A COMPARISON OF APPROACHES TO MULTI DOMAIN CONNECTION MANAGEMENT

Alex Galis
University College London
United Kingdom; E-mail: a.galis@eleceng.ucl.ac.uk

David Griffin
University College London
United Kingdom; E-mail: d.griffin@eleceng.ucl.ac.uk

ABSTRACT

A number of models exist for connection establishment, ranging from traditional control plane solutions based on signalling procedures to connection management architectures based on distributed computing technologies. This paper concentrates on those approaches relevant to the establishment of semi-permanent connections through the activity of the management plane; concentrating, in particular, on the issues related to inter-domain connections requiring federation between connection management systems.

Three approaches are analysed and compared in this paper: the TMN approach, based on the X-user and X-coop interfaces, using OSI systems management as the underlying technology; the TINA approach, using the ConS and LNFed reference points, with CORBA as the base technology for the computing architecture; and an approach based on intelligent agents and mobile code using OMG's MAF as the enabling technology. The three approaches are compared from the points of view of architecture and technology considering the issues of system design, migration and interworking.

Several possibilities for the future integration and/or interworking between systems based on the three approaches are proposed, including: enhancements to TMN and TINA architectures to make them agent-friendly; the use of agent technologies to implement TMN or TINA compliant building blocks; the use of agents to enable interworking between diverse technologies and architectures. The outstanding issues are raised which highlight the areas of future work which are expected to be carried out by a number of research projects which have recently been awarded.

KEYWORDS:

AGENTS, INTERWORKING, MULTI DOMAIN, CONNECTION MANAGEMENT, TINA, TMN.

1.Introduction - Connectivity Management Services

The management of broadband multipoint, multiple bit rate connections has been found to be very complex, especially in a multi-provider, multi-domain environment. Typically, today, the setting up and re-configuration of such connections are performed through manual actions using faxes and telephone calls.

Currently the management of end-to-end services across different telecom operators is largely non-automated. Several large multinational companies in Europe request such end-to-end services (e.g. international voice VPN services). If this trend expands to include more such customers or new services, it will not only become necessary to automate provision of the underlying physical networks, but also to automate their maintenance to provide guaranteed quality of service for end-to-end connectivity.

One of the main TMN goals is to automate the configuration procedures so as to satisfy the end-user requirements in a matter of seconds. This will be achieved by provisioning of open interfaces to the management centres for the necessary co-operation between management systems. This goal is vital to continue the development of the Integrated Broadband Communication (IBC) Infrastructure.

Once the configuration procedures are automated the TMN services will allow ATM/SDH network resources to be used in a co-operative and efficient manner, addressing the needs of the following types of actors:
• Customers and value-added service providers requiring basic broadband connectivity services.
• Network providers with requirements to negotiate co-operative services with their peers;

Providers needing to make optimal use of their resources and allow the timely identification and notification of changes in quality of service.

A number of models exist for connection establishment, ranging from traditional control plane solutions based on signalling procedures to connection management architectures based on distributed computing technologies. This paper concentrates on those approaches relevant to the establishment of semi-permanent connections through the activity of the management plane; concentrating, in particular, on the issues related to inter-domain connections requiring federation between connection management systems.

The ITU-T distinguished between the management control and user planes in the operation of communications networks [17] [18], however it is sometimes ambiguous whether a particular component belongs to the control or management plane as there are several ways to define the contents of the control and management planes.

The following is the management functionality necessary for establishment of multi domain connections at the Network Management level, which is in addition to the intra-domain connection establishment management functionality:

- The Consumer establishment function: The manager sends an order to the destination consumer domain to determine whether the consumer accepts proposed connection.
- The network topology update function: The manager fetches information from a given domain concerning the topology information. The appropriate information is given to the manager.
- Connection reservation function: The manager requests the intra-domain connection a reservation of a subnetwork connection in a domain.
- Connection activation function: The manager orders the activation of a connection, which is acceptable to the consumer.
- Maintenance of reservation function: The manager orders the changes to the characteristics of an active connection.
- Maintenance of connection activation function: The manager orders the modification or cancellation of an accepted connection.
- Connection release function: The manager requests the release the resources allocated to a connection.
- Connection deactivation function: the manager sends the appropriate request for the deactivation of a connection.

Three approaches to multi domain connection management are analysed and compared in this paper: the TMN approach, based on the X-user and X-coop interfaces, using OSI systems management as the underlying technology; the TINA approach, using the cons and LNFed reference points, with CORBA as the base technology for the computing architecture; and an approach based on intelligent agents and mobile code using OMG’s MAF as the enabling technology. The three approaches are compared from the points of view of architecture and technology considering the issues of system design, migration and interworking.

