10

Moving Knowledge Agents for Network and Service Management

T. Zhang and S. Covaci
GMD FOKUS
Hardenbergplatz 2, 10523 Berlin, Germany
+ 49 30 25499 339, +49 30 25499 171
zhang@fokus.gmd.de, covaci@fokus.gmd.de

Abstract

This paper proposes a new framework for developing, marketing and deploying network and service management solutions using Java-based Mobile Intelligent Agents. The management solutions are regarded as knowledge, defined as Java object classes and structured into a hierarchy by the hierarchical specialization relationship among the problem situations which the solutions are addressing. With this hierarchical organization of solutions, we can obtain the Object-Oriented style of reusability/interoperability, and the accumulative construction/standardization of management functionality in the distributed open environment. To manage the real resources, Java-based Mobile Intelligent Agents (IAs) carrying the management solution hierarchies are envisaged, which travel among the managed network sites to solve the problems in the environment by finding out and applying locally the most suitable management solutions. The knowledge of an IA can be further extended by the local, specific management solutions of the network sites it visits, in order to support the optimal management of the local resources.

Keywords

Java, Mobile Agent, Intelligent Agent, Network and Service Management, Knowledge Representation/Processing, Object-Oriented Development.

Management of Multimedia Networks and Services R. Boutaba & A. Hafid (Eds.)

1 INTRODUCTION

The emerging Global Information Infrastructure (NII, 1997), (EII, 1995), (GII, 1994) and the associated telecommunications environments are characterized by the heterogeneity, dynamic nature, the distribution and the dimension of the resources and problem space. Network and service management in this context is a complicated task, which has to deal with the dynamic and complex inter-relationships among the information entities within the managed environments. Sophisticated expertise is needed extensively to realize effective management functions. Lack of enough qualified human experts and sufficient management capabilities are becoming a serious problem for the operation of broadband telecommunications networks and services.

The solution for these problems can be in the following approaches:

- Intelligence
  One solution to cope with the complexity is to provide *artificial intelligence* realized as autonomous functionality that can at least partially replace the human management experts in certain application contexts. The basis of this solution will be a knowledge framework for acquisition, representation, transfer and autonomous processing of management expertise.
- Divide-and-conquering the problem space
  The traditional approach to solve large scale problems is to divide the problem space into manageable ones. In the context of network management, this means to divide the management task into specific sub-tasks for which management solutions can be easily provided, maintained and deployed. Such sub-tasks can be defined by a variety of criteria, which can be technological, administrative, topological etc.
  The complexity and dimension of the tasks and sub-tasks in this context also mean that the solutions have to operate autonomously to fulfil their tasks.
- Mobile and dynamic management solutions
  Within the global telecommunication market, the management expertise and the related management solutions for specific problems can be acquired or developed anywhere in the global network. The same network problems, however, can occur in any remote sites without the necessary expertise and which are far away from the source of the solutions. There must be therefore a solution for moving the management expertise from its provider over the network to the consumer sites.
  Moreover, the changing customer requirements and technologies call for management paradigms which support the dynamical extension or modification of the management solutions. With such distributed and accumulative acquisition of the management expertise, mobility and extensibility will play a key role in the feasibility of the management paradigms.

Traditional research for artificial intelligence or for coping with large scale and distributed problems typically focuses on one of these approaches and have therefore have their limitations in a global telecommunication market.
Intelligent Mobile Agent (called IA hereafter in the paper), which is becoming a buzzword in computer science and especially in the telecommunications communities, offers a promising solution to combine all these approaches.
IA can be defined as an autonomous software entity which, with certain degree of intelligence, and can migrate to and operate in different network nodes to achieve its objectives, e.g. network management objectives.
Compared to the traditional scenarios of network management by travelling experts, network and service management by IAs can be depicted in Figure 1

