Management of E-Commerce Brokerage Services

Jorge E. López·de·Vergara, Víctor A. Villagrá, Juan I. Asensio[†],
José I. Moreno[‡], Julio J. Berrocal.

Dep. of Telematic Systems Engineering
Technical University of Madrid (DIT-UPM). Spain.
{jlopez,villagra,jasensio,berrocal}@dit.upm.es
[†] Visiting researcher from U. Valladolid. Spain.

[‡] Área of Telematic Engineering
Carlos III University of Madrid. Spain.
jmoreno@it.uc3m.es

ABSTRACT

The spread of E-commerce services based on distributed applications has raised the need to analyze what management techniques are suitable to control and monitor the resources implied in their operation. These management methods have the goal of improving the performance and the reliability of these applications, and making easier configuration and security tasks, which are key issues to maximize the quality of service offered. This paper gives an analysis and comparison between different techniques used for the management of distributed applications. These techniques have been applied to the management of E-Commerce brokerage service applications developed in some European projects.

Keywords: Application and Service Management, Distributed Platforms, CORBA, SNMP, JIDM, Management Gateway.

1. INTRODUCTION

Nowadays it is a fact that the network management technology has a good maturity level and it is used in the majority of the network environments. This fact, joined with the high flexibility degree of these technologies, has raised the need of expanding the use of these proven technologies to other fields, such as the services, applications and system. There exist many products in the market providing solutions in these scopes.

However, it is not possible to apply directly the traditional network management technology in some aspects. For example, the distributed applications based on distributed processing platform, such as CORBA (*Common Object Request Broker Architecture*), DCOM (*Distributed Component Object Model*) and RMI (*Remote Method Interface*), in which the manager-agent paradigm is not directly applicable because the resource to be managed is deployed over different infrastructures.

In the following sections the existing problematic will be explained, giving then some approaches to its solution. Next, several proposals, validated in some European projects, will be shown and compared. These proposals deal with the integrated management of the network and those distributed applications deployed on these networks, hiding as much as possible these management aspects to the application developers.

2. IDENTIFIED PROBLEMS

From the management viewpoint, the implantation of distributed environments implies the challenge of obtaining a global view of the applications behavior by means of the management information maintained in those components that compose them. Therefore, it is necessary to standardize the access to that information, allowing the reuse of management service implementations for future applications running in such distributed processing platforms based environments. This also implies the need to instrument the applications to provide the requested management information.

On the other side, the big amount of existing management platforms requires the integration of this application management with the traditional network management technologies, even though this might be a temporal solution until the new emerging management technologies, such as Web based management, are matured. (See Figure 1).

Moreover, an important requirement is to minimize the impact over the applications, in the way the management functionality affects as less as possible their development.

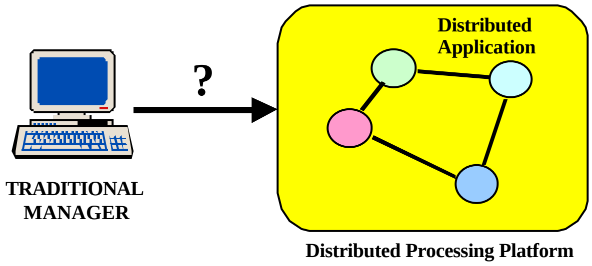

Figure 1. ***Distributed applications management problematical***

3. PROBLEM APROACHES

There exist some general solutions to the integration problem, illustrated in Figure 2, which try to solve the interoperability with the traditional management platforms, maintaining a certain transparency level [3][5]:

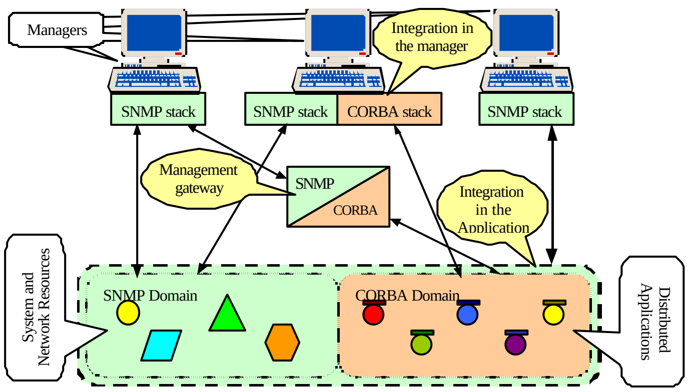

Figure 2. ***Approaches to the integration problem***

1. Using managers capable of supporting all the possible managed resource types, implementing all the necessary information exchange protocols to access the management information of the different components in a distributed application. For instance, a management platform with SNMP (*Simple Network Management Protocol*), CMIP (*Common Management Information Protocol*) and CORBA stacks simultaneously.

