A Hybrid Rule-Based/Case-Based Reasoning Approach for Service Fault Diagnosis

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

***Abstract*— In today's service market the provisioning of high quality services has become a critical issue for providers as the business success of their customers is often based on the well functioning of these subscribed services. Besides a proper service configuration an efficient fault management needs to be in place. Requirements for such a fault management are the timely resolution of problems affecting the service quality and a reasonable balance between the fault management effort and the costs saved by preventing service level agreement violations.**

In order to fulfill these requirements we propose the adaptation of event correlation techniques which have already proven to be useful in the area of network and systems management. Our hybrid architecture consists of a rule-based reasoning module, whose rules are derived from a modeling of services and underlying infrastructure, and a case-based reasoning module. Due to the complexity of today's service provisioning we use the latter one to collect cases that cannot be covered by the rules so far. The experience gained from the cases is used to improve the modeling and therefore to improve the rules. We use a service provisioning scenario at a large IT service provider to show the applicability of our approach.

I. Introduction

The reliability of IT services has become a necessary basis for many companies. In order to focus on their key strengths such companies have often subscribed to IT services from external providers. These business relationships are formalized in service level agreements (SLAs) which contain guarantees for quality of service (QoS) parameters. These guarantees include penalties for not meeting the QoS thresholds.

For such an external provider it is therefore crucial to ensure that the QoS guarantees are met requiring effective service fault management. Fault management has to react quickly onto service malfunctions as SLAs often contain time constraints like the Mean Time To Repair (MTTR). In contrast to this aim, the effort for fault management should be minimized to save costs. An important part of service fault management is the fault diagnosis, i.e. the identification of the root cause which has led to a service malfunction. Solutions to this issue cannot be regarded as satisfactory yet.

In previous work [1] we have proposed to use event correlation techniques for service fault diagnosis and have defined a framework for the interaction of the correlation engine with its environment. Event correlation techniques have proven to be useful in fault management for network infrastructure and end systems. The idea behind the application of these techniques is to automate the fault management as much as possible leading to a minimization of the fault resolution time. Furthermore, the provider is enabled to improve the experience based fault management which is often error prone. The dependency from staff experience is not favorable as people can be temporarily unavailable or might leave the organization together with their problem solving knowledge. The latter benefit is achieved as a side effect since the automation makes it necessary to store service fault management knowledge in a standardized way.

It is important to note the different characteristics of faults in service management in opposition to network and systems management. While the events that are encountered in network and systems management (called *resource events*) denote objective facts that are in most cases defined by the vendors of the devices, *service events* are originated from customer reports. They specify a service quality degradation in terms of the quality that has been defined in the SLA. This definition with respect to the SLAs has been chosen because only customer reports concerning SLA affecting situations are relevant for the provider in the first place. The service events need to be standardized in order to allow for automation of their processing.

In this paper an example scenario is used to show the current situation of service fault diagnosis (Section II). Related work focusing on the examination of event correlation techniques is presented in Section III. The hybrid approach that has been developed for dealing with the service events and a methodology for its application is presented in Section IV, while information about the implementation for the example scenario is given in Section V. Conclusion and future work are subject to the last section.

II. Service Provisioning Scenario

The Leibniz Supercomputing Center (LRZ) is a large ISP in the Munich metropolitan area and offers related services. One of these services is the E-Mail Service which is provided for students and staff of the Munich universities.

Even though the service can currently be regarded as best effort service (i.e., without explicit quality guarantees), its proper operation is highly critical due to the amount of users requiring the timely recognition of critical situations.

The E-Mail Service is dependent on other services like DNS and network connectivity services. Its resources include

servers for sending/receiving mail and their interconnections.

The fault management for this service is currently performed as follows. A user who experiences a problem with her e-mail account can either contact the LRZ service desk directly or can use the web-based problem preclassification tool *Intelligent Assistant* [2]. This tool guides the user to traverse a query tree composed of questions (e.g., how the user accesses the service) and tests (e.g., component ping tests) to gain a problem preclassification and in some cases already a solution. The result of this preclassification is forwarded to the service desk.

Sometimes the problem can already be resolved at the service desk if the customer has made a mistake in the service usage or if the problem is already known and its resolution is under way. Otherwise, a trouble ticket (Remedy ARS Trouble Ticket System) is opened to delegate the problem to other employees responsible for the service. These employees can access management tools like HP OpenView (where an event correlation is performed by using network topology information), IBM Tivoli, and Infovista or examine log files to find the error. The root cause of the problem is reported to the service desk via the trouble ticket system and the user is informed about the service status.

