An Approach to the Transparent Management Instrumentation of Distributed Applications

Víctor A. Villagrá, Juan I. Asensio[†], Jorge E. López-de-Vergara, Julio Berrocal, and Roney Pignaton.
Department of Telematic Systems Engineering.
Technical University of Madrid (DIT-UPM). Spain.
{villagra,jasensio,jlopez,berrocal,roney}@dit.upm.es
[†] Visiting researcher from U. of Valladolid. Spain

Abstract

This paper explains the problem of introducing management instrumentation in distributed application in a way that makes this instrumentation transparent to their developers. The paper proposes different approaches for the achievement of this transparency and describes how they were validated in two scenarios involving the management of distributed applications for e-commerce environments.

Keywords

Application Management, transparent instrumentation, CORBA interceptors, JMX, EJB.

1. Introduction and Motivation

The traditional approaches for managing telecommunication network and services allow the integration of different infrastructures by using an integrated management model (e.g. SNMP/SMI). However, in many cases the use of these common models is inefficient, particularly, where there is a need of managing applications, which require specific management solutions that are beyond the traditional common management models.

There has been a lot of work in the past years concerning the integration of the systems and application management, most notably the effort of DMTF: the WBEM (Web Based Enterprise Management) and its associated information model: CIM (Common Information Model). In order to make these initiatives more successful, the applications should incorporate management infrastructure, e.g. instrumentation, in a way that is as much as possible transparent to the developers of these applications. This paper proposes techniques for achieving that transparency for the management instrumentation.

2. Management Instrumentation of Distributed Applications

Distributed application management is characterised by the fact that managed objects are no longer abstraction of real resources but are actual software modules. This fact is particularly evident in object-based or component-based distributed applications where, in contrast to the traditional network management paradigms, the relationship between managed objects and the managed application resources might be completely specified and standardised. This relationship is supported by the management instrumentation of the applications. In a broad sense, instrumentation can be done in two main different ways:

- Intrusive instrumentation: the functional code of the application component has to be modified in order to be managed. The use of class wrappers and stub instrumentation in object-based distributed applications are two examples of how to reduce the level of intrusiveness.
- Non-intrusive instrumentation: the functional code of the managed application is not modified at all. The use of interceptors for the management of CORBA applications is an example of non-intrusive instrumentation.

3. Transparent instrumentation: case studies

In general, larger management flexibility means more intrusive instrumentation. The following paragraphs present two experiences applying the above techniques transparently. These experiences were obtained by instrumenting and managing distributed applications developed in European research projects.

1. The first scenario consisted of the monitoring of an e-commerce brokerage service, provided by a distributed CORBA application, in the scope of the ACTS ABS (Architecture of an Information Brokerage Service) European project [1]. In this project the instrumentation of the application was achieved by a combination of the following techniques:
   - CORBA interceptors were used in order to get management information derived from interactions among applications objects in a non-intrusive way.
   - Application objects were wrapped in order to offer management interfaces by means of which the management information obtained by the interceptors could be retrieved. The wrapping was based on object-oriented inheritance.

   The management interfaces of the application were generated applying the JIDM Specification Translation algorithms to a defined ABS SNMP MIB. The management information defined in this MIB and accessible through the management interfaces of the wrappers provided different levels of application management functionality. The use of non-intrusive CORBA interceptors allowed the instrumentation of generic application performance. Furthermore, some management information related with internal attributed of application objects could be directly obtained from the inherited wrapper. And all this management functionality could be achieved without forcing the developers of the applications to add a single line of code.

2. The second scenario consisted of the management of an intelligent, knowledge-based multilingual electronic commerce brokerage platform, provided by an EJB-based application, in the scope of the IST MKBEEM (Multilingual Knowledge-Based European Electronic Marketplace) European project [2]. The ongoing development of the first prototype of the MKBEEM platform is based on the J2EE platform that incorporates the EJB component model. JMX (Java Management eXtensions) was chosen as the basis for the management architecture of the MKBEEM EJB-based prototype. Management information and operations of Java applications managed with JMX are made available through Model MBeans (Dynamic MBeans that expose their management interface at run-time).

   For instrumenting EJB-based applications, the fact that the EJB-based applications have to be deployed over the EJB container of a J2EE server is a great advantage. That deployment step implies that the EJB container generates all the necessary stubs and

skeletons for RMI-IIOP communications as well as a wrapper for the EJBs containing the support needed for transactional, security, persistence management, etc. related issues. In this sense, and by introducing the appropriate modifications in J2EE deployment tools and containers, management wrappers similar to the ones described in the previous sections might be generated automatically maintaining, at the same time, the level of transparency. Therefore, the "wrapping" approach was chosen for the instrumentation of the MKBEEM prototype. Using Java class inheritance, the management wrappers could be integrated with the functional code of the application without modifications. With the wrappers installed, each functional method invocation is captured by the wrapper (thus acting as an interceptor), which may take any management decision before passing it to the functional component. In the case of the MKBEEM prototype, the collected management information could be delivered to the JMX-based management architecture. Although this management architecture is proprietary, it is worth mentioning that there is an ongoing effort within the Java Community in order to standardise all management aspects of the J2EE architecture [3].

4. Conclusions

This paper has presented several techniques for the transparent instrumentation of management aspects in distributed object-based and component-based applications. These techniques have been successfully applied to several development projects and have proved to be very useful for hiding management problems to the developers of the functional aspects of the instrumented applications.

References

[1] J.I. Asensio, V.A. Villagrá, J.E. López-de-Vergara, J.J. Berrocal, "Experiences with the SNMP-based integrated management of a CORBA-based electronic commerce application", in *proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management (IM'99)*, Boston, MA, 1999.

[2] J.I. Asensio, V.A. Villagrá, J.E. López·de·Vergara, R. Pignaton, J.J. Berrocal, "Experiences in the management of an EJB-based e-commerce application", in *proceedings of the 8th HP OpenView University Association Plenary workshop (HPOVUA'01)*, Berlin, Germany, 2001.

[3] Java Community Process, "J2EE Management", *Java Specification Request JSR-77*, 2001.