Exploiting the Power of OSI Management for the Control of SNMP-capable Resources Using Generic Application Level Gateways

Kevin McCarthy†, George Pavlou†, Saleem Bhatti†, José Neuman De Souza‡

†Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, UK.
Tel: +44 71-380-7215 Email: G.Pavlou@uk.ac.ucl.cs.
‡Prism, 45 Avenue Des Etats-Unis, 78000 - Versailles, FRANCE.
Tel: +33 1 39 25 40 92 Email: Jose.Neuman@fr.uvsq.prism.

Abstract
A major aspect of Open Systems' network management is the inter-working between distinct Management architectures. This paper details the development of an Object Oriented Generic Application Level Gateway to achieve seamless coexistence between OSI and SNMPv1 management systems. The work builds upon the Network Management Forum's "ISO/CCITT and Internet Management Coexistence" activities. The power of the OSI Systems Management Functions is thus available for the management of SNMPv1 based resources, bringing fully event driven management to the SNMP domain.

Keywords: OSI; SNMP; TMN; Q-Adapter; Gateway.

1. Introduction

Whether driven by technological merit, simplicity of development or Government profiles, considerable investments have been made and will continue to be made into the provision of network management solutions based on the two dominant management protocol architectures, namely SNMPv1[RFC1155,RFC1157,RFC1212] and OSI[X701,720]. They exist together so they must be made to co-exist, so as to achieve global inter-working across heterogeneous platforms in the management domain.

It is the authors' contention that co-existence can most readily be achieved by selecting a semantically rich reference model as the basis for this inter-working. Such an approach can then be readily extended to encompass both up and coming technologies such as CORBA[OMG91], together with architectures that have not yet bridged the synaptic gap in the collective minds of standards bodies and manufacturers' consortia.

The collaborative work of the Network Management Forum's (NMF) ISO/CCITT and Internet Management Coexistence (IIMC) activities[] has provided a sound basis to our efforts in achieving co-existence through automated application level gateways. Through out this paper we shall use the terms "proxy", "application level gateway" and "Q-Adapter" [M3010] synonymously, to indicate the automated translation of information and protocol models, so as to achieve the representation of management objects defined under one proprietary paradigm (in TMN terms) under that of an alternative model (namely OSI).

The development of the gateway has been undertaken by the RACE Integrated Communications Management (ICM) project, to achieve Network Element Management of non-OSI resources. Partners from VTT (Finland), Prism (France), CET (Portugal) and UCL (UK) have been principally involved with this effort. ICM has a mandate to demonstrate the feasibility of integrating Advanced Information Processing technologies for Telecommunication Management Networks. The gateway has been developed using the Object Oriented power of UCL's OSI Management Information Service, development platform [OSIMIS] - the leading public domain implementation.

Finally, it is worth emphasising that this paper is not intended to champion the cause for either of the two dominant network management philosophies. The authors are happy for others to take on the role of Crusader, in what has become a Holy War. We prefer to accept the reality, that both paradigms have achieved a significant acceptance by their respective communities and consider that our task is to achieve coexistence between them, so as to facilitate the provision of OSI functionality to SNMP managed resources.

2. Comparison of the OSI and SNMP Models

The approaches are a result of two distinct (some would say diametrically opposed) underlying tenets. The Internet-

standard Network Management Framework is based on the notion of universal resource deployment. This may be alternatively stated by the fundamental axiom:-

"The impact of adding network management to managed nodes must be minimal, reflecting a lowest common denominator"[Rose91]

Which can be summed up by the two words "simple" and "implementable".

In contrast the OSI standardization process attempts to achieve an all encompassing framework, to meet any future management requirements. Since OSI standardization is a self-perpetuating process a great deal of thought was initially placed into the underlying object oriented model so as to allow for the planned continual expansion. OSI can be summed up by the two expressions "complex yet powerful" and "relatively difficult to implement, relatively easy to extend."

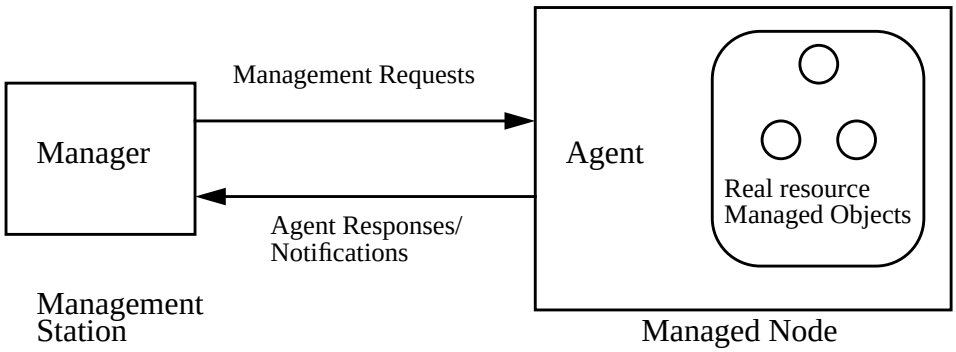

Fig. 1. The Manager/Agent Model.

