Providing Customisable Network Management Services Through Mobile Agents

David Griffin[1], George Pavlou[2], Panos Georgatsos[3]

[1] Department of Electronic and Electrical Engineering, University College London, Torrington Place, London WC1E 7JE, United Kingdom
D.Griffin@ee.ucl.ac.uk
[2] Centre for Communication Systems Research, School of Electronic Engineering, IT and Mathematics, University of Surrey, Guildford, Surrey GU2 5XH, United Kingdom
G.Pavlou@ee.surrey.ac.uk
[3] A.T.T.S. Dept., Algosystems S.A., 4, Sardeon St., 171 21 N. Smyrni, Athens, Greece
pgeorgat@algo.com.gr

Abstract. Telecommunications network management has attracted a lot of attention in terms of research and standardisation in the last decade. TMN and TINA architectural frameworks try to address the management needs of broadband networks and services. They both cater for multi-domain, multi-operator environments while they also allow for electronic customer access to management services. These services, though, are fixed in the sense that new features can only be added after a full research-standardisation-deployment cycle and are static as far as their use: they execute according to their in-built logic and clients may customise their operation only through tuning standardised operational parameters before service execution. The advent of mobile agent technologies opens new possibilities, allowing dynamic and customisable services to be offered to clients. In this paper we explain how mobile agents can enhance traditional connectivity management services, presenting the relevant architecture and design of a customisable network management system based on mobile agents.

1 Introduction and background

A significant amount of research and standardisation activity took place in the area of telecommunications network management over the last decade. The culmination of this work is the Telecommunications Management Network (TMN) [1], with systems based on the relevant standards currently being deployed. More recently, the Telecommunications Information Networking Architecture (TINA) [2], has tried to integrate management with service control for future multimedia teleservices, integrating and evolving the concepts of TMN and the Intelligent Network (IN). While the TMN currently uses the manager-agent model and OSI Systems Management (OSI-SM) [3] protocols as the basis for its interoperable interfaces, TINA advocates the use of more general purpose distributed processing technologies such as the OMG Common Object Request Broker Architecture (CORBA) [4].

Both the TMN and TINA architectures cater for multi-domain multi-operator deregulated telecommunications environments, which allow also for customer access to management services. The TMN defines the inter-domain X interface, specific classes of which can be used between operators in a peer-to-peer fashion; they may also be offered to other Value Added Service Providers (VASPs) and customers e.g. the EURESCOM Xcoop and Xuser interfaces respectively. TINA defines the LNFed and Ret2Ret reference points for peer-to-peer network and service management co-operation respectively; it also defines the ConS reference point for providing connectivity services to Retailers and the Ret and TCon reference points for providing services to end-users [5]. Both TMN and TINA based management systems have been designed, developed and demonstrated in a number of research projects in Europe and elsewhere while management systems offering Customer Network Management (CNM) services have started to be deployed.

In the emerging multi-service telecommunications environment, customers will be able to search for the services they require through electronic brokers. They will subsequently be able to subscribe and customise their services electronically, while the whole set of interactions required for service provisioning will take place almost instantly, in an automated fashion supported by integrated management systems. Customers will be able to monitor the usage of their services, peep/pay their bills, modify service features and unsubscribe electronically. An example of such a service that may be offered to corporate users is a Virtual Private Network (VPN) service.

In such environments, the features of management services offered to customers are first researched, standardised and eventually implemented. This process takes a long time e.g. research on TMN-based VPN services over SDH/ATM has taken place over the last five years but such services are not yet offered to customers. In addition, any modification to the interfaces that support those services, e.g. for providing more sophisticated features that were not thought out in advance, needs to go through the full research, standardisation and deployment cycle.

The recent emergence of “execute-anywhere” languages like Java has made code mobility possible and has given rise to research and development into frameworks for mobile software agents. TMN and TINA use static manager-agent and client-server approaches respectively, static in the sense that the capabilities of the agent or server part are statically defined and cannot be changed. The use of mobile agents offers new possibilities in the sense that logic may be sent to execute in locations which are largely “unaware” of the functions and capabilities of those agents. Such a facility may enable the provision of management services in a different, more flexible and dynamic fashion and this is the subject of intense current research.

Mobile agent frameworks are currently addressed by two standards bodies. The Federation of Intelligent Physical Agents (FIPA) [6] looks at high-level semantically rich interactions between software agents that deploy some form of intelligence and adaptability, having its roots in Distributed Artificial Intelligence (DAI). OMG looks mostly at the issue of mobility according to a standard interoperable framework through its Mobile Agent System Interoperability Facility (MASIF) [7]. In the latter, *agent systems*[1] model the execution environment able to host mobile agents and correspond roughly to OSI-SM *agents* in the TMN manager-agent framework or to the TINA concept of the Distributed Processing Environment (DPE) *node*. Within an agency, *fixed* or *static* agents provide the bare-bones functionality which is statically defined but this can be augmented dynamically through *mobile* agents which are sent to execute there.

The flexibility potentially offered by mobile agents has led a number of researchers to consider their applicability to network and service control and management environments. Breugst [8] considers mobile agents in the context of the Intelligent Network and proposes an agent-based architecture for "active" IN service control. Biesz [9] discusses the general issues of using agents for network management while other researchers have presented specific case studies using mobile [10] and intelligent agents [11]. In this paper we examine how mobile agents can enhance traditional connectivity provisioning management services, allowing flexible customer network management with respect to performance, fault and dynamic reconfiguration which are driven by customer-defined policies. This work examines also the applicability of mobile agents to TMN and TINA-based customer network management services.