2. TINA Connection Management Architecture

The TINA (Telecommunications Information Networking Architecture) [32], [39], [40], [41], [42] initiative aims at providing a framework for all telecommunications software, encompassing components ranging from connection establishment through network and service management to service delivery and operation. TINA strongly supports the concepts of ODP (Open Distributed Processing) [34] via its modelling techniques and viewpoints, and the use of a DPE (Distributed Processing Environment) for providing the generic facilities required by all software running in a distributed fashion.
TINA specifies a ubiquitous software platform for service logic, covering both service operation and service delivery. The idea is that instead of being limited by the IN architecture, the telephony call model and signalling protocols, new advanced services may be deployed directly on a Distributed Processing Environment (DPE) and may be designed and implemented according to object-oriented principles and distributed processing techniques.

TINA's Management Architecture covers the principles and concepts for managing TINA systems and networks and draws heavily on the ITU's TMN architecture [25]. The TINA specifications in the configuration management area are the most developed, especially those for connection management [42], [2], [12], [14], [32], [33]. An interesting observation of the TINA results in this area is that they do not distinguish between the control and management planes in the same way that traditional telecommunications architectures do. Because of this, connection management is included in the Management Architecture as part of configuration management, rather than being part of the control plane of the Network Architecture supported by signalling mechanisms.

The overall business model of TINA is based on a set of actors in different business roles and reference points (RPs) identifying the interactions between those actors. The interactions across the ConS reference point between Retailers and Connectivity Providers (CPs) and across the LNFed reference point between peer CPs is of particular importance when considering inter-domain connection establishment. The Retailer, on behalf of the Consumer makes the request for inter-domain connections across the ConS RP to a single CP who is responsible for the end-to-end connection. Through interactions across the LNFed RP a set of tandem connections (that portion of an end-to-end connection residing within a single CP domain) are established in each CP domain to collectively form the end-to-end trail which forms the inter-domain connection in the TINA environment.

Within a CP domain the Connection Management Architecture (Figure 1) is responsible for establishing the trail. Once the Connection Co-ordinator has determined the most suitable layer network (ATM VP, VC, SDH, etc.) for supporting the requested connection the Layer Network Co-ordinator (LNC) responsible for that layer network is invoked. Through peer-to-peer transactions via the LNFed RP the LNC requests that tandem connections are established in each domain. The tandem connections within a domain are created via the hierarchy of Connection Performers (CPs) each CP responsible for a subnetwork which is an abstraction of a portion of the network within that domain which in turn may be composed of further subnetworks or network elements themselves. In this way connections are established in a top-down fashion through the cascading of operations down the hierarchy of CPs in the CM architecture.

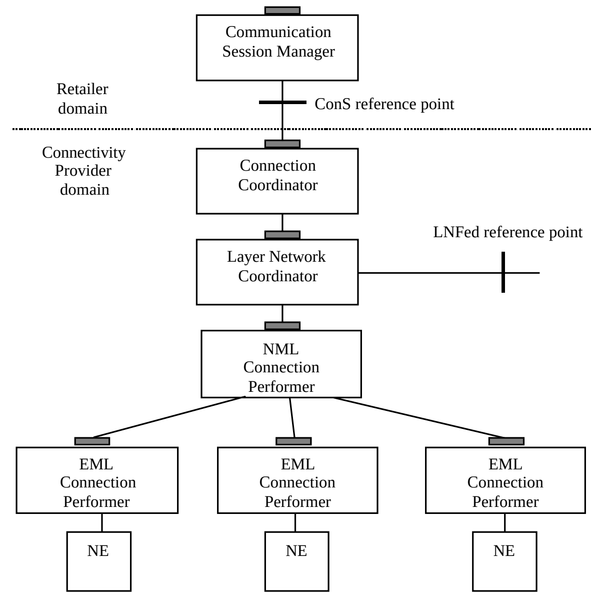

Figure 1 - The TINA Connection Management architecture

The REFORM AC208 project [13] has followed this TINA implementation architecture for multi-domain Connection Management functionality.

3.TMN Connection Management Architecture

The TMN Connection Management Implementation ([30], [31]) approach was followed by the MISA ACTS AC080 Project [24], [13] where a Global Broadband Connectivity Management Service was developed.

Based on the functionality of the Multi-domain federated Connection Management described above MISA project has developed an ATM/SDH network independent Xcoop ensemble for path - provisioning. The Xcoop is the interface between TMN OS of different domains. By means of this interface, the originating TMN OS establishes, modifies, reconfigures and releases CM connections. Each of these management services is supported on component functions.

In MISA the Xcoop interface is being implemented on the base of the manager-agent paradigm. The manager and the agent send and receive CMIS primitives to carry out the tasks specified in each component function. On the other hand external modules to the Xcoop will trigger these functions.