If we consider the manager/agent model shown in figure (1), then under SNMP the burden of management would be placed firmly on the management station, with only minimal impact on the more numerous managed nodes. Under OSI a more significant load is placed on the agents due to a greater expectation of the capabilities of managed nodes.

Both camps set out with the same overall aim of achieving the effective management of heterogeneous resources. One took a pragmatic approach and achieved exceptional market acceptance, the other attempts to provide a complete solution at the expense of its complexity. The future may prove that the simple foundations of the SNMP model can not be continually built upon to meet the management needs of tomorrow's global open systems.

2.1 Management Information

Each agent provides a management view of their underlying logical and physical resources, such as transport connections and power supplies, to the managing applications. Managed Objects provide an abstract view of these real resources. The Managed Object data is held in a management database called the Management Information Base (MIB). Both SNMP and OSI define schemata for the description of Managed Object MIB data, namely the Structure of Management Information (SMI)[RFC1155,RFC1212] and the Guidelines for the Definition of Managed Objects (GDMO) [X722], respectively.

The OSI information model is object-oriented and permits the refinement of existing Managed Object templates via inheritance, see figure (2). Refinement may occur due to an increase in the capabilities of a given Managed Object, perhaps due to the evolution in the technology of the underlying resource. The OSI model supports allomorphism, which facilitates the management of a given object as if it was an instance of any of the object classes in its inheritance hierarchy, thus permitting managing applications that have been coded to an earlier version of the information

model, to continue to exercise control.

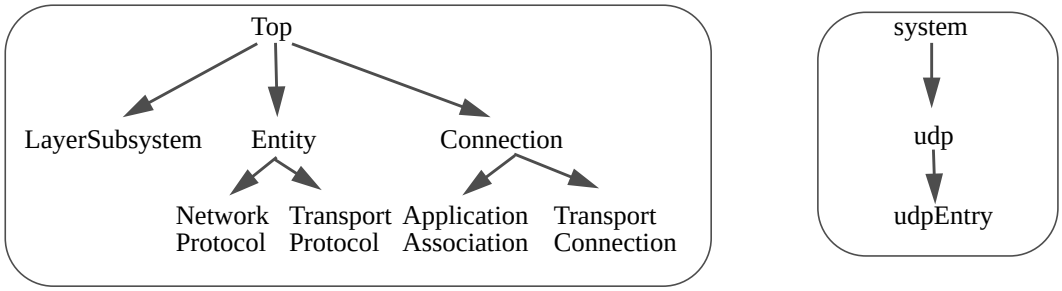

Fig. 2. example Inheritance and Containment Hierarchies

The aggregation relationships between managed objects, such as “kind-of” and “part-of”, are described by Name Binding containment descriptions. These containment descriptions yield a Managed Object instance hierarchy which is termed the Management Information Tree (MIT), see figure (2). The MIT facilitates globally unique instance naming via Distinguished Names.

SNMP’s object-based information model is simpler than its OSI counterpart so as to reduce the complexity of the agent implementations. SNMP objects represent single, atomic data elements that may be read or written to in order to effect the operation of the associated resource. The SNMP SMI permits the variables to be aggregated into lists and tables but there is no mechanism provided by SNMP to enable the manager to operate on them as a whole. Object identifiers are used, some would say misused, to achieve object instance naming, see figure (3). The syntaxes that each MIB variable may hold are a very much reduced subset of the unlimited syntaxes that are permitted by the OSI model.

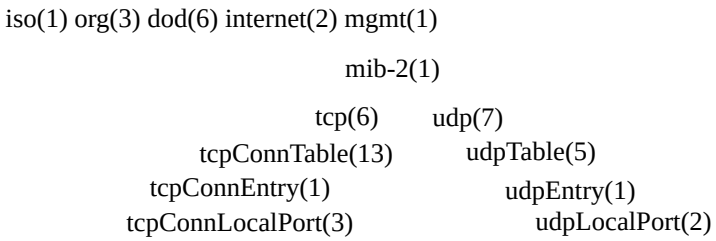

Fig. 3. An Internet Management MIB Object Identifier Instance naming tree

2.2 Protocol Operations

OSI makes a distinction between the service offered by a layer and the underlying protocol that achieves those services, whilst SNMP makes no such distinction. OSI management’s services and protocol are defined by the Common Management Information Service [X710] and the Common Management Information Protocol [X711] respectively.

In placing the emphasis for Manager/Agent communications between asynchronous interrupt-driven and polling based approaches, SNMP selected “trap-directed polling”, whilst OSI adopted an event driven approach. Upon an extraordinary event the SNMP agent emits a simple Trap notification to its manager, which must then initiate MIB polling to ascertain the full nature of the problem. Since Traps are both simple and unacknowledged their generation places only a small burden on the managed node. The manager may still need to poll important attributes periodically if the set of supported Traps are not sufficient to indicate the occurrence of all important error conditions. CMIS supports extremely expressive and optionally acknowledged event reports, to the managing application, via the M-Event-Report operation, thus removing the need for any additional polling. The onus is placed on the OSI agent to inform the manager of significant events.

The requirement for simplicity extends to the number and complexity of SNMPv1 protocol operations, compared with their OSI counterparts. CMIS operations may include specifications for “scope” and “filter” so that the operation may be applied to some subset of the agent’s managed objects. Scoping selects a sub-tree from the agent’s MIT and

filtering specifies a criteria, such as “those routing entries with a valid status”, to select from the scoped objects.

