Defining a Trouble Report Format for the Seamless Integration of Problem Management into Customer Service Management

Michael Langer, Michael Nerb
Leibniz Supercomputing Center
Barer Str. 21, 80333 Munich, Germany
Email: {langer|nerb}@lrz.de

Abstract

Customer Service Management (CSM) offers customers a "logical view" onto the service management of a service provider. CSM describes a management interface between customer and service provider and enables a customer to access service-specific management information about the subscribed service. The information flow across the CSM interface is bidirectional, from customer to service provider and vice versa. This bidirectional flow of infomation is of particular importance in multi-layer service hierarchies, as otherwise players in this service hierarchy cannot determine the QoS parameters from the layer below, but have to provide services to the layer above according to specified SLAs. The information exchanged across the CSM interface covers all functional areas of management.

This paper focuses on the problem management task of Customer Service Management. This task describes the ability of a customer to actively participate in the trouble administration process of the service provider. The paper points out, why existing standards and related research work is not satisfactory to foster inter-domain problem management between customer and service provider. Hence, a methodology that results in a generic interface and a generic data structure called **CSM Trouble Report (CSM-TR)** *is proposed. The CSM Trouble Report can be exchanged across the CSM management interface regardless of the position in a given service hierarchy and the service in question. It enables a customer to report problems, query, view, modify and cancel his trouble reports. The customer gets notified of problems that affect his agreed service level, changes in the state of current trouble reports, and upon problem resolution.*

The object model of the CSM-TR is described using UML notation and is instantiated for a real-life scenario.

1. Introduction

There is an increasing need for service providers to establish an efficient and effective end-to-end service management for their services. However, when talking of service management these days, it usually reflects service processes (such as provisioning, ordering or operation) only and does not account for the management relationship between customer and service provider. As pointed out in [6], *Customer Service Management* adresses the management relationship, as it enables customers to access customer- and service-specific information about the subscribed services. CSM is provided by the service provider to the customer and covers all functional areas of management. In [6], mainly performance and *passive* problem management is addressed, that is, read-only access to the public trouble tickets issued by the service provider.

This paper extends Customer Service Management towards bidirectional, inter-domain problem management. In the context of this paper, this term is used to describe the ability of a customer to actively participate in the trouble administration process of the service provider. It includes the possibility of a customer to report problems, query, view, modify and cancel a problem report and includes notifications issued by the service provider, when the state of a current problem report has changed, or problems occur that affect the agreed service level of the customer.

To facilitate inter-domain problem management, a generic interface and a generic set of information has to be defined, which can be applied independent of the position in the service hierarchy and the service in question. Section 2 describes the main problems associated with trouble administration in IT environments and outlines the current state of the art. Section 3 addresses the critical issues by means of a methodology that results in a generic **CSM Trouble Report** (CSM-TR) object model. Section 4 discusses a scenario, in which the CSM-TR has been used.

2. Problem Description

The trouble administration process is part of the business relationship between customer and service provider. This process is mainly concerned with the identification and resolution of total or partial failures of the individual services. The mutual objective is to ensure, that the service level agreements (SLAs) are met, avoiding QoS violations or keeping them to a minimum.

2.1. Critical Issues

This situation typically occurs between any two layers in multi-level service hierarchies; at each of those points of interaction, information has to be exchanged quickly in case of a problem. An example of a typical service hierarchy can be found in Figure 1. Each of the roles uses a specific service from the lower layer and offers a specific service to the higher layer, where services can range from telecommunication services (e.g. SDH service) to application services (e.g. WWW, news or email service). Accordingly, specific management architectures and management systems are used for the management of these services.

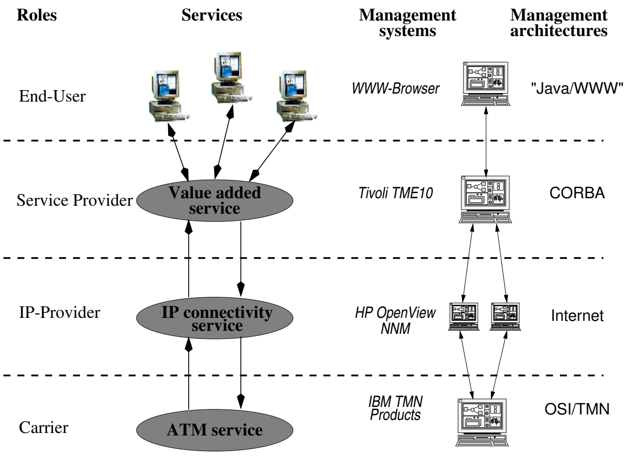

Figure 1. A typical service hierarchy

In order to provide inter-domain problem management independent of the position in the service hierarchy and the type of service, two major problems have to be resolved:

1. The management systems on each layer of the service hierarchy must be interoperable with management systems on the lower (and higher) level.

