

DISTRIBUTED MANAGEMENT BASED ON MOBILE AGENTS

JOSÉ LUÍS OLIVEIRA

Universidade de Aveiro, DET

P-3810 Aveiro

E-mail: jlo@ua.pt

RUI PEDRO LOPES

Instituto Politécnico de Bragança, ESTiG

P-5300 Bragança

E-mail: rlopes@ipb.pt

During the forthcoming years, Internet-based concepts will continue to revolutionize, in an unpredictable way, the mode enterprises provide, maintain and use traditional information technology. Management systems will be a crucial issue in the struggle with this crescent complexity. However, new requirements have to be considered, due to the expectation of enormous quantities of different elements, ranging from an impressive network bandwidth availability to multimedia QoS-constrained services. Many researchers believe that mobile agent paradigm can provide effective solutions on these new scenarios.

This paper presents an implementation of management applications supported upon distribution and delegation concepts. For that it uses the current work of IETF's Disman working group enhanced with mobility provision. The mobility allows the distributed managers to adapt dynamically to a mutable environment optimizing the use of network resources.

1 Introduction

Internet technologies and concepts are dramatically changing the way enterprises provide, maintain and use traditional IT services. Management systems have to accomplish the profound effects of this evolution that will introduce into the net enormous quantities of different elements ranging from low resources devices to large-scale distributed applications. New paradigms must be available to deal with the pressures of these new demands. Many researchers believe that mobile agent paradigm can provide one of such answers.

The term Mobile Agent refers to autonomous programs that can move across networks, from node to node, and assume agent behavior, i.e. act on behalf of users or other entities [1]. Most of the research efforts on this topic has been developed under the context of the telecommunications market. Examples of such research have been: management distribution and delegation [2], network services deliver [3] network traffic optimization and network's fault tolerance [4]

Concerning in particularly network management, the scene has been dominated, during the past decade, by the IETF's Management Framework (shortly SNMP). However this model has suffered from several drawbacks during its

evolution. Even the recently proposed draft standards of SNMPv3 (April 1999) do present a set of lacks that avoids the model to cope well with the new near-coming demands of bandwidth, equipment and services.

Some management operations require large bandwidth from the network, due to the necessity to transmit, for instance, tabular information. Management protocols are not well designed to deal efficiently with this type of transfer load. By changing the paradigm “move data” to “move code” mobile agents seems to save precious network resources. But is it always an advantage to use mobile agents? Several issues and scenarios have to be analyzed in order to infer about that. We point a few:

- Enterprise organization – Management processes tend to mirror the enterprise organizational structure. Most enterprises are divided over several regional agencies that depend upon a central system to perform management operations. On these scenarios mobility of code may help reduce long lines permanent traffic that is typically associated with request-response management approaches.
- Management operations – The type of primitives that is requested also has impact on the efficiency of mobile agents. The processing of huge MIB data is best performed locally, moving code instead of data. On the other hand, the adjustment of a few parameters is more cost-effective if performed on a client/server base.
- Historical knowledge – The agent must keep knowledge about previous operations with the goal to improve its behavior over the time.
- Network resources – The resources needed by an agent to move must be available. Moreover, agents should consider moving when the overall efficiency would be greater than that of the original location.

Other topics, facing the design and development of mobile agents architectures and applications, range from security and privacy to the legality and ethics [1].

There is today a large debate on the use, or not, of mobile. A lesson that was learn by researchers and developers is that mobility is a technology that is, and will be even more, available. The market will decide when and how to use it concerning performance, security, development cost and end-user requirements. We will show an example of one such application.

This paper investigates and promotes the inclusion of mobility in distributed management, using, as a base, the recent work of IETF’s Disman working group.

2 Network Management

Network management issues were dominated over the last decade by two main approaches: the OSI Management Framework or shortly the CMIP/GDMO model and the IETF Management Framework (typically identified as SNMP). Due to its simplicity SNMP soon gained a larger set of followers relegating the CMIP to a

second and also insignificant market share (due to its powerful capabilities, supporters still remain specially across telecommunication operators).

2.1 The Simple Network Management Protocol - SNMP

The IETF network management framework is based on a reduced set of concepts (Figure 1).

The purpose of a network management system is to provide a basis for remote administration of network components and hosts. The basic structure of a standard network management system contains several managed nodes, which provides remote access to management instrumentation. These nodes are traditionally called managed agents.

