Customer Service Management: Towards a Management Information Base for an IP Connectivity Service

Michael Langer, Stefan Loidl, Michael Nerb
Leibniz Supercomputing Center
Barer Str. 21, 80333 Munich, Germany
Email: cnm-team@lrz.de

Abstract

Customer Service Management (CSM) offers a management interface between customer and service provider, which enables customers to individually monitor and control their subscribed service.

This paper presents an approach towards the definition, implementation and instantiation of a CSM management information base for an IP connectivity service (CSM-IP-MIB). The approach uses object-oriented techniques to model the CSM-IP-MIB in order to facilitate re-useability and specialization in real-life scenarios. The CSM-IP-MIB consists of four packages that reflect the relevant details of the IP connectivity service, including contracts, SLAs, topology, current and historical state of the IP connectivity, QoS parameters, QoS violations and problem management as well as authentication and access control to provide customer-specific views.

The CSM-IP-MIB is implemented using the CORBA technology, resulting in a distributed client/server application, which is being instantiated for the IP connectivity service of the German Research Network Organization.

1. Motivation

Customer Service Management (CSM) offers a service-oriented management interface between customer and service provider. CSM incorporates the paradigm shift towards service management, and the growing importance of customer care: It extends the provider’s service management towards the customers and enables customers to monitor and control up-to-date and meaningful information about service-specific QoS parameters [5]. In multi-level service hierarchies, CSM becomes indispensable, as otherwise players in the service hierarchy cannot determine the QoS parameters from the level below, but have to provide services to the level above according to the specified SLAs.

1.1. Scenario: CSM for an IP connectivity service

The relationship between customer and service provider can be seen from different viewpoints: From a contractual point of view, customer and service provider negotiate about a service level agreement (SLA) for the IP connectivity service, which contains specific QoS parameters for a particular customer. The service provider offers the IP connectivity service to the customer, which can use the service or set up value-added-services on top of it. For management purposes, the service provider uses network and system management facilities for managing and provisioning the IP connectivity service. The customer may use similar facilities for the same reasons.

The major drawback in this scenario is, that no management information between customer and service provider can be exchanged. Customers cannot monitor or control the QoS parameters, cannot detect QoS violations and cannot access problem management information electronically. To solve this problem, a suitable management information base (MIB) is needed to represent the IP connectivity service for the customer. This “CSM-IP-MIB” describes a formalized management interface between customer and service provider in respect to the IP connectivity service. The service provider acts in an agent role and implements the CSM-IP-MIB. The customer acts in a manager role and can browse the CSM-IP-MIB information. The CSM-IP-MIB reflects up-to-date, aggregated and meaningful information of the IP connectivity service, including SLAs, topology, current and historical state, QoS parameters, problem management and mechanisms for authentication and access control. Hence, the CSM-IP-MIB becomes indispensable for the customer to set up value-added-services (Email, WWW, . . . ) on top of the IP connectivity service.

Section 2 defines the required information using UML notation. Section 3 transforms the formalized representation into a distributed CORBA application that implements the CSM-IP-MIB. Section 4 instantiates the CORBA application in a real-life scenario.
1.2. Related Work

The management protocol SNMP [2] and its information model are state-of-the-art for managing IP networks and IP resources. However, the SNMP information model is not suitable for modeling complex services, and existing vendor-independent MIBs (e.g. MIB-2 and RMON) offer only part of the required information.

The need for exchanging management information between customer and service provider is called Customer Network Management (CNM) [4]. The main emphasis of CNM lies on providing information for special technologies such as SMDS [8], Frame Relay [9] or ATM [3].

These approaches are not satisfactory, as the standards mentioned above are device-oriented, not service-oriented. The CNM-MIBs are simple views on the technology-specific device MIBs. The information contained is neither condensed nor aggregated and does not reflect the SLAs of the IP connectivity service. Furthermore, these approaches lack a sophisticated access control and do not allow for individual, customer-specific views.

2. Defining the CSM-IP-MIB

The CSM-IP-MIB is an abstract interface definition that does not imply a particular implementation. In many cases, the term "MIB" is associated with SNMP-MIB. In our context, the CSM-IP-MIB is an object-oriented MIB that is defined using UML notation. It consists of four packages:

**IPSystem:** This package contains administrative information to enforce security and privacy of the information contained in the CSM-IP-MIB, and includes a mechanism for issuing asynchronous notifications.

**IPService:** This package contains contractual and organizational information that is necessary to handle the relationship between customer and service provider with respect to the IP connectivity service. It offers customers a convenient way to supervise the QoS parameters. Additionally, it features exception reports that list QoS violations, summary reports on the total flow of IP traffic, and detailed reports on individual flows of IP traffic. These reports could be used for negotiating discounts on major service failures or outages.

