CORBA Scripting in HEP and Beyond

S. Dû[1], *O. Lodygensky*[1], *P. Merle*[2]

[1] LAL-in2p3, Universite Paris-Sud, Orsay, France
[2] LIFL, Laboratoire d'Informatique Fondamentale de Lille, France

Abstract

In this paper, we present our CorbaScript experiments around CORBA applications in BaBar **based on a C++ data server and Java clients. CorbaScript is a new scripting language dedicated to CORBA and its specification IDLscript will be part of the coming CORBA 3.0 standard.

In our context, CorbaScript is not only used for prototyping and testing but also for implementing effective CORBA services and management tools. One of the unique features of CorbaScript is the natural access to CORBA dynamic interface. With the CORBA introspection mechanisms, this allows us to build completely generic services which can handle any CORBA object.

CorbaScript is developped by LIFL in an OpenSource way, allowing the growing users community to extend and enhance a basic set of library modules. The HEP community could benefit more widely from these efforts and maybe contribute to enhance the standard.

Keywords: CORBA, Genericity, Scripting, CorbaScript, OpenSource

1 Introduction

CorbaScript [1, 2] is an answer from the LIFL (Laboratoire d'Informatique Fondamentale de Lille, France) to the Object Management Group Request For Proposals for a CORBA Scripting Language. It has been available since June 1997 and BaBar is using it since February 1998. Our first experiments were presented at CHEP'98 [3]. CorbaScript enhances the developer's productivity in prototyping and testing. It allows one to extend simple Client-Server applications toward much richer and more complex systems.

In this paper, we present new CORBA generic tools, written in CorbaScript, which allowed us to enhance our distributed systems while making them simpler and easier to evolve and maintain. Section 2 briefly describes CorbaScript. Section 3 presents the BaBar distributed applications system. Section 4 gives a description of our developped CORBA generic tools and discutes their uses in our BaBar distributed application context. As conclusions, we summarise our experiences.

2 CorbaScript

2.1 Highlights

CorbaScript is an interpreted scripting language dedicated to CORBA. It runs on top of any C++ CORBA implementation which fully supports CORBA dynamic mechanisms (Interface Repository, Dynamic Interface Invocation, Dynamic Skeleton Invocation) and can interact with any CORBA objects through IIOP (the OMG's Internet Inter-Orb Protocol).

[**] BaBar is an HEP experiment at SLAC, Stanford Linear Accelerator Center, USA

CorbaScript can discover dynamically any kind of CORBA object: There is no need for stubs or skeletons. The only requirement is to load the objects' interface definitions in the CORBA Interface Repository. Moreover, you can use CorbaScript not only as a client but also to implement CORBA objects and services.

The current CorbaScript release [4] is available on top of the most used ORB products: MICO 2.3.1, OAK 5.3.4, omniORB2 2.8.0 , ORBacus 3.2.1, Orbix 2000, TAO 1.0.12, and Visibroker 3.3. It can be compiled (and then run) on Linux, Solaris, HP-UX, Windows 95/98/NT, AIX, IRIX operating systems with various C++ compilers.

2.2 IDLscript and CORBA 3.0

CorbaScript was accepted by the OMG in August 1999 and will be part of the CORBA 3.0 specification [5] under the name ”IDLscript” [6]. Now, CorbaScript can be considered as the first implementation of this standard.

3 BaBar Distributed Systems

3.1 BaBar Applications Overview

- Servers:
  - Standard BaBar Framework C++ applications built with I/O modules and reconstruction sequences deliver either a fully reconstructed event or tracks summaries.
  - A new module converts the components of a BaBar Event into CORBA objects or structures. A BaBar CORBA Event Server is obtained by appending that module to the application framework.
- Clients:
  - WIRED [7] is a general purpose Event Display client written in Java. It has been adapted to the BaBar environment.
  - The BaBar Event Browser [1] allows one to inspect complex data structures which have no graphical representation.
  - These 2 client applications will collaborate in a near future.
- Factories:
  - They are responsible to launch C++ Servers on different DataBases.
  - They manage a pool of Servers ready to be used by clients.
  - The pool parameters (max clients, min free servers) can be dynamically changed.

3.2 Application Deployment with CorbaScript

All the tools to administrate and test applications are developped in CorbaScript allowing them to be upgraded quickly and easily according to the project's evolution.

- Monitor:
  - a persistant service with recovering facilities,
  - registers the Factories and Servers states, and
  - launches new Factories working in various modes.
- Testing:
  Since the early stages of the project, various scripts have been heavily used to easily test the C++ servers. When a detector expert extends the server, he/she only needs to write a few lines of CorbaScript to get and check its new Data.

[1] The BaBar Event Browser is developped at LAPP-Annecy by A.Bazan, D.Boutigny, T.Leflour and S.Lieunard.

- Administration:
  A small shell script allows us to manage interactively the various Factories parameters, to launch new Factories or Servers and to recover dead components.

All these tools have been extended and simplified by using generic tools developped in CorbaScript.

4 CORBA Generic Tools

Generic Tools [1] are aimed at performing operations on any CORBA object. By definition, they are completely re-usable in various applications contexts.

4.1 CORBA and Genericity

CORBA provides mechanisms to discover dynamically what is the interface of any unknown object as long as its description is loaded in the Interface Repository, a standard CORBA service.

Introspection can be done from any language: it is difficult from C++, easier from Java and completely obvious from CorbaScript. Moreover, as a dynamic interpreted language, CorbaScript allows one to invoke the operations of an unknown object.

4.2 Generic Modules and Tools

4.2.1 Basic CorbaScript Modules

CorbaScript modules are useful scripts aimed at being used from other scripts. They provide the same kind of services as a library. CorbaScript is distributed with a set of modules which give access to the posix and standard C libraries. Other modules facilitate IO handling, object access from IOR/URL references or the Name Service, etc.