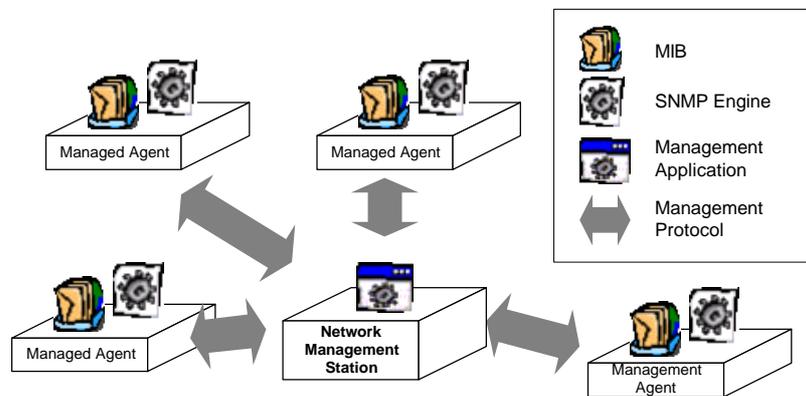


Figure 1 – Classic network management model.

In addition, it contains at least one entity with management applications, typically called Network Management Station (NMS), which concentrates all the data processing functions. The information exchanged between agents and the NMS are conveyed by a management protocol – the Simple Network Management Protocol (SNMP) [5].

The system is highly dependent on the central management station. If the NMS goes offline or the network isn't available for some reason the system is useless.

The SNMP framework defines both the structure and attributes used to face with managed resources (SMI and MIB), and the way these resources are retrieved (management protocol - SNMP). It does not define upper level tasks such as interpretation, correlation, and corrective measures. This type of decision must be performed by specific management applications.

2.1.1 SNMPv3

The SNMPv3 architecture has partially renamed the above-described nomenclature [5]. Each SNMP configuration is classified as a “SNMP Entity” composed by several interacting modules which combinations provide different roles (i.e. an agent, proxy or manager).

SNMPv3 follows the previous versions main goals and adds some features that intend to eliminate some of the earlier drawbacks. Among these, it addresses the need for security support and defines a flexible architecture that allows supporting frameworks already defined or that can be defined in the future. This architecture makes it possible to upgrade some portions of SNMP without disrupting the entire framework (Figure 2).

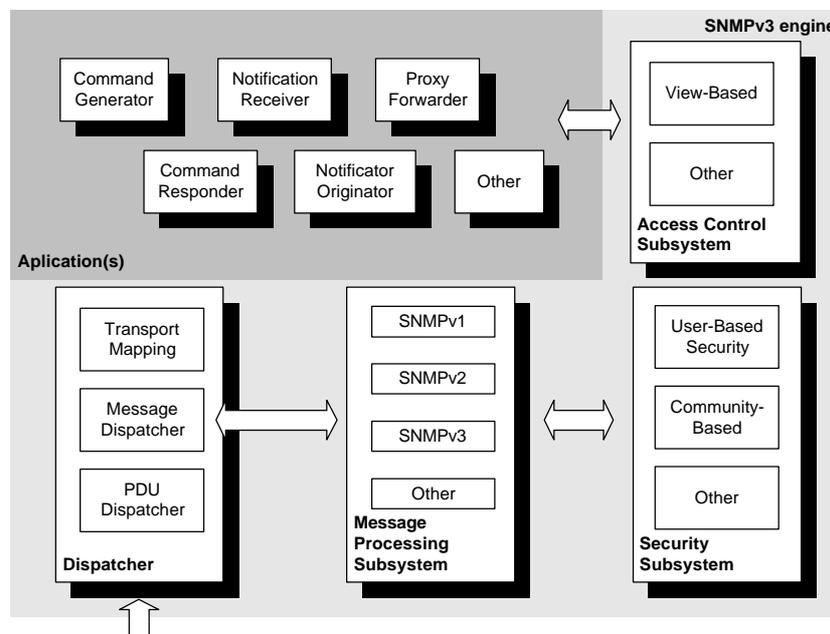


Figure 2 – SNMPv3 Entity.

The Application(s) use services from the SNMPv3 engine to send or receive messages, authenticate, encrypt and control the access to managed objects [6].

The Dispatcher subsystem coordinates the communication between SNMPv3 engine subsystems and differentiates modules belonging to the same subsystem. Based on each PDU, it determines which application should be invoked and coordinates the transport mappings.