2. A common set of information has to be defined, which facilitates the exchange of problem management information independent of a particular service. Especially, the "generic trouble report" has to mediate between the "well-defined and well-formalized" definitions of telecommunication services (typically based on circuit switched networks) and the "application-level" IT services (typically based on packet switched networks), which are far less formalized.

In real-life scenarios, none of these two problems are completely resolved. Often, a hotline telephone number or a website acts as an interface, where customers can access basic information about problems or enter a problem report. It is not possible to track the progress of a problem resolution process, query the estimated duration of the problem, and there is no automated way of getting notified, when a trouble is cleared. Generally speaking, this procedure is inefficient, slow and unsatisfying in regard to the problem resolution process.

These problems call for a standardized interface and a standardized set of information that can be exchanged between the involved organizations. This paper focuses on the information aspects and defines a generic trouble report format that can be used in IT environments.

2.2. State of the Art

ITU-T X.790: The "Trouble Management Function for ITU-T Applications" [4] deals with the management of malfunctions ("troubles") in systems and communication networks. It lists trouble administration functions of fault management that facilitate interaction between customer and service provider in case of a trouble. 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 `Trouble Report (TR)` and two derived classes called `Telecommunication Trouble Report (TTR)` and `Provider Trouble Report (PTR)`. The former is used to model troubles detected by the customer (or service provider), whereas the latter is used by the service provider to notify customers of maintenance activities and scheduled down-times.

X.790 offers valuable concepts and information regarding the critical issues identified above. However, the approach used in X.790 accounts for several implications that limit the applicability in IT environments: X.790 is mainly aimed for use with telecommunication services, and the specified report format attributes and values reflect telecommunication equipment and services only[1]. The defined trouble administration attributes (e.g. `noDialTone, bellRingsAfterAnswer, receivesCallForWrongNumber`) are very

[1] X.790 recognizes that additional attributes may be necessary for other applications.

detailed, and some of them are not required in IT environments. X.790 uses GDMO/ASN.1 to model the report format and maps the functional units onto CMIS/CMIP. These techniques are not common in IT environments, as they are very complex, expensive and difficult to implement.

TeleManagement Forum TMF501 and TMF601: TMF addresses trouble administration in the context of a "Service Management Business Process Model". The "Customer To Service Provider Trouble Administration Business Agreement (TMF501)" [1] states the business objectives and driving problems that call for a business interface agreement between customer and service provider, describes the trouble administration process across this interface and lists technical and functional requirements that have to be met. The "Customer To Service Provider Trouble Administration Information Agreement (TMF601)" [2] defines a report format and workflows across the business interface. The TMF approach applies the X.790 concepts to the specific environment of the service provider and reduces the overall complexity: Related functional units are grouped together, some attributes are ommitted. TMF identifies two interfaces: The `CTT-SP` interface, that has to be implemented by the service provider, and the `CTT-Cust` interface, which has to be implemented by the customer. `CTT-SP` offers methods to create, modify, cancel, verify, view and track the status of trouble reports by the customer. `CTT-Cust` offers methods that can be invoked by the service provider to notify customers of problems. Most of the trouble report attributes are refinements of X.790 attributes, although some attributes have been omitted. Instead of GDMO, UML is used for modeling purposes.

TMF is a very useful source of information: Especially the definition of interfaces (`CTT-Cust` and `CTT-SP`) and the use of UML are suitable for use in IT environments. However, the report format is still aimed at telecommunication services and does not reflect IT environments properly. Furthermore, TMF uses different attribute syntax than X.790 to model similar or related semantics.

IETF RFC 1297: The "NOC Internal Integrated Trouble Ticket System Functional Specification Wishlist" [5] gives a brief overview over the purpose of a TTS, the general structure of trouble tickets and the technical integration of TTS with other NOC tools. RFC 1297 was meant to stimulate discussions, but apparently not very much has been happening since it's original publication in 1992. The document is informal and not very helpful regarding the critical issues identified above.

