

A Generic Event-Driven System for Managing SNMP-Enabled Communication Networks

Aécio P. Braga¹, Riverson Rios², Rossana Andrade³, Javam C. Machado⁴, José Neuman de Souza⁵

¹Núcleo de Processamento de Dados – Universidade Federal do Ceará (UFC)
Campus do Pici, Bloco 901 – 60.455-770 – Fortaleza – CE – Brasil, aecio@ufc.br

^{2,3,4,5}Departamento de Computação – Universidade Federal do Ceará (UFC)
Campus do Pici, Bloco 910 – 60.455-770 – Fortaleza – CE – Brasil
{[riverson](mailto:riverson@ufc.br), [rossana](mailto:rossana@ufc.br), [javam](mailto:javam@ufc.br), [neuman](mailto:neuman@ufc.br)}@ufc.br

Abstract. In the area of monitoring communication networks, GEDSystem is a tool for supporting the development of programs driven to the management of domains that may show up when networks are monitored by SNMP. This tool is supported by an automatic modeling of data and events and automates the grouping of MIB objects from several SNMP agents, producing data structures that represent Domains and Events. The structures are automatically transformed into Tables and Views of a relational database. The views implement the perception mechanisms of the Events defined in the tool. The GEDSystem also provides a Monitoring Agent that automatically recognizes, collects and stores information about the Domains.

1 Introduction

After an analysis of several works in the area of network management, it is possible to notice that, in general, the used approaches try to identify and notice Events as a means for detection, diagnosis and correction of anomalies on the network. They make use, for instance, of probabilistic calculations and graphs as well as case- and rule-based reasoning. The methods are used in the following way:

- Probabilistic approaches are used in the control of the Quality of Service (QoS) and of the resource reservations in ATM networks [4], in the production of fault prediction alarms [11], in the identification of "signatures" from the network traffic and from the behavior of the managed objects [7], and in the anticipation of potential problems in a web server [9].
- Causality and dependency graphs are used in the determination of the simple causes of a chain of alarms or events [5] and [12].
- In [3], rules are formulated as thresholds whose definitions are based on time series of states of the objects in a network.
- Case-based reasoning is applied in maintenance support systems aided by a Knowledgebase of Events [1], in the integration of heterogeneous networks and in the correction of flaws on the nodes of a network [8]. The technique is also integrated into a Trouble Tickets Systems (TTS) architecture so that solutions to flaws based on past episodes that have occurred in a computer network can be formulated [6].

Briefly, these applications are fed by alarms or logs files. Starting from events, they attempt to foresee tendencies and anomalies, to detect, to isolate and to correct the original causes of the events and to map the physical observations with the states of the managed network objects. To accomplish the task, such techniques as thresholds comparisons, statistical analyses, graphs and historical information are used. On the other hand, these techniques employ, for instance, more algorithms in the routers, complex mathematical and statistical models, more layers in the management models etc., which increase computing and message overhead.

There is nowadays a great amount of network management applications whose computational characteristics vary tremendously. Unfortunately, the development of these applications does not take into account the management domains in general [2], in the sense that those approach lacks of a generic and automatic mechanism for preparing data models capable to represent these domains in such a way to facilitate the perception of events.

The main objective of this work is to specify and implement a Generic Event-Driven System for Monitoring Communication Networks (*GEDSystem*). *GEDSystem* is a tool for supporting the construction of Monitoring Systems for Communication Networks that provides:

1. an automatic way to define management domains;
2. an abstraction of the SNMP protocol in the non-reactive phases of the management process (communication, data structures etc.);
3. an automation of the processes of collecting, dating and storing information;
4. an automation of the perception of Events; and
5. an operational Web interface.

This paper is organized as follows. Section 2 depicts the event-driven network management; section 3 describes a generic management model; section 4 presents the *GEDSystem* tool; section 5 outlines some *GEDSystem*'s functionalities and finally the section 6 has the conclusion and further works.

2. Event-Driven Network Management

The main operations of network management consist of tracing, interpreting and manipulating events. In general, an event is defined as an anomalous condition observed in the operation of a network. Usually, these are problems that happen in the hardware and/or software of the nodes [12]. Regarding the SNMP protocol, the events or conditions can be noticed through the variations that happen in the management information stored in the MIBs of the Agents and/or probes of the RMON. When provided by an Agent, this information is obtained as regular instantaneous snapshots. In the case of RMON probes, it is obtained as groups of information collected along a period of time. In both cases, the Manager application is entrusted of periodically requesting the information that needs to analyze, and of executing some control action on the managed nodes.

