Three Common Fault Management Flaws …and how to correct them

Jaime Colom, SE Manager Infosim® US, shares his experience on common flaws of Fault Management and indicates how they can be corrected by a 3rd generation Unified Network Management System

Jaime Colom, SE Manager Infosim® US

Jaime Colom, SE Manager Infosim® US, has more than 25 years of experience in the IT and Telco market. He is a System & Pre-Sales Engineer with strong sales, technical, and project management skills, backed by comfort and success in customer-facing roles. Jaime has a lot of experience working with account managers, program managers and large sales organizations to understand the needs of the client, and provide technical support during contractual negotiations.

Flaw #1: Inadequate MTTR efficiency

Commonly, a huge amount of the Mean Time to Repair (MTTR), i.e., the time that network engineers and administrators spend to "fix" a problem in their network, is actually not really related to fixing this problem by applying an adequate solution but rather to finding out where the problem originated from. The more complex a system gets, the more complex it is to actually locate the problem before being able to start solving it. This, in general, leads to an inadequate MTTR efficiency, i.e., too much time is spent on things that could have possibly been automated.

Network managers and operators have traditionally performed failure management by simply using network status monitoring tools. This methodology, although useful, has less accuracy nowadays due to the complexity of interconnected networks. For this reason, operators must be assisted in their tasks by also monitoring in parallel events from all sorts of devices.

An event is an unsolicited message from a device, typically indicating a problem in the system that requires attention. A single fault may produce a cascade of events from the affected network elements. In fact, a fault can easily lead to another one in a chain reaction fashion; thus, it increases innumerous events masking the really important ones. This phenomenon is better known as an Event Storm.

As a network manager or operator, it is imperative to navigate through such a storm of events and be able to find the root cause in the shortest time possible.

A good discovery engine should have a mechanism that prioritizes each device it discovers and assigns a weight-value to that device that automatically maps its hierarchical relationship and overall importance within the network topology.

This device topological weighting can be used as a base for creating automated dependency rules among different elements in the network and the IT infrastructure. Based on this automated process a full Root Cause Analysis (RCA) is possible.

A state-of-the-art fault management system automatically correlates alarms and events to determine their root cause. This reduces the MTTR and downtime of mission-critical applications.

Altogether, the RCA allows the user to spend more time on fixing the problem rather than spending time finding where the problem originated from.

Flaw #2: Lost in Translation or "not seeing the wood for the trees"

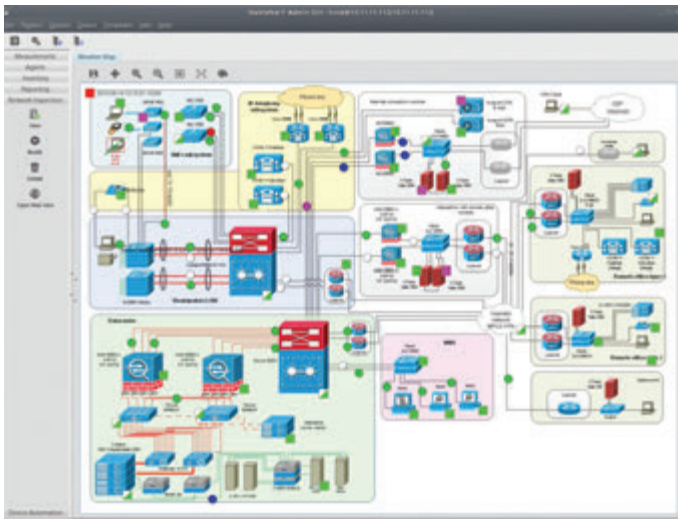
As just explained, a key element for an efficient Fault Management process and for a reduction of the MTTR is the automated correlation of events and alarms to provide a fully automated RCA. This, however, directly leads us to the next common flaw in Fault Management. A prerequisite of being able to automatically correlate events and alarms is, of course, to have the right event data as input for the analysis. This event data commonly consists of syslog and trap information from various distributed systems that needs to be collected, interpreted, and stored adequately for further processing.

What sounds quite clear in theory, turns out to be extremely challenging in practice. Experience shows that syslog and trap information highly differs across different platforms and vendors. The differences can be of various types: metrics are provided in different number formats or even different units or information might be provided by some event sources and omitted by others. Furthermore, text messages might be provided in different languages (English, German, etc.). That way, one sometimes feels literally lost in translation. The challenge is further exacerbated as often devices or systems do not offer a lot of configuration options for how the syslogs or traps should look like or which of them should be sent and which not. An extreme – yet not unusual – example are devices that offer only two "debug levels" for syslogs: they can either send only very limited information that is not enough at all or send out all information which is far too much and prevents from seeing the woods for the trees.

