Service-Oriented Event Correlation - the MNM Service Model Applied to E-Mail Services

Andreas Hanemann, David Schmitz
Munich Network Management Team
Leibniz Supercomputing Center
Barer Str. 21, D-80333 Munich, Germany
{hanemann, 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 addresses mainly the correlation of events as reported from management tools. However, the correlation of trouble reports from users needs to be addressed as well, because different reports could have the same cause. In such a case the reports could be linked together and a processing has to performed only once. Therefore, the response time for trouble reports could be improved and service level guarantees could be kept with less effort. We refer to such a type of correlation as service-oriented correlation.*

In this paper we motivate the necessity of such a type of correlation. We use the MNM Service Model, a generic service management model proposed by the MNM Team, to retrieve an appropriate modeling of the necessary correlation information. To show the benefits of the service-oriented correlation we present a real-world scenario, the E-Mail Service offered by the Leibniz Supercomputing Center.

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.

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*. A *service level agreement (SLA)* is 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. Section 2 describes the motivation for service-oriented event correlation and its benefits. After having motivated the need for such a type of correlation we present our proposal for an appropriate workflow modeling (see Section 3). In Section 4 we present our information modeling which is derived from the MNM Service Model. This modeling is applied to an e-mail service scenario at the Leibniz Supercomputing Center in Section 5. The last section concludes the paper and presents future work.

2 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 zero, one, or more services.

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 the SSH, web hosting service and video conference service are not available.

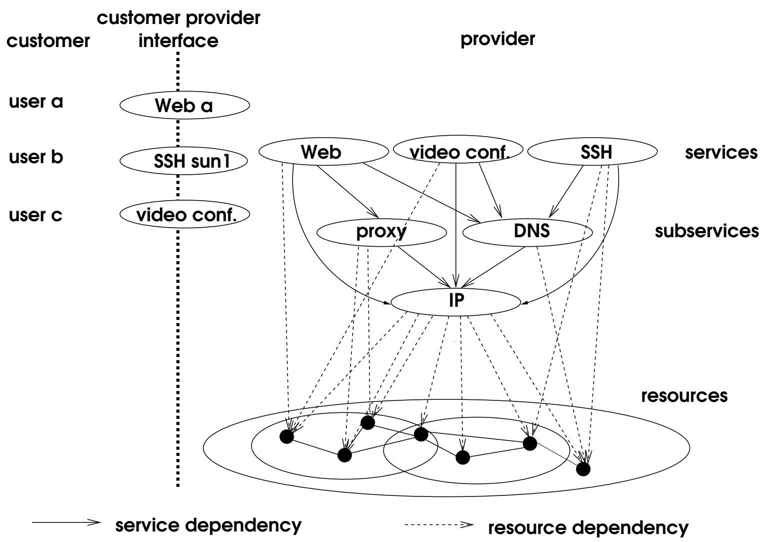

Figure 1. Scenario

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. One example of such an aspect could be that a fault (e.g. breakdown of a device) 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. This is also the case if a degradation in service quality is caused by high traffic load on the backbone. In the resource-oriented perspective it would be possible to define events which indicate that a link usage is 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 a guaranteed 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 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 currently is a resource failure) or long term (e.g. network optimization) considerations.

At this point we have motivated the necessity of a service-oriented event correlation. In the next sections we are approaching an appropriate workflow and information modeling to perform such a type of correlation.

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. 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).

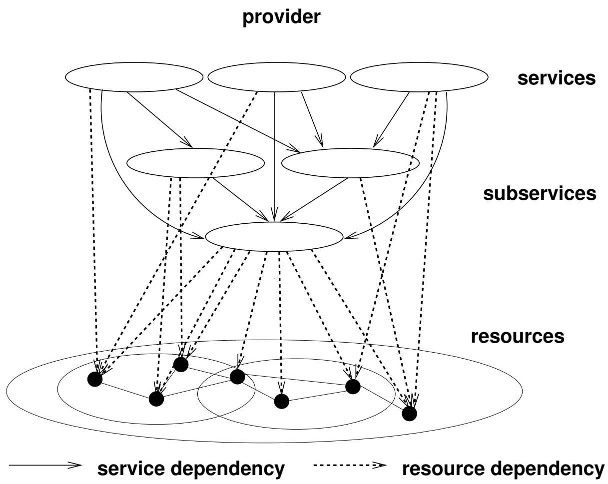

Figure 2. Different kinds of dependencies for the service-oriented event correlation

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 e.g. if 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.