The compilation of the information from the RMON and RMON2 probes is accomplished through package selection together with the scanning of their contents. The aim is to generate statistics and/or events in conformity with the definitions of events from the probes themselves. These events can trigger traps that send information from the RMON/RMON2's MIBs to some Manager.

As mentioned before, systems managed by SNMP are fed with information from the MIBs of the Agents and/or from the RMON probes. Each MIB is a general collection of information that change over time. In other words, they should be grouped, based on some supposed relationships, so that a Management Domain can be defined. The most immediate relationship than one can notice among them is time.

According to [10], the RMON and RMON2 probes use circular buffers. Consequently, the Manager application must have a more accurate control on the polling so that the buffers do not overflow, thereby avoiding the loss of previously compiled information. The probes, too, are computationally burdened with the execution of processes that generate statistics, events and traps. The buffers and processes could be left under the responsibility of another processing entity. This way, they would be freed from tasks other than "listening" to the interfaces and scanning packages.

3. A Generic Monitoring Model

Up to now, this paper has touched upon the data sources for management applications and the diversity of domains managed by the applications. This work proposes a tool to support the development of programs capable of managing domains deriving from SNMP-enabled communication networks, based on events.

The tool allows the gathering of information from the several nodes of the networks with SNMP agents. This operation results in the creation of a database to keep this information in a transparent way. In other words, it provides the modeling of data structures that represent management domains, so as to feed network management processes.

Dating attributes are also automatically included into the data models, indicating time of requisition and time of reception of the information from the modeled domains. This information can help, for instance, performance analyses, historical behavior and statistics procedures.

Rules that evaluate the possible variations of the grouped information can be defined and eventually used in the formulation of expressions that can represent several types of events. Additionally, Views of the repositories, based on the expressions, can be dynamically defined in such a way to provide a perception mechanism of the events. The feasible management processes can consult these Views in order to detect, diagnose and correct any anomaly in the monitored management domain.

The built tool provides a monitoring agent that automatically recognizes, monitors and stores the information from the management domains. This way, the applications can be exclusively dedicated to the management procedures, freeing themselves from doing polls. Their decisions are based only on the perception of the events implemented as database Views or on other analyses done on the information contained in the repositories. Small countless applications, therefore, can manage their own domain, facilitating the development of a Network Management System. The tool is based on such technologies as Domain Modeling, Event Modeling, Database Modeling, Web Modeling and Client/Server Modeling.

4. The GEDSystem Tool

A Generic Event-Driven System for Monitoring Communication Networks (*GEDSystem*) was born from the main issues involving a monitoring model explained in Section 3. *GEDSystem* is a tool composed of technological components produced by recent progresses in the areas of operating systems, web systems, databases systems, graphical user interfaces, programming languages, communication protocols and network management. *GEDSystem's* architecture is composed of one or more Clients in the first layer, a Monitoring Server in the second layer and one or several Information Management Servers (SNMP agents) in the third layer, as can be seen in Figure 1.

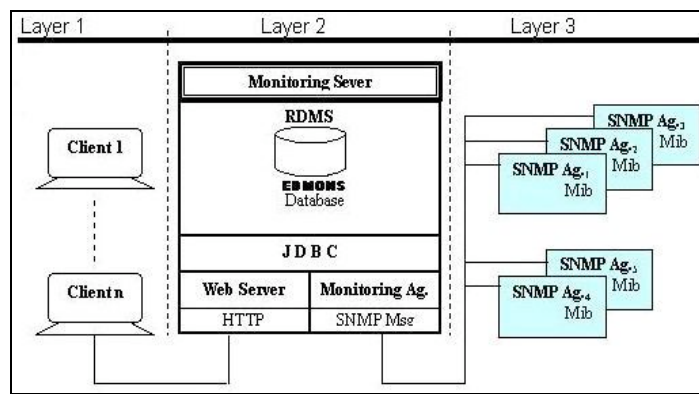


Figure 1. GEDSystem 's Client/Server Architecture

The First Layer - The Client Layer. A Client can be any computer whose Internet browser can access to a Java Virtual Machine (JVM). Such a browser is enough to provide the communication between the first and second layers of the architecture, additionally offering a graphical user interface that makes it possible for the network administrator to easily interact with the Monitoring and the Database Servers.

