Articles & White Papers

Deeper Inside Corba Services

As the basic ORB becomes a commodity, attention turn toward services.

by Max Dolgicer Author's note

Part 2 of 2

Growing support for Internet InterOrb Protocol (IIOP) and the Portable Object Adapter (POA) almost assures that the basic functionality offered by the Object Management Group?s (OMG?s) Common Object Request Broker (Corba) will become a commodity. This puts a special light on individual Corba products? support for Corba services. Support for these services will become the next battleground among Corba vendors, the area where they will really compete.

The bad news in this battle is that all Corba vendors still have a way to go to support a few important Corba services. The day when every vendor supports all Corba services is in the remote future. Furthermore, there are additional middleware services (for example, load balancing and fail-over) outside the realm of Framingham, Mass.-based OMG?s defined services that will need to be supported. The good news is that no vendor can do it all, and cooperation among vendors will result in a number of best-of-breed middleware solutions.

As the battleground of vendor-supported Corba services heats up, it must be remembered that not all Corba services are of equal importance. Among the more crucial services are the Naming, Event and Transaction services as discussed in Part 1 of this series. Also of special importance are the Security and Trader services, discussed here. Of the services of lesser importance, OMG has already defined a substantial number, including the Concurrency Control, Relationship, Externalization, Life-Cycle, Persistence, Licensing, Time, Properties and Query services.

Not only are the Corba services of unequal importance; their definitions are of unequal quality. Some services, such as Transaction, are defined in very detailed fashion, while others, such as Persistence, require updates. Despite these differences, we expect a secondary market for Corba services to emerge. As long as these services work with the Corba product that the service was intended for, this is probably very good news for the corporate developers who are the ultimate consumers of the technology. Support for OLE/COM is also becoming key to Corba services.

Today, support for Corba services is far from being a commodity. Vendors differ widely on what services they support and how good support for a specific service is. Fortunately, every Corba vendor is in the race to support at least what appears to be "must have" services. Note, however, that support for Corba services is optional. These services should be used only when appropriate.

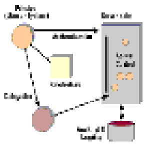
Distributed objects and security

It is somewhat ironic that the specifications for Corba Object Security Services (OSS) have been defined relatively recently. Security has long been a hot topic. However, it is only now, as corporations have finished experimenting with Corba and have begun to deploy production-quality applications, that the issue of security has become very important.

Security in distributed environments is complex, and the Corba environment is no exception. In fact, the requirements for security in a Corba environment apply to objects in addition to users, and in theory at least, security is required at almost every level of the Corba specification. For example, object creation may depend on the user's credentials, and the Corba Trader Service may be required to return to callers only that information that they are allowed to see or access.

Even in distributed Corba environments, standard security concerns still need to be addressed. These include the famous three A?s -- authentication, authorization (including access control) and auditing/logging -- as well as encryption and non-repudiation. While the definition of each "A" is straightforward, its implementation in a distributed environment is not trivial. Extra requirements such as encryption and non-repudiation further complicate the situation.

It is important that the Corba Object Security Service addresses the security of the basic transports by which objects communicate with each other. The Security Service can be implemented at three levels. (Click to expand Security Service view.)



Level 0 specifies the authentication and session encryption, which is equivalent to the functionality provided by Mountain View, Calif.-based Netscape Communications Corp.'s Secure Socket Layer (SSL) library. Using SSL, which operates by layering a security protocol on top of standard TCP sockets, object servers are capable of being registered with the Object Request Broker (ORB) as 'secure,' so that only clients with appropriate authentication credentials can connect to them. This also requires the implementation of a Secure Internet InterOrb Protocol (SecureIIOP).

Level 1 supports Corba client objects and Corba server objects that can be secured without being aware of the presence of the Security Service. Applications running under a secure ORB are a good example of the Level 1 implementation.

