Integrated Expert Management Knowledge on OSI Network Management Objects

Antonio Martín, Carlos León, and Félix Biscarri

Escuela Superior de Ingeniería Informática, Universidad de Sevilla,
Avda. Reina Mercedes, 41012 Sevilla, Spain
{toni, cleon, fbiscarri}@us.es
http://www.dte.us.es

Abstract. The management of modern telecommunications networks must satisfy ever-increasing operational demands. We propose a study for the improvement of intelligent administration techniques in telecommunications networks. This task is achieved by integrating knowledge base of expert system within the management information used to manage a network. For this purpose, an extension of OSI management framework specifications language has been added and investigated. For this goal, we shall use the language Guidelines for the Definition of Managed Objects (GDMO) and a new property named RULE which gathers important aspects of the facts and the knowledge base of the embedded expert system. Networks can be managed easily by using this proposed integration.

1 Introduction

Current communications networks support a large demand of services for which the traditional model of network management is inadequate. It is thus necessary to develop new models, which offer more possibilities. These models are called *Integrated Management Expert Systems*.

We propose a new technique which integrates the Expert System completely within the Management Information Base (MIB) [1]. The expert rules that make up the Knowledge Base are joined to the management objects definitions that belong to the network. These definitions integrate the specifications of management objects representing the network resource and the management expert rules which allow for the intelligent control and administration of the resources represented. In this document we explain the main aspects of this proposal. To achieve this we have used the OSI network management model and the Guidelines for the Definition of Managed Objects, GDMO (ISO/IEC 10165-4 (ITU X.722)) [2].

We present an extension of the standard GDMO, to accommodate the intelligent management requirements. We describe how to achieve this goal using a new extension called GDMO+. This extension presents a new element RULE, which defines the knowledge base of the management expert system.

2 GDMO and Expert Management

Information architecture is based on an object-oriented approach and the agent/ manager concepts that are of paramount importance in the open system inter-connection (OSI) systems management [3]. The denominated Managed Objects have an important role in the normalization. A managed object is the OSI abstract view of a logical or physical system resource to be managed. These special elements provide the necessary operations for the administration, monitoring and control of the telecommunications network. The managed objects are defined according to the International Standardization Organization (ISO) Guidelines for the Definition of Managed Objects (GDMO), which defines how network objects and their behavior are to be specified, including the syntax and semantics [4].

Within the OSI (Open Systems Interconnection) management framework [5], the specification language GDMO (Guidelines for the Definition of Managed Objects) has been established as a means to describe logical or physical resources from a management point of view. GDMO has been standardized by ITU (International Telecommunication Union) in ITU-T X.722 and is now widely used to specify interfaces between different components of the TMN (Telecommunication Management Network) architecture [6].

Basically, a GDMO specification defines a management information model by specifying a set of so called managed object classes that describe all different kinds of network resources. For each managed object class, one has to specify the properties that characterize the objects of this class, i.e. their attributes and their operations as well as their relationships to other objects. Such a managed object class description is meant to supply all the information that is needed to enable a managing system to effectively control the network resource represented by a managed object, and to trigger suitable management operations, if necessary. In this context, it is important that a GDMO specification clearly prescribes the capabilities of managed systems that have to be implemented as well as the meaning of all messages that can be exchanged between the managing system and the managed system. To this purpose, it is essential the knowledge base that the expert systems supply.

GDMO is organized into templates, which are standard formats used in the definition of a particular aspect of the object. A complete object definition is a combination of interrelated templates. There are nine of these templates: class of managed objects, package, attribute, group of attributes, action, notification, parameter, connection of name and behavior [7].

3 Extension of the GDMO Standard

The elements that at the moment form the GDMO standard do not make a reference to the knowledge base of an expert system. To answer these questions, it will be necessary to make changes on the template of the GDMO standard. Specifically, by means of a new item named RULE. This template groups the knowledge base supplied by an expert in a specific management dominion. It allows the storage of the management knowledge in the definition of the resources that form the system to be managed [8].

The standard we propose contains the singular template RULE and its relations to other templates. Two relationships are essential for the inclusion of knowledge in the component definition of the network: Managed Object Class and Package Template. In the standard we propose, both templates have the new property RULES. Let us study both relationships.