The Second Layer - The Monitoring System. The Monitoring Server is a computer running any operating system with a Web Server that supports Java Servlets and JDBC (Java Database Connectivity). The support for any RDBMS is also mandatory. To implement a prototype of GEDSystem, the Apache Server's Tomcat, AB's MySQL database server and Java servlets / JDBC and AdventNet APIs were adopted. In addition, the tool provides a Monitoring Agent implemented in the Java language. The functional base of the process will be shown later in this paper along with the definition of the Generic Event-Driven Data Model (MDGE) for monitoring communication networks. Through the MDGE, the Monitoring Agent accomplishes the following tasks: Collection of information of the managed elements' MIBs; Grouping of information in domains; Storage of information in a database.

The Third Layer - The Layer of the Management Information Servers. The third layer of the GEDSystem 's architecture is composed of the nodes of the network with

SNMP Agents. This way, the agents are Management Information Servers, which provide the values of the managed objects that are found in the MIBs. In other words, the second layer's server takes the role of a management Information Client.

GEDSystem's Database is based on the Generic Event-Driven Data Model (MDGE). It is composed by the supporting structures of the monitoring process, by the information obtained by the monitoring agent, and by the mechanisms used for noticing Events. In the database, data and historical events coming from SNMP Agents are stored as time series. By doing this, the database can help in the production of management reports about the behavior of the network.

The Generic Event-Driven Data Model (MDGE) represents a group of entities in the form of Tables whose attributes and relationships(Figure 2) specify which, where, when and how the managed Objects are collected and stored by the Monitoring Agent. These specifications represent the administrator's monitoring needs. Besides, MDGE in an event-driven model and, thus, allows the perception of the variations of the values collected at any moment by means of a certain restrictive rule of those values.

Some entities of the Model are created in a transparent and dynamic way, depending on the configuration of the monitoring process previously defined by the Administrator. The entities are the repositories of the values of the managed SNMP Objects.

MDGE is composed by the following entities: *Community*, *Domain*, *PrimitiveState*, *CompositeState*, *Index*, *Formula* and *GeneralTable*. In addition, an uncertain number of entities named *Repositories* are also part of the Model.

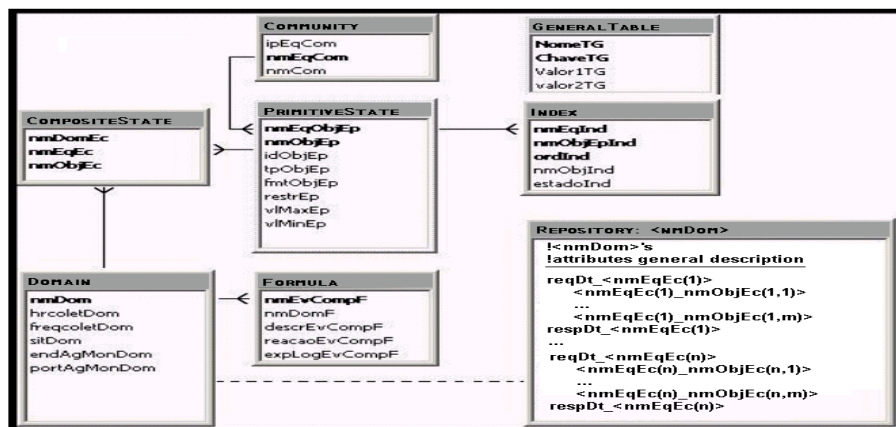


Figure 2. General view of MDGE's Entity-Relationship Model

Figure 3 shows the relationship among the entities that provide the bases for the automatic construction of the repositories. In the upper part of the illustration, one can see MDGE's entities whose relationships represent the definition of Management Domains. The lower part shows this relationship in the form of connections between tables that implement the entities presented in the upper part.

