IMANetMS - An Intelligent Mobile Agent-Based Network Management Approach for a Real Scenario

Joaquim Ricardo Azevedo Teixeira and Maria João Viamonte

Abstract—**Current Network Management paradigms are rigid and lack flexibility. Therefore, a new kind of management system and strategy must be provided to enhance the reliability and performance of Network Management. This paper presents the results of our work on Mobile Agent technology as a new paradigm for developing Network Management applications. Problems existing in traditional Network Management are discussed and a distributed solution is proposed. In this new approach, Mobile agents are the lead characters, in a world where networks grow in a larger and complicated way. Finally, we present the results from an implemented architecture for a real network scenario. Our efforts resulted in IMANetMS, an Intelligent Mobile Agent Network Management System.**

Index Terms—**Mobile agents, network management, simple network management protocol.**

I. Introduction

Network management essentially involve configuration, monitoring and controlling the devices connected in a network by collecting and analyzing data from these devices. The data is locally stored in Management Information Bases (MIB). On the other hand, most network management systems operate Simple Network Management Protocol (SNMP) with a centralized configuration. When we use the Client/Server paradigm for network management purposes, it requires transferring large amount of management data between the manager and agents. This not only requires considerable bandwidth, but also can cause a bottleneck at the manager.

In addition, today's networks grow in a larger and complex manner and the problem becomes more severe. When we focus on the conventional centralized architecture, the conclusion can only be one: it is not sufficient to manage in an effective way, because a central entity routinely requests the status information of local units, which consumes a significant amount of bandwidth. Further problems are: lack of scalability; excessive processing load at the manager; heavy usage of network bandwidth by network management actions and management intelligence too centralized. [1]

However, there are several solutions which have already been investigated such as Remote Monitoring (RMON) and Management by Delegation (MbD), which introduce some degree of decentralization. Another approach, the use of software agents, and particularly mobile agents, to distribute and delegate management tasks, has also been investigated [2]-[4].

To achieve all that's been referred in common networks, the authors propose the use of intelligent MAs to manage the distributed network system. The mobile agent (MA) can be used to retrieve data from the MIBs to monitor the network flow in the distributed environment. Tasks are assigned to an agent and after, it can be sent to remote hosts completing the assignments (which may include not only monitoring but also configuration tasks). To conclude the results are carried back to the sender by the agent.

The solution aims to distribute the management mechanism to overcome the limitations of the centralized Client/Server architecture. A lot of research is currently being carried out to evaluate the applicability of agent technology to network management. MA can be used to ease the manager workload and reduce the bandwidth usage by delegation of authority from the manager to MA. MA is flexible and can be customized by user's requirements and launched from the manager. It can visit each network element according to the itinerary table, compute and compress the management data locally, only returning the result to the network manager. By moving a portion of the "intelligence" to the nodes where data is resident, the management decisions could be taken locally, avoiding the transferring of large amounts of data from the remote nodes to the central one [2], [5]-[9].

II. Centralized Implementations

Current network management systems such as SNMP for data networks, with a centralized model, are characterized by lack of distribution, a low degree of flexibility, re-configurability, efficiency, scalability, and fault tolerance. They also require network administrator to make real-time decisions and find solutions for all kinds of problems in the network. These network management applications deal only with data gathering and reporting methods, which in general involves substantial transmission of management data. This also causes computational overhead, a considerable strain on the network and a cause for traffic jam at the manager host. These management activities are limited, since they cannot do intelligent processing such as judgment, forecasting, decision making, analyzing data and make positive efforts to maintain quality of service. Therefore, all these problems recommend distribution of management intelligence by using MA to overcome the limitations of centralized management and meet today's requirements [2]-[4].

One important functional area of network management is Performance management, which has its basis on gathering statistics about network traffic and schemes to concentrate

Manuscript received November 15, 2012; revised January 18, 2013. This work is supported by FEDER Funds through the "Programa Operacional Factores de Competitividade - COMPETE" program and by National Funds through FCT "Fundação para a Ciência e a Tecnologia" under the project: FCOMP-01-0124-FEDER-PEST-OE/EEI/UI0760/2011.

The authors are with the Knowledge Engineering and Decision Support Research Center, Porto, Portugal (email: jra@isep.ipp.pt, mjv@isep.ipp.pt).

and present data. In some conditions like times of network stress, this client-server implementation and communication, generates considerable traffic that may overload the management station. Network Elements (NEs), as well as computers, gateways, routers, among others, often record a great quantity of data. Taking the conventional approach, all the data recorded in these remote nodes must be frequently transmitted to the central unit responsible for the network management [2]-[4].

