

A Comparison of Distributed Object Technologies CORBA vs DCOM

Filip Nowak, Mohsin Qasim

Abstract.

This essay compares two technologies: CORBA and DCOM. This two different distributed object technology has presented various approaches to component software model. DCOM most widely used standard developed by Microsoft and CORBA open distributed object computing infrastructure developed by OMG-Object Management Group. Goal of this essay is to show the similarities and difference of those two standards. To achieved that we compare them in different categories: such as programming languages, development support, platforms, etc..

Introduction.

CORBA and DCOM are extension of traditional object-oriented system- by allowing objects to be distributed across different network, which is very important nowadays, when Internet become so popular. The objects my stay in their own address space outside of an application or on different node then the application and still be referenced as a part of application. CORBA is the specification for an emerging technology known as distributed object management (DOM).DOM provides a higher level object oriented interface on the top of the basic distributed computing services. CORBA has emerged as the leading standard among DOM solutions with only one significant competitor. DCOM formally know as network object linking and embedding is the natural evolution of component object model

CORBA overview. [3]

- OMG created CORBA
- OMG group non profit organization with over 800 members primarily from industry.
- Main goal of OMG was to develop, define system that could solve problems with integrating networked applications.
- First CORBA specification becomes available in the beginning of the 1990's.
- current version of CORBA is 2.4
- It address the two most prominent problem faced in the software industry
 - The difficulty of developing client server applications
 - By rapidly integrating legacy systems off-the-shelf application and new development

DCOM overview.

- DCOM is described as COM with a longer wire; this is an extension to Component Object Model (COM).
- DCOM- Distributed Component Object Model- supports remote object by running on protocol called the Object Remote Procedure Call (ORPC).
- 10 millions of people using Windows daily in networked environment DCOM become widely used
- main goal of DCOM is to support the development of components that can be dynamically activated and that can interact with each other.
- DCOM object model is centered on the implementations of interfaces.
- COM provides operation through which a client can connect to one or more server applications.[2]

CORBA in detail.

Developer can easily and quickly integrate network resident software module and applications to create new more powerful applications. It combines object technology with a client server model to provide a uniform view of an enterprise computing system every thing on the network is an object. The highest level specification is referred to as the object management architecture (OMA). As in the diagram given [1]

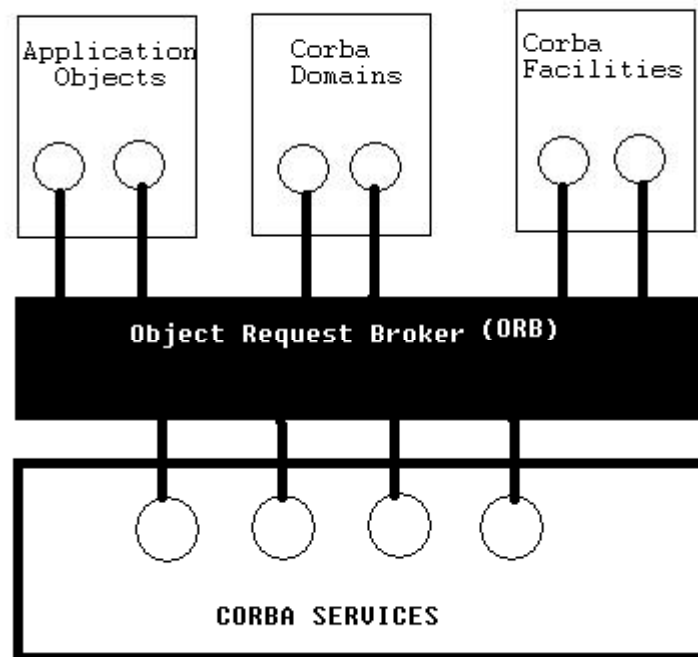


FIGURE 1 Object management architecture

The term CORBA is often used to refer to the object request broker (ORB) itself as well as to the entire OMG architecture.