Level 2 supports Corba client objects and Corba server objects that are aware of the presence of the Security Service.

Since OMG develops only specifications and not software, the Corba OSS only suggests a security architecture. It does not enforce how the security is implemented; rather, it only indicates that there is a security service that exposes its interfaces in a standardized way. As a result, a vendor's actual implementation of OSS can leverage the implementation of an existing security solution, often one provided by a third party. In fact, it is entirely possible that Corba vendors will leverage the security offered by DCE for authentication and authorization, or the public key encryption offered by security vendors who have implemented RSA encryption algorithms. The highly robust and functional Generic Secuirty Service Application Programming Interface (GSS-API) may also be employed.

In some industries it is hard to imagine distributed, mission-critical applications running in production without support for end-to-end security. To many people's surprise, however, many applications using different types of middleware are running in production today without comprehensive security. Corba-based applications are not an exception. Still, application designers and developers need to be aware that security can be very expensive, both in computer processing time and also in operational overhead. For example, to authenticate each request may add additional, unnecessary overhead, while authentication only at the time of initial connection adds almost no overhead and may provide adequate security in some cases. Application designers need to strike the right balance, given the nature of the applications and general policies, between the exposure presented by security risks and the possible negative effect that bullet-proof security might have on the application performance and operational overhead.

When Naming doesn't cut it

The Corba Trader Service provides an alternative, or more of an enhancement, to the existing Naming Service (see Part 1). It allows clients to locate objects using complex search criteria that are expressed as a sequence of attributes. The search criteria are matched against the attributes of registered objects to locate a suitable match. Furthermore, when a server object wishes to make itself known (that is, advertise) to the Trader Service, it calls the registration function, passing its object reference and a description of the service it provides. In sum, the Trader Service not only supports the advertising and discovery of objects at runtime, but also allows applications to specify a preference that will determine the order in which the set of matching objects is returned to the application.

Compared to the simple and straightforward specifications of the Naming Service, the Trader Service provides more robust functionality. In some cases, this can be overkill in terms of additional application complexity and performance. For example, applications that merely need to obtain an object reference to an already-known object should use a simple name-to-address resolution mechanism offered by the Naming Service. On the other hand, if an application needs to obtain an object reference to a specific object from a large set of similar objects, then the Trader Service should be considered. Be aware, however, that while some other types of middleware, such as Message-Oriented Middleware (MOM), already support this capability, "off-the-shelf" Corba products do not yet support this service.

Not all Corba services are equal

Because all Corba services are not of equal importance, vendors offer spotty support for second-tier services such as LifeCycle, Relationship, Licensing, Externalization, Properties and Persistence (the latter being, in our opinion, an important service). Here we cite the first-tier services as Naming, Security, Transaction, Event and, arguably, Trader. Some vendors provide either partial or full support only for selected services. For example, Expersoft Corp., San Diego, provides full support for Relationship Service and Iona Technologies Inc., Dublin, Ireland, provides adapters for almost every possible ODBMS that is used to support Persistence Service.

Moreover, OMG is in the midst of defining and finalizing a number of additional services. The most important, in our opinion, is the Asynchronous Messaging Service. This service extends the traditional, synchronous Corba model of interactions with a notion of asynchronous, "store-and-forward" processing that supports a variety of Qualities of Service (QoS), such as guaranteed and assured (once and only once) message delivery, levels of message confirmation, message tracking and reporting, to name a few. We believe that extending Corba with messaging capabilities will result in a very powerful best-of-breed middleware combination.

A number of vendors are racing to support Asynchronous Messaging capabilities even before OMG finalizes the specification. Driven by a strong market demand, Iona is integrating with IBM?s MQSeries, as well as integrating with San Francisco-based PeerLogic Inc.'s PIPES Platform. An Orbix Adapter for MQSeries also will be shipping later this year. Expersoft is integrating with niche MOM player Modulus Technologies Inc., Houston, with its InterAgent product. More activity is expected in this arena.

