Service Monitoring and Alarm Correlations

Oliver Jukić
Virovitica College
Virovitica, Republic of Croatia
oliver.jukic@vsmti.hr

Ivan Heđi
Virovitica College
Virovitica, Republic of Croatia
ivan.hedi@vsmti.hr

***Abstract*—Services provided by telecom operators rely on operator's network infrastructure. We can consider parts of telecommunications network as resources used for service providing. Global network problem is represented as sequence of alarms. Sources of alarms are network elements, in this context – service resources. If given alarm sequence is filtrated and correlated, and problem's root-cause is detected, it is reasonable to think about integration of such root-cause in service monitoring process. In author's previous works, we have proposed ABCDE - Alarm Basic Correlations Discovery Environment. Some basic concepts of that environment are mentioned here in first part of this paper. Second part of this paper should open new perspectives of potential usage of given environment in service monitoring process, together with service structure model and additional sources of information, proposed in our service monitoring architecture.**

***Index Terms*—Service monitoring architecture, service structure, problem root-cause, alarm correlations.**

I. INTRODUCTION

"Fault management primarily covers the detection, isolation and correction of unusual operational behaviors of telecommunication network and its environment" [6]. After network problem's appearance, network generates large number of unsolicited events carrying information about malfunction called alarms. For instance, in the case of transmission link failure, nodes from both sides of transmission link will generate alarms (e.g. "Loss of signal"). Typically, alarms from the whole network are delivered to the network operation and management center, where the alarms are processed by network operator. In that case, we talk about centralized fault or network management.

Services provided by telecom operator over telecommunications network can be impacted by network faults. There are at least two conditions that must be fulfilled in order to evaluate network fault impact on some specific service: network faults should be obtainable and service structure should be known.

Under term "service structure", we think about way how service is "spread" over telecommunications network. Namely, network elements from telecommunications network can be considered as resources needed for service implementation. Hence, network elements should be mapped into resources in service structure description. Alarms from network elements will be mapped to service resources alarms.

One of the most important issues is to recognize the problem's root-cause correlating incoming alarms. Alarms can be correlated by its starting/ending time (when alarm started/ended?), location (where alarm happened?), probable cause (what is alarm nature?) or by another criteria.

When problem root-cause is recognized, it can be propagated to service management in order to take a part in service monitoring process. In author's previous work [13] we have proposed alarm correlations environment, called ABCDE – Alarm Basic Correlations Discovery Environment. This paper is focused on potential usage of ABCDE in service monitoring process.

II. OVERVIEW OF ABCDE

A. ABCDE architecture

Architecture of ABCDE is shown on figure 1:

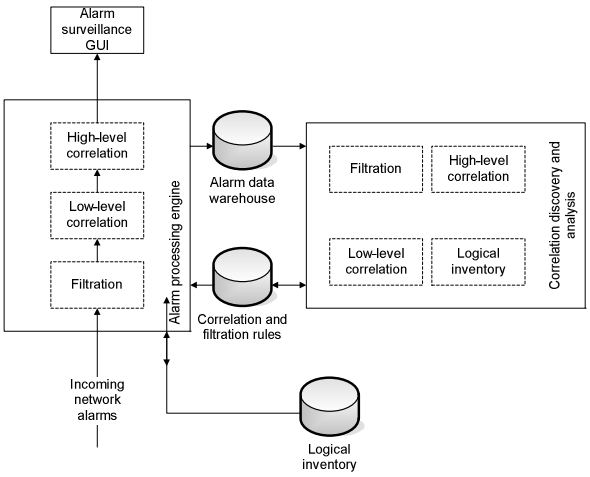

Fig. 1. *Basic ABCDE architecture*

Telecommunication network is source of incoming network alarms. Alarm processing engine performs filtration as well as low and high-level correlation of incoming alarms. Processed alarms appear in alarm surveillance GUI monitored by the network operator. Correlation and filtration rules used by alarm processing engine are stored in database. Incoming alarms are stored into alarm data warehouse. Logical inventory database is database containing data about network structure. It can be used for more efficient alarm correlation as well as for

978-1-4673-2017-7/12/$31.00 ©2012 IEEE

enhancement of incoming alarm data. Namely, alarms often contain numeric representation of alarm source (e.g. port number) while network operators want to see alarm source "friendly" name in alarm surveillance GUI. Logical inventory database allows tying relevant inventory information with alarm data. Logical inventory data should be maintained by network operator. If it doesn't exist, there is proposed technique of logical inventory data extraction from alarm history [7].

