Construction of a TMN Network Element for the Configuration Management of ATM Networks

Luiz Fernando Kormann
kormann@lrg.ufsc.br

Ketter Ohnes Rogerio
ohnes@lrg.ufsc.br

Carlos Becker Westphall
westphal@lrg.ufsc.br

UFSC - CTC - INE - LRG
Laboratório de Redes e Gerência
Cx. Postal 476 CEP 88040-970 Florianópolis - SC - Brasil
Phone: +55 48 231-9739 FAX: +55 48 231-9770

Abstract

Network management is a difficult task and becomes even more complex when involves the requirements of a high performance network as in the case of the ATM architecture. The work presented in this paper proposes the implementation of recently released ITU-T recommendations and investigates the usage of TMN concepts for the management of ATM networks. Within this context, efforts were concentrated on the development of configuration and fault functions for the control of virtual channel and virtual path connections. More specifically, the work under development covers the monitoring and control of establishment and release of an end-to-end ATM connection, the allocation of corresponding VP/VC identifiers and the activation of the cell flow applied to the connection. Furthermore a testing mechanism through the usage of OAM cells was also implemented.

Keywords

ATM, TMN, Configuration Management, VP/VC connections.

1. Introduction

The ATM (Asynchronous Transfer Mode) represents the main support for the B-ISDN (Broadband Integrated Service Digital Network) architecture. The benefits of a scaleable bandwidth, performance improvements and guarantees of Quality of Service justify the tendency in employing ATM over the existing LAN (Local Area Network) and WAN (Wide Area Network) technologies. Emphasis may also be focused on the current and future requirements for the provision of different types of services within a singular high performance network architecture.

Recent researches have employed the concepts inherent to TMN (Telecommunications Management Network) for the management of ATM networks [AyTa 96] [Ku 96] [WeKo 96]. The interfaces provided by the CMIP (Common Management Information Protocol) [X.711] allied with the distributed characteristic of OS (Operation Systems) and NE (Network Elements) contribute for the construction of complex applications which are suitable to the traffic demand maintained in an ATM network. Moreover, different views of a same ATM component are subject to be modelled by distinct network elements under the control of a management application.

The ATM network element described in the paper encompasses relevant functionalities involving the configuration management of VP (Virtual Path) and VC (Virtual Channel) connections in an ATM equipment. More specifically, the work under development covers the monitoring and control of establishment and release of an end-to-end ATM connection, the allocation of corresponding VPI/VCI (Virtual Path Identifier/Virtual Channel Identifier) and the activation of the cell flow applied to the connection. Methods were implemented in order to guarantee that parameters associated with bandwidth and Quality of Service may suffer changes even after the connection has already been established. In addition, facilities regarding the management of the establishment and release of VP/VC trail termination as well as the subsequent binding to a VP/VC link connection endpoint are also under construction.

The ATM network management also considers aspects of fault management which includes the immediate notification of failures that may happen in the ATM network element and the implementation of a testing mechanism through the usage of OAM (Operations, Administration and Maintenance) cells. In the latter, specific loopback cells are inserted into the physical medium for further analysis of their return. Additional functionalities are, at moment, being investigated in order to control services related to bandwidth allocation and the collection of usage metrics for the appropriate utilisation by performance and accounting management applications.

Within this context, a project is under development at Universidade Federal de Santa Catarina in order to install an ATM backbone for the interconnection of the whole set of sub-networks enclosed by the UFSC network. At present, the project involves two ATM IBM 8260 switches and a third Bay Networks Centillium 100 ATM switch which are interconnected over links of 100 and 155 Mbps.

The activities involved in the construction of those previously mentioned management applications correspond to part of the objectives proposed by the multi-institutional PLAGERE Project [West 94] - Platforms for Network Management, funded by the Brazilian National Research Council.

This paper is organised as follows. Section 2 outlines the context where the management functions implemented in this work take place while the following section describes the ATM network environment under analysis. Section 4 presents the development environment encharged to support the ATM management services, the information model adopted on the construction of the configuration functions for the virtual path and virtual channel connections as well as outlines aspects involved during the construction of the network element. Finally, the following section presents the conclusion and previews some future works.

2. Configuration Service Provisioning

The TMN model defined by the ITU-T (International Telecommunication Union - Telecommunications) proposes a distributed operation system architecture with standardised operation interfaces between the management components. Such components employ the CMIP protocol to exchange operation messages and are modelled by a completely object oriented management information model which is structured in three different containment, inheritance and register hierarchies. Managing systems are represented by operation systems which are responsible for monitoring and controlling the network elements distributed in a network. Both components, operation systems and network elements, are connected through a data communications network that is encharged for the proper exchange of information. In addition, a whole set of interfaces is also defined for intra and inter-network communications.