The proposed enhancements to connectivity provisioning management services make these services usable in a 'dynamic' rather than 'static way', where the terms 'static' and 'dynamic' are used as follows. In traditional service provisioning systems, customers subscribe to certain network management services according to their needs. During service subscription, customisation of certain management aspects to a particular set of customer requirements may take place - by selecting a subset of offered service features. However, this presents a 'static' approach with respect to the use of the services by the customers, who use the services as they have been subscribed, and services execute according to their 'built-in' logic. This approach is valid as long as the management needs of the customer environment are static, which however is not generally the case. Alternatively, a more flexible approach should be followed. Network management services should be built and offered to customers so that they leave several 'degrees of freedom' for use. Customers will modify the use of the offered services not only during subscription time but also during service operation, according to the dynamics of their environment.

In this paper, we demonstrate how customers may be offered an initial set of management services which could then subsequently be customised by sending customer-owned mobile agents to execute in the provider's environment. With mobile agents it is possible for a client to deploy specific functionality *at run-time* in a server to add value above and beyond the server's basic facilities. Following deployment, the mobile agent may act autonomously to interact with the local environment and make *local* decisions which may then be implemented as *local* management actions without needing to interact with the remote client every time a decision is required. The provided facilities correspond roughly to the TMN Xuser interface and the TINA

---

[1] While MASIF uses the term *agent system*, some implementations - notably Grasshopper - use the term *agency*. Our implementation is based on Grasshopper and we use the term *agency* in our system and in the remainder of this paper.

ConS reference point but they support extensible and customisable functionality. The proposed architecture and associated realisation are being pursued in the ACTS MIAMI project (Mobile Intelligent Agents in the Management of the Information infrastructure) [12], which examines the impact and possibilities of mobile agent technology to network and service management in general. MIAMI has defined a case study and associated environment which will allow co-operating customers dynamically to form Virtual Enterprises (VEs) for providing a particular service to end-users [13]. The VE makes use of services offered by a Active Virtual Pipe Provider (AVPP), a business role similar to the TMN VASP or the TINA retailer. The AVPP provides a programmable, dynamic virtual private network, as needed by the virtual enterprise. The AVPP makes use of connectivity services offered by a Connectivity Provider (CP), which is a business role similar to the TMN Public Network Operator (PNO) or the TINA Connectivity Provider.

The architecture presented in this paper concentrates in the CP domain of the MIAMI system. Mobile agents are used both in reference points offered to customers and also internally within the CP domain, trying to implement better performance and fault management functionality. The work presented here is being verified and validated through an implementation which will be followed by a field trial. The status of the implementation work, which aims to provide a proof of concept for agent-based customisable network management services, is discussed in more detail in the summary and conclusions section. It should be finally noted that the types of agent-based services discussed in this paper require strong security guarantees, especially as they target telecommunications environments which allow customers some degrees of freedom. Security is orthogonal to the architecture presented here and is not considered in this paper; the reader may consult [14] for a general consideration of security issues and associated mechanisms in agent-based systems.

The rest of this paper is structured as follows: section 2 introduces an example network environment to be managed; section 3 introduces our architecture for agent-based network management systems; section 4 discusses the way in which the management services offered by a Connectivity Provider may be customised by their users; section 5 illustrates the flexibility of customisable management services with an example; section 6 discusses our implementation approach; and section 7 presents the summary and conclusions, examining the benefits and drawbacks of such an approach.

2 A typical networking scenario

In the MIAMI project, and in this paper, we assume that the network scenario consists of an end-to-end IP network which interconnects the participants of a Virtual Enterprise (VE). In addition to standard best-effort, Internet quality connectivity for general email and web browsing the VE users also require access to higher quality connectivity facilities for real-time services such as high bandwidth video conferencing or for high speed access to large files. The users expect to pay a premium rate for guaranteed quality services, but they also wish to use lower cost, and correspondingly lower quality services for more general purpose communications.

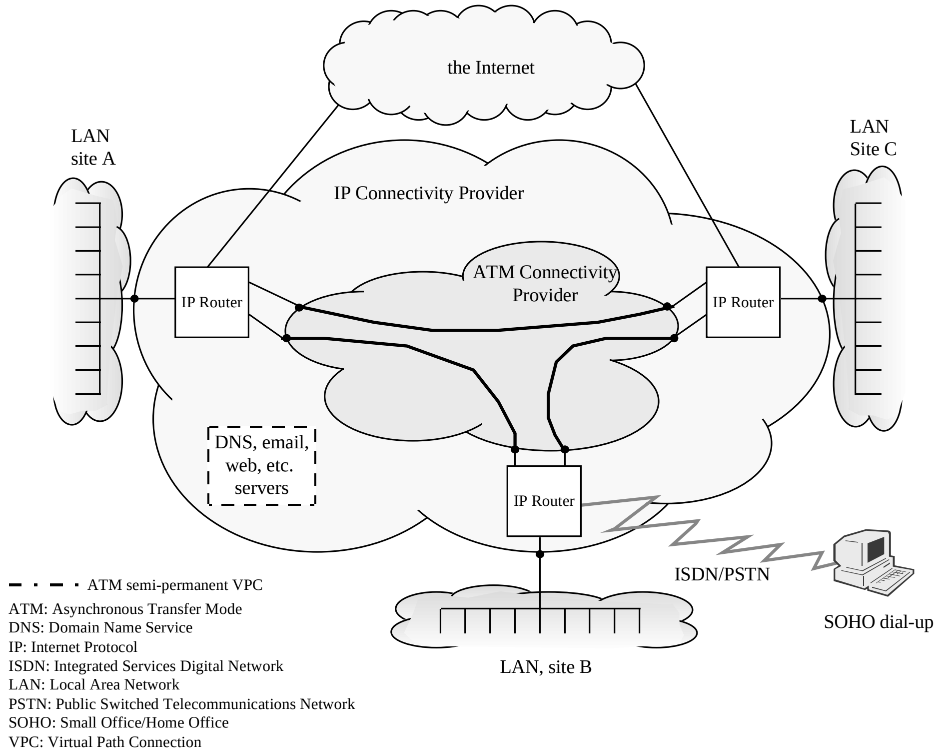

Figure 1 A possible networking scenario

In addition to providing access to the Internet, the IP CP also makes use of the services of an underlying CP - offering semi-permanent ATM connections in this case - who is able to provide guaranteed quality leased lines between the IP CP's routers. Figure 1 shows the network scenario we assume in the rest of this paper.

