

## Simple Experiments of DCOM, JAVA IDL and RMI

Akira Nakamura, Ryotaro Takano, Satoru Simizu

International Christian University  
Department of Information Sciences  
3-10-2 Osawa, Mitaka, Tokyo, Japan

### 1. Introduction

A Cambridge graduate, Andrew Watson, *Director of Architecture, OMG*, suggested me how the U.S military investigated three *middleware* DCE, DCOM and CORBA and made recommendations [1]. In the document, DCE played important roles and influenced both of DCOM and CORBA, but is becoming historical. Therefore we concentrate on DCOM and CORBA. The documents describes interesting comparisons as follows:

#### Key features of DCOM:

- Support for object-oriented language interfaces.
- Ready availability on Windows operating systems.
- Minimal availability outside of Windows platforms.
- Ability to intercommunicate (bridge) easily with DCE.

#### Advantages of DCOM:

- Strong tool and system support,
- Lower cost of DCOM-compatible networks,
- Easier interfacing to object-oriented languages than DCE,
- Good separation of interfaces and implementations.

#### Disadvantages of DCOM:

- Platform dependence,
- Sole source lock-in.
- Complicated non-intuitive programming style for objects and interfaces.

#### Key features of CORBA:

- Support for a broad range of platforms and programming languages,

- Broad and diverse market support,
- Standardization of a flexible Internet protocol for inter-object communications,
- Support for object-oriented distributed applications,
- Maturity of its object concepts,
- Emphasis on network transparency (the ORB concept)

Advantages of CORBA:

- Support by about 800 vendors, developers, and users,
- Platform independence,
- Open, public process for creating and approving specifications,
- Long-term definition and support of middleware services,
- Easier to understand, program and support at the software coding level,
- Support for object-oriented languages and designs,
- Support of the Internet Inter-ORB Protocol (IIOP)
- Ability to integrate legacy software applications,
- Good separation of interfaces and implementations,
- Strong focus on providing dynamic interfaces.

Disadvantages of CORBA:

- Less integrated tool support than DCOM,
- Weak vendor-to-vendor interoperability of CORBA products.

The report lead recommendations based on the investigations, however we would like to ignore them; firstly we would like to do simple Internet experiments using both of DCOM and CORBA base systems, secondly that emphasizes strongly on one product called 'VisiBroker'.

## 2. DCOM -- *Distributed Component Object Model*

DCOM[2] is the Microsoft middleware which is extended from the Microsoft Component Object Model (COM)[3]: COM is the Microsoft proprietary object-oriented technology to manage Microsoft document components and their associate GUI control. I was surprised that MS-IDL is exploited by COM rather DCOM. The philosophy of COM makes an association to a greatest development of the object-oriented analysis and design areas: '*Design Patterns*' [4] and the Unified Modeling Language, *UML* [5]. The both of them deny the traditional object-oriented essentials: classes and inheritances. DCOM exploits the concepts of components and interfaces. UML exploited concepts of types and interfaces.

We avoid the heavy mathematical foundations about classes, inheritances, types, and interfaces [4]. We would like to discuss very intuitive ideas from system programming.

Window NT is a layered micro kernel based operating system [6] supported by

- (1) message oriented mechanisms (LPR),
- (2) the object management
- (3) kernel threads,
- (4) heavy using software interrupts.

We remember the failure of Smalltalk-80 and the success of C++.

### 3. CORBA – *Common Object Request Broker Architecture*

CORBA [8] is an open standard of the Object Management Group(OMG) established 1989. The role of CORBA is defined in the object management architecture(OMA) reference model. The central component of OMA is the Object Request Broker(ORB) for unified accesses to application services. Moreover OMA defines four categories of objects: CORBA servers, Application Objects, CORBA domains and CORBA facilities.

The CORBA architecture is a dynamic model of understanding how CORBA works

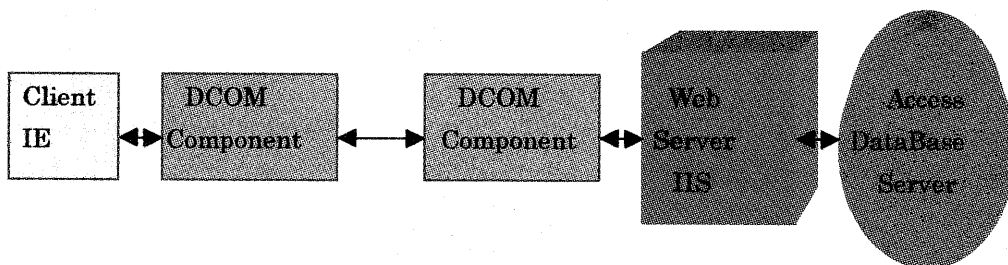
It shows how application software communicates through ORB.

### 4. Simple experiments

#### 4.1 An experiment of DCOM

DCOM supports distributed objects by splitting a single application into a number of different components objects running on different machines. DCOM works by using a proxy to intercept an interface call to an object and then issuing a Remote Procedure Call(RPC) to a stub to make the actual call to the real instance of the object that is running in another process on another machine.