All raw alarm history data as well as correlated alarm history data are stored in alarm data warehouse. It contains data for a certain time period, predefined by the operator (e.g. 1 year). Alarm data warehouse is starting point for analysis of typical correlation patterns from alarm historical data.

Correlation and filtration rules database contains data about correlations and filtrations rules that will be performed in real-time manner. Rules are proposed by Correlation discovery and analysis module based on alarm data warehouse performing data mining algorithm on historical alarm data. Algorithm used was Apriori [3], with proposed improvements related to logical inventory database integration.

Potential filtering patterns are discovered and evaluated by Filtration part of Correlation discovery and analysis module. Not all incoming alarms are relevant for further processing. Alarm classification and filtration are described in details in [11], and will not be discussed here more detailed. Filtering is also not always statically related to predefined, concrete network element; it can be rather dynamically changed, based on certain circumstances in network, such as scheduled maintenance procedure on some network elements.

After filtration is done on historical alarm data, low-level correlation discovery and evaluation can be performed. This is primarily related to discovery of general patterns, such as alarm overlapping or alarm jittering.

High-level correlation will cope with concrete alarm patterns, coming from specific network elements. At this stage, alarm clusters are detected first. Alarm cluster is set of alarms received from the network within certain time interval fenced with cluster borders. Namely, we have detected "long enough" time periods without alarms. Those periods are considered as cluster borders. What does it mean "long enough"? If problem occurs, alarms from network elements are generated. However, alarms are not generated at the same time – there is time interval between alarms belonging to the same cluster (typically couple of seconds, even minutes). "Long enough" time interval should ensure that all alarms characterizing problem are received and situation is balanced. Future alarms will characterize another network problem and will be suited into another cluster.

Alarms suited between two cluster borders are members of the same cluster [2]. Cluster is input for the mathematical Apriori algorithm, but in order to improve algorithm performance, we have proposed usage of logical network inventory data to split clusters in smaller parts containing alarms from interconnected alarm locations only. In that case, all interconnections will be taken under consideration while creating alarm clusters: total number of clusters will increase, while average number of alarms in one cluster will decrease. It will drastically improve performance of data mining algorithm execution.

When clusters are generated, the Apriori algorithm is performed. The final result is the number of alarm sequences that occurred frequently in the past. Those sequences are potential high-level correlation rules candidates for future alarm processing. Criteria for acceptation of those candidates can be rule frequency, but also rule can be accepted based on network expert's opinion.

B. Experimental results

Experimental proof of concept was done on real alarm data sample, obtained from access network of GSM system. Base stations are connected to Base Station Controllers via multiplexing transmission system. In this case, connections were realized using microwave radio transmission links. Hence we have opportunity to find real interesting patterns, potentially caused by heavy weather conditions, impacting transmission performance.

Total number of incoming alarms for processing was 36639. If we consider that every correlated alarm sequence can be replaced with one alarm with value-added information "sticked" to it, discovering of sequence with length=N means reduction of (N-1) alarms. At low-level correlation, we have reduced total number of alarms by 23983. It is 65.46% of total number of alarms [13].

Self-solving alarm is alarm that appears and disappears within very short time interval. "Very short" means that alarm duration is so short that network operator is unable to react during alarm activity period. Concrete value depends on network maintenance center organization. Number of self-solving alarms was 5008 alarms. Together with low-level correlation reduced alarms, we have detected 28991 potentially reduced alarms. This is 79.12 % of total number of alarms. Conclusion is that filtration together with low-level correlations can decrease number of alarms in great percent, almost 80 % in this case.

