

1. What is componentware, and how will it evolve?
2. What are the componentware strategies of the major system software vendors?
3. How will the industry's business infrastructure evolve to support componentware?
4. How can users prepare for componentware?

The battle for architectural control at the desktop and workgroup continues to rise from operating systems to middleware. The next battlefield where the system software giants will be competing is component software infrastructure (componentware). Microsoft's entry, OLE, is facing Apple and IBM's OpenDoc as well as a plethora of other IBM technologies augmented by the acquisition of Notes with Lotus. Now that componentware is all the rage, it too is subject to bloated expectations that obscure the much more realistic facts.

Much has been said about componentware being a revolution. We see it as more of an evolution of existing technologies and disciplines.



What is componentware, and how will it evolve?

Reader Notes

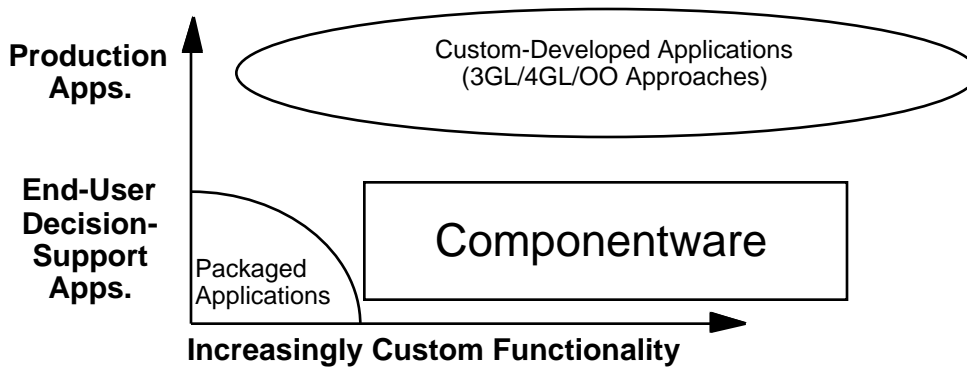
Market Conditions

- Huge and rapidly growing base of PCs
- Few high-volume system software platforms (Windows, OS/2 and Macintosh)
- Lack of differentiation and feature saturation in core personal productivity applications

Technical Advances

- Infobases (Notes, Cairo)
- Object/component models (OLE, OpenDoc, COM, CORBA)
- Object/component tools (VB, LotusScript, Taligent and NextStep)

User Requirements



Source: Gartner Group

Three forces are driving the PC industry toward componentware: 1) ISVs are encouraged to break up their applications into pieces so they can reuse the pieces (e.g., a chart function) in multiple applications; 2) users are generating a seemingly infinite demand for custom applications and are eager to leverage off the functionality of their packaged applications in their custom applications; and 3) system software is evolving toward an object model that explicitly enables small, reusable components.

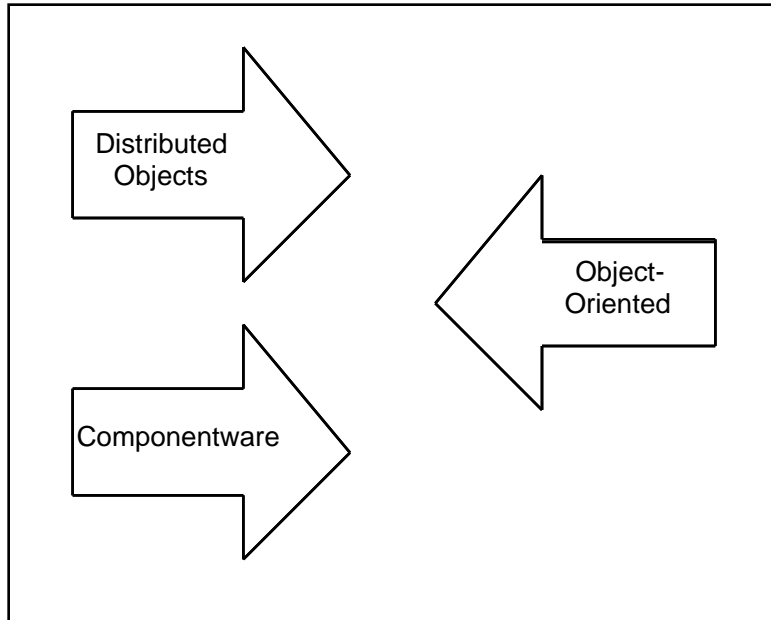
Componentware involves applying the economics and business model of packaged application software to the problems of custom applications. Vendors will be challenged to make the right technical and strategic choices regarding all layers of the componentware hierarchy, while developing their business models in the right directions at the right rates. Organizations will be challenged to manage the potential of more end users turning out greater numbers of “tactical” applications in less time.



Componentware and distributed objects are similar and are examples of object-based technologies.

Reader Notes

“The Big Picture” of Object Technology



Object-based
 Deployment focus
 “Black Box” reuse
 Coarse-grained objects

Object-oriented
 Development focus
 Code reuse
 Fine-grained objects

Source: Gartner Group

Key Issue: What is componentware, and how will it evolve?

Componentware and distributed object computing are examples of object-based technologies. Componentware is a PC-oriented term similar in concept to distributed object computing, although there are two differences in emphasis.