**IPNetwork:** This package models resources and the topology of an IP network that is necessary to implement an IP connectivity service. It reflects the current and historical configuration, fault, performance and state information of IP resources. The customer can use this information for traffic analysis, bandwidth management and network planning.

**IPTroubleAdministration:** This package enables customers to get involved in the problem management process. It offers functionality to exchange trouble reports between customer and service provider.

For the remainder of this section, these packages are described in detail. In accordance with other MIB specifications, it is not described, how the classes, methods and attributes can be implemented, as this depends on the implementation of the IP connectivity service (local matter).

2.1. IPSystem: Administrative information and security functionality

As depicted on the left-hand side of Figure 1, this package contains system information about the CSM-IP-MIB itself (class System). Before a customer can access information about the IP connectivity service, he has to authenticate to the CSM-IP-MIB. Class AuthorizationCenter provides the necessary methods. It issues an unique signature used for access control during the session. The signature is passed on as a parameter with each request to the CSM-IP-MIB. Thus, individual, customer specific “views” of the CSM-IP-MIB can be created, restricting amount and granularity of the information accessible by the customer. If a customer wants to receive asynchronous notifications automatically, he can register with NotificationCenter, which offers a “push-mechanism” that distributes information (e.g. updates, trouble reports) to all registered customers.

2.2. IPService: Modeling the relationship between customer and service provider

The right-hand side of Figure 1 formalizes the relationship between customer and service provider: Customer and ServiceProvider are derived from an abstract superclass Organization. Class User models employees that can access the CSM-IP-MIB. To provide customer-specific views on the CSM-IP-MIB, CustomerProfile grants certain access rights. The PrivilegeLevel attribute determines the amount and granularity of the accessible information.

Contract is an association class between customer and service provider. ContractNumber uniquely identifies a contract and can be used for many purposes, e.g. assigning accounting information to customers. Part of a contract are SLAs, which contain QoS parameters and a service access point (SAP) for a given customer. ServiceLevelAgreement describes the technical specification of the IP connectivity service, especially QoS parameters and escalation mechanisms. Any violation of these values results in an exception that can be queried by the customer using getQoSViolation.
**QoSParameter** describes the characteristics of the IP connectivity service. So far, no methods and attributes have been specified, as the entire QoS management process (definition, establishment, monitoring, maintenance, control, . . . ) and the mapping of QoS parameters to measurable values is currently being studied. Besides this, standardization activities regarding prioritization of traffic flows like *Classes of Service* (CoS) or *Type of Service* (ToS) have to be taken into consideration.

The SAP references an interface of a router (see section 2.3), where a customer is connected to the IP network. ServiceAccessPoint is represented by an own class that combines ServiceLevelAgreement, Customer and Interface. ServiceAccessPoint offers statistical information on the customer-specific flow of IP traffic (identified by the IP header source and destination field). It supports three types of statistics:

**getTopNStatistic:** The TopN statistic is used to prepare summary reports for a given customer $C_X$ that describe the destinations $D_1..D_N$ that top a list ordered by the exchanged IP packets and IP octets over the customers SAP in a given time interval (i) from $C_X$ as source to $D_1..D_N$ as destinations or (ii) from $D_1..D_N$ as sources to $C_X$ as destination.

**getXYTrafficRelationship:** This method returns statistical information on the traffic flow (IP packets, IP octets), customer $C_X$ has exchanged with another destination $D_Y$ over a given time interval.

**getSentReceivedTraffic:** This method provides the amount of traffic a given customer $C_X$ has sent over or received at his SAP over a given time interval. It offers summary reports on the total amount and detailed reports on the individual flow of traffic.

### 2.3. IPNetwork: Network configuration and current and historical network state

This package (see Figure 2) defines managed objects that represent the topology and the current and historical state of the IP network that provides the IP connectivity service. Common attributes of routers, subnetworks and locations (such as Name, current State or a list of associated TroubleReports) are modeled by the abstract superclass NetworkElement. Large IP networks are comprised of many network elements. To reduce the complexity and facilitate easy navigation, Locations are introduced to logically group network elements. Locations contain NetworkElements, and allow a hierarchical or-
ganization of IP networks. As each network element has a state, the state of a location is aggregated from all containing network elements. Backbone is a special location that serves as an entry point for navigating the topology.

As depicted on Figure 2, Router inherits attributes from NetworkElement and offers specific attributes such as Domainname, DeviceType, and a (read-only) CommunityString. Furthermore, current performance and fault information regarding received, forwarded or discarded datagrams reflects the current state of the router. This information has to be updated frequently. A Router is comprised of several Boards, and each Board is composed of several Interfaces. Depending on the hardware architecture of a router, one or several CPUs are available. The Utilization of a CPU is an indicator for the performance of the corresponding board or the entire router.

