Using Mobile Agents for the Management of Telecommunication Networks

L. Moura e Silva[+], P. Simões[+], J. Gabriel e Silva[+]
J. Boavida[*], P. Monteiro[*], J. Rebhan[*], C. Renato[*], L. Almeida[*], N. Sthoer[*]

[+]Dep. de Engenharia Informática, Universidade de Coimbra, Pólo II, 3030 Coimbra, Portugal
[*]Siemens S.A., BU.NET, Rua Irmãos Siemens Nº 1, 2720-093 Amadora, Portugal

Abstract

This paper presents a preliminary overview of a research project called JAMES where we exploit the use of mobile agents technology in the management of telecommunication networks. The project is partially funded by *Agência de Inovação*, and was accepted as en Eureka Project (Σ!1921). The project partners are Siemens SA, University of Coimbra and Siemens AG.

I. INTRODUCTION

The main goal of the JAMES project is to develop an infrastructure of Mobile Agents with enhanced support for network management and try to exploit the use of this new technology in some telecommunications software products, namely performance and network management. A Mobile Agent corresponds to a small program that is able to migrate to some remote machine, where it is able to execute some function or collect some relevant data and then migrate to other machines in order to accomplish a task. The basic idea of this paradigm is to distribute the processing throughout the network: that is, send the code to the data instead of bringing the data to the code.

The existing applications in the management of telecommunication networks are usually based on static and centralized client/server solutions, where every element of the network sends all the data to a central location that executes the whole processing over that data and provides the interface to the user operator. By consequence, they are not flexible, they have problems of scalability and they produce too much traffic in the network.

The use of Mobile Agents in this kind of applications represents a novel approach and potentially solves most of the problems that exist in centralized client/server solutions. The applications can be more scalable, more robust, can be easily upgraded and customized, they reduce the traffic in the network and all this is achieved by distributing the processing functions over the network through the use of Mobile Agents instead of relying on a centralized location.

The JAMES project will try exploit all these technological advantages and see how the mobile agents technology can be used in software products that are been developed by Siemens SA. The project involves the development of a Java-based software infrastructure for the execution of mobile agents. The use of Java was motivated for reasons of portability and robustness of the applications.

In the last few years the use of Mobile Agent technology has received an extraordinary attention from several Universities and research institutes and a notable investment from leading companies, including IBM, Oracle, Digital and Microsoft [1]. Mobile agents have been applied in several areas, like mobile computing, electronic commerce, Internet applications, information retrieval, workflow and cooperative work, network management and telecommunications [2-5]. Several commercial implementations of mobile agents have been presented in the market, including Aglets from IBM [6], Concordia from Mitsubishi [7], Odyssey from General Magic [8], Voyager from ObjectSpace [9] and Jumping Beans from AdAstra [10]. Although these software products have some interesting features they are too much general-purpose and do not provide any special support for network management.

In our project, we are developing from scratch a new Mobile Agent infrastructure that is being tuned and customized for the applications we have in mind in the area of telecommunications and data network management. Our platform departs from other existing ones by a set of features that can be enumerated as follows: high-performance, security, fault-tolerance and robustness, support for network management, easy upgrading, disconnected operation, a 100% pure Java implementation and the inclusion of CORBA.

The platform will then be used in two software products: one in the area of TMN and another one for data network management.

The first prototype is already finished and we are now doing a benchmarking study to compare the use of mobile agents over traditional client/server solutions to see if we corroborate some of the advantages of this new paradigm in the field of distributed computing.

We are also developing a prototype application for performance management that will be used to evaluate the capabilities of Mobile Agents in this field of applications. This application will be presented in Section III. The next Section will provide a small overview of the JAMES Platform.

II. JAMES PLATFORM

The JAMES Platform provides the running environment for mobile agents. There is a distinction between the software environment that runs in the manager host and the software that executes in the network elements: the central host executes the JAMES Manager while the nodes in the network run a JAMES Agency. The agents are written by application programmers and will execute on top of that platform. The JAMES system will provide a programming interface that allows the full manipulation of Mobile Agents. Figure 1 shows a global snapshot of the system, with a special description of a possible scenario where the mobile agents will be used.