The service provided by the ATM CP is an end-to-end Virtual Path (VP) service offering PVPs (permanent VPs) between specified termination points. Associated with each VP is a number of parameters which defines the capacity of the connection and the level of performance to be provided in terms of end-to-end delay, delay variation and tolerable cell loss ratio. VPs may be created, deleted and modified through client management actions. The clients may monitor VP usage and performance statistics, and initiate fault monitoring activities on their resources.

Issues associated with inter-administration connectivity and federation of management systems are outside the scope of this paper although the dynamic customisable approach presented for a single CP domain could also apply to a multiple CP, inter-domain environment. The remainder of this paper will concentrate on the ATM CP, although the issues discussed are also generally relevant to CPs offering managed services for any network technology.

3 An architecture for agent-based customisable management systems

When treated as a black box, the CP's management system can be seen as offering two main classes of interface (see Figure 2): the interfaces to its clients (Dynamic Connectivity Management (DCM) and Contract Negotiation (CN) interfaces in the figure which are described later in section 3.2) over which it offers a set of services for the management of ATM connectivity; and the interfaces to the underlying Network Elements (NEs) which are ATM VP cross-connects in our scenario. The interfaces to the NEs will be based on the technology offered by the vendors of the switches - Simple Network Management Protocol (SNMP) or Common Management Information Protocol (CMIP); if CORBA-based interfaces are available then this is also an option. It is unlikely that commercial switches will have embedded agent enabled interfaces in the immediate future.

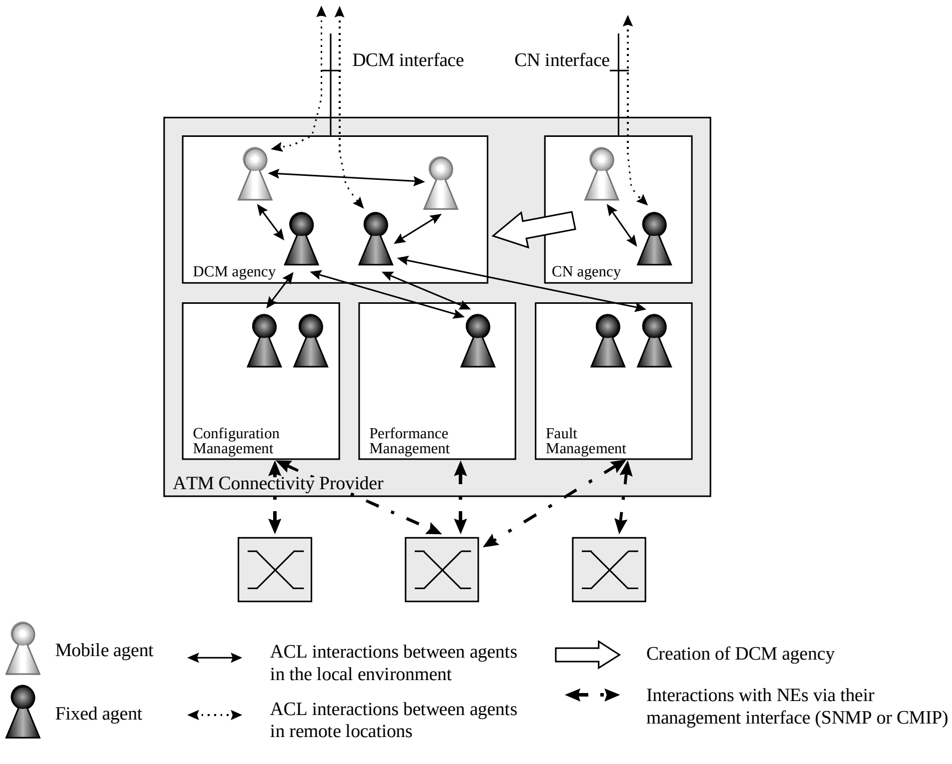

Figure 2 High-level architecture of an agent-based CP

The architecture for our system draws heavily on traditional network management systems, TMN [1] and TINA [2] in particular. There is a network management level which contains functionality for Fault, Configuration and Performance management activities; this is above the network element management level which performs element specific management activities on the set of network elements below it (with possibly one or more sub-network layers between [see Figure 3]). The management software at the network and element management levels resides in general-purpose computing workstations which are interconnected and also connected to the underlying network equipment by a communications network. The latter network, termed the Data Communications Network (DCN) in TMN and the Kernel Transport Network (kTN) in TINA, is logically, if not physically, separate from the underlying *managed* telecommunications network.

The main difference of our architecture compared to traditional network management systems is that we have replaced a *static* service management level with a programmable layer - the DCM agency - above the network management level. This is where client mobile agents may execute and customise the offered management services according to the requirements of the clients. In summary, compared to a TMN system, the architecture shown in Figure 2 may be mapped as follows: the Configuration, Fault and Performance Management agencies each span the Network- and Network Element Management layers while the DCM and CN agencies map to the Service Management layer.