3.1 Template for Management of Object Classes

This template is used to define the different kinds of objects that exist in the system. The definition of a managed Object Class is made uniformly in the standard template, eliminating the confusion that may result when different persons define objects of different forms. This way we ensure that the classes and the management expert rules defined in system A can be easily interpreted in system B.

```
<class-label> MANAGED OBJECT CLASS
   [DERIVED FROM  <class-label>   [,<class-label>]*;]
   [CHARACTERIZED BY <package-label> [,<package-label]*;]
   [CONDITIONAL PACKAGES                                          (1)
           <package-label>  PRESENT IF condition;
          ,<package-label>] PRESENT IF condition]*;]
 REGISTERED AS object-identifier;
```

DERIVED FROM plays a very important role, when determining the relations of inheritance which makes it possible to reutilize specific characteristics in other classes of managed objects. In addition, a great advantage is the reusability of the object classes and therefore of the expert rules which are defined.

This also template can contain packages and conditional packages, including the clauses CHARACTERIZED BY and CONDITIONAL PACKAGES.

3.2 Package Template

This template is used to define a package that contains a combination of many characteristics of a managed object class: behaviours, attributes, groups of attributes, operations, notifications, parameters, attributes, groups of attributes, actions, behaviour and notifications. In addition to the properties indicated above, we suggest the incorporation of a new property called RULES, which contains all the specifications of the knowledge base for the expert system [9].

All the properties that we define in the package will be included later in the Managed Object Class Template, where the package is incorporated. A same package can be referenced by more than one class of managed objects.

Next definition shows the elements of a package template, in which it is possible to observe the new property RULES.

```
<package-label> PACKAGE
   [BEHAVIOUR <behaviour-label> [,<behaviour-label>]*;]
   [ATTRIBUTES <attribute-label> propertylist [,<parameter-label>]*
        [,<attribute-label> propertylist [,<parameter-label>]*]*;]
   [ATTRIBUTE GROUPS   <group-label> [<attribute-label>]*
                [<group-label> [<attribute-label>]*]* ;]
   [ACTIONS <action-label> [<parameter-label>]*
                [<action-label> [<parameter-label>]*]* ;
   [NOTIFICATIONS <notification-label> [<parameter-label>]*
         [<notification-label> [<parameter-label>]*]* ;]
   [RULES     <rule-label>   [,<rule-label>]*;]
REGISTERED AS object-identifier;                                       (2)
```

The property RULES allows a treatment similar to the other properties, including the possibility of inheritance of rules between classes. Like the rest of the other properties defined in a package, the property RULES need a corresponding associated template.

4 Expert Rule Template

This template permits the normalised definition of the specifications of the expert rule to which it is related. This template allows a particular managed object class to have properties that provide a normalised knowledge of a management dominion. The structure of the RULE template is shown here:

```
<rule-label> RULE
   [PRIORITY        <priority> ;]
   [BEHAVIOUR    <behaviour-label> [,<behaviour-label>]*;]
   [IF           occurred-event-pattern [,occurred-event-pattern]*]
   [THEN      sentence [, sentence]* ;]
REGISTERED AS object-identifier;                                       (3)
```

The first element in a template definition is headed. It consists of two sections:
- <rule-label>: This is the name of the management expert rule. Rule definitions must have a unique characterizing name.
- RULE: A key word indicates the type of template, in our case a definition template and the specifications for the management expert rule.

After the head, the following elements compose a normalised definition of a expert rule.
- BEHAVIOUR: This construct is used to extend the semantics of previously defined templates. It describes the behaviour of the rule. This element is common to the others templates of the GDMO standard.
- PRIORITY: This represents the priority of the rule, that is, the order in which competing rules will be executed.
- IF: It contains all the events that must be true to activate a rule. Those events must be defined in the Notification template. The occurrence of these events is necessary

for the activation of the rule and the execution of their associated actions. We can add a logical condition that will be applied on the events occurred or their parameters.
- THEN: This gives details of the operations performed when the rule is executed. Those operations must be previously defined in the Action template. These are actions and diagnoses that the management platform makes as an answer to network events occurred.
- REGISTERED AS is an object-identifier: A clause identifies the location of the expert rule on the ISO Registration Tree. The identifier is compulsory.

