# Assured Service Quality by Improved Fault Management

## Service-Oriented Event Correlation

Andreas Hanemann
Munich Network Management Team
Leibniz Supercomputing Center
Barer Str. 21, D-80333
Munich, Germany
hanemann@lrz.de

Martin Sailer
Munich Network Management Team
University of Munich (LMU)
Oettingenstr. 67, D-80538
Munich, Germany
sailer@nm.ifi.lmu.de

David Schmitz
Munich Network Management Team
Leibniz Supercomputing Center
Barer Str. 21, D-80333
Munich, Germany
schmitz@lrz.de

## ABSTRACT

The paradigm shift from device-oriented to service-oriented management has also implications to the area of event correlation. Today's event correlation mainly addresses the correlation of events as reported from management tools. However, a correlation of user trouble reports concerning services should also be performed. This is necessary to improve the resolution time and to reduce the effort for keeping the service agreements. We refer to such a type of correlation as service-oriented event correlation. The necessity to use this kind of event correlation is motivated in the paper.

To introduce service-oriented event correlation for an IT service provider, an appropriate modeling of the correlation workflow and of the information is necessary. Therefore, we examine the process management frameworks IT Infrastructure Library (ITIL) and enhanced Telecom Operations Map (eTOM) for their contribution to the workflow modeling in this area. The different kinds of dependencies that we find in our general scenario are then used to develop a workflow for the service-oriented event correlation. The MNM Service Model, which is a generic model for IT service management proposed by the Munich Network Management (MNM) Team, is used to derive an appropriate information modeling. An example scenario, the Web Hosting Service of the Leibniz Supercomputing Center (LRZ), is used to demonstrate the application of service-oriented event correlation.

## Categories and Subject Descriptors

C.2.4 [**Computer Systems Organization**]: Computer-Communication Networks—*Distributed Applications*

*ICSOC'04,* November 15–19, 2004, New York, New York, USA.
Copyright 2004 ACM 1-58113-871-7/04/0011 ...$5.00.

## General Terms

Management, Performance, Reliability

## Keywords

Event Correlation, Fault Management, Service Management, Service Level Agreements

## 1. INTRODUCTION

In huge networks a single fault can cause a burst of failure events. To handle the flood of events and to find the root cause of a fault, event correlation approaches like rule-based reasoning, case-based reasoning or the codebook approach have been developed. The main idea of correlation is to condense and structure events to retrieve meaningful information. Until now, these approaches address primarily the correlation of events as reported from management tools or devices. Therefore, we call them *device-oriented.*

In this paper we define a *service* as a set of *functions* which are offered by a *provider* to a *customer* at a *customer provider interface.* The definition of a "service" is therefore more general than the definition of a "Web Service", but a "Web Service" is included in this "service" definition. As a consequence, the results are applicable for Web Services as well as for other kinds of services. A *service level agreement (SLA)* is defined as a contract between customer and provider about guaranteed service performance.

As in today's IT environments the offering of such services with an agreed service quality becomes more and more important, this change also affects the event correlation. It has become a necessity for providers to offer such guarantees for a differentiation from other providers. To avoid SLA violations it is especially important for service providers to identify the root cause of a fault in a very short time or even act proactively. The latter refers to the case of recognizing the influence of a device breakdown on the offered services. As in this scenario the knowledge about services and their SLAs is used we call it *service-oriented.* It can be addressed from two directions.

**Top-down perspective:** Several customers report a problem in a certain time interval. Are these trouble reports correlated? How to identify a resource as being the problem's root cause?

**Bottom-up perspective:** A device (e.g., router, server) breaks down. Which services, and especially which customers, are affected by this fault?

The rest of the paper is organized as follows. In Section 2 we describe how event correlation is performed today and present a selection of the state-of-the-art event correlation techniques. Section 3 describes the motivation for service-oriented event correlation and its benefits. After having motivated the need for such type of correlation we use two well-known IT service management models to gain requirements for an appropriate workflow modeling and present our proposal for it (see Section 4). In Section 5 we present our information modeling which is derived from the MNM Service Model. An application of the approach for a web hosting scenario is performed in Section 6. The last section concludes the paper and presents future work.

## 2. TODAY'S EVENT CORRELATION TECHNIQUES

In [11] the task of event correlation is defined as "a conceptual interpretation procedure in the sense that a new meaning is assigned to a set of events that happen in a certain time interval". We can distinguish between three aspects for event correlation.

**Functional aspect:** The correlation focuses on functions which are provided by each network element. It is also regarded which other functions are used to provide a specific function.

**Topology aspect:** The correlation takes into account how the network elements are connected to each other and how they interact.

**Time aspect:** When explicitly regarding time constraints, a start and end time has to be defined for each event. The correlation can use time relationships between the events to perform the correlation. This aspect is only mentioned in some papers [11], but it has to be treated in an event correlation system.

In the event correlation it is also important to distinguish between the knowledge acquisition/representation and the correlation algorithm. Examples of approaches to knowledge acquisition/representation are Gruschke's dependency graphs [6] and Ensel's dependency detection by neural networks [3]. It is also possible to find the dependencies by analyzing interactions [7]. In addition, there is an approach to manage service dependencies with XML and to define a resource description framework [4].

To get an overview about device-oriented event correlation a selection of several event correlation techniques being used for this kind of correlation is presented.

**Model-based reasoning:** *Model-based reasoning (MBR)* [15, 10, 20] represents a system by modeling each of its components. A model can either represent a physical entity or a logical entity (e.g., LAN, WAN, domain, service, business process). The models for physical entities are called *functional model*, while the models for all logical entities are called *logical model*. A description of each model contains three categories of information: attributes, relations to other models, and behavior. The event correlation is a result of the collaboration among models.

