IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4693

Adapting UP for CORBA Application Development

Bencomo, N. Matteo, A.
Laboratorio TOOLS – Centro ISYS
Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela
Apdo. 48093, Los Chaguaramos 1041-A, Caracas, Venezuela.
Phone number: 0058-212-6051659, Fax number: 0058-212-6051131
e-mail: nelly@acm.org, amatteo@isys.ciens.ucv.ve

ABSTRACT

The design of distributed applications in a CORBA based environment can be carried out by means of an incremental software process, which starts from requirements specification and leads to architectural design and implementation. In such a process, activities related to communication and integration mechanisms defined in the CORBA standard have to be executed. This paper discusses an adaptation of the Unified Process (UP) for the developing of CORBA applications. The adaptation proposes a clear traceability from the Use Case model through Analysis, Design, Deployment and Implementation models.

INTRODUCTION

During the past decade, distributed computing has gained increasing importance in the Information Technology domain. One of the most important results in this field is the Object Management Group (OMG) Common Object request Broker (CORBA) [13]. CORBA provides the mechanisms by which objects transparently make requests and receive responses. The heart of the architecture is an Object request Broker (ORB) which fulfils this function. The OMG has also specified a complete architecture (OMG/OMA) addressing both general issues and particular needs of application domains (e.g., Telecom, Banking, and Medicine) by defining high level library modules and frameworks. However, OMG mainly addresses the technological aspects in O-O distributed computing with little emphasis on the development process or notation.

On the other hand, the concept of software process has evolved over the past few years. Nowadays, it is accepted that a process has to focus not only on such aspects as workflows, devices, and activities, but also on other aspects such as the nature of the company and technology to be used in the project. Generally speaking, a process is defined to be adapted by an organization considering its culture, social dynamics, abilities, clients, complexity of the project, tools, techniques, and the application domain, among others. In O-O software, a generic process known as Unified Process (UP) [7] has been defined. UP can be adapted by taking into account the aspects mentioned above.

This paper reports our experience in specifying an adaptation of UP for the development of CORBA applications. The adaptation focuses on use cases, which are used to establish a clear traceability among analysis, design, deployment and implementation models. In this sense, control objects of the analysis have a direct correspondence with distributed components in implementation and deployment models. Currently, the adaptation affects Analysis, Design and Implementation workflows by tailoring and introducing activities and artefacts. We are also interested on studying how Requirements workflow can be adapted to fulfil CORBA applications development.

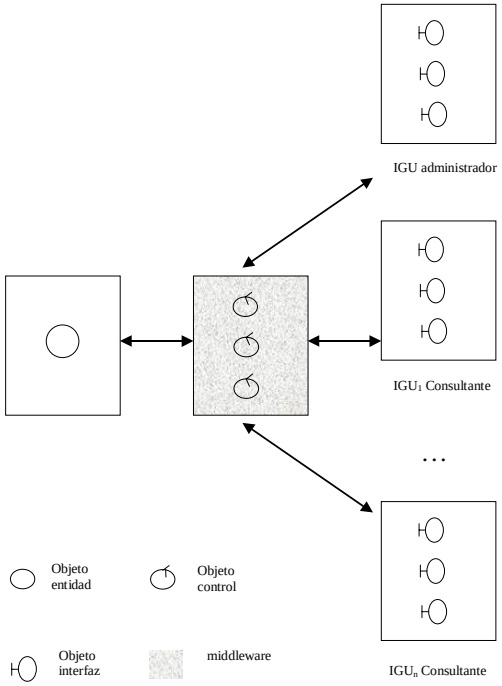
The reminder of this paper is organized as follows: in section 2 we present an overview of the adaptation and section 3 draws some preliminary conclusions and the research directions we plan to explore in the future.

ADAPTATION

Overview

Figure 1 shows an overview of the adaptation. CORBA icons highlight the activities affected by the adaptation. CORBA has a strong influence on the architecture of a system; this is the reason why Architectural Design activity is adapted in all workflows. In the Use Case Analysis activity, control classes responsible for remote communication are identified and specialized. These control classes are given the name of Remote Communication Control Classes (RCCC) and are the abstractions of components in charge of remote communication in the implementation. In Use Case Design, Subsystem Design and Class Design activities, the design classes corresponding to the specialized control classes in analysis and their IDL interfaces are specified. Finally, in Subsystem Implementation and Class Implementation activities, are the steps related to the implementation of the IDL interfaces in design, and the mapping of components onto nodes.

Figure 1. Adapted workflows and activities



1.1 Workflows

In this section we briefly explain the adaptations done in the Analysis, Design and Implementation workflows.

Analysis

The adaptation is tailored to aspects related to control classes. A control class plays the role of intermediary. The adaptation mainly proposes the iden-

tification of control classes responsible for remote communication among the analysis classes; this has to be done for each relevant use case in the Use Case Model. These RCCCs are specializations or adaptations of common control classes in the Use Case Model.

