Generic tools: a new way to use Corba

Philippe Merle, Christophe Gransart, Jean-Marc Geib
Laboratoire d'Informatique Fondamentale de Lille - URA CNRS 369
Université des Sciences et Technologies de Lille
Bâtiment M3, 59655 Villeneuve d'Ascq Cedex - France
Tél.: (+33) (0)3 20 43 47 21 - Fax: (+33) (0)3 20 43 65 66
E-mails: {merle, gransart, geib}@lifl.fr

Abstract

This paper first presents Corba and our Generic Object-Oriented Dynamic Environment, second it discusses the interests of a generic environment and introduces our scripting language CorbaScript, and finally presents CorbaWeb as an integration tool between the WWW and Corba worlds.

Keywords: Corba objects, Generic Object-Oriented Dynamic Environment, Scripting Language, Objects in the World Wide Web.

1 Introduction

The goal of the GOODE project [1] is to study and design basic software allowing design, deployment and access to large scale distributed services. These services are composed of a set of distributed applications which cooperate on the client/server model. They demand to choose and to uniformize 1) models for resources design, 2) protocols for applications deployment and 3) flexible access mechanisms taking account users' evoluting activities.

Our current work studies generic tools (i.e. application independent) to navigate through large object spaces distributed on wide area networks. These proposals are tested on a Corba distributed object bus and on the World Wide Web infrastructure. Currently, two tools are available (http://corbaweb.lifl.fr): CorbaScript and CorbaWeb [MGG96a, MGG96b, MGG97]. CorbaScript is a scripting language to interactively exploit Corba objects. CorbaWeb is a generic gateway for Corba and WWW integration. It allows users to access and to navigate through object graphs using the Web user friendly interface. Moreover, it offers to designers a flexible environment to develop new Web services using object technologies from Corba.

2 GOODE: Generic tools to access to Corba services

In the OMG's architecture context [OMG96], the object oriented approach is used to design resources and the distributed objects bus is used for deployment and to establish connections between distributed applications. Objects modelizing services are specified with the interface definition language (IDL). Then, these interfaces are compiled to generate stubs for client and server components. These stubs are using the Object Request Broker (ORB) to transport requests through the network. This whole (stubs + ORB) can be compared with an object oriented RPC mechanism taking in charge heterogeneity problems: differences between hardware architectures, operating systems and implementing languages for programs.

[1] This work is supported by "Région Nord/Pas-de-Calais" in the R&D program on Advanced Communication "Ganymède".

Nowadays, service design and deployment with Corba are totally achieved using this "static" approach. Interactions between clients and servers are freezed at compilation time: clients contain stubs for a limited number of Corba object types which will be invoked at execution time. We consider that this approach is not well suited to support large scale generic services for which components must adapt themselves to information and business evolutions. With the static approach, when a service evolves, one must modify and redistribute server programs and also each client program distributed to users. As Corba is not easy to manage and use (for example: mapping rules to use stubs), these evolutions must be done by specialized programmers. All these points make that the use of the Corba static approach for general purpose services is not well suited.

To give an answer to this problem, we submit to use a dynamic approach where client programs are disconnected from services that they will use at execution time. Then, these clients are generic programs which adapt themselves, at execution time, to objects involved in the required service. Our user-oriented tools are built on top of the Corba dynamic mechanisms that are the dynamic invocation interface (DII) and the interface repository (IR). DII is an API for a generic stub which allows invocation of any object. The interface repository is an object database to discover IDL interfaces of objects available on the bus. Our proposal named GOODE (Generic Object-Oriented Dynamic Environment) has been tested and validated with two tools: CorbaScript and CorbaWeb.

3 CorbaScript: A scripting language for Corba objects

CorbaScript is a scripting language specially designed for Corba. It allows every user to access simply and interactively to objects available on the bus. Thus, a user is totally autonomous to exploit, administrate, configure, connect, create and destroy distributed objects on the bus.