As described, there is currently no automated method in place to map customer reports concerning the same root causes together to prevent a parallel processing of these reports. The Intelligent Assistant is an important basis for doing so as it has introduced a standardization to customer reports. This standardized service problem report reflects a service event in our terms. In addition, an automated way to match the customer reports to events on the network and systems management level (resource events) is missing for a service-orientation.

III. Related Work

Our examination of IT process management frameworks (ITIL, eTOM) in [1] has shown that these frameworks currently provide a high level description what needs to be done for the service fault diagnosis. Due their genericity they do not focus on the process realization.

We have examined event correlation techniques for their adaptability to deal with service-oriented events. In the following, the criteria for selecting the event correlation techniques are motivated and a table to summarize the examination results is given. The criteria are ordered according to decreasing importance.

Maintainability: The way services are provided today has become very dynamic, i.e. there are frequent changes in the collaboration of services as well as in the configuration of underlying resources. Therefore, the correlation technique should allow for an easy update of correlation information when changes in the service provisioning are performed. This requirement is crucial for achieving an effort reduction in service fault management.

Modeling: The relationships that are found in the provisioning of services are quite complex. There are inter-service dependencies, dependencies between services and resources as well as relationships on the resource level which have to be considered. As a consequence, the modeling which is used for the correlation technique has to be able to reflect this complexity especially with respect to redundancy and quality degradations.

	Maintainability	Modeling	Robustness	Performance
MBR	–	++	0	0
RBR	-	+	-	+
Codebook	-	-	0	+
CBR	+	+	+	-

TABLE I
USABILITY OF EVENT CORRELATION TECHNIQUES FOR SERVICE-ORIENTED EVENT CORRELATION

Robustness: Due to the complexity of service provisioning that is encountered in real-world scenarios, it cannot be assumed that the correlation knowledge base is always complete. Therefore, the correlation technique should be flexible to deal with unknown situations. After the root cause has been identified it should be easily possible to improve the future treatment of similar situations. In addition, a misguided correlation may be caused by incorrect information from the correlation knowledge base. An improvement of correlation information should be feasible by backtracking the misguided correlation.

Performance: While the correlation speed is very critical in the area of network and systems management where hundreds or thousands of events per second have to be processed in an event storm, the number of service events resulting from customer reports is usually much lower. However, these very critical events have to be matched to a potentially large number of resource events. Furthermore, tests to improve the correlation result may be requested during the correlation which generate additional events.

Table I shows the results of the examination of model-based (MBR), rule-based (RBR), and case-based reasoning (CBR) as well as the codebook approach (see [1] for further information about the approaches and their advantages/disadvantages).

It is not necessary to select just one techniques for a correlation task. In [3] a hybrid approach combining RBR and CBR has been proposed to deal with highly dynamical situations. In the proposed architecture an RBR and a CBR component run in parallel. The RBR engine uses temporal and spatial dependencies to correlate reported events, while the CBR engine makes use of prior situation templates. As no implementation details have been provided yet, it remains open how the collaboration of the engines is performed. According to the authors this work has been the first attempt to combine RBR and CBR techniques in the network and systems management domain.

A good insight into the current situation concerning service-orientation in the industry is given by a Netcool whitepaper [4]. After a filtering step, which is used to reduce the number of events (by de-duplication and association of event pairs

like link down/up), the Netcool suite performs a rule-based correlation of events with respect to devices and network topology. The service correlation that is mentioned in the whitepaper is only basic compared to what we are addressing. Events on the resource layer are related to services so that some kind of impact can be estimated, but a service modeling appropriate for this is not part of Netcool. The user is required to provide a suitable service modeling, but it remains open which requirements have to be taken into account. The event correlation does not integrate service events into the event correlation. Therefore, no customer reports about a service quality degradation can be mapped onto resource problems yet.

IV. Improvement by Event Correlation Techniques

After providing information about our proposed event correlation framework (compare [1]) including the choice of a hybrid event correlation architecture the rule maintenance and service modeling is addressed in this section. In addition, a methodology to adapt the approach to a concrete scenario is provided.

