CORBA WRAPPERS FOR A-POSTERIORI MANAGEMENT

An Approach to Integrating Management with Existing Heterogeneous Systems

Steffen Lipperts and Dirk Thißen

Department of Computer Science (i4)
Ahornstr. 55, 52056 Aachen
Aachen University of Technology, GERMANY

{lipperts,thissen}@i4.informatik.rwth-aachen.de

Abstract: Network and system management aim at planning, surveying, and coordinating resources of communication networks and systems, in order to ensure availability and reliability of the systems and services involved. The structure of both networks and systems, however, is dramatically changing, and the overall complexity is rapidly increasing. Today, it is therefore aimed at finding new approaches to management, and the Common Object Request Broker Architecture (CORBA) has shown to offer new capabilities for management purposes. A critical issue in deploying CORBA as a management middleware is the integration of existing networks and systems into the management solution aimed at. This paper discusses the problems involved with the integration process of legacy applications, especially with regard to the management aspects. The concept of management wrappers is introduced and it is shown how it enables a flexible management approach based on CORBA.

Keywords: CORBA, management, wrapper, a-posteriori, mobile agent

1 INTRODUCTION

Due to the development of network and system technologies, the concept of distributed systems has become increasingly important. In the last years existing small networks have been integrated into bigger ones. This results in more complex and heterogeneous systems, which have additional requirements concerning availability and reliability of the systems and services involved. CORBA offers a suitable basis for handling the heterogeneity of these systems. For the same reasons, it can be used as a

L. Kutvonen et al. (eds.), *Distributed Applications and Interoperable Systems II*

basis for management applications. But CORBA is a concept which only has existed for a few years now; thus, most applications and systems have not been constructed and equipped with the capability of using CORBA. Often, when implementing a CORBA management system, this is ignored, as merely CORBA applications are considered. But there also is the problem of handling existing technologies with a CORBA solution. In this paper, the integration of legacy applications with a CORBA platform is discussed. The concept of management wrappers is introduced, which enables a CORBA manager to access legacy application in a uniform way.

The paper is structured as follows. In section 2, an overview of distributed systems' management is given. Besides standard management protocols like SNMP and CMIP, new concepts using CORBA and mobile agents are presented. Section 3 specifies solutions for adding management functionality to applications, and problems caused by off-the-shelf and legacy applications. To give a concrete example of CORBA deployment and the management involved, the management of development tools in chemical engineering is described. A solution for the identified problems is given in section 4, which introduces the concept of management wrappers. Finally, section 5 concludes the work and presents an outlook on future tasks.

2 MANAGEMENT OF DISTRIBUTED SYSTEMS

Two main protocols for system and network management exist today, the Simple Network Management Protocol (SNMP) and the Common Management Information Protocol (CMIP). Although both SNMP and CMIP provide or are extended by features to decentralise management, they still follow an overall platform-centric manager-agent paradigm. There are several factors, which make such an approach no longer feasible as systems grow increasingly complex. First of all, the amount of operational data that must be monitored and processed in real time is increasing dramatically. Therefore, bandwidth is becoming a critical factor when trying to transfer this information to managers and processing it there. In case of congestion due to heavy traffic, for example, additional notifications will be sent and the situation will deteriorate. Another problem is given with the information overflow at the manager. The required data needs to be explicitly requested and processed by the manager. In many cases, this can be more than the manager is able to handle.

Allowing the agents to pre-process information and merely send out notifications aims at freeing the manager from this flood of information as well as reducing the network load. However, trying in advance to include all code required in the agents will first lead to overweighted agents and eventually does no longer scale. The growing system complexity causes the need to observe countless parameters and to provide a continuously changing functionality.

Another problem draws from the changing nature of the underlying networks. As mobile communication is becoming an increasingly important part of today's applications, problems of intermittent connectivity need to be addressed. It has to be assured that the management functionality is executed reliably and correctly, even if the connection is temporarily lost. In client/server based approaches such as SNMP and CMIP, loss of connectivity is an immanent problem. Moreover, as systems and networks are becoming increasingly complex due to the operating systems, programming languages and protocols involved, a means for addressing this heterogeneity must be provided.

The overall conclusion therefore is that it is no longer necessary or useful to centralise management processing. The role of agents should no longer be restricted to collecting, pre-processing, and reporting instrumentation data [7,8]. Mechanisms are needed to delegate tasks from managers to agents and to allow agents to follow these tasks autonomously, which allows them to continue their work, even if the connection to the manager is temporarily lost. At the same time, issues of integrating existing applications into systems and networks must be addressed.