Finally, after number of alarms was reduced, we have 7648 alarms as input for high-level correlations discovery module.

This number can be reduced if we discover some relevant alarm sequences (frequently repeated), and replace it by one alarm. For that purpose, we have used Apriori algorithm, as we discussed in our previous work [2]. However, after sequences are detected, it is necessary to "judge" which sequence is relevant for future and which is not. One of criteria can be frequency of alarm sequence appearing. Also, some sequences can be very relevant, event if those are not repeated very frequently. ABCDE can be used for discovery and statistical processing of alarm sequences, while final decision should be made by human operator.

According to our previous and other related works [12], reduction rate at high-level correlations can be rather high, up to 70%. Using test data sample and finding several alarm sequences confirmed by network experts, reduction rate was 25.41 %.

After alarm sequences presenting network problems were replaced by one alarm with value added information, it can be

considered as problem's root-cause, and it can be used within service monitoring process.

III. ABCDE ROLE IN SERVICE MONITORING

ABCDE system was originally aimed to be network management tool, primarily used for incoming network alarms filtration and correlation as well as analysis of potential correlation and filtration rules candidates. According to results presented, we have achieved great ratio of reduced alarms, "hidden" from network operators, allowing easier manipulation of incoming alarms. On the other hand, high-level correlations lead us toward problem root-cause determination.

If we assume that alarm filtration and correlation process gave us problem's root-cause as its output, logical question is how root-cause just detected is impacting services provided to our customers over managed network?

Since all services provided by telecommunication operators rely on network infrastructure, it is obvious that network problems will have some kind of impact on services. Generally, network monitoring (in this paper we refer on "network monitoring" as on fault management on network level) can be considered as monitoring of service resources. Network monitoring is one of base components that should be included in service management (or "service monitoring": monitoring of service failure/degradation). It is also visible from "TMN pyramid", shown on figure 2.

Network operators included in network monitoring process should be focused on monitoring of fault data from specific part of given network: radio access network, transmission network, GSM core network, Value added services infrastructure (servers, switches, firewalls,…), etc.

Service monitoring operators should be focused on service monitoring. It means, monitoring of service failures or service degradations. However, service failure or degradation can be caused by network errors from different parts of network, attacks, intrusions, etc. Hence it is necessary to stimulate good co-operation between network and service monitoring systems and personnel as well as other systems.

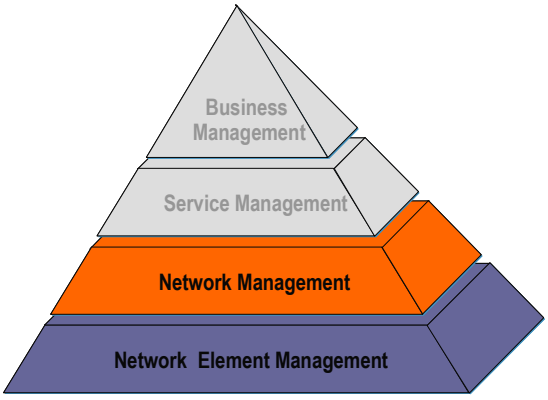

Fig. 2. *Service management in TMN pyramid*

Service "spreading" over different network resources of different types is shown on figure 3. There are two services using network resources from four typical network parts - access, transmission and core network as well as value added services infrastructure:

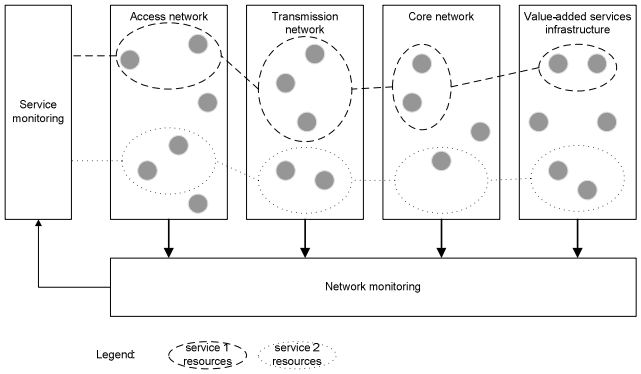

Fig. 3. *Service spreading over network resources*

Discussion mentioned above has triggered creation of potential service monitoring architecture that will include network problems into service current status information. Architecture is presented on figure 4.