A basic approach for the construction of management services consists on the provisioning of a notification mechanism that supports the emission of reports informing the current state of the network element being controlled. In this context, primary notifications may inform the management application when the ATM network element has been initialised and is then available for subsequent operation. These notifications need only to provide a simple indication that the initialisation of the ATM network element is completed. Once management information had been collected and the operational state of the network element had been analysed, distinct operations may be performed in order to update control parameters for the current and future demands of the network.

The relationship between the elements involved in the establishment of Virtual Path and Virtual Channel connections are illustrated on figure 1. The Virtual Channel Layer contains the adaptation function for the trail access point and the subsequent trail termination function for the appropriate binding of the input/output port of the VC trail termination point to the corresponding VC connection point. Similar proceedings are performed at the Virtual Path Layer for the configuration of the parameters associated with establishment of a VP connection.

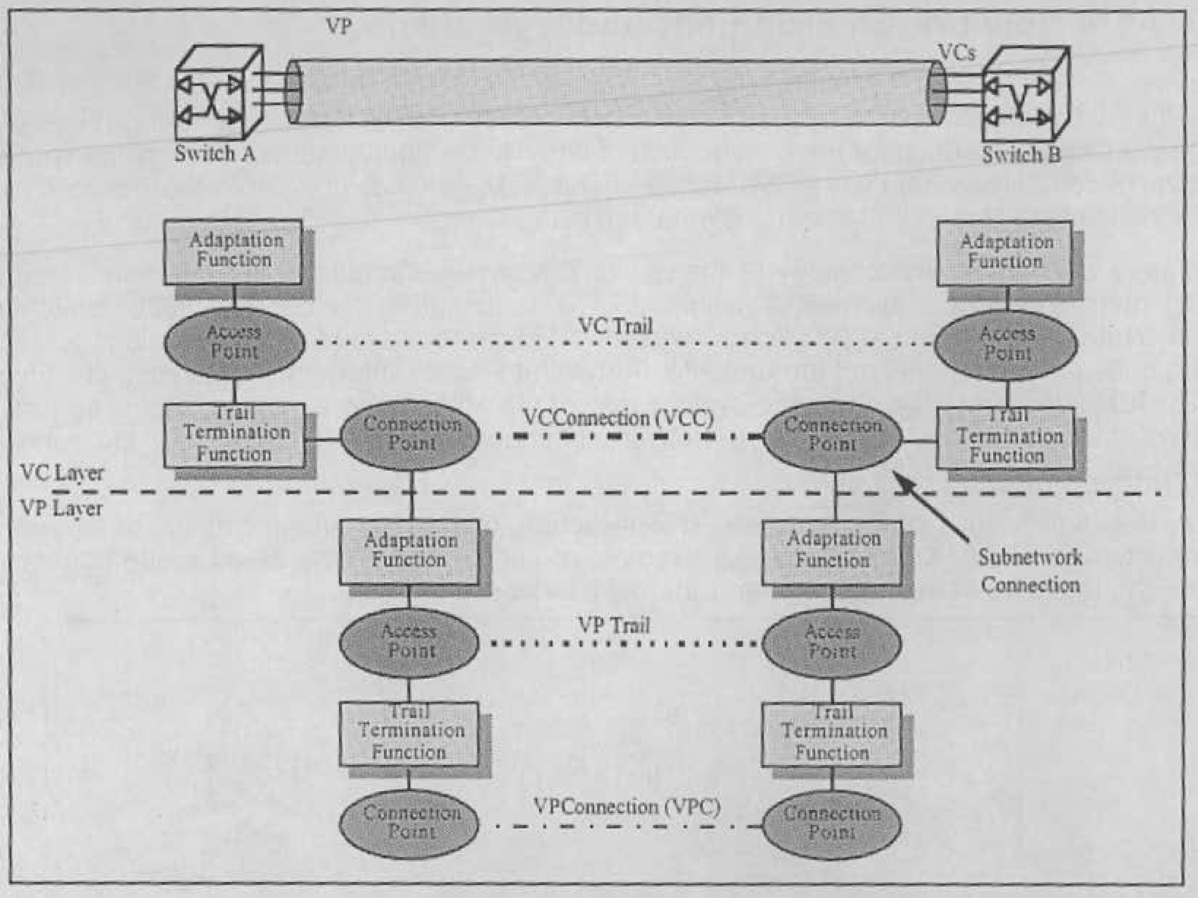

Figure 1: VP and VC Connection Model Between ATM Switches.