This language is a high level tool allowing procedural, modular and object-oriented (classes/instances, multiple inheritance and polymorphism) programming. This language is interpreted and typing is checked dynamically at runtime. When a user invokes a Corba object, the interpreter checks type conformity for parameters with information extracted from the interface repository. These information are locally cached to improve future checks. Then, invocations are achieved by the DII generic stub. Moreover, IDL interface implementations can be done with CorbaScript classes. The CorbaScript interpreter uses the DSI mechanism (Dynamic Skeleton Interface) to trap and decode requests for objects implemented with CorbaScript.

The current implementation of the interpreter is done with C++ language (27000 lines) and uses the Orbix 2.0 product as Corba bus. It is composed of more or less 100 C++ classes in 3 modules: the ObjectScript module is the interpreter nucleus (syntax and execution), the CorbaScript module realizes the interface repository cache and calls to Corba dynamic mechanisms (DII and IR) and finally, the CorbaTools module to encapsulate specificities of the ORB. We are porting our tools on HP Orb+ and on OmniBroker. The interpreter code is modular and extensible allowing the add of new functionnalites in the CorbaScript language.

This approach for Corba has also been evaluated in some other environments like ILU+Python [Jan95] and TclDii [ASG+94]. ILU is a compatible Corba bus developped by Xerox Corporation. Python is an interpreted language from the CWI at Amsterdam which can exploit the ILU bus. The link between ILU and Python allows one to access interactively to Corba objects. But, this environment is based on the static approach: Python stubs are pre-generated, so type discovering at execution time is not allowed and application evolutions are not easy to do. TclDii is an interface between the Tcl interpreted language and Corba. It uses the dynamic approach DII from Corba. But it does not use the interface repository so this tool doesn't realize type checking and is complex to use for a non Corba expert.

4 CorbaWeb: Access to Corba objects from the WWW

To offer access to large scale information systems, we have developped the CorbaWeb environment. This environment allows Web browsers (taking in charge the computer human interface) to consult and apply operations on any Corba objects. CorbaWeb is a generic gateway between these two worlds. Users can access and navigate through object graphs using the Web user friendly interface. Moreover, it offers to software designers a flexible environment to build new Web services using Corba object technologies. The gateway is composed of a set of CorbaScript scripts to:

- view the state of an object or a graph of objects,
- navigate through object graphs,
- invoke operations on objects,
- view the functional interface of an object and
- migrate mobile scripts using the Web.

Users exploit Corba objects using documents (HTML and MIME) dynamically generated. These documents can be object views, forms and Java applets to invoke operations and works on these objects. The relations between objects are shown as anchors. The URLs are generated on the fly to allow navigation.

The link between the Web HTTP bus and the CorbaScript language can be done using several ways ; for example using the standard CGI protocol (Common Gateway Interface). We have tested two implementations of CorbaWeb. The first one is done using the CGI protocol and the second one is realized with a specialized HTTP server. With these two versions, we are able to evaluate the worst and the best performances of CorbaWeb.

Many related works have also studied the mix of Web and objects. AnsaWeb [ER94] studies the interoperability between Web protocols (HTTP, FTP, ...) and Corba protocols (IIOP). A lot of Web servers are now extensible using object-oriented software components (for example the Sun Jeeves server [SMI96]). The Web* environment [ASM+95] uses dynamic documents including some TclDii code (as interface with Corba) in HTML documents. Some others are based on mobile code languages which can be used as implementation languages for IDL specifications (e.g. Java with OrbixWeb [ION96]). All these works are the first steps for future distributed object-oriented infrastructures for large scale services on Internet. The main idea of CorbaWeb is to offer a generic solution based on a dynamic approach to solve deployment problems.

5 Conclusion: A new vision of large scale client/server

All our work (study and implementation) has permitted us to test and to validate a new vision of large scale client/server through generic and dynamic tools. The GOODE environment, composed of the CorbaScript language and the CorbaWeb environment, gives some answers to solve problems encountered during the life cycle of a large scale service: the design, the deployment and the use [Mer97]. Moreover, this environment is open and based on standardized technologies: Corba and WWW.