The configuration of the MISA managed network envisaged for the MISA trials in 1998. In this trial the demonstration components are communicating across the Internet with generic replicable middleware servers, who in turn interface via CMIP/X.25 and CMIP/RFC1006 to the location and potentially platform-specific Operations Systems representing the ATM/SDH managed networks in the National Host environments

The middleware developed in the project, Webbin, is a gateway that converts http requests to CMIP requests and it can be used for communication with the designed agents/managers. The Webbin CMIP and Java [11] language/environment are used for the integration of the demo components with the graphical user interfaces. The user requests to this server are converted to CMIP requests by the Webbin CMIP gateway resulting in management information flow.

4. Agent Connection Management Architecture

Recently the popularity of the WWW and JAVA has created a new momentum towards mobile agent applications [38], [46], [4], [5], [43], [16], [26], [27]. Mobile agents are intelligent programs that can migrate on computer networks. The concept of having mobile agents carrying out tasks for their owners is creating a new paradigm for network-enabling computing. Mobile agents not only enable more efficient utilisation of network resources, but also provide o more scaleable model for implementing distributed applications on wide-area networks.

Among the more compelling advantages that mobile agents can provide over client-server computing are network traffic reduction, reduction in connectivity requirements, and enabling control of remote operations with real - time constraints. To fully exploit the advantages that mobile agents provide, they must be design in the context of application domain. A mobile agent should utilise knowledge about the problem that it is addressing.

In order to realise a mobile agents based Connection Management a number of agents have to be created with the functionality identify in the introduction and an engine to run the agents. The engine for the CM agents could be constructed using JAVA language or based on the JAVA. A number of Agents Platform are based on JAVA: Telescript and Odyssey from General Magic, Aglets Workbench [1] from IBM, Voyager from ObjectSpace, and Grasshopper from IKV++. The main reason for the selection of the defacto standard JAVA based engine for the CM agent implementation is two fold: firstly in a case that a TMN environment has no mobile agent engine, it is easy to construct its environment by downloading a mobile engine composed of JAVA language based agents which interoperate with CMIP/SNMP resources through middleware like Webbin [24]. The WEBBIN is a platform-independent plug-in for Web servers, which aims to simplify the way network management is performed. Webbin is based on the idea that the complexity of protocols such as CMIP or SNMP has to be hidden by the system.

A mobile agent operation environment for Connection Management is represented in the following figure in line with the MAF [29] specification.

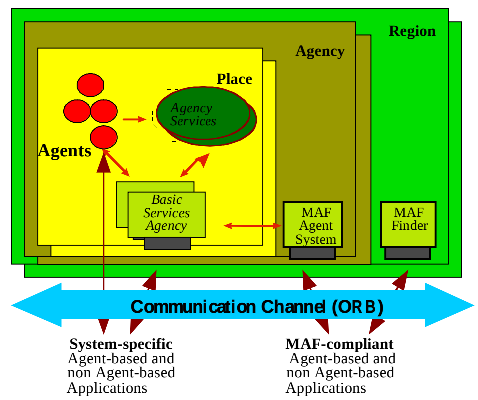

Figure 2 - MAF Compliant Mobile Agents Architecture

In order to realise a mobile agents based Connection Management a number of agents have to be created with the functionality identify in the introduction. They are as follows.

CM Agency : An agency is placed at each management domain. The agency is a place where fixed agents exist and mobile agents are received processed and dispatched to the next management domain. Issuing requests to a Factory creates agents. The basic structure for an inter - agency communication is an agreement. The agreement is composed o a general component and a dynamic component. The general component consists of a set of network resource descriptors such as network topology information and domain specific connection establishment resources. The dynamic component showed the current values for the descriptions found in the general component. Agents interact by processing agreements and agreements travel from agency to agency inside mobile agents.

Mobile CM Agents

Achieving an efficient transaction for multi domain connection management negotiation and management is difficult because of the number and complexity of the entities involved in the process. With mobile agents, it is possible to simplify the transaction by partitioning many functionality into autonomous entities (i.e. agents), keeping simples the communication among such entities. Adding mobility to these entities is a step further in simplification by balancing between message exchange and code dispatching.

There are three classes of mobile agents, which are instantiated when some specific activities need to be carried out. These activities encompass connection establishment, monitoring and re-negotiation.