Generally, we have perceived five main sources that should be included in service monitoring process:

Root-cause information from ABCDE (from fault management) is result of completed alarm correlation process. Network alarms should be processed, filtrated and correlated, and after that only relevant information should be propagated to service monitoring. We believe that environment such as ABCDE is of great importance in service monitoring, since number of irrelevant information will be excluded from service monitoring. Service monitoring will be more reliable, more efficient and processing requirements will be decreased. However, in the case of attacks or intrusion, other sources for service monitoring are needed.

Performance management should cover collection and analysis of performance data. Key performance indicators should be defined, together with appropriate threshold values. Measured value that is out of predefined borders should trigger input into service monitoring process, carrying information about performance threshold violation.

End-to-end testing activities are based on typical production scenarios, from customer point of view. Test results are delivered to centralized location for analysis. If analysis detects violation of predefined desired testing values, that information should be propagated to service management.

Even if there is no indication of service failure/degradation from network monitoring, performance management and end-to-end test results, there is possibility that service is in failure state, or service quality is degraded. Namely, those three inputs rely mostly on hardware and software systems. Hence, we must take into account possibility of hardware/software errors in management systems. That's reason why we have included **trouble-tickets from CRM system** as potential input in service monitoring. For instance, if call-center (service desk) is occupied by customer's complaints related to service

degradation, it is reasonable to generate trouble-ticket to service monitoring directly. Even in the case that information from other sources is already received, trouble-tickets from CRM system can enhance already received information with customer's experience of service failure/degradation.

Finally, last but probably the most important is **service structure**. Since service rely on network resources, it is necessary to make clear relationship between every particular network element (service resource) and service structure. For instance, if we have three servers on disposal for service provision, it is possible to make backup system. In that case, root-cause information about critical problem on first and second server will not make any impact on service quality. Those servers are in AND relation: problem root-cause should be received for all three servers in order to make service impact.

Generally, service structure should be presented in service-tree, where information from network monitoring and performance management will be sticked on tree leafs. Failured status from leafs will be propagated to "higher" levels, according to predefined service structure.

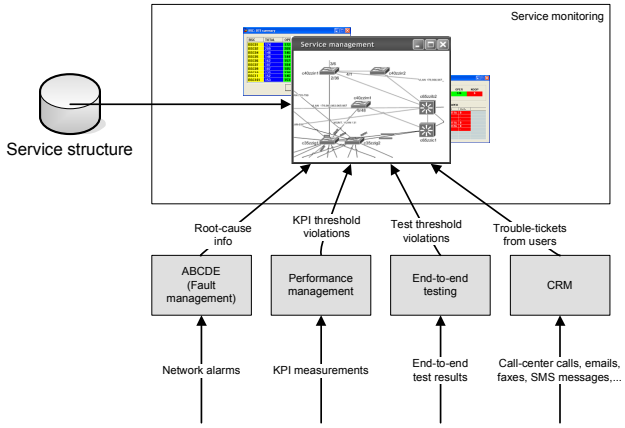

Fig. 4. *Service monitoring architecture with ABCDE included*

Service structure on figure 5 is presented with two different nodes on 1st layer. Every node can be in operational or failured state. Service is in operational state if all nodes from 1st layer are in operational state. Otherwise, service is in degraded/failured state.

On 1st layer, nodes present "links" to 2nd layer. It means, nodes don't present concrete service resource; rather, nodes are "symbols" that can be explored to 2nd layer, where concrete network resources can be shown (in this case, we have defined 2 levels only. However, number of levels is not limited).