3.1 Agent-based communications

We use the term Agent-Based Communication (ABC) to refer to the mechanism by which agents communicate with one another. The term implies that specific protocols and interface definitions are used which could either be based on general distributed systems techniques for remote method invocation (CORBA or Java RMI, for example) or on higher level semantic/AI languages such as KQML [15] or FIPA’s Agent Communication Language (ACL) [16] which support interactions with “semantic heterogeneity”. The specific languages and communications protocols are dependent on the chosen agent platform and are not of direct concern in this paper. Mobile agents may communicate with fixed agents and mobile agents may communicate with other mobile agents using ABC.

The communication between agents may take several forms: to raise asynchronous notifications, to query specific agents to retrieve information and to invoke operations. It is assumed that there is a mechanism for publishing the facilities offered by an agent operating in the server role - i.e. a way of formally specifying an agent’s interface. In the design of our system, the Unified Modelling Language (UML) [17] is used to specify formally an agent’s interface, being subsequently mapped to the Java language. It is further assumed that an event/notification service is offered by the host environment for disseminating the events raised by agents based on filtering criteria. Today's agent platforms, including those based on OMG's MASIF specifications, do not currently offer event/notification services. In our environment it has been necessary to implement notifications in a non-generic and fairly inelegant way, on a case-by-case basis. It should be noted that the MIAMI project is currently in the process of extending the Grasshopper agent platform [18] to allow communication between agents using FIPA’s ACL.

ABC extends beyond the local agency to allow communication with agents in remote execution environments. This implies two methods for communications in agent systems: either remote operations may be invoked through ABC (in a similar way to traditional distributed systems based on statically located objects); or mobile agents may physically travel to the remote agency where they may run in the local environment and invoke the *same* operations through *local* (i.e. intra-node) rather than remote (i.e. inter-node) ABC mechanisms.

By relying on mobile agents, an active and dynamically adaptive management system can be built which is not fixed and limited by initial deployment decisions at system design or build time. The choice of which communications method to use - remote or through mobile agents - is an issue which may even be decided dynamically, even at system *runtime*. It is possible to create and deploy a mobile agent when the communications overhead between remote systems rises above a certain threshold, for example. This, however, would be at the cost of physically transferring the agent to the remote execution environment.

3.2 High-level architecture

Figure 2 shows the overall architecture of the CP's management system. The management system consists of three separate agencies for the main management activities of the CP: one each for configuration management, performance management and fault management[2]. In addition there are two agencies which represent the two classes of interface to the clients of the CP: the CN (Contract Negotiation) agency supports the CN interface, and the DCM (Dynamic Connectivity Management) agency supports the DCM interface. Within these latter two agencies a number of fixed and mobile agents may execute. The fixed agents are provided by the CP, at initialisation time, and form the agent-based interfaces to the basic management services of the CP. The mobile agents belong to the clients of the CP and are dynamically created by remote clients.

The role of the CN interface is to provide access to the CN agency for *negotiating* the contract between the CP and its clients. The contract defines the set of agreed basic management services to be provided through the DCM interface.

Following the completion of the contract negotiation phase, a DCM agency and a DCM interface will be instantiated to provide the client with access to the agreed management services. It is through the DCM interface that the client may dynamically invoke management operations and that active network management services are realised through client programming and customisation via mobile agents.

---

[2] Other management functional areas and management services such as accounting and security are also envisioned, but we currently limit ourselves to these three.

3.3 Operational scenario

Initially, a potential client is unable to invoke CP management services for two reasons: physically it does not have access to a management interface, and legally it does not have a contract with the CP. The first step is to negotiate a contract. A fixed agent in the CN agency offers an interface to allow contract negotiation. This negotiation can be achieved in two ways: either the client creates a mobile agent to move to the CN agency and negotiate locally with the fixed agent; or the client may communicate remotely with the fixed agent.

Following successful contract negotiation, the CP creates an agency and DCM interface for the client. This involves the creation of one or more fixed agents in the DCM agency to offer specific interfaces to the management services which feature in the contract. The fixed agents tailor (in a *static* sense) the management services of the CP to the requirements of the client and to limit access to the services according to the terms of the contract. For example, not all management services may be made available to all clients, or the geographical coverage of, say, configuration management may be limited to specific locations. In other words, the fixed agents operate as *proxies* to the configuration, performance and fault management services of the CP.

When a management service - to create a new VP, for example - has been invoked, either by a locally running mobile agent or by a remote operation from the client, the fixed (proxy) agent invokes the corresponding operations on the agents in one of the configuration, performance or fault management agencies within the CP. It is within the latter agencies that the real management work - such as the creation of a VP - is achieved. Through the activity of the CP's configuration, performance or fault management systems, modifications are made to the network elements through their management interfaces (SNMP, CMIP) to reflect the original requests made by the clients at the DCM interface.

3.4 Implementation approaches to agent-based management systems

In the scenario above the interactions between the client and the CP were discussed for contract negotiation, tailoring of offered management services and dynamically invoking specific management services. This section discusses the way in which the configuration, performance and fault management systems *within* the CP are organised.

In general, network management systems are hierarchical with a network-wide view at the top of the hierarchy and an element-specific view at the bottom with zero, one or more intermediate levels according to the needs of the system. This hierarchical approach can be seen in both TMN and TINA architectures. It is assumed that the configuration, fault and performance management systems in the agent-based CP will also follow a hierarchical architecture for many reasons including scalability and compatibility with existing management architectures and information models. There are two ways in which the CP's management systems could be deployed: either through building agent wrappers on existing management software - to represent the highest level of the TMN or TINA compliant system in the agent environment; or through building the entire management system from scratch in an agent based way and through building agent wrappers to represent the SNMP or CMIP interfaces of the network elements in the agent environment.