M-Get and M-Set are provided to retrieve and update attribute values. Since the usage of scoping and filtering means that the number of responses to an M-Get (which are received in linked replies) will not necessarily be known when the request is sent, an M-Cancel-Get operation is provided to prevent the possibility of the manager being over loaded. M-Create and M-Delete cause the creation and deletion of managed objects. M-Action facilitates the execution of any supported imperative command such as running diagnostic tests.

SNMPv1 supports the retrieval of management information via Get and Get-Next primitives, the latter facilitating MIB traversal. Retrieval responses are limited to a single packet, which ensures that the manager will not be overloaded with response data, at the expense of requiring multiple retrieval requests to traverse an entire table. The Set primitive is used to update MIB objects which, via side-effects, achieves the control of imperative actions and the creation or deletion of table entries.

2.3 Transport Mappings

Although SNMP is transport protocol independent, the connectionless User Datagram Protocol is the principal transport for SNMP. The “end-to-end” argument[Sal84] makes a very strong case for leaving the selection of aspects such as the transport protocol to the application level, since only the application (in this case management) has a complete appreciation of its transport requirements.

By selecting a connectionless protocol such as UDP the management implementation is free to produce its own timeout and retransmission mechanisms. At times of network congestion the SNMP implementation can then configure an appropriate level of retransmissions to increase the chances of successful management when the network itself is failing. The application can more readily determine when some form of out-of-bands communication is essential. This approach requires that each SNMP implementation must attempt to produce its own transport mechanism that will not end up accentuating any problems of network congestion.

OSI management is association based, requiring association establishment and removal phases, in addition to the transfer of management requests. It should be born in mind that manager/agent associations are intended to be held open for a period of time, thus spreading the cost of the association over a number of management requests. The Transport level implementation, whether TPx or TCP (if RFC1006 is followed), is entrusted with achieving the efficient delivery of management messages whatever the underlying network conditions.

2.4 Generic Functionality

OSI management standardization has greatly surpassed the more ad-hoc efforts of the SNMP community in defining functionality through an ever growing series of Systems Management Functions (SMFs). The Event-Reporting SMF [X734] permits the managing application to create Event-forwarding-discriminators at the agent, which control the selection and destination of all event reports that the agent generates. A related SMF is the Log-Control function [X735], which permits event logging according to manager configurable criteria.

To reduce requirements for remote polling and data retrieval the Metric Monitor [X738] and Summarization SMFs[X739] have been developed. Together they permit manager app[X722] ITU X.722,Information Technology - Structure of Management Information: Guidelines For The Definition

of Managed Objects, January 1992.lications to configure agents to undertake localised polling, threshold checking, data summarization and statistical analysis.

The X.500 Directory[X500] provides a global, hierarchically structured data repository. By incorporating the Directory into the OSI management model, the distributed transparencies, such as faults, replication and location transparency, can be achieved.

3. Management Coexistence

At an early stage in the design of the gateway the decision was made to build upon the work that has been undertaken by the Network Management Forum's ISO/CCITT and Internet Management Coexistence (IIMC) activities. The IIMC package currently consists of five documents [IIMCIMIBTRANS,IIMCOMIBTRANS,IIMCMIB-II,IIMC-PROXY and IIMCSEC]. Two of these documents are of the greatest significance to our work, namely "Translation of Internet MIBs to ISO/CCITT GDMO MIBs" and "ISO/CCITT to Internet Management Proxy".

As intimated above, although it was out intent to follow the IIMC specifications in full, a number of instances arose where we selected options that either differed with or continued on from the IIMC work. For example the IIMC define a "stateless" proxy, whilst our gateway is "stateful" and can thus take advantage of caching. Other issues such as achieving maximum efficiency in the protocol translation, improved automation and the inter-working with non-conformant SNMP agent implementations, have been given greater consideration by our research.

3.1 Mapping the Management Information Model

The procedures used in converting a MIB defined under the SNMP SMI into one using ISO GDMO templates are those defined by the IIMC [IIMCIMIBTRANS]. We shall first consider a short example of mapping the OBJECT-TYPE templates for the "udpTable" and "udpEntry" objects into the corresponding GDMO Managed Object Class (MOC) and Name Binding templates.

```
udpTable OBJECT-TYPE
        SYNTAX SEQUENCE OF UdpEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION "A table containing UDP listener information."
::= { udp 5 }

udpEntry OBJECT-TYPE
        SYNTAX UdpEntry
        ACCESS not-accessible
        STATUS mandatory
        DESCRIPTION "Information about a particular current UDP listener."
        INDEX { udpLocalAddress, udpLocalPort }
::= { udpTable 1 }

UdpEntry ::=
        SEQUENCE {
                udpLocalAddress IpAddress,
                udpLocalPort INTEGER (0..65536)
        }
```

