HP OEMF: Alarm Management in Telecommunications Networks

This article explains the HP OpenView Element Management Framework concept, which is based on the HP OpenView Fault Management Platform (FMP) and complements the functionality of the FMP to provide an integrated network management solution. This article also explains the FMP, which facilitates efficient management of alarms in a telecommunications network, and the open APIs provided in the FMP, which allow seamless integration with other applications.

by Sujai Hajela

There has been an unprecedented growth in the telecommunications industry around the globe. The rapid evolution of new technologies, the offering of a broad spectrum of data services, and the need to have fast access to information are some of the factors that have contributed to a tremendous increase in the number of subscribers to telecom services. This has imposed great demands on the telecommunications networks of both public and private operators. To keep up with the demand, telecom operators are expanding their existing infrastructure at a hectic pace. Furthermore, deregulation of the telecommunications industry has led to the emergence of a number of private service providers, and this has created keen competition within the industry. A good quality of service at an economical price has become a key factor for service providers to increase their customer bases.

Telecommunications Management Network

Offering a high quality of telecom services and at the same time generating high revenues requires efficient management of telecommunications networks by the service providers. The Telecommunications Management Network (TMN) defines activities that aid in managing a telecommunications network. According to ITU-T Recommendation M.3010, a TMN is intended to support a wide variety of management areas including planning, installation, operations, administration, maintenance, and provisioning of telecommunications networks and services. The following five functional areas have been identified in TMN (ITU-T Recommendation M.3400):

- Fault management
- Configuration management
- Performance management
- Security management
- Accounting management.

Fig. 1 shows the TMN functional blocks and components. The TMN architecture consists of the functional architecture, the information architecture, and the physical architecture. The TMN functional architecture defines the following blocks:

- Operations systems function (OSF)
- Mediation function (MDF)
- Network element function (NEF)
- Workstation function (WSF)
- Q adapter function (QAF).

The TMN information architecture defines the information exchanged between these functional blocks.

The TMN physical architecture provides a means to transport and process information. The physical architecture is made up of the following types of physical components:

- Operations system (OS). Performs OSF.
- Mediation device (MD). Performs MDF.
- Q adapter (QA). Performs QAF, that is, connects network elements and operations systems with noncompatible interfaces to OSI Qx and Q3 interfaces.
- Data communications network (DCN). Performs data communications function (DCF), which is used by the TMN functional blocks to exchange information.

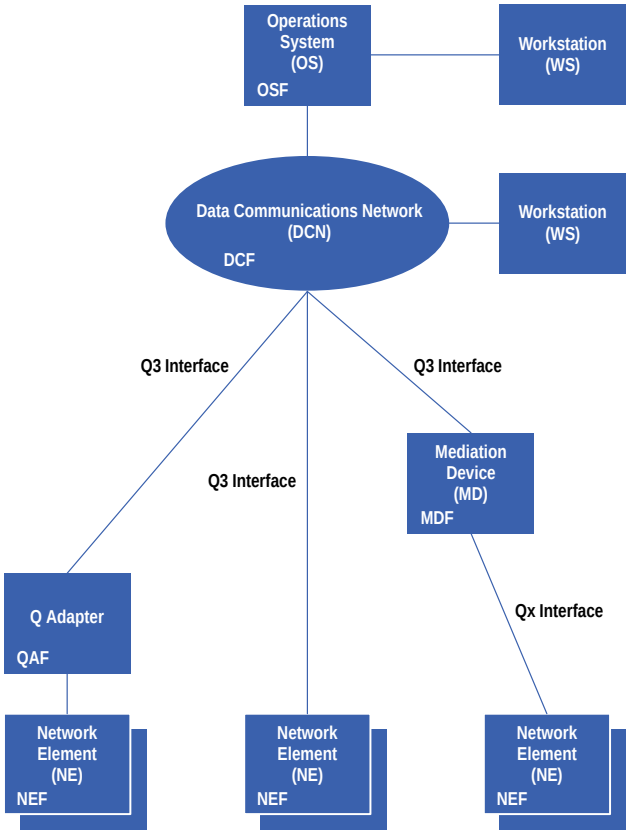

Fig. 1. *Telecommunications Management Network (TMN) functional blocks and components. (F = function, e.g., OSF = operations system function.)*

- Network element (NE). Performs NEF.
- Workstation (WS). Performs WSF.

OpenView Element Management Framework

The HP OpenView Element Management Framework (OEMF) aims to provide a set of management activities defined in ITU-T Recommendation M.3400 to facilitate efficient management of a telecommunications network. The functional areas covered within the OEMF are fault management (including trouble management), performance management, and other third-party applications to complement the existing set of applications under the OEMF umbrella—for example, configuration management and asset management.

OEMF is an open system that makes possible the detection, isolation, and correction of abnormal operation of the telecommunications network. OEMF consists of the HP OpenView Fault Management Platform (FMP) integrated with the Trouble Ticketing System provided by Remedy and the Performance Management System from Metrica. Other third-party applications for inventory, asset, and configuration management have also been integrated. Integration with test and measurement products like HP AcceSS7 further enhances the OEMF functionality.

Fig. 2 illustrates the physical architecture of the OEMF. OEMF has a distributed architecture in which different management activities can reside on different servers or on the same server. OEMF offers application availability, that is, if one of the management activities ceases to function, the operator can still execute the functionality provided by the other applications.