To sum up, the concepts used by X.790 are very useful for our purposes, but due to the complexity of the top-down approach and the focus on telecommunication environments, the model can not be applied onto IT services. TMF reduces the complexity and develops a good set of interfaces which can be used. But still, the relevant trouble information that has to be exchanged between customer and service provider in IT environments is not covered completely, as TMF focuses on the service provider view.

3. The CSM Trouble Report

The discussion in the previous section pointed out, that the trouble report formats of existing standards are not sufficient for the trouble administration process in IT environments. However, the identified information and the concepts used in the standards provide a useful baseline for the definition of a generic data structure applicable in IT environments. This generic data structure is called **CSM Trouble Report (CSM-TR)** and can be used to exchange problem information between customer and service provider in IT environments. Section 3.1 describes the methodology for the definition of the CSM-TR, section 3.2 details the CSM-TR object model using UML notation, and section 3.3 describes the CSM-TR in OMG-IDL.

3.1. Methodology

As depicted in Figure 2, we borrow from the information and concepts used in the standards analyzed in section 2.2: The top-down approach (as used in X.790) ensures, that all aspects of problem management are covered independent of a particular environment or scenario. The use of GDMO promotes re-useablility and specialization for different environments and scenarios. The TMF interface defintions `CTT-SP` and `CTT-Cust` abstract from a particular implementation of problem management and promote generic applicability and interoperability in distributed and heterogeneous environments. The use of UML provides a common understanding independent of a specific programming language. The bottom-up analysis of existing trouble ticket systems (TTS) ensures, that the CSM-TR can be easily mapped onto the trouble report formats of existing TTS, i.e. the CSM-TR can be implemented using these products. By using these concepts, we can assure, that the resulting CSM-TR conforms to the mentioned standards *and* is suitable to meet the requirements of IT environments.

A similar approach is taken to determine the CSM Trouble Report format. The information is drawn from three sources (see Figure 2): The attributes defined in X.790 act as a baseline for the modeling process. As discussed in section 2.2, these attributes reflect the trouble administration

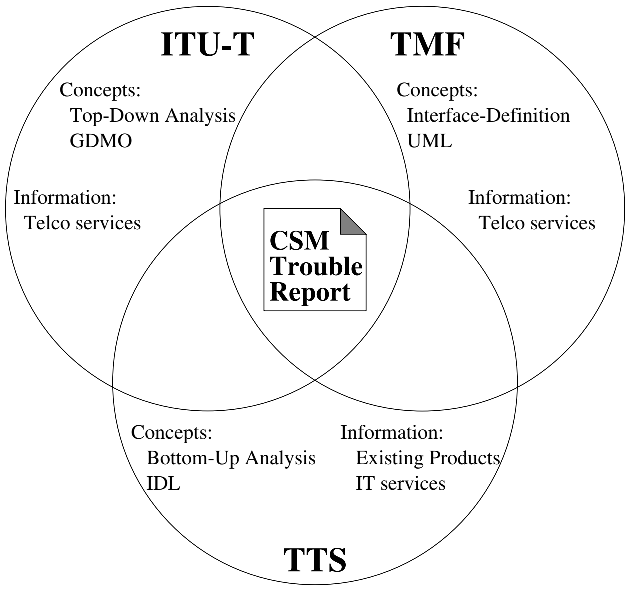

Figure 2. The methodology for defining the CSM Trouble Report

process in telecommunication environments. Due to the complexity of X.790 and the requirement, that the attributes can be easily mapped onto existing TTS, several attributes must be modified or omitted. The second source of information is TMF601. As described in section 2.2, the specified report format is aimed mainly at telecommunication services and does not consider the specific problems in IT environments. Nevertheless, TMF601 gives useful hints. The attributes of existing TTS are the third set of information. This set of attributes is of particular interest, as it reflects information that is of practical relevance in IT environments. We consider the TTS Remedy ARS and GNATS, which are used by the Leibniz Supercomputing Center (LRZ). From these sources, all those attributes are selected that are necessary for the exchange of problem management information in IT environments. Technically speaking, the three sets are intersected based on the similarities in the semantic meaning of the contained attributes. The resulting set of the intersection process contains a small number of common, generic and reasonable attributes, which are relevant in IT environments and still defined according to the standards. This set forms the CSM Trouble Report.