- Connection Negotiators. These agents operate during the establishment phase and negotiate the agreement on behalf of an consumer / initiator of connections in other words, the consumer delegates the responsibility of agreement negotiations to these agents. Connection negotiators carry the agreement and travel from agency to agency where the resources states in the agreement are managed. The negotiation is conducted in two phases: reservation and commitment. In the first phase the reserved resources are committed to the initiator of the connection. The agent leaves a copy of the negotiated agreement in the agencies in involved in the negotiation. If the agent is not able to reserve or commit a resource it simply travels again releasing all the resources already reserved or committed. In the commitment phase the agent optimises the network resources. By imposing time-outs or committing the already reserved resources a simple deadlock avoidance algorithm is implemented in the resource managers. Connection Negotiators are instantiated when connection establishment session starts and are destroyed after the agreement has been negotiated.
- Connection Monitors. These agents re instantiated after the Connection Negotiator establishes the agreement and are kept activated during all the management phase of a connection. Connection Monitors travel from time to time in order to check if the parameters stated in the agreement are being honoured. In order to accomplish this task, the agent of this class travels to the agencies where the agreement has acquired resources and inspects the parameters stated in the agreement The connection monitor interacts with the local CM Monitors in order to get a picture of the optimum use of the network resources. When this agent comes back, a global picture of the parameters in the agreement is evaluated, and the results of this evaluation are presented to the consumer of the connections employed in the setting-up of the connections.
- Connection Re-negotiators. These agents travel to the agencies where resources must be renegotiated. The agent is autonomous since it carries all the information it needs for agreement re-negotiation. Agreement termination is a special case of re-negotiation and is performed by this class of agents.

Fixed CM Agents

- The CM Mapper Agent is responsible to map multi domain network wide connection establishment parameters into domain specific parameters. The mapping can be based on simple translation techniques such as tables, or sophisticated, knowledge –based schemes such as those based on expert systems.
- The CM Resource Agent has the mission of mapping between the multi – domain network resources parameters and local domain resources. Like the CM Mapper, this agent ca employs a simple or a sophisticated mapping scheme.
- The CM Resource Manager Agent is responsible for negotiating the part of the agreement that involves local resources. This agent receives the resources needed from the Resource agent and interacts with the Resource Manager agent for resource reservation and allocation.
- The CM Monitor Agent checks permanently if the if the connection parameters stated in the agreement are within their bounds. It interacts with the low-level resources management in order to measure the level of the information generated by the network.
- The CM Adaptor Agent is activated when some parameter in the agreement goes out of bounds. If the parameter is managed locally it interacts with the Resource Manager in order to try to correct the problem locally. Otherwise, if the parameter relates to a remote resource, this agent instructs the CM Mapper Agent to start the agreement re-negotiation activity.

Except for the Resource Manager, the agents listed above are instantiated when a connection establishment session starts and lasts until the end of the session.

5. Co-existence Scenarios

5.1 Possible approaches of TMN-TINA CM co-existence

Different scenarios of TMN CM systems interconnection with TINA architecture based CM Systems were described in [37], [14], [22], [35], [36].

The exchange of information between OSI/TMN-based CM components and CM TINA-based components may involve two different semantics :

In the top-down relation, a TMN-based component offers a specific, application-oriented access to a TINA component. This topology makes interoperation possible through a particular (ad hoc) interface for a specific application. The top-down relationship between TINA and OSI components represents a realistic interconnection TMN - TINA CM scenario, because it inherently preserves existing OSI installations and allows to provide TINA service layer applications to the customer at the cost of a mediation device (CMIP-CORBA gateway). In the top-down interaction, the TMN-based components and the TINA-based ones behave exactly in the same manner, with respect to the DPE. In this case, all applications have access to the managed objects disregarding whether they reside in TMN agents or in TINA components.

The peer-to-peer relation projects mixed TMN systems bearing both ISO components and TINA components on the same level of responsibility regarding resource management. The DPE insures communication between all components. A mediation couples the DPE with the OSI components. Depending on the tightness of the relation between TMN and TINA components, two interaction types can be distinguished : loose and tight coupling. In the loose coupling interaction, the two types of components reside in distinct protocol domains, like ´ pure ‘ entities (they communicate each in their own protocol). The interoperation requires therefore an additional component, the mediation. Loose coupling enables out-of-the-box or existing applications to communicate easily without any modification. This allows to envision CORBA interoperability in the context of deployed TMN software at a minimum redesign cost. The loose coupling interaction attempts to keep either interoperating system completely unaffected by the respective other protocol, concentrating all the translation mechanisms into the mediation component. Loose coupling enables to reuse existing OSI components directly.

At the opposite, in the tight coupling interaction some bi-domain entities are able to communicate both through an ORB and through CMIP. In a sense, these entities encapsulate their own mediation. In this situation, a CORBA manager is able to communicate directly with a CMIP agent. Tight coupling, when combined with the ability to recompile/link existing application code, makes existing applications access transparently new CORBA entities. Indeed, as this layout proposes really an extension of the communication classes of the TMN framework towards CORBA, tight coupling brings the advantage of reusability of existing TMN software. Tight coupling introduces a new kind of components not any more bound to one distribution technology. More precisely, the ability to communicate with respects to these protocols is a property of the object instances themselves, instead of a centralised scheme where some dedicated service would provide communication capabilities. Implementations make use of the inheritance and polymorphism properties of OO languages in order to confer to the objects the new behaviour in a transparent manner.