These two approaches are shown in Figure 3. Although the figure shows options for configuration management, the same holds for fault and performance management. Option A shows the approach of wrapping legacy systems with agents at the highest level; option B shows agent wrappers at the NE level and the entire system being built using agent technology. Option A retains existing management systems and therefore builds upon the operator's previous investment while option B allows a more agent-centric management system to be designed and built without being constrained by interworking with existing systems.

It is also possible that a hybrid approach (not shown in Figure 3) may be taken. In this case agent wrappers would be provided *at each level* of the hierarchy of a legacy management system. Mobile agents would then be able to visit the hierarchical level that is relevant to their operation and interact with fixed agents representing the legacy software at that level. Mobile agents in such a hybrid environment could relocate by traversing the hierarchy horizontally between subnetworks, or management functional areas, or they could migrate vertically to “zoom-in” or “zoom-out” the level of detail with which they are concerned.

The hierarchical nature of management systems in TMN and TINA (option A, Figure 3) is fixed at system design time and to a certain extent at standardisation time. For each management service to be deployed, the system designers make decisions on the placement of functionality at each hierarchical level and whether to distribute or centralise functionality within a particular hierarchical layer. These decisions are based on many factors, including: the degree of parallelism required; the quantity and complexity of information to be passed between components and whether existing information models need to be modified to support the required information flows; the scalability of the solution; balancing of processing load between management workstations; the complexity of each component.

A promising application of mobile agents for network management is in deploying each management component as a set of co-operating agents (Option B in Figure 3). The way in which these agents are grouped and placed is initially determined by the system designers according to similar criteria as those for the design of static hierarchical TMN or TINA systems. However, now it is possible to revise the grouping and placement decisions during the operation of the management system through the mobility of agents. There could be a number reasons for migrating agents on-the-fly or for spawning new agent instances: e.g. to reduce the processing load on an overloaded management workstation, to cater for an expanding set of managed resources or to reduce the quantity of management traffic, or information lag, between remote systems when it crosses a certain unacceptable threshold. This aspect of agent mobility for network management deserves to be studied further but is not the subject of this paper.

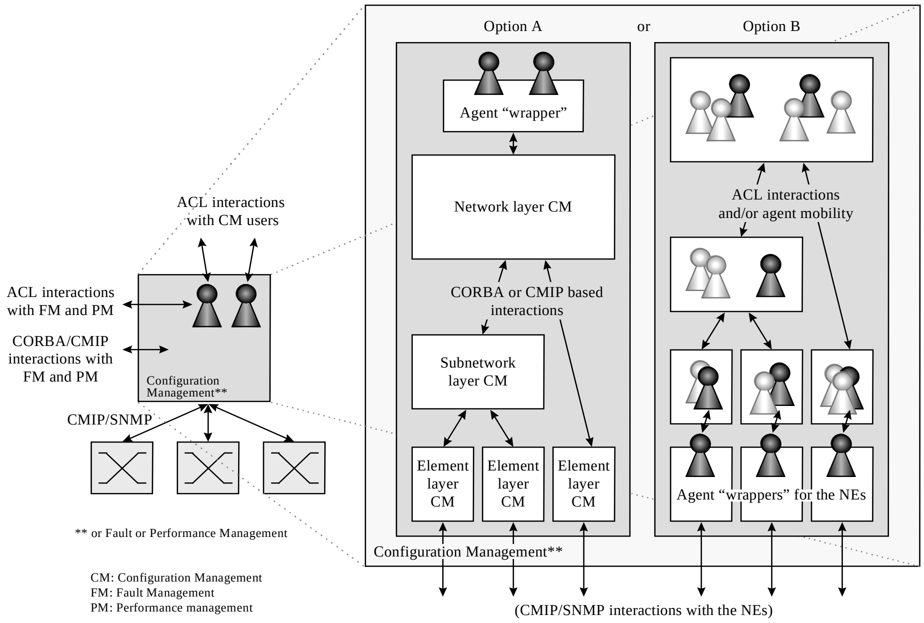

Figure 3 Implementation options for agent-based management functions

4 Customisation of management services

So far the basic operation of the CP's configuration, performance and fault management systems have been presented together with the means by which clients in the IP CP or AVPP may access them. This section presents the mechanism by which the management services may be customised and the following section presents three integrated scenarios to demonstrate the power of the mobile agent based approach to network management.

With the basic operation described in the previous section there is very little apparent advantage in adopting an agent based management system - very similar facilities are available in a traditional system based on distributed systems: TMN or TINA for example. However, there was one distinguishing advantage in the system as presented above, that was the way in which the DCM interface could be *customised* - with client-owned agents - to *tailor* the services offered to specific clients - this is one area where agents gain an advantage over traditional software systems.

In the scenarios below mobile agents are able to autonomously interact with one or more management functional area in the server to add value to the original management services. In traditional client-server[3] distributed systems for network

---

[3] The terms "client" and "server" are used rather than "manager" and "agent" to avoid confusing mobile or intelligent *agents* with OSI management or SNMP *agents*.

management (e.g. SNMP, TMN, TINA) the client is limited to working with the in-built facilities of the server. Any further manipulation of management information, beyond that which was generically provided by the relevant standards or by the developer of the server, must be performed in the client application code.