2. Making the application components able to adopt the traditional network management technology being used, although its information exchange protocols are different from those used internally to communicate the different application components. An example will be a CORBA distributed application that can answer SNMP requests directly.

3. Placing an intermediate layer between manager and application, a gateway, which is in charge of dispatching the management information requests to the different components of the multiple applications and other existing network and system agents, obtaining therefore the desired transparency. This can be shown as a gateway that translates SNMP requests referred to the applications performing CORBA invocations to those applications, and forwarding the other ones to the corresponding agents via SNMP.

It is possible to think of other intermediate solutions, which use different aspects from those shown above. Applying one or another will depend on each case. However, the last one seems to be conceptually more adequate for generic issues, given that it is possible to increase the number of applications or resources, or to use different management platforms without having to perform big changes in the implied domains.

On the other hand, there also exists the transparency problem for the applications management instrumentation. This means, how to maintain and manipulate the management information in the different components that belongs to it, minimizing the impact to its functional development.

A possible solution consists in employing Object Oriented design in these applications: Using inheritance, each object can extend other one that implements standard interfaces to access the management information. Developers other than the application programmers can implement these interfaces. This solution, named wrapper instrumentation in [4], gives flexibility in the instrumentation.

The chance to monitor the incoming calls to the functional interfaces, using CORBA interceptors [6], automates the support of a part of the management information and adds transparency to the wrapping solution, avoiding that the developer of the functional aspects of the application to perform this task.

4. ADOPTED SOLUTIONS

Some implemented and tested approaches adopted in distributed applications management are shown in the following subsections. Specifically, four approaches are detailed: three are based on gateways and the other one in traditional management interfaces placed on the application. These solutions have been applied and validated in two ACTS (*Advanced Communication Technologies and Services*) European Projects:

- ABS (*Architecture for Information Brokerage Service*). It is focused on the application of the brokerage concept to the electronic commerce world.

- ABROSE (*Agent Based Brokerage Services in Electronic Commerce*). It has taken the results obtained in ABS to develop a brokerage service using intelligent agents technology.

These projects have used different technologies (CORBA distributed systems and agent-based/multithreaded systems, respectively) for providing an information brokerage service, so the management architecture has had to face up these heterogeneities.

The adopted solutions have had also the requirement, as mentioned in section 2, of making use of an existing SNMP management platform, which also deals with common network elements.

Thus, the manageable parameters of the applications were identified and modeled using SMI (*Structure of Management Information*), making it easier the integration of these services management with such conventional management platforms. Those defined MIBs (*Management Information Base*) contained information related to all management functional areas, being possible examples for each one:

- Fault: To know if a component was running or not.

- Configuration: To start or stop a component or the whole system.

- Accounting: How many times a user logs into the system, how many requests are done each time.

- Performance: How long a request takes to give a response. How many responses are given to a request.

- Security: A user can log in the system or not.

In fact, the defined information dictated what parameters of the brokering systems could be managed, giving an outline about the required management functionality to be implemented.

Gateway with direct relationship access

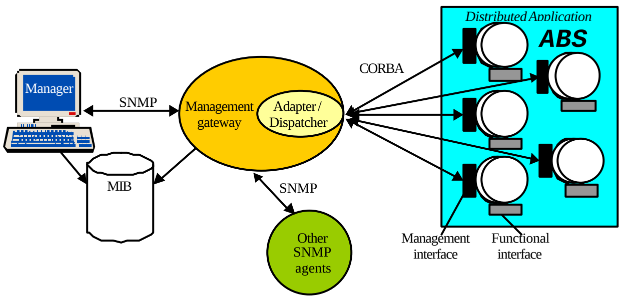

Figure 3. ***Solution adopted in the first version of ABS***

The first approach, used in the first ABS prototype [1] and illustrated in Figure 3, has used a gateway that accessed the different components of a CORBA distributed processing application. In this case, there was a *matching* between the component names and the defined management information. With this, the gateway, an extensible agent, forwarded the requests from a network management platform to the different components in a direct way, and also for network and system agents. The application components should have a well-known management interface, so the access and modification to previously defined variables is possible.