2.2 *SNMP Drawbacks*

The SNMP management framework has been guided by simplicity and minimalism (SNMPv1). These characteristics have led it to a successful trail on industry.

But, and because not ever “small is beautiful”, when users and vendors gradually start claim more powerful and security, the SNMPv1 reveal itself not enough to provide the right answers to upcoming requests.

The SNMP standardization roadmap has drive today to a more complex model, larger set of documents although more complete specification. We envisage now a more closed-coupled framework to the CMIP/GDMOs that on the past. The number of documents has increase and the modularity of the framework allows easy a crescent number of new specifications.

Considering its drawbacks, a first one is related with to the lack of extensibility and scalability of the model on very large networks. This constraint results from the inability of a centralized manager to handle huge amounts of management information and also because centralized polling across geographically distributed sites is infeasible and expensive [7]. Moreover, system updates usually entail the modification of several agents or of the management station itself. In addition, there are occasions where it is necessary to cope with situations where the management station is not accessible. The classic management architectures are not well suited for low-bandwidth or disconnected operation.

Yet another problem is related to the successive increase of management information. Since MIB-II a large number of MIBs have been added to the Internet standardization track, driving to an impressive growing on the overall management information that manager system have to handle. There are currently nearly 100 standards-based MIB with a total number of defined objects approaching 10,000 [8]. Those numbers do not include an even larger and growing number of enterprise-specific MIB modules defined unilaterally by various vendors, research groups or consortia.

Several authors have addressed these problems along the past years [2][9][10] resulting in ad-hoc and partial solutions typically based on management distribution and delegation. Inside the IETF, the Distributed Management (Disman) WG was chartered to define an architecture where a main manager can delegate control above several distributed management stations thus improving scalability through distribution and allowing “off-line” operations.

2.3 *Disman*

The management distribution allows reducing the processing load on the traditional centralized management station (NMS) by delegation upon several Distributed Managers (DM) or upon more powerful agents. A DM is an SNMP entity that receives requests from another manager and executes those requests by performing management operations on agents or other managers.

Since the management entities are split over the network and collaborate between themselves by assignment, a hierarchy of several “islands” is created increasing the robustness and fault tolerance of the overall management system (Figure 3). Although if the access to the central manager is not possible, each DM may handle locally critical situations.

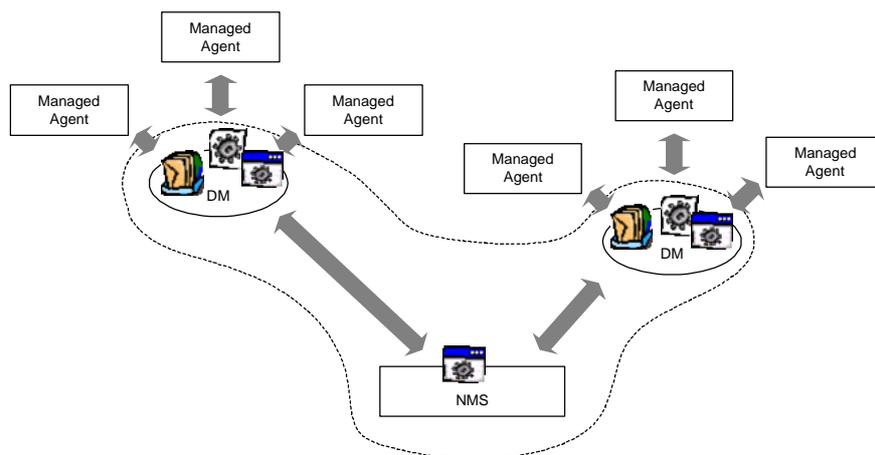


Figure 3 – Disman architecture.

The IETF Disman framework is based on distributed management applications and services. A distributed management application performs some management function, often by monitoring and controlling managed elements. The distributed management services can perform functions or store information once for all applications on the local system thus making a set of applications more efficient. Each service is provided by a specific MIB interface.

The framework is currently defining a set of services to address a large number of management operations [11]:

- Known Systems – provides a list of all systems that the distributed management system knows about and also stores type and attribute information for each of the known systems. The IETF PTOPO WG is providing even deeper information on those management entities by creating topological information of the network [12] [13].
- Management Domains – allow creating subsets of the list of know systems where systems are filtered by special criteria based on organizational boundaries (commercial section, administration, provision, etc.).
- Management Operation Targets – while organizational constraint bound the domain, targets are defined according to criteria that select several specific systems across one or more domains (maintains a list of known systems according to the kind of management function).

