ESSAY TITLE: MIDDLEWARE TECHNOLOGIES IN WINDOWS XP, WINDOWS VISTA

MODULE NAME AND NUMBER: CNM018 DISTRIBUTED OBJECT ENGINEERING

STUDENT ID: 0853756

STUDENT NAME: MAREPALLY ARUNKUMAR REDDY
MIDDLEWARE TECHNOLOGIES IN WINDOWS XP AND WINDOWS VISTA

ABSTRACT:

The essay here discussed is about the Middleware Technologies which are acting a key role in Windows and high level description about the technologies used in Windows XP and windows Vista. This paper also includes the description of popular Distributed Object Technologies or Middleware Technologies like CORBA, JAVA/RMI, and COM. The main aspects which I have covered in this paper are about the BUGS and CASE STUDIES on Current Companies which are adopting Middleware.

KEYWORDS:
MIDDLEWARE, MIDDLEWARE TECHNOLOGIES, CORBA, JAVA/RMI, COM, WINDOWS XP, VISTA, 7.

INTRODUCTION

MIDDLEWARE:

In the interest of competitive and development of technologies every company had adopted new software’s and updated technologies according to the new releases. Earlier companies used software’s for exchange of information on different platforms and different operating systems. Thus, there was a problem of integration of software from various sources.

Now the companies are using Middleware as an interface or communicating medium between ‘x’ number of applications. Any number of applications can be connected to middleware interface which reduces the time for exchange of data. But, there are some important services which the Middleware should concentrate on such as:

- **Availability** of communication medium for all the applications which are connected to it. The applications which are connected to it are heterogeneous i.e. of different types.

- **Reliability** here refers to assurance or trust as the applications sends message to middleware, it is the responsibility of the middleware interface to take that message to the destination.

- **Traffic** between the applications and middleware is very high as there is only one medium of communication between them.

In a simple way a middleware can be defined as software that combines the applications and operating systems e.g. Windows,
UNIX. It is used in wide areas because of it can ease the task of designing, programming, deploying and managing distributed applications. The three most used distributed object technologies are CORBA, COM, and JAVA/RMI. (Daniel Serain, 1999).

AN OVERVIEW OF MIDDLEWARE TECHNOLOGIES:

CORBA: Common Object Request Broker Architecture is a set of standard mechanisms for naming, locating, and defining objects in a distributed computer environment. CORBA is standardised by Object Management Group (OMG) and is the most used software in Non Windows market. OMG was founded in 1989 to promote the adoption of object-oriented technology and reusable software components.

COM: Component Object model is the object based middleware from Microsoft. COM is used by the developers to create re-usable software components, link components together to build applications, and take advantage of windows services. Variety of programming languages are used to create COM objects. Object oriented languages like c++, which gives the mechanisms that simplify the implementation of COM objects. COM technologies include COM+, Distributed COM (DCOM) and ActiveX controls.

JAVA/RMI: Java/Remote Method Invocation provides a distributed computing platform specially focused on java based clients and servers. Due to java's inherent platform independent capabilities, RMI-based applications are capable of running on a wide variety of computing platforms.
DESCRIPTION OF MIDDLEWARE TECHNOLOGIES:

CORBA: Object Based Middleware
CORBA describes architecture and gives the specification for processing objects distributed over a network. CORBA is also an international standard for which several companies offer software products. CORBA belongs to a new generation of software which helps in connecting the software modules together. This can be achieved by combining two technologies: Object Technology and Distributed Processing. (Daniel Serain, 1999).

Object Management Group (OMG) defined the Object Oriented Architecture called the Object Management Architecture (OMA) which is characterized by four elements:

**OBJECT REQUEST BROKER (ORB):** This broker forms the controlling element of the architecture because it supports the portability of objects and their interoperability in a network of heterogeneous systems. This is the first element in OMA to be implemented as a software product.

**OBJECT SERVICES:** Allows the creation of objects and the control of access or manipulating their addresses. Their goal is to simplify the process of constructing application.

**APPLICATION SERVICES:** These services offer a set of facilities for allowing applications access to databases, to printing services, or to synchronize with other applications.

**DOMAIN INTERFACES:** fill roles similar to Object Services and Common Facilities but are oriented towards specific application domains like manufacturing, telecommunication etc.

Fig: Object Management Architecture
CORBA ARCHITECTURE:

Let us see a simple request sent by a client and the destination is the server.