The lack of automation and fault tolerance in network management is another issue, when this one obligates the administrator to continuously monitor some network management activities. More problems can be found in the traditional way of implementing a centralized network management solution, such as detecting and removing device failures. These procedures are very inefficient, as if we encounter difficulties only by looking at monitoring tasks, and the need of healing a network node with a specific problem arises, our network management solution becomes more ineffective [2]-[4].

III. Proposed Approach

A. Case Study

This Architecture was developed for a real scenario, the Knowledge Engineering and Decision Support Research Center (GECAD) network. This unit is a R&D centre settled at the Institute of Engineering – Polytechnic of Porto (ISEP/IPP. GECAD has two work areas located in different buildings, separated physically by a considerable distance within the campus of ISEP (Fig. 1). The first area is located on the fourth floor of building I. Consisting of two laboratories (soon there will be three) with several working places, a general meeting room, a laboratory of ambient intelligence for decision-making, the director's office and a space where a variety of network equipment, servers and workstations coexist and work together to support all network services. The second area is located in the entire N building, containing laboratories, offices and an IT space. The connection between I and N was facilitated by the fact that there is already a fiber optic link between the building I and the ISEP Datacenter - where the Campus network core resides - as well as another one that connects the Datacenter to N. The remaining path is completed in CAT5e UTP cable, allowing speeds up to 1Gbps.

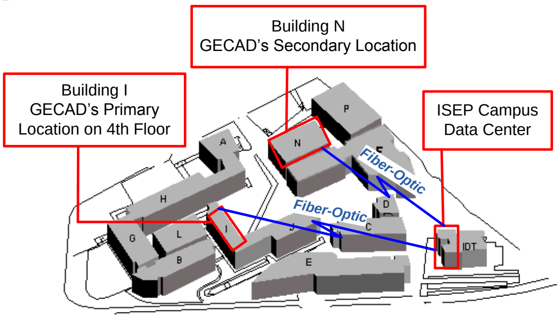


Fig. 1. GECAD network overview

The GECAD network is composed by a significant amount of Desktop computers (near 65, some of these are workstations with characteristics close to servers). The network is composed by 13 servers, where 4 of them are for purposes of High Performance and Massive Parallel Processing, and shared by several users. In addition, 5 network printers, 1 router, 7 switches and 3 access points that broadcast our wireless networks.

B. Network Management

Network management in GECAD is done mainly by a centralized sophisticated management solution. That solution was introduced to guarantee the full functioning of the network described based on a set of applications, which support the traditional paradigm. However, when it was configured it was planned to introduce a second paradigm, the mobile agents.

The tool is "GroundWork Monitor" [10]. This is an Open Source tool that integrates several other Open Source projects, such as "Nagios", "Apache" and "NMAP". The monitoring engine is "Nagios" and "GroundWork Monitor" was configured in a Linux Operating System. One of the major reasons that made this tool the chosen one relates to the fact that we can configure it to our liking or specific needs of our network. GroundWork Monitor, like any traditional application management, has tasks that involve the transfer of bulk data, such as SNMP extensive tables. These tables are obtained by using repeatedly the operation "GET-NEXT". For every "GET-NEXT", we need to wait for the answer so we can move to the next "GET", and for each "GET", only a table row is returned. If the table is large, this procedure has an impact on network resources, causing latency. There is an improvement over a more complex operation, the "GET-BULK", introducing enhancements by transferring several lines simultaneously. However, it requires that the users know the maximum number of data needed; otherwise at the station manager will arrive more data than necessary.