The host machine that runs the JAMES manager is typically a Windows NT or a Solaris workstation. This host machine is responsible by the whole management of the mobile agent system and provides the interface to the end-user operator, together with a Graphical User Interface (JAMES GUI) for the remote control of agents, places and applications. The central host provides a graphical interface to the end-user. Through that interface it will be possible to chose and configure a specific report, send the mobile agents to the network and present the final result in a graphical format.

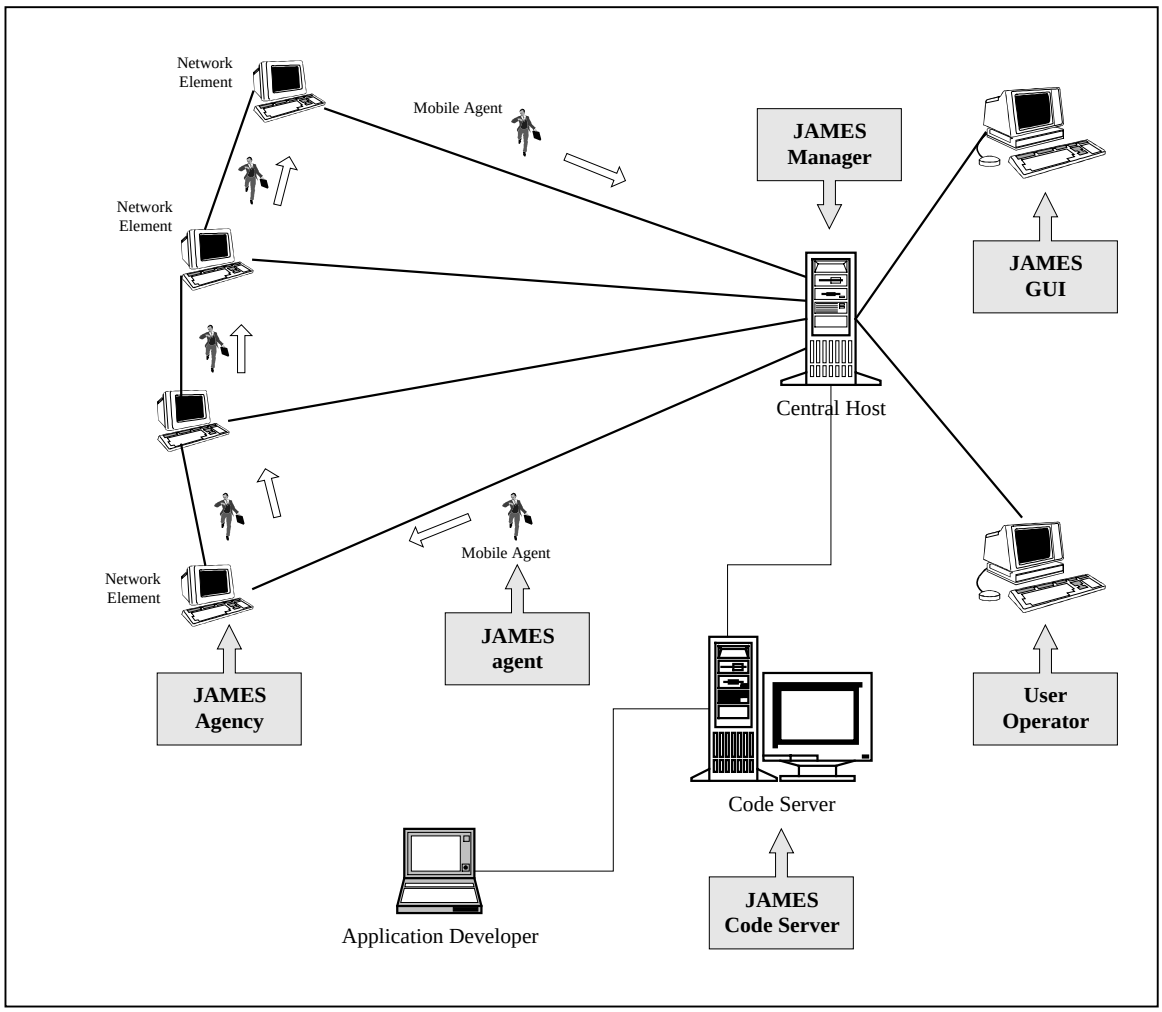

Figure 1: An Overview of the JAMES Platform

Every Network Element runs the Windows NT operating system and executes a JAMES Agency that enables the execution of Java-based agents. This machine will keep most of the data in the local disk instead of sending it to the central host. The JAMES agents will migrate through these machines to access this data and will produce short-size reports that will be sent back to the JAMES Manager. There is mechanism of authentication in the JAMES Agencies to control the execution of agents and to avoid the intrusion of non-official agents. The communication between the different machines is done through sockets TCP/IP. A special protocol was developed to transfer the agents across the machines.

The application developer writes the applications that are based on mobile agents. These applications are fully written in Java and should use the JAMES API for the control of mobility. After writing an application the programmer should create a JAR with all the classes that make part of the mobile agent. This JAR file is placed in a JAMES Code Server. This server can be a different machine or in the same machine where the JAMES Manager is executing. In both cases, it maintains a code directory with all the JAR files available and the mapping to the corresponding mobile agents. The communication between the client program that is used by the application developer and the Code Server is done by the HTTP protocol. The Code Server interacts with the JAMES Manager through a dedicated TCP socket.

III. PROTOTYPE APPLICATION

Since the JAMES platform targets the management of data and telecommunication networks, the evaluation of its behavior in real PM/TMN environments was considered as a crucial issue.

A. *Selection of the Application Field*

The specification of the benchmark application to be developed by Siemens had two major constraints:

- it should consider a real network management problem;
- there should exist prior applications addressing that problem by the use of traditional approaches.

With these constraints in mind, a prototype application was designed to provide O&M Destinations Reporting, a component of TMN Performance Management Traffic measurements, using the Mobile Agents Paradigm. For that purpose, Network Elements (NE) from the Siemens Digital Switching System (EWSD) have been selected.

A mature TMN application from Siemens (the SPOTS product) already addresses this problem, making it possible to evaluate the advantages/disadvantages of Mobile Agents versus the use of the client/server paradigm in a real TMN environment.