- The role of ORB is to route request among the other architectural components.
- CORBA services, CORBA facilities and CORBA domain are also defined as part of specification.
- CORBA SERVICES provide some basic system level services such as
 - naming,
 - persistence
 - Event notification.
- CORBA FACILITIES are the set of higher level functions that cover a broad range of generically applicable facilities in area such as user interface and information management. CF are oriented towards end-user applications
 - DDCF- Distributed Document Component Facility- allows for the presentation and interchange of objects based on document model
- CORBA DOMAINS are the specific to particular application domains, such as manufacturing finance and telecommunication.
-
- APPLICATION OBJECT provides the new business capabilities that are created by the system implementers.

The key to integrating application object is the specification of standard interfaces using the interface definition language (IDL). Once all applications and data have an IDL-compliant interface communication is independent of

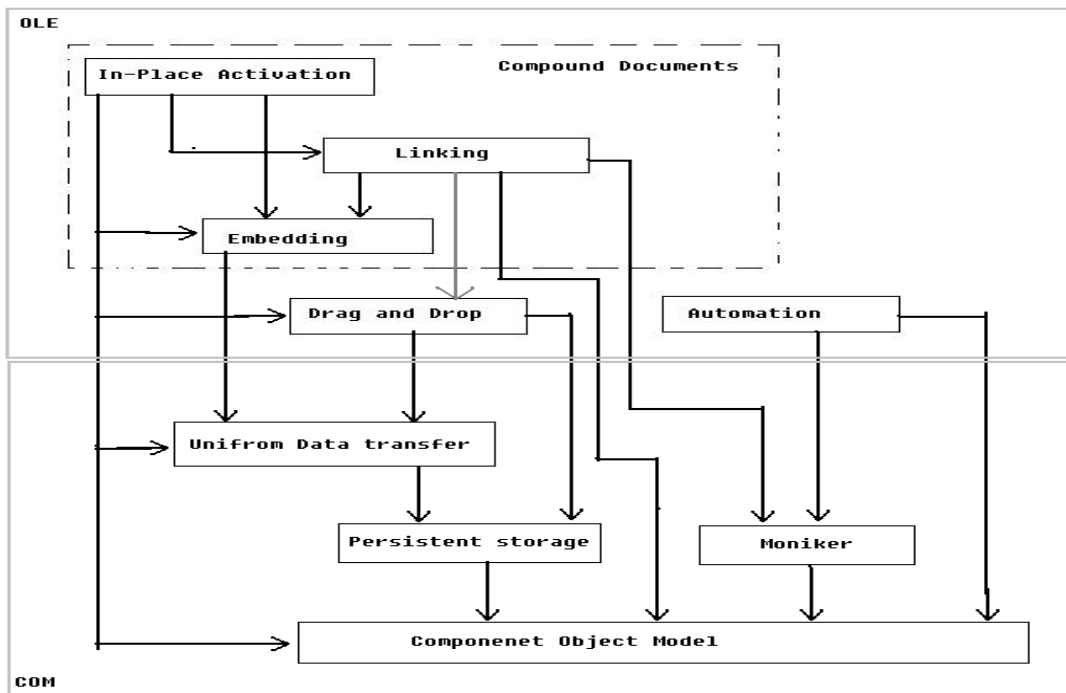
- physical location
- platform type
- networking protocol
- programming language

An information system is created by using CORBA to mediate the flow of control and information among these software objects

DCOM Details.