As all components of a network are represented with their behavior in the model, it is possible to perform simulations to predict how the whole network will behave.

A comparison in [20] showed that a large MBR system is not in all cases easy to maintain. It can be difficult to appropriately model the behavior for all components and their interactions correctly and completely.

An example system for MBR is NetExpert[16] from OSI which is a hybrid MBR/RBR system (in 2000 OSI was acquired by Agilent Technologies).

**Rule-based reasoning:** *Rule-based reasoning (RBR)* [15, 10] uses a set of rules for event correlation. The rules have the form *conclusion* **if** *condition*. The condition uses received events and information about the system, while the conclusion contains actions which can either lead to system changes or use system parameters to choose the next rule.

An advantage of the approach is that the rules are more or less human-readable and therefore their effect is intuitive. The correlation has proved to be very fast in practice by using the RETE algorithm.

In the literature [20, 1] it is claimed that RBR systems are classified as relatively inflexible. Frequent changes in the modeled IT environment would lead to many rule updates. These changes would have to be performed by experts as no automation has currently been established. In some systems information about the network topology which is needed for the event correlation is not used explicitly, but is encoded into the rules. This intransparent usage would make rule updates for topology changes quite difficult. The *system brittleness* would also be a problem for RBR systems. It means that the system fails if an unknown case occurs, because the case cannot be mapped onto similar cases. The output of RBR systems would also be difficult to predict, because of unforeseen rule interactions in a large rule set. According to [15] an RBR system is only appropriate if the domain for which it is used is small, nonchanging, and well understood.

The GTE IMPACT system [11] is an example of a rule-based system. It also uses MBR (GTE has merged with Bell Atlantic in 1998 and is now called Verizon [19]).

**Codebook approach:** The *codebook approach* [12, 21] has similarities to RBR, but takes a further step and encodes the rules into a correlation matrix.

The approach starts using a dependency graph with two kinds of nodes for the modeling. The first kind of nodes are the faults (denoted as problems in the cited papers) which have to be detected, while the second kind of nodes are observable events (symptoms in the papers) which are caused by the faults or other events. The dependencies between the nodes are denoted as directed edges. It is possible to choose weights for the edges, e.g., a weight for the probability that fault/event A causes event B. Another possible weighting could indicate time dependencies. There are several possibilities to reduce the initial graph. If, e.g., a cyclic dependency of events exists and there are no probabilities for the cycles' edges, all events can be treated as one event.

After a final input graph is chosen, the graph is transformed into a correlation matrix where the columns contain the faults and the rows contain the events. If there is a dependency in the graph, the weight of the corresponding edge is put into the according matrix cell. In case no weights are used, the matrix cells get the values 1 for dependency and 0 otherwise. Afterwards, a simplification can be done, where events which do not help to discriminate faults are deleted. There is a trade-off between the minimization of the matrix and the robustness of the results. If the matrix is minimized as much as possible, some faults can only be distinguished by a single event. If this event cannot be reliably detected, the event correlation system cannot discriminate between the two faults. A measure how many event observation errors can be compensated by the system is the Hamming distance. The number of rows (events) that can be deleted from the matrix can differ very much depending on the relationships [15].

The codebook approach has the advantage that it uses long-term experience with graphs and coding. This experience is used to minimize the dependency graph and to select an optimal group of events with respect to processing time and robustness against noise.

A disadvantage of the approach could be that similar to RBR frequent changes in the environment make it necessary to frequently edit the input graph.

SMARTS InCharge [12, 17] is an example of such a correlation system.

**Case-based reasoning:** In contrast to other approaches *case-based reasoning (CBR)* [14, 15] systems do not use any knowledge about the system structure. The knowledge base saves cases with their values for system parameters and successful recovery actions for these cases. The recovery actions are not performed by the CBR system in the first place, but in most cases by a human operator.

If a new case appears, the CBR system compares the current system parameters with the system parameters in prior cases and tries to find a similar one. To identify such a match it has to be defined for which parameters the cases can differ or have to be the same. If a match is found, a learned action can be performed automatically or the operator can be informed with a recovery proposal.

An advantage of this approach is that the ability to learn is an integral part of it which is important for rapid changing environments.

There are also difficulties when applying the approach [15]. The fields which are used to find a similar case and their importance have to be defined appropriately. If there is a match with a similar case, an adaptation of the previous solution to the current one has to be found.

An example system for CBR is SpectroRx from Cabletron Systems. The part of Cabletron that developed SpectroRx became an independent software company in 2002 and is now called Aprisma Management Technologies [2].

In this section four event correlation approaches were presented which have evolved into commercial event correlation systems. The correlation approaches have different focuses. MBR mainly deals with the knowledge acquisition and representation, while RBR and the codebook approach propose a correlation algorithm. The focus of CBR is its ability to learn from prior cases.

## 3. MOTIVATION OF SERVICE-ORIENTED EVENT CORRELATION

Fig. 1 shows a general service scenario upon which we will discuss the importance of a service-oriented correlation. Several services like SSH, a web hosting service, or a video conference service are offered by a provider to its customers at the customer provider interface. A customer can allow several users to use a subscribed service. The quality and cost issues of the subscribed services between a customer and a provider are agreed in SLAs. On the provider side the services use subservices for their provisioning. In case of the services mentioned above such subservices are DNS, proxy service, and IP service. Both services and subservices depend on resources upon which they are provisioned. As displayed in the figure a service can depend on more than one resource and a resource can be used by one or more services.