5 Application of the GDMO+ Standard System Network Management

The traditional expert management uses management knowledge and management information separately. Integrating both elements is the main purpose of our work.

This section present a tool based on the proposed GDMO+ standard, which helps administrators in expert network management. Our tool understands transceivers and multiplex equipment. We will describe the basic structure and concepts of our software, especially the knowledge base.

ISO classifies the systems management activities into five functional areas: fault management, accounting management, configuration management, performance management and, security management [10]. We can categorize the expert systems within these five groups. The expert system that we have built would be included in the area of work: fault management. Related work is briefly discussed in the next section.

5.1 Related Work

We present a rule-based expert system applied to error diagnosis in the communications system of SEVILLANA-ENDESA (a major Spanish power utility). Part of

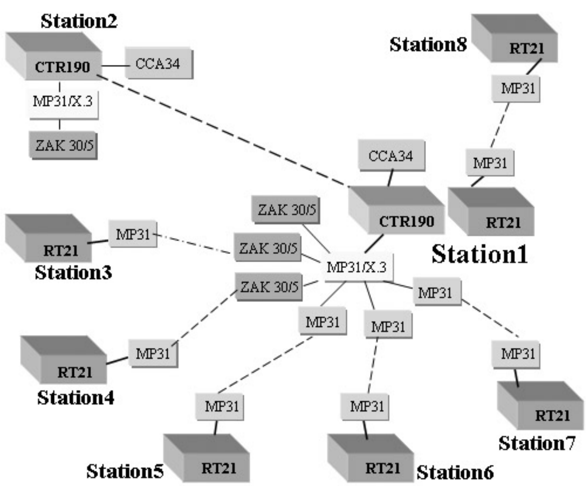

Fig. 1. Power Company Network

SEVILLANA-ENDESA's long-distance traffic is controlled by a wireless System distributed throughout the Endesa network. Expert systems are part of the system dedicated to the management of a power utility's communications system, which we call NOMOS [11]. NOMOS is implemented in Brightware's ART*Enterprise. ART*Enterprise is a set of programming paradigms and tools that are focused on the development of efficient, flexible, and commercially deployable knowledge-based systems. NOMOS+ is an extension for intelligent decision-making and diagnostic reasoning controlled by its own integrated expert system. NOMOS+ is the first production software written and integrated in GDMO+, Fig.1.

The knowledge base is included within the specifications of the managed resources, following the proposed prescriptions in standard GDMO+. These new specifications contain management information of managed resources and include also the set of expert rules that provides the knowledge base of the expert system.

5.2 The System Features

NOMOS+ operation, uses a supervision system called SSC (Communication Supervisory System). This system can monitor, in real time, the network's main parameters, making use of the information supplied by a Supervisory Control and Data Acquisition (SCADA), formed by a Control Center (placed on the main CSE building), and Remote Terminal Units (RTUs) installed into different stations. The use of a SCADA system is due to the management limitations of network communication equipment, Fig.2.

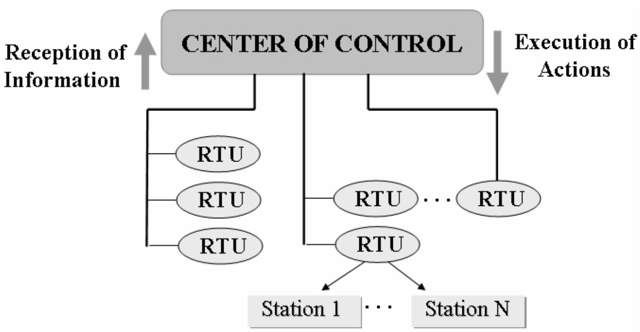

Fig. 2. NOMOS+ System Features

The SSC allows the operator to acquire information, alarms or digital and analogical parameters of measure, registered on each RTU. Starting from the supplied information, the operator is able to undertake actions through the SSC in order to solve the failures that could appear or to send a technician to repair the stations equipment.

5.3 The System Architecture

Our tool has three major components: a knowledge base, an inference engine and a user interface [12], Fig.3.