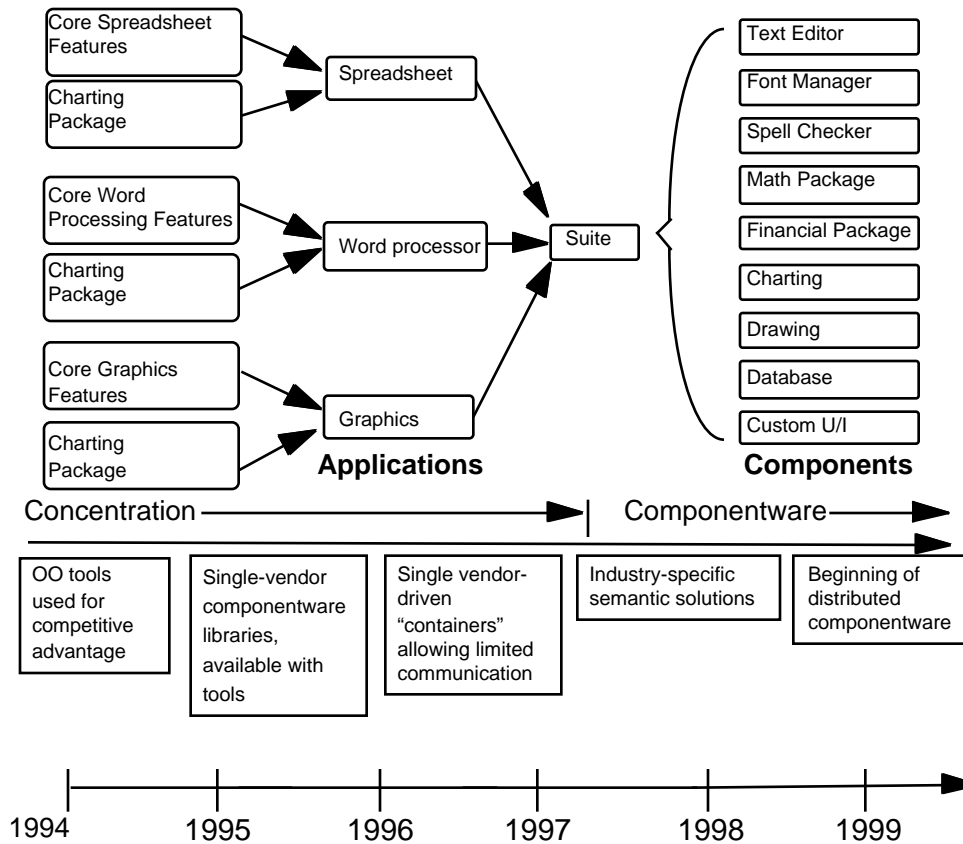
Componentware concentrates initially on linking medium-sized modules within a single PC; distributed componentware will happen over time. For example, Microsoft’s OLE/COM works within a PC (although numerous add-on middleware products enable communication with other environments using non-COM software mechanisms). Although it will not use CORBA interfaces, COM will become a type of ORB. The initial emphasis of OLE is on linking components from the same vendor (e.g., Microsoft) or from different vendors that agree to use a common interface, generally defined by the primary application vendor (e.g., Microsoft).

In distributed object computing, CORBA-style ORBs are often used to link coarse-grained objects from different developers; the interface details are unspecified, and are often tailored by the user. The emphasis is on providing the results of functions to the calling object.



Componentware itself will solve only a limited set of semantic communication issues (0.8 probability).

Reader Notes



Source: Gartner Group

Key Issue: What is componentware, and how will it evolve?

Componentware itself will solve only a limited set of semantic communication issues (0.8 probability). As long as one vendor sets the standard, communication can occur for what is specified. “Any-to-any” component interaction is a very difficult (perhaps unsolvable) problem. The industry focus on suites is based on a limited perspective of both the PC application software market and the long-term role of suites. Suites are successful because they represent highly integrated, user-interface-consistent bundles of the components most users will need — at an attractive price. The long-term role of suites will be: 1) a bundle of shrink-wrapped applications; 2) a bundle of reusable components that end users can recombine into custom applications; and 3) a platform that vendors of value-added components can extend. The suite sets the semantic rules.



Limited forms of componentware will become mainstream by 2000, but the full “any-with-any” ideal will never be realized (0.8 probability).

Reader Notes

Claims for Componentware

Reality

- | | |
|--|---|
| 1. Componentware will allow mixing and matching from any source. | • By itself, componentware only will solve limited semantic issues. |
| 2. Componentware and objects will simplify software distribution and management. | • Componentware actually will make the environment more complex. |
| 3. Componentware will solve bloated application issues. | • Functionality driven by market, componentization may adversely affect performance. |
| 4. Componentware is a revolution. | • Componentware is an evolution of existing technologies such as RPCs, DLLs and compound documents. |
| 5. OpenDoc is cross platform, OLE is not. | • Issues are grossly oversimplified; success of OpenDoc depends on complex issues. |
| 6. Conformance to OLE (or to OpenDoc) guarantees interoperability. | • OLE conformance means many things, mostly misunderstood. |

Source: Gartner Group

Key Issue: What is componentware, and how will it evolve?