5.2 Possible approaches of TMN/TINA and Agents CM co-existence

Many mobile-agent systems are closed systems (i.e. Telescript). They prescribe a given language foe agent development and use proprietary protocols for agent mobility and interaction. An open approach to agent interaction and movement is required and envisaged for CM applications through the use of CORBA and /or Hypertext Transport Protocol (HTTP) [3] , an Internet Standard [2].

The Common Object Request Broker Architecture has been designed by the Object Management Group (OMG) with the goal to provide a truly open object distribution. The CORBA architecture is composed of a software bus, the Object Request Broker (ORB), on which sit CORBA clients (referencing distant objects) and CORBA servers (holding objects). The clients and servers share one unique definition of their interface, which is specified in a particular language: the Interface Definition Language (IDL) [19]. IDL is independent from implementation languages, thus the developer has a choice of IDL bindings (to C++, etc.) from which to select.

In addition, the CORBA architecture offers a number of services, which are, from bottom to top:

- core object services which are generic: naming, persistence, events etc. which can be thought of as augmenting the functionality of the ORB;
- Common Facilities, which define application-level frameworks: Information Management, Task Management etc.
- Business Objects, which are at application level and perform business-specific functions.

The Common Object Request Broker Architecture (CORBA) has been established as an important standard, enhancing the original RPC based architectures by allowing relatively free and transparent distribution of service functionality.

The MAF standard [29] was developed to handle interoperability between different agent systems via CORBA thus enabling interoperability of agents based CM components to interact with TINA applications or TMN CORBA based applications. This standard is complementary to the FIPA standard [9], [10].

This approach of TINA applications interaction with agent based applications prescribes the need to have the agent platform provide additional middleware for a common agent CORBA API as depict in the following figure.

The runtime environment needed by CORBA -based applications is relatively heavy-weight, whereas HTTP is, by comparison, a lightweight protocol that every agent can easily carry its own protocol implementation.

The agents can interact between themselves by issuing HTTP request and it follows that agents can interact and interoperate with CMIP/SNMP resources through middleware like Webbin [24]. Every URL corresponds to a remote method invocation; arguments can be wrapped up in the URL, and results are returned just like WWW pages. In this way agents have the option to declare whether they want to move to another agent, or whether it is faster to remotely invoke its methods. This decision is a trade -off between the cost of agent migration and the amount of data to be exchange between TMN objects and agent. The advantage of this approach is that it facilitates the integration of mobile agents with stationary agents by using the interaction mechanism native to the stationary agent's environment.

5.3 Co - existence Scenarios

When examining the different scenarios of TMN systems interworking with the TINA and Agents systems, different co-existence options come to mind. These perspectives, or migration paths, differ regarding the relative topology of OSI , TINA and Agents systems one to another on one hand, and regarding the way in which information is really exchanged between both worlds on the other.

The following figure summarises the co-existence scenarios. The architecture of the interworking units is for further study.

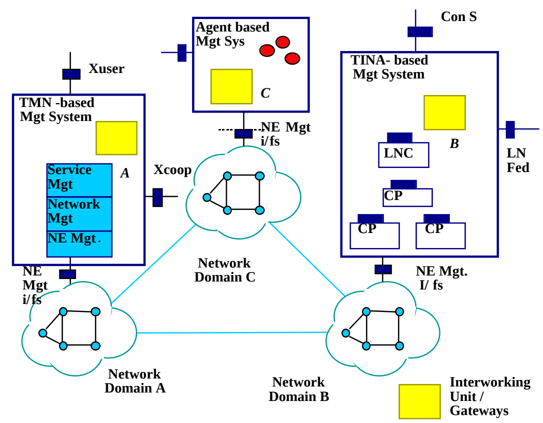

Figure 3 – Co-existence Scenario

The interworking units in the above figure must interwork in all possible combinations.
The interworking issues involve consistency of information models between management systems based on different technologies and the compatibility of operations across system boundaries.

For interworking from an agent-based environment (operating in a client role) to either a TMN or TINA management environment (operating in a server role) the style of interworking unit depicted in the following figure is envisioned.