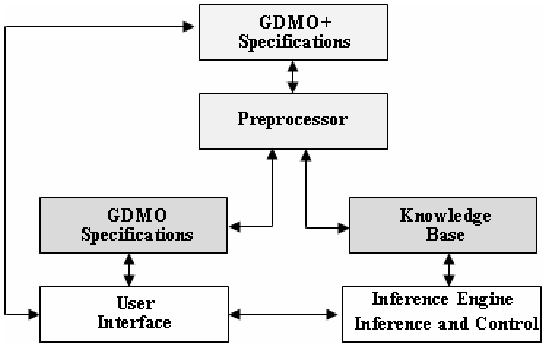

Fig. 3. Elements of the prototype NOMOS+

This structure is typical of expert systems. Those elements are briefly discussed in the following:

The knowledge base: The core of the system, this is a collection of facts and if-then production rules that represent stored knowledge about the problem domain. The knowledge base of our system is a collection of expert rules and facts expressed in the ARTScript programming language ART*Enterprise.

The knowledge base contains both static and dynamic information and knowledge about diferent network resources and common failures. The knowledge base of our system can be extended by adding new higher level rules and facts. To this purpose we can employ user interface.

The inference engine: This is the processing unit that solves any given problems by making logical inferences on the given facts and rules stored in the knowledge base. In our tool we used the ART*Enterprise. By using an existing general purpose tool we were able to build a standard and extensible platform with proven performance and quality.

The user interface: This controls the inference engine and manages system input and output. The user interface of our tool contains a preprocessor for parsing GDMO+ specification files, a set of input and output handling routines, and a simple command prompt interface for managing the system. Also, the user interface components allow administrators to inspect the definitions of management object classes interactively, this allows to modify or include new experts management rules in the managed objects definition.

The prototype has a preprocessor module. A previous phase to the inference is realized by a unit processor-translator, which processes the file that contains the GDMO+ specifications and extracts the normalized knowledge from the expert system. Two exits are obtained: a file with management expert rules and another file with GDMO definitions of the managed object classes. The preprocessor also translates the expert rules into a valid syntax for the programming language of inference engine. Procedures are coded in ART*Enterprise's ARTScript language, a dynamic interpreted language similar in syntax to LISP.

6 Example of a Management Expert Rule

This section shows a complete example of an expert rule integration in the GDMO+ proposed standard; it defines the managed object of a CTR 190-type radio transceiver. This device belongs to a private network that gives service to a power utility.

In the next example a Class of Managed Object is radioTransceiver, which defines the properties corresponding to the radio transceiver. This class includes the compulsory transceiverPackage which contains all the specifications corresponding to the device. The most important properties that we can indicate are the two expert rules that have been associated with the defined class by means of the RULES clause. The two rules are defined by using the RULE template.

```
radioTrasnceiver MANAGED OBJECT CLASS
  DERIVED FROM "rec.X721": top;
  CHARACTERIZED BY transceiverPackage;
REGISTERED AS {nm-MobjectClass 1};

transceiverPackage PACKAGE
 ATTRIBUTES
    receptionPower     GET,
    statusTransmission GET, speedTransmission  GET, ...
 NOTIFICATIONS
    damageFeeding, inferiorLimit, repairAction;
 RULE powerError, transmissionError;
REGISTERED AS {nm-package 1};

powerError RULE
 PRIORITY 3;
 BEHAVIOUR powerErrorBehaviour;
 IF (?date ? ?local 7_F_ALIM_2 ?remote ALARM)
   (NOT (?date ? ?local CCA?34_AIS_DE_BB ?remote ALARM))
 THEN ("Severity:" PRIORITY),
 ("Diagnostic: It damages in the electric feeding of the
           station" ?local),
 ("Recommendation:To  revise  the  electric  connection",
?local);
REGISTERED AS {nm-rule 1};

transmissionError RULE
  PRIORITY 4;
  BEHAVIOUR transmissionErrorBehaviour;
  IF (?date ?time1 ?local 7_TX_C2 ?remote ALARM)
     (?date ?time2 ?local 7_TX_C2 ?remote ALARM
            & : (<(ABS(? ?time1 ?time2)) 1.00))
 THEN ("Severity:" PRIORITY),
  ("Diagnostic:
   "It damages in the modulate transmission", ?local),
  ("Recommendation "Revision transceiver");
REGISTERED AS {nm-rule 2);
```

Both rules detect anomalies or defects of operations produced in the transceiver and suggest the necessary measures for solving the problem. The first rule powerError, is in charge of detecting failures in the power supply of the device; the second rule transmissionError, is devoted to the detection of errors in the data transmission module. Both rules give recommendations on how to solve the failures.