Design

In Design, some classes can be initially sketched from analysis classes; this is the case of design classes that deal with remote communication. A RCCC associated with a use case i in analysis will correspond to a pair of design classes, see Figure 2. Corresponding objects related to this pair of design classes will be intermediaries for remote component communications in the implementation.

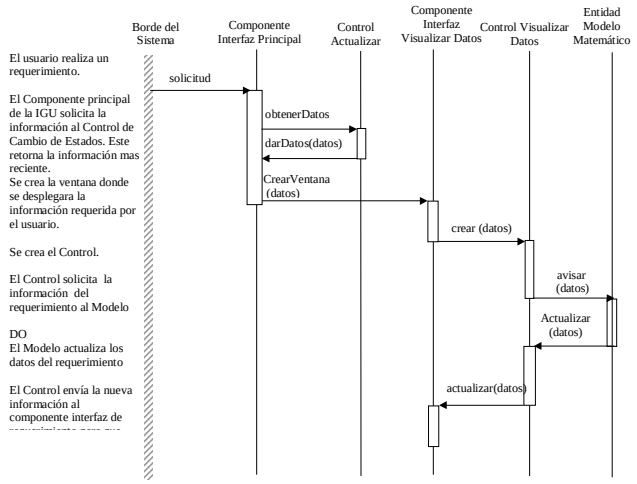
Each design class in Figure 2 traces to a CORBA object in the implementation. These classes expose basically two kinds of interfaces; one is a common UML semantic interface and the other is an IDL interface. IDL interfaces let CORBA objects in the implementation send/receive the (remote) messages. The common UML interfaces let components on each end to send/receive the messages without taking care of CORBA communication mechanisms.

The methods of both kinds of interfaces are specified using the interaction diagrams that were specified early in design.

Implementation

In implementation, we have to program the code associated with CORBA objects and components based on the IDL interfaces in design. Currently, we are working on the definition of CORBA Components Diagrams to describe the functionality and interactions among CORBA components.

Figure 2: Trace relationship between Use Case Model, Analysis Classes (RCCC) and Design Classes



3. CONCLUSIONS

Currently, the specification of the adaptation is almost finished, leading to a formal adaptation named CORBAdapted-UP. The adaptation is based on practical experience, [1][2][3]. One such successful experience has been the development of a distributed graphical interface for a Real-Time system.

One of the issues we are refining is the formal notation of new stereotypes, artefacts and diagrams related to CORBA to be included in UP. Another issue is the application of this adaptation for the development of new applications based on legacy applications.

We are also extending the adaptation to include Requirements workflow and to specify a list of parameters tailored to CORBA application development.

One of the most difficult issues in this work has been describing our adaptation of a process that is so detailed. For this reason, we are also interested on developing a formal specification of a template UP adaptation specifications.

4. REFERENCES

Bencomo, N,: Interfaz Gráfica Basada En Objeto Distribuidos Para Un Sistema En Tiempo Real, Tesis de Maestría, Postgrado en Ciencias de la Computación, UCV, Venezuela, (1998)

Bencomo, N., Blanco, A., Correa, E., Matteo, A.: An experience using OOSE for a Graphical Multi-User Interface based on Distributed Objects, 5th. International Conference on Information Systems Analysis and Synthesis ISAS'99, Proceedings de la Conferencia, Volumen II, Orlando, Florida, USA, (1999)

Bencomo, N., Correa, E., Matteo, A.: An experience using CORBA and OOSE in the construction of a Graphical multi-user Interface based on Distributed Objects, Proceeding of Practical Experience Segment of Objetos Distribuidos 2000, Sao Paolo, Brazil, (2000)

Bencomo, N., Correa, E., Matteo, A.: Comparación de CORBA y X Web en el Desarrollo y Desempeño de Una Interfaz Gráfica Multiusuario, Publicado en Memorias XXVI Conferencia Latinoamericana de Informática CLEI200, (2000)

Finkelstein, A., Kramer J., Nuseibeh, B.: Software Process Modelling and Technology, Research Studies Press LTD (1994)

Fowler, M., Scott, K.: UML Distilled Applying the Standard Object Modelling Language, AW, (1997)

Jacobson, I., Booch, G., Rumbaugh J..: The Unified Software Development Process, Addison-Wesley (1999)

Jacobson, I., Magnus, C., Patrik, J., Gunnar, O.: Object-Oriented Software Engineering: A Use case driven Approach, Addison-Wesley (1993)

Kendall, S.: Modern Systems Analysis & Design, Prentice Hall, Tercera Edición (2002)

Leszek, A. Maciaszek: Requirements Analysis and System Design Developing Information Systems using UML, ADDISON WESLEY, (2001)

OMG Unified Modelling Language Specification, Version 1.4, OMG, (2001)

Shatz, S.: Developing of Distributed Software, Macmillan Publishing Company, (1993)

Slama, D., Garbis J., Russelm P.: Enterprise CORBA, Prentice Hall, (1999)

Stevens, P., Pooley, R.: Using UML Software Engineering with Objects and Components, AW, (2000)