In the TMN hierarchy, OEMF resides between the network management level and the element management level (Fig. 3). It can manage the network elements directly or can be interfaced to an existing element manager to manage the network. Providing this flexibility to OEMF are a rich mediation service and APIs (application programming interfaces) for integrating with customer-specific data collection mechanisms.

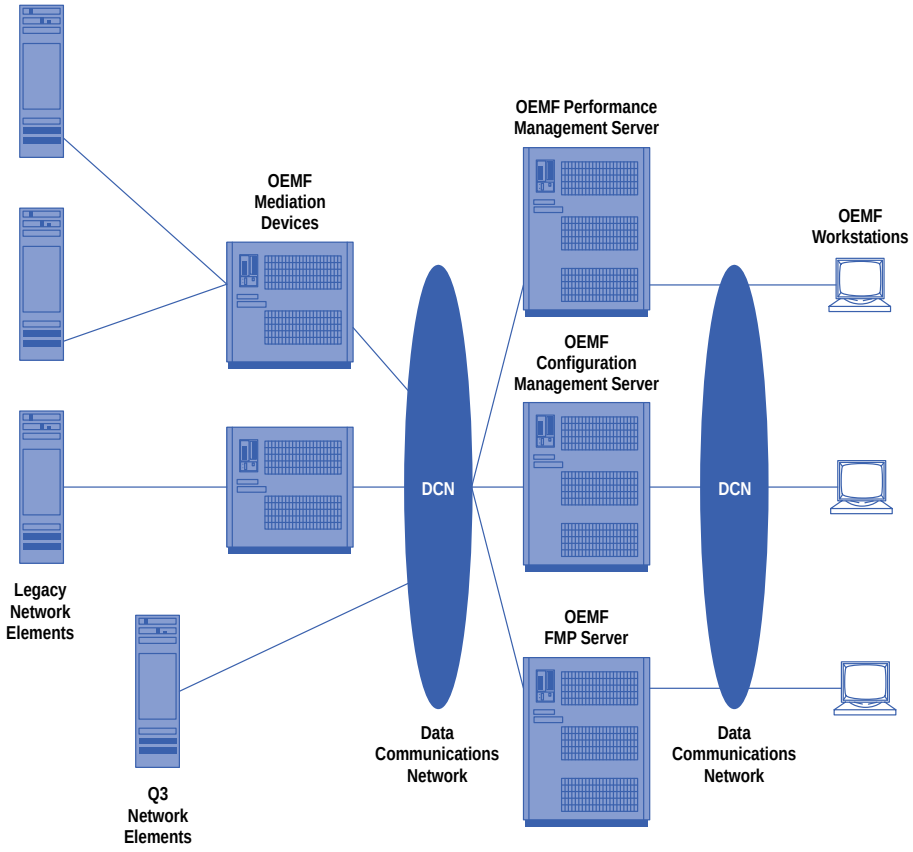

Fig. 2. *Physical architecture of the HP OpenView Element Management Framework (OEMF). FMP is the HP OpenView Fault Management Platform.*

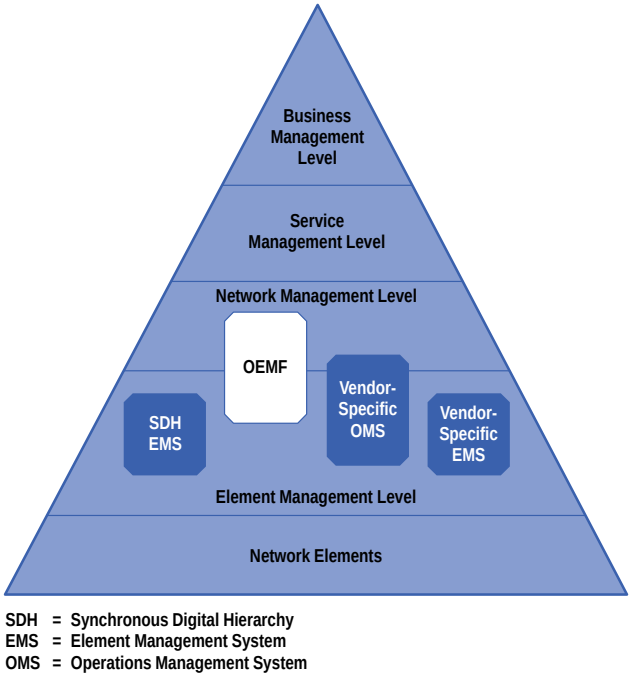

Fig. 3. *In the TMN hierarchy, the HP OEMF resides between the network management level and the element management level.*

Fault Management Platform

The FMP is a fault management platform that provides utility tools for managing alarms from multivendor devices and network element managers. It is based on the HP OpenView Distributed Management Platform. It has an extremely open architecture, which facilitates a seamless integration of third-party applications, as manifested by the OpenView Element Management Framework described earlier. Fig. 4 illustrates the FMP functional blocks. The main components of the FMP are the mediation device block, the FMP server block, and the graphical operator interface.