Relevant parameters regarding aspects related to the configuration of the UNI (User Network Interface) and NNI (Network Node Interface) may be set up for the proper functionality of the ATM components. The information in question mainly defines the interface identifier, the underlying trail termination point, the maximum number of simultaneously active VPC/VCC (Virtual Path Connection/Virtual Channel Connection) supported and the number of allocated VPI/VCI bits. The latter is essential to adapt addressing constrains to the low supported VPI/VCI values of the equipment on each end of the interface. In the case of UNI and NNI cells, two types of VP and VC processing are performed according to the end functionality of the cell interface. Configuration functions must identify the cell interface and allocate proper VPI bit space. Such functionality attends a requirement of the NNI cells for a larger VP identifier .

Subsequent activities associated with the establishment of an end-to-end connection involves the configuration of the VP/VC connection termination points at the input and output ports of the ATM network element. The parameters settled will configure the connection of both termination points and maintain the flow of information. Configuration functions shall then support the creation of appropriate VP/VC connection termination points through identifying the input and output ports of the respective VP/VC connection to be established. As previously mentioned, VP and VC link parameters may be altered at any time between the establishment and release of the connection assuring flexibility in the configuration of the network and also be configured as a segment or non-segment endpoint according to the requirements of the ATM network management application.

Similar proceedings shall be taken with relation to the establishment and release of trail termination points. In this case, configuration functions are required to identify the VP/VC link connection endpoint and instance the VP/VC trail termination function. Additional functions should also be provided in order to bind the input/output port of the VP/VC trail termination function to the VP/VC link connection endpoint, and finally initialise the relevant VP/VC trail termination function parameters. Such parameters may be then modified to reflect the current state of the traffic demand.

Special proceedings shall be adopted for the establishment and release of VP/VC cross connections and provide the functionality to reserve, create, release the connections in question and deallocate the corresponding resources.

3. The ATM Network Environment under Analysis

This section describes the testbed environment proposed at Universidade Federal Santa Catarina (UFSC) as an initial approach for the construction of an ATM backbone interconnecting the whole set of sub-networks contained within the UFSC network. The ATM switches present in the network represent the main focus of the Network Element implemented in this work.

Figure 2 illustrates the topology of the network where workstations are interconnected by ATM switching hubs. The ATM equipment available at UFSC is composed of two IBM 8620 switching hubs and a Bay Network Centillium 100 switch, in addition ATM cards are found in workstations and PCs. The switching hubs are interconnected through NNI (network-to-node) interfaces over links of 155 and 100 Mbps, The IBM 8260 are interconnected at a bit rate of 100Mbps, whereas the connection to Centillium 100 is performed at 155Mbs. UNI (user-to-network) interfaces are only employed for the connection of the ATM cards to the switching hubs.

A relevant feature of those three ATM switches consist on the provision of ports for the interconnection to non-ATM interfaces. An example of such facility is observed in the connection of a PC to the Centillium 100 switch through an Ethernet interface.