The server consists of the code and the information that forms the implementation of the object. The broker (ORB) is responsible for all the mechanisms which allow the location of the server preparing it to receive the request and also transfer the data to the server. Let’s see a Student interface which can send a request to an instance (e.g. Chris) to which it holds the reference asking for the execution of an operation (e.g. Pass). The request can be generated in two ways: Statically and Dynamically. [2]
Static Invocation:
Static invocation is similar to the RPC mechanism because it allows synchronous and one way communication and it assumes the existence of stubs at the client and server levels. The client stub makes the connection between the client and the agent. The server stub makes the connection between agent and the server.

Static invocation generally assumes that the communication between the client and server is pre-defined which means that it knows why they are in construction process. Client code is linked to the stub in order to obtain a single executable module. Similarly the executable module of the server is the outcome of linking code for the methods with that for the skeleton.

**Fig: Example of request sent by the client and transmitted to the server by the agent**
Dynamic Invocation:
The client can also invoke the server in a dynamic method. The dynamic invocation is similar to the mechanism which is used in message based middleware. The database containing the interface is loaded from descriptions of the interfaces in IDL. Its main aim is to allow the client to determine the existence of interfaces that were not present while the construction.

In dynamic invocation the client has no stub but on the other hand the server must have a skeleton.
JAVA/RMI: REMOTE METHOD INVOCATION
Remote method call in java allows objects belonging to one java virtual machine (JVM) to transparently invoke methods belonging to objects in another JVM. The originality of the Java/RMI comes from the fact that the call parameters to an object can be themselves java objects.

The Java/RMI uses the technique called serialization to pass objects. This technique allows the representation of an object in the form of a character string that is directly interpretable by the receiver object. [3]

JAVA RMI ARCHITECTURE:
The Java RMI allows communications between two entities located on two different Java virtual machines. The Java RMI implements the
client-server model in which the client is either an applet or a Java application and the server is an object belonging to a Java application.

In order to access the object server, the Java client must obtain its address. It obtains this by asking for an entity which is called the Registry. The Registry acts as a name server. Each reference contains the interface and the address of the object. By using the object reference, the client is then able to ask for the execution of a method on this object. The request passes through layers such as:

- **Stub/Skeleton layer**: The stub represents an image of the remote object. It possesses the same interface. Its function is to receive calls and transmit them to the real object the intermediary of the layer immediately below the remote object.
- **Remote reference layer**: The remote reference layer deals with the semantic aspect of the communication. It handles ordering function and multi cast function which means the ability to send the same request to all objects offering the same interface.
- **Transport layer**: transport layer is for establishing the connection as well as its management.

Every Java entity whether applications or applets can use the Java RMI system which contains three layers Stub/skeleton layer, Remote reference layer and Transport layer and also runs on the java virtual machine.

**COM: COMPONENT OBJECT MODEL**

COM is the Object-based middleware from Microsoft. The COM model defines mechanisms for the creation of objects as well as for the communication between clients and objects that are distributed...
COM implies the existence of code at the machine’s operating system level. The code forms the COM middleware which is organised as a library which is used in Microsoft windows using Dynamic link Libraries (DLL) and consists of:

- A set of functions used by the clients and servers forms the API (Application Programming Interface) of the COM infrastructure.
- The server location service determines on the basis of class identifier of the name and the location of the server which handles the particular class. This service is called Service Control Manager (SCM).
- Remote Procedure Call is used when the client and object are operating in two different memory spaces.
- A mechanism for controlling names and for naming new objects. The name allows the clients to access an object in the system no matter where it is.
- A uniform transfer mechanism for data. This mechanism is implemented using an interface through which the client and objects exchange data.

COM middleware allows the creation, storage and the naming of objects. It also allows communication between objects and the exchange of data. [1]

Fig: COM Middleware
PAST, PRESENT AND FUTURE BUGS:

BUGS RELATED TO MIDDLEWARE IN WINDOWS XP, WINDOWS VISTA AND WINDOWS 7:

