

Object-based System or Object-based Environment, that is the Question

Daniel Hagimont⁽¹⁾

André Freyssinet

Bull-IMAG/Systèmes, 2 av. de Vignate, 38610 Gières – France

Internet : {Daniel.Hagimont, Andre.Freyssinet}@imag.fr

Abstract:

In this position paper, we argue that it is painful and needless to develop an object based system that provides all the traditional functions of an operating system. A distributed object-oriented system should rather appear as a portable environment that cooperates with some well known popular systems such as the Unix system.

1 Introduction

Many research projects have attempted to provide a distributed object based system for the support of high level applications. Two main approaches have been followed:

- In the first approach, the implemented system provides a platform on which all the executed applications are structured in terms of object.
- In contrast, an object based system can be implemented as an environment alongside a popular existing system. This environment provides the support for object-based applications while many applications can still run on the traditional system.

In the following, these approaches are respectively called the "system approach" and the "environment approach". Since one of the critical problems of system implementors is to experiment with *real* users and applications, our claim in this position paper is that the environment approach allows the cohabitation and cooperation of a research prototype with a commonly used operating system, and that this is a strong argument to make a system accepted by potential users.

The remainder of the paper is organized as follows. Section 2 discusses respectively the advantages and drawbacks of the system and the environment approaches. Section 3 presents our experience from the Guide project. Section 4 presents our conclusions.

(1) This author is currently a post-doctoral invitee at the University of British Columbia, in the Department of Computer Science, 2366 MainMall, Vancouver, B.C. Canada V6T 1Z4. This position is funded by INRIA (Institut National de Recherche en Informatique et Automatique), Domaine de Voluceau – Rocquencourt, B.P. 105, Le Chesnay Cedex, France.

2 Discussion

One of the best arguments for the system approach is the provided freedom at design time. For instance the design of a system from scratch is very enthusiastic because many functions can be implemented at a very low level. The design of an environment that coexists with a traditional system requires at least the sharing of a common denominator between both, which limits the investigation field and makes more difficult the possible improvements on the environment.

The advantage of the environment approach is that most of the distributed systems users already use standard systems such as Unix with additional tools for network management (e.g. NFS). It makes more sense and it is more realistic to extend such a configuration with a distributed object-based environment, rather than to ask users for a full replacement of their system configurations. Moreover, applications that run in these two different worlds can cooperate, thus avoiding the re-implementation of many system services in the object-based environment. Furthermore, it seems preferable for such an environment to be portable, i.e. not to be dependent on some low level internal layers of the hosting system.

3 The Guide experience

Several prototypes have been implemented in the framework of the Guide project.

Although the first prototype (Guide-1 [1]) was implemented on top of the Unix system, its design followed the system approach: all the supported applications were developed with the Guide object-oriented language [3] and there wasn't any cooperation between the Guide and the Unix worlds at the application level. This implementation aimed at hiding the underlying Unix system. Moreover, a small part of the Unix system had to be modified for some performance reasons.

In a second step, we implemented a prototype (Guide-2 [2]) on top of the Mach 3.0 microkernel. Its design was closer from the environment approach, since this prototype allows applications either to call Unix (the OSF-1 single server on Mach) or Guide-2 (our environment) services. Applications run as Unix processes and are free to make use or not of the Guide-2 facilities. However, Guide-2 requires a OSF-1 Unix system based on Mach 3.0.

The industrial implementation which is under development in the framework of the Oode project is running on the AIX Unix system, that is the Unix of the Bull distribution. The integration in an existing product distribution is a strong requirement.

4 Conclusion and open issues

We think that the environment approach is the best suited for the system to be accepted by users. Therefore, potential users can see object-based services as additional facilities rather than as a new system generation.

The main problem is then to be able to improve object management by a low level implementation without compromising the compatibility with an existing system.

Bibliographie

- [1] R. Balter, J. Bernadat, D. Decouchant, A. Duda, A. Freyssinet, S. Krakowiak, M. Meysembourg, P. Le Dot, H. Nguyen Van, E. Paire, M. Riveill, C. Roisin, X. Rousset de Pina, R. Scioville and G. Vandôme, Architecture and implementation of Guide, an object-oriented distributed system, *Computing Systems*, 4(1), pp. 31–67, Hiver 1991.
- [2] P.Y. Chevalier, A. Freyssinet, D. Hagimont, S. Krakowiak, S. Lacourte, J. Mossière and X. Rousset de Pina, Persistent Shared Object Support in the Guide System: Evaluation & Related Work, *Accepted for publication at ACM Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA)*, Octobre 1994.
- [3] S. Krakowiak, M. Meysembourg, H. Nguyen Van, M. Riveill, C. Roisin and X. Rousset de Pina, Design and implementation of an object-oriented strongly typed language for distributed applications, *Journal of Object-Oriented Programming (JOOP)*, 3(3), pp. 11–22, Octobre 1990.