B. *Data Collection in the TMN Environment*

Data collection in the TMN environment is a primary issue for both telecommunication suppliers and providers. Administrative and maintenance tasks (billing, performance management, maintenance, etc.) require data collection from proprietary NE nodes and data processing for reporting purposes (either online[1] or offline[2]).

The software for network management is currently dominated by systems based on Client/Server technologies. In most cases, this approach results in monolithic, not scalable and hardly flexible solutions.

Typically, there are several network devices distributed across the network. In each device it is installed a static server (or agent) that is responsible for collecting raw data the local device, with little or even no pre-processing at all. At the manager host there is a client process that interacts with all the servers in the network devices, collects the information from them and provides the information required to the end-user. These servers (or agents) are static and proprietary processes. They are very difficult to upgrade or to customize. This client/server approach is not flexible, although in several cases it would be desired to extend/increase its functionality.

Some of the current solutions also suffer from an information bottleneck at the manager host due to its centralized nature. Some of the applications are very inefficient in the use of the network resources, since most of the data has to be sent from the network devices to the manager site in order to be processed.

Most performance management applications follow this client/server approach and suffer from several of those drawbacks, namely:

- they impose a huge traffic in the network;
- the management agents are stationary and difficult to upgrade or modify;
- the manager site is usually a point of congestion;
- the software lacks flexibility.

In some cases, it is also very difficult to correlate management data between sub-networks in order to provide a global view of the network.

Additionally, raw data information generated in the TMN Network Elements has some “unpleasant” characteristics:

- proprietary raw data formats;
- huge amount of raw data produced at a daily basis;
- variation of data formats depending on customer and/or software version installed.

Given this scenario, the use of Mobile Agents for TMN data collection provides several potential benefits. The application

1 executed immediately upon craft person request on a database
2 executed according to a pre-defined schedule on a database

developer can create diverse collector agent versions, each one matching the proprietary raw data format involved, the software versions involved and the customer involved. This means that several versions of the collector agent can coexist. Upon launching a task collection, the most suitable agent version is selected according to the NE's properties or, furthermore, an itinerary is assigned to the agent with missions depending on each NE specific properties.