- Credential Delegation – this service deals with security, an issue that allows user credential delegation on DM applications. Basically, different users with different access control rights must be able to use an application. The idea is to concentrate SNMP access control policy on the manager and use this service to download credentials on DM.
- Delegation Control – it allows to define restrictive policies that limit average and burst polling, notification and broadcast rates on DM framework (defines policies to limit the abusive use of resources from management application).
- Scheduling – this service, currently addressed by [14], allows the execution of DM applications to be enabled and disabled at specified dates and times, periodically or based on the occurrence of events.
- Reliable Notifications – the recording of notification information is a solution against lost notifications (trap or inform PDUs that do not reach their destiny). The storage can occur on the sender of notifications or on receivers. The Notification Log MIB addresses this service.

In addition to this set of services, the Disman framework provides currently mechanisms to define management operations (scripts) so that the raw information can be processed locally [15]. Scripts are defined in a language supported by the DMs (examples can be TCL, Java or native code). This facility allows increasing the abstraction level and helps to simplify the management applications development. There is yet a mechanism to perform remote operations, such as ping, traceroute and lookup, from a DM site [16].

Currently there are being proposed eight MIB modules to address management operations distribution [17].

2.4 Disman evaluation

Although the distribution of management tasks by several autonomous stationary agents, as proposed by the Disman framework, is an improvement when compared to centralized architectures, it also suffers from some problems:

- The communication protocol, SNMP, uses IP addresses to identify the peer. If a firewall exists in between, it may be used invalid IP addresses (192.168.0.xx, for example). In this case it is not possible to contact the peer unless some tunneling mechanism is supported [18]. Moreover, the IP addressing scheme is too rigid in terms of peer location. Using higher level naming mechanisms makes it is possible to get references to objects inside or outside the firewall in a location independent way.
- System updates usually entail the modification of several services and management applications.
- Appending features usually makes the management applications or the management services more complex.
- The stationary agents may suffer from overloads either by limited capacity or insufficient computing power to process the tasks [19].

- The correlation of information from several DM is possible but only through an hierarchical flow. There must be a node that gathers all the information from instrumentation and correlates it.

Nevertheless, Disman is the right way to handle the crescent complexity of networks management – “Divide to conquer”. Considering just the technological issues associated to Disman implementation we see here a great field where the mobile agent paradigm can prove it capability with enrichment of the architectural functionality.

3 Mobile Agents

As computer systems evolve new features are added turning the system more and more complex. Far goes the time when computers were exclusive of teams of experts. Nowadays, users with diverse abilities and different levels of knowledge may use computer and network systems. As result, the number of tasks and the number of issues to keep track is continuously growing. In this context, users usually feel a need for extra eyes, hands, time and even brainpower.

In this scenario agent technology can make the difference. With the ability to perform actions on behalf of other programs or on behalf of the user, agents can be used on areas such as information retrieval [20], electronic commerce [21], personal mobility, telecommunications and user interfaces [22] just to name a few. A mobile agent is an example of these software agents that have the ability to migrate from host to host (Figure 4).

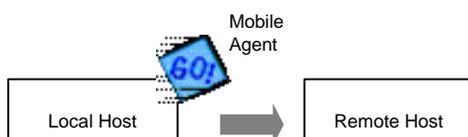


Figure 4 – Mobile Agent.

Mobile network agents are pieces of code that can be dispatched from one computer and transported to a remote computer for execution. Arriving at the remote computer, they present their credentials and obtain access to local services and data (maintaining the previous execution state). The remote computer may also serve as a broker by bringing together agents with similar interests and compatible goals, thus providing a meeting place at which agents can interact.

Several approaches consider mobile agents for managing networks and services due to the distributed nature, efficiency savings, traffic reduction and robustness [23-25]. Besides adding flexibility and the possibility to improve management efficiency it also brings some difficulties, namely it increases the agent management difficulty [26] and also introduces some kind of usability challenges and possibly threats [27].

4 Distributed Management based on Mobile Agents

Before addressing this section main issue it is important to establish a clear distinction between several concepts that use the same acronym (MA): Mobile Agents, Managed Agents and Management Agents.