A. Service-Oriented Event Correlation Framework

Besides the service event correlator itself the framework contains a component called Customer Service Management [5] which is used for the communication with the customer. The tasks of this component comprise in particular the preprocessing of customer problem reports (formalization, plausibility checks) and their forwarding as service events to the service event correlator. Other tasks are to inform the customer about the status of the service events, the general service status, and to provide SLA reports.

A QoS measurement component [6] is in place to assume the role of a virtual customer and therein to perform service quality tests. These tests simulate typical user interactions which result in the generation of service events in case the service quality is violated or can be tests on demand if the correlation result needs to be improved by correlating additional events.

On the resource level a resource fault management system like the commercial ones mentioned before is applied. Correlated resource events are transferred to the service event correlator to allow for correlation with the service events. The result of the service-oriented event correlation, i.e. a list of resources that could be the problem's root cause is reported back to resource fault management.

A repository called *ServiceMIB* contains all information relevant to service management. For fault management it contains the dependencies of services from other services and their dependencies upon resources. This comprises information which features of resources are needed in order to reach the guaranteed service quality.

B. Hybrid Event Correlation Approach

As a result of the examination (see Section III) we have chosen a hybrid event correlation architecture (see Figure 1).

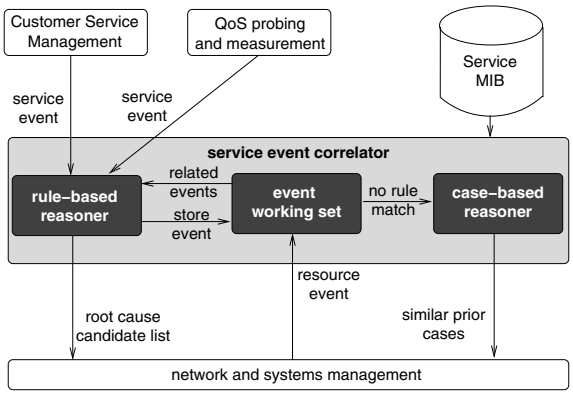

Fig. 1. Hybrid event correlation architecture

It consists of a rule-based reasoner and a case-based reasoner combining the strengths of both approaches.

Besides the existence of efficient correlation algorithms, the flexible representation of knowledge in rules has led to the choice of a rule-based reasoning module. Rule maintainability, which has been identified as a crucial issue for this technique, is ensured by the ServiceMIB which makes it possible to automatically derive the rules from service modeling. As the provider needs to have the information about the offered services for the configuration management anyway the additional effort for the rule generation and maintenance is low. Automated rule derivation ensures that unintended rule interactions become less likely opposed to encoding rules by hand.

Due to the complexity of service provisioning there are situations where the ServiceMIB and therefore the rules do not correctly reflect the current situation. As a consequence, a case-based reasoner is used to collect those events that cannot be covered by the rules. For these cases the root causes have to be found by operation staff. It can be helpful to match the current event to prior cases in order to adapt a previous solution.

In our architecture the case-based reasoner can be seen as a backup in case of an incorrect modeling causing the rule-based reasoner to fail. In contrast, both reasoners run in parallel in the architecture for highly dynamical situations and the case-based reasoner permanently tries to match the current situations onto situations seen before.

C. Rule Generation

Corresponding to the architecture there are two ways to generate the rules which are depicted in Figure 2. The usual way is to derive the rules from the ServiceMIB which is performed by a rule definition component. This component stores the updated rules in the rule database.

The situation gets more complicated if it is necessary to modify the service modeling. This necessity arises if an event cannot be matched to resource events and is therefore forwarded to the case-based reasoner. In the case-based reasoner this event is matched to prior cases. In some situations, an adaptation to a prior solution can be found, while a completely