The instrumentation has been performed, as shown in section 3, with wrapped objects that implement the management interfaces and maintain the managed variables. Those variables have been defined in a MIB containing all the management information related to this distributed application.

This approach has been also adopted in the first ABROSE prototype [9], where the distributed application performs some tasks using threads (see Figure 4). In this case, the access to the application will be performed from a single point that redirects the requests, also using a direct relationship, to the different components, which have been implemented as threads in the same address space. On the other hand, the gateway, as it is an extensible agent, is still in charge of forwarding requests related to standard management information to other existing agents in the same machine.

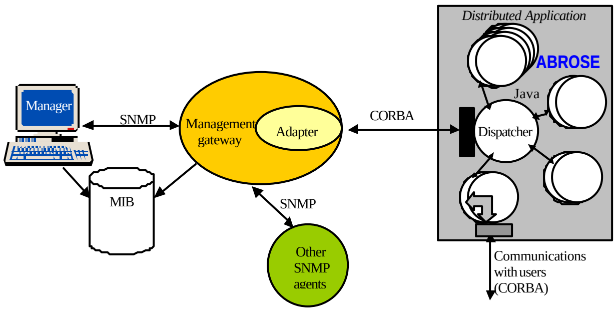

Figure 4. ***Solution adopted in the first version of ABROSE***

Generic interoperability between CORBA and network management frameworks

To allow the interoperability between the traditional management frameworks and CORBA-based distributed processing platforms, the Open Group set up the JIDM (*Joint Inter-Domain Management*) working group, whose proposals has been adopted later by the OMG (*Object Management Group*). This working group has been studying how to carry out this interoperability, concluding that it can be possible by solving two questions [7]:

- Standardize the management information Specifications Translation (JIDM-ST), detailing the translation between the data types and structures used in CMIP and SNMP, traditional network management protocols, and those used in CORBA. This implies that beginning with a MIB, written in GDMO (*Guidelines for the Definition of Management Objects*) for OSI (*Open System Interconnection*) or in SMI for Internet, it is possible to generate an IDL (*Interface Definition Language*) module defining what CORBA interfaces have to be implemented in an object that is going to be managed with that management information. Besides, it is possible to do an inverse translation from an IDL module to a GDMO MIB.

- Standardize the Interaction Translation (JIDM-IT) between the different domains: CORBA, CMIP and SNMP. This implies to define a set of algorithms and services to allow the translation and routing of the requests and answers generated in different domains. For example, in an interaction between SNMP and CORBA, some services are defined to translate an ASN.1 object identifier (OID) to its bound name and using this name, obtain the CORBA object reference (IOR, *Interoperable Object Reference*) that maintains the information related to that name.

These questions simplify the management of distributed applications that run in a CORBA-compliant platform using a traditional network management platform, through gateway-like solutions like that one applied in the second prototype of ABS [2], illustrated in Figure 5:

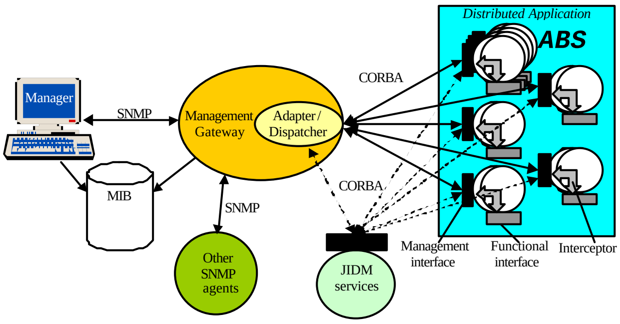

Figure 5. ***Solution adopted in the second version of ABS***

In this approach, when the gateway, still an extensible agent, receives a request from the management platform to the distributed application it uses the JIDM algorithms. Those algorithms defined in the Interaction Translation allow the access to the manageable resources, translating SNMP primitives to CORBA invocations to objects, obtaining their references through the services created for this purpose. Furthermore, the components of these resources implement the IDL interfaces generated from the management information previously defined in a SMI MIB, using the Specifications Translation. This mechanism produces one IDL management interface by each MIB group, which supposes certain restriction because it is necessary to specify the management information of each CORBA component in a different MIB group.