CorbaScript users can contribute to the development of modules which are validated by the LIFL team and published in the standard distribution [4].

4.2.2 New Generic Modules

These modules developped at LAL-Orsay are built on top of basic modules. They will be distributed after validation by LIFL.

- `nsTools`: easy access and administration of the Name Service.
- `irTools`: discover easily any object interface from the Interface Repository.
- `Cloner`:
  - any remote CORBA object is duplicated in a local instance. The clone's operations are implemented by invoking remote object operations.
  - makes a local instance of any CORBA object description. A living but inactive object is created: its state is conform to the description, it will implement all the expected operations but they will be inactive. There is still no Implementation Repository to decide how the operations must be implemented.
  - uses `irTools`.
- `ioTools`:
  - reads/writes the state of any CORBA object into a file.
  - restores any CORBA structure from a file.
  - if CORBA objects are dead, a living but inactive object can be restored using the previous `Cloner` module.

4.2.3 Generic Browser

The CorbaScript distribution provides the powerful CorbaWeb environment. This environment provides a generic gateway to integrate CORBA and World Wide Web worlds. More precisely, CorbaWeb allows CORBA users/programmers to display objects attributes in HTML pages and to invoke their operations through HTML links and forms. Then the association of standard WWW browsers and the CorbaWeb environment provides a generic CORBA browser.

We use CorbaWeb with our new generic modules to interact more easily with any CORBA objects.

4.3 Generic Tools Usage in BaBar

4.3.1 Cache, Import, Export

We have presented during CHEP'98 [3] a project to develop generic services aimed at facilitating the data migration on wide area networks: any CORBA objects are converted to structures which are faster to export and restored in local caches.

These services are not really used today: data transferts are done essentially by cartridges and servers trend now to deliver structures instead of objects.

Anyway, they are effective, ready to be used and very easy to maintain: they are based on generic modules and depend only on the C++ server interface definition.

4.3.2 Testing, Dumping

Regression tests can be produced and automatized without any change inside the server code. There is no need for additional dump methods: simple scripts based on `ioTools` module produce reference outputs.

4.3.3 Playback

A generic Playback Server can create and deliver CORBA objects from the reference outputs described above. It uses the `ioTools` and `Cloner` modules.

A client interested exclusively by structures or objects attributes does not see any difference with the original server. This is very useful for Client development which becomes independant from the real server and DataBase servers states.

4.3.4 Recovering and Configuration

Generic modules like `ioTools` and `Cloner` are used to implement recovering facilities:

- Monitor can restart all dead Factories, restore their state, reconnect all servers still alive and restart dead servers.
- It offers the facilities of a ”Session Manager”: a whole configuration with many Factories and Servers connected to various DataBases can be restored automatically.

4.4 Towards Components

- Generic DataFilter service:
  An interface derived from the original C++ Server offers now an operation *void doFilter (in string key, in string mobileCode);* This interface, implemented in CorbaScript, provides a Filter Service, able to apply selections on any CORBA object.
  By connecting a client to the Server through the Filter service, we can extend the original distributed application without changing anything in the Clients' or Servers' code.

- Scripts are a “software glue”:
  Initial simple client/server applications have been extended with filtering and persistance features without changing the original applications. The key is to dynamically plug new services on the existing CORBA objects. Those are not really CORBA Components [8] as they are specified now by the OMG.

  According to our experience, the CorbaScript language is a key to build “glue elements“ to combine and extend distributed applications.

4.5 Conclusions

In this paper, we summarise our various CorbaScript experiments and developments around CORBA applications for the BaBar project. These experiments show that CORBA generic tools are an important key to speed up the development of complex distributed systems by extending simple ones.

Then, we argue that this kind of **Genericity is a key for reusability**. CORBA Components are another key for reusability. In this context, CorbaScript is also a powerful tool to design and implement component-based systems.

The specification of CorbaScript is now an international OMG standard. Its modules library is growing fast with contributions from various domains. HEP experiments could benefit greatly from this OpenSource activity coordinated by LIFL.

References

1 Philippe Merle, Christophe Gransart, Jean-Marc Geib, “CorbaScript and CorbaWeb: A Generic Object Oriented Dynamic Environment upon CORBA” , Proceedings of TOOLS Europe 96, Paris, June 1996.

2 Philippe Merle, Christophe Gransart, Jean-Francois Roos, Jean-Marc Geib, “CorbaScript: A Dedicated CORBA Scripting Language” , CHEP’98, Chicago, Autumn 1998.

3 Serge Dû, Philippe Merle, “What we can get from a CORBA Scripting language in HEP?”, CHEP’98, Chicago, Autumn 1998.

4 LIFL, “CorbaScript Home Page”, `http://corbaweb.lifl.fr/CorbaScript/`.

5 OMG, Recently Adopted Specifications, CORBA 3.0:
`http://www.omg.org/techprocess/meetings/schedule/tech2a.html`

6 Object Oriented Concepts, LIFL, “CORBA Scripting - Joint Revised Submission” , OMG TC Document orbos/99-07-17, Object Management Group, August 1999.

7 M.C. Coperchio, M. Donszelmann, N. de Groot, P. Gunnarsson, M. Litmaath, D. McNally, N. Smirnov, “ WIRED - World-Wide Web Interactive Remote Event Display ”, Computer Physics Communications 110 (1998), 155-159.

8 BEA *et al.*, “CORBA Components: Joint Revised Submission” , OMG TC Document orbos/99-07-{01..03,05} orbos/99-08{05..07,12,13}, Object Management Group, August 1999.