A Computational Model for a Distributed Object-Oriented Operating System Based on a Reflective Abstract Machine

Lourdes Tajes-Martínez, Fernando Álvarez-García, Marián Díaz-Fondón, Darío Álvarez-Gutiérrez, Juan Manuel Cueva-Lovelle

Department of Computer Science, University of Oviedo
{lourdes, falvarez, fondon, darioa, cueva}@pinon.ccu.uniovi.es

Abstract. The design of an object-oriented operating system (OOOS) involves the design of a model that governs the objects method execution. In this paper we show the design of an OOOS based in an OO abstract machine: specifically, the design of the computational model. We propose the adoption of an active object model and we think *reflectivity* is a helpful tool to achieve a flexible OO computational system.

1 Introduction

The aim of the OVIEDO3[1] [1] project is to develop an OO integral system where every layer is designed and developed using the OO paradigm. The two lowest layers are an abstract machine, named *Carbayonia*, and an Operating System (OS), named *SO4*. The computational system of SO4 defines an *active object model* and extends the default behavior of Carbayonia in some areas by means of *reflectivity*.

2 The Computational Model: Reflectivity

Carbayonia is the lowest support level. It provides objects with the most basic mechanisms for the execution of their methods:

1. Basic classes. Their execution is atomic, without interruption, and synchronization is not necessary. The *Thread* class is one of the basic classes.
2. Instructions for method invocation (*call*) and exception handling (*handle*, *throw*).

SO4 provides objects with the mechanisms needed to define the behavior of their environment and modify or extend it if needed. We think reflectivity is a fundamental concept to get it. SO4 defines a set of OS classes and extends the *Object* class defined

[1] This work has been supported in part by the 2nd plan (FICYT) of research of the Principado de Asturias, Spain, under project PBP-TIC-97-01 "Sistema Integral Orientado a Objetos: OVIEDO3"

by Carbayonia, allowing user objects to construct some aspects of the execution environment. There are some projects, like ApertOS [2] and Merlin [3], studying the use of reflectivity to design and construct OS, abstract machines, languages, ...

3 Object Environment

Reflectivity is used to extend the machine behavior and is expressed attaching to an object a set of objects, which will be named the *object environment*.

The key idea in the design will be to divide the object world in two levels: *base level* (where base objects exist) and *meta-level* (where meta-objects exist). Each one of the meta-objects will describe some aspect of the base-level behavior of the object. Control is transferred to the meta-object when some specific events (method invocation, exceptions, ...) happen.

In the first stage of our system, the object environment is composed by the following meta-objects:

1. *Concurrence*: It defines the synchronization policy specific to the object.
2. *Scheduler*: It defines the scheduling policy the base object applies to its tasks.
3. *Communication*: It takes charge to send, receive and manage the messages.

4 Advantages of Using Reflectivity to Organize an OOOS

By offering the OS mechanisms as extensions of the abstract machine, a number of advantages are obtained: *Flexibility* [4] is the most important benefit and is achieved because each object can define its own environment using specific meta-objects and so, adapting the behavior of Carbayonia. *Uniformity* around the object concept because meta-objects are objects themselves. Besides, persistence and distribution of computation are straightforward extensions to traditional mechanisms.

References

1. Álvarez-García, F., Álvarez-Gutiérrez, D., Tajes-Martínez, L., Díaz-Fondón, M.A., Izquierdo-Castanedo, R., Cueva-Lovelle, J.M., “An Object-Oriented Abstract Machine as the substrate for an Object-Oriented Operating System”, Workshop on Object-Orientation and Operating Systems, 11th European Conference on Object-Oriented Programming (ECOOP’97), Jyväskylä (Finland), June 1997
2. Yokote, Y., The Apertos reflective operating system: The concept and its implementation. Proceedings of the 1992 Conference on Object-Oriented Programming Systems, Languages and Applications. ACM Special Interest Group on Programming Languages, ACM Press, October 1992.
3. The Merlin Project, URL http://www.lsi.usp.br/~jecel/merlin.html, November, 1996
4. Cahill, V., Flexibility in Object-Oriented Operating Systems: A Review, 3rd CaberNet Radicals Workshop, Connemara (Ireland), May 1996