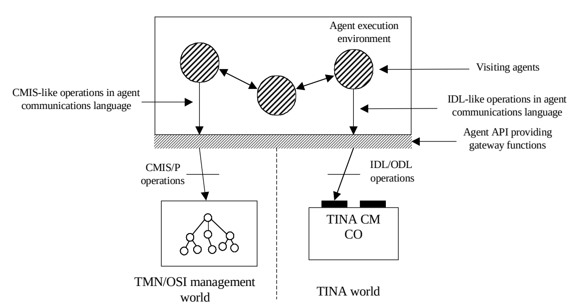

Figure 4 – Interworking Units

There are two ways in which this can be achieved. Either the agents physically move from the agent-based management domain to the remote (TMN or TINA) domains and execute in the interworking units provided by the remote domains; or the agents execute locally with a local agent API and interworking unit issuing CMIP or IDL operations "on the wire" between domains. In the latter case the X interface or LNFed reference points are preserved between management domains, in terms of interface specification, syntax and semantics of operations and communications protocols used. In the former case the semantics of the X or LNFed operations are embedded within the mobile agent migrating to the remote domain and interacting with the agent API in the remote domain's agent execution environment where a local X interface (or LNFed reference point) is supported between the agent interworking unit and the local management system. Both options are shown in the following figure in the case of agent interworking with TMN-based management domains. A very similar figure could be drawn to illustrate interworking with TINA-based management systems.

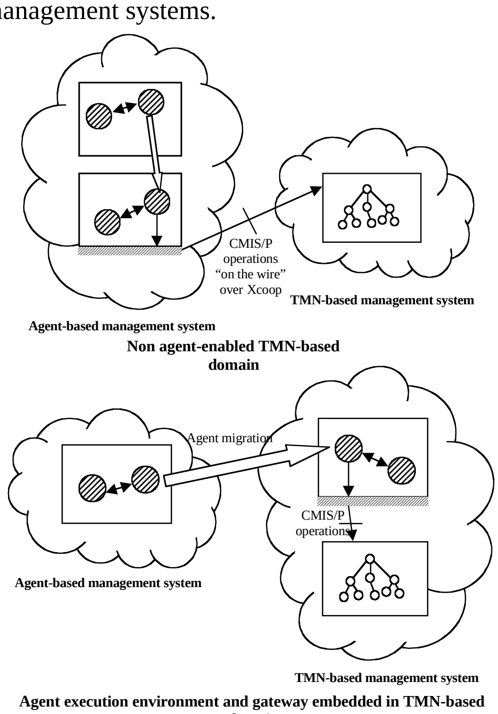

Figure 5 – Interworking with TINA based management system

In order to allow interworking between agents and TMN or TINA based systems in the reverse direction (the agents-based domain is operating as a server to client TMN or TINA-based domains) a slightly different approach is required for the gateway/interworking unit. This is shown in the following figure.

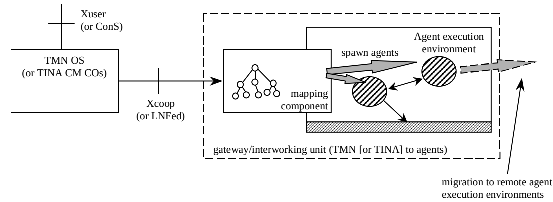

Figure 6 – Interworking with agents system

In the above figure, the mapping component accepts Xcoop operations (gets, sets, actions, etc.) and packages the operations into a mobile agent either singly or as a set. The resulting mobile agent is therefore programmed to specifically implement the requested operations in either a local or remote agent execution environment. In the case of remote execution environments the mobile agent will undergo migration between execution environments in different domains. As before there are two possible locations for the interworking unit. If it is placed in the agent-based domain then the inter-domain communications are still across either the X interface or the LNFed reference point. If there is a local gateway within the TMN or TINA domains then the remote operations will be embedded in migrating agents between domains.

Agents can be programmed to interwork with either TMN or TINA interfaces and information models provided that the underlying TMN or TINA interface is modelled in the agent API of the agent execution environment associated with the TMN or TINA based management system. However a mapping is required for converting CMIS/GDMO and IDL to the agent API language, or agent communication language (ACL). While extensive work has been undertaken for mappings between CMIS/GDMO and IDL the issues of agent API translations of TMN or TINA based interfaces and associated information models is still largely for further study.

Assuming that agent technologies, together with appropriate interworking units, gateways and mapping components, are suitable for a common approach to interdomain connection management across heterogeneous management technologies, there remain a number of outstanding issues regarding procedures and sequences of interactions to achieve connectivity across several domains. Here, it is assumed that using migrating agents to visit the server management domains performs all interdomain management operations. Two approaches are envisioned: the first assumes N different agents are spawned to visit N domains in parallel; the second assumes a single agent is spawned who visits each of the N domains in turn before returning to its home domain. These two styles of operation are illustrated in the following figure.