If customers wish to add value to the basic management services offered by the provider they must have the capability of running a management platform in their premises which supports the protocols and information models offered by the server. In addition, the customers must deploy suitable applications running on their local platform to house the required logic. The client applications must interact with the remote server to receive notifications and to initiate management operations. The quantity of information to be exchanged between client and server is a function of the management activities being undertaken and on the efficiency of the protocols and information models supported by the server for the task in hand. The delay and cost associated with each remote operation is a function of the network interconnecting the client and server. Given the particular protocols and information models supported by the server and the characteristics of the network interconnecting them, it may be not be cost effective or even possible to perform certain management tasks remotely in the client. For example, the cost of communication may outweigh the benefits of performing some management tasks (such as fine grain, real-time monitoring of performance parameters) or the information may be out of date by the time an appropriate course of action has been determined by the client application, if the network delays are too large.

With mobile agents it is possible for a client to program a mobile agent with specific functionality which may then be deployed *at run-time* in the server to add value above and beyond the server's basic facilities. Following deployment, the mobile agent may act autonomously to interact with the local environment and make *local* decisions which may then be implemented as *local* management actions without needing to interact with the remote client every time a decision is required. Through this approach it is possible for a remote client to deploy management behaviour and algorithms *inside* the remote server. Note that there are issues of server scalability and/or distribution to be considered further but which are not tackled directly in this paper.

5 Examples of customisable network management services

To illustrate the use of dynamically deployed agents we have identified the following three integrated scenarios which are being studied further in the MIAMI project:

- Intelligent reporting
  Mobile agents may respond to reports from both the performance and fault management systems. According to their programmed policies, rather than relaying *all* fault and performance reports back to the remote client, they will only report

when certain conditions have been fulfilled. An example might be a performance degradation on one connection following the failure of a connection in a remote part of the network which forms an alternate route. Only the correlation of these two events might be relevant to the client. Alternatively, observed performance degradations might cause the agent to initiate tests to verify that unreported failures have not taken place. This scenario integrates the facilities of fault and performance management.

- Fault repair

  A mobile agent is programmed to listen to fault reports from the fault management system when connections have been interrupted by network failures. According to a pre-programmed policy it can initiate new connection requests between the same end points as the failed connection to restore connectivity. This scenario integrates fault and configuration management.
- Bandwidth management

  Assuming that performance monitoring agents have been deployed (either by the client or the bandwidth management agent itself) to monitor and report on the utilisation of connections, a bandwidth management agent will be programmed to listen to utilisation reports for certain connections. Depending on the policy for a specific connection the bandwidth management agent may decide to request increased bandwidth on highly utilised connections or to reduce the capacity of a lightly utilised connection. The decision may depend on the cost of changing the bandwidth and so a negotiation between the configuration management agents and the bandwidth management agent may take place. This scenario integrates performance and configuration management.

The integrated scenarios introduced above combine the basic facilities of the configuration, fault and performance management services offered by the server with additional customer specific logic. In other words the customer is able to program the offered management service to a certain degree. This concept has its parallel in traditional management systems through the use of the OSI management Systems Management Functions (SMFs) [19] for event forwarding and logging, resource monitoring [20] [21], and testing [22], albeit in a more limited way. Previous research work [23] [24] has demonstrated how clients can take advantage of these generic facilities to simplify the construction of intelligent clients.

We now consider how these basic facilities could be implemented through the use of mobile agents. Rather than being restricted to standardised capabilities such as the SMFs it is now possible to build entirely arbitrary and powerful behaviour into mobile agents which will be physically located in the managed system’s environment. This embedded intelligence not only allows event reports tailored to the client's requirements to be emitted but it enables the migration of the client's logic and decision making algorithms to the server. This has an obvious impact on reducing the quantity of management traffic between remote systems and achieves more timely access to information generated by the remote server.

Considering the network example in Figure 1 a number of ATM connections (VPs) are required between a limited set of endpoints to interconnect the IP routers in the IP CP domain to support higher quality VE traffic. The degree of interconnection and the characteristics of each connection are subject to change throughout the lifetime of the VE. To achieve this a fixed contract for specific connections is not appropriate. The contract between the ATM CP and its customer should be flexible to allow the resources to be managed in a dynamic way: e.g. to allow the creation, deletion and modification of connections and to allow them to be monitored and tested.

The initial contract negotiation over the CN interface results in the creation of a DCM interface and agency where semi-permanent VPs may be established and subsequently managed. The contract should specify the allowed connection termination points and the set of basic management services to be offered across the DCM interface. The contract may limit the number of simultaneous VPs that may exist at any one time and determine the costs associated with resources and management operations.

In our MIAMI work we have used UML for object specification with mappings to the Java language in the context of the Grasshopper platform for implementation. However, in the following example we are using a pseudo-procedural style to represent interfaces and logic.

Example service contract:

```
List of termination points: a, b, c ... n

Management services:

      request_trail(source, destination, capacity, qos parameters)
      delete_trail(trail_id)
      modify_trail_capacity(trail_id, new_capacity)
      monitor_trail(trail_id, performance_parameter, polling_interval,
      thresholds)
      modify_monitoring_characteristics(trail_id, performance_parameter,
      polling_interval, thresholds)
      test_trail(trail_id)

Costs: ...
```

Given that these basic management services are available, the customer could create a mobile agent to capture his logic and execute in the remote domain. The following example uses the configuration and performance management services to dynamically manage the capacity of a trail:

```
monitor_trail(trail_1, utilisation, 10 seconds, {upper threshold: 85%,
lower_threshold: 45%})
during_the_hours_9am_to_5pm:
{
        if (upper threshold crossing on trail_1)
        if (old_capacity+20% > some_limit) then raise_notification_to_user()
        else modify_trail_capacity(trail_1, old_capacity+20%)
        if (lower threshold crossing on trail_1) then
        modify_trail_capacity(trail_1, old_capacity-10%)
}
during_the_hours_5pm_to_9am:
{
        if (upper threshold crossing on trail_1)
        if (old_capacity+10% > some_limit) then raise_notification_to_user()
        else modify_trail_capacity(trail_1, old_capacity+10%)
        if (lower threshold crossing on trail_1) then
        modify_trail_capacity(trail_1, old_capacity-50%)
}
```

During working hours the customer wishes the capacity of his trail to increase if his users consume more than 85% of the trail's capacity unless this would push the capacity above a pre-defined limit which might cause the cost of the trail to be greater than he is prepared to pay, even if this causes his users to experience reduced quality. In this case a notification is raised to either the customer or to other agents in the DCM.