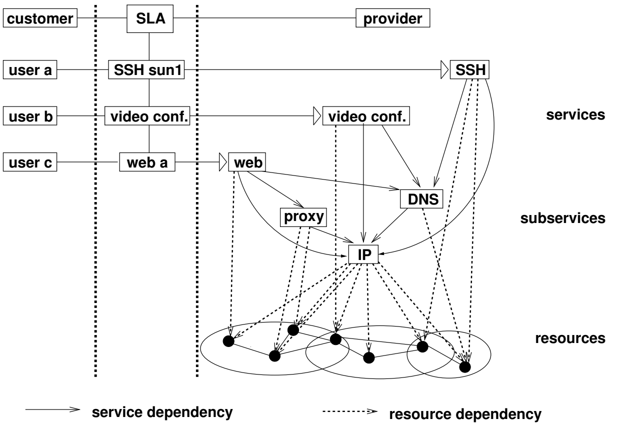

**Figure 1: Scenario**

To get a common understanding, we distinguish between different types of events:

**Resource event:** We use the term *resource event* for network events and system events. A network event refers to events like `node up/down` or `link up/down` whereas system events refer to events like `server down` or `authentication failure`.

**Service event:** A *service event* indicates that a service does not work properly. A trouble ticket which is generated from a customer report is a kind of such an event. Other service events can be generated by the provider of a service, if the provider himself detects a service malfunction.

In such a scenario the provider may receive service events from customers which indicate that SSH, web hosting service, and video conference service are not available. When referring to the service hierarchy, the provider can conclude in such a case that all services depend on DNS. Therefore, it seems more likely that a common resource which is necessary for this service does not work properly or is not available than to assume three independent service failures. In contrast to a resource-oriented perspective where all of the service events would have to be processed separately, the service events can be linked together. Their information can be aggregated and processed only once. If, e.g., the problem is solved, one common message to the customers that their services are available again is generated and distributed by using the list of linked service events. This is certainly a simplified example. However, it shows the general principle of identifying the common subservices and common resources of different services.

It is important to note that the service-oriented perspective is needed to integrate service aspects, especially QoS aspects. An example of such an aspect is that a fault does not lead to a total failure of a service, but its QoS parameters, respectively agreed service levels, at the customer-provider interface might not be met. A degradation in service quality which is caused by high traffic load on the backbone is another example. In the resource-oriented perspective it would be possible to define events which indicate that there is a link usage higher than a threshold, but no mechanism has currently been established to find out which services are affected and whether a QoS violation occurs.

To summarize, the reasons for the necessity of a service-oriented event correlation are the following:

**Keeping of SLAs (top-down perspective):** The time interval between the first symptom (recognized either by provider, network management tools, or customers) that a service does not perform properly and the verified fault repair needs to be minimized. This is especially needed with respect to SLAs as such agreements often contain guarantees like a mean time to repair.

**Effort reduction (top-down perspective):** If several user trouble reports are symptoms of the same fault, fault processing should be performed only once and not several times. If the fault has been repaired, the affected customers should be informed about this automatically.

**Impact analysis (bottom-up perspective):** In case of a fault in a resource, its influence on the associated services and affected customers can be determined. This analysis can be performed for short term (when there is currently a resource failure) or long term (e.g., network optimization) considerations.

# 4. WORKFLOW MODELING

In the following we examine the established IT process management frameworks IT Infrastructure Library (ITIL) and enhanced Telecom Operations Map (eTOM). The aim is find out where event correlation can be found in the process structure and how detailed the frameworks currently are. After that we present our solution for a workflow modeling for the service-oriented event correlation.

## 4.1 IT Infrastructure Library (ITIL)

The British Office of Government Commerce (OGC) and the IT Service Management Forum (itSMF) [9] provide a collection of best practices for IT processes in the area of IT service management which is called ITIL. The service management is described by 11 modules which are grouped into Service Support Set (provider internal processes) and Service Delivery Set (processes at the customer-provider interface). Each module describes processes, functions, roles, and responsibilities as well as necessary databases and interfaces. In general, ITIL describes contents, processes, and aims at a high abstraction level and contains no information about management architectures and tools.

The fault management is divided into Incident Management process and Problem Management process.

**Incident Management:** The Incident Management contains the service desk as interface to customers (e.g., receives reports about service problems). In case of severe errors structured queries are transferred to the Problem Management.

**Problem Management:** The Problem Management's tasks are to solve problems, take care of keeping priorities, minimize the reoccurrence of problems, and to provide management information. After receiving requests from the Incident Management, the problem has to be identified and information about necessary countermeasures is transferred to the Change Management.

The ITIL processes describe only what has to be done, but contain no information how this can be actually performed. As a consequence, event correlation is not part of the modeling. The ITIL incidents could be regarded as input for the service-oriented event correlation, while the output could be used as a query to the ITIL Problem Management.

## 4.2 Enhanced Telecom Operations Map (eTOM)

The TeleManagement Forum (TMF) [18] is an international non-profit organization from service providers and suppliers in the area of telecommunications services. Similar to ITIL a process-oriented framework has been developed at first, but the framework was designed for a narrower focus, i.e., the market of information and communications service providers. A horizontal grouping into processes for customer care, service development & operations, network & systems management, and partner/supplier is performed. The vertical grouping (fulfillment, assurance, billing) reflects the service life cycle.

In the area of fault management three processes have been defined along the horizontal process grouping.