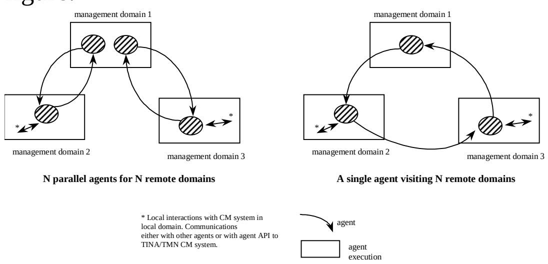

Figure 7 – Agents migration

6. Conclusions

We have described in this paper three approaches for the implementation of multi-domain federated Connectivity Management functionality namely a TINA compliant approach, a TMN compliant approach and a Mobile Agent based approach. Full implementations are needed for a complete evaluation and comparison for the provision of the CM service.

The proposed architectures and enterprise model make possible the development and provision of the CM Services as distributed systems. This will allow many PNOs to co-operate in providing end-to-end manageability of their broadband networks.

Three approaches are analysed and reviewed in this paper: the TMN approach, based on the X-user and X- coop interfaces, using OSI systems management as the underlying technology; the TINA approach, using the ConS and LNFed reference points, with CORBA as the base technology for the computing architecture; and an approach based on intelligent agents and mobile code using OMG's MAF as the enabling technology. The three approaches are reviewed from the points of view of architecture and technology considering the issues of system design, migration and interworking.

Several possibilities for the future integration and/or interworking between systems based on the three approaches are proposed, including: enhancements to TMN and TINA architectures to make them agent-friendly; the use of agent technologies to implement TMN or TINA compliant building blocks; the use of agents to enable interworking between diverse technologies and architectures. The outstanding issues are raised which highlight the areas of future work which are expected to be carried out by a number of research projects, which have recently been awarded.

ACKNOWLEDGEMENTS

The work presented in this paper has been carried out in the context of ACTS projects: MIAMI [47], MISA [13] and REFORM [14]. The ACTS programme is partially funded by the Commission of European Union.

REFERENCES

[1] IBM (1996) Programming Mobile Agents in Java(tm), http://www.trl.ibm.co.jp/aglets

[2] J. Bloem, et al, The TINA-C Connection Management Architecture, proceedings of TINA’95, Melbourne, Australia, Feb. 1995

[3] Bernes - Lee T., Fielding, R. and Frystyk H. - HTTP/1.0 RFC 1945, Network Working Group, May 1996.

[4] Chess D et. al. (1995) Itinerant Agents for Mobile Computing, IEEE Personal Communications Magazine 2(5) 34-59

[5] Conradie L. and Mountzia M.-A (1996) A Relational Model for Distributed Systems Monitoring using Flexible Agents in Proceedings of the Third International Workshop on Services in Distributed and Networked Environments.

[6] “CORBAservices: Common Object Service Specification”, Object Management Group, 1996

[7] “Common Object Request Broker: Architecture and Specification”, r2.0, Object Management Group, 1995

[8] Finin T et. al. (1994) KQLM as an Agent Communication Language, Proc CIKM'94, ACM Press

[9] FIPA (1996) FIPA's First Call for Proposals, http://drogo.cselt.stet.it/fipa/

[10] FIPA (1997) First set of FIPA specifications http://drogo.cselt.it/fipa/spec/fipa97.htm

[11] Frost RH (1996) Java agent template, http://cdr.stanford.edu/ABE/JAVAAgent.html

[12] L. de la Fuente, et al, Application of the TINA-C Management Architecture, in A. Sethi, Y. Raynaud and F. Faure-Vincent (eds.), Integrated Network Management IV, Chapman & Hall, pp. 424-435, 1995.

[13] Galis A, Brianza C, Leone C, Salvatori C, Gantenbein D, Covaci, Mykoniatis G, Karayannis F - Towards Integrated Network Management for ATM and SDH Networks supporting a Global Broadband Connectivity Management Service in “Intelligence in Services and Networks: Technology for Cooperative Competition“, Mullery A (Eds.)- Springer- Verlag, Berlin, 1997, ISBN 3-540-63135-6

[13] Griffin, D., Pavlou, G., Tin, T., "Implementing TMN-like Management Services in a TINA Compliant Architecture - A Case Study on Resource Configuration Management," Intelligence in Services and Networks: Technology for Cooperative Competition, ed. A. Mullery, M. Besson, M. Campolargo, R. Gobbi, R. Reed, pp. 263-274, Springer Verlag, 1997.

[14] Genesereth M R and Fikes R E (1992) Knowledge Interchange Format Version 3.0 Reference Manual, Technical Report Logic-92-1, Stanford University

[14] Guideline NIM-G4, "TMN and TINA Co-existence", NIM Chain, version 2, June 1997

[15] ITU-T Recommendation G.803, “Architectures of Transport Networks Based on the SDH.”