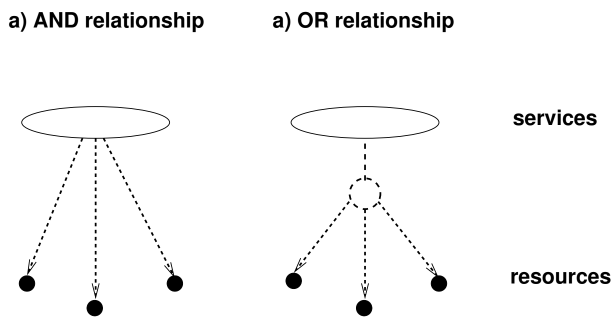

Figure 3. Modeling of no redundancy (a) and of redundancy (b)

As current process frameworks like ITIL and eTOM contain no appropriate workflow for service-oriented event correlation [19], we propose the following design for such a workflow (see Fig. 4). The workflow is divided into the three phases fault detection, fault diagnosis, and fault recovery. In general, we have two kinds of events: Resource events, which contain information about failures in resources, and service events, which contain information about service problems.

In the fault detection phase these 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).

An important part of the fault diagnosis phase is the event correlation. The correlation contains 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 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.

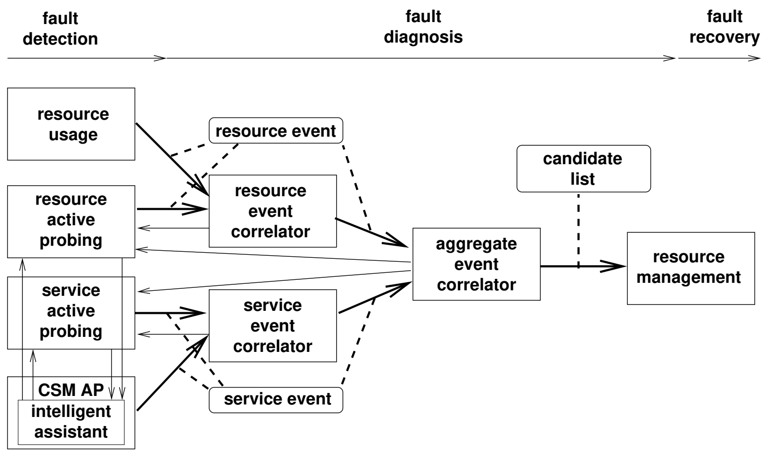

Figure 4. Event correlation workflow

3.1 Customer Service Management and Intelligent Assistant

The MNM Service Model contains a Customer Service Management (CSM) access point as a single interface between customer 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.

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 trouble. 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 start a "ping" request to the mail server. But also more complex requests could be possible, e.g. requests of a combination of SNMP variables.

3.2 Active Probing

Active probing (see Fig. 5) 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

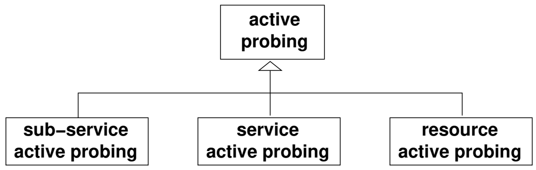

Figure 5. Active Probing

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.

3.3 Event Correlator

Because we have to deal with two types of events (resource events and service events) in the service-oriented scenario, the event correlation should be performed in 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, e.g. caused by the network topology. 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 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. 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 the video conference service and e-mail service 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.

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

Fig. 6 shows the different event correlators.

4 Information Modeling

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

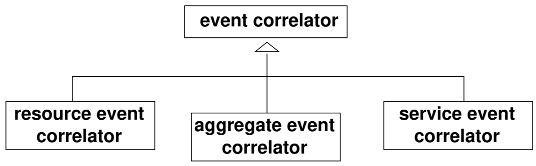

Figure 6. Event Correlators

4.1 MNM Service Model

The MNM Service Model [20], which was developed by the Munich Network Management Team, is a generic model for IT service management. It distinguishes 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. 7) shows a common perspective of the service for customer and provider. Everything that is only important for the realization of the service is not contained in this view. For these details another perspective, the *Realization View*, is defined (see Fig. 8).

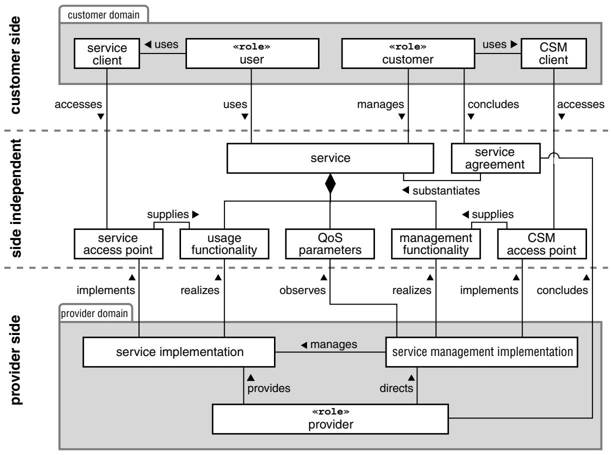

Figure 7. Service View

The Service View contains the *service* for which the functionality is defined for management as well as for usage. 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.