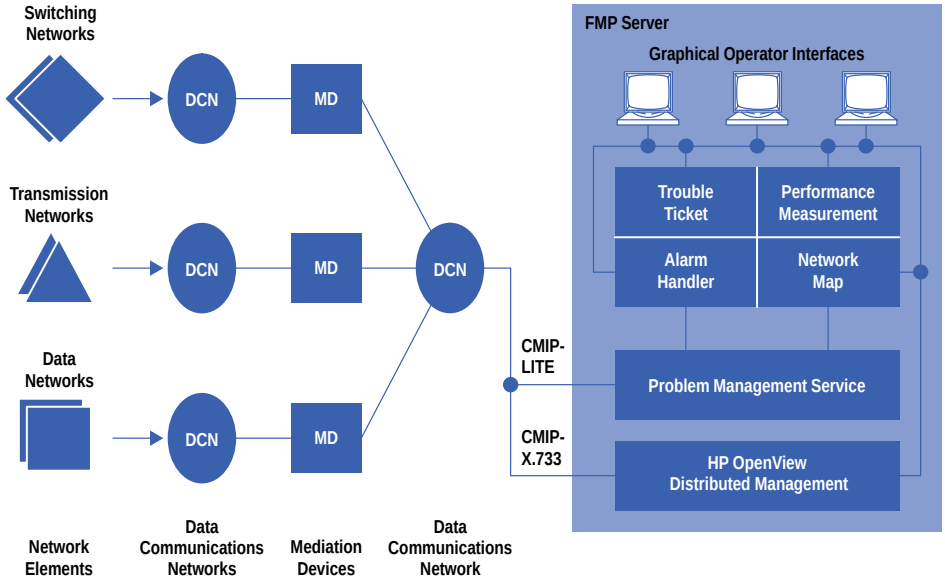

Fig. 4. *Functional blocks of the fault management platform (FMP) of the OEMF.*

The mediation device block provides the mediation and Q adapter functions and the FMP server provides the operations systems function. The mediation device logs, formats, filters, maps, and finally correlates all alarms it receives from network elements into ITU-T X.733 alarm reporting format and sends these alarms to the FMP server for alarm management. The mediation device can send the alarms to the FMP server using the CMISE protocol over the CMIP stack provided by the HP OpenView Distributed Management Platform, or optionally, using the CMIP-LITE protocol (an FMP representation of the X.733 alarm report) over TCP/IP.

The FMP server performs the problem condition management services. It provides graphical operator interfaces to aid in the management of the alarms being received from the network elements (which are performing the network element function). These graphical interfaces provide the means to interpret TMN information for the management information user. They perform the workstation function.

The FMP provides the fault management activities in a telecommunications network. However, to manage a telecommunications network, other management activities such as trouble management, performance management, and configuration management are also required. This requirement contributed to the OEMF concept, which allows a broad spectrum of best-in-class applications, regardless of manufacturer, to be integrated with FMP to provide an integrated network management solution. This integration is made possible by a range of open APIs provided in the FMP. The HP OpenView Distributed Management Platform APIs further enhance the integration capabilities.

Mediation Device Block

The mediation device logs raw alarms, formats and filters alarms from events, correlates these alarms, and then forwards them to the FMP server in the X.733 alarm reporting format. The mediation function is extremely important as the FMP server receives and manages alarms in a heterogeneous, multivendor, multinetwork environment in which the network elements send events in varying formats. Fig. 5 illustrates the functional blocks within the mediation device.

The mediation device provides a set of data collectors, which collect data over RS-232, TCP/IP, and SNMP. Reports in X.733 format can be sent directly to the FMP server using CMIP protocol. For data collection over X.25 and other types of networks, customer-specific data collectors can be written using mediation device connection APIs. These data collectors forward the events to the event logging module, which logs them into raw log files. The event logging module forwards the valid events to the event formatting module, which parses the incoming events and then classifies and formats them into message classes based on the parsing rules defined in the configuration. These formatted events are then logged into message class files corresponding to the message classes. The event formatting module forwards the events to the event mapping module, which filters the alarms from events, converts the alarms into the X.733 alarm report format and forwards them to the event filtering and correlation module. The correlation module correlates repeated alarms, transient alarms, and related alarms for a network element. The FMP supports a two-stage correlation approach in which the correlation