Work is in progress to enhance the client/server-based management and to define new approaches to a decentralised management [1,2,3]. A language-independent approach to management by delegation can be based on the concept of elastic servers. The main feature of such an elastic server is the ability to support translation and dynamical linking of delegated agents, i.e. delegated agents are executed as threads inside an elastic server on the destination host. In order to do so, elastic servers are realised as multithreaded processes with a program code and a process state that can be modified dynamically on execution. A three-layered architecture of the elastic processing runtime environment is presented in [4].

Another closely related approach aims at implementing delegated management services in CORBA environments as a base for intelligent agents [5]. Management functionality is isolated in objects, which are distributed across a CORBA-based environment and shared between managing and managed system as services. The delegation of agents then requires no special delegation protocol, but relies on CORBAservices, either the Externalisation Service, which is used for flattening objects into streams, or alternatively on the LifeCycle Service, which offers a move and copy operation for objects.

Similarly, the integration of mobile agent technology with CORBA aims at providing both a management middleware and the benefits of delegated tasks. In [6] we have presented an approach to a CORBA management based on mobile agents and have shown the efficient deployment of migration to reduce management complexity and load. We have also discussed the communication issues involved and how they can be mapped to CORBA mechanisms. In the next section, we will focus on the issues involved in the integration of applications into management systems, especially as part of an a-posteriori development of a management system, and will give an example how this is applied in the context of a heterogeneous, process-centred development environment for chemical engineering.

3 REQUIREMENTS SET BY THE INTEGRATION PROCESS OF LEGACY APPLICATIONS

In order to add management functionality to a system, which is being developed, there are several potential ways of integrating the management extensions with the original system.

- Direct integration: This is a very specific approach, in which the management code is directly embedded in the original source code. It allows full control over the application and provides all the information necessary for the management evaluation. Such an embedded approach, however, is not advisable, as it is application specific and very difficult to be maintained and modified.

- Interface extension: Objects in CORRBA have their interfaces defined in the Interface Definition Language (IDL), in order to hide implementation details and address heterogeneity of distributed systems. By extending the IDL specification with management functions, a modular approach is given. The management functionality provided by the object can be accessed by management components via the standardised interface and thus be modified without having to reconstruct the entire system. However, although the process of adding the code can be automated as part of the IDL-compilation process, it still requires the explicit modification of source code.
- Linking: To add management functions to an object at a later point, this can be done in the linking process. Pre-configured libraries containing the management part can be linked to the original application.

Although the last approach does not require an explicit modification of an object's source code, it still requires the code to be available for re-compilation. This, however, is not given with many off-the-shelf and legacy application. Such applications cause major problems and therefore, a different approach needs to be taken. One example given here tackles the work of the Collaborative Research Centre "Information Technology Support for Collaborative and Distributed Design Processes in Chemical Engineering" (CRC 476 IMPROVE). This project has the goal to enhance the productivity of developers and to improve the quality of design processes and products in chemical engineering by using support of database technology, software engineering, and communication systems management [7]. The focus of this project is on the early development phases of a chemical engineering process. Developers located at different departments and different companies are interworking in these phases. The range of tools deployed reaches from editors for the development of alternatives for processes, and simulators for reaction processes to tools for analysis. These applications are intended to interwork in a joint complex process, which causes a highly heterogeneous environment.

A co-ordinated interworking of tools is necessary, because the failure or blockage of one component must not defer the entire development process. Hence, bottlenecks and breakdowns have to be found and eliminated in the earliest possible way. Consequently, a service management system for managing the execution and interaction of the development tools and the supporting services is needed to guarantee high availability, fault tolerance, efficiency, and reliability. For the reasons mentioned in the last chapter, CORBA was chosen as basis for this system to handle the heterogeneity and to enable the co-operation of the development tools. However, trying to provide a management of the development tools, problems arise from the fact that an a-posteriori integration is to be made; since the source code for most development tools is not given, the approaches mentioned above cannot be used. Instead, a new approach is needed which is capable of adding a uniform management functionality to each of the development tools. Unfortunately the tools offer different capabilities for adding management functionality. Some provide OLE interfaces, which enable access over a CORBA-COM interface. But although the general approach is the same, no uniform access functions are given, because the OLE interfaces of the applications are different. Some applications offer CORBA-bridges, which usually provide limited access operations. Some applications, however, offer neither of these

two accesses. In order to provide a uniform management functionality which allows managers to access these applications via standardised interfaces and thus avoids proprietary and application specific management solutions, a new approach has to be chosen, which is presented in the following section.