A Mobile Agent is a program or process that uses some set of functions or methods (API) that allows it to transfer the code and the current state to other location. Mobile agents are usually executed in a specific environment (mobile agent platform), which also provides security features, cloning and copying capabilities. At this time, we choose to consider that a mobile agent is composed only by mobility aspect and leave the agency aspect for latter on. For now on we identify it as a MobAg.

SNMP Agents are computer entities responsible for collecting and storing management information local to the node and responding to requests for this information from the management station via a management protocol. For these reasons, they implement some set of MIB modules and an SNMP engine. The agent is as simple as possible so that it can be embedded in restricted environments, such as bridges or hubs. For now on we call it a MA (managed agent).

A Management Agent is a program or process (agent) that provides an abstraction to network resources with the purpose of facilitating management operations. It usually acts on behalf of another program or application, performs instrumentation operations and executes management operations. Note that a management agent can implement some SNMP module and thus have characteristics of SNMP agent. From this point on we will use the term ManAg to name it.

The development of a network management application, namely tools to gather and process management information, is a difficult task due to the ever-changing technology and standards. Moreover, aiming the interoperability it is important to maintain compatibility between the successive standards.

In the last years we have observed the emergence of some proprietary network management solutions, such as Java Management API [28] and Web-based management [29]. Although these architectures were developed over some of the SNMP weaknesses (special upon the SNMPv2 defect) they did not find, yet, a wide acceptance on network management community.

The IETF, with the Disman framework, addresses some of the architecture faults as a way to improve scalability, robustness and flexibility from the classical centralized architecture. A mobile agent approach can complement this framework by providing location transparency and facilitating some cumbersome aspects such as the SNMP engine installation, data correlation and others (see Table 1).

Table 1 – Problems solving both by Disman and MobAg approaches.

Task	Disman solution	MobAg solution
DM installation	The user must install each engine in every host.	Each engine may be distributed by agent's cloning.
Inter-agent and NMS Communication	SNMP, HTTP, FTP – based on IP.	Method invocation (either remote or local), KQML, etc. – more independent from the network protocol.
Data processing	Each engine processes data from a hierarchical set of other engines or distributed managers.	Each agent may process data from several sources. The processing is horizontal.
Data correlation	Must be done hierarchically by a higher level DM based on data collected by SNMP engines or other DMs.	A unique agent may move from node to node correlating the previous collected information and results with the local information.
Heavy load processing tasks	It is difficult to implement some kind of task distribution mechanism.	Agent cloning followed by migration to a platform with low usage – better use of network processing resources.
Engine updates	Using script and expression MIB.	Code on demand. Agent customization followed by migration to the destination.
Network faults	Inside the DM “island” lacks on the network lead to unmanageable situations	A mobile DM can adapt its position according to traffic, processing targets and network faults.

4.1 The Design Requirements for a Management System

The architecture meets the following objectives:

- Encompass current management framework, particularly the Simple Network Management Protocol. The SNMP framework is the most known and implemented architecture for network management. For this reason it is important that the system provides an SNMP engine. The Disman architecture, in particular, solves some distribution problems. It is important not to start from scratch but to reuse existing technology and improving it in scalable form.
- Let different management models be interconnected. The management community still did not find a consensus on the choice of the adequate network management model. There are several different solutions available, either

proprietary or standard. Maintaining the system architecture as modular as possible allows the interoperability between different models.

- Provide a set of migration services. The ability to move around is important for some management operations so it should not be disregarded. The architecture should provide a set of services to allow the migration and cloning of management agents, regardless of the implemented management models.
- Maximize flexibility in supported services. By providing an open architecture it is possible to add or change modules implementing different services with the same kernel. The agency in the MobAg paradigm is similar to an operating system, which can provide the same resources to different applications/services.
- Maximize usability. The increase of network and applications complexity is a reality and a never-ending process. Repetitive and boring management tasks should be performed automatically, without the user intervention. Moreover, the system should be permitted to take some decisions by itself (intelligent MobAg).
- Provide support for different policies. According to different sections of the network it may be necessary to apply different policies. For example, in an accounting department security management should be prioritized while in the multimedia section the management focus must be on bandwidth and QoS.
- Provide adaptive features to the management systems. For example, if some network section is having communication problems, it may be necessary to move the DM to a different location, where it can operate.
- The following model tries to cope with the above requirements through the use of MobAg upon Disman architecture.