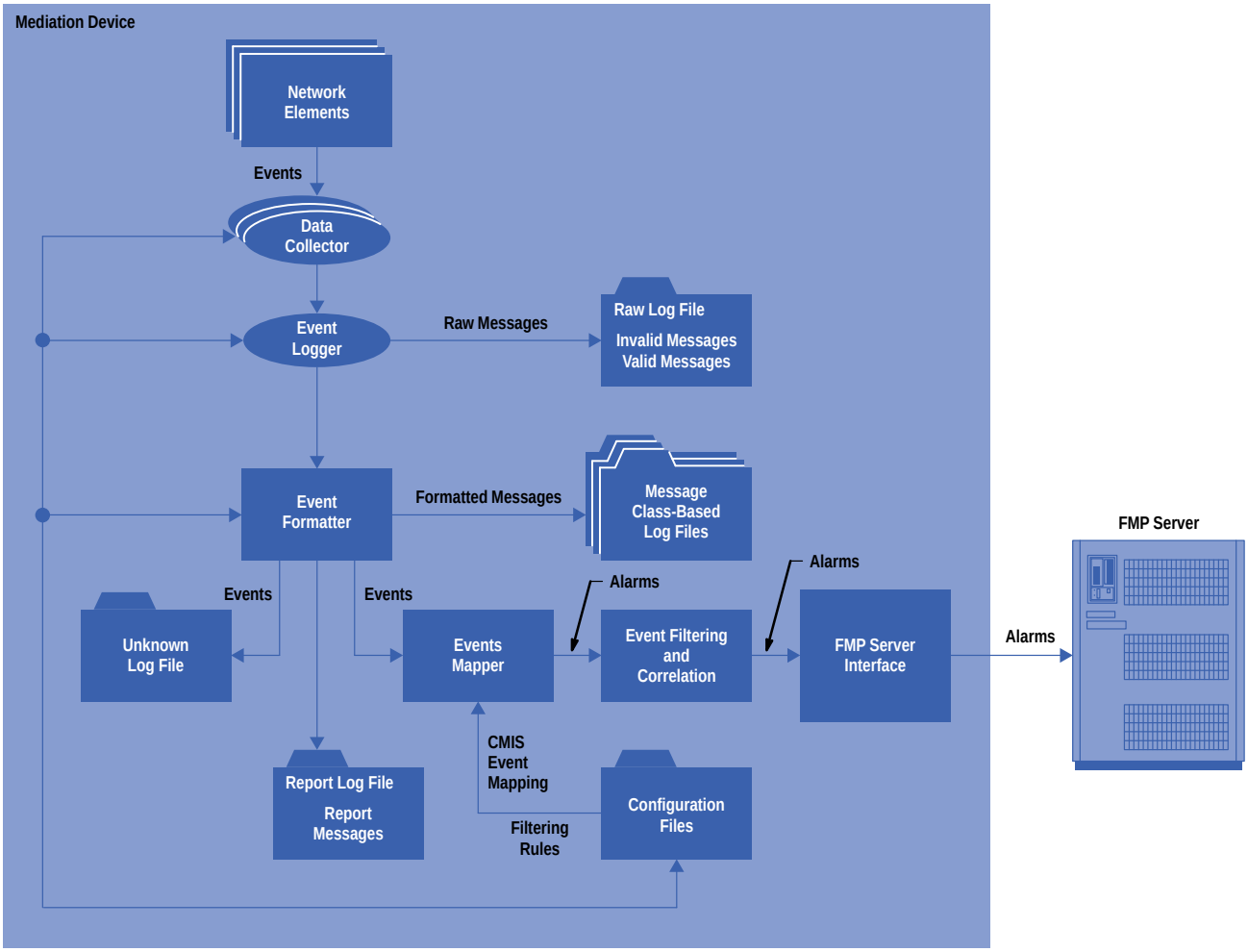

Fig. 5. *Mediation device functional block diagram.*

functionality is provided at the mediation device and at the FMP server. The correlation module forwards the alarm to the module interfacing to the FMP server. This module encodes the alarm into CMIP and sends it over the CMIP stack provided by the HP OpenView Distributed Management Platform or optionally (depending upon the configuration) encodes the alarm into CMIP-LITE and sends it over TCP/IP to the FMP server.

Correlation in the FMP

The FMP allows two stages of event correlation: one at the mediation device level and the other at the FMP server level. The correlation at the FMP server is done across the network being managed because the server has access to the topology database. The correlation at the mediation device is restricted to the network elements to which the mediation device is connected. The correlation at the mediation device level is done primarily to prevent not-so-important data from being forwarded from the mediation device to the FMP server.

The FMP has two types of correlation: repeated/transient correlation and root-cause/related correlation. Repeated/ transient correlation correlates alarms that are identical (they may have different severities) and are being emitted continuously by a network element. Root-cause/related correlation correlates alarms that have occurred because of a root-cause alarm and are not as important to the operator.

Let's take an example of root-cause/related correlation in a GSM network. Assumptions:

- MSC-1 is connected to BSC-1 which is connected to BTS-1 through BTS-4.
- An alarm A:MSC-1 (that is, an alarm of type A:MSC at MSC-1) causes an alarm B:BSC-1 (an alarm of type B:BSC at BSC-1).
- The alarm B:BSC-1 causes an alarm C:BSC-1.
- The alarm C:BSC-1 causes alarms D:BTS-1, D:BTS-2, D:BTS-3, and D:BTS-4.
- Of all these alarms, only A:MSC-1 is significant.

Fig. 6 illustrates the scenario. Based on the above assumptions, if an alarm A:MSC-1 occurs, the operator will also receive the alarm B:BSC-1 which will further cause alarms C:BSC-1 and D:BTS-1 through D:BTS-4. Even though the real problem is with MSC-1, the operator receives numerous alarms, many of which are of no significance.