Users and vendors have been searching for the elusive silver bullet technology that will solve all. Now that component software is the rage, it too is subject to bloated expectations and much more realistic facts. Componentware is not a revolution but rather an evolution of technologies such as RPCs and DLLs. As with all over-hyped technologies, componentware will bring some value to users but will not solve all software issues.

The most misunderstood claim is of “plug-and-play” components. Reality is that any software will allow only semantically consistent communication. Two cooperating pieces of software (whether on the same system, or even within the same process) need to agree on the semantics of the conversation to be meaningful.



What are the componentware strategies of the major system software vendors?

Reader Notes

Categories	Vendors			
	Microsoft	Novell/ WordPerfect	IBM/Lotus	Apple
Visual Programming Metaphor	Visual Basic, others	AppWare	Taligent, Bart Notes/VIP	Taligent, Denali
Script Language	VBX, OCX	PerfectScript	Lotus Script ?	Open Scripting Architecture
Component Library	Microsoft Office	Perfect Office	SmartSuite	Claris?
Compound Document Model	OLE	OLE OpenDoc	OpenDoc OLE Interop.	OpenDoc OLE Interop.
Class Library	MFC	?	Taligent, OpenClass?	Taligent
Object Model	COM	COM, SOM	SOM, COM CORBA	SOM CORBA
Info-base	Cairo, OLE DB	NetWare, AFS	Notes, Bento	Bento

Source: Gartner Group

As the desktop and workgroup software industry moves toward componentware, vendors will be forced to compete in new and different categories. Some vendors, such as Lotus and Microsoft, will differentiate themselves via the synergy between their component suite and their visual programming tools. Vendors will have to decide whether to compete with Lotus and Microsoft on the basis of having a broad and comprehensive component library. We believe that only a few vendors (Lotus and Microsoft) will be able to compete on the basis of having the most comprehensive library of components. Other vendors will have to compete either on the basis of having the best set of components or on the ability to add value to the Lotus or Microsoft suites through components not offered by the major vendors. Some vendors also will choose to market components from the broad industry of component foundries, providing a component integration service as their added value to the customer.



Use of the term OLE by Microsoft is evolving from a compound-document focus to that of a software integration architecture. This leads to even more confusion over a set of already complex technologies.

Two different meanings of OLE

OLE, the compound document architecture

- Object Linking and Embedding

OLE, the marketing umbrella, or “MAA”

- OLE Compound Documents
- OLE Custom Controls
- OLE Structured Storage
- OLE DB
- OLE Transactions
- Network OLE
- OLE DS

Source: Gartner Group

Key Issue: What are the componentware strategies of the major system software vendors?

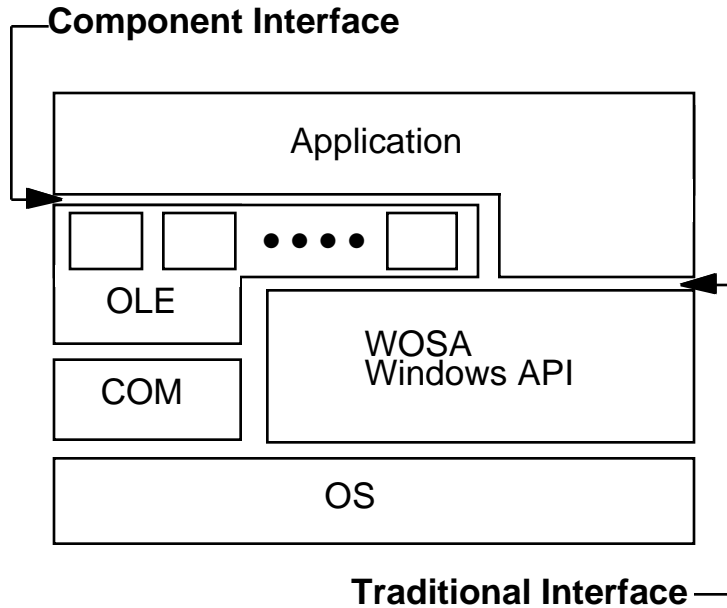
Microsoft uses OLE in many ways. OLE has two very different meanings:

1. The acronym OLE, which originally stood for Object Linking and Embedding, describes the methods most commonly invoked when dealing with compound documents. OLE is best known for its ability to provide a compound-document capability (linking and embedding). OLE 1.0 provided these capabilities and used DDE for rudimentary messaging. OLE 2.0 was rearchitected to use COM, which is partially compatible with DCE RPC.
2. OLE is being repositioned as a marketing umbrella, not a product. Like WOSA, OLE is not a product for sale. Microsoft no longer uses the term OLE 2.0 because it implies a product and versions. Instead, Microsoft is using the term OLE to refer to the software integration utilizing sets of interfaces that layer on COM. It is analogous to IBM's former SAA strategy that Gartner Group calls “MAA.”



OLE will be the leading componentware model but will remain inadequate for distributed workgroup applications until 1998 (0.7 probability).

Reader Notes



Microsoft is emphasizing OLE and the component paradigm to protect and increase its control of system software.