Another additional service, the Notification Service, should provide the ability for an object to subscribe and receive automatic notification on specific, definable events such as "tell me when the airplane left the gate" or "tell me when Microsoft stock exceeds $150." Both the Notification Service and the Messaging Service are closely associated with Event Service. The combination of events, notifications and fast message distributions (which is offered by some products) is, in our opinion, the key to having robust and scalable push technology.

Does all this service activity reflect poorly on "standard ORB" vendors? No. In fact, it is neither feasible nor advantageous for all Corba vendors to be the same. After all, each is competing for the same corporate dollars in this lucrative and fast growing middleware space. Furthermore, corporate developers cannot count on OMG to define everything that they may possibly need, or on each Corba vendor to implement what is defined by OMG. In our opinion, unique features, even though they are proprietary, should be welcomed by corporate developers. These should continue to exist and should be used if it makes developing applications easier and faster. The good news is that there are some powerful product extensions out there. San Mateo, Calif.-based Visigenic Software's SmartAgent technology; Iona's Filters, Smart Proxies and Loaders; and

Expersoft’s passing objects by value are just a few examples of powerful product specific extensions.

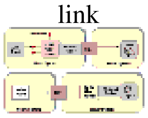
OLE/COM internetworking

Whether we like it or not, the programmatic ease of use of Corba can best be judged by how seamless the integration is between Corba objects and a litany of desktop applications. As an example, let’s assume that a number of key distributed applications have been developed using a Corba paradigm and that objects such as "book a flight," "money market deposit" and "calculate financial risk" are Corba objects (written in Java or C++). The ability to access these objects from Visual Basic, PowerBuilder or Microsoft Office 97 makes tremendous sense. Today, it is common practice to develop server side objects of applications without predefined dependency on the clients accessing them. Often the server objects are being developed by different groups then the application front end that needs to access them. This is where the interoperability with OLE/COM becomes very handy.

Since everything (or almost everything) on the Microsoft desktop is OLE/COM enabled, COM/Corba interoperability makes tremendous sense. After all, business managers don’t care whether it is COM, Corba, or both. All they care about is how quickly applications can be developed to gain competitive advantage. OMG understands the realities of these market dynamics, and there are specifications in place to address the issue of COM/Corba interoperability (COM/Corba internetworking).

COM/Corba internetworking consists of two parts. The specification for Part A has been approved and is now part of the Corba 2.0 specification. Part A does not support distribution through OLE/COM, which means that the internetworking components live on a COM platform, and communication across the network is accomplished via a Corba protocol. Part B of the protocol, now being considered for approval, will include support for Distributed COM (DCOM).

Click here to link



The bottom line of the OLE/COM internetworking specification is that an OLE/COM client can invoke methods on an object in a Corba server, while a Corba client can seamlessly invoke methods on an object in an OLE/COM server with similar ease - all with minimal programming effort. Also important is that the OMG specification calls for full bi-directional communication between OLE/COM and Corba. The most visible shortcoming of the Part A specification is that it doesn’t address the issue of DCOM. However, this should not preclude Corba vendors from trying to address the issue before OMG does.

Naming names

There is not much you can do without at least rudimentary support for naming. It is safe to say that every Corba vendor supports the Corba Naming Sevice it in one form or another, although strictly speaking, not all Corba vendors comply with OMG specifications for the Naming Service (see Part 1). Whether or not all Corba vendors comply with OMG’s specifications, the implementation of the Naming Service among the vendors is quite different.

Iona’s Orbix, for example, supports the hierarchical naming of objects. This feature, combined with the elegant proprietary Object Locator mechanism, allows specific individual hosts and even groups of hosts to search for registered implementations if an object cannot be found on the host from which it was originally requested. The Visigenic Corba product does not support hierarchical naming yet, but it does provide an interesting feature called SmartAgent, where information pertinent to object registration with the Naming Service is propagated across multiple hosts. This is valuable especially in cases where a client request for an object execution needs to be dynamically routed and bound to another host should the original host terminate prematurely.