4.2 Mobile Disman Architecture

The proposed architecture uses MobAg to implement Disman specifically in what concerns DM characteristics. The mobility support in DMs allows them to adapt to a changing environment and simplifies tasks such as agent and tasks distribution.

Figure 5 presents two situations of DMs with mobile characteristics. In the first situation (one) the DM choose to clone to a different management domain because the instantaneous load increased. Situation two presents an approach where the communication with the upper management station is interrupted. As an example, the DM may have detected a problem in the platform where it was installed and choose to migrate to a different location to continue its operation without assistance from the management station. When the station gets back on-line it may migrate to the original host and continue its operation. Other situation where this move may occur is when the interaction between the DM and some agent delivers high volumes of traffic. In this case, instead of generating traffic across several links the DM can move near the agent and interact with it locally. The DM MobAg can dynamically infer about these condition in order to adapt to the best network position and the best host.

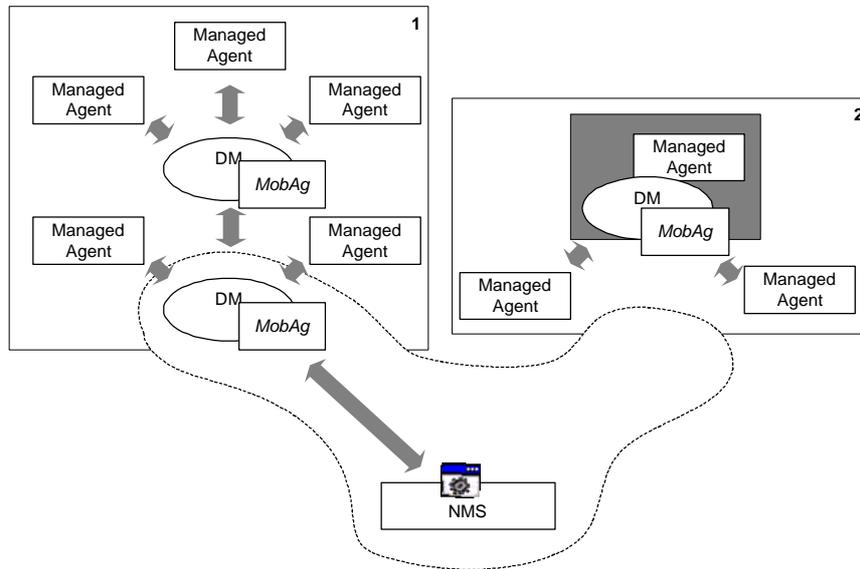


Figure 5 – Mobile Disman Architecture.

From the usability point of view, the user (manager) may define DMs in the topmost Management Station and set locally its behaviour. After creating the desired DM he can define an itinerary to be followed or some kind of distribution policy.

For this architecture to work as expected it is necessary to define an architecture for the DM so that it can implement the Disman framework and also have mobility characteristic. It must be considered that the SNMP framework does not expect the agents to move, so it is necessary to maintain knowledge of the current location of the agents. This fact also helps managing the DMs.

4.2.1 DM Scheme

The DM architecture was planned as open as possible but keeping in mind primary, the Disman conformance statements.

The main goal of this special agent is to collect raw data from several MAs, or from itself, and provide local management task and processing over that data. This gathering may be done by SNMP, method invocation (remotely or locally) or any other mechanism. By providing communication engines with a common external interface to communication services it is possible to inter-exchange modules maintaining the same architecture (Figure 6).

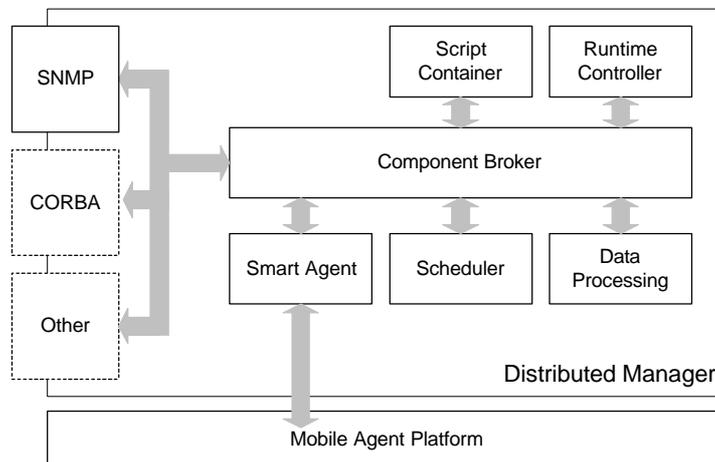


Figure 6 – Distributed Manager Scheme.