Six requisites were identified to evolve the network management solution at GECAD with the aid from mobile agents: diminish the excessive consumption of network resources. The use of a solution based on mobile agents allows us to move processing to the network elements, executing operations like "GET-NEXT" locally. This is a fundamental requirement, the ability to have mobile agents that "speak" SNMP and being able to filter what really interests us, distributing the load caused by the management process; increase the efficiency with management processes that involve getting data about resources such as CPU, RAM and disk space. The current tool has to repeatedly query the network elements. With mobile agents, processing can be moved and actions taken when problems are detected, without the administrator's intervention; create graphs illustrating the behavior of certain resources in a given network element. However, a traditional management solution, to obtain such information, uses polling and after receiving the response generates the corresponding graphs. A solution based on mobile agents eliminates the polls, since the agent is sent to the specific network element, migrating only when it has obtained all necessary data; collect data on the network equipments (router, switches, etc) that communicate via SNMP, more efficiently. Such tasks can be performed taking into account a polling distribution, eliminating the centralized approach. The agents can move to a network segment, for a given element (PC, server, etc.), which is as close as possible to the router or switch targeted, executing tasks from that point, avoiding the so called "bottleneck"; ensure fault recovery related to DNS and ARP caches, among other issues with network adapters, which cause many problems in communication in the network elements as they arise; monitor all processes vital to the smooth functioning of the network. These processes are located in various network elements and need to be monitored regularly. The load on the existing traditional application is very high, because it has to do a constant polling to consult ten, fifteen or twenty processes per network element. Generally, if one is stopped, no action is taken, and when it is, demands response, analysis, send the command to run for recovery and a reply with the result. The mobile agent can act in order to process all these queries in parallel and locally, seeking to be autonomous and intelligent, still recovering services on the network element and only after that, transmit the results.

C. *Architecture Proposed*

As shown in Fig. 2, the system architecture consists of three main parts: Core, Network Elements and the IP-Based Network for supporting the whole system.

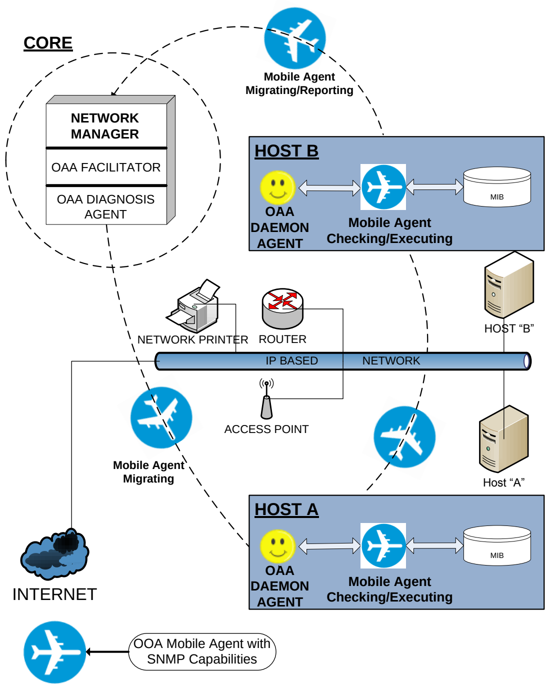


Fig. 2. Architecture proposed

The Core is the control center of the system. In it, is included the traditional management tool, "GroundWork Monitor". The core is responsible for the creation and initiation of mobile agents, as well as displaying the results returned by them. The vast majority of network elements have SNMP capabilities. These were also added to the mobile agents.

OAA Diagnosis Agent must be capable of create and control MAs, among other tasks. After some MAs have been sent, their state must be monitored along with the actions being performed by these ones. In addition to all the functions mentioned before, the OAA Diagnosis Agent receives the reports from the MAs and must be able to interact with its hierarchical superior, the network manager. It is a virtual gateway that provides this feature, a way for the network manager application to involve in the whole process of administration, giving the ability of communication between these two fundamental elements in the system.

The OAA Facilitator agent coordinates the community of agents. Each agent of the community registers their capabilities or features in the OAA Facilitator. When services are requested by an agent, instead of asking a specific one to perform a task, only the request is made, and then the OAA Facilitator decides which of the agents is/are available and able to respond. The request for a task may or may not be divided into subtasks to be performed by different and distributed agents.

The network elements may be routers, switches or workstations; all the components of a network that the administrator needs to monitor and configure.

The MA object has behavioral description, state information and attributes (static/permanent information). MAs need to be location aware in order to decide when and where to move. Moving an agent involves sending its code and state through an IP-Based network and this is patent in the trajectory shown in Fig. 2. Also in the same figure, we can see the MA making some checking's and executions along the way. This happens because of its ability to be intelligent, deciding where it needs to go and if it needs to do an execution based on previous analysis done at the arrival. MAs have the potential to improve the retrieval of SNMP tables in terms of network overhead. A MA can be moved to the network element where we wish to retrieve SNMP values. We can additionally create a type of MA to be used for global filtering by comparing and merging the results already saved with those just fetched. With this, MA brings more benefits. Not only the network manager and OAA Diagnosis Agent are relieved from handling heavy processing tasks, but also the MAs state size is prevented from growing rapidly.