3.2. The CSM-TR Object Model

After the definition of the CSM-TR format, the semantic of the identified attributes has to be determined. For a detailed discussion, the attributes are grouped into one of three different subsets:

Common Attributes have been adopted unchanged from X.790/TMF601. The semantic meaning (behaviour) of these attributes is not discussed here. It can be found in [2, 4].

Modified Attributes are based on X.790/TMF601 but have been modified syntactically or semantically. These attributes will be explained by describing the semantic meaning (behavior), followed by a short discussion on differences to X.790 and TMF601.

Other Attributes have mainly been adopted from existing TTS. The rationale for these attributes will be explained.

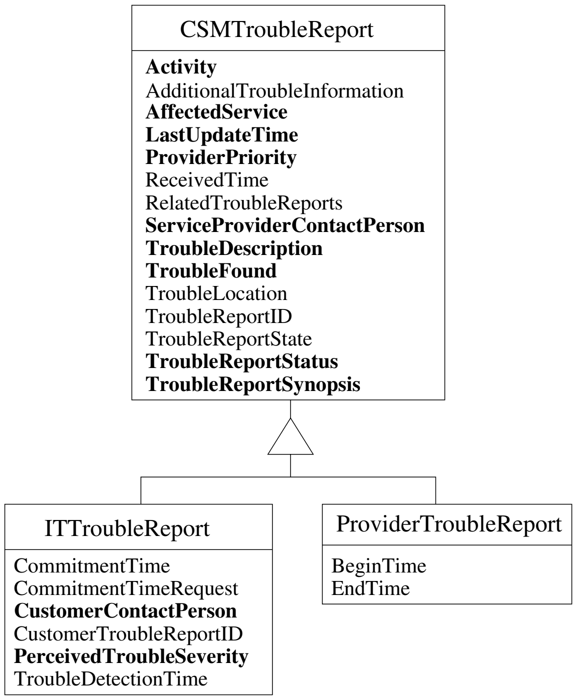

Figure 3. The CSM-TR Object Model

In Figure 3, the *Modified* and *Other Attributes* have been highlightened for better readability. The CSM-TR is modeled in accordance with X.790, using an abstract superclass called `CSMTroubleReport` (in X.790 parlance: `TroubleReport`), and two derived classes called `ITTroubleReport` and `ProviderTroubleReport` (in X.790 parlance: `TelecommunicationTroubleReport` and `ProviderTroubleReport`).

3.2.1 Superclass CSMTroubleReport

`CSMTroubleReport` is used as an abstract superclass for modeling generic troubles in IT environments. `CSMTroubleReport` contains the following attributes:

Common Attributes
The following attributes have been adopted from X.790/TMF601: AdditionalTroubleInformation (X.790: AdditionalTroubleInfoList), ReceivedTime, RelatedTroubleReports (X.790: RelatedTroubleReportList), TroubleLocation, TroubleReportID and TroubleReportState.

Modified Attributes
These attributes have been modified to address the critical issues identified in section 2.1:

Activity: This attribute describes (i) the specific activity performed as plain text, (ii) who performed them, (iii) the date and time when they where performed and (iv) the resulting new state and status. The corresponding X.790/TMF601 attribute (RepairActivityList) is intended to provide supporting details only on repair activities but we focus on generic activities, that means each performed activity during the trouble resolution process should be represented by a list entry. In contrast to X.790 and TMF601 the activity is not described by an activity code. In X.790 there are only a few codes defined, which are insufficient for our purpose of describing all possible activities in IT environments. Instead we represent this information by a combination of the textual activity description and the new state and status (in OMG-IDL notation):

```
struct Activity {
  Person   operator;
  DateTime entryTime;
  Text     description;
  State    newState;
  Status   newStatus;
};
typedef sequence<Activity> ActivityList;
```

AffectedService: This identifies the service for which the trouble report is created. In contrast to telecommunication environments there is no formal service description available for most services in IT environments. Hence, the affected service has to be described textual. A possible corresponding X.790 attribute might be UnavailableServicePointer. Unlike our approach, the X.790 UnavailableServicePointer attribute is part of the Provider Trouble Report. The corresponding TMF601 attribute is called TroubleObject.

LastUpdateTime: The LastUpdateTime attribute identifies the date and time of the last update (Activity or TroubleReportStatus) made to the trouble report by either the service provider or the customer. It combines the two X.790/TMF601 attributes LastUpdateTime and TRStatusTime.