In addition to the logic in the above example the agent could be programmed to offer an interface to other agents or the remote customer to set and modify some of the parameters. For example the customer could dynamically change the value of the utilisation thresholds and the percentage increase/decrease to be applied in the case of threshold crossings.

In OSI management, SMFs were standardised by international organisations and encapsulated in the compiled functions of OSI agents; in CORBA-based management systems the SMF-like facilities could be determined by the designers of the systems and embedded at design and system-build time; with mobile agents and intelligent reporting in agent-based management systems the SMF-like facilities can be enhanced and extended almost infinitely and deployed *at run time*!

As seen in the example above, it would be very difficult to capture such behaviour in traditional TMN or TINA systems without standardising such a bandwidth management service at the Xuser or ConS interface. If such a service was to be standardised it would be difficult to capture all possible potential behaviours that clients may request without making a comprehensive and therefore complex specification of the service in GDMO or IDL. However, through the use of programmable, intelligent agents based on mobile code for dynamic and customisable network management this is achievable and deployable on the fly and at the whim of the client. This is clearly a very powerful application of mobile agents for telecommunications management.

6 Implementation issues

The initial approach for the configuration management domain was to base it on an existing TMN system for ATM PVP set-up originating from the ACTS MISA project [25], to which an agent interface would be added (option A in Figure 3). For logistical reasons, the final implementation is based on *static* agents internally (Figure 3, option B), which communicate using remote method calls. This implementation could be also based on distributed object technology e.g. CORBA. Agents were chosen for two reasons: first for uniformity, since there is no need for an adaptation agent-based interface; and second for evaluating mobile agent platforms in the same role as distributed object frameworks. It should be finally noted that we do not see any immediate benefits from applying mobile agent technology to configuration management.

On the other hand, the performance and fault management systems use agents internally (Figure 3, option B) in a way that mobility is exercised and exploited. In the performance management domain, customised agents replace, augment and allow customisation of the functionality of the TMN/OSI-SM metric monitoring and summarisation objects [20][21], while in the fault management domain customised agents replace, augment and allow customisation of the TMN/OSI-SM testing objects [22]. In both domains, mobile agents are instantiated at the “network management level” of a management hierarchy according to requests originating from the DCM domain, migrate to network elements and perform relevant tasks locally, enjoying minimal latency and reducing network traffic. Details of the performance and fault management approaches are described in [26] and [27] respectively.

The DCM domain uses agent mobility in a similar fashion to the performance and fault management domains, supporting the programmability of connectivity services by clients across the DCM interface as described in section 5.

At the time of writing, this system is in the final integration and testing stage of the various components with a field trial planned for December 1999. On the other hand, some experimentation with individual domains has already taken place, with results reported in [27] for the fault and in [26] for the performance management domain. These results point to a performance and resource overhead for mobile agent platforms in comparison to distributed object platforms, which partly outweighs the programmability advantages. On the other hand, there is ongoing research work to integrate distributed object and mobile agent platforms. Such an integration would allow the performance benefits of distributed object platforms for static objects/agents, with the additional flexibility of mobility for mobile agents/objects.

7 Summary and Conclusions

Traditional and emerging frameworks for network management such as TMN and TINA allow customers electronic access to management services. These services, however, are fixed in the sense that new features can only be added after a lengthy research-standardisation-deployment cycle. In this paper we have discussed the advent of mobile agent technologies and how they may enhance traditional connectivity management services making them dynamically customisable by clients. We presented three examples which add value to the offered management services as perceived by and required by the *client* rather than by the researchers, standardisation bodies, equipment vendors or service providers. Clients may introduce their own value-added logic during service operation to cater for the dynamics of their environment and to enforce their own policies.

This prompts for a new paradigm for building network management services. Instead of providers building services that attempt to encapsulate the requirements of all clients, they build the necessary hooks and let the clients apply their logic. Customisation and programmability of management services was always possible in traditional systems based on client-server paradigms through the development of client applications in the customers' premises management platform to capture the required logic and intelligence. However, the cost and efficiency associated with such remote operations compared to the proposed agent-based approach should be considered.

The advantages and disadvantages of agent based methods for tailoring, customising and programming management services compared to more traditional client-server management approaches are currently being evaluated through prototype development and subsequent experimentation. This work is being undertaken in the context of the ACTS MIAMI [12] project.

In summary, agent mobility in the presented network management architecture is used in a fashion which we would term “constrained mobility”: mobile agents are instantiated at a control point by a master static agent and then move to another point (i.e. network node) where they stay until their task is accomplished, this can be considered as an intelligent software deployment activity. The key benefit of this approach is *programmability*, allowing clients to “push” functionality to a point offering elementary hooks which can be accessed to provide derived, higher-level services. In a similar fashion, we could term “full mobility” as a situation in which a mobile agent moves from point to point using its built-in logic, adapting to changing situations in the problem domain where it is involved. We have not yet found convincing cases in our network management research where full mobility could offer tangible benefits.