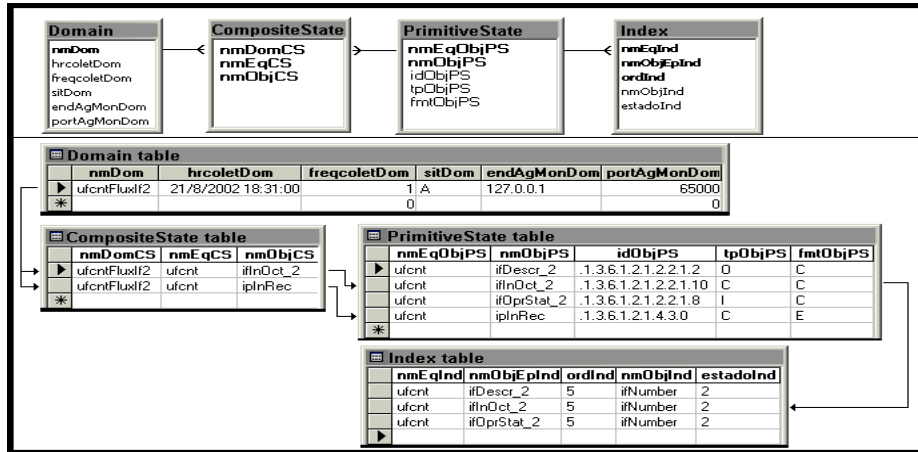


Figure 3 illustrates the definition of a domain called *ufcntFluxIf2*. This domain contains the objects *ifInOct_2* (columnar) and *ipInRec* (scalar), both of a node called *ufcnt*. This definition leads to the creation of a table, the repository of the domain being treated, in a database called *R_ufcntFluxIf2* with the following lay-out:

R_ufcntFluxIf2 : repository table	
Attribute Name	Data Type
reqDt_ufcnt	Date
ufcnt_ifInOct_2	Double
ufcnt_ipInRec	Double
respDt_ufcnt	Date

Figure 4 illustrates, in the upper part, the entities whose relationship defines the possible Events of a Domain. The lower part shows this relationship in the form of connections between tables that implement the entities presented in the upper part.

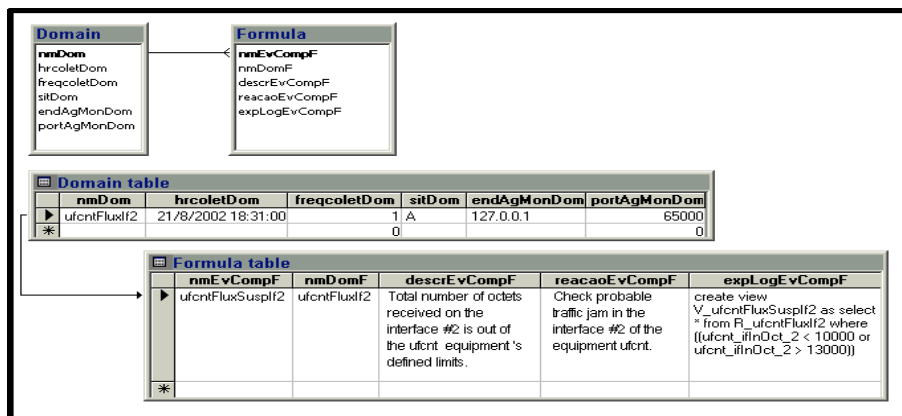


Figure 4 illustrates the definition of an Event called *ufcntFluxSuspIf2*. The Rule that characterizes it is the same one that limits the 'where' part of the SQL clause that creates the view of the database. This can be clearly seen in the contents of the *expLogEvCompF* attribute shown in Figure 4.

5. GEDSystem's Functionalities

The tool is composed by three functional groups of services that allow *GEDSystem* to reach its objectives. The first group configures the monitoring process of the network nodes and defines Events. The second one defines and dynamically generates management reports from the data stored in its repositories. The last group, formed only by the Monitoring Agent, executes the monitoring process itself.

The configuration of the monitoring process allows the Administrator to define the SNMP Objects that he/she wants to monitor. These definitions are made by the services: Communities Registration, Primitive States Registration, Indexing Registration, Domains Registration, Collection Control Registration and Events Registration.

The Communities Registration service maintains the information that allows the Monitoring Agent to request data from the SNMP Agents; the Primitive States Registrations service defines the SNMP Objects to which the network administrator's attentions are directed. The service also allows the establishment of restrictions or Rules that evaluate the possible variations of the instances of the Objects. Similarly, in the case of the registration of columnar Objects, the Indexation Registration service allows the definition of its indices; the Domains Registration service contains the SNMP Objects or Primitive States that, according to the Administrator's opinion, have some kind of logical relationship. Each group forms one Management Domain or simply a Domain, and the values of its components are stored in a repository. One repository is built for each Domain; the Collection Control Registration service fires a monitoring process for each of the defined Domains, at the same time that creates their repositories; the Events Registration service defines formulas, based on the Rules established for the Primitive States that lead to the definition and creation of Views on the repositories of the previously registered Domains. The views constitute the perception mechanisms of the Events.