ServiceProviderContactPerson: This attribute identifies a person in the service providers organization who can be contacted regarding the reported trouble. It includes both the identity of the person and the organization he is with. The corresponding X.790 attribute is called AgentContactPerson, the TMF601 counterpart is called SPRoleAssignment.

TroubleDescription: This attribute gives a description of the reported trouble. There is no matching X.790 attribute; the X.790 TroubleType attribute might have a similar semantic meaning. But in contrast to telecommunication environments it is almost impossible to describe all possible problems in IT environments by means of a single error code (as the data type of TroubleType), due to missing formal service descriptions. Because of this fact the data type for TroubleDescription is plain text. The TMF601 equivalent is called TroubleReportDescription.

TroubleFound: The TroubleFound attribute identifies the resolved problem. The data type for TroubleFound is plain text in contrast to the corresponding attribute in X.790. X.790 defines an integer value for most of the possible problems in a telecommunications environment. For an IT environment this is much harder to do and furthermore a single error code describes only insufficiently most of the problems.

TroubleReportStatus: The TroubleReportStatus attribute is set by the service provider. It describes the state of the trouble report in a finer granularity. This attribute associates a value to each state of a trouble report, which gives a short description of the currently executed activity at the service provider, the reason for a delay of the processing of the trouble report or the last carried out activity. A detailed description of a trouble report status may be deduced from the last item in the Activities list. The data type for TroubleReportStatus has been deduced from the corresponding data types in X.790 and TMF601. Some values have been merged in one value and a few have been omitted in order to reduce complexity. But essentially the data type is very similar: One status value is only allowed in a special trouble report state.

```
enum Status {
  //state open
  testing, dispatched, pendingTest,
  pendingDispatch, repairing, requestedRepair,
  referOther, failureFound, backOrder,
  //state deferred
  noAccessOther, delayedMaintenance,
  //state cleared
  temporaryOk, customerNotAdvised, customerAdvised,
  //state closed
  closedOut, customerVerified, customerDenied,
  canceled,
  // if none of the above is appropriate
  other
};
```

Other Attributes
The following attributes have been added to the CSM-TR due to requirements of IT environments:

ProviderPriority: This attribute defines the urgency with which the service provider works at problem resolution. In addition the Service Provider is able to define the escalation level of the trouble resolution process in conjunction with specified, local policies. The values for the `Priority` data type have been taken from GNATS [8] plus an additional `undefined` value.

```
enum Priority { high, medium, low, undefined };
```

This definition differs from the X.790/TMF601 PreferredPriority attribute. The PreferredPriority attribute defines the urgency with which the customer requires resolution of the problem (see PerceivedTroubleSeverity in the next subsection).

TroubleReportSynopsis: A short textual description of the problem.

3.2.2 IT Trouble Report (ITTR)

`ITTroubleReport` is derived from `CSMTrouble-Report`. It is used to model troubles that are detected by the service provider or the customer. `ITTroubleReport` contains the following attributes:

Common Attributes
The following attributes have been adopted from X.790/TMF601: CommitmentTime, CommitmentTimeRequest, CustomerTroubleReportID and TroubleDetectionTime.

Modified Attributes
These attributes have been modified to address the critical issues identified in section 2.1:

CustomerContactPerson: This attribute identifies a person in the customer organization that can be contacted regarding the reported trouble. The corresponding X.790 attribute is called ManagerContactPerson, the TMF601 counterpart is called CustRoleAssignment.

PerceivedTroubleSeverity: This attribute allows the customer to indicate the effect of the reported trouble from his point of view. This attribute is common to X.790 and TMF601. Nevertheless, the values for the Severity data type have been taken from GNATS [8]. The differences of the values to the corresponding X.790 values are only syntactical, but the value `backInService` in X.790 has been omitted.

No *Other Attributes* have been introduced to the IT Trouble Report.

3.2.3 Provider Trouble Report (PTR)

`ProviderTroubleReport` is derived from `CSM-TroubleReport`. It is used by the service provider to inform customers of scheduled maintenance activities. There are no modified or other attributes in the Provider Trouble Report. The attributes BeginTime and EndTime have been adopted from X.790.