4 THE CONCEPT OF MANAGEMENT WRAPPERS

For the integration of legacy applications with a CORBA-based management system, different approaches are possible. The simplest approach is to use only the management information which can be provided by the ORB. Such information could be the number of requests to a server or the response time of the server. This can for example be achieved via the Interceptors specified in the CORBA specification [8]. But this approach merely gives minimal access to servers, and only a limited set of management information can be used. Furthermore, this approach requires applications to provide a CORBA interface.

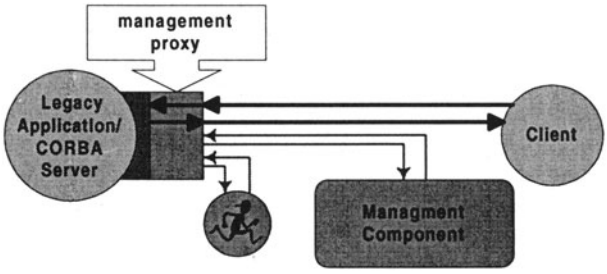

Figure 1. Flexible Management Proxies.

A more promising approach deploys the new concept of management wrappers, or management proxies, see figure 1. Such a wrapper hides the details of an application by encapsulating its interface and offering a new CORBA interface to clients and management applications. The requests made by clients are simply passed to the original interface of the application. Internally, the wrapper contains management functionality, which uses the application-specific interface to present a uniform CORBA interface to management applications. The manager itself does not need to adapt to each managed application; it has one general management interface, which is mapped to the original interface by the wrapper. This concept allows a manager to obtain more detailed management information and to perform more complex management actions by using the specific interface of an application and additional features of the wrapper, for example measurements, surveying of thresholds, and statistics. Thus, this concept is suitable as a-posteriori approach. Moreover, the uniform interfaces allow this concept to be deployed together with the new management approaches tackled in section 2. For instance, a mobile agent locally executing a management task can access the applications management data via the same methods as a central management component. Based on the needs in the CRC 476 IMPROVE, this

structure of the wrapper's interface was determined and the implementation started for the development tools discussed in section 3.

It must be taken into consideration that the introduction of the management wrappers adds an overhead to the underlying system. Therefore, this solution will cause an overhead concerning execution times. Although this is expected to be comparatively small, it has to be examined in the context of real time applications. Therefore, this issue will be addressed as work progresses, but more detailed information will not be available until the wrapper implementation for the development tools has actually been completed.

5 CONCLUSION AND OUTLOOK

In this paper, the concept of management wrappers for the integration of legacy applications with a CORBA-based management system was presented. While CORBA offers an appropriate basis for the management of distributed systems, which are implemented from scratch, the handling of legacy applications still remains a problem. The management wrappers presented here are used to realise a standardised management interface for such applications.

Currently, the interface structure of the management wrappers has been determined for the CRC IMPROVE, and the implementation of the development tools mentioned in section 3 has started. After finishing this implementation work, the next task is to evaluate the overhead caused by the deployment of the management wrappers. It is also intended to refine the structure of the management wrappers to contain two main parts. The first one is to contain generally necessary operations, which are valid for all applications. The specification of the second part is to address mechanisms, which are relevant for specific applications or groups of applications.

References

[1] BALDI M., GAI S., PICCO G. Exploiting Code Mobility in Decentralized and Flexible Network Management, *Mobile Agents, First International Workshop*, Berlin, April 1997.

[2] SAHAI A., MORIN C., BILLIART S. Intelligent Agents for a Mobile Network Manager (MNM) *IFIP International Conference on Intelligent Networks and Intelligence in Networks*, Chapman & Hall, Paris, France, September 1997.

[3] KAHANI M., BEADLE P., Decentralised Approaches for Network Management, *Comp. Commun. Rev.*, vol.27, no.3, July 1997.

[4] GOLDSZMIDT G., YEMINI Y., Delegated Agents for Network Management, *IEEE Communications: Management of Heterogeneous Networks*, Vol.36, No.3, March 1998.

[5] KELLER M., Service-based Systems Management Using CORBA as a Middleware for Intelligent Agents, *Proceedings of the IFIP/IEEE International Workshop on Distributed Systems*. L'Aquila, Italy, October 1996.

[6] LIPPERTS S., CORBA for Inter-Agent Communication of Management Information, *5th International Workshop on Mobile Multimedia Communication*, Berlin, Germany, October 1998.

[7] NAGL M., WESTFECHTEL B., *Integration of Development Systems in Engineering Applications (in German)*. Springer, December 1998.

[8] OBJECT MANAGEMENT GROUP, *CORBA 2.2 Specification*, http://www.omg.org/corba/corbaiiop.html.