Source: Gartner Group

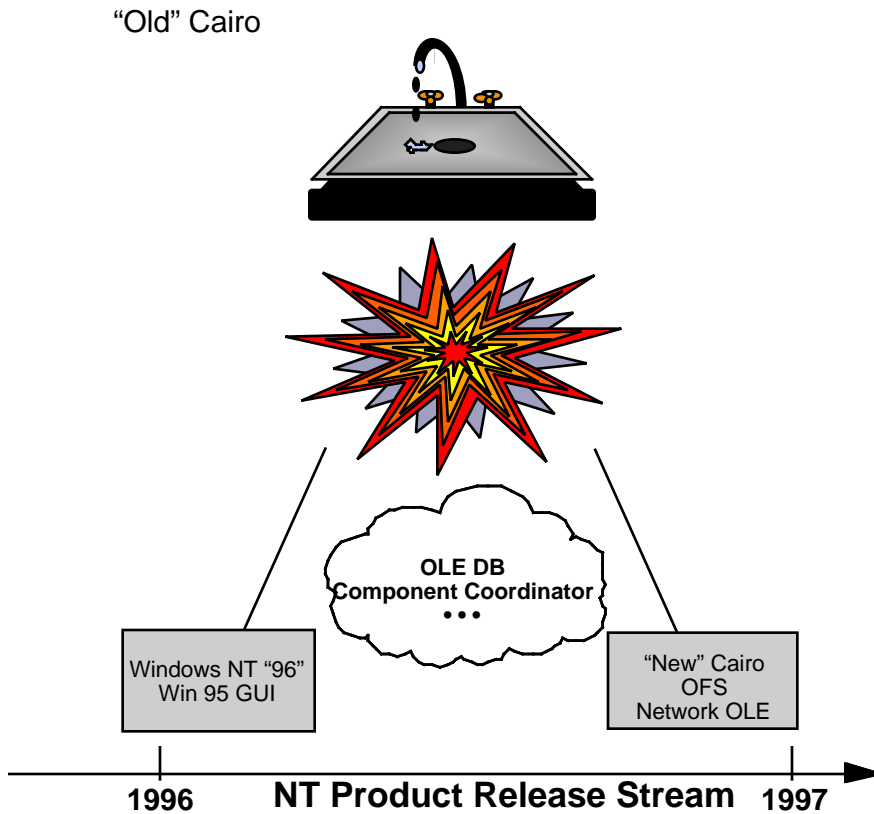
Key Issue: What are the componentware strategies of the major system software vendors?

Microsoft's strategy has been based on controlling the "important" (i.e., most strategic) layer of computing. Increasingly, the middleware (i.e., infrastructure) layer is replacing the OS layer as the more "important" layer. As this occurs, Microsoft is at risk, since its control has been at the OS and GUI levels, and based on the traditional programming paradigm of the Windows and WOSA API architectures. OLE, in conjunction with the marketing of Office and BackOffice as platforms, is Microsoft's strategy for extending its dominant desktop system software position into the workgroup, and eventually into the enterprise, via components. Microsoft recognizes the importance of controlling the software infrastructure. With software becoming more layered and isolated from the OS, Microsoft's system software control is at risk, most notably from Lotus with Notes.



The Cairo vision functionality will be delivered incrementally, via middleware such as OLE DB (0.8 probability).

Reader Notes



Source: Gartner Group

Key Issue: What are the componentware strategies of the major system software vendors?

The vision and technology formerly known as Cairo will be brought to market in different packages. Previously, Cairo was the answer to virtually any question. Microsoft will position Cairo more clearly as the next major release of Windows NT.

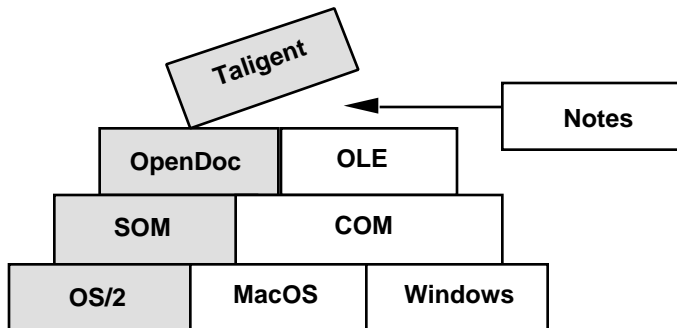
Microsoft has repositioned Cairo for three reasons:

1. To reset expectations and deal with the hype and FUD generated over it and other Microsoft system software
2. To mask schedule slips and provide flexibility in delivery schedules
3. To continue to blur the line between OS and applications by moving functionality to nebulous layers of middleware.



OpenDoc will enable the first credible alternative to Microsoft APIs by 1997 (0.6 probability).

Reader Notes



This alternative will not depend on better technology, but on:

- Consistent marketing of a clear alternative set of APIs
- Distribution of the technology
- Leveraging a top-down as well as bottom-up strategy

Topdown vs. Bottom-up Approaches

Top-down refers to an approach based on central design and control. Integration with existing applications and infrastructure are major parts of the top-down approach.

Bottom-up refers to an approach based on end-user-driven activities “trickling up” to the workgroup and beyond. Personal productivity and use of mass market technologies are major parts of the bottom-up approach.

Source: Gartner Group

Key Issue: What are the componentware strategies of the major system software vendors?

The shaky IBM tower is limiting the rallying of the alternative. The flexibility allows for insulation from the potential failure of products that may not be strategic to all the vendors (e.g., OS/2), but limits consistent marketing messages. We believe OpenDoc will enable the first credible alternative to Microsoft APIs by 1997 (0.6 probability). OpenDoc by itself cannot provide the alternative; it is part of an overall architecture.