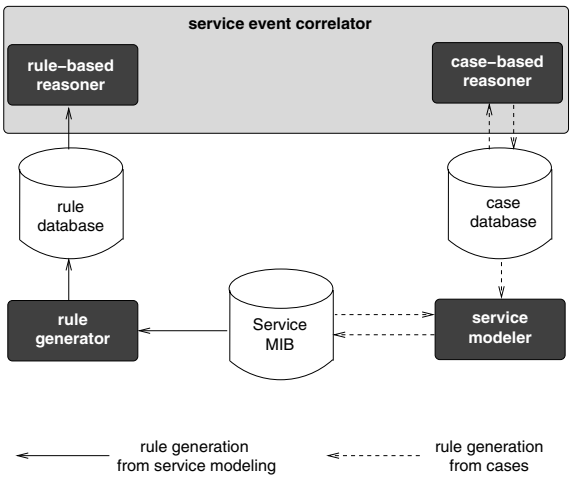

Fig. 2. Possibilities for rule generation

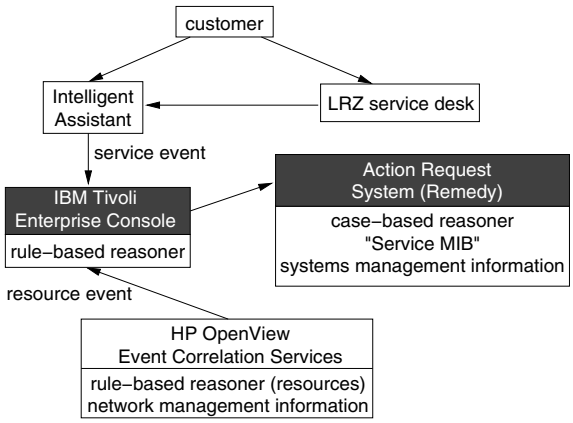

Fig. 3. Prototypical implementation at the LRZ

new solution has to be determined by hand, otherwise. The current event together with its solution is used by the service modeler to improve the service modeling in the ServiceMIB. The rule generation using the ServiceMIB is then applied to update the rule base in order to be able to cope with this and similar events in the future.

D. Application Methodology for a Given Scenario

In order to apply the proposed framework to a concrete scenario we propose the following adaptation methodology.

After a provider has selected services according to an assumed benefit of the service-oriented event correlation, the dependencies for the services and used resources are needed as input. The dependencies may already be known to the provider as part of the configuration management or need to be identified. In the latter case automated approaches like dependency detection by analyzing interactions [7] or neural network based techniques [8] can be used.

Also following the identification of services appropriate service events have to defined. As these events indicate a degradation of the service quality which is described by QoS parameters, the events have to be defined corresponding to a degradation of one or more QoS parameters. As a service often consists of several functionalities (offered user interactions), the event has to be related to one of the service functionalities.

By using the service-related dependencies the rule set for the rule-based reasoner has to be derived in the following. As motivated earlier, the rule definition should happen in an automated way by using the service modeling.

A simple methodology to initialize the case database would be to leave it empty and to wait until it is filled with current events. A more sophisticated method is to derive a set of representative cases from the rule set together with known root causes. These cases are useful to find an adaptation of a prior solution which should be easier than to start without such knowledge.

To notice a service malfunction prior to customers the provider's own service monitoring has to be installed. Typical customer transactions need to be estimated or derived from real user traffic. Then, a schedule has to be set up to test the offered functionalities. While the events which are derived from customer reports usually only specify negative events (something does not or only poorly work), the majority of tests performed by active probing can be presumed to show that a functionality is working properly (positive event). Even though negative events from the provider's own service monitoring should be forwarded to the correlation engine in any case, there is a trade-off how many positive events should be transferred to the correlation engine (improvement of results vs. correlation slowdown).

V. Application to the Service Provisioning Scenario

In this section an application of the service-oriented event correlation framework for the LRZ E-Mail Service is presented. The adaptation of tools for the framework is outlined as well as an example processing of a customer report.

A. Prototypical Implementation

For implementing the framework at the Leibniz Supercomputing Center an architecture (see Fig. 3) has been chosen which integrates previously used tools. The rule-based reasoning module is realized by IBM Tivoli Enterprise Console [9] which has not been part of the LRZ management environment before. This tool has been chosen as it offers a variety of rules and is flexible enough for the definition of service-related rules. The case-based reasoning module is realized by the Remedy's ARS [10]. This tool has been applied for the management of trouble tickets for several years and is also in place for storing the LRZ's hardware configuration. Therefore, the tool has also been chosen for storing service management information for this scenario, i.e. for storing fault and performance management information according to the ServiceMIB approach.

For correlation input LRZ's Intelligent Assistant tool is adapted. As mentioned before its output can be regarded as a service event. In practice, the tool can either be used by the user directly or by the LRZ service desk staff if the user has

reported the problem by phone (especially needed in case of connectivity problems).

For event correlation on the resource level, in particular for the downstream suppression of resource events, HP OpenView Event Correlation Services [11] is used.