The semi-automatic IIMC Internet MIB translation procedures the produce:-

```
udpEntry MANAGED OBJECT CLASS
        DERIVED FROM "Rec. X721|ISO/IEC 10165-2:1992":top ;
        CHARACTERISED BY udpEntryPkg PACKAGE
        BEHAVIOUR udpEntryPkgBehaviour
        BEHAVIOUR DEFINED AS
                !BEGINPARSE
                        REFERENCE !!
                        This managed object class maps to the "udpEntry" object
                        with object identifier {udpTable 1} in modeule RFC1213-MIB!!
                        DESCRIPTION !!
```

```
                    Information abou a particular current UDP listener. !!
                    INDEX  RFC1213-MIB.udlLocalAddress,
                           RFC1213-MIB.udpLocalPort
            ENDPARSE!;;
        ATTRIBUTES
            udpEntryId      GET     -- IIMC naming attribute --,
            udpLocalAddress GET,
            udpLocalPort    GET ;;;
        REGISTERED AS { iimcAutoTrans 1 3 6 1 2 1 7 5 1 };

    udpEntry-udpNB NAME BINDING -- RFC 1213-MIB --
        SUBORDINATE OBJECT CLASS udpEntry AND SUBCLASSES;
        NAMED BY SUPERIOR OBJECT CLASS udp AND SUBCLASSES;
        WITH ATTRIBUTE udpEntryId;
        BEHAVIOUR udpEntry-udpNBBehaviour
        BEHAVIOUR DEFINED AS
        !BEGINPARSE
            INDEX  RFC1213-MIB.udpLocalAddress,
                   RFC1213-MIB.udlLocalPort;
        ENDPARSE!;;
        REGISTERED AS { iimcManagementNB 1 3 6 1 2 1 7 5 1 }
```

It is worth emphasising certain aspects of the above translation. Firstly, information that is contained within the SNMP SMI, but can not be directly represented by the corresponding GDMO, is held in “BEHAVIOUR” clause “PARSE” statements, e.g. the objects used for entry indexing. Secondly, conceptual table objects (those that do not contain any MIB variables, such as the MIB-II “udpTable” object), are not mapped to GDMO MOCs. This means that the “udpEntry” MOC is bound directly below “udp”.

A fundamental requirement when mapping between management models is the ability to translate between a CMIS Distinguished Name (DN) and their equivalent SNMPv1 MIB Object Identifier (OID). The Relative Distinguished Name components of DNs consist of either an ASN.1 NULL, for single instanced managed object classes, or an ASN.1 SEQUENCE of the INDEX variables contained in the corresponding SMI OBJECT-TYPE template†.

The following is an example of a full DN:-

```
{{ systemId             = “uk.ac.ucl.cs.synapse” }
   ipId                 = NULL }
   ipNetToMediaEntryId = SEQUENCE{
                             ipNetToMediaIfIndex {2},
                             ipNetToMediaNetAddress{ 128.16.8.170}
                         }
}
```

Should we need to refresh the “ipNetToMediaType” attribute for the MOC defined by this DN, then we first obtain the IIMC defined OID for this OSI attribute, namely { iimcAutoObjAndAttr.1.3.6.1.2.1.4.22.1.2 }. The leading “iimcAutoObjAndAttr” sub-identifiers are removed, before appending on the SMI instance sub-identifiers, which for this case are “2.128.16.8.170”, yielding the correct SNMPv1 OID. Producing the OID for a single instanced MOC would have required the appending of the “.0” sub-identifier instead.

The reverse mapping from SMI OID to CMIS DN must be undertaken when translating Traps to Event-Reports. The correct system object is determined by checking the Trap source address and community strings that have been regis-

---

†. Foot note: For convenience our proxy also permits “dot” format strings to be used for RDN components.

tered for a given remote system. The hierarchical MIB information for the MIBs supported by this remote system is then traversed for all bar the instance sub-identifiers. The instance sub-identifiers are then converted to either a NULL or SEQUENCE syntax as in the example DN above.

In terms of the Telecommunications Management Network [M.3010] the information model produced by the IIMC translation rules is Q_x rather than Q_3. For example the GDMO produced for an ATM switch MIB would be semantically similar to, but not exactly the same, as that produced by the ITU, leading to a requirement for a Mediation Function to achieve a full Q_3 information model.

3.2 Protocol Mapping

When translating CMIS operations to SNMP requests it is immediately apparent that a one to many mapping exists, due to the presence of scoping and filter parameters. An efficient mapping requires the minimalisation of the number of the generated SNMP requests, especially since MIB traversal using the Get-Next primitive necessitates a wait state until the response to the current retrieval request is received before a further request can be emitted. The number of object instances listed in an SNMP request must be maximised. Attempts to achieve this may cause a “Too-Big” error response, leading to the generation of smaller requests. The managed objects that are present in the scoped MIT subtree must be refreshed in a top-down manner. The filter can then be applied to their state to permit selection of those instances that will have the current CMIS operation applied.

Since usage of the SNMPv1 Get-Next does not cause an error response if the specified object(s) are not instantiated at the SNMPv1 agent (unless the MIB has been fully traversed), it is utilised in preference to the Get primitive, unless a refresh is required for a single table entry object. When retrieving the variables of a single instanced object the corresponding SNMP Object Identifiers for each OSI attribute instance are determined without including a trailing “.0” sub-identifier, so that a Get-Next request can be utilised.