The alternative faces the task of distributing its technology in a world of \$99 operating systems and applications. Hence, it will need to rely on bundling in operating systems, applications and tools.



Microsoft's OLE will be the leading component software architecture for the next five years (0.9 probability). OpenDoc will be a viable alternative by 1997 (0.7 probability).

Reader Notes

OLE vs. OpenDoc Five-Year View

	OLE	OpenDoc
Attractiveness to In-House Developers	2	4
Quality of Enterprise Glue	3	4
Ease of Programming	3	4
Breadth of ISV Support	5	2
Depth of Microsoft Applications Support	5	0
Depth of IBM/ Lotus Applications Support	4	4
Depth of Novell Applications Support	5	4
Viability on Windows	5	3
Viability on OS/2	0	4
Viability on Mac OS	4	5
Technical Prowess	3	4
Marketing and Message	5	2
Availability	Now	1H96

5 = Excellent
 4 = Good
 3 = Average
 2 = Below Average
 1 = Poor

Source: Gartner Group

Key Issue: What are the componentware strategies of the major system software vendors?

The OpenDoc platform has technical advantages over OLE. Yet, we believe OpenDoc will obtain at best a "viable alternative" status vs. OLE. As is so often the case in the desktop software industry, the best product rarely wins. The most important aspect in the desktop software market is not the product's features (as long as the market believes these will improve over time), but the aggressive support of large numbers of ISVs. This is, unfortunately for Apple and IBM, a replay of the Windows vs. OS/2 story. The inferior product won because of the depth and breadth of ISV support. A superior product coupled with inferior ISV marketing is not enough to carry the day.

Beyond the personal and workgroup space, factors such as enterprise glue and application interoperability make the prospects for OpenDoc and IBM middleware a bit brighter.



IBM's lack of vision will drive a collaborative Lotus/IBM Notes strategy that will suffer from conflicting goals and the force-fitting of IBM technology (0.55 probability).

Reader Notes

Scenario 1 — IBM rules (0.05 probability). IBM dictates a complete rearchitecting of Notes, utilizing its “more robust enterprise” middleware technologies. Notes fades away quickly.

Scenario 2 — Lotus rules (0.1 probability). IBM allows Lotus to run totally autonomously and Lotus wants absolutely nothing to do with any IBM technologies. Any potential synergy is lost.

Scenario 3 — Lotus' vision drives Notes strategy and utilizes IBM technologies as it deems appropriate (0.3 probability). Lotus uses IBM technologies judiciously in engineering the evolution of Notes. Lotus utilizes IBM technology only when there is a demonstrated benefit to Notes and its users.

Scenario 4 — IBM's lack of vision drives a “collaborative” Lotus/IBM Notes strategy (0.55 probability). In the worst case scenario, IBM and Lotus engage in lengthy technology and strategy debates. IBM technologies are “force fit” into the evolution of Notes because IBM deems them to be more technically elegant or strategic. Conflicting goals compromise the overall strategy.

Key Issue: What are the componentware strategies of the major system software vendors?

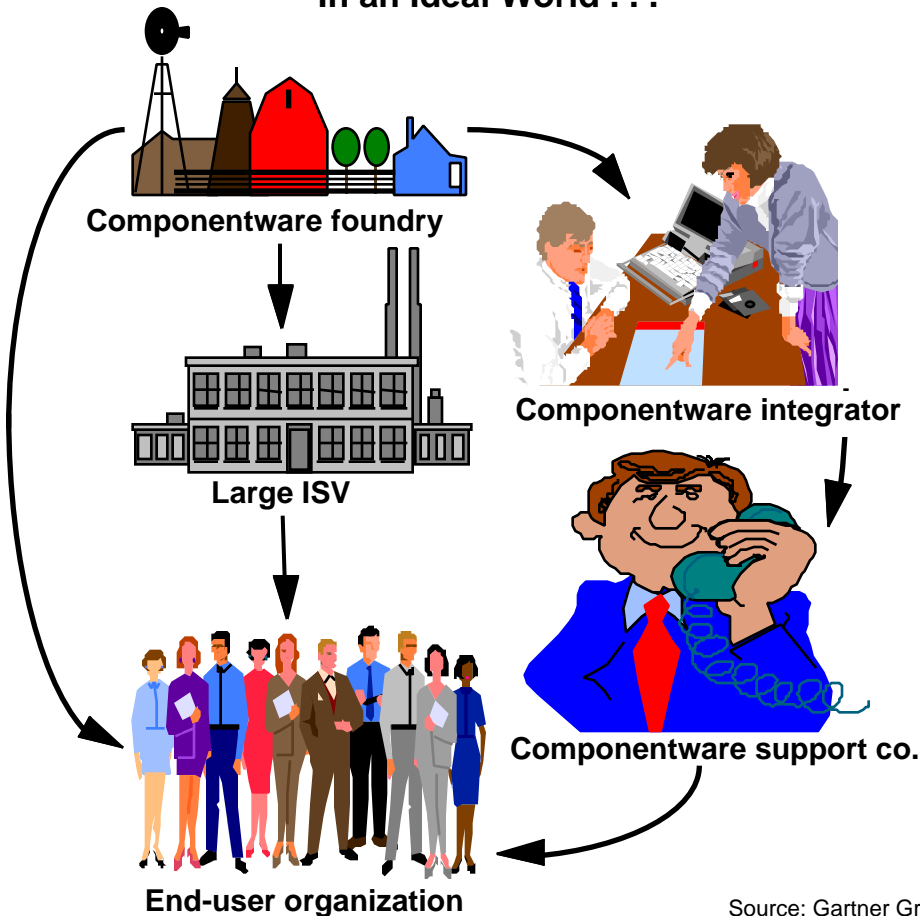
Integrating Notes with IBM's software technology arsenal will be challenging. IBM has a vast array of advanced middleware technologies, but thus far has been unable to leverage cohesively these technologies in its quest once again to influence user architectures. Although it has been sold primarily as an application, Notes is many things, e.g., middleware or infrastructure. Notes is an infrastructure because applications are written to the APIs it exposes. In the personal and workgroup space, in general, most infrastructure is not sold separately but purchased “unconsciously” as part of an OS, application or development tool. In the case of Notes, the infrastructure is really two things: APIs that “Notes applications” are written to, and a storage model. Application vendors see the APIs; users see the storage model as part of the user interface. These two parts of Notes define the infrastructure to developers and users. At these levels, IBM is trying to redefine its competition with Microsoft.



