

# SNMP Management of MASIF Platforms

*Rui Pedro Lopes*

*Instituto Politécnico de Bragança, ESTiG, 5300 Bragança, Portugal  
rlopes@ipb.pt*

*José Luis Oliveira*

*Universidade de Aveiro, DET/IEETA, 3810 Aveiro, Portugal  
jlo@det.ua.pt*

## Abstract

In this paper we describe the architecture, the development and the assessment results of an SNMP agent that allows MASIF compliant platforms to be managed through SNMP. The major outcome of this work is a simple integration of mobile agent technology with any well-established commercial network management system.

## 1. SNMP Management of Mobile Agents

The SNMP is, so far, the framework of choice from a large number of equipment (hardware, software and technical knowledge), so any near upcoming management architecture must deal with SNMP standards. Thus, the coexistence of mobile agent technology with SNMP equipment and management tools will be welcome. Although the Distributed Management charter of the IETF is considering the SNMP centralized architecture inconvenient, we see in this issue a great area where the mobile agent paradigm can prove its capabilities [1]. This enrichment is easier to achieve if the mobile agents are themselves managed by SNMP thus facilitating the integration.

Several approaches consider mobile agents for managing networks and services due to the distributed nature, efficiency savings, traffic reduction and robustness [2][3]. Besides, adding flexibility and the possibility to improve management efficiency also brings some difficulties, namely it increases the agent management difficulty and introduces yet some usability challenges and possibly threats [4]. This leads to the fact that mobile agents also need to be managed.

Why should mobile agents be managed by SNMP? First of all, SNMP is specifically designed for management operations. Moreover, there are available several SNMP based powerful applications and quite a number of installed systems rely on them. This way it is possible to integrate both the mobile agent technology and the SNMP management architecture thus getting the advantages of both [5][6]. On the other hand, MASIF interfaces specify a common access method to mobile agent platforms. So, any MASIF platform can be managed by following the same procedures, with the same tools. By adding an SNMP layer over Mobile Agent

Facility (MAF) interfaces we are able to manage any compliant platform using a well-known SNMP NMS.

## 2. The MASIF MIB

The Mobile Agent System Interoperability Facilities specification (MASIF) is a first attempt to standardize agent system, i.e. the middleware that allows agents to stop, to move and to restart execution in another place. MASIF uses OMG IDL to declare two interfaces, which are the base to all the operations on the agent system and on the region (some sort of domain in mobile agent technology): the *MAFAgentSystem* and the *MAFFinder*.

The *MAFFinder* interface, which “lives” inside the Region, consists on an access point to a naming service for agents systems, places and agents. Catalogue and lookup are the main functionalities of this interface.

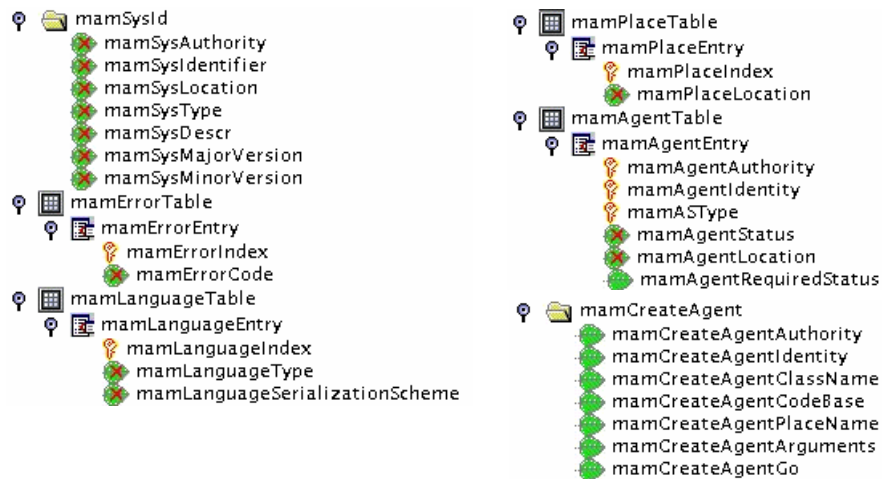
The *MAFAgentSystem* interface defines methods and objects that support agent management tasks such as fetching an agent system name and receiving an agent.

Following the MASIF specification, the MASIF-MIB is structured in two groups: *mamObjects* and *mamLookup*. The *mamObjects* group is related to agent system information, retrieved through the *MAFAgentSystem* interface. The *mamLookup* group allows directory service queries through *MAFFinder* interface method calls.

### 2.1 The *mamObjects* group

This group is mainly based on the *MAFAgentSystem* interface. It starts with a *mamSysId* section, which has information about the agent system authority, identification, type, description and version (Figure 1).

While the agents travel, several errors can occur. The table *mamErrorTable*, maintains a record of these errors.