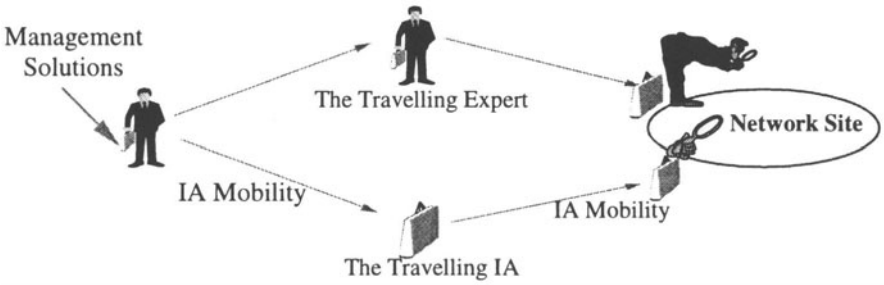

Figure 1 IA vs. Travelling Experts

The key features of IAs are the autonomy, intelligence and mobility. In the context of network and service management, the combination of these features can:

- reduce the requirement on customer/consumer intelligence during the installation, operation and maintenance of the management solutions via the intelligence and autonomy of the IAs;
- reduce the requirement of traffic load and the availability on the underlying networks during the deployment of the IA, by migrating the IA as close as possible to the managed resources using the autonomy and mobility of the IAs;
- enable „on demand" provision of dynamic/customized management services, via dynamic IA migration from the provider system to the customer system;
- increase the flexibility, reusability and effectiveness of the software-based network management solutions using distributed knowledge-based approaches.

This paper presents an IA framework currently under development and validation in GMD FOKUS, which combines the functionality of both intelligence and mobility based on Java technology.

2 THE FRAMEWORK OF INTELLIGENT MOBILE AGENT FOR NETWORK AND SERVICE MANAGEMENT

Following the definition of IA, an IA Framework will be supported by two platforms:

- The *agent intelligence platform* supports the acquisition, representation, and processing of the application specific IA knowledge. Such IA knowledge defines the real functionality of the IAs.
- The *agent management platform* supports the common facilities which will be needed for the management and migration of the IAs. Such common facilities can include:
  - ⇒ lifecycle management (creation, deletion, suspension, activation)
  - ⇒ agent naming and location service,
  - ⇒ agent transportation, and agent message transportation,
  - ⇒ security.

The agent management platform is the major issue in most current developments of mobile agent platforms including the OMG MAF (Mobile Agent Facility) specification (OMG, 1997). Usually this platform will be realized via a number of distributed agent management environments, which will be *called agent systems* hereafter. Each agent system offers the agent management facilities to all the IAs visiting or residing in the corresponding network node.

Figure 2 The IA Framework

The agent intelligence platform can be regarded as a client layer of the agent management platform as depicted in Figure 2. In this configuration, the agent intelligent platform is implemented as a set of *agencies* associated to the agent systems. Each agency can host the incoming IAs and support their operations by utilizing the facilities offered by the underlying agent system.

E.g. to move an IA from one network node to another, the agency first delivers the IA to the local agent system, which will contact the destination agent system to transfer the agent. The destination agent system, after receiving the IA, will delegate the IA to the appropriate agency for IA operation.

Usually the agent management platform tends to be application context independent, while the agent intelligence platforms and the agencies can be very different for different application areas.
This paper focuses on the agent intelligence platform for network and service management.
The agent intelligence platform has two key functions:

- IA knowledge representation and
- IA operation,

where the IA knowledge representation framework supports the specification of IA functionality, and the IA operation environment provides the infrastructure for the operation of the IAs.
In an open environment, numerous players can contribute to the management solutions with their own experiences and views. A suitable framework supporting such a global and co-operative acquisition of knowledge will therefore determine the success or failure of the IA-based approach. The following characteristics of such a knowledge framework will be important in this context:

- Reusability and Interoperability

The global telecommunication environment encompasses a tremendous number of heterogeneous sub-environments. The success of any solutions in this world depends on their capability of integrating themselves into the different operation environments, i.e. the ability to be applied in a wide range of contexts.
Reusability and interoperability in the distributed environments are usually achieved through standardization at different syntactical or semantic levels, and through Object-Oriented constructions.