Subnetworks are connected by interfaces to routers. Interfaces are characterized by the attributes Name, State, Bandwidth and IPAddress. The specific attributes of subnetworks are configuration parameters like SubnetworkAddress and SubnetworkMask, and performance attributes like Availability. Further details on subnetworks depend on the underlying technical implementation (OSI layer 2). We subclassed PointToPointNetwork and MultipointNetwork. PointToPointNetwork represents IP subnetworks using dedicated circuits with a specific Bandwidth, Throughput, and Utilization for each direction. MultipointNetwork is for further study.

Besides this current “snapshot” of the IP network, methods for querying historical state and statistic information for each network element are provided:

**Historical state of network elements:** The `getStateHistory()` method returns the values $S_1, S_N$ for the historical state $[up, warning, down, unknown]$ of the specified network element in a given time interval. Additional information on the availability of the network elements are provided.
Historical availability of routers and subnetworks:
Method getAvailabilityHistory() returns the percentage $A$ for the historical availability of routers and subnetworks in a given time interval.

Historical utilization of CPUs: The getUtilHistory() method returns the values $U_1, U_N$ for the historical utilization [%] of CPUs in a given time.

Historical statistics of PointToPointNetworks: These methods return the values $V_1, V_N$ for the historical and aggregated $\min V_1, V_N / \avg V_1, V_N / \max V_1, V_N$ values of $\text{[Bandwidth][Utilization][Throughput]}$ for a particular instance of PointToPointNetwork in a given time interval.

2.4. IPTroubleAdministration: Enter and access trouble reports

The IPTroubleAdministration package (see Figure 3) facilitates interaction between customer and service provider in case of a problem.

![Figure 3. Package “IPTroubleAdministration”](image)

It defines a report format that allows the customer to track the trouble resolution process within the service provider. The report format is modeled using an abstract superclass called TroubleReport, and two derived classes called ITTroubleReport and ProviderTroubleReport. The former is used by customer and service provider to model detected troubles, whereas the latter is used by service providers to notify customers of scheduled down-times. TroubleReportCenter is used for creating, deleting and searching trouble reports.

The definition of this package is based on a methodology presented in [6] with respect to relevant standards and existing trouble ticket systems. Most of the attributes are derived from these sources. However, some attributes have been added or modified to suit the requirements of the IP connectivity service. For more details see [6].

3. Implementing the CSM-IP-MIB

As outlined in [5], the CORBA technology was used for implementing the CSM-IP-MIB. This section gives a brief overview over the steps necessary to transform the CSM-IP-MIB into a distributed CORBA application using IIOP for communication. The entire process should be automated to support rapid prototyping and minimize inconsistencies when changing or extending the CSM-IP-MIB.

Step 1 – Transferring UML to IDL: We used the CASE tool StP from Aonix to model the CSM-IP-MIB. StP offers a possibility to generate IDL code from the class diagrams. However, IDL support is pretty poor, and the generated IDL code had to be edited manually afterwards. Several scripts helped to automate this process.

Step 2 – Transferring IDL to implementation languages: We used the CORBA implementation NEO of Sun Microsystems to transform IDL code into C++ and Java. Generating stubs, skeletons, holder classes and get/set methods for all attributes could be automated without problems. The C++ code was used for implementing the server (i.e. the CSM-IP-MIB), the Java code was used to implement the client (i.e. the CSM-IP-MIB browser).

Step 3 – Adding functionality to client and server: For the implementation of the CSM-IP-MIB, the CORBA Naming Service, Event Service and Persistence Service were used. On the client-side, a suitable application had to be implemented to access (browse) the CSM-IP-MIB. Several options are feasible: The most “primitive” access is by means of native IIOP, e.g. for the integration with existing management platforms. A dial-in text-based client is suitable for narrow-bandwidth, outband-access to the CSM-IP-MIB in case of a major failure of the IP connectivity service itself. The standalone clients browses the CSM-IP-MIB graphically. There is a choice of platform dependent solutions (e.g. Windows, UNIX) and platform-independent solutions using the WWW/Java techniques.
4. Instantiating the CSM-IP-MIB

"Customer Service Management" is the main subject of a research project [1, 5] at the Leibniz Supercomputing Center (LRZ) in Germany. The project is supervised by the German Research Network Organization ("DFN-Verein"), who operates a nationwide network that connects the German universities and research organizations to the worldwide internet. The “Broadband-WiN” (B-WiN) is a virtual private network based on the ATM cross connect network of the Deutsche Telekom AG. The DFN-Verein offers various services based on the B-WiN, most notably the IP connectivity service.

Within the scope of this research project, the CSM-IP-MIB has been defined, implemented and instantiated. Currently, over 70 customers (250 users) can access the instantiation of the CSM-IP-MIB.

