%
OpenDoc	na	na	na	20	38	53	68	82	na
TOTAL	98	69%	166	246	307	355	405	475	23%

Source: INPUT

Exhibit C-4

Programmer Expenditures on Component Software—Worldwide, 1995-2000

Type of Component	1994 ($M)	Growth '94-95	1995 ($M)	1996 ($M)	1997 ($M)	1998 ($M)	1999 ($M)	2000 ($M)	CAGR 95-00 (%)
OLE	100	60%	160	240	312	406	507	634	32%
CORBA	10	400%	50	65	85	118	166	215	34%
OpenDoc	na	na	na	25	50	70	91	109	na
TOTAL	110	91%	210	330	447	594	764	958	35%

Source: INPUT

C
User Expenditures on Software that Supports OLE, OpenDoc or CORBA

To forecast the user expenditures on software that uses one or more of the technologies discussed in this report, the software is divided into three categories:

- Application development tools
- Packaged software
- Enterprise software

Application development tools is a broader category than visual development tools, as described in INPUT's report, *Object-oriented Platforms for Client/Server Computing*. Forecasts from this report are given here for comparison in Exhibit C-5. Some visual development tools do not support OLE, OpenDoc or CORBA because they are from different language environments such as Smalltalk. Even in that forecast some of the Smalltalk tools support OLE, as virtually any development tool for Windows will support OLE.

Exhibit C-5

Market Forecast for Visual Development Tools, 1994-1999

	$M	%	$M	$M	$M	$M	$M	%
U.S.	640	48%	950	1,362	1,770	1,947	1,977	0
Worldwide	777	57%	1,221	1,990	2,996	3,445	3,790	0

Source: INPUT

Packaged software is shrink-wrapped or electronically delivered software that is generally available for either single users or group users (typically sold as a package for some number of users) and sold through retailers, mail order catalogs, resellers and distributors. It is usually for LAN or PC/workstation platforms. Its price is usually based on a published discount from list.

Enterprise software is typically delivered by a systems integrator, VAR, or software developer to a corporation. It may not be widely distributed and may or may not be shrink-wrapped. Often it will be delivered on CD ROM or magnetic tape, and it may be leased. Examples of enterprise packages are larger databases, enterprise systems management software, minicomputer and mainframe software. Enterprise software is typically priced on a contract that may include maintenance services and support.

Exhibit C-6

User Spending on Software that Supports OLE, OpenDoc or CORBA Standards—U.S., 1995-2000

Type of Component	1994 ($M)	Growth 94/95	1995 ($M)	1996 ($M)	1997 ($M)	1998 ($M)	1999 ($M)	2000 ($M)	CAGR 95-00 (%)
Application Development Tools	1,503	72%	2,584	3,634	4,355	4,031	3,686	3,506	6%
Packaged Software	4,008	41%	5,636	9,469	11,363	12,878	17,385	22,600	32%
Enterprise Software	334	824%	3,085	5,797	13,318	14,099	12,713	12,845	33%
TOTAL	5,845	93%	11,305	18,900	29,036	31,008	33,784	38,951	28%

Source: INPUT

Exhibit C-7

User Spending on Software that Supports OLE, OpenDoc or CORBA Standards—Worldwide, 1995-2000

Type of Component	1994 ($M)	Growth 94/95	1995 ($M)	1996 ($M)	1997 ($M)	1998 ($M)	1999 ($M)	2000 ($M)	CAGR 95-00 (%)
Application Development Tools	1,670	93%	3,230	4,845	6,222	6,718	7,371	7,790	19%
Packaged Software	5,010	50%	7,515	13,527	18,938	25,755	34,770	45,201	43%
Enterprise Software	1,670	224%	5,405	8,628	16,320	19,206	19,284	24,912	36%
TOTAL	8,350	93%	16,150	27,000	41,480	51,680	61,425	77,903	37%

Source: INPUT

The next two charts show user spending on software that supports component technologies (i.e., OLE, OpenDoc or CORBA) by platform. The main platforms are Windows and UNIX, with the MacOS having

a much lower market share. In the component software world the Macintosh market is behind the Windows market and these forecasts make the assumption that it will be hard to catch up. From an analysis of development tool vendors' revenues, only 4% can be expected to support the MacOS.