In the Realization View the service implementation and the service management implementation are described in detail. For both there are provider-internal resources and

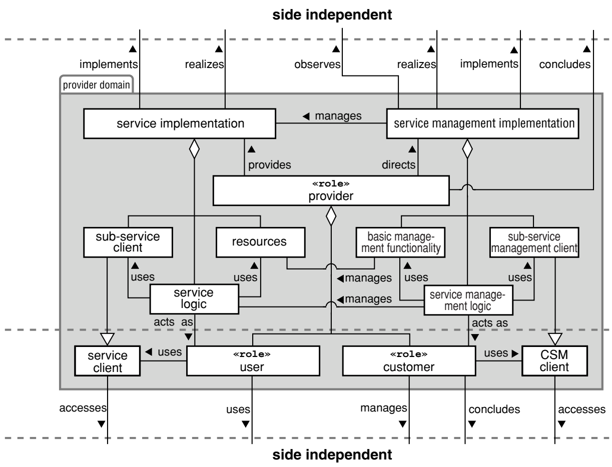

Figure 8. Realization View

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 [21] 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.

4.2 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 (see Fig. 9).

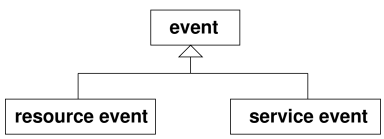

Figure 9. Events

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

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 be possible to add an informal description of the problem.

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 (see below). 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.

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).

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.

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

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.

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

5 Application of Service-Oriented Event Correlation for an E-Mail Scenario

The LRZ, which is the computing center of the Munich universities and runs the scientific network in Munich, offers e-mail access for staff and students from the Munich universities and other research institutions. To perform a service-oriented event correlation, it is important to identify the dependencies of the service from subservices and resources. In case of the E-Mail Service subservices are DNS, SSH Service (for secure service access), IP Service, and Storage Service. The E-Mail Service is provided on two central mail relays with a load balancer and some dedicated servers for different user groups as well as the network between these components. The application software is also part of the resources.

The service-oriented event correlation should be able to handle user requests e.g. reports that e-mail cannot be received. This could have different root causes such as mail relay failure, load balancer failure, network link failure or wrong routing information in the DNS. The list of possible root causes has to be identified in the event correlation process.

For example (see Figure 10), a customer reports that new e-mails cannot be received. This report is transferred to the service management. At the same time the service management has already got the information by performing own tests that there are also problems with the SSH Service. As in commercial products today (e.g. HP ECS for HP OpenView) a correlation is performed at first for events on the resource level. In this example we have no events from the network management and an event that an authentification at a mail server has failed for the systems management. As we have only one event, no further correlation can be performed on this level. Then, a correlation using only service events is performed. It is identified that SSH is a subservice of the E-Mail Service and therefore it can be assumed that the event from the E-Mail Service is a consequence of a failure inside the SSH Service. After that, a correlation between the resource events and the service events is conducted. The resource event (authentification failure at a mail server) and the service event (problem with SSH service) make it seem likely that there is a failure in the mail server's authentification process. Even though both problems could also be explained by a network problem, this does not seem likely, because no network events have been reported. Besides testing the mail server it depends on the provider's policy (e.g. addictive to the effort) whether other tests are also performed as it could be the case that an event has been lost. Maybe some quick "ping"-tests could be used to make sure that the error is not caused by a broken link. Assuming that in this case the check shows that the mail server's authentification process has crashed, a repair of the failure begins. It depends on the provider's policy (e.g. with respect to desired transparency to the customers) whether a report about finding the root cause is issued to the customers or whether this is only done