4.1. Customer authorization and access control

A graphical Java client was implemented to access the CSM-IP-MIB. Before a customer can access information about the IP connectivity service, he has to authenticate to the CSM-IP-MIB. Within the Java-GUI, this is done by entering a Customer/User/Password combination, which is validated by AuthorizationCenter. After successful authentication, a unique and customer-specific profile is assigned for this session. The profile restricts the information the customer can access. Furthermore, the Java-GUI automatically registers with NotificationCenter and thus receives update notifications (see section 4.3) and trouble information (see section 4.4).

4.2. Visualizing SLAs and QoS violations

The relationship between the DFN-Verein and the respective customer is manifested by means of a contract (an instance of Contract); it contains the SLAs (instance of ServiceLevelAgreement). Within the SLAs, QoS parameters are defined. In our scenario, QoS parameters concerning the IP connectivity service are very informal or even unspecified yet.

Additionally, for each customer, an instance of ServiceAccessPoint (SAP) is assigned to the router interface that connects the customer to the B-WiN. The configuration and state of this router interface is displayed within the Java-GUI. Based on the SAP, the Java-GUI offers statistical information on the customer-specific flow of IP traffic. A particular customer can access TopN IP traffic partners, IP traffic with a given destination and sent/received IP traffic; each statistic type offers aggregated values and IP traffic over the course of time. The former statistics are displayed as bar and pie charts, the latter as line charts. The Java-GUI does not offer reports about QoS violations on the individual SLAs yet, as there are no strict QoS parameter definitions in our scenario. Nevertheless, we are currently investigating, how to extract meaningful reports based on the available information.

4.3. Visualizing and navigating the topology, its current and historical state and QoS parameters

The Java-GUI visualizes the topology of the B-WiN by means of hierarchically organized maps (see Figure 4). Maps represent instances of Location. Starting from the IP backbone map (an instance of Backbone), users can navigate the maps, zooming in and out. Circular symbols in maps represent zoomable submaps and visualize instances of Location. Square symbols in maps represent IP routers and visualize instances of Router. Lines between submaps or IP routers represent IP connections and visualize instances of PointToPointNetwork. The current state of each IP router and IP connection is visualized by means of a colour scheme: Green indicates “ok”, yellow indicates “warning”, red indicates “down”, blue indicates “unknown”. Each IP connection is attributed with

![Figure 4. State and utilization of the backbone](image-url)
the current values of the identified QoS parameters (configured bandwidth \([\text{MBit/s}]\), throughput \([\text{MBit/s}]\) and utilization \(\%\)). The customer can switch between these QoS parameters at any time. All information is updated every five minutes. Maps aggregate the state of all containing elements and propagate this information. For example, the IP backbone map in Figure 4 offers the current state of the entire IP network at a single glance.

![Throughput of an IP connection](image)

**Figure 5. Throughput of an IP connection**

All information described above is stored permanently in the CSM-IP-MIB and can be retrieved by the customer at any time: The Java-GUI offers a convenient way to query historical statistics as described in section 2.3. The query is processed online by the CSM-IP-MIB, and the results are displayed in the Java-GUI using chart diagrams. A screenshot for a typical historical IP connection statistics can be found in Figure 5. The chart diagram features two graphs, each representing the directed throughput \([\text{MBit/s}]\) for the IP connection between the IP routers in Stuttgart and Munich on December 6th, 1998. This type of statistic has been used extensively by customers for traffic analysis, bandwidth management and network planning [7]. Similar statistics are available for historical state statistics of IP routers and IP connections.

### 4.4. Detecting, propagating and resolving troubles

The Java-GUI enables customers to participate in the problem management process. They can enter trouble reports and access a list of current (public and customer-specific) trouble reports. Each TroubleReport contains information about the current state, and the affected network elements. Customers get notified of state changes and/or upon problem resolution by means of notifications issued by NotificationCenter. If trouble reports have been assigned to network elements, the reports are automatically displayed when the customer points on the graphical symbol. This functionality supports quick detection, propagation and resolution of problems associated with the IP connectivity service.

### 5. Summary and Open Issues

This paper defines a management information base for an IP connectivity service (CSM-IP-MIB) that offers management information specifically tailored to the needs of the customer. The CSM-IP-MIB offers relevant details, including contracts, SLAs, topology, current and historical state of the IP connectivity, QoS parameters, QoS violations, problem management and mechanisms for authentication and access control.

The CSM-IP-MIB definition is implementation-independent. To prove for the applicability of our approach, an implementation of the CSM-IP-MIB using the CORBA technology is presented, along with an instantiation of the CSM-IP-MIB for a real-life scenario.

The process of integrating QoS parameters into the CSM-IP-MIB is an open issue: Modeling QoS parameters for the IP connectivity service (common definition and description) and implementing QoS management functions (such as establishment, monitoring, maintenance, control and reporting) is subject to further studies.

### References