- Accumulative Construction

Management will be based on complicated knowledge. Such knowledge can only be derived from profound experiences which have to be collected gradually during the evolution the resources. The result is that management has to, at the beginning, be based on some very primitive management knowledge and operations. With the time passes by, new knowledge will be obtained concerning the deeper inter-relationships between the components of the environment and the detailed behavior of the components, new management operations can be added due to new requirements and new facilities. All these new knowledge and functionalities have to be gradually integrated into the management stations.
In this sense, accumulative construction of the intelligent management systems is also an important factor in realizing effective management of future networks. The most important technology in this context is again the Object-Oriented style of development.
Another important aspect of IA framework is to enable the migration of the IAs and the interpretation of IA functionality in all the agencies an IA visits.
The necessary and sufficient pre-condition in this case is that all the network nodes support the same IA representation, either directly or via some proxy functions. A representation language which is supported by most popular platforms will certainly have advantage over other less popular languages. With the dominance of Internet in the context of telecommunication and telecommunications management,

any language platforms that are widely accepted and supported in this community will be the best candidate.
Java, due to its strong connection to WWW, and also due to its simplicity in Internet-oriented programming, is currently very popular within the Internet community and is also supported by most platforms. We select, based on this consideration, Java as the base language for our IA-based management solutions.

3 THE KNOWLEDGE FRAMWORK FOR AGENT-BASED NETWORK AND SERVICE MANAGEMENT

The knowledge for the management IA is the set of management solutions that address the problems that can occur in the network. Each management solution can be constructed from two key components:

- the *pre-condition* for the solution, in terms of the network situations that require or support the application/invocation of the solution, and
- the *management plan*, defined as a pre-defined sequence of management actions, including the generation of new Event Reports.

3.1 The Situation Theory for IA-based Network and Service Management

Most currently dominating network and service management frameworks, including the ITU-T TMN (ITU-T, 1995a)or Internet SMI (IETF, 1990), view the managed network environment with its set of resources as a set of Managed Information Entities.
In this paper, we discuss only the TMN/CMIP-based management environment. The framework, however, can be easily adapted to a SNMP-based environment.
In a TMN/CMIP-based environment, the Managed Information Entities are represented as Managed Object (MO) Instances. Each of such MOs can possess the following information capabilities:

- the *attributes*, which contain the status and characteristic information concerning the managed resources,
- management *actions*, which support the management operations upon the underlying resources, and
- the *notifications* with their parameters, which the managed resources can issue to the managers for signaling special events in the network. In the following we will call notifications received by the managers *Event Reports (ERs)* to maintain a simple terminology.

A manager in this context can make the following basic operations on the MOs:

- *get* for obtaining the value of an attribute by providing the Identifier of the attribute
- *set* for setting the value of a MO attribute
- call a management *action* which is supported by the MO
- receive an *ER* from the TMN management agent representing the resources

In such an environment, the current situation in the network is characterized by the set of MO instances (with their attribute values) and the generated ERs (with their parameters).
If we denote the set of all possible sets of MO instances (with all their attribute values) as Ψ and the set of all possible sets of generated ERs as Φ, we can have the following definitions (Zhang, 1996a):
A *situation* is the union of an element from Ψ and an element from Φ. The *universe* of situations is

$$E = \{ \lambda \cup \alpha \mid \text{where } \lambda \in \Psi \text{ and } \alpha \in \Phi \}.$$

A *situation test* t is a function (i.e. an operation) from E to true or false, i.e.