How will the industry's business infrastructure evolve to support componentware?

Reader Notes

In an Ideal World . . .



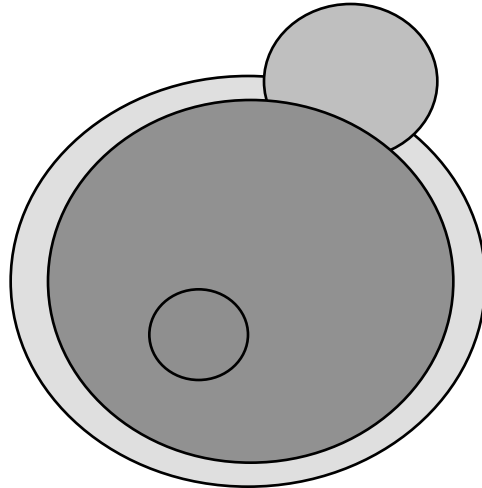
Source: Gartner Group

Componentware will precipitate dramatic changes in how vendors market and support software, and how users buy and integrate it. Users will need to focus on procuring the correct sets of components to allow their end users the appropriate degree of flexibility for developing and customizing applications. The concept of a “standard” desktop will be challenged and possibly invalidated by componentware, since each end user may require slightly different sets of components to achieve maximum productivity. Given that component-level heterogeneity will occur in the end-user community, user organizations will need to ensure that the cost of end-user computing, and especially the cost of end-user operations, does not skyrocket as “casual” applications development becomes another form of “fooling with the computer.” Vendors will see traditional methods of packaging software become invalidated, and they will also experience a change in the manner in which software is distributed.



Application suites will become the base library of components that every desktop needs.

Reader Notes



- Base Functions, e.g., OS plus Suite
- Specialty Product, e.g., multimedia authoring, desktop publishing and vertical solutions
- Replacement Component, e.g., text editor, table viewer and financial functions
- Horizontal Extension, e.g., drawing tools and video editing

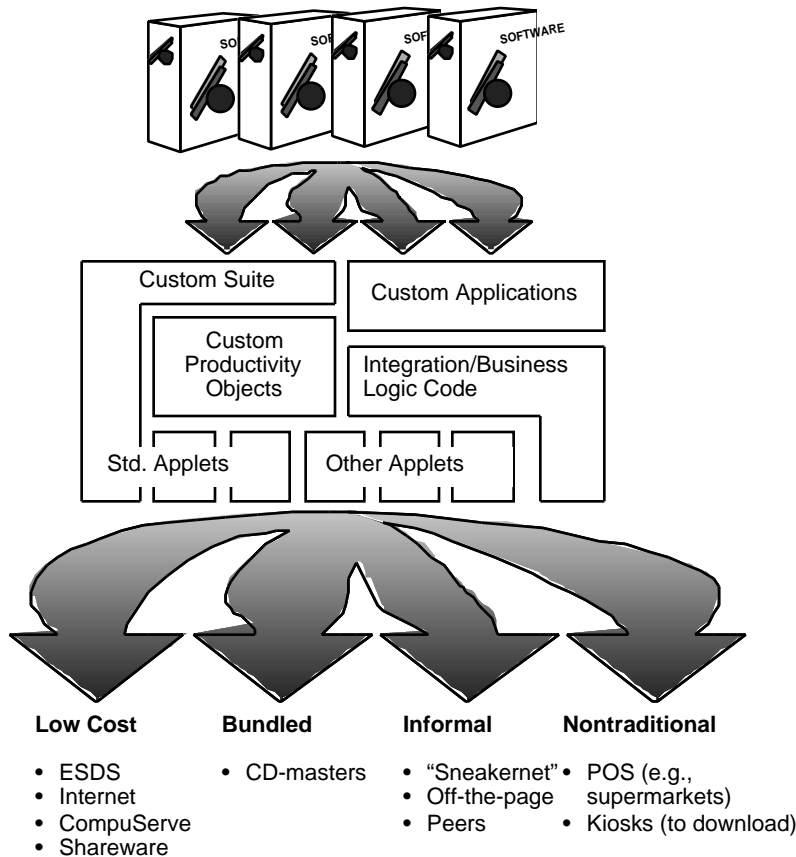
Source: Gartner Group

Key Issue: How will the industry’s business infrastructure evolve to support componentware?