Looking at all this together, the legacy Network Fault Management approach has a built-in dilemma: Scalability vs. Aggregation. On the one hand, it is unfeasible to setup rules for all possible event information while on the other hand, not having enough rules will leave NOC personnel with insufficient data to troubleshoot complex scenarios.

A two-fold solution addresses this dilemma:

1. "Distributed Trap & Syslog Preprocessing" helps to shed light on the vast amount of trap and syslog information sent out by the devices. Information coming from various devices can be pre-aggregated adequately before being further processed. Among others, fine-grained filters can be defined to separate the important from the unimportant event information. Transformation scripts can be used to convert different metrical units or to translate text to a common language.
2. Once the traps and syslogs have been preprocessed, the most important information should be extracted from the filtered events. For some event types, it might be enough to count the number of appearances. For others, it might be necessary to do calculations on the content of some of the event information fields. Yet for others, textual content might have to be interpreted with regard to alarm severity or impacted users/services. Based on a customizable master rule set, a fault management system should provide a mechanism which can robotically expand and contract rules to count event numbers, do calculations on event information or interpret text content. This way, troubleshooting data is kept at optimum levels constantly without human intervention. Based on this, it is also possible to automatically generate tickets and report alarms raised by dynamically generated rules.



Automated Root Cause Analysis (RCA) for complex service maps with StableNet®

Flaw #3: Friction losses through silo-based management solutions

Having reached a correct processing of events and their correlation by an automated RCA, still one challenge remains that can impact the MTTR and harden the work of network engineers and administrators: the seamless integration of fault management with other service and network management workflows. If such a seamless integration cannot be guaranteed, a lot of friction losses are to be expected. Experience shows that this is actually the case in many companies using a set of different silo-based management solutions rather than one integrated solution.

In the following, three key examples of such integrations are briefly discussed:

1. Integration of Fault Management and Configuration Management: An enormous percentage of all IT & network problems actually originates from human errors. Therefore, a fault management solution should offer a fully integrated possibility to not only look at alarms and events related to devices but to also check their configurations and recent configuration changes. In case this is part of the solution to the identified problem, the management system should furthermore allow to directly interact with the devices, e.g. to restore previous configurations.
2. Integration of Fault Management and Performance Management: Throughout the last years, network services have become more and more complex running across disparate technologies, vendors and device types on a multitenant basis. However, the tools of network engineers/administrators often still limit their view to the device level and the monitoring and event data available from there. They often do not even know about the impact of the current problem to a higher level, like impacted business processes, services, etc. Therefore,

- a fault management solution should fully integrate with performance management to solve this issue and provide a holistic end-to-end service visibility and corresponding impact analysis.

3. Integration of Fault Management and Customer Service Management: Most of the system or network problems that are currently worked on involve a lot of communication between different engineers, engineers and the customer support team, as well as the customer support team and the end customers. Information is sent back and forth, tickets are updated, and information is copied and pasted. This is not just time consuming but also very error-prone. Therefore, a fault management solution should integrate with customer service management/ticketing system to automatically update tickets based on the current network/alarm status and to mitigate as much as possible any manual processes.

Why Infosim® StableNet® makes the difference

StableNet® is the ideal choice to overcome the described flaws. As a 3-in-1 Unified Network Management solution, it combines Fault, Performance, and Configuration Management in a single software – integrated by design with a unified underlying data structure.

The concepts of RCA, syslog preprocessing and filtering, as well as the integration of the different network management areas Fault, Performance, and Configuration are not completely new anymore and also offered by a lot of other solutions out there.

So, what is the key difference of StableNet® to all those other solutions?

1. In StableNet®, the RCA is completely automated, i.e., it automatically deduces all the necessary dependencies and RCA rules from discovery information, network topology, as well as the Infosim® long term best practices. That way, the setup of any manual rules is not a "must have" anymore, but just a "nice to have" for reaching additional customization.
2. The Distributed Trap and Syslog Preprocessing as well as the StableNet® Dynamic Rule Generation offer a generic module to process, transform, or filter any kind of trap and syslog information agnostic to the underlying vendor and device type. The functionality can be obtained by a onetime license fee (independent of the number of components considered) and all the processes can be defined by the users directly without a need of customized implementation efforts by the software vendor. This on the one side adds a lot of flexibility and on the other side saves a lot of licensing fees that in other solutions often have to be paid by number of devices and on a per vendor/device type base.
3. Many solutions offer a set of various tools to cover performance, fault, and configuration management that has to be put together and integrated as far as possible. Infosim® StableNet® is an integrated solution by design. It has been developed from ground up on a single code base and data structure and covers all the areas of performance, fault, and configuration management in one single solution with one licensing module and one single installation.

For the above mentioned reasons, already many customers from various business areas have selected StableNet® as their preferred choice for fault management. The high degree of automation and customizability notably increases the MTTR efficiency and significantly speeds up common network management workflows.

For further information on Infosim® products and services, please visit: www.infosim.net or email: info@infosim.net