Of the other Corba players, two are worth mentioning. First is the forthcoming IBM Component Broker (CB), which has implemented the Naming Service using DCE's Cell Directory Services (CDS). Second is Anaheim, Calif.-based Suite Software's SuiteValet (formerly SuiteDome), which is the only Corba product that has implemented a fully federated Naming Service, although the implementation does not comply with the OMG specification yet.

Event Service

As discussed in Part 1, the Event Service defines an implicit, asynchronous and decoupled way of communication between objects. Using an Event Service, applications can subscribe and publish information without knowing about each other. As of this writing, most Corba vendors have implemented the Event Service. However, implementation of the basic Event Service does not guarantee development of scalable, publish-subscribe applications. The reason is simple -- an Event Service uses a point-to-point (connection-oriented) approach, which limits its scalability and usefulness when information needs to be distributed to hundreds or even thousands of consumers. This is why development of additional functionality will be required for any Corba vendor to address the issue of scalable, publish/subscribe applications.

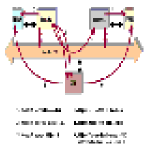
Iona's OrbixTalk implementation is similar in some respects to the Event Service specification. However, instead of using point-to-point communication, it uses IP Multicast, which is much more efficient and scalable. Using OrbixTalk, the world of applications is divided into talkers (publishers) and listeners (subscribers). The information destined to reach the listeners is distributed using OrbixTalk, which also makes sure that the messages reach the destination reliably and in the correct order. As of this writing, Orbix is the only Corba product that extends the Event Service beyond the standard specification.

Transaction Service

The Transaction Service or Object Transaction Service (OTS) takes the notion of distributed transactions to the world of Corba. OTS is clearly one of the most important services in Corba. It is well defined and it adds very little complexity to the application code. In fact, OTS does not require changes to the IDL. It only introduces one new interface, called Transactional Object, from which any other IDL interface inherits From a vendor’s perspective, OTS is very sophisticated. However, it is not easy to implement. People who have implemented distributed transactions before, namely inside databases or TP Monitor engines, will probably concur with this statement. Corba’s OTS has been modeled after the Distributed Transaction Processing (DTP) standard specification defined by The Open Group (formerly X/Open), Cambridge, Mass.

In OTS, objects (for example, Object A and Object B as shown at right participate in the transaction register with OTS, which in turn implicitly propagates the transactional context during the transaction.

Click to link



This begins when Object A starts a transaction and continues when Object A updates a database and invokes Object B, which in turn updates another database. This continues until the end of the transaction, at which time Object A commits (or rolls back) the transaction. At this point OTS coordinates the distributed two-phase commit. As stated, the good news is that, if necessary, OTS can be quite transparent to applications. It will pass the transactional context implicitly through the transaction. However, there may be situations where application designers will be ready to compromise the transparency and ease of use of OTS for more control. In these cases, OTS also offers the possibility of applications passing the transactional context explicitly, as a parameter throughout the transaction, where OTS is less transparent to the applications, since the parameter list to each relevant operation within the transaction must be changed.

It is not easy to navigate through the conflicting vendor claims and marketing literature. However, as stated in Part 1, a new breed of middleware called Object-Transactional Middleware (OTM) is coming up strong. To be specific, OTM is either object-oriented (Corba) or object-based (COM) middleware that also supports distributed transactions. OTM is being positioned by many middleware industry experts as a sort of ultimate middleware, a cure to all application scalability and integration evils.

Iona's Orbix Transaction Monitor (Iona's interpretation of OTM as an acronym) will be shipping later this year (most likely beginning Q4'97) with a number of Corba services implemented. OTS is one of them. In cooperation with IBM's Pittsburgh-based Transarc unit, Iona has developed OTS with the Encina Transaction Manager as the underlying foundation. OTS is 100% Corba-compliant, and it supports flat as well as nested transactions. OTS is bundled as part of both Orbix OTM and Encina Monitor, which can be purchased from Transarc.