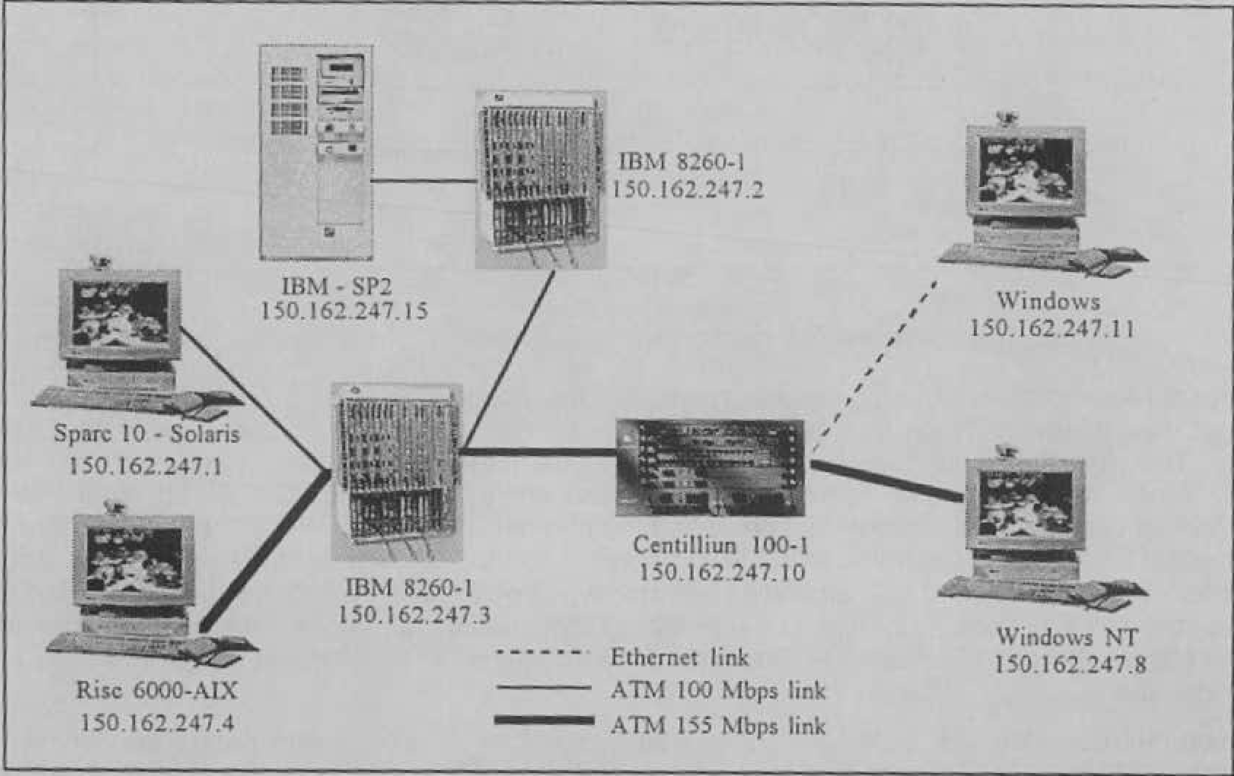

Figure 2: ATM Testbed Environment Available at UFSC Network.

IBM 8260 adopts both types of virtual connections, PVC (Permanent Virtual Connection) and SVC (Switched Virtual Connection). The support to Permanent Virtual Connections allows the usage of external management systems for the proper configuration of the parameters required on the establishment and release of virtual path connections and virtual channel connections. A sub-set of these parameters are used to identify a connection termination point and describe relevant features related to guarantees of QoS and bandwidth allocation. Such parameters are subject to be configured through an SNMP MIB II which allows the usage of a management platform to control the virtual connections. As a consequence, specific tables contain values that identify the connection and define the type of ATM interface.

The information model embedded in the IBM 8260 and Centillium 100 ATM switches is not in conformity with the TMN architecture, such characteristic implies the usage of a Q-adaptor to make possible the control of the switches through a TMN management platform. The OSIMIS platform employed in this work provides a CMIP/SNMP gateway which implements the functionality inherent to a Q-adaptor. Nevertheless, additional methods must be added to the Q-adaptor in order to implement the mapping of information required by the ATM network element.

4. Information Model and Implementation Aspects

The ITU-T has recently released a set of recommendations proposing new functions for the management of both Virtual Channel and Virtual Path layers. The information model specified in those recommendations basically describes managed object classes that inherit characteristics previously defined in the object classes proposed by the [M.3100][Q.821][Q.822] recommendations. The collection of the such managed objects provides management information specifically for ATM network elements and defines VP and VC management functions for the ATM layer, as well as an information model for the transmission convergence sub-layer.

The generic network information model provided by the [M.3100] recommendation contains managed object classes that are useful to describe the information exchanged across all interfaces defined in the TMN architecture. This information model mainly identifies TMN object classes that are common to manage telecommunication networks, and is generic enough to be used in the management of networks at a technology independent level. Such feature guarantees the viability of the TMN concepts for the abstraction, and subsequent management, of components specific of an ATM network.

4.1 Development Environment

The implementation of configuration functions for the ATM VP/VC connections was developed in C++ language and employed the services provided by the OSIMIS (OSI Management Information Service) platform **[PaKn 95] [PaTi 94]**. The choose of the OSIMIS platform as the development environment for the mentioned management functions is justified by the set of high level Application Program Interfaces present in the platform which considerably hides part of the complexity of the CMIP protocol. Within this context, distinct interfaces may be employed for the construction of managed and managing systems and respectively correspond to the GMS (Generic Managed System) interface in the case of a network element, and the RMIB (Remote MIB) and SMIB (Shadow MIB) for managing activities. Moreover, the application developer is also supplied with common structures for the development of both managing and managed processes, such structures implement aspects that include the support for asynchronous event-driven applications and transparent handling of the ASN.1 (Abstract Syntax Notation. One) syntax. In addition, a GDMO (Guidelines for the Definition of Managed Objects) compiler **[GDMO]** was employed for the construction of managed object classes.

The use of GDMO specifications for the design of all object classes significantly reduced the efforts in modelling new management functions in two main aspects: a) the automatic generation of C++ code from an original GDMO specification; and b) the inherent concept of object orientation which induces, besides other features, the benefits of code reutilization. In the latter, an expressive part of the ATM network element being implemented consists on common structures that may be used by a variety of management applications within distinct network technologies.

4.2 The TMN Information Model

In the construction of the objects present at the M.3100 information model, special treatment were adopted for the classes that would be later employed by the ATM network element. Such classes are used, besides other functionalities, to model relevant logical aspects that support the initialisation of a network element, to provide mechanism for event manipulation, and to identify the basic components of a network equipment. In the latter, object classes may represent different ensembles within the same network element through the abstraction of networks, managed elements, termination points and specific details associated with switching and transmission techniques. The inheritance hierarchy of the management information model contains within its branches, a group of managed objects that provides the basic attributes, actions and operations for the implementation of a network element view of trail and connection termination points required during a connection establishment. Due to the fact that virtual channel and virtual path connections are defined by configurable parameters, such parameters and correlate activities may be straightly controlled by the managing system according to the network present condition [GrGe 95].