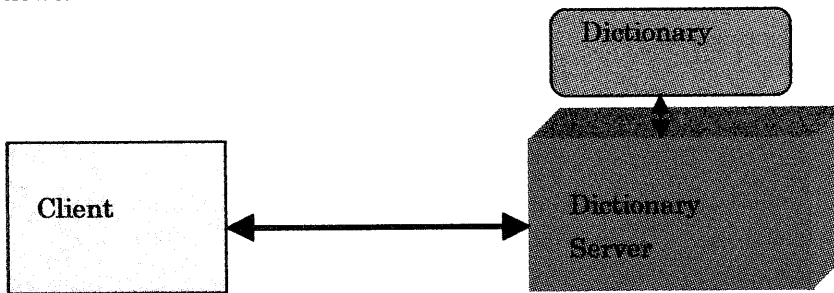
The following figure is a simple example using DCOM.



The experiment is an ordering system which a client orders a product and the Web server inquires the Access Database system by using ODBC and issues an invoice. . All components are written by C++ provided by the Visual Suite and runs on a Windows NT server and Clients.

#### 4.2 An experiment of JAVA-IDL

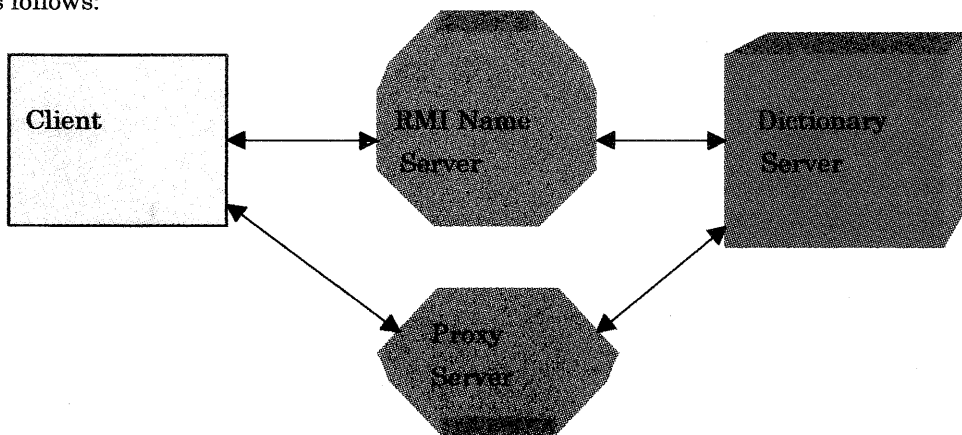
JAVA-IDL is an implementation of CORBA in the JAVA language platform. JAVA-IDL includes OMG-IDL (idltojava) and IIOP. A simple experiment is depicted as follows:



A client requests a meaning of a word to the Dictionary Server which looks for several dictionary and reply to the client.

#### 4.3 An experiment of JAVA RMI

Remote Method Invocation (RMI) enables JAVA applications to communicate by ordinal method calls without special interfaces. A model of Dictionary Sever based RMI depicted as follows:



The following list compares between CORBA and RMI.

Capability	<i>CORBA (JAVA IDL)</i>	<i>RMI (over IIOP)</i>
Dynamic stub download	No	Yes
Pass by Value	No	Yes
Distributed Dynamic Discovery	Yes	Yes
Distributed Dynamic Invocations	Yes	Yes
Multiple Transport Support	Yes	Yes
URL Naming	No (ORB-dependent)	Yes
Firewall Proxy	No (ORB-dependent)	Yes
Language Independent	Yes	No
Language Neutral Wire Protocol	Yes (IIOP)	Yes (IIOP)
Persistent Naming	Yes	No
Wire-level Security	Yes (CORBA Security)	Yes (SSL by JDK1.2)
Wire-level Transaction	Yes (CORBA OTP)	Yes (JTS)
Services	Yes (naming Service)	Yes (registry)

## 5. Conclusions

We implement simple experiments of the Internet oriented applications by using DCOM, CORBA(JAVA-IDL) and JAVA-RMI.

## 6. References

- [1] Defense Information System Agency, “*Recommendations for Using DCE, DCOM, and CORBA Middleware*”, April 13, 1998.
- [2] G. Eddon, H. Eddon, “*Inside Distributed COM*”, Microsoft Press, 1998.
- [3] D. Box, “*Essential COM*”, Don Box, Addison Wesley, 1998.
- [4] E. Gamma, R. Helm, R. Johnson, J. Vlissides, “*Design Patterns*” Addison Wesley, 1995.
- [5] G. Booch, J. Rumbaugh, I. Jacobson, “*The Unified Modeling Language User Guide*”, Addison Wesley, 1999.
- [6] D. Solomon, “*Inside Windows NT, Second Edition*”, Microsoft Press, 1998.
- [7] M. Abadi, L. Cardelli, ‘*Theory of Objects*’, Springer, 1996.
- [8] T. Mowbray, W. Ruh, “*Inside CORBA*”, Addison Wesley, 1997.