**Problem Handling:** The purpose of this process is to receive trouble reports from customers and to solve them by using the Service Problem Management. The aim is also to keep the customer informed about the current status of the trouble report processing as well as about the general network status (e.g., planned maintenance). It is also a task of this process to inform the

QoS/SLA management about the impact of current errors on the SLAs.

**Service Problem Management:** In this process reports about customer-affecting service failures are received and transformed. Their root causes are identified and a problem solution or a temporary workaround is established. The task of the "Diagnose Problem" subprocess is to find the root cause of the problem by performing appropriate tests. Nothing is said how this can be done (e.g., no event correlation is mentioned).

**Resource Trouble Management:** A subprocess of the Resource Trouble Management is responsible for resource failure event analysis, alarm correlation & filtering, and failure event detection & reporting. Another subprocess is used to execute different tests to find a resource failure. There is also another subprocess which keeps track about the status of the trouble report processing. This subprocess is similar to the functionality of a trouble ticket system.

The process description in eTOM is not very detailed. It is useful to have a check list which aspects for these processes have to be taken into account, but there is no detailed modeling of the relationships and no methodology for applying the framework. Event correlation is only mentioned in the resource management, but it is not used in the service level.

## 4.3 Workflow Modeling for the Service-Oriented Event Correlation

Fig. 2 shows a general service scenario which we will use as basis for the workflow modeling for the service-oriented event correlation. We assume that the dependencies are already known (e.g., by using the approaches mentioned in Section 2). The provider offers different services which depend on other services called subservices (service dependency). Another kind of dependency exists between services/subservices and resources. These dependencies are called resource dependencies. These two kinds of dependencies are in most cases not used for the event correlation performed today. This resource-oriented event correlation deals only with relationships on the resource level (e.g., network topology).

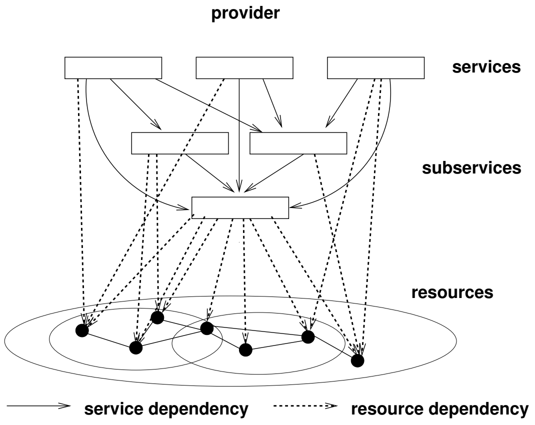

**Figure 2: Different kinds of dependencies for the service-oriented event correlation**

The dependencies depicted in Figure 2 reflect a situation with no redundancy in the service provisioning. The relationships can be seen as AND relationships. In case of redundancy, if e.g., a provider has 3 independent web servers, another modeling (see Figure 3) should be used (OR relationship). In such a case different relationships are possible. The service could be seen as working properly if one of the servers is working or a certain percentage of them is working.

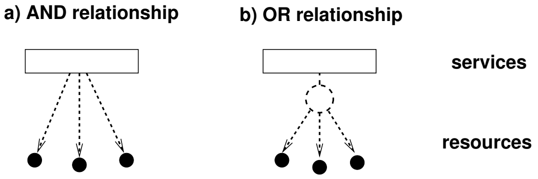

**Figure 3: Modeling of no redundancy (a) and of redundancy (b)**

As both ITIL and eTOM contain no description how event correlation and especially service-oriented event correlation should actually be performed, we propose the following design for such a workflow (see Fig. 4). The additional components which are not part of a device-oriented event correlation are depicted with a gray background. The workflow is divided into the phases fault detection, fault diagnosis, and fault recovery.

In the fault detection phase resource events and service events can be generated from different sources. The resource events are issued during the use of a resource, e.g., via SNMP traps. The service events are originated from customer trouble reports, which are reported via the Customer Service Management (see below) access point. In addition to these two "passive" ways to get the events, a provider can also perform active tests. These tests can either deal with the resources (resource active probing) or can assume the role of a virtual customer and test a service or one of its subservices by performing interactions at the service access points (service active probing).

The fault diagnosis phase is composed of three event correlation steps. The first one is performed by the resource event correlator which can be regarded as the event correlator in today's commercial systems. Therefore, it deals only with resource events. The service event correlator does a correlation of the service events, while the aggregate event correlator finally performs a correlation of both resource and service events. If the correlation result in one of the correlation steps shall be improved, it is possible to go back to the fault detection phase and start the active probing to get additional events. These events can be helpful to confirm a correlation result or to reduce the list of possible root causes.

After the event correlation an ordered list of possible root causes is checked by the resource management. When the root cause is found, the failure repair starts. This last step is performed in the fault recovery phase.

The next subsections present different elements of the event correlation process.

## 4.4 Customer Service Management and Intelligent Assistant

The Customer Service Management (CSM) access point was proposed by [13] as a single interface between customer

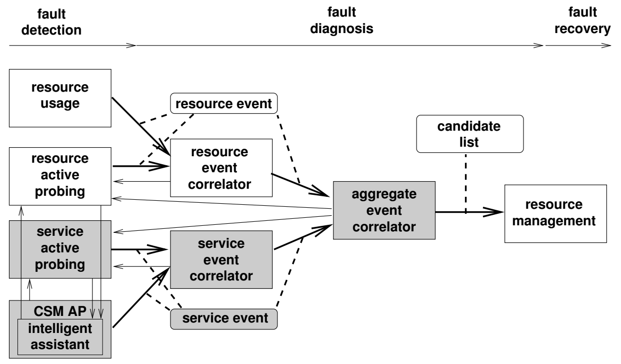

**Figure 4: Event correlation workflow**

and provider. Its functionality is to provide information to the customer about his subscribed services, e.g., reports about the fulfillment of agreed SLAs. It can also be used to subscribe services or to allow the customer to manage his services in a restricted way. Reports about problems with a service can be sent to the customer via CSM. The CSM is also contained in the MNM Service Model (see Section 5).