In this version, the interceptors mentioned in section 3 to account the operations exchanged between the different components have been used. Another improvement in relation to the first prototype is the possibility of managing different instances of the same component, thanks to the JIDM services defined in the Interaction Translation. However, the increment of the number of interactions between the different system components affects negatively to the performance. To enhance it, a more complex instrumentation has been introduced in this approach, which is able of measuring locally some QoS (*Quality of Service*) metrics (e.g. response times) without the action of a manager. Then, that manager only has to get the values of those metrics already measured.

Approaches based on components technology

Another approach, which has been applied in the management of the second ABROSE prototype [10] (see Figure 6), is based on components technology. This approach facilitates the software development using objects previously developed, with modifiable attributes accessing to interfaces that fulfill specific design patterns. In the particular case of Java, *management beans* have been defined in JMX (*Java Management Extensions*) [8] for that purpose.

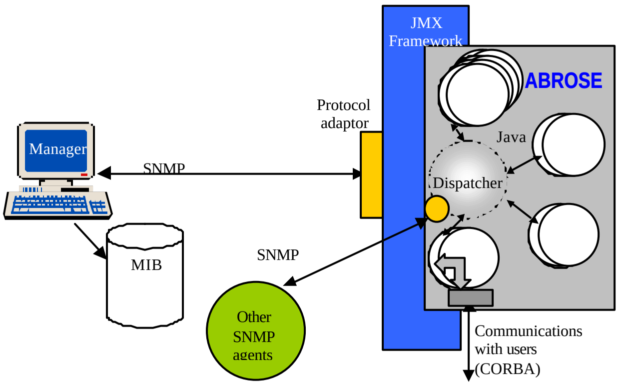

Figure 6. ***Adopted solution for the second ABROSE prototype***

With this approach, it is possible to manage a multi threaded application in a transparent way to manager and programmer, creating Management Beans or M-bean based on the defined management information and finally, adding them to the different threads that compose the application. In this case, the SNMP request are received through a protocol adapter and transparently dispatched to the components. The existence of other components that forward requests to other applications allows using external agents to obtain standard management information.

In relation to CORBA, the communications with service users were monitored using interceptors again. A thread was in charge of monitoring the users' connection, accessing the users' CORBA interface, so the manager could know in every moment who was connected, from where and how much time, being these data useful as stated before.

5. APPROACHES COMPARISON

This section is intended to compare the approaches previously shown, giving pros and cons of all them in terms of performance, complexity, flexibility and instrumentation. Performance is relative to the execution of management tasks in the same hardware. Complexity is related to the way in which this solution can be applied. Flexibility is related to the functionality of the solution. Instrumentation is related to the way in which the functionality has to be developed.

With respect to the gateway with direct relationship access, this solution is the easiest one to adapt to any scenario. Its performance is relatively good, comparing it to the JIDM approach. Its biggest problem is its lack of flexibility, with reference both to the names of the components and to the functionality. In this case, the instrumentation is very easy to implement, because it is a very simple interface.

The adoption of JIDM as a solution supposes a rise in the flexibility: those things that cannot be done with the previous approach, such as managing different instances of an object, can be now performed thanks to the JIDM services. However, the performance decreases, due to the number of interactions to complete a management operation. With reference to the instrumentation, as JIDM is a CORBA-based solution, it just generates IDL interfaces. This question gives independence of the language used to implement the components, but also implies an additional work to the management instrumentation developers, which have to implement the interfaces generated with the IDL compiler.

JMX is also very flexible, and with the advantage of lesser complexity, which has been hidden inside the *JMX framework* that dispatches the requests. The performance is relatively good. The instrumentation code is directly generated and only few changes have to be done. However, this solution is only valid for Java applications. The interceptors can be applied just to those components with a CORBA interface.

The following table summarizes the comparison.

Approach	Performance	Complexity	Flexibility	Instrumentation
Ad Hoc	+	-	-	+
JIDM	-	+	+	-
JMX	+	-	+/-	+

6. CONCLUSIONS AND FURTHER RESEARCH

The management of distributed applications (their monitoring and control), permits an improvement in the quality of service demanded by their users. The existing technology allows performing this management by means of multiple approaches, being each of them more adequate for certain kind of problems. They can be multiarchitectural management platforms, applications with a traditional management interface, and gateways for the information exchange between different domains, created ad-hoc for a concrete application, or absolutely general.