The information model depicted on figure 3 illustrates part of the inheritance hierarchy of the objects concerning the configuration of VP/VC connections. These objects cover the construction of an ATM network element with the support of pre-defined structures specified in the [M.3100] recommendation. Such object classes model generic features that are subjected to be employed by distinct network transmission techniques as in the case of the ATM technology.

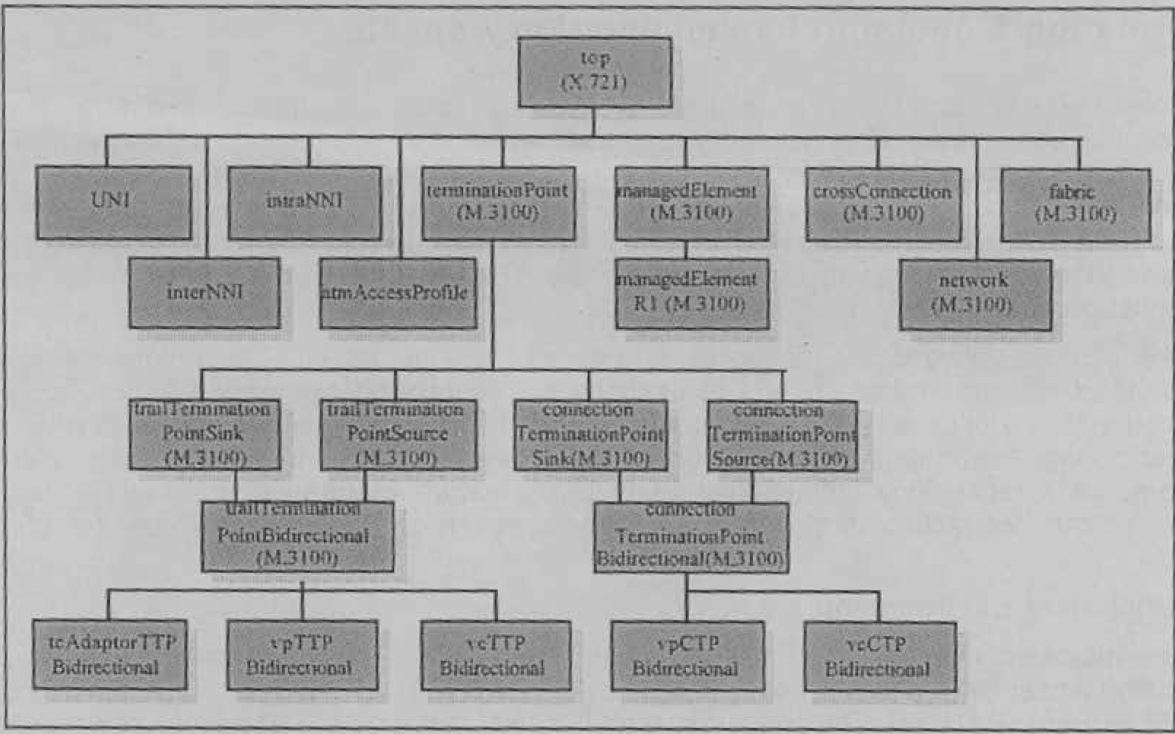

Figure 3: G.atmm Information Model Inheritance Hierarchy

TerminationPoint is the superclass of all managed objects that represents the trail termination or the connection termination point of a transport entity. The conditional and mandatory packages included in this class mainly identifies control parameters and correlate notifications associated with creation and deletion of the managed object as well as any alteration on the value of its attributes. The subsequent level in the inheritance hierarchy consists on the *trailTerminationPointSource* and *trailTerminationPointSink* object classes whose main objective is the identification of the access point where a trail is respectively originated or terminated. More complex functionalities are driven to the objects related to the establishment and release of a connection, in this case, *connectionTerminationPointSource* and *connectionTerminationPointSink* classes provide the appropriate methods to initialise and terminate a link connection. Moreover, conditional packages comprise structures which inform the channel number and the connection termination point identifier.

The methods for the generation and termination of a trial end point may be derived into a single *trailTerminationPointBidirectional* object class through the multiple inheritance of the corresponding source and sink superclasses. An analogy may also be made for the *connection-TerminationPointBidirectional* object whose main features are multiply inherited from the corresponding *connectionTerminationPointSource* and *connectionTerminationPointSink* object classes.