Exhibit C-8

User Spending on Software Supporting OLE, OpenDoc or CORBA Standards by Operating System—U.S., 1995-2000

Type of Component	1994 ($M)	Growth '94-'95 (%)	1995 ($M)	1996 ($M)	1997 ($M)	1998 ($M)	1999 ($M)	2000 ($M)	CAGR '95-'00 (%)
Windows	5,323	50%	7,985	11,977	14,518	13,954	13,821	14,022	12%
UNIX	251	287%	969	3,240	6,222	6,460	7,678	9,738	59%
MacOS	na	na	175	420	700	1,860	1,843	2,337	68%
Other	271	766%	2,351	3,263	7,596	8,734	10,442	12,854	40%
TOTAL	5,845	93%	11,305	18,900	29,036	31,008	33,784	38,951	28%

Source: INPUT

Exhibit C-9

User Spending on Software Supporting OLE, OpenDoc or CORBA Standards by Operating System—Worldwide, 1995-2000

Type of Component	1994 ($M)	Growth '94-'95 (%)	1995 ($M)	1996 ($M)	1997 ($M)	1998 ($M)	1999 ($M)	2000 ($M)	CAGR '95-'00 (%)
Windows	7,098	50%	10,646	15,969	20,740	23,256	27,641	31,161	24%
UNIX	418	50%	1,615	5,400	10,370	12,920	15,356	19,476	65%
MacOS	na	na	250	600	1,000	3,101	3,686	4,674	80%
Other	835	50%	3,639	5,031	9,370	12,403	14,742	22,592	44%
TOTAL	8,350	93%	16,150	27,000	41,480	51,680	61,425	77,903	37%

Source: INPUT

Some software will support more than one technology. Exhibits C-10 and C-11, forecast the total software market that is based on OLE, OpenDoc or CORBA. The OLE line refers to software that supports OLE, but not OpenDoc. The OpenDoc line refers to software that supports OpenDoc and that probably implies support for OLE. The CORBA line refers to software that primarily supports the CORBA standards. Some software may be designed to support both OLE and CORBA standards, in which case it may appear in both the OLE and CORBA rows. Therefore, there are no totals in Exhibits C-10 and C-11 that summarize these forecasts, to avoid duplication.

Exhibit C-10

User Spending on Software Supporting OLE, OpenDoc or CORBA Standards by Technology—U.S., 1995-2000

Type of Component	1994 ($M)	Growth 94-95	1995 ($M)	1996 ($M)	1997 ($M)	1998 ($M)	1999 ($M)	2000 ($M)	CAGR 95-00 (%)
OLE	4,950	100%	9,900	14,400	24,000	29,120	31,200	32,500	27%
CORBA	2,400	100%	4,800	9,000	11,760	13,800	14,950	19,435	32%
OpenDoc	na	na	na	1,800	3,200	6,000	7,800	9,360	na

Source: INPUT

Exhibit C-11

User Spending on Software Supporting OLE, OpenDoc or CORBA Standards by OperatingSystem—Worldwide, 1995-2000

Type of Application	1994 ($M)	Growth 94-95	1995 ($M)	1996 ($M)	1997 ($M)	1998 ($M)	1999 ($M)	2000 ($M)	CAGR 95-00 (%)
OLE	5,500	100%	11,000	18,000	32,000	41,600	52,000	65,000	43%
CORBA	3,000	100%	6,000	12,000	16,800	23,000	29,900	38,870	45%
OpenDoc	na	na	na	2,000	4,000	8,000	10,400	12,480	na

Source: INPUT

The forecasts presented in this section are based on analyses of vendor revenues, assumptions about industry trends and INPUT's standard forecasts from the Market Analysis Program.