Retrieving all the entries within a table requires the generation of an initial request that specifies the OID for each table entry attribute, but excludes any trailing instance sub-identifiers, which yields the first table entry instance. The OIDs from this response are used as the parameters to the second Get-Next request, so as to retrieve the second table entry, and so on until the table has been fully traversed. An important optimisation can be achieved when refreshing existing tables since multiple Get-Nexts can be fired off in parallel, each starting from a different existing entry, e.g. from every fifth entry.

Providing that the GDMO Managed Object Class and Attribute templates indicate that the operation may be applied, CMIS M-Create, M-Delete and M-Set operations are all mapped to SNMP Set requests. To ensure that the semantics of the original CMIS requests are not infringed, M-Set requests that would cause the creation or deletion of a multi-instanced SNMP object are prevented.

SNMP Traps are mapped to an “internetAlarm” [IIMCIMIBTRANS] CMIS Event-Report. This notification contains the list of name/value pairs that are provided by the Trap’s list of variable-bindings. The proxy is also required to determine the Distinguished Name of the object instance that is associated with each variable-binding. The completed event report must then be forwarded to any manager that has previously requested such reports, and may also be logged locally.

4. The OSIMIS platform support

The OSI Management Information Service [OSIMIS,Pav93] provides a generic and extensible management platform. The support provided for the development of management agents is known as the Generic Managed System (GMS). The GMS is a fundamental aspect of OSIMIS as it provides the facilities to construct agents as it supports the rich facilities of the OSI management model e.g. scoping, filtering and multiple replies.

A primary advantage in selecting the OSIMIS platform is the provision of a large number of Systems Management Functions (SMFs). These include the Event-Report-Management [X734], Log-Control[X735], Object-Management [X730], Metric-Monitor [X739] and Summarization [X738] SMFs.

4.1 Support for event-driven system organisation

The Coordinator/Knowledge Source abstraction may be used to realise OSI's event driven infrastructure. An OSI agent can be implemented with one Coordinator object instance listening to all external communication endpoints (e.g. for incoming CMIP and Trap messages) or internal alarms and exercising a fully event-driven policy serialising all requests and ensuring that they will be taken to completion. Knowledge Sources are general application objects that may register external communication endpoints on which they expect to receive information and also request "wake-up" calls periodically to implement a polling regime.

4.2 Support for transparent abstract syntax handling

A general ASN.1 object may be defined with methods for encoding, decoding, freeing, comparing and parsing. These methods may be automatically generated by ASN.1 compilers and the manipulation of values through the management protocol at the syntax level can be automated.

4.3 Support for CMIS agent functionality

Managed objects in an OSI agent are autonomous entities that need only be concerned with the interactions with the associated real resource and not with details of access through the management protocol. The knowledge of the protocol and associated issues such as parameter checking, object addressing, access control, scoping, filtering and the handling of errors and linked replies can be encapsulated by an instance of an object known as the OSI agent. This is a specialised Knowledge Source that listens on management associations and performs the requested operation on the selected managed objects. Its interface to the managed objects may be asynchronous for loosely coupled resources (e.g. when remote SNMP agents are the resource), in which case it keeps state of the pending operations in order to eventually return a result or failure.

Managed object instances are represented internally as C++ object instances linked in a containment tree. Operations such as scoping and filtering are handled by the agent, which may request that an object refreshes its subordinates if these are transient objects or refresh the contents of an object before filtering.

4.4 Managed Object Support

The provision of a GDMO Compiler together with the high level of automation of checking, ensures that MIB implementers are presented with a very simple interface when constructing the real resource dependent Managed Object code. The real resource access may utilise one or more of the following methods -

- •) access upon an external management request
- •) a "cache-ahead" mechanism through periodic polling
- •) updates through asynchronous reports from the real resources

5. The ICM OSI/SNMPv1 Gateway

A fundamental design requirement for the gateway is to achieve seamless inter-operability between OSI management Operations Systems (OS) or Mediation Functions (MF) and SNMPv1 managed resources. An efficient mapping is essential given the fact that the gateway introduces an intermediate hop in the manager/agent communication path,

see figure(4).

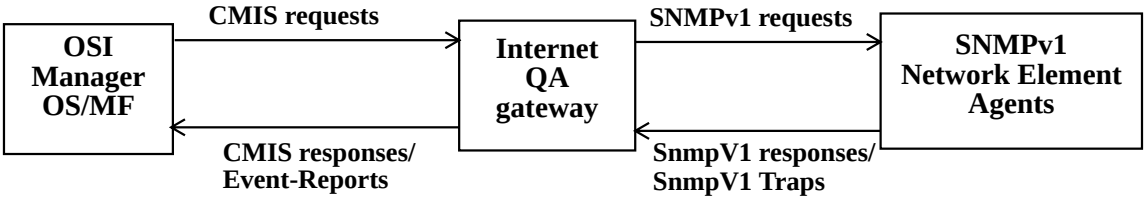

Fig. 4. Manager Agent communication paths

5.1 The Internet Q-Adapter gateway in operation

A fundamental aim of our research is to maximise the level of automation in generating a Q-Adapter that proxies for the desired remote SNMPv1 agents. Three stages are required, namely “translate”, “convert” and “run”, see figure (5).