- Windows XP had to undergo changes in Middleware products and Windows Live Messenger.
- Windows XP SP1 made it possible to drain once mandatory programs or middleware such as Internet Explorer, Windows Media Player, Microsoft Messenger and Outlook Express.
- Many tests have been done before and after the release of Windows Vista Service Pack 1, WMP 11 and Internet Explorer 7 due to its middleware issues.
- Microsoft has corrected most of the issues related to Middleware and the Technical Committee has fixed those issues. Microsoft has included these corrections in XP SP3 and Vista SP1.
- A bug in the ‘Open With’ feature is said to be corrected in Windows 7 and the Technical Committee has found some Middleware issues in Windows 7 which indeed will be modified and update from Windows 7 M1 (Milestone 1) to Windows 7 M2. ([http://news.softpedia.com/news/Windows-XP-SP3-Vista-SP1-and-Windows-7-in-the-Same-Boat-80669.shtml](http://news.softpedia.com/news/Windows-XP-SP3-Vista-SP1-and-Windows-7-in-the-Same-Boat-80669.shtml)).
- Microsoft initially changed Windows XP in order to quell antitrust concerns and these changes carried over Vista. Using a semi-simple control panel, users would be able to configure built-in or third-party applications to handle such ‘Middleware’ duties as Web Browsing, email, instant messaging and media playing. Windows Vista is more elegant than XP in this regard because vista lacks the built-in shell links to certain digital media-related services.
- One of the major Vista features Microsoft showed off, incidentally, was instant desktop search. The competitors of Microsoft released many other search feature applications after Microsoft delayed to update in Vista. Later Microsoft added the search feature in Windows XP, called Windows Desktop Search.
- Windows 7 is slightly faster than Vista on identical hardware. But it’s also still significantly slower than Windows XP, while generating twice as many threads and consuming nearly three times as much RAM as XP to run the same application load. ([http://itexpertvoice.com/home/smart-cards-lead-windows-7-drive-for-security/](http://itexpertvoice.com/home/smart-cards-lead-windows-7-drive-for-security/)).
CURRENT CASE STUDIES:

USE OF MIDDLEWARE IN PRESENT DAYS:

HEATHROW TERMINAL 5 SERVICES is using Progress Software middleware to integrate airport services and provide a hassle-free experience for passengers.

Progress Sonic Enterprise Service Bus (ESB) enables BAA to integrate many different back-office operations that guarantee the smooth running of any airport. Now that British Airport Authority (BAA) runs Heathrow must ensure that flights are in right place, there are steps for aircraft and the catering and cleaning staffs is in right location- all of these are handled through the Airport Operational Database.

Altogether, Heathrow handles 67 million passengers a year and daily manages the movement of 1,600 planes. It’s a huge logistical challenge that needs a coordination approach within each terminal and communication with other terminals. Progress software is adopted by the Heathrow airport because of its highly reliable and robust nature to ensure business continuity. ([http://middleware.cbronline.com/news/progress_links_heathrow_t5_services_281009](http://middleware.cbronline.com/news/progress_links_heathrow_t5_services_281009)).

PrismTech, Nextel Engineering to provide real-time data management offerings. Focuses on OpenSplice DDS real-time middleware.

PrismTech, a provider of software integration and infrastructure services, has partnered with Nextel Engineering Systems, systems integrator providing offerings to the defence, aerospace, security and transport sectors, to deliver real time data management offerings. As part of its software and services offerings, Nextel will deliver and support PrismTech’s suite of middleware products.

The focus will be on OpenSplice DDS, a real-time publish/subscribe middleware that enables quality-of-service enabled service oriented architecture and net-centric applications. OpenSplice DSS offers the ability to provide low-latency, high-throughput data dissemination across distributed systems. The OpenSplice application can add to performance critical systems in other vertical markets such as financial services, manufacturing, transportation, and telecommunications.
MuleSoft releases Mule Data Integrator for data mapping and transformation.

MuleSoft a Web Middleware Company has released Mule Data Integrator, which helps to simplify data integration and transformation tasks. It is integrated with open source enterprise service bus, Mule ESB, providing users with the infrastructure for application integration and SOA.

According to MuleSoft the Data Integrator complements ESB’s flexibility in messaging formats, providing a data transformation service that supports a variety of structured and unstructured data. (http://middleware.cbronline.com/news/mulesoft_releases_mule_data_integrator_for_data_mapping_and_transformation_091119).

CONCLUSION:

Next Generation Middleware

Middleware technologies like CORBA, JAVA/RMI and COM are which refer to object oriented technologies are used to develop lot of middleware applications which are used by companies like Microsoft for developing their operating systems (Windows). But the challenge from present applications and updated applications, including the multimedia, real world requirements and mobility seems that the need for defining a new architecture for open distributed systems is needed. The structure of the new architecture should be concentrating initially on flexibility and adaptability services and this can be achieved by defining an open engineering middleware platform that is run time configurable and allows inspection and adaption of the underlying components.

Every company should adopt Middleware which is suitable to the company on reasons like infrastructure, technology etc. The future could depend on the High level interface, services associated with each interface and object oriented based middleware.
REFERENCES:


http://biz.yahoo.com/msft/p7.html
http://www.ll.mit.edu/HPEC/agendas/proc02/abstracts/hanes.pdf
http://www.jot.fm/issues/issue_2004_05/article2/
http://www.winsupersite.com/showcase/winvista_google_changes.asp
http://middleware.cbronline.com/
http://www.uninet.net/~blaisdel/whistler3.htm
http://www.microsoft.com/com/default.mspx
http://itexpertvoice.com/home/smart-cards-lead-windows-7-drive-for-security/