On 2nd layer, eventual root-cause information or KPI threshold violations are "sticked" to concrete network resource. According to that information, it is possible to make "aggregate" status of the 2nd layer that will be propagated to appropriate node on the 1st layer.

There is possibility to define propagation (or status aggregation) rules. For instance, propagate failure if all resources at the same level are in failured state (AND), or propagate failure if at least one resource is in failured state (OR).

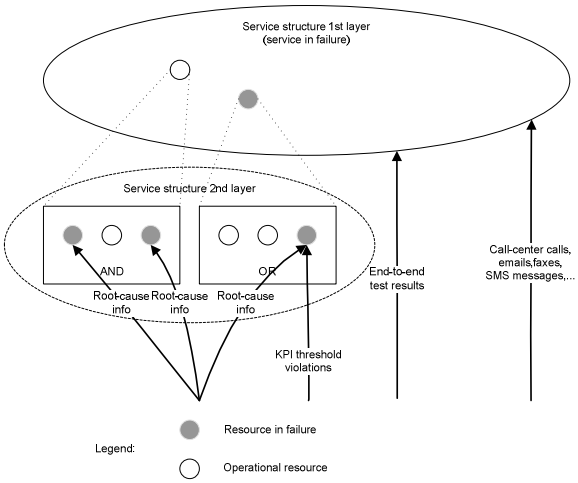

Fig. 5. *Service structure: general approach*

CRM information as well as end-to-end test violations are "sticked" directly to the service; however, it is possible to define nodes for that information on 1st layer also.

Service structure description may cover service local nature also. Namely, service can be in failure state, but not for all users. Failure can be localized, and appropriate service structure description can allow service monitoring operators detection of locally oriented service failures.

Investigation of possible ways for presentation of service structures, that will be realistic for implementation also, will be subject of our future works. It is necessary step for experimental proof of ABCDE role in service monitoring.

IV. Implementation

Based on service monitoring architecture shown on figure 4, experimental tool for service monitoring was developed (figure 6). Tool is launched in telecommunication network environment at one GSM operator in Croatia. Graphical user interface is based on HP Open View Network Node Manager (NNM), with number of utility applications connected via NNM API (Application programming interface).

Sources currently included in service monitoring are root-cause information (fault management), performance management and end-to-end testing results. Trouble-ticketing information will be included in future work.

Root-cause information is integrated using SNMP northbound interface from existing fault management system. Alarms coming from fault management system are filtered and correlated. Performance data are collected by parsing of MML (Man-Machine Language) outputs, periodically launched for interrogation of performance indicators from different network elements. End-to-end testing results are also integrated through SNMP northbound interface.

During exploitation phase, detection of service degradation was result of data from different sources (fault and

performance management, end-to-end testing). When service degradation was detected, we have checked data from different sources. General conclusion is that all sources contained data indicating service degradation, but some sources indicated service degradation with time delay. For instance, after BSC crash, fault data came with even 10 minutes delay, while performance data collected periodically shown service degradation almost immediately.

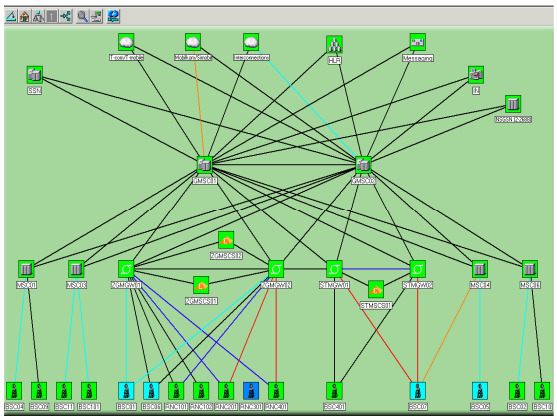

Fig. 6. *Experimental tool for service monitoring – GUI part*

Finally, after service degradation was detected, data from all sources gave complete picture. For instance, after storm in some geographic area, fault management data indicated possible service degradation, since number of network elements reported critical alarms. However, performance data gave quantitative aspect of service degradation – how many base stations (users) were out of service during BSC crash.