To reduce the effort for the provider's first level support, an Intelligent Assistant can be added to the CSM. The Intelligent Assistant structures the customer's information about a service problem. The information which is needed for a preclassification of the problem is gathered from a list of questions to the customer. The list is not static as the current question depends on the answers to prior questions or from the result of specific tests. A decision tree is used to structure the questions and tests. The tests allow the customer to gain a controlled access to the provider's management. At the LRZ a customer of the E-Mail Service can, e.g., use the Intelligent Assistant to perform "ping" requests to the mail server. But also more complex requests could be possible, e.g., requests of a combination of SNMP variables.

## 4.5 Active Probing

Active probing is useful for the provider to check his offered services. The aim is to identify and react to problems before a customer notices them. The probing can be done from a customer point of view or by testing the resources which are part of the services. It can also be useful to perform tests of subservices (own subservices or subservices offered by suppliers).

Different schedules are possible to perform the active probing. The provider could select to test important services and resources in regular time intervals. Other tests could be initiated by a user who traverses the decision tree of the Intelligent Assistant including active tests. Another possibility for the use of active probing is a request from the event correlator, if the current correlation result needs to be improved. The results of active probing are reported via service or resource events to the event correlator (or if the test was demanded by the Intelligent Assistant the result is reported to it, too). While the events that are received from management tools and customers denote negative events (something does not work), the events from active probing should also contain positive events for a better discrimination.

## 4.6 Event Correlator

The event correlation should not be performed by a single event correlator, but by using different steps. The reason for this are the different characteristics of the dependencies (see Fig. 1).

On the resource level there are only relationships between resources (network topology, systems configuration). An example for this could be a switch linking separate LANs. If the switch is down, events are reported that other network components which are located behind the switch are also not reachable. When correlating these events it can be figured out that the switch is the likely error cause. At this stage, the integration of service events does not seem to be helpful. The result of this step is a list of resources which could be the problem's root cause. The resource event correlator is used to perform this step.

In the service-oriented scenario there are also service and resource dependencies. As next step in the event correlation process the service events should be correlated with each other using the service dependencies, because the service dependencies have no direct relationship to the resource level. The result of this step, which is performed by the service event correlator, is a list of services/subservices which could contain a failure in a resource. If, e.g., there are service events from customers that two services do not work and both services depend on a common subservice, it seems more likely that the resource failure can be found inside the subservice. The output of this correlation is a list of services/subservices which could be affected by a failure in an associated resource.

In the last step the aggregate event correlator matches the lists from resource event correlator and service event correlator to find the problem's possible root cause. This is done by using the resource dependencies.

The event correlation techniques presented in Section 2 could be used to perform the correlation inside the three event correlators. If the dependencies can be found precisely, an RBR or codebook approach seems to be appropriate. A case database (CBR) could be used if there are cases which could not be covered by RBR or the codebook approach. These cases could then be used to improve the modeling in a way that RBR or the codebook approach can deal with them in future correlations.

# 5. INFORMATION MODELING

In this section we use a generic model for IT service management to derive the information necessary for the event correlation process.

## 5.1 MNM Service Model

The MNM Service Model [5] is a generic model for IT service management. A distinction is made between *customer side* and *provider side*. The customer side contains the basic roles *customer* and *user*, while the provider side contains the role *provider*. The provider makes the service available to the customer side. The service as a whole is divided into usage which is accessed by the role user and management which is used by the role customer.

The model consists of two main views. The *Service View* (see Fig. 5) shows a common perspective of the service for customer and provider. Everything that is only important for the service realization is not contained in this view. For these details another perspective, the *Realization View*, is defined (see Fig. 6).

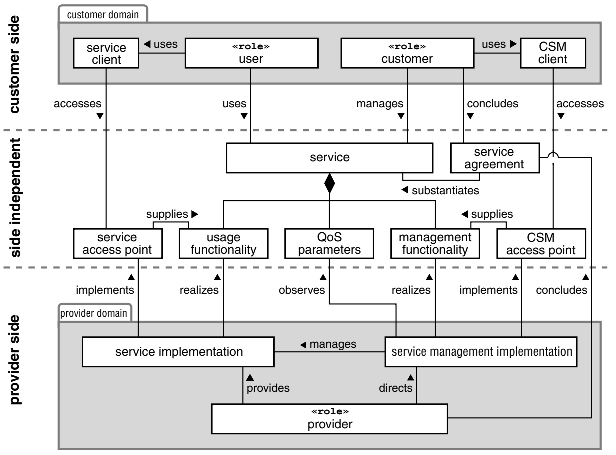

**Figure 5: Service View**

The Service View contains the *service* for which the functionality is defined for usage as well as for management. There are two access points (service access point and CSM access point) where user and customer can access the usage and management functionality, respectively. Associated to each service is a list of QoS parameters which have to be met by the service at the service access point. The QoS surveillance is performed by the management.