• Translation involves the usage of VTT’s SMI to GDMO converter (“imibtool”) to produce an OSI representation of the MIB that is to be managed.

• Conversion concerns the generation of a simplified MIB description file. The OSIMIS GDMO compiler achieves this task.

• Run - the gateway reads in the simplified input file MIB description(s) and is ready to provide an “OSI view” of the SNMPv1 managed real resources.

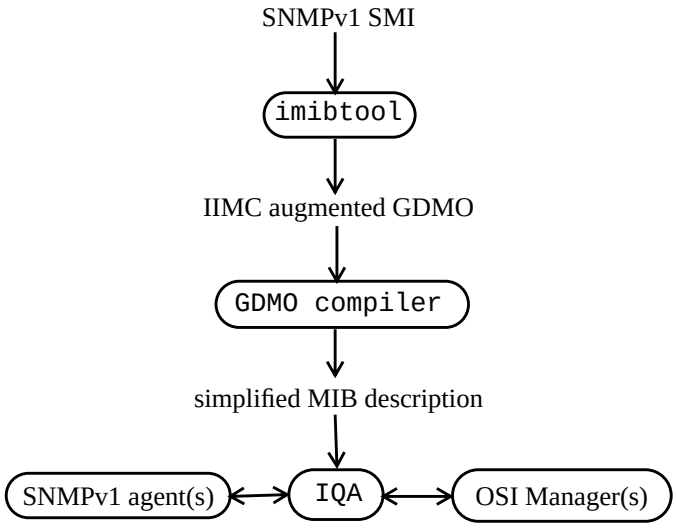

Fig. 5. The Internet Q-Adapter’s execution cycle.

5.2 Implementation Aspects.

The structural decomposition of the Internet Q-Adapter (IQA) gateway is shown in figure (6). We shall now endeavour to describe these components in some detail. At start-up an instance of each of the IQA `system`, `proxySystem`, `cmipsnmpProxy` and `remoteSystem` classes is instantiated. The `proxySystem` object represents the gateway’s local resources, whilst the `remoteSystem` object(s) represents the remote SNMPv1 systems.

The `cmipsnmpProxy` object reads the initial configuration requirements and creates a `cmipsnmpProxyAgent` object, a `remoteSystem` object and an `SnmpRMIBAgent` object for each remote SNMPv1 system. The `remoteSystem` objects can only be created successfully if a poll of the remote SNMPv1 agent receives a response. The `SnmpRMIBAgent` objects encapsulate an SNMPv1 protocol interface.

A tree of `SnmpImageMO` Managed Objects, corresponding to objects held at the remote SNMPv1 agent, will be built up below the respective `remoteSystem` objects in response to incoming Cmis requests. Managed Object Class descriptions are held within the `SnmpImageMOClassInfo` meta-class instances, which are themselves constructed into an MIT during the IQA's initialisation phase.

The `SnmpImageMO`s utilise the meta-class information to determine whether the corresponding SNMP SMI objects are single or multiply instanced. If multiply instanced then the INDEX attributes are indicated so that retrieved object values can be converted into Relative Distinguished Names in Cmis responses. Meta class information is also kept on which attributes are supported by the remote SNMPv1 agents so that the IQA does not request attributes that have been determined to be non-existent already.

Asynchronous Trap to Event-Report translation utilises the `proxyKS` object. This is an instance of an OSIMIS Knowledge Source and listens on the incoming SNMPv1 Trap UDP ports (e.g. 162), that have been configured for each remote SNMPv1 agent.