6.1 Final Prototype Verification

The purpose is to achieve a functionally correct prototype. To verify the system we feed it with an alarms arbitrary amount. The results of this proof are included in Table 1. From these results we can establish the following conclusions:

- Filtration process effectiveness is very high: almost 90% of the whole. This has the advantage of a decreasing percentage in the amount of indications presented to the operator.
- The speed of the system improves diminishing the number of alarms on which the rest of rules act.

Table1. Prototype Testing Results

Alarms Initial Number	Number After Filtration	Filtered Alarms	Fired Rules	Preceding time	Rules/Sec.	Indications to the Operator
100	1	99	51	0,118 Sec.	432,2034	1
200	10	95	102	0,412 Sec.	247,5728	6
300	31	89,6	155	1,250 Sec.	124,0000	20
400	31	92,25	201	1,438 Sec.	139,7775	16
500	32	93,6	254	2,975 Sec.	85,3782	19
600	38	93,66	293	5,249 Sec.	55,8202	16
700	44	93,71	346	17,982 Sec.	19,2415	18
800	55	93,125	394	26,938 Sec.	14,6262	23

The expert system, with over 150 operation rules, has produced excellent results which, after extensive field-testing, proved to be capable of filtering 93% of produced alarms with a precision of 95% in locating them.

7 Conclusions

Current networks are very complex and demand ever-increasing levels of quality, making their management a very important aspect to take into account. The traditional model of network administration has certain deficiencies that we have tried to overcome by using a model of intelligent integrated management. To improve the techniques of expert management in a communications network, we propose the possibility of integrating and normalising the expert rules of management within the actual definition of the managed objects. Through the integration of the knowledge within the new extension of the GDMO standard, we can simultaneously define the management information and knowledge.

Thus, the management platform is more easily integrated and allows a better adaptation for the network management. We conclude pointing out an important aspect of the obtained integration: by using only and exclusively the extended GDMO specification, the administration platform will be able to obtain the management necessary information with respect to the managed objects as well as the expert rules of management that make up the knowledge base of the expert system.

References

1. Morris, Stephen B.: Network Management, MIBs and MPLS: Principles, Design and Implementation By Publisher: Addison Wesley (2003).
2. CCITT X. 722 / ISO 10165-4 ISO, Structure of management information. Part 1: Guidelines for the definition of managed objects.
3. ISO/IEC and ITU-T, "Open Systems Interconnection - Systems Management Overview," Standard 10040-2, Recommendation X.701. 1992.
4. Hebrawi, B.: GDMO, Object modeling and definition for network management. Technology appraisals (1995)
5. ITU-T Recommendation X.700, Management Framework for Open Systems Interconnection (OSI). CCITT Applications (1992).
6. ITU-T Rec. M.3010, Principles for a Telecommunications Management Network (TMN). Study Group IV (1996).
7. Black, U.D.: Network Management Standards. McGraw Hill (1995).
8. Stallings, William.: SNMP, SNMPv2, and CMIP: the practical guide to network. Publication Reading, Mass. [etc.] Addison-Wesley, (2000).
9. Garcia, R.C.; Cannady, J.: Boundary expansion of expert systems: incorporating evolutionary computation with intrusion detection solutions. SoutheastCon 2001. Proceedings. IEEE, (2001) 96-99.
10. ITU-T Recommendation M.3400, TMN Management Functions, 1997
11. Leon, Carlos. Mejias, Manuel. Luque, Joaquin. Gonzalo, Fernando.: Expert System for the Integrated Management of a Power Utility's Communication System. IEEE Trans on Power Delivery, Vol. 14, No. 4, Octubre, (1999), pp 1208-1212
12. Joseph C. Giarratano, Gary D. Riley.: Expert Systems: Principles and Programming. Book, Brooks/Cole Publishing Co. (2005).