DCOM object model is centered around the implementations of interfaces: MIDL-Microsoft Interface Definition Language – is used to define interfaces for the DCOM object. It supports two different forms of interface descriptions:[1]

- basic *IUNKNOWN* – every object must be implement in this interface
 - Query interface – is a method to navigate among the interfaces.
- *IDISPATCH*- extends *IUNKNOWN*- is a sort of gateway interface- to many more interfaces
 - *AddRef()*
 - *Release()*
- (*MIDL*) associates classes to the interfaces.
- Benefit of binary interfaces is that interfaces are programming independent.
- Libraries- store the type of information of the object.
 - Can be also used to invoke objects implementing the *IDISPATCH* interface.
- each interface in DCOM has unique 128-bit identifier called Interface Identifier *IID*



OLE is best known as compound document framework. Recent extensions to OLE extended this notation to a fully functional desktop integration framework. The services provided by OLE are: [1]

- Compound documents- The ability to link information in a document through three services
 - In place Activation –Allow containers application to display component object. Providing the user with an ability to edit component application operations.
 - Linking- provides access to common data with updates
 - Embedding- Provides a separate copy of source data into container objects
- Drag-and-drop -Adds OLE objects to windows drag-and-drop
- Automation dispatch service to control application and provides features similar to the dynamic innovation mechanism of CORBA
- The Com provides the basic communication mechanism on which all the other services rely:
 - Uniform data transfer- Extends the clipboard to handle OLE objects
 - Monikers- Naming of objects; Limitations include names being server specific
 - Persistent storage- Stores multiple objects with in a container objects provided by the compound files services

COMPARISSION BETWEEN CORBA AND DCOM

One of the DCOM key distinctions is its specification of binary interface standards that is supposed to provide better portability then CORBA. However the reliance on the win32 API eliminates portability in circumstances in which it is not available on UNIX or MVS

Similarities between CORBA and DCOM [1]

SIMILARITIES	DCOM	CORBA
Object model	Yes	Yes
Standards body	Recently made formal; managed by the active group, an open group affiliate	Formal; managed by the object management group
Interface similarities	Microsoft IDL allows for separation of interface and implementation and providing a repository for storage of interface (although the IDL interface is not complete.)	CORBA IDL allows for separation of interface and implementation and provides a repository for storage of interface
Language independency	YES	YES
Compound document model	Yes (OLE)	Yes –open disc
Location transparency	Yes	Yes

Difference between CORBA and DCOM [1]

DIFFERENCE	DCOM	CORBA
Focus	Desktop first Enterprise second	Enterprise first Desktop second
Platforms	Windows NT; Future support for windows (all version) Macintosh, Unix(various) and MVS	MVS, UNIX(various),Windows (all version),Macintosh
Availability	Single vendor; Availability from other vendor expected	Multi-vendor
Services differences	Active X—interface content standard	Significant number of additional services, including query, trader, transactions, as well as facilities in the area of information management and systems management. Lastly service in areas such as finance distributed simulation and computer integrated manufacturing
Maturity	NT shipped in 1996; decade-long evolution of OLE and COM products, most services and facilities under construction	Product since 1992; Many services and facility under construction
Language binding	C ,C++: working on JAVA, visual basic, ADA	C++, Smalltalk, Ada95:JAVA and COBOL in process
Interface inheritance	Supports aggregation but not inheritance; interface are not classes	Multiple inheritance; Interfaces are classes

Microsoft dominance in the personal computer industry will make DCOM a significant player in distributed object architecture. However the focus on the desktop will not be enough for most integrations problems. Microsoft ability to make DCOM an enterprise-wide solution remains questionable. In the end both DCOM and CORBA will have to coexist. DCOM does not currently support the entire enterprise and Microsoft ability to deliver feasible technology across the enterprise has been marginally successful. However Microsoft Dominance on the desktop cannot be ignored, requiring coexistence between two.

References

- [1] Inside Corba by Mowbray Thomas 1997
- [2] Microsoft: The Component Object Model Specification,
<http://www.microsoft.com/com/resources/specs.asp>
- [3] The Common Object Request Broker: Architecture and Specification, Revision 2.4.2 February 2001
<http://www.omg.org/technology/documents/formal/corbaiop.htm>

Additional resources

- <http://www.corba.org/>
- <http://www.microsoft.com/>
- <http://www.microsoft.com/com/>
- <http://www.w3.org/>
- <http://www.omg.org/>
- <http://www.biztalk.org/>