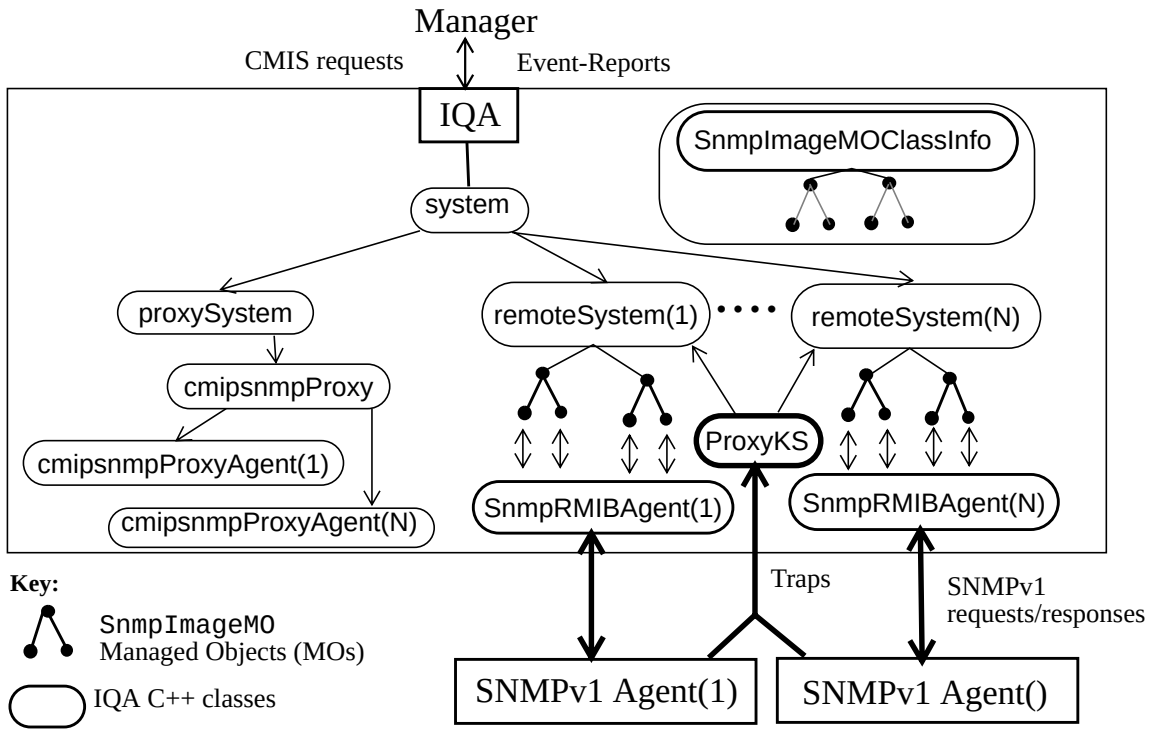

Fig. 6. IQA structural decomposition.

5.3 Performance Trials

Since the supply of management information is time critical, we have carried out a number of performance trials to confirm the validity of utilising the IQA for management purposes. We shall consider two comparative cases in the retrieval of the "ip" and "tcp" groups [RFC1213] from a remote ISODE SNMPv1 agent. The comparisons are against direct SNMPv1 retrieval using the "dump" command of ISODE's "snmpi" manager application, see figure (7).

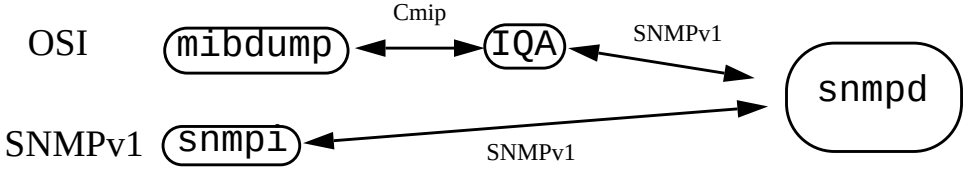

Fig. 7. Test components for the SNMPv1 vs OSI trials

Table 1: SNMPv1 vs OSI retrieval

Test Case	Manager	Test Runs	Minimum(s)	Maximum(s)	Mean(s)
IP	SNMPv1	21	2.539629	2.767339	2.601846
IP	OSI	25	1.975986	4.233499	2.284255
TCP	SNMPv1	35	1.698751	2.274993	1.830825
TCP	OSI	46	1.545147	3.652789	1.754175

Note 1: The OSI timings do not include the association setup and tear down components, which are around 0.2s and 0.02s respectively. Clearly these components can be amortized over far larger data transfers than have been considered in these trials.

Note 2: Those test runs which experienced a UDP response failure and thus the lock up of “snmpi” have been excluded from these results.

6. Management Scenario Evaluation

A centralised manager application is required to poll outlying agents to determine whether the values of certain MIB objects have exceeded some threshold. Let us now consider how this requirement can be achieved using both SNMPv1 and OSI managers. Usage of an Internet Q-Adapter means that the management protocol utilised at the real resource is not relevant.

6.1 Using an SNMPv1 Manager

If the manager polls too rapidly then it is in danger of taking a significant share of the transmission path’s capacity. Whilst if the manager does not poll frequently enough then there is every chance that the event that it was monitoring for, so as to permit it time to take evasive action, will be missed.

Even if the remote agents have a hard-wired enterprise specific Trap generation capability for certain thresholds then the unconfirmed UDP Trap may not even reach the manager despite there being only relatively minor network congestions levels. Also the manager can not remotely configure the agent to monitor a threshold that has not been hard-wired in.

The manager might be utilising a remote monitoring agent [RMON94] to achieve its goals - this is fine for transmission paths that offer a promiscuous mode of operation, but what of the ATM networks that are currently being introduced?

6.2 Using an OSI Manager

Localised polling can be remotely configured by the creation of metric monitor objects at the OSI agent or Internet Q-Adapter. Should the value or some weighted average of the values of a monitored attribute cross a defined threshold then an event report will be automatically be emitted without further management intervention. This idea is taken significantly further by the Summarization Function, which facilitates the summarization and statistical analysis of the data contained within the agent’s MIB, without the need to upload considerable amounts of data so that analysis can be undertaken at the managing application.

Since OSI supports both confirmed Event Reports and the designation of a backup event sink should the primary location fail, the OSI agent can be informed when its report has reached an appropriate manager, or can re-direct the report elsewhere if the first manager is off-line. Even if we take the worst case scenario, when the transmission path

itself goes down, then the generic OSI logging facilities still permit the management application to ascertain the agent state up to and beyond the failure, just as soon as the path is re-instated, so that diagnosis can be pursued.