Software upgrade requires only the coding of new agents or the automatic deployment of new agent versions over the managed network. This way, extension of functionality or installation of new software versions becomes simple, effortless and remotely performed.

C. *Overview of the Application*

The SPOTS application collects and processes traffic data from the EWSD switches in order to produce a set of performance reports. Right now, data collection and report building are two dissociated tasks: data is collected to and organized in a central relational database. This database is later used to produce the reports. This two-step approach, as well as the huge amount of data involved, significantly increases the amount of time necessary to produce each report. Despite the introduction of several optimization techniques, the system does not scale well.

The benchmark application, designated as "EWSD-Destination Reporting Application", is designed to reproduce a representative subset of the SPOTS reports using mobile agents to collect and process the management data. The structure of the application is shown in Figure 2.

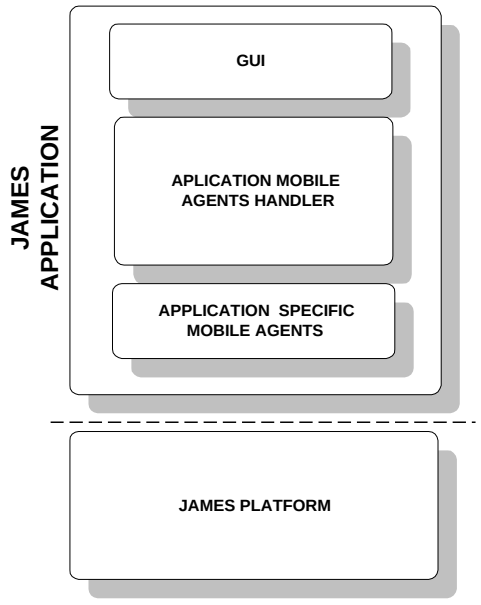

Figure 2: Application Structure

According to the Figure, the application is structured into three different modules:

- an application GUI that handles the end-user requests and outputs the reports (either in numerical and graphical format);
- an application mobile agents handler, responsible for the control of the *reporting features* through the usage of application specific mobile agents;
- and the application specific mobile agents which are assigned to fulfil the required reports.

D. *Reporting Features Description*

The benchmark application produces two different types of reports: *on-demand* and *predefined*/*scheduled* reports.

On-demand reports correspond to requests to be immediately executed over the *traffic destination raw data files* stored in the remote Network Elements. Two types of such reports are supported:

- Full Network Best / Worst Counters Reports, which allows to request either the best or the worst value of a specific *traffic destination counter* or two combined *traffic destination counters* (through a *math operation* regarding the specified measured object);
- and Time Behavior Reports, that allow the reporting of the evolution of one specific *traffic destination counter* or two combined *traffic destination counters* (also through a *math operation* regarding the specified measured object). Figure 3 presents a snapshot from such a report.

Predefined reports are a sort of report *templates* with a pre-defined behavior. This report templates can be used at a later time with user provided attributes like time window to be evaluated, Network Elements to be considered, type of traffic destination to be analyzed, etc.

Scheduling of a pre-defined report is possible. At the scheduled time, the concerned *mobile agent* is launched to the *network* with the user-provided attributes. The *scheduled report* is then considered to be *active*.

E. *Benchmark Study*

The development of the application is now complete, and several benchmarking tests are currently under way (using release 1.0 of the JAMES Platform). These tests are focusing on the following benchmark metric criteria:

- scalability and traffic bottlenecks, which are problems of the referred PM/TMN Product;
- dynamic execution, impact on mobile agent processing time (for each NE node) due to variation in the amount and complexity of the data to be processed;
- performance and mobility;

- network traffic; search for network bottlenecks conditions at both the manager site and agencies sites under huge traffic;
- persistency of agents and application data (under NE or network malfunction and disconnected computing);
- security;
- safeguarding and recovery mechanisms;
- robustness under stress testing;
- upgrading mechanisms for both agents and agencies (analysis of the qualitative and quantitative impact for a typical TMN environment);
- easiness to use and develop applications using the JAMES API;
- customization ability;
- interfaces to managed resources and environment.

This benchmark study, whose preliminary tests are already quite promising, is expected to show how effectively mobile agents can replace traditional approaches for this kind of applications. Furthermore, it will provide valuable feedback for the development of new releases of the JAMES Platform.