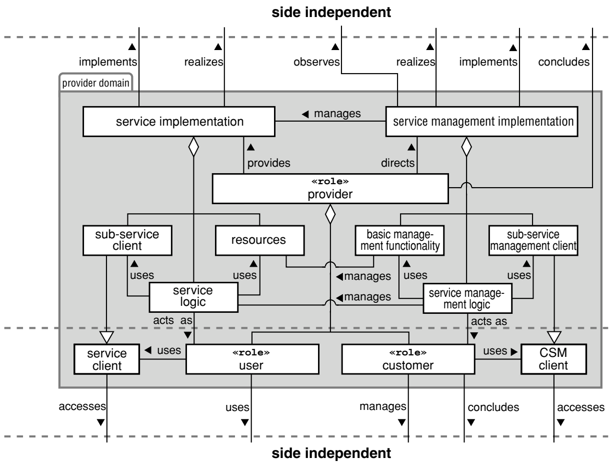

**Figure 6: Realization View**

In the Realization View the service implementation and the service management implementation are described in detail. For both there are provider-internal resources and subservices. For the service implementation a service logic uses internal resources (devices, knowledge, staff) and external subservices to provide the service. Analogous, the service management implementation includes a service management logic using basic management functionalities [8] and external management subservices.

The MNM Service Model can be used for a similar modeling of the used subservices, i.e., the model can be applied recursively.

As the service-oriented event correlation has to use dependencies of a service from subservices and resources, the model is used in the following to derive the needed information for service events.

## 5.2 Information Modeling for Service Events

Today's event correlation deals mainly with events which are originated from resources. Beside a resource identifier these events contain information about the resource status, e.g., SNMP variables. To perform a service-oriented event correlation it is necessary to define events which are related to services. These events can be generated from the provider's own service surveillance or from customer reports at the CSM interface. They contain information about the problems with the agreed QoS. In our information modeling we define an event superclass which contains common attributes (e.g., time stamp). Resource event and service event inherit from this superclass.

Derived from the MNM Service Model we define the information necessary for a service event.

**Service:** As a service event shall represent the problems of a single service, a unique identification of the affected service is contained here.

**Event description:** This field has to contain a description of the problem. Depending on the interactions at the service access point (Service View) a classification of the problem into different categories should be defined. It should also be possible to add an informal description of the problem.

**QoS parameters:** For each service QoS parameters (Service View) are defined between the provider and the customer. This field represents a list of these QoS parameters and agreed service levels. The list can help the provider to set the priority of a problem with respect to the service levels agreed.

**Resource list:** This list contains the resources (Realization View) which are needed to provide the service. This list is used by the provider to check if one of these resources causes the problem.

**Subservice service event identification:** In the service hierarchy (Realization View) the service, for which this service event has been issued, may depend on subservices. If there is a suspicion that one of these subservices causes the problem, child service events are issued from this service event for the subservices. In such a case this field contains links to the corresponding events.

**Other event identifications:** In the event correlation process the service event can be correlated with other service events or with resource events. This field then contains links to other events which have been correlated to this service event. This is useful to, e.g., send a common message to all affected customers when their subscribed services are available again.

**Issuer's identification:** This field can either contain an identification of the customer who reported the problem, an identification of a service provider's employee (in case the failure has been detected by the provider's own service active probing) or a link to a parent service event. The identification is needed, if there are ambiguities in the service event or the issuer should be informed (e.g., that the service is available again). The possible issuers refer to the basic roles (customer, provider) in the Service Model.

**Assignee:** To keep track of the processing the name and address of the provider's employee who is solving or solved the problem is also noted. This is a specialization of the provider role in the Service Model.

**Dates:** This field contains key dates in the processing of the service event such as initial date, problem identification date, resolution date. These dates are important to keep track how quick the problems have been solved.

**Status:** This field represents the service event's actual status (e.g., active, suspended, solved).

**Priority:** The priority shows which importance the service event has from the provider's perspective. The importance is derived from the service agreement, especially the agreed QoS parameters (Service View).

The fields date, status, and other service events are not derived directly from the Service Model, but are necessary for the event correlation process.

## 6. APPLICATION OF SERVICE-ORIENTED EVENT CORRELATION FOR A WEB HOSTING SCENARIO

The Leibniz Supercomputing Center is the joint computing center for the Munich universities and research institutions. It also runs the Munich Scientific Network and offers related services. One of these services is the Virtual WWW Server, a web hosting offer for smaller research institutions. It currently has approximately 200 customers.

A subservice of the Virtual WWW Server is the Storage Service which stores the static and dynamic web pages and uses caching techniques for a fast access. Other subservices are DNS and IP service. When a user accesses a hosted web site via one of the LRZ's Virtual Private Networks the VPN service is also used. The resources of the Virtual WWW Server include a load balancer and 5 redundant servers. The network connections are also part of the resources as well as the Apache web server application running on the servers. Figure 7 shows the dependencies of the Virtual WWW Server.

### 6.1 Customer Service Management and Intelligent Assistant

The Intelligent Assistant that is available at the Leibniz Supercomputing Center can currently be used for connectivity or performance problems or problems with the LRZ E-Mail Service. A selection of possible customer problem reports for the Virtual WWW Server is given in the following:

- The hosted web site is not reachable.
- The web site access is (too) slow.
- The web site contains outdated content.
- The transfer of new content to the LRZ does not change the provided content.
- The web site looks strange (e.g., caused by problems with HTML version)