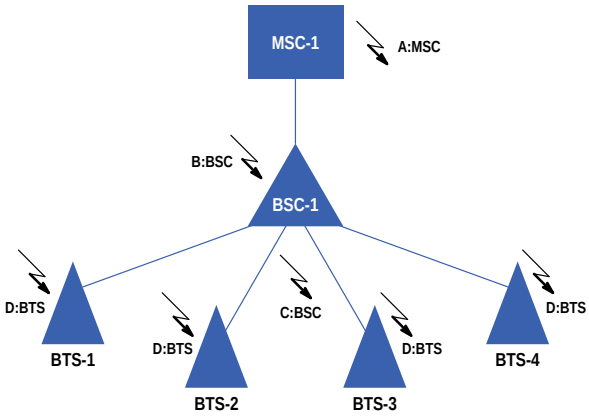

Fig. 6. *Scenario of alarm generation in an example GSM network without correlation. Many extraneous alarms can be generated in addition to the root-cause alarm.*

Let's specify the correlation rules to be as follows (the format of the event correlation rules specification has been simplified to explain the concept):

```
Rule 1: ROOTCAUSE : A:MSC RELATED : B:BSC
Rule 2: ROOTCAUSE : B:BSC RELATED : C:BSC
Rule 3: ROOTCAUSE : C:BSC RELATED : D:BTS

Correlation window: 20 seconds
```

With these correlation rules in effect, the operator will receive only the alarm A:MSC-1, which is the significant alarm. This behavior is illustrated below:

```
Correlation window (total elapsed time) = 20 seconds
|<------------------------------------------------------------>|
                                                               |
Arrival of A:MSC-1 -->|-->A:MSC-1 sent                         |
Arrival of B:BSC-1 --------->|                                 |
  (correlated by rule 1)                                       |
Arrival of C:BSC-1 ------------>|                              |
  (correlated by rule 2)                                       |
Arrival of D:BTS-1---------------------->|                     |
  (all BTS correlated by rule 3)                               |
Arrival of D:BTS-4 -------------------------->|                |
                                                               |
 End of Correlation Window---------------------->|
```

All alarms matching the correlation rules and occurring within the correlation window are subject to correlation. The correlation window can be a fixed or a sliding window. The arrival order of alarms is not important—in the above example, the alarms could have arrived in any order. As long as they arrive within the correlation time window, they get correlated. If a related alarm arrives before a root-cause alarm, it is held in the correlation module until the end of the correlation window. If the root-cause alarm does not arrive within the time window, the related alarm is sent out as uncorrelated. If the root-cause alarm arrives before the related alarms, it is sent out immediately. In the scenario above, the A:MSC-1 is sent out by the correlation module immediately and is not held back until the end of the correlation window.

Event correlation services are available in HP OpenView Distributed Management Platform 4.21 as an option (see ***Article 4***). Event correlation services further complement the event correlation provided by the FMP and, when integrated with the FMP, greatly enhance the correlation functionality of the FMP.

FMP Server Block

The FMP server provides problem management services. It logs, correlates, and distributes the alarms to the graphical operator interfaces. Fig. 7 shows the functional blocks within the FMP server.

An alarm is received from the mediation device (there may be one or more mediation devices) either in CMIP or CMIP-LITE by an interfacing module, which decodes and forwards it to the alarm logging module. The alarm logging module logs the alarm in the alarm database. This alarm is then passed to the alarm correlator module for correlation. This is the second

