

# Integrated Network Management Using Java Development Environment

*Jae-Oh Lee*

*Switching Technology Research Lab, Korea Telecom  
17 Woomyun-Dong, Suhcho-Ku, Seoul, Korea, 137-792,  
Tel: +82-2-526-6068, Fax: +82-2-526-6389,  
E-mail: jolee@infortel.kotel.co.kr*

## **Abstract**

There are some necessary requirements for implementing network management system such as network management protocols, object-oriented information modelling, management functions/services, and Graphical User Interface(GUI) in a distributed environment. Java™ is designed to meet the challenges of application development in the context of heterogeneous, network-wide distributed environments and provides the capabilities of simplicity, object-oriented concept, reliability, portability, multi-threading, and viewer capability. The Java distributed environment is very useful to deploy the Integrated Network Management System(INMS) which can provide the management communication interfaces of SNMP and CMIP. In this paper, we design the application structure of INMS using the Java development environment with object-oriented methodology. Especially, it focuses on the implementation of management services and functions in simple and uniform way without the needs to use the complex and static technologies. And, it will be applicable to manage ATM networks in a management domain.

## **Keywords**

Java, Java Management API, SNMP/CMIP, Management Information, Management Service/Function, ATM Network Management

## 1 INTRODUCTION

In accordance with the increasing complexity and heterogeneity of networks, the environment of distributed computing has required industry and research to make an uniform way of managing them. In recent, there are so many efforts to make the application of network management useful from the viewpoints of modelling and viewing. Java is one of very promising development kits in realising the application and service of network management. In recent, many researchers have been developing the network management world with the Web. One of their efforts is the Java Management API(JMAPI) which provides a rich set of extensible Java objects and methods for building applets that can manage an enterprise network over Internets. JMAPI is a core set of Application Programming Interface(API) that can be used across a diverse array of computing environments involving numerous operation systems, architectures, and network protocols, enabling the development of low maintenance, heterogeneous software from a single source. JMAPI provides the user interface guidelines, Java classes, and specifications for developing seamlessly integrated system, network, and service management applications, but it has not supported a CMIP interface for TMN.

Network management is one of areas which might integrate many technologies such as GUI standards, networking facilities, distributed processing, database, object-oriented modelling and so on. But, it is very difficult to make the universal architecture for systems and network management to be enlarged into the domain network management. Network management application might support the ease of programming, the extensibility, the object-oriented concept, the simple description of management service and so on. Since the Java object model is easier to learn CMIP and SNMP, anyone who is able to create Java applications can immediately use services offered by CMIP/SNMP agent systems, given a Java interface to them, without having to specific knowledge about CMIP and SNMP. Therefore, it is proper to make the environment of INMS by using Java programming technique and JMAPI which are embedded in the Java distributed environment.

In this paper, we are willing to design the architecture of INMS by the help of Java development environment. As a result of using this architecture, we can make the simple realisation of complex management services by expressing them as defined objects which can be divided into their functional units. They can be activated as the combination of management communication object, managed information object according to their behaviours. This architecture provides the following advantages: 1) the integrated management communication interfaces to both manager and agent system conforming to SNMP/CMIP, 2) the translation of GDMO/ASN.1 to Java being interpreted in the Java environment. 3) the method of dynamic configuration of management behaviour by maintaining management services/functions represented as Java classes.

## 2 AN ARCHITECTURE OF INMS

The Java programming language environment provides a portable, interpreted, high-performance, simple, object-oriented programming language and supporting run-time environment. The Java system that emerged to meet these needs is simple, so it can be easily programmed by most developers; familiar, so that current developers can easily learn Java; object oriented, to take advantage of modern software development methodologies and to fit into distributed client-server applications; multithreaded, for high performance in applications that need to perform multiple concurrent activities, such as multimedia; and interpreted, for maximum portability and dynamic capabilities. Java programming language supports exceptions and programmers may be accustomed to manipulating them, so it is appropriate to replace protocol errors with exceptions and to avoid to execute a lot of code or to convert information many times just to get the value of an simple-type attribute. By taking this approach, the one who embody the INMS to be protocol-independent can implement simple management applications and manipulate them in a dynamic way.