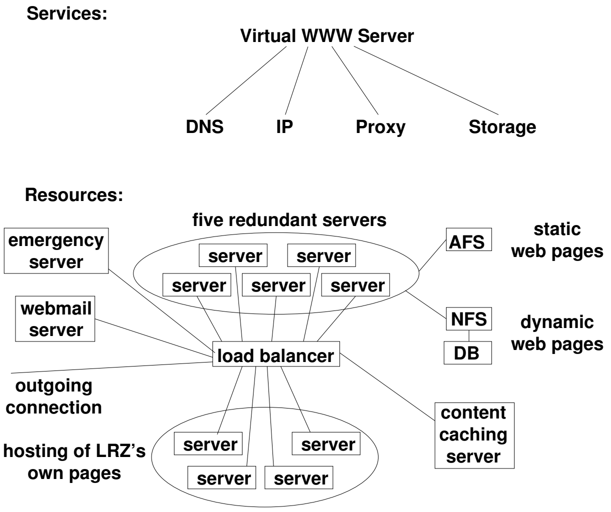

**Figure 7: Dependencies of the Virtual WWW Server**

This customer reports have to be mapped onto failures in resources. For, e.g., an unreachable web site different root causes are possible like a DNS problem, connectivity problem, wrong configuration of the load balancer.

### 6.2 Active Probing

In general, active probing can be used for services or resources. For the service active probing of the Virtual WWW Server a virtual customer could be installed. This customer does typical HTTP requests of web sites and compares the answer with the known content. To check the up-to-dateness of a test web site, the content could contain a time stamp. The service active probing could also include the testing of subservices, e.g., sending requests to the DNS.

The resource active probing performs tests of the resources. Examples are connectivity tests, requests to application processes, and tests of available disk space.

### 6.3 Event Correlation for the Virtual WWW Server

Figure 8 shows the example processing. At first, a customer who takes a look at his hosted web site reports that the content that he had changed is not displayed correctly. This report is transferred to the service management via the CSM interface. An Intelligent Assistant could be used to structure the customer report. The service management translates the customer report into a service event.

Independent from the customer report the service provider's own service active probing tries to change the content of a test web site. Because this is not possible, a service event is issued.

Meanwhile, a resource event has been reported to the event correlator, because an access of the content caching server to one of the WWW servers failed. As there are no other events at the moment the resource event correlation

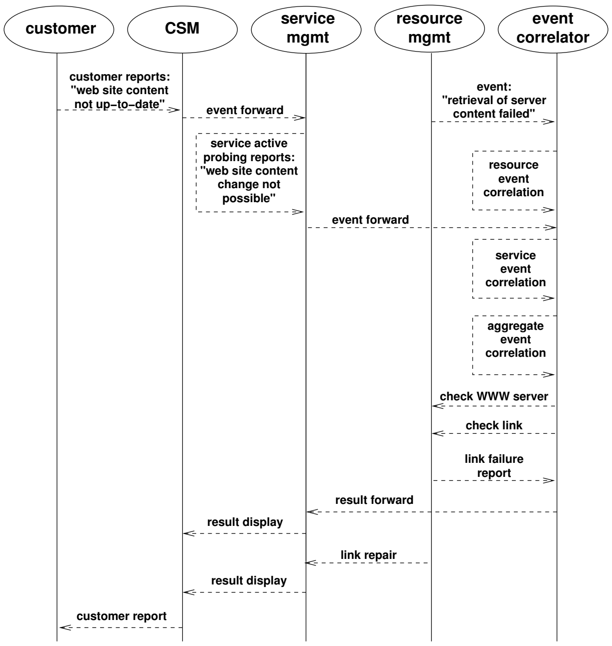
customer
CSM
service mgmt
resource mgmt
event correlator
customer reports: "web site content not up-to-date"
event forward
event: "retrieval of server content failed"
service active probing reports: "web site content change not possible"
resource event correlation
event forward
service event correlation
aggregate event correlation
check WWW server
check link
link failure report
result forward
result display
link repair
result display
customer report

**Figure 8: Example processing of a customer report**

cannot correlate this event to other events. At this stage it would be possible that the event correlator asks the resource management to perform an active probing of related resources.

Both service events are now transferred to the service event correlator and are correlated. From the correlation of these events it seems likely that either the WWW server itself or the link to the WWW server is the problem's root cause. A wrong web site update procedure inside the content caching server seems to be less likely as this would only explain the customer report and not the service active probing result. At this stage a service active probing could be started, but this does not seem to be useful as this correlation only deals with the Web Hosting Service and its resources and not with other services.

After the separate correlation of both resource and service events, which can be performed in parallel, the aggregate event correlator is used to correlate both types of events. The additional resource event makes it seem much more likely that the problems are caused by a broken link to the WWW server or by the WWW server itself and not by the content caching server. In this case the event correlator asks the resource management to check the link and the WWW server. The decision between these two likely error causes can not be further automated here.

Later, the resource management finds out that a broken link is the failure's root cause. It informs the event correlator about this and it can be determined that this explains all previous events. Therefore, the event correlation can be stopped at this point.

Depending on the provider's customer relationship management the finding of the root cause and an expected repair time could be reported to the customers. After the link has been repaired, it is possible to report this event via the CSM interface.

Even though many details of this event correlation process could also be performed differently, the example showed an important advantage of the service-oriented event correlation. The relationship between the service provisioning and the provider's resources is explicitly modeled. This allows a mapping of the customer report onto the provider-internal resources.

## 6.4 Event Correlation for Different Services

If a provider like the LRZ offers several services the service-oriented event correlation can be used to reveal relationships that are not obvious in the first place. If the LRZ E-Mail Service and its events are viewed in relationship with the events for the Virtual WWW Server, it is possible to identify failures in common subservices and resources. Both services depend on the DNS which means that customer reports like "I cannot retrieve new e-mail" and "The web site of my research institute is not available" can have a common cause, e.g., the DNS does not work properly.