7. Concluding Remarks.

Until the day arrives when a single Network Management architecture reaches 100% market penetration, there will always be a necessity to achieve meaningful inter-working between diverse management paradigms. The authors' research has attempted to meet this goal for the OSI and SNMPv1 models in a highly automated manner.

We have found that the OSI's powerful management functionality can be utilised successfully in enriching the SNMPv1 information model, by providing generic functions such as localised polling, remotely configurable event generation criteria and logging. The SNMP community wishes to retain the simplicity of their agents and by utilising generic OSI Q-Adapters **the agents can remain simple**, whilst the managers can be presented with a very powerful management architecture -the best of both worlds?

Acknowledgements

The research work detailed in this paper was produced under the auspices of the Integrated Communication Management (ICM) project, which is funded by the European Commission's Research into Advanced Communications in Europe (RACE) research program.

The authors would like to acknowledge the work of Jim Reilly of VTT (Finland) who achieved a significant level of automation with his SMI to GDMO MIB converter. James Cowan of UCL must be congratulated for developing the innovative GDMO compiler.

It would be remiss of us to sign off without re-emphasising our appreciation to the NMF, and Lee LaBarre, Lisa Phifer and April Chang in particular, for the excellence of the IIMC document package.

References

[IIMCMIBTRANS] Lee LaBarre (Editor), Forum 026 - *Translation of Internet MIBs to ISO/CCITT GDMO MIBs*, Issue 1.0, October 1993.

[IIMCSEC] Lee LaBarre (Editor), Forum 027 - *ISO/CCITT to Internet Management Security*, Issue 1.0, October 1993.

[IIMCPROXY] April Chang (Editor), Forum 028 - *ISO/CCITT to Internet Management Proxy*, Issue 1.0, October 1993.

[IIMCMIB-II] Lee LaBarre (Editor), Forum 029 - *Translation of Internet MIB-II (RFC1213) to ISO/CCITT GDMO MIB*, Issue 1.0, October 1993.

[IIMCOMIBTRANS] Owen Newman (Editor), Forum 030 - *Translation of ISO/CCITT MIBs to Internet MIBs*, Issue 1.0, October 1993.

[M3010]ITU M.3010, *Principles for a Telecommunications Management Network*, Working Party IV, Report 28, 12/91.

[OMG91] T*he Common Object Request Broker: Architecture and Specification*, OMG Draft 10 December 1991.

[OSIMIS] Pavlou, G., S. Bhatti and G. Knight, *OSIMIS User Manual Version 1.0 for System Version 3.5*, July 1994.

[Pav93] Pavlou G., *The OSIMIS TMN Platform: Support for Multiple Technology Integrated Management Systems*, Proceedings of the 1st RACE IS&N Conference, Paris, 11/93

[RFC1155] M.Rose, K.McCloghrie, Request for Comments: 1155, *Structure and Identification of Management Information for TCP/IP-based Internets*, May 1990.

[RFC1157] J.Case, M.Fedor, M.Schoffstall, J.Davin, Request for Comments:1157, *A Simple Management Protocol (SNMP)*, May 1990.

[RFC1212] M.Rose, K.McCloghrie (editors), Request for Comments:1212, *Concise MIB Definitions*, March 1991.

[RMON94] S.Waldbusser, Internet Draft, *Remote Network Monitoring Management Information Base*, June 1994.

[Rose91] M.Rose, The Simple Book, *An introduction to Management of TCP/IP-Based Internets*, Prentice-Hall, 1991.

[Sal84] J.H.Saltzer, D.P.Reed and D.D.Clark, *End-To-End Arguments in System Design*, ACM Transactions on Computer Systems, Vol.2, No. 4, November 1984.

[X500] ITU X.500, *Information Processing, Open Systems Interconnection - The Directory: Overview of Concepts, Models and Service*, 1988.

[X701] ITU X.701, *Information Technology - Open Systems Interconnection - Systems Management Overview*, 7/91

[X710] ITU X.710, I*nformation Technology - Open Systems Interconnection - Common Management Information Service Definition*, Version 2, 7/91

[X711] ITU X.711, *Information Technology - Open Systems Interconnection - Common Management Information Protocol Definition*, Version 2, 7/91

[X720] ITU X.720, *Information Technology - Structure of Management Information - Part 1: Management Information Model*, 8/91.

[X722] ITU X.722,*Information Technology - Structure of Management Information: Guidelines For The Definition of Managed Objects*, January 1992.

[X730] CCITT Recommendation X.730 (ISO 10164-1) *Information Technology - Open Systems Interconnection - Systems Management - Part 1: Object Management Function (for CCITT Applications)*, 10/91.

[X734] CCITT Recommendation X.734 (ISO 10164-5) *Information Technology - Open Systems Interconnection - Systems Management - Part 5: Event Report Management Function*, 8/91.

[X735] CCITT Recommendation X.735 (ISO 10164-6) *Information Technology - Open Systems Interconnection Systems Management - Part 6: Log Control Function*, 6/91

[X738] Revised Text of DIS 10164-13, *Information Technology - Open Systems Interconnection - Systems Management - Part 13: Summarization Function*, March 1993.

[X739] ITU Draft Recommendation X.739, *Information Technology - Open Systems Interconnection - Systems Management - Metric Objects And Attributes*, September 1993.