In summary, we believe that agent-based, customisable, programmable services provide a new range of opportunities to service providers. The three examples presented are just the tip of the iceberg, an enormous number of potential services are possible. By combining the concepts of intelligent reporting agents with fault repair agents and bandwidth management agents with other possibilities, it is certainly possible to deploy significant, complex, “active” management applications in a dynamic and flexible manner. This would move away from the client-server, manager-agent paradigms of today’s systems to a fully dynamic system of interacting agents which can be built layer-by-layer with increasingly complex behaviour to fulfil the demands of sophisticated clients.

Acknowledgements

This paper describes work undertaken in the context of the ACTS MIAMI (AC338) project. The ACTS programme is partially funded by the Commission of the European Union.

References

[1] ITU-T Rec. M.3010, Principles for a Telecommunications Management Network (TMN*)*, Study Group IV,1996.

[2] TINA consortium, Overall Concepts and Principles of TINA, Document label TB_MDC.018_1.0_94, TINA-C, February 1995.

[3] ITU-T Rec. X.701, Information Technology - Open Systems Interconnection, Systems Management Overview, 1992.

[4] Object Management Group, The Common Object Request Broker: Architecture and Specification (CORBA), Version 2.0, 1995.
[5] TINA Business Model and Reference Points, version 4.0, TINA Consortium, May 1997.
[6] Foundation for Intelligent Physical Agents, web page: http://www.fipa.org/
[7] Mobile Agent System Interoperability Facilities Specification OMG TC Document orbos/97-10-05, November 10, 1997. ftp://ftp.omg.org/pub/docs/orbos/97-10-05.pdf
[8] Breugst, M., Magedanz, T., Mobile Agents - Enabling Technology for Active Intelligent Network Implementation, IEEE Network, Vol. 12, No. 3, May/June 1998.
[9] Bieszczad, A., Pagurek, B., White, T., Mobile Agents for Network Management, IEEE Communications Surveys, Vol. 1, No. 1, 4th Quarter 1999, http://www.comsoc.org/pubs/surveys
[10] M. Zapf, K. Herrmann, K. Geihs, Decentralised SNMP Management with Mobile Agents, in Integrated Network Management VI, Sloman, Mazumdar, Lupu, eds., pp. 623-635, IEEE, 1999.
[11] Gurer, D., Lakshminarayan, V., Sastry, A., An Intelligent Agent Based Architecture for the Management of Heterogeneous Networks, in Proc. of the 9th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, Sethi, ed., Univ. of Delaware, USA, October 1998.
[12] Web pages: main project page: http://www.fokus.gmd.de/research/cc/ima/miami/
page at UCL: http://www.ee.ucl.ac.uk/~dgriffin/miami/
page at UniS: http://www.ee.surrey.ac.uk/CCSR/ACTS/Miami/
[13] Covaci, S., Zell, M., Broos, R., A Mobile Intelligent Agent Based Virtual Enterprise Support Environment, in Intelligence in Services and Networks - Paving the Way for an Open Service Market, Zuidweg, Campolargo, Delgado, Mullery, eds., pp. 536-549, Springer, 1999.
[14] Vigna, G., (ed.), Mobile Agents and Security, LNCS 1419, Springer-Verlag, 1998.
[15] Labrou, Y., and Finin, T., A semantics approach for KQML -- a general purpose communication language for software agents , Third International Conference on Information and Knowledge Management (CIKM'94), November 1994.
[16] Agent Communication Language, FIPA 97 Specification, Version 2.0, October 1998, http://www.fipa.org/spec/FIPA97.html
[17] Booch, G., Rumbaugh, J., Jacobson, I., The Unified Modelling Language User Guide, Addison-Wesley, 1999.
[18] http://www.ikv.de/products/grasshopper/index.html
[19] ITU-T Recommendations X.730-750, Information Technology - Open Systems Interconnection - Systems Management Functions.
[20] ITU-T X.738, Information Technology - Open Systems Interconnection - Systems Management: Metric Objects and Attributes, 1994.
[21] ITU-T X.739, Information Technology - Open Systems Interconnection - Systems Management: Summarisation Function, 1994.
[22] ITU-T X.745, Information Technology - Open Systems Interconnection - Systems Management: Test management function, 1994.
[23] Georgatsos, P., Griffin, D., Management Services for Performance Verification in Broadband Multi-Service Networks, in - Bringing Telecommunication Services to the People - IS&N'95, Clarke, Campolargo and Karatzas, eds., pp. 275-289, Springer, 1995.
[24] Pavlou, G., Mykoniatis, G., Sanchez, J., Distributed Intelligent Monitoring and Reporting Facilities, IEE Distributed Systems Engineering Journal, Vol. 3, No. 2, pp. 124-135, IOP Publishing, 1996.

[25] Karayannis, F., Berdekas, K., Diaz, R., Serrat, J., A Telecommunication Operators Inter-domain Interface Enabling Multi-domain, Multi-technology Network Management, Interoperable Communication Networks Journal, Vol. 2, No. 1, pp. 1-10, Baltzer Science Publishers, March 1999.

[26] Bohoris, C., Pavlou, G., Cruickshank, H., *Using Mobile Agents for Network Performance Management*, to appear in the Proc. of the IFIP/IEEE Network Operations and Management Symposium (NOMS'00), Hawaii, USA, IEEE, April 2000.

[27] Sugauchi, K., Miyazaki, S., Covaci, S., Zhang, T., Efficiency Evaluation of a Mobile Agent Based Network Management System, in Intelligence in Services and Networks - Paving the Way for an Open Service Market, Zuidweg, Campolargo, Delgado, Mullery, eds., pp. 527-535, Springer, 1999.