4.3 The ATM Network Element Information Model

Specific objects for the configuration management of VC/VP connections implemented in this work are in conformity with the specifications present in the [G.atmm] ITU-T recommendation. More specifically, emphasis were given to the provision of methods for the bidirectional connections required in point-to-point transmissions. Within this context, the *vcTTPBidirectional* class is employed to delimit the virtual channel connection and provides operations to request the termination point in order to insert an OAM cell for downstream loopbacking and report whether the cell was returned within the required time. The information supplied by the network management must identify, among others features, the cell insertion point, the location for the cell loopback and an indication whether the OAM cell is restricted to a segment or represents an end-to-end cell. In the latter, the cell PT (Payload Type) header field shall be configured with the appropriate value. Similar functionalities are presented by the *vpTTPBidirectional* class where the same actions are performed for OAM cell loopback testing within the scope of a virtual path trail termination point. Additional operations involving the trail termination point are included in the *tcAdaptorTTPBidirectional* class whose main focus is the representation of the point where the adaptation of the ATM Layer to the underlying physical structure takes place. Moreover, the object may also inform whether the activity of cell scrambling is being performed over the ATM interface.

In conformity with the definitions proposed by the [G.atmm] recommendation, the *vcCTPBidirectional* object class presents the proper methods to delimit virtual channel links. The attributes included in this class basically describe the VCI value, a collection of traffic descriptors and the Quality of Service class assigned to the virtual channel link termination being represented. A relevant characteristic introduced in this class consists on the assertion of the link connection VCI value to the virtual channel connection termination point identifier attribute which is employed by the object for the construction of the naming hierarchy. In addition, attributes describing performance metrics as the following peak cell rate, sustainable cell rate, burst tolerance, cell delay variation tolerance and Quality of Service class are also under the monitoring activities carried out by the object. From the fault management perspective , this object class also provides an OAM cell loopback package for the appropriate link connection test. The whole set of attributes, operations and behaviours mentioned at the virtual channel connection termination point bidirectional object class was maintained to the *vpCTPBidirectional* class in the virtual path level. Such feature guarantees the usage of separate objects with similar processing for the distinct virtual channel and virtual path sub-layers included in the ATM layer.

Once configured the trail and connection termination points in both VC and VP sub-layer, the *atmFabric* object class, shown on figure 3, may be created to actually control the establishment and release of an ATM cross-connection. In this case, the termination points to be connected or disconnected are identified by the specification of the respective *vcCTPBidirectional* object or *vpCTPBidirectional* object, or by specifying the characteristics of the termination end points. Additional information may be provided to configure the network element as an intra-NNI, inter-NNI or UNI interface, in this case the respective *intra-INN*, *inter-NNI* and *UNI* object classes are encharged to identify the managed system through providing a reference to the *tcAdaptorTTPBidirectional* object for the proper representation of the point where the adaptation of the ATM layer to the underlying physical infrastructure takes place. A conditional package contained in those three classes is used to establish the point where an OAM cell is looped-back, i.e. a point where the cell shall begin its return.

Finally, as shown on figure 3, the *atmAccessProfile* object class is employed to identify the maximum number of VPCs or VCCs that may be simultaneously active while the connection is maintained, as well as the maximum number of supported bits in the VPI or VCI fields.

4.4 Construction Aspects of the ATM TMN Network Element

As previously mentioned, attributes in the OSIMIS platform are implemented through object classes in C++ language. Each object class makes reference to generic methods provided by support mechanism for the transparent ASN.1 syntax notation handling. This methods implement the aspects to consult and change the attribute values, as well as codify (from C structure to an ASN.1 presentation element), and decodify (from an ASN.1 presentation element to a C structure) the assigned values[RoCo 96]. The attributes are represented through object classes, and as a consequence the ASN.1 specification associated with the attribute syntax must be implemented according to the object oriented paradigm. The first step for implementing the attributes consist on the automatic generation of C structures by the *Pepsy* ASN.1 compiler provided by the ISODE platform. A sub-set of the generated C structures are identified in the second column present in Tables I and II. Following to the construction of the data structures are implemented the functions for the manipulation of the ASN.1 syntax. The final step in the attribute construction consist on the development of the C++ object classes for the proper representation of the semantic aspects of the attributes. The third column of Tables I and II outlines the new attribute classes implemented in this work.