3.3. IDL Definition of the CSM-TR Object Model

To foster the implementation of inter-domain problem management in distributed and heterogeneous environments, the UML definition of the CSM-TR format is mapped onto OMG IDL. The IDL definition is a baseline for an implementation using the OMG CORBA technology.

```
interface CSMTroubleReport {
  attribute ActivityList  Activities;
  attribute TextList      AdditionalTroubleInformation;
  attribute DateTime      LastUpdateTime;
  attribute Priority      ProviderPriority;
  attribute DateTime      ReceivedTime;
  attribute IDList        RelatedTroubleReports;
  attribute Person        ServiceProviderContactPerson;
  attribute Text          TroubleDescription;
  attribute Location      TroubleLocation;
  attribute Text          AffectedService;
  attribute ID            TroubleReportID;
  attribute State         TroubleReportState;
  attribute Status        TroubleReportStatus;
  attribute Text          TroubleReportSynopsis;
  attribute Text          TroubleFound;
};
interface ITTroubleReport  : CSMTroubleReport {
  attribute ID            CustomerTroubleReportID;
  attribute DateTime      CommitmentTime;
  attribute DateTime      CommitmentTimeRequest;
  attribute Person        CustomerContactPerson;
  attribute Severity      PerceivedTroubleSeverity;
  attribute DateTime      TroubleDetectionTime;
};
interface ProviderTroubleReport : CSMTroubleReport {
  attribute DateTime      BeginTime;
  attribute DateTime      EndTime;
};
```

4. Instantiation of the CSM-TR

"Customer Service Management" is the main subject of a research project [6, 7] at the Leibniz Supercomputing Center (LRZ) in Germany. The project is supervised by the *German Research Network Organization* ("DFN-Verein") and funded by the Federal Ministry for Education, Science, Research and Technology. Within the scope of this research project, the CSM-TR object model has been developed.

The DFN-Verein offers various services to his customers (e.g. LRZ), most notably an IP connectivity service (see Figure 4). These services are provided using a nationwide network that connects the German universites and research organizations to the worldwide internet. The "Broadband-WiN" (B-WiN) is a virtual private network (VPN) based on

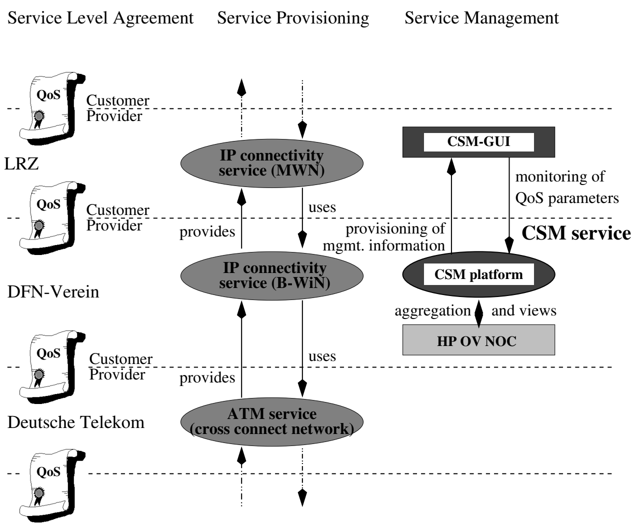

Figure 4. Service hierarchy in the scenario

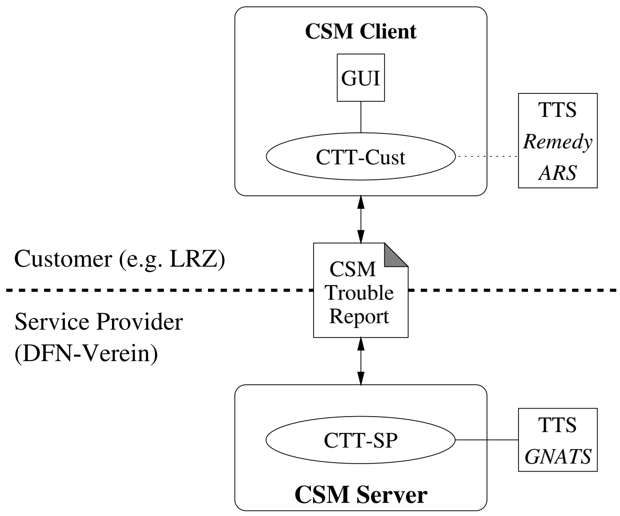

Figure 5. CSM-TR exchange using the CSM platform

the ATM cross connect network of the Deutsche Telekom AG. Only this part of the service hierarchy is depicted in Figure 4. As a matter of fact, this is a simplification of the real situation, as the service hierarchy can be extended towards the lower end (the ATM cross connect network of the Deutsche Telekom AG is based on an underlying SDH network also owned by the Deutsche Telekom AG) and towards the higher end (the LRZ offers the IP connectivity service to the Munich Universities, which, in turn, offer the service to their end-users).