Given IBM's experience in transaction processing, it is hard to imagine the company not delivering OTS. So when Component Broker ships it will fully support OTS, including flat and nested transactions. Since Component Broker is a reincarnation of different technologies that have been integrated, the $64,000 question is when it will ship and on what platforms. We expect that Component Broker will ship Q4'97 on selected platforms, namely Microsoft Windows NT and IBM's AIX and OS/2.

The integration of ObjectBroker from BEA Systems Inc., Sunnyvale, Calif. (the Corba-compliant ORB acquired earlier this year from Digital Equipment Corp.) and Tuxedo (the TP Monitor exclusively licensed two years ago from Novell Inc., Provo, Utah) will result in yet another Object-Transactional Middleware sometime in 1998, with an early support program beginning in Q4'97. Although the task ahead of BEA Systems is monumental -- any post factum integration is difficult and the integration of two products developed by two separate companies at different times is especially difficult -- it is doable. Also, today neither Tuxedo nor ObjectBroker supports multithreading, and an OTM is expected to be at least multithreaded.

When it comes to other vendors, the OTS situation is less clear. Visigenic is in the midst of developing an implementation of OTS that is positioned as part of its Integrated Transaction Services (ITS), and Expersoft has announced a partnership with Tokyo-based Hitachi, using its TPBroker. It is not possible to verify the practical ramifications of this partnership yet. However, it should be noted that Visigenic implied support for TPBroker in earlier releases of VisiBroker.

In the end, people in need of OTS support should draw their own conclusions as to whether to wait for vendors to deliver OTS or to proceed without it. The good news is, that compared with the landscape of just one year ago, most vendors have made substantial progress and a few vendors are just steps away from making this important service generally available.

Security Service

What about vendor support for security in general and for Corba Object Security Service in particular? As noted, at least today, Corba products vary widely in their support for security.

With the exception of ObjectBroker from BEA Systems and Suite Software's SuiteValet, there is nothing to write home about. ObjectBroker supports any GSS-API enabled security for authentication, with underlying security being either DCE or any other third-party security product such as CyberSafe. SuiteValet is fully integrated with Kerberos implementation, which includes support for authentication, authorization and full auditing.

Most vendors are bundling support for SSL (what is termed Level 0 Corba security) with their base products. For example, Iona is about to release support for GSS-API with DCE as an underlying security mechanism with support for additional third-party security providers using GSS-API to follow shortly. When IBM's CB ships it will support security based on DCE, and it is reasonable and realistic to expect that more vendors will ship support for different levels of Corba security (Level 1 and Level 2) in 1998.

Looking ahead

We have neither covered all Corba services nor addressed many additional features such as load balancing, fault tolerance, or different angles of system management. All of these features are required by enterprise-class middleware. These features have only recently started to be addressed by the Corba vendors..

Still, both Corba specifications and Corba products are maturing very fast. In fact, OMG as an organization as well as all Corba vendors now understand that time-to-market is key. What corporate decision makers need to remember, however, is that real-world product maturity is not only measured by announced features. Product maturity is measured by a vendor's ability to incorporate customer feedback into its products. To determine what a certain Corba product offers today -- and to assess what it will offer tomorrow -- requires in-depth research and development of a prototype.

Is compliance to Corba services critical? Is functionality more important than compliance? In our opinion, what matters more in the end is the implemented functionality. But, since OMG only defines interfaces to services and does not dictate the underlying implementation, the time when we could have our cake and eat it too might not be too remote.

---

 1 Apple Hill, Suite 301, Natick, MA 01760
(508) 652-1100 FAX (508) 652-1220 e-mail:general@spgnet.com http://www.spgnet.com