# 7. CONCLUSION AND FUTURE WORK

In our paper we showed the need for a service-oriented event correlation. For an IT service provider this new kind of event correlation makes it possible to automatically map problems with the current service quality onto resource failures. This helps to find the failure's root cause earlier and to reduce costs for SLA violations. In addition, customer reports can be linked together and therefore the processing effort can be reduced.

To receive these benefits we presented our approach for performing the service-oriented event correlation as well as a modeling of the necessary correlation information. In the future we are going to apply our workflow and information modeling for services offered by the Leibniz Supercomputing Center going further into details.

Several issues have not been treated in detail so far, e.g., the consequences for the service-oriented event correlation if a subservice is offered by another provider. If a service does not perform properly, it has to be determined whether this is caused by the provider himself or by the subservice. In the latter case appropriate information has to be exchanged between the providers via the CSM interface. Another issue is the use of active probing in the event correlation process which can improve the result, but can also lead to a correlation delay.

Another important point is the precise definition of "dependency" which has also been left out by many other publications. To avoid having to much dependencies in a certain situation one could try to check whether the dependencies currently exist. In case of a download from a web site there is only a dependency from the DNS subservice at the beginning, but after the address is resolved a download failure is unlikely to have been caused by the DNS. Another possibility to reduce the dependencies is to divide a service into its possible user interactions (e.g., an e-mail service into transactions like get mail, sent mail, etc) and to define the dependencies for each user interaction.

## 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 web server is located at wwwmnmteam.informatik.uni-muenchen.de.

## 8. REFERENCES

[1] K. Appleby, G. Goldszmidt, and M. Steinder. Yemanja - A Layered Event Correlation Engine for Multi-domain Server Farms. In *Proceedings of the Seventh IFIP/IEEE International Symposium on Integrated Network Management*, pages 329–344. IFIP/IEEE, May 2001.

[2] Spectrum, Aprisma Corporation. http://www.aprisma.com.

[3] C. Ensel. New Approach for Automated Generation of Service Dependency Models. In *Network Management as a Strategy for Evolution and Development; Second Latin American Network Operation and Management Symposium (LANOMS 2001)*. IEEE, August 2001.

[4] C. Ensel and A. Keller. An Approach for Managing Service Dependencies with XML and the Resource Description Framework. *Journal of Network and Systems Management*, 10(2), June 2002.

[5] M. Garschhammer, R. Hauck, H.-G. Hegering, B. Kempter, M. Langer, M. Nerb, I. Radisic, H. Roelle, and H. Schmidt. Towards generic Service Management Concepts - A Service Model Based Approach. In *Proceedings of the Seventh IFIP/IEEE International Symposium on Integrated Network Management*, pages 719–732. IFIP/IEEE, May 2001.

[6] B. Gruschke. Integrated Event Management: Event Correlation using Dependency Graphs. In *Proceedings of the 9th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM 98)*. IEEE/IFIP, October 1998.

[7] M. Gupta, A. Neogi, M. Agarwal, and G. Kar. Discovering Dynamic Dependencies in Enterprise Environments for Problem Determination. In *Proceedings of the 14th IFIP/IEEE Workshop on Distributed Sytems: Operations and Management*. IFIP/IEEE, October 2003.

[8] H.-G. Hegering, S. Abeck, and B. Neumair. *Integrated Management of Networked Systems - Concepts, Architectures and their Operational Application*. Morgan Kaufmann Publishers, 1999.

[9] IT Infrastructure Library, Office of Government Commerce and IT Service Management Forum. http://www.itil.co.uk.

[10] G. Jakobson and M. Weissman. Alarm Correlation. *IEEE Network*, 7(6), November 1993.

[11] G. Jakobson and M. Weissman. Real-time Telecommunication Network Management: Extending Event Correlation with Temporal Constraints. In *Proceedings of the Fourth IEEE/IFIP International Symposium on Integrated Network Management*, pages 290–301. IEEE/IFIP, May 1995.

[12] S. Kliger, S. Yemini, Y. Yemini, D. Ohsie, and S. Stolfo. A Coding Approach to Event Correlation. In *Proceedings of the Fourth IFIP/IEEE International Symposium on Integrated Network Management*, pages 266–277. IFIP/IEEE, May 1995.

[13] M. Langer, S. Loidl, and M. Nerb. Customer Service Management: A More Transparent View To Your Subscribed Services. In *Proceedings of the 9th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM 98)*, Newark, DE, USA, October 1998.

[14] L. Lewis. A Case-based Reasoning Approach for the Resolution of Faults in Communication Networks. In *Proceedings of the Third IFIP/IEEE International Symposium on Integrated Network Management*. IFIP/IEEE, 1993.

[15] L. Lewis. *Service Level Management for Enterprise Networks*. Artech House, Inc., 1999.

[16] NETeXPERT, Agilent Technologies. http://www.agilent.com/comms/OSS.

[17] InCharge, Smarts Corporation. http://www.smarts.com.

[18] Enhanced Telecom Operations Map, TeleManagement Forum. http://www.tmforum.org.

[19] Verizon Communications. http://www.verizon.com.

[20] H. Wietgrefe, K.-D. Tuchs, K. Jobmann, G. Carls, P. Froelich, W. Nejdl, and S. Steinfeld. Using Neural Networks for Alarm Correlation in Cellular Phone Networks. In *International Workshop on Applications of Neural Networks to Telecommunications (IWANNT)*, May 1997.

[21] S. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie. High Speed and Robust Event Correlation. *IEEE Communiations Magazine*, 34(5), May 1996.