Table I: Attribute Classes from G.tmn Recommendation

Attribute Syntax	C Structure	C++ Class
BurstTolerance	type_AtmMIBMod_BurstTolerance*	BurstTolerance
CDVTolerance	type_AtmMIBMod_CDVTolerance*	CDVTolerance
PeakCellRate	type_AtmMIBMod_PeakCellRate*	PeakCellRate
QosClass	type_AtmMIBMod_QosClass*	QosClass
SustainableCellRate	type_AtmMIBMod_SustainableCellRate*	SustainableCellRate
OAMPeakCellRate	type_ASN1TypeModule_OAMPeakCellRate*	OAMPeakCellRate

Table II: Attribute Classes from M.3100 Recommendation

Attribute Syntax	C Structure	C++ Class
AlarmStatus	type_ASN1DefinedTypesModule_AlarmStatus*	AlarmStatus
CurrentProblemList	type_ASN1DefinedTypesModule_CurrentProblemList*	CurrentProblemList
SystemTimingSource	type_ASN1DefinedTypesModule_SystemTimingSource*	SystemTimingSource
SystemTitle	type_ASN1DefinedTypesModule_SystemTitle*	SystemTitle
ObjectList	type_ASN1DefinedTypesModule_ObjectList*	ObjectList
ConnectivityPointer	type_ASN1DefinedTypesModule_ConnectivityPointer*	ConnectivityPointer
CrossConnectionObjectPointer	type_ASN1DefinedTypesModule_CrossConnectionObjectPointer*	CrossConnectionObjectPointer
DownstreamConnectivityPointer	type_ASN1DefinedTypesModule_DownstreamConnectivityPointer*	DownstreamConnectivityPointer
SupportableClientList	type_ SMI_ObjectClassList*	SupportableClientList
PointerOrNull	type_ASN1DefinedTypesModule_PointerOrNull*	PointerOrNull

Alarm status and *Current Problem List* attribute types are both employed to cover aspects of fault management and respectively describe the severity of the alarm emitted by an object class and identify current problems that may be observed by the Network Element. The information contained in the *Current Problem List* include a large variety of alarms that may vary from a simple notification about the inexistence of an object instance to more complex types of notification as the indication that two termination points were previously connected.

The following *Connectivity Pointer*, *Cross Connection Object Pointer* and *Down Stream Connectivity Pointer* attribute types required in the Network Element mainly identify the object instances that represent the termination point of a connection or trail. Performance parameters in the network are mainly required by the [G.atmm] G.atmm recommendation and describe the information related to the maximum rate of cell as well as inform the sustainable flow of cells in a network. Tolerance of cell flow assigned to the VP and VC link being terminated. The Quality of Service is also assured through the identification of the class of service being provided.

Table III outlines a set of additional attributes required by the ATM Network Element which make use of some pre-defined classes present in the OSIMIS platform. The sub-set of these attributes characterised by the *SimpleNameType* syntax is basically employed to identify the object instances enclosed in the containment hierarchy. Some distinct attributes are used on different aspects of the network element and may indicate whether the appropriate object instance was configured to represent a segment end-point or indicate an eventual cell delay variation assigned to an OAM ingress and egress direction. An additional attribute is also provided in order to specify the number of a channel connection.

Finally, optional attributes were implemented in order to inform the site where the managed element is installed, the vendor of the resource under control, the version associated with the equipment and friendly name defined by a user to identify the resource.

Table III: Pre-Defined Attribute Syntaxes Present in the OSIMIS Platform.

Attribute	Attribute Syntax	Recommendation
oamEgressCDVTolerance	Integer	G.ATMM
oamIngressCDVTolerance	Integer	G.ATMM
segmentEndPoint	Boolean	G.ATMM
tcTTPId	SimpleNameType	G.ATMM
vcCTPId	SimpleNameType	G.ATMM
vcTTPId	SimpleNameType	G.ATMM
vpCTPId	SimpleNameType	G.ATMM
vpTTPId	SimpleNameType	G.ATMM
channelNumber	Integer	M.3100
characteristicInformation	OID	M.3100
cTPId	SimpleNameType	M.3100
locationName	GraphicString	M.3100
managedElementId	SimpleNameType	M.3100
networkId	SimpleNameType	M.3100
networkLevelPointer	ObjectInstance	M.3100
tTPId	SimpleNameType	M.3100
userLabel	GraphicString	M.3100
vendorName	GraphicString	M.3100
version	GraphicString	M.3100

Actions are performed by the *Fabric* object in order to allow the establishment and release of a point-to-point connection either in the Virtual Path layer or in the Virtual Channel layer. The syntax implemented for the action contains a set of information that includes specific object instance that may reflect an UNI, an intra-NNI or an inter-NNI connection, in addition, further arguments are employed to describe the above mentioned performance parameters. Moreover, the behavior under construction in the Network Element is basically embedded in the previously mentioned attributes and actions.

5. Conclusions

The management services under development in this work provide the complete functionality for the management of a virtual channel and virtual path connections and constitute the support for the construction of new ATM network management function involving aspects associated with bandwidth allocation, traffic monitoring, path optimisation, and a variety of additional services related to security and accounting management. The information on the configuration of virtual channel and virtual path connection may be then obtained from a management information base. Such proceeding considerably reduces the costs for the construction of a managing system since the major part of the management application intelligence is performed by the functions included in the network element.