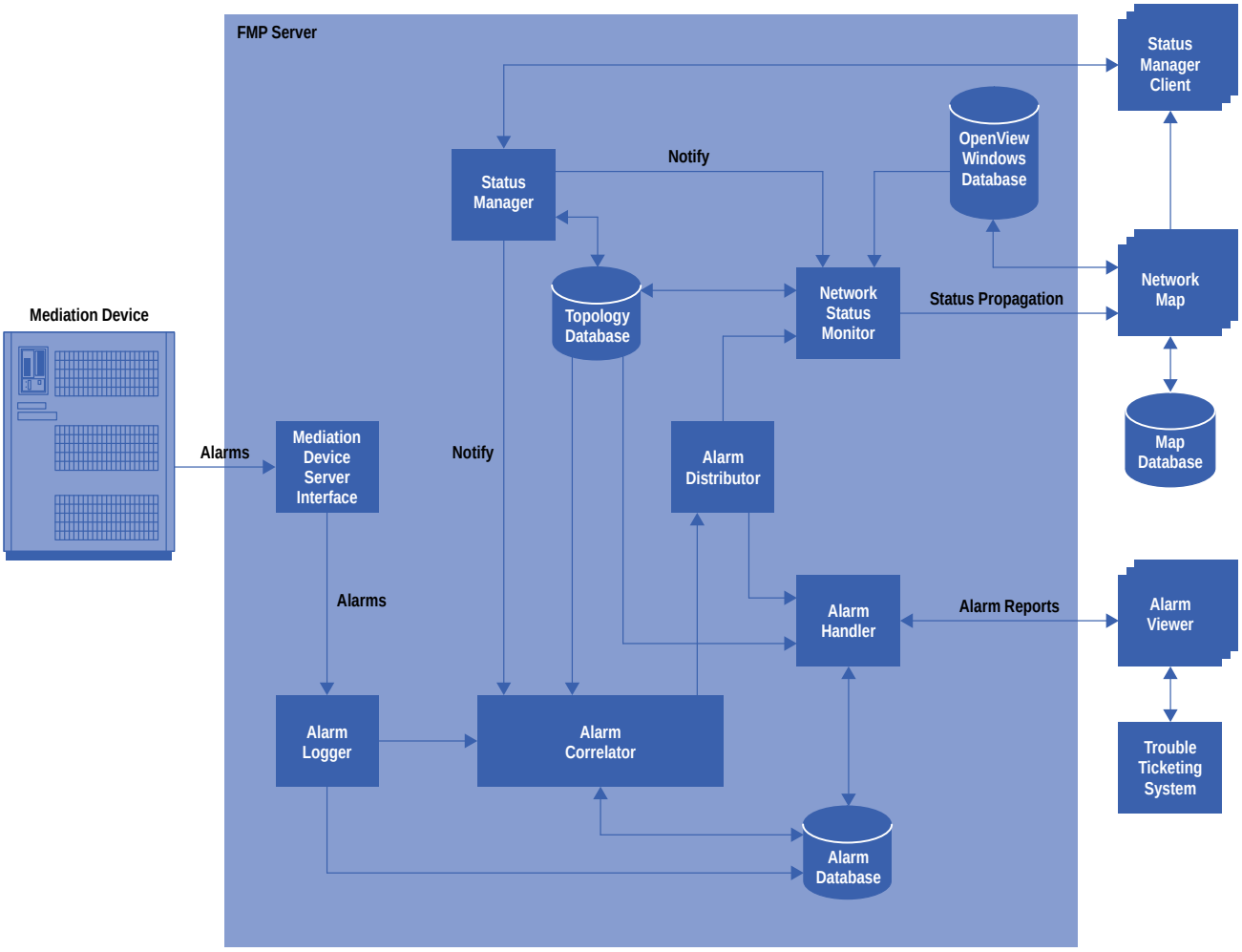

Fig. 7. *FMP server functional block diagram.*

stage of correlation, the first being at the mediation device. Correlation at the FMP server is performed across the network because the server has access to the topology information stored in the topology database, which resides at the server. After correlation, the alarm is distributed to the alarm handling module and the network status monitor module. The alarm handling module manages problems. Every alarm need not be a new problem—many alarms may be sent for the same problem. The alarm handling module identifies the alarm as belonging to a problem if it originates from the same network element and has the same probable cause and specific problem fields as the problem (these fields are specified by ITU-T X.733). The alarm handling module checks whether a problem condition already exists for the alarm received. If it does, then an update to the problem is sent to all the connected alarm viewers. Otherwise, a new problem condition is created and sent to the alarm viewers depending upon the span of control defined for each alarm viewer.

The network status monitor is responsible for status propagation according to the configuration rules. Depending upon the severity of the alarm, the network status monitor calculates and sets the status of the alarming network element on the network map. The network status monitor calculates the status of objects based upon the severity of objects both on the map and not represented on the map (*nonmap objects*). The need to support the concept of nonmap objects arises because in a telecommunications environment, the number of objects being managed can be very large. Also, the operator may prefer not to see all of the objects being managed on the map, but would want the status of the nonmap objects to be considered while calculating the status of the higher-level objects (in the containment hierarchy) that are represented on the map.

The status manager server (or simply status manager) facilitates the submission of outage schedules and maintains information regarding the outage of the network elements. Operators submit the outage information through the status manager clients.

Graphical Operator Interfaces

The FMP includes a set of utility tools that provide a graphical interface for the operator to manage the telecommunications network. The tools provided are the alarm viewer, the network map application, and the status manager clients for submitting outage schedules.

Alarm Viewer. The alarm viewer is an X11/Motif-based application that allows the operator to view the faults occurring in the network being managed and provides facilities to take corrective actions to handle these faults. Fig. 8 shows the alarm viewer window.

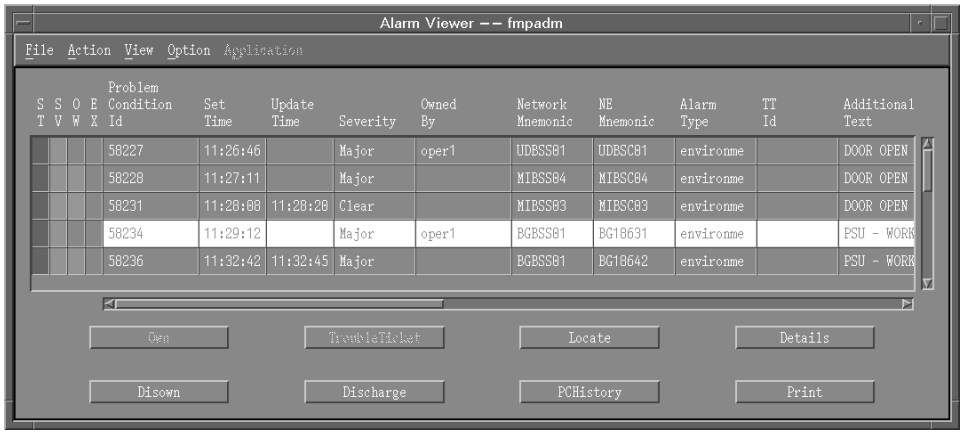

Fig. 8. *Alarm viewer window.*

The alarm viewer receives the problem condition from the alarm handling module. It allows the operator to select and perform the following actions on the selected problem condition:

- Own
- Disown
- Discharge
- Locate
- View Problem Condition History
- Create Trouble Ticket
- View Details
- Print.

By owning a problem condition, the operator acknowledges the presence of a fault. The operator can then locate the alarming network element on the network map and create a trouble ticket for the problem. Once the problem has been rectified, it can be discharged. On being discharged, the problem disappears from the alarm viewer. An audit trail is maintained in the alarm database regarding the actions performed on the problem. A list of all the alarms corresponding to the selected problem condition is displayed by clicking the Problem Condition History button. The alarm viewer can be configured to display only the fields in which the operator is interested. The Details button can be clicked to view the details of the problem condition. A hard copy of the selected problem condition can be obtained by selecting the print option.