$$t : E \rightarrow \{ \text{true, false} \}$$

The *denotation* (semantics) of a situation test t is the set of situations ε, such that an element χ is in ε iff we have:

$$t(\chi) \rightarrow \text{true}$$

I.e. a *situation test* can be used to the check to validity of situations in the network.
A *situation test* τ is a *specialization* of another *situation test* γ iff the denotation for τ is a subset of the denotation for γ.
The *conjunction* of a situation test τ and any other situation tests is a *specialization* of τ.
Intuitively, a specialized situation test contains more restrictions on its denotation and can therefore be satisfied by a smaller set of situations (Zhang, 1996a), (Zhang, 1996b). Management plans based on specialized situation tests can therefore support safer, and more effective solutions of the current problems in the network.
In our Java-based management environment, the pre-condition for a management solutions is specified by the *situation tests* that characterize the network situations for the management actions. To guarantee the side-effect free testing of the managed network environment, we restrict the management operations within a *situation test* to the CMIP *get*. Although some implementations also use CMIP get operation to modify the environment, we believe this is not compliant to the TMN paradigm.
A *situation test* in our Java-based IA framework is just a Boolean Java method which uses only the *get* methods from the CMIP API and any other Java basic methods. This property will be guaranteed by the editing environment for IAs.

3.2 The IA Management Solution

As discussed above, a *management solution* consists of a *situation test*, which checks the situation in the environment, and a *management plan* for addressing the corresponding problems in the network by issuing management operations.
For this purpose, a management solution will be a subclass of the following abstract class definition:

```
public abstract class ManagementSolution {
        public EventMemory eventMemory;
        public Event currentEvent;
```

```
        public CmipAgent agentAddress;
        public abstract boolean situationTest ();
        public abstract AgentSystem managementPlan ();
}
```

In this definition, the *EventMemory* class is used to keep a list of *ERs*, each of which belonging to the class *Event*. The *EventMemory* class also supports the manipulation of events stored via operations like filtering, searching, deleting events, adding ERs, refreshing the memory, etc.

The *currentEvent* variable is used to hold the ER that currently drives the processing of the IA, while the *eventMemory* maintains the list of ERs that can be relevant in the environment of the IA.

The *agentAddress* information serves as the contact address for all the CMIP operations that are to be carried out within the *situation test* and *management plan* methods.

The result of the *managementPlan* can be either *null*, in this case the IA will stay in the local environment, or it can be an *AgentSystem* - the next place to which the IA should be moved.

A real *management solution*, which extends this abstract class, can also contain other private variables that can be manipulated by the *situation test* method to carry extra test result information from the test phase to the *management plan*.

Based on the hierarchical structuring of the *situation tests*, we can have also the hierarchical relationship among the management solutions. For building up such a hierarchy of management solutions, we impose the following conventions on the development of management solutions:

- Each management solution will be contained in a separate Java package. Each management solution package for a more specific problem is positioned under the package for a more generic solution.
- The *situation test* within each management solution implies automatically the *situation tests* in all the ancestor (more generic) management solutions.

Following the Java convention, the management solutions are stored in a hierarchy of directories in the file system.

In this way we build up a hierarchy of management solutions which starts with generic management solutions at the upper part, and grows with specific management solutions positioned under more generic solutions.

The management solutions will be pre-compiled to the Java *class* files. At the time of IA initialization, these class files will be loaded to build a corresponding hierarchical Java data structure, called the *managementHierarchy*, which contains the management solution classes.

An IA consists of the following key components in addition to the components of the mobile agents:

- the *eventMemory* for holding the set of ERs collected by the IA during its life time, and which are still relevant for the IA processing,
- the *currentEvent* which is driving the inference of the current IA,
- the *managementHierarchy* of management solutions.

4 THE IA OPERATION ENVIRONMENT FOR NETWORK AND SERVICE MANAGEMENT

The IA operation will be realized by the environment depicted in Figure 3.

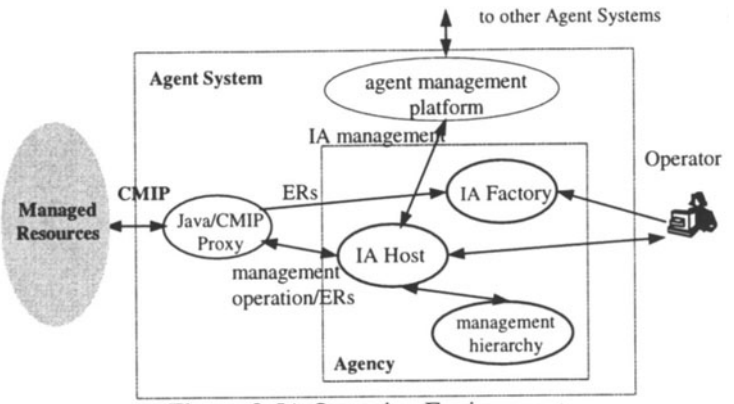

Figure 3 IA Operation Environment

In this configuration:

- the *IA Factory* is responsible for receiving ERs from the managed resources via a Java/CMIP proxy and the TMN Event Forwarding (ITU-T, 1993) service. Upon receiving such an ER or upon initiative of a human operator, the IA Factory generates an IA with an empty *managementHierarchy*, and only with one ER in the eventMemory referring to the currentEvent. This initial IA will be then sent to the associated IA Host for this IA Factory.
- the *IA Host* is responsible for inferencing based on the knowledge in *managementHierarchy*, and for finding out the most suitable management solutions for the current situation in the network. For this purpose, the IA Host also maintains an *eventMemory* of the ERs received in the local environment.

It is possible that some network nodes have only IA Factory, which means that the initial IAs will be sent to some associated IA Hosts in remote network nodes.

Each time when an IA enters an IA Host, the IA Host first integrates the local management solutions hierarchy into the IA, and then starts inferencing on knowledge hierarchy to find the suitable management operations to be issued to the managed resources via the Proxy.

To update the management solution hierarchy, the IA Host traverses through the local hierarchy of the Java management solution package file directories, creates an record of each management solution class, and builds up a hierarchy of management solutions (implemented in Java as vector of vectors) to be integrated into the corresponding IA variable.

After the initialization of the *managementHierarchy* and the initialization of other variables, the IA Host starts the execution of the IA, i.e. the traversing of the hierarchy to find a management solution for the current situation in the network.

The traversing will be done in pre-order and started at the root of the hierarchy. At each management solution node the IA Host will do the following:

1. assign the *eventMemeory*, *currentEvent* of the management solution to the corresponding variables in the IA instance;
2. call the *situationTest* method;
3. if the result of *situationTest* is true, and all the children nodes were already searched, call the *managementPlan* method in that solution, otherwise continue the traversing of the hierachy on the next management solution node.

The result of the *managementPlan* method has three possibilities:

1. the method returns *null*, and sets the currentEvent variable to *null*
   In this case the IA will be terminated.
2. the method returns *null*, but sets the *currentEvent* variable to a new ER (one from the eventMemory, or a newly generated one)
   In this case the IA Host will set the hierarchy to the sub-hierarchy under the current node and restart the process of traversing, based on the intermediate result coded in the new ER.
3. the method returns a new *AgentSystem* address
   In this case, the IA host sets the *managementHierarchy* to the sub-hierarchy under the current node and moves the whole IA instance to the new IA Host via the Java Serialization mechanism .

In the new network node, the new IA Host will restart the operation of the IA by repeating the above procedure on the incoming IA.

The advantage with this whole scheme is that each local network node can maintain its proprietary solutions for its local resources. The travelling IA will consider these local solutions even if these are not available in the origin node for the IA.

5 CONCLUSION

In the above we presented a Java-based agent platform for agent-based network and service management solutions, which is based on management knowledge acquisition in Object-Oriented style and on the corresponding knowledge processing framework.

The whole approach has, among others, the following major advantages:

- Automation and delegation of management processes via mobile, knowledge-based Agents

The management framework enables the delegation of autonomous management solutions, which can greatly reduce the complexity and costs for managing the global network resources.

- Compatibility with existing environments

The approach is designed to be based on the TMN management framework and can easily be adapted to a SNMP environment. Each IA communicates with the managed resources via standardized management protocols and operates on the management view offered by the Management Information Base (MIB). The co-

operation among the IAs is realized via TMN Event Reports and Event Forwarding (ITU-T, 1993) service, instead of novel co-operation languages like KQLM (Finin, 1994). This approach can therefore be easily embedded in a standard conform network and service management environment.