The instrumentation raw data must be filtered and processed in order to extract useful information. The Data Processing unit is based on the Expression MIB module and it collects algorithms, expressions and functions to process management information obtained from the managed nodes.

Management operations performed upon network components can be defined as scripts, according to the Script MIB module. The Script Container module maintains a collection of object references and it is used to carry the scripts and macros along when the DM migrates. The containers also provide a persistence service i.e. non-volatile storage. The Runtime Controller is responsible for recognizing the scripts language and version, so that it can choose the correspondent interpreter to execute it. In addition it provides methods to initialise, start, stop, suspend, resume and get the status of the script execution.

The DM MobAg have some level of autonomy so that it can make decisions on its own, such as when to move, when to clone, move to where. This characteristic is implemented by the Smart Agent module, which provides algorithms and functions to help deciding according to results from the Data Processing unit. If, for instance, the conclusion is that the DM platform is under heavy load, an agent will be dispatched to continue processing a specific task on another less loaded platform, or it can create, through cloning, several DM that operate in a hierarchical way under this DM's control.

The Scheduler launches notifications to registered modules either periodically or at a given time, according to the defined schedule providing time operational information that helps to guide management tasks activation and repetition.

The Component Broker is a lightness communication bus that controls and interconnects the other DM modules.

Since the DM do have mobility and cloning capabilities it must report do the upper level DM, the NMS at the uppermost node, each of such operations it executes (mobility monitoring). By doing this it is possible to maintain under control all the collaborating DMs.

4.2.2 Load Balancing

How well a distributed computing system performs, depends a great deal on an efficient usage of network resources. Load balancing consists on the balancing of the total workload among the various processors of the underlying sub-systems, in this particular case among the various DMs of the network.

In a static approach the manager distributes the estimated load along several points. Future re-planning must pass through the user/manager evaluation, which means, “managing the management system”.

Moreover, network management systems are intrinsically dynamic thus increasing the load-balancing problem. Considering moving load between processors we arrive to a not so simple task as have been shown by several authors [30-32]. This approach, however, uses balancing at the application level that, in spite of having much lower granularity, simplifies significantly the processing model (monitoring of traffic impact, estimation of “near” real-time management operations, output delays, etc.).

The information needed to perceive this not always is easy to define and to obtain, and is currently being matter for further investigation. Several parameters are being considered mainly based on two resources: the host performance and the network throughput and reliability.

5 Conclusions

Network management has been led by IETF SNMP framework, during the past years. SNMPv3 revised specifications have been approved as Draft Standard by the IESG in February 1999s and despite some limitations that can be pointed yet to this framework, there is now a great expectation upon it, after the SNMPv2 failure. The Distributed Management WG (Disman) is providing specifications that allow enhancing this architecture specifically through the delegation of management operations.

This paper presents an implementation of the Disman framework based on mobility. The system allows the distributed managers to adapt in a changing environment, increasing management efficiency, reducing management traffic and providing better use of network resources.