OAA Daemon Agent is not mobile, however, belongs to the same community. The purpose of this agent is to receive mobile agents by providing features that allow them to restore their status, capabilities and attributes. The mobile agents belonging to this solution are part of a community of agents known as OAA, and have SNMP capabilities, so that they can perform management tasks among the network elements.

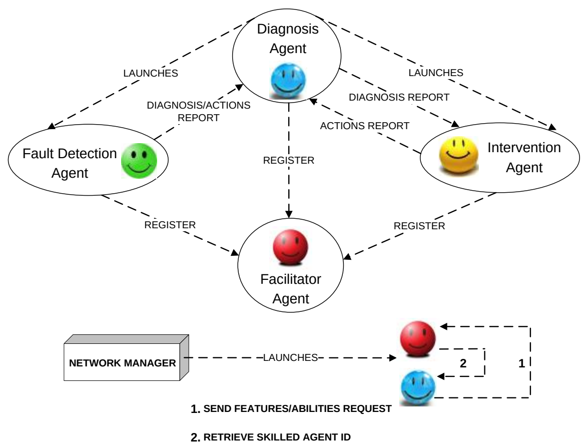


Fig. 3. Community agents & interactions

Fig. 3 illustrates all kinds of agents existing in the community. The agents with mobile capabilities are the Fault Detection Agent and the Intervention Agent. Fault Detection Agent is scheduled to travel on a particular route for the purpose of monitoring its operation. It is also able to resolve any problems identified, i.e. carrying the "intelligence" to the network elements. The Intervention Agent is called by a Fault Detection Agent when the first one cannot solve the problems detected. This model allows us to design “lighter” agents that move across the network with built-in fault detection competences, since they are endowed only with capabilities to solve the most frequent problems, becoming the Intervention Agents more sophisticated, and only moved under request or special conditions. The Diagnosis Agent and the Facilitator are launched by Network Manager, which also starts the mobile agents’ application. Whenever an agent is created in the community, it registers with the Facilitator, informing of its name and respective capabilities. The first agent to be instantiated is the facilitator. Diagnosis agent only then joins the community.

Once fully operational, it is possible to move multiple agents of the same type, and the Diagnosis agent is responsible for making the request to create the new ones, communicating those same intentions to the Facilitator. In the previous figure, Fig. 3, we see multiple interactions between the Diagnosis Agent and Intervention Agents, as well as with Fault Detection Agents. When a Fault Detection Agent is released, it has a certain route and several management tasks to perform. It is necessary at the end of these tasks, to compile a report on the actions taken and retrieve the results. This report is delivered to the Diagnostic Agent, which analyzes it to check whether there are additional needs. The interaction with the Intervention agent is different, because it requires more information to work. A diagnostic report, with problems on the network, resulting from operations carried out by Fault Detection Agents is needed.

IV. Implementation

Implementation was done in our approach with the help of Java language programming and two API’s: Open Agent Architecture (OAA) and AdventNet SNMP. The OAA platform was developed to integrate a community of heterogeneous software agents in a distributed environment. The platform is Open - the agents can be written in various languages (Prolog, Java, ANSI C / C + +, LISP) and operating systems. In this case, the barriers of language and operating system are minimal; is distributed - agents can be distributed across multiple machines on a network. This is a very important advantage, as in this case, it allows each task to be divided into several subtasks, and these subtasks will be executed by several agents in various machines; extensible - you can add and remove agents in time execution, allowing the creation of flexible scenarios [11]. AdventNet SNMP API gave the OOA agents the ability of “speaking” SNMP language. The purpose of the high-level APIs is to make it easier to develop management applications with the SNMP libraries [12].

Fig. 4. IMANetMS interface

A very important feature in our approach is the addition of Artificial Intelligence techniques to the MAs. Not only the mobility model was included, but also the capacity of a MA to leave a Host, get to another one and depending on the issues found there, act adequately and immediately.

Fig. 4 shows the Interface of IMANetMS, where all main functions can be done, before sending the agents, and after their execution. Several types of agents can be created, or removed, at any time, whether they are Intervention or Fault Detection Agents: there are agents that take care of resources, like RAM or CPU, agents that focus on network adapters and network equipment and even agents prepared to analyze a network element retrieving statistical data. We can create a path by adding IP addresses to a list. That list will contain a final route with all destinies that the agent(s) need to visit.