In this article some applied solutions have been shown to manage distributed platform based applications, through the use of platforms and existing network management tools, using solutions based on gateways with different level of generalization, and also components technology. The management instrumentation of them has been performed using concepts like the wrapping of components and the CORBA interceptors.

Many concepts shown in this paper are being applied in another E-Commerce research project, IST MKBEEM (*Internet Society Technologies Multilingual Knowledge Based European Electronic Marketplace*), in which an EJB (*Enterprise Java Beans*) distributed application has to be managed. In this case, the JMX framework is being used as a gateway to the EJB platform, existing one M-bean, acting as proxy, per EJB component. The EJB are wrapped again to obtain any needed management information. Figure 7 shows how this is performed.

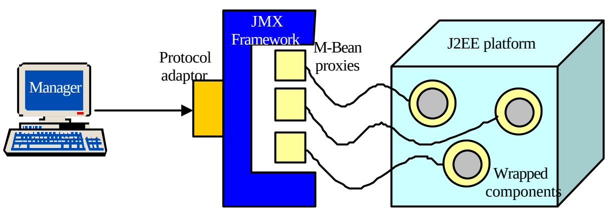

Figure 7. ***Solution being developed for MKBEEM prototype***

Nevertheless, other general problems are still unsolved: the distribution of the managed variables, inherent to this kind of platforms, poses new questions about how to share management data and aggregate the distributed managed information in order to obtain consolidated data about the managed aspects of the application.

7. ACKNOWLEDGEMENTS

This work has been partially funded by the European Commission through ACTS ABS (AC206) and ABROSE (AC316).

8. REFERENCES

[1] Juan I. Asensio , Víctor A. Villagrá. *A Simplified Approach to the Management of a CORBA-based Electronic Brokerage Application. Proceedings of the 5th HP OpenView University Association Plenary Workshop (HPOVUA'98)*, ENST Bretagne, Rennes, France, 19-21 April 1998.

[2] Juan I. Asensio , Victor A. Villagrá, Jorge E. López de Vergara, Julio Berrocal. *Experiences with SNMP-based integrated management of a CORBA-based electronic commerce application. Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management (IM'99)*, Boston Park Plaza Hotel, Boston, Massachusetts, U.S.A, 24-28 May 1999.

[3] P. Kalyanasundaram, A.S. Sethi, *Interoperability Issues in Heterogeneous Network Management*, en M. Malek (editor), *Journal of Network and Systems Management*, vol. 2, pp. 169-193, June1994.

[4] M.J. Katchabaw, S.L. Howard, H.L. Lutfiyya, A.D. Marshall, M.A. Bauer, *Making Distributed Applications Manageable Through Instrumentation. Proceedings of the 2nd International Workshop on Software Engineering for Parallel and Distributed Systems (PDSE'97)*, Boston, Massachusetts, U.S.A., 17-18 May, 1997.

[5] Alexander Keller, *Managing the Management: CORBA-based Instrumentation of Management Systems, Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management (IM'99)*, Boston Park Plaza Hotel, Boston, Massachusetts, U.S.A, 24-28 May 1999.

[6] Jorge E. López·de·Vergara, Víctor A. Villagrá, Juan I. Asensio, José I. Moreno, Julio J. Berrocal. *Experiencias en la Gestión de Aplicaciones Distribuidas. Actas de las IX Jornadas Telecom I+D*, Barcelona, Spain, 16-17 November 1999.

[7] The Open Group, *Inter-Domain Management: Specification and Interaction Translation*. Open Group Document C802, January 2000

[8] Sun Microsystems, Inc. *Java Management Extensions White Paper*. http://java.sun.com/products/JavaManagement/wp/, December 1999.

[9] Francisco Valera, José I. Moreno, Victor A. Villagrá, Julio J. Berrocal. *Mecanismos de comunicación y gestión de servicio de un broker de información multiagente. Comunicaciones de las II Jornadas de Ingeniería Telemática JITEL'99*, Leganés , Madrid, 15-17 September 1999.

[10] Francisco Valera, Jorge E. López·de·Vergara, José I. Moreno, Victor A. Villagrá, Julio J. Berrocal. *Communication and Management Experiences in an E-Commerce MAS-Based Environment.* Communications of the Association for Computing Machinery (CACM). Special Section for Agents in Telecommunications. April 2001.