The alarm viewer provides visual aids for quick identification of the severity of the problem. The columns in the problem condition row are color-coded and signify the outage, the severity, and the ownership status of the problem.

An operator can invoke an alarm viewer and perform actions on the selected problem conditions. However, operators need to be registered with the FMP server, and their spans of control, or *management domains*, need to be defined. Their control can be defined on the basis of the network instance, the network element class, and the network element instance. The alarm handler ensures that all the alarm viewers are consistent in case there is overlap in the operators' spans of control (it is common to have multiple operators responsible for the same network elements). For example, if a problem condition is owned at one alarm viewer, this owned status is propagated to other alarm viewers displaying this problem.

Alarm viewer menu options allow an operator to customize the alarm viewer. Operators can set their own sorting criteria for problem condition display. They can also set their own view preferences, for example to view problems from a certain network, view problems of a certain severity, view problems they own, and so on. These menu options offer flexibility and convenience to help the operators efficiently manage the problems occurring in the network.

The alarm viewer allows external and customer-specific applications to be registered with it. A problem condition can be selected and any of the registered applications can be invoked for the selected problem condition. This is an extremely useful feature which allows integration of the FMP with best-in-class applications from any vendor to complement the FMP functionality.

Network Map. The network map is an OpenView Windows application that displays the network being managed and the status of the network elements within it. The network elements are represented by icons and the colors of the icons represent their severity. The network map gets status updates from the network status monitor module. It allows the operators to navigate through the managed network and isolate the network elements generating the problem conditions.

It also allows updating of the topology either interactively through the menu options provided or programatically through the map loader APIs. The network map can be customized by creating logical views of the network. Thus, an operator can navigate through the whole network hierarchy while a manager can choose to look only at the status of the network at a higher level without going into the details of the network elements within the network.

The network map has the FMP registered as an application. The FMP menu option allows the operators to invoke various applications like the alarm viewer and the status manager clients.

Fig. 9 shows a network submap for an example GSM network.

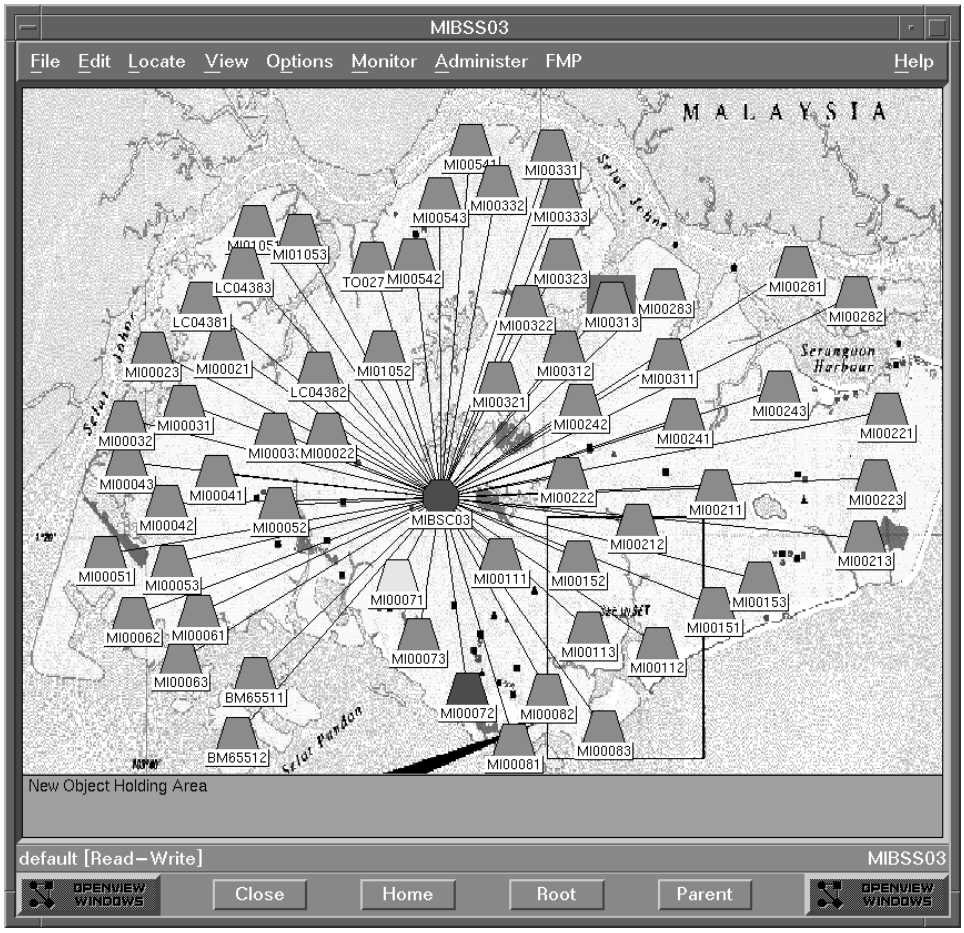

Fig. 9. *Network submap for a region in an example GSM network.*

Status Manager Client. The status manager client is an X11/ Motif application that allows an operator to submit outage schedules, that is, an operator can submit a proposal to put a network element out of service or restore a network element back into service. The operator needs to be configured for the status management capability to submit the outage schedules. The operator can specify the start and end times of the outages and whether the network element will be restored manually or automatically to in-service status. If an alarm is generated by a network element that is in the outage state, it is flagged by a different color in the first column of the alarm viewer. The FMP server can also be configured such that alarms from network elements in outage status are not sent to the alarm viewers at all. Information regarding the outage status of the network elements is maintained by the status manager server.