IV. CONCLUSIONS

The JAMES Project exploits a promising approach for the development of applications in the field of management of telecommunication networks: the use of Mobile Agents. We are developing a Java-based platform that is mainly oriented to this field of applications and will be used by two software products: one in the area of TMN and another for data network management. Within this project we expect to show that Mobile Agents can overcome the problems that exist with traditional Client/Server solutions.

ACKNOWLEDGMENTS

This project is partially supported by *Agência de Inovação* and was accepted in the Eureka Program (Σ!1921). Special thanks to the team of Java programmers that are developing the JAMES Platform: Paulo Martins, Victor Batista, Guilherme Soares and Luis Santos.

REFERENCES

[1] "Agent Product and Research Activities", http://www.agent.org/pub/activity.html

[2] "Intelligent Agents", Communications of the ACM, Vol. 37, No. 7, July 1994

[3] Björn Hermans, "Intelligent Software Agents on the Internet", http://www.hermans.org/agents/index.html

[4] T.Magedanz, K.Rothermel, S.Krause. "Intelligent Agents: An Emerging Technology for Next Generation Telecommunications", Proc. INFOCOM'96, March 1996, San-Francisco, CA.

[5] V.A.Pham, A.Karmouch. "Mobile Software Agents: An Overview", IEEE Communications Magazine, pp. 26-37, July 1998

[6] IBM Aglets Workbench, http://www.trl.ibm.co.jp/aglets/

[7] Concordia, http://www.meitca.com/HSL/Projects/Concordia/

[8] General Magic Odyssey, http://www.genmagic.com/agents/

[9] Voyager, http://www.objectspace.com/voyager/

[10] Jumping Beans, http://www.JumpingBeans.com/

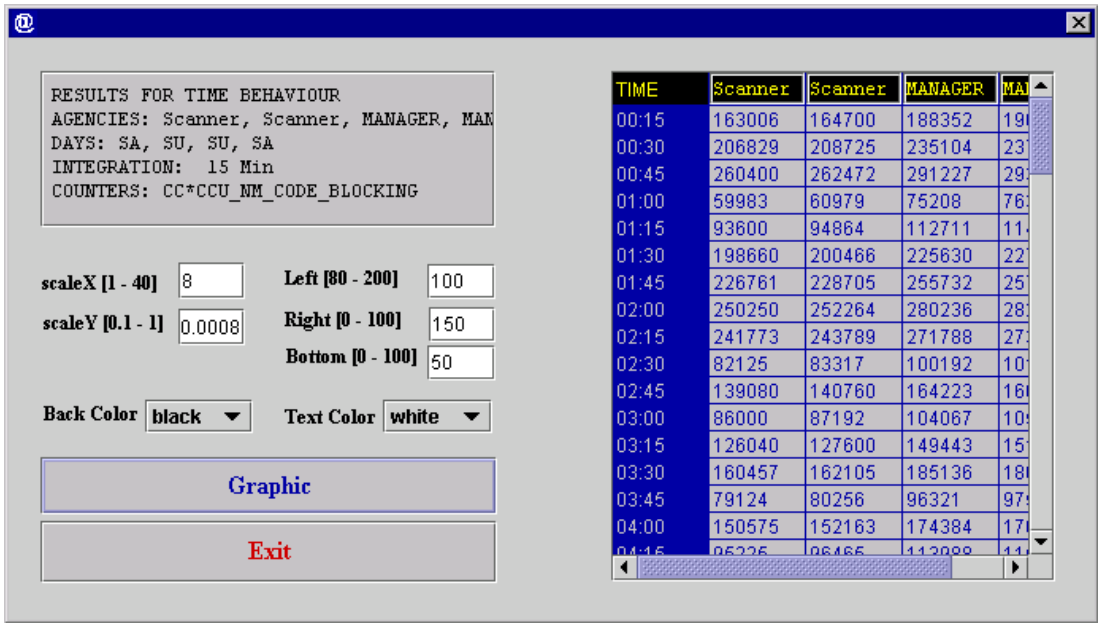

Figure 3: Snapshot from the Prototype Application: a Time Behavior Report in Raw Table Format