6 References

1. Pham V. and Karmouch A., Mobile Software Agents: An Overview, *IEEE Communications*, Vol. 36, No. 7 (1998) pp. 26-37.
2. Goldszmidt G. and Yemini Y., Delegated Agents for Network Management, *IEEE Communications Magazine*, Vol. 36 No. 3 (1998) pp. 66-71.
3. Krause S. and Magedanz T., Mobile Service Agents enabling Intelligence on Demand in Telecommunications, *Proc. IEEE GLOBECOM'96* (1996).
4. Lopes R. P. and Oliveira J. L., Software Agents in Network Management, *Proc. of the 1st International Conference on Enterprise Information Systems – ICEIS'99* (Setúbal, Portugal, 1999).
5. Case J., Mundy R., Partain D. and Stewart B., RFC2570 (I), Introduction to Version 3 of the Internet-standard Network Management Framework (1999).
6. Wijnen B., Harrington D. and Preshun R., RFC2571 (DS), An Architecture for Describing SNMP Management Frameworks (1999).
7. Sprenkels R. and Martin-Flatin J-P, Bulk Transfer of MIB Data, *The Simple Times*, Vol. 7, N. 1 (1999).
8. Postel J. and Reynolds J., Internet Official Protocol Standards, STD 1 (1998).
9. Oliveira J. L., *Arquitetura para Desenvolvimento e Integração de Aplicações de Gestão*, PhD Thesis, University of Aveiro (1995).
10. Brites A., Simões P., Leitão P., Monteiro E. and Boavida F., A Protocol-independent Notation for the Specification of Operations and Management Applications, *Proc. DSOM'94 - Fifth IFIP/IEEE International Workshop on Distributed Systems: Operations & Management* (1994).
11. Bierman A., Greene M., Stewart B. and Waldbusser S., Distributed Management Framework, *draft-ietf-disman-framework-02.txt* (August 1998).
12. Physical Topology MIB,
<http://www.ietf.org/html.charters/ptopomib-charter.html>
13. Oliveira J. L. and Martins J., A Methodology to Represent Logical Topology Information, *Proc. NOMS'94 - IEEE/IFIP 1994 Network Operations and Management Symposium* (1994), pp. 196-206.
14. Levi D. and Shoenwaelder J., RFC 2591 (PS), Definitions of Managed Objects for Scheduling Management Operations (1999).
15. Levi D. and Shoenwaelder J., RFC 2592 (PS), Definitions of Managed Objects for the Delegation of Management Scripts, (1999).
16. White K., Definitions of Managed Objects for Remote Ping, Traceroute, and Lookup Operations Using SMIV2, *draft-ietf-disman-remops-04.txt* (November 1998).
17. Distributed Management (disman) Charter,
<http://www.ietf.org/html.charters/disman-charter.html>
18. Mobile-IP: Supporting Transparent Host Migration on the Internet,
<http://anchor.cs.binghamton.edu/~mobileip/LJ/>

19. Shehory O., Sycara K., Chalasani P. and Jha S., Agent Cloning: An Approach to Agent Mobility and Resource Allocation, *IEEE Communications Magazine*, Vol. 36, N. 7 (1998).
20. Nekrestyanov I., O'Meara T., Patel A. and Romanova E., Building Topic-Specific Collections with Intelligent Agents, *Proc. of the 6th International Conference on Intelligence in Services and Networks, IS&N'99* (Barcelona, Spain, 1999).
21. Gleizes M., Léger A., Athanassiou E. and Glize P., Self-Organization and Learning in MultiAgent Based Brokerage Services, *Proc. of the 6th International Conference on Intelligence in Services and Networks, IS&N'99* (Barcelona, Spain, 1999).
22. Green S., Hurst L., Nangle B., Cunningham P., Somers F. and Evans R., *Software Agents: A review* (1997).
23. Perpetuum Mobile Procura,
<http://www.sce.carleton.ca/netmanage/perpetum.shtml>
24. Collective Intelligence
<http://www-iasc.enst-bretagne.fr/PROJECTS/SWARM/>
25. Bieszczad A., Pagurek B. and White T., Mobile Agents for Network Management (Carleton University, Canada, 1997).
26. Breugst M. and Choy S., Management of Mobile Agent Based Services, *Proc. of the 6th International Conference on Intelligence in Services and Networks, IS&N'99* (Barcelona, Spain, 1999).
27. Kaasinen E., Usability Challenges in Agent Based Services, *Proc. of the 6th International Conference on Intelligence in Services and Networks, IS&N'99* (Barcelona, Spain, 1999).
28. Java™ Management Home Page,
<http://www.javasoft.com/products/JavaManagement/index.html>
29. WBEM Initiative,
<http://www.dmtf.org/wbem/index.html>
30. Tantawi A. and Towsley D., Optimal static load balancing in distributed computer systems, *Journal of the ACM* Vol. 32, No. 2 (1998) pp. 445-465.
31. Ghosh B. and Muthukrishnan S., Dynamic load balancing in parallel and distributed networks by random matchings, *Proc. of the 6th annual ACM symposium on Parallel algorithms and architectures, SPAA '94* (1994) pp. 226-235.
32. Muthukrishnan S. and Rajaraman R., An Adversarial Model for Distributed Dynamic Load Balancing, *Proc. of the tenth annual ACM symposium on Parallel algorithms and architectures, SPAA '98* (1998) pp. 47-54.