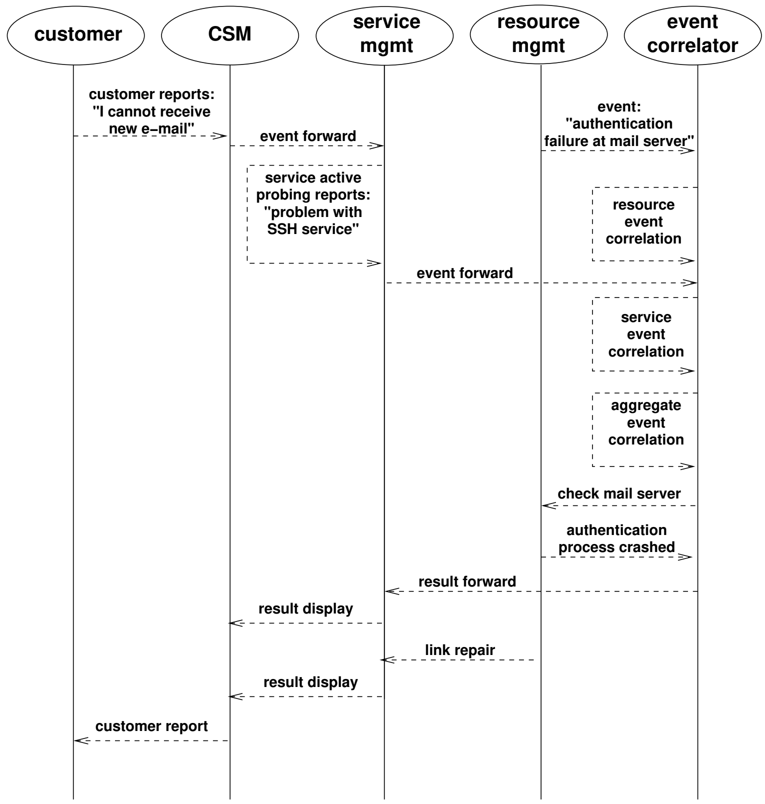

Figure 10. Example processing of a customer report

when the service is available again.

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] Jakobson, G. and Weissman, M.: Real-time Telecommunication Network Management: Extending Event Correlation with Temporal Constraints. In Sethi, A.S., Raynaud, Y., and Faure-Vincent, F. (eds.): Proceedings of the IEEE/IFIP Fourth Inter-

national Symposium on Integrated Network Management, pages 290-301, Chapman and Hall, May 1995.

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

[3] Ensel, C.: 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 Publishing, IEEE, Belo Horizonte, Brazil, August, 2001.

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

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

[6] Lewis, L.: Service Level Management for Enterprise Networks. Artech House, Inc. 1999. ISBN 1-58053-016-8.

[7] Jakobson, G., and Weissman, M.D.: Alarm Correlation. IEEE Network, pages 52-59, Nov. 1993.

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

[9] http://www.agilent.com/comms/OSS

[10] Appleby, K., Goldszmidt, G., and Steinder, M.: Yemanja - A Layered Event Correlation Engine for Multi-domain Server Farms. In: Pavlou, G., Anerousis, N., and Liotta, A. (eds.): Integrated Network Management, VII, pages 329-344, IEEE/IFIP, May 2001.

[11] http://www.verizon.com

[12] Kliger, S., Yemini, S., Yemini, Y., Ohsie, D., and Stolfo, S.: A Coding Approach to Event Correlation. In: Integrated Network Management IV, pages 266-277, Chapman & Hall, 1995.

[13] Yemini, S.A., Kliger, S., Mozes, E., Yemini, Y., and Ohsie, D.: High Speed and Robust Event Correlation. IEEE Communications Magazine, pages 82-90, Volume 34, Issue 5, May 1996.

[14] http://www.smarts.com

[15] Lewis, L.: A Case-based Reasoning Approach for the Resolution of Faults in Communication Networks. In: H.-G. Hegering and Y. Yemini (eds.): Integrated Network Management, III (C-12), Elsevier Science Publishers B.V. (North-Holland), 1993.

[16] http://www.aprisma.com

[17] Dreo Rodosek, G.: A Generic Model for IT Services and Service Management. In: Goldszmidt, G. and Schönwälder, J. (eds.): Integrated Network Management VIII, pages 171-184, Kluver Academic Publishers, March 2003.

[18] Hanemann, A. and Schmitz, D.: Why is Service-Orientation Necessary for Event Correlation? Proceedings of the DAIS/FMOODS PhD Student Workshop, Paris, France, November 2003.

[19] Hanemann, A. and Schmitz, D.: Service-Oriented Event Correlation - Workflow and Information Modeling Approached. Proceedings of the Third International Workshop on Distributed Event Based Systems (DEBS 2004). Edinburgh, Scotland, May 2004.

[20] Garschhammer, M., Hauck, R., Hegering, H.-G., Kempter, B., Langer, M., Nerb, M., Radisic, I., Rölle, H., and Schmidt, H.: Towards generic Service Management Concepts - A Service Model Based Approach. In: Pavlou, G., Anerousis, N., and Liotta, A. (eds.): Integrated Network Management, VII, pages 719-732, IEEE/IFIP, May 2001.

[21] Hegering, H.-G., Abeck, S., and Neumair, B.: Integrated Management of Networked Systems - Concepts, Architectures and their Operational Application. Morgan Kaufmann Publishers, ISBN 1-55860-571-1, 1999.