[16] Harrison CG, Chess DM and Kershenbaum (1995) Mobile agents: are they a good idea? TR 1995 IBM Research Division, http://www.research.ibm.com/xv-d953-mobags-ps

[17] ITU-T Recommendation I.320, “ISDN protocol reference model”

[18] ITU-T Recommendation I.321, “B-ISDN protocol reference model and its application”

[19] “Specification of the Interface Definition Language (IDL)”, Object Management Group.

[20] ITU-T Q.1200 series, Intelligent Networks.

[21] “CORBA Internet Inter-Operability Protocol.”, Object Management Group

[22] X/Open/NMF, Joint Inter-Domain Management (JIDM) Specifications - “SNMP SMI to CORBA IDL, ASN.1/GDMO to CORBA IDL and IDL to GDMO/ASN.1 translations”, 1994.

[23] Li W (1996) Java-To-Go: Interactive Computing Using Java, http://ptolemy.eecs.berkeley.edu/dgm/javatools/java-to-go/

[24] ACTS MISA AC080- Consortium and Deliverables- Management of Integrated SDH and ATM Networks, http://misa.zurich.ibm.com/

[25] ITU-T Recommendation M.3010 - 1995 - Principles for a Telecommunication Management Network

[26] Magedanz T and Eckardt T (1996) Mobile software agents: a new paradigm for telecommunication management, IEEE/IFIP Network Operations and Management Symposium (NOMS), Kyoto, Japan

[27] Magedanz T and Krause S (1996) Intelligent Agents: An Emerging Technology for Next Generation Telecommunications?, Proc IEEE INFOCOM'96, IEEE

[28] Magedanz T: "On the Impacts of Intelligent Agent Concepts on Future Telecommunication Environments", in: Lecture Notes on Computer Science 998 - "Bringing Telecommunication Services to the People - IS&N'95", pp. 396 - 414, A. Clarke et al. (Eds.), ISBN: 3-540-60479-0, Springer Verlag, 1995

[29] OMG (1996) Mobile Agent Facility Specification, Joint Submission by Crystaliz Inc., General Magic Inc., GMD FOKUS, IBM and The Open Group ftp://ftp.omg.org/pub/docs/orbos/97-10-05.pdf

[30] ITU-T Recommendation M.3010, “Principles for a Telecommunications Management Network (TMN)”, SG IV,1996.

[31] ITU-T Recommendation M.3020, “TMN interface specification methodology”, 1995

[32] “Network Resource Architecture,” The TINA Consortium, Version 3.0, Feb. 1997.

[33] “Network Resource Information Model,” The TINA Consortium, Document No. TB_LRS.011_2.0_94, Dec. 1994.

[34] ITU-T Recommendation X.903/ISO 10746-3, “Basic reference model of Open Distributed Processing”, 1994

[35] “Comparison of the OSI management, OMG and Internet management Object Models”, Joint X/Open/NM Forum Inter-domain Management Task Force, Dec. 1993.

[36] “Evolution, Migration Paths and Inter-working to TINA, Deliverable 1”, Project P508, EURESCOM, 1996.

[37] Ranc, D., Pavlou, G., Griffin, D., Galis, A., "A Staged Approach for TMN to TINA Migration", proceedings of the TINA'97 Conference, Santiago, Chile, November 1997, Proceedings to be published by IEEE

[38] Kurt Rothermel, Radu Popescu Zeletin (Eds.) - Mobile Agents, First International Workshop, MA'97, Berlin, April 1997, IBSN 3-540-62803-7 Springer Verlag

[39] "An Overview of the Telecommunications Information Networking Architecture (TINA)", TINA'95 Conference, Melbourne, Australia, 1995.

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

[41] "TINA Reference Points", The TINA Consortium, Version 3.1, June 1996.

[42] TINA consortium, Connection Management Architecture, Draft, Document label TB_JJB.005_1.5_94, TINA-C, March 1995.

[43] Tokoro M (1996) Agents: Towards a Society in which Humans and Computers Cohabit, in Perram TW and Muller J-P (eds) Distributed Software and Applications, LNCS 1069 Springer Verlag

[44] White JE (1994) Telescript Technology: The Foundation for the Electronic Market-place, General Magic Inc., http://wwwgenmagic.com/Whitepapers

[45] ITU-T Recommendation X.701, "Information Technology - Open Systems Interconnection - Systems Management Overview", 1992.

[46] Zhang T and Covaci S (1996) Intelligent Agents in Network and Service Management, Proc IEEE Globecom'96, IEEE, 1855-1861

[47] http: //www.fokus.gmd.de