The Definition and the Generation of Management Reports provide the construction of the repositories of the data obtained by the monitoring processes. This way, the definition and report generation services can generate HTML forms already filled with a list of attributes of the repositories that can be easily selected to compose the reports. The Administrator can also specify time filters for the reports. And all is done in a very friendly way. Besides the definitions of the structures of the repositories, MDGE also comprises the definitions of the Views. Thus, the transactions Definition and Generation of Management Reports can consult the Views to list the events that happened in the network in a given period of time.

The Monitoring Process is executed by one of GEDSystem's component called the Monitoring Agent. In order to have a total independence of the hardware and the operating system the Monitoring Agent is implemented in the Java language. It continuously verifies the current situation of the Domains, enabling/disabling their monitoring as requested by the Administrator. Different process threads make the surveillance of each Domain individually.

6. Future Work

As future work, the possibility of implementing the idea of Domain-based polling and the concept of Events based on database views directly in the SNMP protocol and in the RMON/RMON2 probes could be verified. The use of Triggers and Stored Procedures, in the automation of reactive procedures, could also be investigated.

In the conventional so much as in the active networks, one enhancement could be the implementation of an database where the definitions of the Domains that the Administrator wants to monitor would be contained. This way, the protocol could obtain the data of the domains locally and store them directly into the specified database. In active networks context, the active packages will constitute of data collectors of the monitored Domains. In both cases one could expect a decrease on the traffic of SNMP messages in the network, the non-use of the UDP protocol and the continuity of the support for the construction of management applications fed by relational databases.

References

1. BURGESS, John, GUILLERMO, Ray. (2000). "Raising Network Fault Management Intelligence". In *IEEE/IFIP Network Operations and Management Seminar*.
2. HARIRI, Salim, KIM, Yoonhee. (2000). "Design and Analysis of a Proactive Application Management System (PAMS)". In *IEEE/IFIP Network Operations and Management Seminar*.
3. HO L.Lawrence, CAVUTO, David.J, PAPAVASSILIOU, Symeon [et al] (2000). "Adaptive and Automated Detection of Network/Service Anomalies in Transaction-Oriented WAN's: Network Analysis, Algorithms Implementation and Deployment". In *IEEE Networks Journal*; v. 18, n. 5.
4. LI, Jung-Shian (2000). "Measurement and in-service monitoring for QoS violations and spare capacity estimations in ATM network". *Computer Communications*; v.23, p162-170.
5. LO, Chi-Chum, CHEN Shing-Hong.(1998) "Robust Event Correlation Scheme for Fault Identification in Communications Network". IEEE.
6. MELCHIORI, C., TAROUCO, L.M.R. (2000). "Troubleshooting Network Faults Using Past Experience". In *IEEE/IFIP Network Operations and Management Seminar*.
7. PAPAVASSILIOU, Symeon., SAVANT, V.S., TUPINO, J.J. [et al] (1998). Enhanced Network Management for Online Services. In *Proc. IEEE International Conference on Computer Communications and Networks IC3N'98*, Louisiana, Oct.
8. PENIDO, G., NOGUEIRA, J.M, MACHADO, C. (1999). "An automatic fault diagnosis and correction system for telecommunications management". In *Integrated Network Management VI*.
9. SHEN, Dongxu, HELLERSTEIN, Joseph.(2000). "Predictive Models for Proactive Network Management: Application to a Production Web Server". In *NOMS 2000 IEEE/IFIP Network Operations and Management Seminar*.
10. STALLINGS, William. *SNMP, SNMPv2, SNMPv3, and RMON1 and 2*. Massachusetts: Addison Wesley, c1999. 619p.
11. THOTTAN, M., JI, C. (2000). "Fault Prediction at the Network Layer using Intelligent Agents". In *Proc. of Sixth IFIP/IEEE International Symposium on Integrated Network Management IM'99, Boston, May*.
12. YEMINI, Shaula Alexander, KLIGER, Shmuel, MOZES, Eyal [et al] (1996). "High Speed and Robust Event Correlation".