**Figure 1:** Organization of the *mamObjects* group.

The agent system supports agents written in one or more programming languages, such as Java, TCL and others. For a manager to know which language is supported he has to take a look at *mamLanguageTable*.

The two following tables, *mamPlaceTable* and *mamAgentTable*, provide a list of currently existing places and agents in this particular agent system. Moreover, the agent table allows the user to suspend, resume or terminate an agent by setting the required agent status (*mamAgentRequiredStatus*). The index to *mamAgentTable* is composed of the agent authority, its identity and its agent system type. This triplet ensures that every agent has a globally unique name.

The last section, *mamCreateAgent*, holds the necessary objects for creating an agent. The first two objects, *Authority* and *Identity*, will be ignored if the user does not have naming responsibility. In this situation, the authority and identity will be assigned by the agent system. The *Name* and *CodeBase* provides the agent system with the necessary information to download the code and construct the agent. The agent will be created on the provided place name (*mamCreateAgentPlaceName*) or on a default place if no place has been provided. It is also possible to define arguments to be passed to the agent constructor (*mamCreateAgentArguments*). After setting all the required values, the agent is created by setting the *mamCreateAgentGo* to 1 (*create*).

## 2.2 The *mamLookup* group

To suspend or terminate the agent the user has to know its location. This information can be retrieved from the *mamLookup* group. It focuses on lookup operations, performed mainly with *MAFFinder* interface method calls and is not restricted to a specific agent system. It can be used to perform lookup operations anywhere up to the region limits.

Each section (*mamLookupPlaces*, *mamLookupAS* and *mamLookupAgents*) has a location table, which is updated according to the value on an associated object (*mamPlaceName*, *mamASFilter* and *mamAgentFilter*).

According to MASIF, it is sufficient to locate places based on names. On the other hand, with this MIB, more elaborate queries may be built for agent systems and agents. For example, search rules based on supported languages, agent names, serialization method and so on. This is achieved through a filter with the following syntax:

```
filter = item (|,&) filter      /* or, and */
item = key comparator value
comparator = (=,^,$,~)         /* equal, starts with, ends with, contains*/
key = string with              /* Name, Codebase, Language, ... */
value = string
```

For example, the filter “*Name=pingAgent&Codebase~myHost*” is a query to *MAFFinder* looking for an agent called “pingAgent” with code base containing “myHost”. It issues a *mamAgentFilter* set operation with this string and finds its current location on the *mamLocAgentTable*.

While this functionality is not explicitly stated in the scope of MASIF it does not impair itself MASIF conformance and it enriches significantly the proposed MIB.

### 3. A MASIF Agent Implementation

The MASIF MIB was supported through a MASIF-to-SNMP gateway that is structured in three large blocks: a) the Communication module, responsible for receiving and sending SNMP commands (SNMP stack or other); b) the Message Processing module, responsible for interfacing the communication module to the agent data; c) the Agent virtual Data structure, responsible for implementing the agent behaviour.

With a well-established interface between the communication mechanism and the Message Processing module we can switch communication modules maintaining the agent. This feature is useful, for instance, to check CORBA or RMI performance, or to add mobility to the agent.

The present implementation of the MASIF agent was developed upon the Grasshopper platform using yet the JDK1.3 ORB and the JDK CORBA naming service. The code and some examples can be retrieved from <http://nms.estig.ipb.pt>.

### 4. Conclusions

While the added value of the mobile agent technology is still to be proven, several areas and application are being exploited as scenarios for such contribution. Network management is one of these areas. Considering the impact SNMP have today on management systems, the coexistence of mobile agent technology with SNMP seems a natural choice. However, only a few works have been done in that direction.

In this paper we proposed a model that allows MASIF compliant platforms to be managed through SNMP. The proposed model brings together the advantages of the IETF's network management architecture, namely the current availability of commercial products, with the robustness and flexibility of mobile agent technology.

### 5. References

- [1] J. Oliveira, R. Lopes, "Distributed Management based on Mobile Agents", *Proc. of the 1st International Workshop on Mobile Agents for Telecommunications Applications – MATA'99*, October 1999, Ottawa, Canada.
- [2] Perpetuum Mobile (<http://www.sce.carleton.ca/netmanage/perpetum.shtml>).
- [3] A. Bieszczad, B. Pagurek, T. White, "Mobile Agents for Network Management", Carleton University, Canada, 1997.
- [4] E. Kaasinen, "Usability Challenges in Agent Based Services", *Proc. of the 6th International Conference on Intelligence in Services and Networks, IS&N'99*, Barcelona,
- [5] Goldszmidt G., Yemini Y., "Delegated Agents for Network Management", *IEEE Communications Magazine*, Vol. 36 No. 3 (1998) pp. 66-71.
- [6] R. Lopes, J. Oliveira, "On the use of Mobility in Distributed Network Management", *Hawaii International Conference on System Sciences – HICSS*, January 2000, Maui, Hawaii, EUA.