In the first phase of the research project, the CSM Trouble Report had to be implemented between the DFN-Verein and his customers (e.g. LRZ). At this point we took advantage of an already installed CSM platform, which offers CSM information and functionality for the IP connectivity service of the DFN-Verein. The CSM platform is implemented using CORBA/Java/WWW techniques (see [6] for more details).

As depicted in Figure 5, the CSM server implements the `CTT-SP` interface, which offers a possibility to create, query, view, modify and cancel a problem report. The `CTT-SP` implementation provides a mapping of the CSM trouble report format onto the attributes supported by the trouble ticket system GNATS, which is used by the technical provider of the IP connectivity service (DFN Network Operations Center, DFN-NOC).

The CSM client implements the `CTT-Cust` interface and offers a possibility to receive notifications from the DFN-NOC. In the first step, a Graphical User Interface (GUI) (integrated in the CSM client) is used to access trouble information. In the second step, the `CTT-Cust` implementation is going to be extended to integrate the TTS used by the customers of the IP connectivity service. Therefore, the `CTT-Cust` implementation must map the CSM-TR attributes onto the attributes provided by the TTS used by customers. The LRZ, for example, uses Remedy ARS.

5 Conclusions and Further Work

To facilitate inter-domain problem management for IT services in multi-level service hierarchies, a generic interface and generic trouble report format has to be defined. Standards regarding problem management offer useful concepts, but are not fully suitable for IT environments. Thus, we use the concepts of these standards as a baseline and define a CSM Trouble Report (CSM-TR) for IT services, which can be exchanged over standardized interfaces. The interfaces and the CSM-TR have been implemented as part of an already existing CSM platform.

So far, the exchange of problem management information is limited to the interface between the DFN-Verein and the customers of the IP connectivity service (see Figures 4 and 5). We are going to apply our approach onto the other levels of the service hierarchy, especially the interface between DFN-Verein and Deutsche Telekom AG. However, this is a very difficult goal to achieve, as various organizational and administrative domains have to cooperate.

Customer Service Management covers all functional areas of management. So far, we have focused on *perfor-*

mance and *problem management*. Currently, *accounting mangement* functionality is being added to the CSM platform, which offers customer-specific accounting information about the IP connectivity service [7] . Regarding *configuration* and *change management*, an automated service ordering process for IT services (e.g. ATM-PVC service) is being considered for the integration in the CSM platform.

Acknowledgments

The authors wish to thank the members of the Munich Network Management (MNM) Team for helpful discussions and valuable comments on previous versions of the paper. The MNM Team, directed by Prof. Dr. Heinz-Gerd Hegering, is a group of researchers of the Munich Universities and the Leibniz Supercomputing Center of the Bavarian Academy of Sciences. Its webserver is located at `http://wwwmnmteam.informatik.uni-muenchen.de.`

References

[1] Network Management Forum. Customer to Service Provider Trouble Administration Business Agreement, Issue 1.0 (NMF 501), Aug. 1996.

[2] Network Management Forum. Customer to Service Provider Trouble Administration Information Agreement, Issue 1.0 (NMF 601), Mar. 1997.

[3] *Proceedings of the 4th IEEE Symposium on Computers and Communications (ISCC'99), Sharm El Sheikh, Red Sea, Egypt*, July 1999.

[4] ITU-T. (X.790) Data Networks and Open System Communications — OSI Management: Trouble Management Function for ITU-T Applications, Nov. 1995.

[5] D. Johnson. RFC1297: NOC Internal Integrated Trouble Ticket System Functional Specification Wishlist ("NOC TT Requirements"), Jan. 1992.

[6] M. Langer, S. Loidl, and M. Nerb. Customer Service Management: A more Transparent View to Your Subscribed Services. In Sethi [9].

[7] M. Langer, S. Loidl, and M. Nerb. Customer Service Management: Towards a Management Information Base For an IP Connectivity Service. In ISCC 99 [3].

[8] J. Osier and K. B. *Keeping Track - Managing Messages With GNATS*, Jan. 1996. The GNU Problem Report Management System, Version 3.104.

[9] A. S. Sethi, editor. *Proceedings of the 9th Annual IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM'98), Newark*, Oct. 1998.