V. Results

Since the transmission of a MA to destiny creates less traffic, instead of transmitting all data, we can reduce significantly the network bandwidth for most of management traffic. On the other hand, to take advantage of this feature, each MA must be small and designed for meticulous tasks. This obligates some caution when designing an agent and customizing it, because the execution cost could be very high if we exceed it in some way. Each MA can be controlled in a decentralized approach and carry out its management tasks independently of its source node. With IMANetMS, our new approach is found on Fig. 5. IMANetMS is benefiting from local resources found in all network elements as well as processing locally and transporting code, state and reports from one node to another.

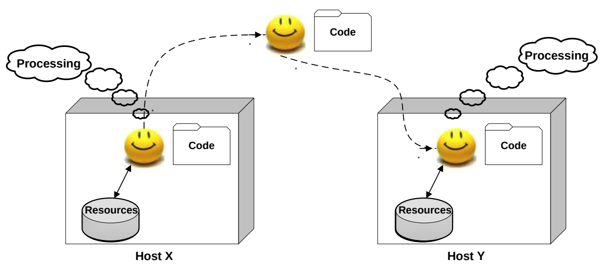


Fig. 5. IMANetMS approach

Was simulated the network management of GECAD for 5 weeks with the proposed approach. Services in the traditional centralized management application to check CPU load, RAM, vital processes working, and disk space on system partitions at the network elements, were removed. Mobile agents were in charge of this kind of management. All other management functions were intact in GroundWork Monitor. We now present some illustrative numbers of gains obtained by adopting this proposed “Hybrid” solution: we went from 255 services in “Nagios” to 190 services, where some of them had to check with a single service, if ten processes were running on a given network element. The centralized application had about 800 service checks every 15 minutes. Now it has 570. It is obvious that in addition to decreasing the polling, a portion of processing was moved to the network elements, where it’s done locally, taking advantage of all resources available in the network.

The next two figures are graphics automatically generated by the centralized application. Week 33 should be ignored, as well as 36, where there was a serious hardware failure in the machine where the “Core” of this solution resides. Before the simulation, the situation was a traditional week like 34 and 35. After implementing mobile agents, we can see a clear drop of warnings and some decrease in alarms (See Fig. 6). In week 38, despite only being processed data for Sunday, Monday and Tuesday until noon, the trend is clearly to lower these numbers. Fig. 7 provide new insights regarding this decrease per network element and service.

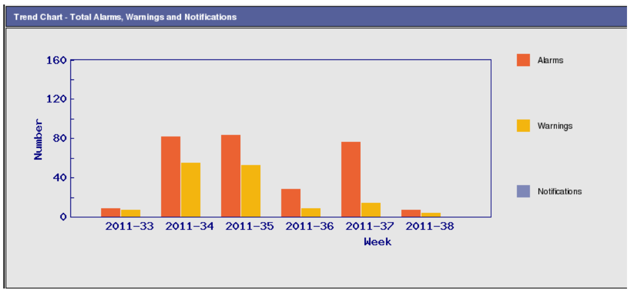


Fig. 6. Alarms & warnings

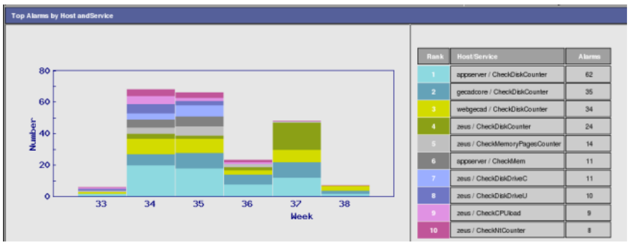


Fig. 7. Outages by network element & service

The proposed solution ensures a wide distribution management. The mobile agents responsible for performing management tasks related to RAM, CPU, disk space, among others, do it locally on the network element, consuming the resources that each element has to offer, returning the results at the end, having them transported by our IP network. It was reduced some of the characteristic excessive polling featuring such traditional applications.

The work with network equipments like Routers or Switches has been improved, thanks to the fact that an agent can migrate to a network element that is directly connected to the asset in question. Thus, we achieved two things: first, remove the processing and polling from the traditional; second, different types of mobile agents can work together to perform analyzes that complement each other by providing more complete data on the equipment. Finally, there are about 30 performance graphics on the traditional solution, built based on polling continually each network element. This type of work can be replaced by mobile agents, which study and process locally, and recreate the graphics only in the Core, at a later stage.

