Abstract Description of Distributed Object Systems*

Thorsten Hartmann
Ralf Jungclaus

Abt. Datenbanken, TU Braunschweig
P.O. Box 3329, D-W-3300 Braunschweig, FRG
E-mail: {jungclau,hartmann}@infbs.uucp

Abstract

In this paper we propose an abstract object-oriented model to describe distributed systems in an actor-oriented way. We introduce Basic Object Structures (BOS) with the basic concepts base objects to represent entities and channels as representations of communications between base objects. Objects are regarded as processes that can be observed. Based on these concepts we show how BOS can be uniformly described using an abstract language, the HOL-kernel.

1 Introduction

It is rather straightforward to regard a (distributed) system as a collection of dynamic autonomous objects that communicate. This approach dates back to [Hew77] and has initiated different approaches towards a computational model for objects. The ACTOR model of computation [Agh86] promotes asynchronous communication between actors, whereas in POOL-T [Ame87] objects are considered as processes that communicate synchronously. Other approaches are ABCL/1 [YSTH87], that models objects as active entities with a single thread of control which may communicate synchronously and asynchronously, or Abacus [Nie90, NP90b], that uses a computational model of communicating agents based on CCS [Mil89]. The behavioral description of systems of communicating agents has been a vivid field of research since the presentation of CSP [Hoa85] and CCS.

In this paper we propose a more abstract view of dynamic object systems. We regard objects as units of concurrency encapsulating a single process without internal concurrency (this is known as the trace model). The state of an object is described by a number of (structured) attributes that take values from predefined codomains. The attribute values are determined by the history of the object — in this way advantages of process models are combined with data modeling capabilities. Mathematical models of objects in our sense are described in [SE90, ES91].

*This work is supported by Deutsche Forschungsgemeinschaft under Sa 465/1-1 and Sa 465/1-2.
Based on this view we develop Basic Object Structures (BOS) to serve as an abstract view of loosely coupled distributed systems. The only concepts supported by BOS are simple (non-composite) objects and communication objects representing communication channels between simple objects. The goal is to provide a suitable interface to implement BOS on a network of workstations.

BOS are the basis for high level abstractions like specialization, generalization, temporary roles, and complex (composite) objects. These abstractions are supported by a language called ROL, particularly suited to the abstract description of interactive information systems [JSH91, JHSS91, SJE91]. Declarative ROL specifications will be decomposed into equivalent specifications of BOS described using a ROL-sublanguage, the ROL-kernel. In this sense BOS can be seen as a step towards implementation of distributed information systems. They provide a link between the conceptual model of the application area and the implemented system.

The advantage of this approach is basically the uniform language framework. In the process of describing a system we start with a consistent model of the application using language features of ROL. Then we will go on with transformations that do not violate the constraints given by this model. Each step can (at least in principle) be verified against the former step. The result of this procedure is a model of the system in terms of Basic Object Structures described with language features of the ROL-kernel.

In information systems, we typically have a huge number of persistent objects of which just a few are active (i.e. providing services and are given CPU resources) at one point in time [SE90]. Thus, the implementation of BOS will be supported by object stores managing the currently passive objects, object managers scheduling the active objects and several communication primitives. This viewpoint also provides a way to weaken the artificial boundary between active and passive objects.

The concept of object used as a basis for ROL and the ROL-kernel has been developed in [SSE87, SFSE89b, SE90] and [SFSE89a] which takes a temporal perspective of abstract object types and concurrency. This approach is accompanied by work on a categorical semantics [ES89, EGS90, ES91]. Based on this formal concept, work has been done towards a logical framework of structured theories over a suitable logical calculus [FSMS90, FM91].

The paper is organized as follows: In section 2 we describe the general concept of object in a formal representation. Then we introduce our notion of Basic Object Structures using a simple example of communicating objects. In the main section we describe the language features of the ROL-kernel used to model BOS providing a modeling of the simple communication example between a server and a possible client. In the last section we give some concluding remarks and state our future plans.

2 Formalization of Object-Oriented Concepts

In this section, we describe how a model, in the mathematical sense, representing a system as a collection of interacting objects is characterized. In our approach, emphasis is put on the representation of the behavior of objects through time and of their evolution depending on their behavior resulting in a complete representation of structural and