V. Conclusion

In this paper we continue the research of alarm correlations [2], [7] and [11]. Last step of our research was creation of ABCDE – Alarm Basic Correlations Discovery Environment. We have used test data from real telecommunication network, in order to make experimental proof of this concept.

According to [13], we have achieved great number of reduced incoming network alarms, with opened possibility for "high-level" alarm correlations. It has leaded us toward determination of network problem root-cause. That information is of great importance for network personnel monitoring telecommunications network. We have decided to try to use it in service monitoring process too.

In this paper we have presented first, general idea of ABCDE usage in service monitoring. Together with data from other sources, ABCDE-detected root-cause information is very useful for integration into service current status information, according to the service structure. Data from other sources were included in service monitoring, and experimental tool was developed and launched at one GSM operator in Croatia.

Further research efforts should be invested into the full implementation of proposed architecture. First logical step should be well-defined way of presentation of service structure.

References

[1] Kunštić, M., O. Jukić and M. Bagić, "Definition of formal infrastructure for perception of intelligent agents as problem solvers", *Proceedings on International Conference on Software, Telecommunications and Computer Networks*, Nikola Rožić and Dinko Begušić (ed.), Split, 2002.

[2] Jukić, O., M. Kunštić, "Network problems frequency detection using Apriori algorithm", *Proceedings of the 32rd International Convention MIPRO 2009*., Golubić S. et al. (ed.), pp. 77-81, Opatija, Republic of Croatia, 2009.

[3] Goethals, B., "Survey on frequent pattern mining", *Department of Computer Science, University of Helsinki, Finland*, 2009.

[4] Agrawal R., T. Imielinski and A.N. Swami, "Mining association rules between sets of items in large database", *Proceedings of the 1993 ACM SIGMOD International Conference on Management Data*, P. Buneman and S. Jajodia (ed.), ACM Press, 1993.

[5] Kowalski, R., Logic for problem solving, North Holland, New York 1979.

[6] Udupa, K.D., *TMN – Telecommunications Management Network*, McGraw-Hill Telecommunications, New York, 1999.

[7] Jukić, O., M. Kunštić, "Logical inventory database integration into network problems frequency detection process", *Proceedings of the 10th International Conference on Telecommunications CONTEL 2009*., Podnar Žarko, Ivana; Boris, Vrdoljak (ed.), pp. 361-365, Zagreb, Republic of Croatia, 2009.

[8] Burns, L., J.L.Hellerstein, S.Ma, D.J.Taylor, C.S.Perng, D.A.Robenhorst, "Toward Discovery of Event Correlation Rules", IBM T.J. Watson Research Center, Hawthorne, New York USA

[9] ITU T, *Recommendation X.733: Alarm Reporting Function*, Geneva 1992.

[10] Garofalakis, M., R. Rastogi, "Data mining meets network management – The Nemesis project", *Bell Laboratories, USA*, 2001.

[11] Jukić, O., M. Špoljarić, V. Halusek, "Low-level alarm filtration based based on alarm classification", *Proceedings of the 51stInternational Symposium ELMAR 2009*., Grgić, Mislav et al. (ed.), pp. 143-146, Zadar, Republic of Croatia,2009

[12] Costa, R., N. Cachulo, P. Cortez, "An Intelligent Alarm Management System for Large-Scale Telecommunication Companies", EPIA 2009, L. Seabra Lopes et al. (ed.), pp. 386-399, Berlin 2009

[13] Jukić, O., M. Kunštić, "ABCDE – Alarm Basic Correlations Discovery Environment", *Proceedings of the 33rd International Convention MIPRO 2010*., Golubić S. et al. (ed.), pp. 298-303, Opatija, Republic of Croatia, 2010. – Awarded paper