The base level of functionality that ISVs have been able to assume exists on the users desktop and has grown consistently as the PC market has matured. In the move to a component-based desktop, the next level of assumed functionality will be defined not only by the operating system but by the base functions currently associated with the standard productivity suite applications (0.8 probability). While the suite vendor is concentrating on the core set of functions the user requires, other ISVs must develop a strategy that utilizes the availability of these functions. Apart from the base suite, we believe three distinct categories will emerge: 1) replacement components — the main type will be targeted at providing the user with extended facilities in particular areas, e.g., power charting and extended thesaurus; 2) horizontal extensions — certain groups of users will require a range of additional core functionality not provided in the base suite, e.g., advanced drawing capabilities, video editing and fax creation; and 3) speciality products (e.g., vertical solutions, multimedia authoring) targeted at a niche community that requires extensive features in a specialist area.



The changing economics of software distribution will force ISVs to focus on low-cost distribution methods, bundling and nontraditional channels (0.9 probability).



Source: Gartner Group

Key Issue: How will the industry's business infrastructure evolve to support componentware?

The PC software market has reached a major turning point for vendors. The market is rapidly starting to polarize (in the productivity segment) into two, or at most three, companies that can survive in the \$1 billion-plus bracket with the majority remaining with annual revenues below \$150 million. Most of these smaller vendors will not be able to compete with the few remaining giants for shelf space in traditional distribution channels.

Concurrent with these changes will be the growth in componentware, generating large volumes of small applications, some as small as a few hundred megabytes. Creating an individual shrink-wrapped package for each "applet" will be prohibitive. Bundles of components will be targeted as solution packs for specific markets with a variety of content and size (ranging from point products to full suites).

At the same time, structural changes in the PC market are creating classes of users who will demand new outlets in which to acquire their software.



How can users prepare for componentware?

Developer Affiliation	Developer	Technologies
Business Unit ↓	End user	Compound documents
	Power user or occasional developer	Macro languages, simple AD tools, desktop databases
Central IS ↓	Professional developer, business/implementation focused skills	Workgroup AD tools, workgroup databases
↓	Professional developer, wide range of AD skills	Enterprise AD tools

	Centralized	Decentralized	Hidden End User	Total
1994	2.0%	2.3%	1.5%	5.8%
1999	1.6%	5.6%	1.6%	8.9%

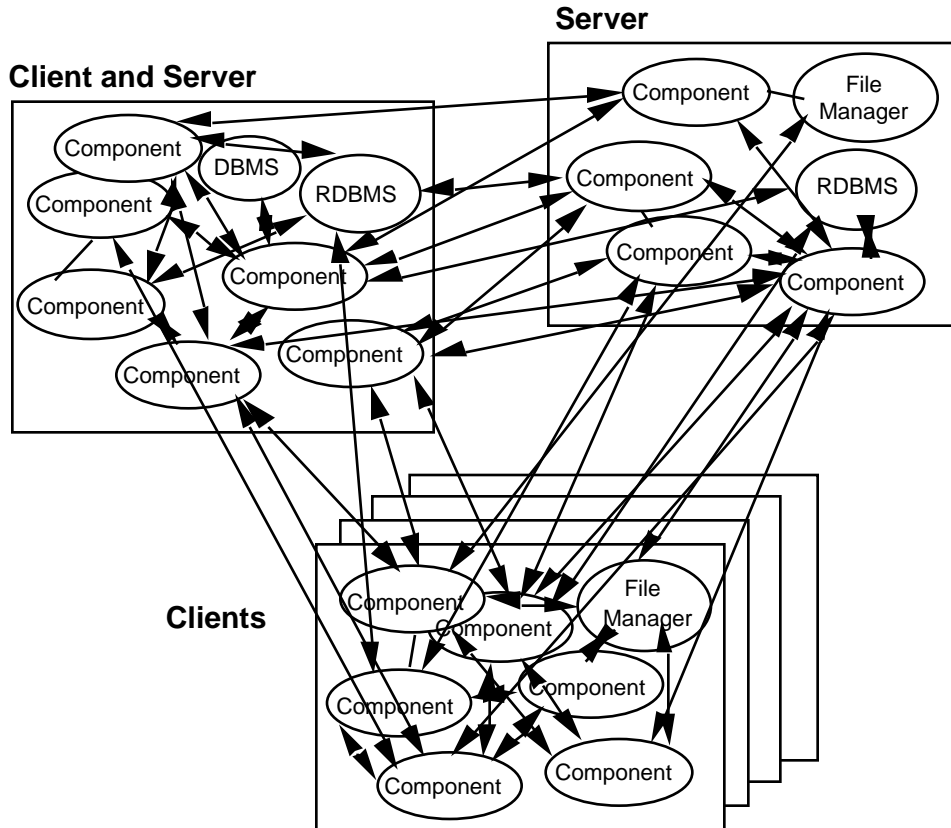
Source: Gartner Group

We are seeing a reduction in the role of centralized IT and an increase in development carried out by smaller teams within business groups, including a type of decentralized IS. By the year 2000, we expect to see 70 percent of development carried out at the business group level supported by specialized skills supplied by central IS. In parallel, we see technological power moving toward business groups with new and more-capable tools, permitting smaller numbers of developers to create more complex and more functional systems.