B. Example Processing

The LRZ E-Mail Service is used to demonstrate the application of the presented approach. A customer would like to download an e-mail with a large attachment, but this data transfer seems to make nearly no progress from the customer's point of view. Therefore, the customer decides to abort the download and to report the problem to the LRZ. By using the Intelligent Assistant a service event is generated. At this point, it is again possible to observe the subjectivity of service events as it will usually not be defined which duration for the e-mail delivery has to be expected. Customers will report the problem depending on their personal service quality expectations or the importance of the attachment.

The rule-based reasoning component contains a set of rules to match this service event to events on the resource level. Examples for possible root causes are high mail server CPU load or problems with the server's I/O. In addition, there could be a low bandwidth for the connection either caused by a high utilization at some links or by a limited bandwidth due to a slow user link.

Example rules for this scenario are the following.

- `if` **slow e-mail transfer** `and` **mail server CPU load high**
  `then` **check processes at mail server** (possible root cause)
- `if` **slow e-mail transfer** `and` **high path utilization (user IP address, mail server IP address)**
  `then` **check routing** and **check intrusion/DOS attack** (possible root causes)
- `if` **slow e-mail transfer** `and` **low connection bandwidth (user IP address, mail server IP address)**
  `then` **report bandwidth** (no failure)

Another condition that could lead to the delay in the data transfer is a wrong configuration in the access network which is used to connect to the mail server. If this possibility has not been considered before and therefore no appropriate rule exists, there is no rule match and the problem would be transferred to the case-based reasoner.

Here, the problem has to be solved by hand at first and is then stored in the case database. It needs to be decided whether this problem should be the basis for an update of the rules. In case of reoccurrence it can be matched to this former situation in the case database which is sufficient for seldom and low impact problems. Otherwise, the case should be used to update the service modeling and the rule base.

VI. Conclusion and Future Work

After demonstrating the necessity of automation in service fault diagnosis, the idea to adapt event correlation techniques for this task has been motivated. The focus in the presentation of the hybrid RBR/CBR architecture has been the rule generation and the application concept for a given scenario. The implementation of the approach for services at the LRZ is currently carried out to quantify the benefits of the approach.

The output of the service-oriented event correlation - resource failures - should serve as input for an impact analysis [12] which determines the effect of resource failures for the offered services and their customers to decide about recovery measures.

Acknowledgment

The author wishes to thank the members of the 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, the University of Federal Armed Forces in Munich and the Leibniz Supercomputing Center of the Bavarian Academy of Sciences. Its webserver is located at `http://www.mnm-team.org`.

References

[1] A. Hanemann and M. Sailer, "Towards a framework for service-oriented event correlation," in *Proceedings of the International Conference on Service Assurance with Partial and Intermittent Resources (SAPIR 2005)*. Lisbon, Portugal: IARIA/IEEE, July 2005.

[2] G. D. Rodosek and T. Kaiser, "Intelligent assistant: User-guided fault localization," in *Proceedings of the 9th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM 98)*. Newark, DE, USA: IFIP/IEEE, October 1998, pp. 119–129.

[3] G. Jakobson, J. Buford, and L. Lewis, "Towards an architecture for reasoning about complex event-based dynamic situations," in *Proceedings of the Third International Workshop on Distributed Event Based Systems (DEBS 2004)*. IEE, May 2004.

[4] "Managing today's mission-critical infrastructures: Discovery, collection, correlation, and resolution with the netcool suite," http://www.micromuse.com/downloads/pdf_lit/wps/Muse_Discovery_-Correlation_Resolution_Jan04.pdf, Micromuse Incorporated, January 2004.

[5] 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: IFIP/IEEE, October 1998, pp. 195–206.

[6] M. Garschhammer, "Dienstguetebehandlung im Dienstlebenszyklus: von der formalen Spezifikation zur rechnergestuetzten Umsetzung - in German," PhD thesis, University of Munich, Department of Computer Science, Munich, Germany, August 2004.

[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] 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)*. Belo Horizonte, Brazil: IEEE, August 2001.

[9] "IBM Tivoli Enterprise Console," http://www-306.ibm.com/software/tivoli/products/enterprise-console/, IBM.

[10] "Action Request System," http://www.remedy.com, BMC Remedy.

[11] "HP OpenView Event Correlation Services," http://www.managementsoftware.hp.com/products/ecs/, Hewlett Packard.

[12] A. Hanemann, M. Sailer, and D. Schmitz, "A framework for failure impact analysis and recovery with respect to service level agreements," in *Proceedings of the IEEE International Conference on Services Computing (SCC 2005)*. Orlando, Florida, USA: IEEE, July 2005.