Sidebar – Middlewares for Distributed Computing

Davide Brugali

Universitá degli Studi di Bergamo, Italy brugali@unibg.it

1 Introduction

A framework for distributed computing consists of an integrated set of service components that allow distributed systems to operate together. Typically, distributed component frameworks offer at least the following services:

- Distributed event management supports dynamic notification of events raised by remote objects.
- Location-transparent access to remote objects allows distributed objects to cooperate regardless of their network location, of the operating platforms where they are executed, and of their implementation language.
- Distributed object location allows client objects to determine at run time which server objects offer the functionality they need. Client and server objects can appear and disappear on the network dynamically.
- Persistency and transaction management supports persistent storage, access to distributed data, data replication and data consistency.

Using a distributed component framework consists in developing end-user applications by delegating the execution of common functionalities to middleware services. The following distributed component frameworks are commonly used to build large-scale distributed systems.

2 OMG CORBA

The Common Object Request Broker Architecture (CORBA) (Vinoski 1997) is a standard for distributed middleware defined by the Object Management Group (OMG), a consortium of more than 700 organisations including software industry leaders such as Sun, HP, IBM, Microsoft and Rational. This architecture has reached a good level of maturity and is now implemented in more than ten commercial products.

D. Brugali (Ed.): Software Eng. for Experimental Robotics. STAR 30, pp. 395–398, 2007.

The basic component of the architecture is the Object Request Broker (ORB), which should be installed on each connected host. The ORB uses the Stub/Skeleton mechanism for remote communication between client and server objects. The stub and the skeleton of a server object are generated at compile-time from a declarative specification of the server's interface in the language-neutral Interface Definition Language (IDL). The interface describes which methods the server object supports, which events the object can trigger, and which exceptions the object raises. The server object can be implemented in any programming language (C, C++, Java, etc.). The ORB installed on the server side is in charge of translating the incoming IDL service requests from the remote clients into the server's method invocations. The IDL supports the declaration of only basic type arguments, which have to be passed to a remote server. The connection between the stub and the skeleton is established using the Remote Procedure Call (RPC) mechanism.

The ORB supports location transparency, as it provides the client with a reference to the server object regardless of its network location. The client side ORB dispatches service requests to the server side ORB transparently. Since CORBA can be implemented using different technologies, the Internet Inter-ORB Protocol (IIOP) defines the standard communication protocol for inter-vendor ORB compatibility.

The ORB supports the control of the threading policy used by the servers, such as one-thread-per request and one-thread-per-objects. This threading capability is at the basis of recent CORBA extensions towards a real-time ORB (see [FWDC00] for a comprehensive survey on recent results in developing a standard real-time CORBA).

Server objects can publish the IDL specifications of their interface using the Interface Repository API. The Interface Repository is used to implement two services for object location.

The *Trader Service* is similar to the yellow pages. It allows client objects to find out which distributed server objects support a given interface. The Trader Service returns a list of references to the objects, which have registered that interface and a description of the usage constraints for each object.

The *Naming Service* allows client objects to identify which interface is supported by an object that has been registered with a given name. In some cases, the client object has to request the service of a server object, whose interface was not known at compile-time. This means that the client object does not have a stub of the remote server object, but it can use a generic interface, called Dynamic Invocation Interface that makes it possible to construct method invocations at run-time [Vin97].

3 Sun Java

SUN Java Java is the Object Oriented programming language from Sun Microsystems [Joy00]. The most relevant characteristic (called Write Once/Run

Anywhere) is its portability: Java programs written on one type of hardware or operating system can run unmodified on almost any other type of computer. This is possible because Java programs are compiled in an intermediate format called byte-code that is interpreted by the Java Virtual Machine. The most important packages for distributed computing are:

The Remote Method Invocation (RMI) is the Java mechanism for remote object communication. It implements the Stub/Skeleton approach but, different to CORBA IDL, the server's interface is written in Java. The stub is not linked within the client's address space at compile time, but it is downloaded from the server side when the client needs a connection to the server object.