The following INMS architecture can enable the one who want to develop the NMS to easily write light network management applications that fully support both CMIP and SNMP using a single and simple object model. Also, these management applications can be deployed into the world-wide web using the Java bindings by providing the classes for them except the main body.

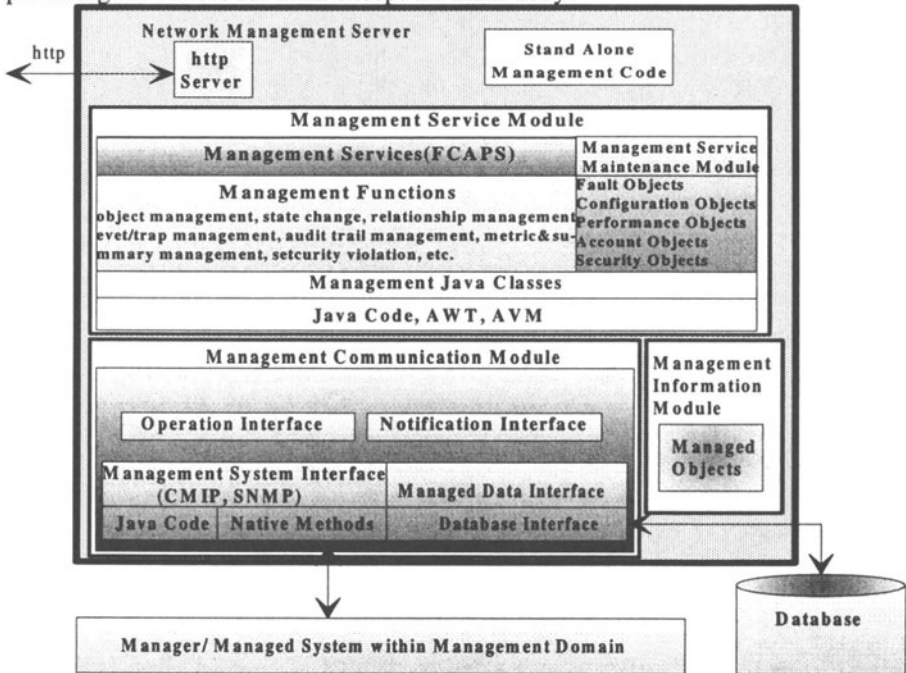


Figure 1 Architecture of Integrated Network Management System

- Management Communication and Information

The maintenance facility provides the operations of create, delete, set attribute, get attribute, search, object identifier conversion, and so on. The management communication is composed of Java classes combined in the native method using our previously developed communication library interfacing with SNMP and CMIP. Each Managed Object(MO) is maintained in the form of Management Information Tree(MIT). Also, GDMO/ASN.1 representing the CMIP and SNMP management information is given the class of Java language. Therefore, we need some automatic tools for the translation of GDMO/ASN.1 to Java class.

- Management Function and Service

A management service is associated with the multiple of management functions. For instance, configuration management is mainly related to the object management, state change management, and relationship management functions. And, the management services and their associated management functions are maintained as Java object classes among which there exists a containment relationship. Therefore, it is suitable to keep them like the methodology of MOs in the form of Management Java Classes derived from Java code, AWT, and AVM classes. And, each management function is constructed from the use of management communication services and MOs.

To satisfy the above requirements, it is proper to write the management service/function and application described in Java language with the support of JMAPI under the Java distributed environment and by the provision of translation method from GDMO/ASN.1 to Java object. Therefore, a management application uses the object classes provided from the management service and function which has their class constructors and methods.

### 3 REMARKS

As networks grow and management services increase in complexity, the dynamic maintenance of them is essential in the INMS which must provide multiple management protocols, management services, and management relationships. Providing the capabilities of simplicity, object-oriented concept, reliability, portability, and multi-threading, the Java distributed environment is so desirable to encompass the requirements of INMS.

By using the proposed model using Java development environment, we can accomplish the INMS that supports the provision of dynamic management services with a variety of management relationships in the management domain which includes managed ATM networks and other networks. In current, we verify the translation method of GDMO/ASN1 to Java for information model in the environment of Java development environment and study the script language like Java script to express management function/service. Also, we must consider other existing services to support the distributed environment.