- Reusability and Interoperability

A management solution hierarchy starts with some generic knowledge/solutions which are applicable in different environments, and it derives specific knowledge by adding specific information about the individual environment classes. If applied in a different environment, the generic management knowledge in the upper part of the hierarchy will continue to be valid, while the lower part situation knowledge will not be satisfied and will be simply ignored. The result is that the IA with this hierarchy operates also in a new environment, although with some loss of effectiveness.

For the same reason, one management solution hierarchy can be reused in different environments to build there the management systems.

- Accumulative and distributed construction of the global knowledge

When starting to manage an environment with an IA-based framework, we usually use a partial management solution hierarchy to initialize the management functionality. Later, with accumulating experiences and expertise, we can add more specific management solutions with new situation tests and their associated management plans, to improve the performance of the IA system.

Moreover, the management solutions can be developed/advertised by any players anywhere in the open world. The players can also decide to maintain such solutions only in the proprietary environments, and to ask IAs who visit their sites to use such proprietary solutions instead of the generic ones.

- Standardization of global available management solutions

The hierarchical structuring of the management solutions in the Object-Oriented style enables the standardization of the relatively generic management solutions in various contexts and domains for the upper part of the hierarchy under the root. Such management solutions can be further made available to all users via hierarchical information services like the ITU-T X.500 Directory Services (ITU-T, 1995b).

With these characteristics and possibilities, we expect the approach to provide a realistic management framework for the current and emerging telecommunication networks and services.

6 REFERENCES

ETSI (1995) SRC6 Final Report on European Information Infrastructure.

Finin, T. (1994) KQLM as a Agent Communication Language, in *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, Amsterdam.

FIPA (1997) Agent Communication Language. FIPA'97 Specification Part 2.

GII (1994) The Global Information Infrastructure: Agenda for Cooperation, http://www.iitf.nist.gov/documents/docs/gii/giiagend.html.

ITU-T (1992) Recommendation X.722, Information technology - Open System Interconnection - Structure of Management Information: Guidelines for the Definition of Managed Objects.

ITU-T (1993) Recommendation X.724, Information Technology - Open Systems Interconnection - System Management: Event Report Management Function.

ITU-T (1995a) Recommendation M.3010, Principles for a Telecommunications Management Network.

ITU-T (1995b) Recommendation X.500, Information technology - Open System Interconnection - The directory: Overview of concepts, models, and services.

NII (1997) United States National Information Infrastructure Virtual Library, http://nii.nist.gov/nii.html.

OMG (1997) Crystaliz, General Magic, GMD FOKUS IBM and The Open Group. Mobile Agent Facility Specification (Joint Submission). Draft 7.

Zhang, T and Covaci, S. The Semantics of Network Management Information, in *Proceedings of the 1996 IEEE INFOCOM Conference*, San Francisco.

Zhang, T, Covaci, S and Popescu-Zeletin, R. (1996b) Intelligent Agents in Network Management, in *Proceedings of the 1996 IEEE GLOBECOM Conference*, London.

7 BIOGRAPHY

Tianning Zhang, born 1963, studied computer science at the Shanghai Jiao Tong University and the Technical University Berlin, where he received his Bachelor's and Master's Degree in 1985 and in1993 respectively. Since 1994, he has been working in the Research Institute for Open Communication Systems (FOKUS) of the German National Research Center for Information Technology (GMD), joining several national and international projects in the areas of Distributed Directory Service, ATM/SDH network and Global Broadband Connectivity service management, intelligent and mobile agents for network and service management.

Stefan Covaci received his Dipl.-Ing. diploma in 1979 from the faculty of Electronics & Telecommunications at the Polytechnic Institute of Bucharest, Romania, and his PhD in Electrical Engineering in 1987. He is currently a senior researcher and project group manager at GMD FOKUS. His research interests include European/Global Information Infrastructure (EII/GII), B-ISDN (ATM, SDH) and Gigabit networking technologies, OSI/TMN integrated network management, management of INs, ODP and TINA, intelligent and mobile agents in telecommunications management.