Technological and financial power moving to business units is not necessarily matched by appropriate skills; such groups are likely to include more occasional developers and implementation-focused developers with weak IT skills but strong business knowledge. Hybrids (particularly those exploiting weaker workgroup scale technology) provide a way for less skilled developers to deliver complex functionality.



Componentware and distributed object computing will increase the complexity of user architectures (0.8 probability).



Source: Gartner Group

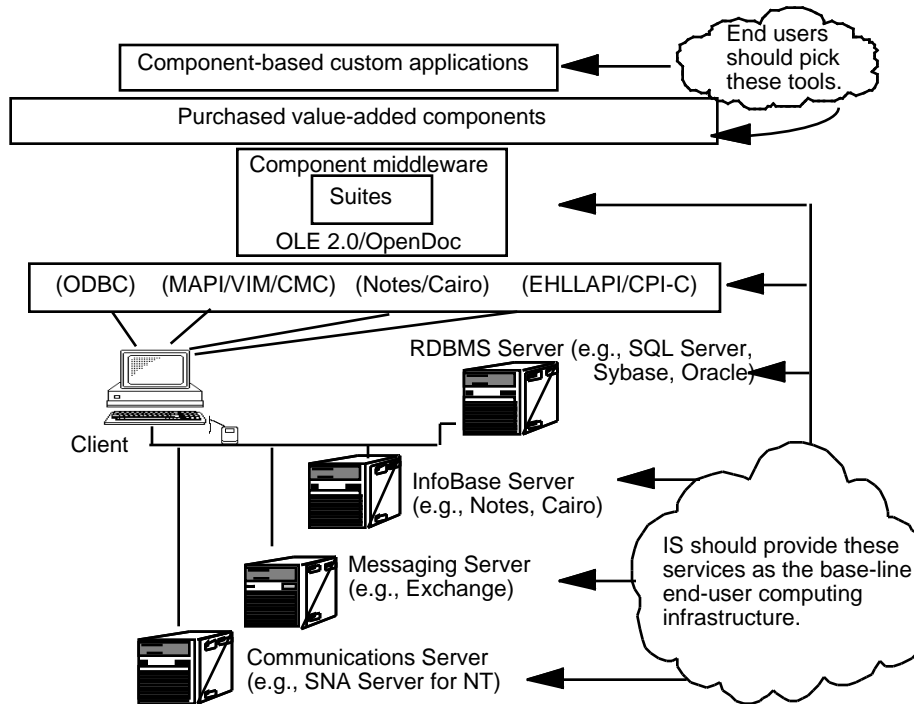
Key Issue: How can users prepare for componentware?

Componentware will make user environments more complex. Users are already losing links to documents on the same system using OLE. Proliferation of component technology and extensions over the network will increase complexity tremendously. Without adequate addressing of management issues up front by vendors and users, deploying component software has more potential to increase, not decrease, costs of software distribution and management, at least initially.

As componentware becomes distributed, these issues will increase. There is a threshold of manageability that management technologies cannot cross. As systems become more complex they tend to become unmanageable. Complexity can approach, and in some cases cross, the threshold of manageability; introduction of componentware increases complexity. Manageability as a function of user population size decreases, while willingness to spend as a function of user size increases.



Organizations can optimize the exploitation of a common infrastructure, or acquire applications that offer the best fit for business needs. They will not be able to do both.



Source: Gartner Group

Key Issue: How can users prepare for componentware?

Organizations will not be able to have a single infrastructure. Every purchase of an application, tool or middleware has the potential to bring in another. This also is true of desktop-oriented or workgroup solutions. While it is accepted that larger, production applications bring infrastructures, it is increasingly true that bottom-up solutions also bring infrastructures such as OLE, Notes and NetWare and their potential integration with top-down infrastructures such as DNS naming, Oracle, DB2, and CICS.

While it would be helpful if there were useful standards for naming and directory, organizations can take some steps by recognizing that the issues surrounding naming are organizational as well as technological. This means that when issuing names, even for local specific-purpose naming services, consistency and planning for integration into a global name space should be considered. Clients that do not take these steps will pay more in operational costs.



- Componentware and distributed object computing are similar and are examples of object-based technology.
- Componentware will solve only a limited set of semantic communications issues (0.8 probability).
- OLE will be the leading componentware model but will remain inadequate for distributed workgroup applications until 1998 (0.7 probability). The Cairo vision functionality will be delivered incrementally via middleware such as OLE DB.
- OpenDoc will enable the first credible alternative to Microsoft APIs by 1997 (0.6 probability).
- IBM's lack of vision will drive a Notes strategy that will suffer from conflicting goals and the force-fitting of IBM technology.
- Application suites will become the base library of components that every desktop needs.
- The changing economics of software distribution will force ISVs to focus on low-cost distribution methods, bundling and nontraditional channels (0.9 probability).
- Componentware and distributed object computing will increase the complexity of user architectures (0.8 probability).
- Organizations will have to balance a common infrastructure with applications that offer the best fit for business needs.