FMP Configuration

The mediation device has to be configured to be able to receive, format, and map events received from multivendor network elements in the X.733 alarm format. The object model of the network being managed, the different types of network elements, event correlation rules, operators' spans of control, and status propagation rules all have to be configured. Information regarding the mediation devices connected to the FMP server (the FMP server can be connected to more than one mediation device) and any customer-specific data collectors connected to the mediation devices also needs to be supplied.

FMP provides a screen-based GUI utility—the configurator—to aid in configuring the FMP and customizing it to manage a heterogeneous telecommunications network.

FMP Application Programming Interfaces

Throughout the design of the FMP, an open architecture and ease of integration were always given maximum importance. The FMP allows seamless integration with other applications as a result of its rich set of C and C++ APIs. The various APIs are described in the following paragraphs.

Data-Collector-to-Mediation-Device Connection APIs. These APIs can be used to write customer-specific data collectors to send events received from the network elements to the mediation device. Apart from data collection, these data collectors can also use these APIs to inform the mediation device regarding their operational status and to get information regarding the operational status of the mediation device.

High-Level Parsing APIs. High-level parsing APIs facilitate the validation of events received from the network elements. The data collectors can use these APIs to find the validity of an incoming event and flag the event as valid or invalid. This information is used by the event logging module in the mediation device to tag the event as valid or invalid in the raw log.

Pseudocode for a sample data collector using the data collector and high-level parsing APIs is as follows:

```
main()
{
   Open the network element port
   for(;;)
     {
       If (Data received from network element
           port)
         {
            //High-level parse the input data
            //received
            FaultBuf = HLPParse(Data Received)
            //Send the structure returned by the
            //HLPParse to the mediation device
            DCSendFaultToMD(FaultBuf)
         }
       //Inform the mediation device if the
       //network element port is not OK
       If (Error in network element port)
         DCSendPortStatusToMD(Port is not OK)
       //If the mediation device is shutting
       //down, shut down this data collector
       If (((msg = DCReceiveFromMD(control
             message from MD )) == MM_SHUTDOWN)
         ShutdownThisDC();
     }
}
```

Log APIs. The mediation device logs raw data received from the network elements. It then classifies these events into message classes and logs them in the corresponding message class file. The log APIs can be used to access these files. A number of applications can use the mediation services of the FMP and have the data collected at the mediation device in a format desired by them. These applications can then access this formatted information using the log APIs. Many interesting and useful applications can be written using the mediation and logging services provided by the FMP. An example is a raw log browser, which allows the operator to select an X.733 alarm from the alarm viewer and then extract and browse the raw alarm data corresponding to the selected alarm.

Application Registration APIs. These APIs allow the registration of external applications with the alarm viewer and facilitate passing information about the selected problem conditions to these applications. Application registration APIs can be used to integrate customer-specific applications. Once registered, the applications can be invoked from the Applications menu option of the alarm viewer. Taking the example of the raw log browser, the browser would first be registered with the alarm viewer. Then a problem condition can be selected and the raw log browser application can be invoked. Assuming that a raw log index is passed as some field in the X.733 alarm (e.g., rawlogindex as a part of additional information), this application can use the APIs to extract this index, which can be used to access the raw log. The trouble management system provided under the OEMF also uses these APIs. When a problem condition is selected and a trouble ticket is created for it, the trouble ticketing application uses these APIs to get information regarding the problem condition.

Problem Condition Management APIs. These APIs allow an application to interface to the problem condition management services. The alarm viewer uses these APIs. Customized alarm viewers, an automatic trouble ticketing application, an automatic problem condition discharge application, and other applications can be written using these APIs. Take, for example, an automatic problem condition discharge application for managing problems for which the switch normally does

not send a clear event. Based on certain criteria like the age of the problem condition, its severity, and so on, this application can be designed to discharge the problem condition automatically without any manual intervention from the operator.

Map Loader APIs. These APIs allow the customer's topology to be loaded into the FMP without having to add the objects manually into the FMP topology. This is extremely useful because the number of objects being managed can range anywhere from 4000 to more than 40,000. Map loader applications can be written to access the topology database of the customer and then populate the FMP topology using the map loader APIs.

Conclusion

The FMP APIs have led to the expansion of the number of products integrated under the OEMF umbrella. The FMP integrated with other best-in-class applications provides an enhanced telecom network management solution for efficient management of the multivendor, multi-equipment telecommunications networks of today.

Acknowledgments

I would like to acknowledge the members of the FMP team who have made it all happen, especially Chew Chye-Guan, Tok Wu-Chuan, Kuan Siew-Weng, Lin Chee-Kheong, Ho Yong-Boon, Vasu Sankhavaram, Prem N. Devadason, and not the least, project manager David Chua. Special thanks to the TMN, Integration Solutions Center, and Application Integration Center teams whose valuable suggestions have contributed to the growth of this product.

▶ Go to Article 4
▶ Go to Table of Contents
▶ Go to HP Journal Home Page