Applets are small graphical applications that run within a Web browser. When a remote user accesses a web page containing a reference to an applet, the applet's code is downloaded from the web site and executed inside the browser. The applet is not allowed to access the local resources on the client side (e.g. its file system), but it exchange data with the remote server.

Servlets are small applications that run within a Web server. The programmer implements specific servlets on the server side and gives them network names in the same way as is done for standard web pages. A remote user connects to a servlet's URL using a web browser. JavaServer Pages (JSP) are an extension of Servlet technology.

Enterprise Java Beans (EJB) are a component-based technology for developing server-side applications. The basic concept is the EJB Container, a database management application that provides persistence and transaction services to the enterprise beans.

IBM Aglets [LO98]. Aglets are active objects (mobile agents) that can suspend their activity, migrate to a remote host and resume their execution on that host. Before dispatch, the aglet serializes its internal state into a standard form. When the aglet arrives at the receiver side, it is assigned a new thread and executed. An Aglet belongs to an AgletContext, a container class that provides an interface for the runtime environment on the local host.

The *Sun JINI Platform* [Arn99] is an extension of the RMI framework. The stub (called proxy) is no longer a simple interface. Instead, it is a fully-fledged object, which embeds the co-operation protocol between the client and the server and executes (part of) the service logic within the client's address space. Clients download the proxy object from the corresponding device and are ready to use its services. The Jini platform implements the LookUp service in order to identify available services on the network.

4 Microsoft.NET

Microsoft.NET [TL01] is a new technology for developing distributed systems. It borrows many successful concepts from the Java world and from CORBA in order to achieve interoperability of heterogeneous applications. The following four aspects characterize the .NET framework:

The Common Type System (CTS) is an object model that extends the previous COM and DCOM models with the goal of supporting multiple language software development.
The Intermediate Language (IL) is an object-oriented language that conforms to the CTS. Various Microsoft and third-party language compilers (for C++, Java, etc.) generate code in the IL language.
The Common Language Infrastructure (CLI) is a run time environment (similar to the Java Virtual Machine) that executes code compiled in IL.
The .NET Software Development Kit (SDK) is an object-oriented collection of reusable components. It provides run time hosts for the CLI for a variety of execution platforms. Internet Explorer is an example of a run time host. It also supports the development of customer run time hosts.

The essence of the Microsoft proposal is the possibility of compiling existing code and new code written in the programmer's preferred language. The resulting applications can interoperate with class libraries and components written in different languages using the .NET run time environment.

.NET Remoting allows objects to interact over the Internet using binary encoding where performance is critical, or XML encoding where interoperability with other applications is essential. Similar to Java RMI, it is based on the Stub/Scheleton model: the stub (called proxy) is created at run-time using the metadata published by the server.

Web Services are reusable components which are used to develop server-side applications. They combine the distributed computing capability with the Web portal concepts and can be compared to the Java Servlets and JSP. The ASP.NET is the hosting environment for Web services. Clients use the ubiquitous wsdl.exe utility (a Web Service Directory supplied with the .NET SDK) to discover and find Web Services.

Serialization allows a memory object or graph of objects to be converted into a linear sequence of bytes that can be stored on a disk or sent to another networked computer. This mechanism is at the basis of the .NET persistence service that uses metadata information to automatically store and to reconstruct memory object.

References

[Arn99] Ken Arnold et al., *The jini(tm) specification.*, Addison-Wesley, 1999.
[Joy00] Bill Joy et al., *The java language specification.*, Addison-Wesley, 2000.
[FWDC00] V. Faye-Wolfe, et al., *Real-time corba*, IEEE Transactions on Parallel and Distributed Systems **11(10)** (Oct. 2000).
[LO98] D.B. Lange and M. Oshima, *Programming and deploying mobile agents with java and aglets.*, Addison-Wesley, 1998.
[TL01] Thuan Thai and Hoang Lam, *.net framework essentials.*, O'Reilly, 2001.
[Vin97] S. Vinoski, *Corba: Integrating diverse applications within distributed heterogeneous environments.*, IEEE Communications Magazine **14(2)** (Feb. 1997).