VI. Conclusions and Future Work

Common network management solutions based on centralized architecture have showed in the last years a lot of inefficiency when the managed networks are medium/big in scale. MA based approaches may have the solution to solve those problems. These implementations bring numerous benefits comparing to the traditional approach were we need to work harder and with some bad consequences in the whole system. Having a lot of network elements makes the network manager become the system bottleneck. MAs can be launched with the task(s) delegated to them, doing executions based on an analysis, and finish the job only returning the report with outcome to the network manager. We start avoiding in the network manager overloads and it can do more work concurrently. Besides, we are executing much management work locally, avoiding move large amount of data through the network.

Network management is being shared between the two paradigms (Centralized and mobile), which leads to a distribution with GroundWork Monitor and IMANetMS, with all the advantages that come from this junction. Tasks that were performed by traditional management solution can now be performed by IMANetMS, with more efficiency in the overall process of Network Management.

References

[1] A. Clemm, *Network Management Fundamentals*, Indianapolis, Ind.: Cisco Press, 2007.

[2] P. J. Martins, “Comparação dos paradigmas cliente/servidor e agentes móveis: um estudo em gerência de redes,” MsC, Universidade Federal de Santa Catarina, 2002.

[3] G. Goldszmidt and Y. Yemini, “Distributed management by delegation,” in *Proc. of the 15th International Conf. on Distributed Computing Systems*, pp. 333-340, 1995.

[4] D. Perkins, *RMON: remote monitoring of SNMP-managed LANs*, Prentice Hall PTR, 1999.

[5] T. Joaquim, “Uma Aplicação de Gestão de Redes baseada no paradigma dos agentes móveis,” MsC, Dept. of Informatic Engineering, ISEP, Porto, 2011.

[6] M. Bernichi and F. Mourlin, “Network Management with mobile agent toolkit,” in *Proc. Third 2008 International Conf. on Convergence and Hybrid Information Technology*, vol. 2, pp. 1012-1017, 2008.

[7] P. Braun and W. Rossak, *Mobile agents: basic concepts, mobility models, and the Tracy toolkit*, San Francisco, CA: Elsevier : Morgan Kaufmann ; Heidelberg : Dpunkt.verlag, 2005

[8] D. Gavalas *et al.*, “A mobile agent platform for distributed network and systems management,” *Journal of Systems and Software*, vol. 82, pp. 355-371, Feb 2009.

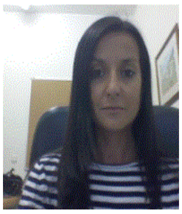
[9] H. Li and L. Meng, “Reconstruction mechanism based on distributed intelligent agents for network management,” in *Proc. 2008 7th World Congress on Intelligent Control and Automation*, vol. 1-23, pp. 5143-5147, 2008.

[10] GroundWork Monitor Community Edition (GWMCE). [Online]. Available:http://www.gwos.com/forums/index.php?/forum/43-groundwork-monitor-community-edition-gwmce/

[11] AdventNet. (2007). ADVENTNET SNMP UTILITIES 5. [Online]. Available: http://www.webnms.com/snmputilities/help.html.

[12] (2008). OAA® 2.3.2 Documentation. [Online]. Available: http://www.ai.sri.com/oaa/distribution/v2.3/2.3.2/.

Joaquim Teixeira received the MSc degree in Informatics Engineering - Systems and Networks from the Polytechnic Institute of Porto (ISEP/IPP), Portugal in 2011. Presently, he is a Network and Systems Administrator and also an Assistant Researcher at GECAD – Knowledge Engineering and Decision-Support Research Center of ISEP/IPP. His research interests include Network Management, Multi Agent Systems and Intelligent Mobile Agents in Network Management.



Maria João Viamonte is an adjunct professor of computer engineering at the Polytechnic Institute of Porto's Institute of Engineering. Her research interests are in multi-agent simulation, agent mediated electronic commerce and decision support systems. She received her PhD in electrical engineering from the University of Trás-os-Montes and Alto Douro. Contact her at Inst. Superior de Engenharia do Porto, Dep. de Engenharia Informática, Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal; viamonte@dei.isep.ipp.pt.