In the case of management of virtual path and virtual channel connections, was observed some aspects which involved, among others features, the existence of useful pre-defined constructions in TMN recommendations, the emerging interest for the extension of the present standardised information models and the crescent definition of new standards which mainly focus on the network entities within the ATM layer. Recent researches have largely employed the components involved in the TMN concept with extensions to the each particular requirement related to the functional area of the management application.

Future works will concentrate on the development of applications that will employ the facilities provided by the network element and increment the information model with new objects for the management of the ATM transmission sub-layer. Additional activities may involve a comparison of the TMN concepts to the similar functionalities within a TINA-C architectural framework.

Acknowledgements

The construction of the components inherent to the ATM Network Element required a considerable effort of C/C++ language implementation. The authors would like to thank Mrs. Ariadne M. Raschcowetzki, Mrs. Clarissa Carneiro Mussi, Mr. Juliano Fontoura Pereira, Mr. Nelson Sinibaldi Filho, Mrs. Vivianne Noemi Mazzorca, Mr. Robinson Mittmann and Mrs. Suzy Pascoali for their important contribution on the construction of the services under development in this work.

References

[AsWe 93] Assoul, S.; Westphall, C.B.; **Management in Broadband Networks: ATM Networks Fault Management**. Proceeding of the IFIP TC6/WG 6.4 International Advanced Processing Techniques for LAN and WAN Management. Versailles, France. 04/1993.

[AyTa 96] Aydemir, B.; Tanzini, J.; **An ATM Network-View Model for Integrated Layer-Network Management**. IEEE/IFIP Network Operations and Management Symposium NOMS´96, Kyoto, Japan, 04/1996.

[DeKo 96] De Franceschi, A.S.M.D.; Kormann, L.F.; Westphall, C.B. **Performance Evaluation for Proactive Network Management**. IEEE ICC'96 International Communications Conference. Dallas, Taxas, 06/1996.

[G.atmm] **ITU-T Draft Recommendation G.atmm:** Asynchronous Transfer Mode (ATM) Management of the Network Element View, 1995.

[X.710] **ITU-T Recommendation X.710**: Open Systems Interconnection - Common Management Information Service Definition. 07/1991.

[X.711] **ITU-T Recommendation X.711:** Open Systems Interconnection - Common Information Protocol Specification, Version 2, 07/1991.

[GDMO] **OSIMIS GDMO Compiler User Manual**, Department of Computer Science, University College London, 1993.

[GrGe 95] Griffin, D. P.; Georgatsos, P.; **A TMN System for VPC and Routing Management in ATM network.** International Symposium on Integrated Network Management - ISINM'95, Santa Barbara, California - USA, May 1995.

[KoWe] Kormann, L.F.; Westphall, C.B.; Coser, A.; **OSI Usage Metering Function for the OSIMIS Management.** Platform. Journal of Network and Systems Management. Plenum Publishing Corporation. Meddletown (to appear in the 3Q96, Vol. , No. 3) 1996.

[ku 96] Ku, B. S.; **Use of TMN for SONET/SDH Network Management**. IEEE/IFIP Network Operations and Management Symposium NOMS´96, Kyoto, Japan, 04/1996.

[M.3100] **ITU-T Recommendation M.3100** - Generic Management Information Model, 1995.

[PaKn 95] Pavlou, G.; Knight, G.; McCarthy, K.; et al, **The OSIMIS Platform: Making OSI Management Simple**, International Symposium on Integrated Network ISINM'95, Santa Barbara, California. USA. 05/1995.

[PaTi 94] Pavlou, G.; Tin, T.; Carr, A.; **High-Level Access APIs in the OSIMIS TMN Platform: Hernessing and Hiding**. 2nd Conference on Intelligent Services and Networks, Aachen, Alemanha, 09/1994.

[Q.821] **ITU-T Recommendation Q.821** - Stage 2 and Stage 3 Description for the Q3 Interface-Alarm Surveillance, 1993.

[Q.822] **ITU-T Recommendation Q.822** - Stage 1, Stage 2 and Stage 3 description for the Q3 Interface: Performance Management, 1993.

[WeKo 96] Westphall, C. B.; Kormann, L.F. **Usage of TMN Concepts for the Configuration Management of ATM Networks** , in Broadband Strategies and Technologies for Wide Area and Local Access Network, Berlin - Germany, October 1996, Roberto Vercelli, Editor ISBN 0-8194-2357-2, Proceedings SPIE vol. 2953, pp 250-7 (1996)

[West 94] Westphall, C.B. et al. **Detail Description of the PLAGERE Project (to ProTeM-CC-II/CNPq**, UFSC, CEFET-PR, CPqD-Telebrás and UFPB; in portuguese), 04/94.