Currently, we are working on some other related topics: a mechanism to control access to objects and another one for notification (using the related Corba services and also with smart card technology [MVD96]). Resources management in a distributed environment is difficult, we are going to study script migration between CorbaWeb sites to avoid network overhead. We are beginning a new work to define a component integration language which will be used with IDL to describe services and links between them.

References

[ASG+94] George Almasi, Anca Suvaiala, Cristian Goina, Calin Cascaval, and V. «Juggy» Jagannathan. Tcldii: A tcl interface to the orbix dymanic invocation interface. Technical report, Concurrent Engineering Research Center (CERC), 1994. <URL:http://www.cerc.wvu.edu/dice/iss/TclDii/TclDiiDoc.ps>.

[ASM$^+$95] Georges Almasi, Anca Suvaiala, Ion Muslea, Calin Cascaval, Tad Davis, and V. «Juggy» Jagannathan. Web* - a technology to make information available on the web. In *Proceedings of the Fourth IEEE Workshop on Enabling Technology: Infrastructure for Collaborative Entreprises: WETICE'95*. Concurrent Engineering Research Center (CERC), 95. <URL:http://webstar.cerc.wvn.edu/>.

[ER94] Nigel Edwards and Owen Rees. Distributed objects and the world wide web. The AnsaWeb Project, Document Identifier APM.1283.00.08, Architecture Projects Management Ltd, Cambridge, U.K., 1994. <URL:http://www.ansa.co.uk/phase3-doc-root/sponsors/APM.1283.00.08.html>.

[ION96] IONA. *OrbixWeb: Programmer's Guide*. IONA Technologies Ltd., 8-34 Percy Place, Dublin 4, Ireland, 1996. <URL:http://www.iona.com>.

[Jan95] Bill Janssen. *Using ILU with Python: A Tutorial*. Xerox Palo Alto Research Center (Xerox PARC), May 1995.

[Mer97] Philippe Merle. *CorbaScript - CorbaWeb : propositions pour l'accès à des objets et services distribués*. Thèse de doctorat en informatique, Laboratoire d'Informatique Fondamentale de Lille (LIFL), Université des Sciences et Technologies de Lille (USTL), France, January 1997.

[MGG96a] Philippe Merle, Christophe Gransart, and Jean-Marc Geib. CorbaScript and CorbaWeb: A Generic Object-Oriented Dynamic Environment upon CORBA. In *Proceedings of TOOLS Europe'96, Palais des Congrès, Paris, France, February 26–29, 1996*. Prentice-Hall, February 1996.

[MGG96b] Philippe Merle, Christophe Gransart, and Jean-Marc Geib. CorbaWeb: A Generic Object Navigator. In *The Fifth International World Wide Web Conference (WWW5), Paris, France, May 6–10, 1996*, volume 28 (7-11), pages 1269–1281, Paris, France, May 1996. Elsevier, Computer Networks and ISDN System.

[MGG97] Philippe Merle, Christophe Gransart, and Jean-Marc Geib. How to Make Corba Objects User-Friendly with a Generic Object-Oriented Dynamic Environment? In Max Mühlhäuser, editor, *Special Issues in Object-Oriented Programming, Workshop Reader of the 10th European Conference on Object-Oriented Programming, ECOOP'96, Workshop on Putting Distributed Objects to Work, Linz, Austria, July 1996*, pages 222–229. Dpunkt Verlag, 1997.

[MVD96] Philippe Merle, Jean-Jacques Vandewalle, and Eric Dufresne. Intégration de systèmes hétérogènes : World Wide Web, Carte à microprocesseur et Corba. In *XIVe congrès INFORSID*, pages 235–248, Bordeaux, France, June 1996.

[OMG96] Object Management Group. The Common Object Request Broker: Architecture and Specification - revision 2.0. OMG TC Document 96-7-X, Object Management Group, Headquaters, 492 Old Connecticut Path, Framingham, MA 01701, U.S.A., July 1996.

[SMI96] Sun Microsystems Inc. Jeeves alpha2 documentation, 1996. <URL:http://java.sun.com/products/jeeves/alpha2/doc/index.html>.