INTEGRATION OF EXPERT SYSTEM RULES INTO STANDARDIZED OBJECT DESCRIPTION MODELS FOR TELECOMMUNICATION NETWORK MANAGEMENT

S. Martín, C. León, M. Mejías, M. C. Romero, A. V. Medina
Facultad de Informática y Estadística - Universidad de Sevilla
Avda. Reina Mercedes s/n - 41012 - SEVILLA - Spain
e-mail: smartin@cica.es

Abstract: The work described in this paper is focused on the attainment of a product that: a) is able to manage a communication network according to ISO ant ITU-T standards (Telecommunication Management Network, TMN, model); b) automatically locates network faults based on a set of rules embedded in an integrated expert system; and c) represents the expert knowledge using procedures and tools to define object and object classes, following the internationally standardized model. The most outstanding characteristic of the product is the availability of expert functionalities integrated into the very object definition of the management system.

Key words: expert systems, TMN, network management, object-oriented technology.

1. INTEGRATED MANAGEMENT PLATFORMS

At this time, telecommunication networks are present everywhere. These networks are not only rather complex, but they are also made up of heterogeneous components; in other words, they incorporate incompatible hardware architectures, interfaces, protocols... from different manufacturers.

The complex task of network management can only be fulfilled if it is computer assisted. The wealth of different management tasks has already led to a wealth of tools. Due to the network heterogeneity, the same management function is supported by various tools, which may differ considerably in their use (interfaces, languages) from manufacturer to manufacturer.

The network manager faces not only with a variety of control consoles and redundant network descriptions, but also with the lack of a global view of the network. It follows from the problems discussed above that the functionality of isolated management tools (isolated in terms of manufacturers and functionality) does not meet the needs of the network manager.

As discussed above, traditional management tools only support specific application areas for products from specific manufactures and they have not open interfaces based on recognized standards that may be used to extend the functional scope or application area of the tool.

The objective must be an integrated management [6]. The requirements on integrated management include:

- Integration of architectures and system and network types.
- Integration of management functional areas.
- Integration of organizational aspects (domain concept)
- Common concept for a management data repository.
- Extensively standardized concepts for network and system managements.
- Support for distributed applications and systems.
- Common programming interfaces and user interfaces.

A necessary condition for an integrated network management is that the components to be managed in a heterogeneous environment should provide information that may be interpreted in a manufacturer-independent manner, and this information should be accessible via well-defined interfaces and protocols. In other words, manufacturer-independent integrated management is only possible on the basis of recognized standards [2]. In recent years, in connection with progress in the standardization of management protocols an management information, management platforms have gained increasing importance as carriers systems for integrated management systems.

Management platforms have the following properties:

- They run in open system environments, in other words, on systems with open interfaces, such as the UNIX operating system.
- They are not restricted to the management of resources from specific vendors.
- They are not restricted to functional areas.
- They are based on object-oriented data models.
- They provide well-defined interfaces based on international standards and allow the integration of new modules (applications in a specific functional area or modules for managing the components of specific vendors).
- Application may access resources via these interfaces, without needing to know which vendor implemented the resources or how the resources are reached over the network.

In particular, the project described in this paper makes use of HP OpenView, that is one of the most widespread integrated management platforms.

One of the project phases consists of customizing this platform to manage in an integrated way a test network: the network of the Facultad de Informática y Estadística at the University of Seville. This way, a basic management of the network is made and some experience in integration tasks is achieved.

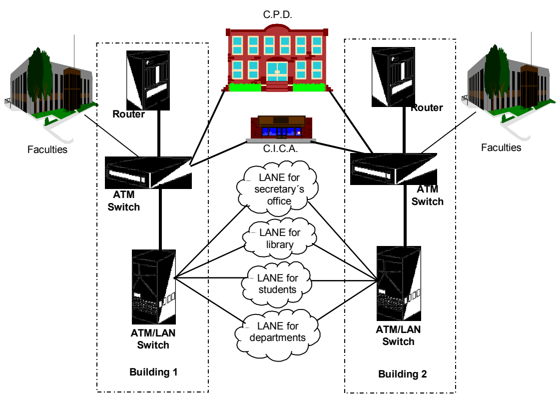

Figure1: Test network

Although integrated management platforms let us integrate several elements, they often do so at the expense of increased development resources.

Expert alarm management is one of the most important functionalities to integrate into the management platform.

2. DESCRIPTION OF THE TEST NETWORK

The test network belongs to the RIUS network (Red Informática de la Universidad de Sevilla), that is integrated into Internet via the RICA network (Red Informática Científica de Andalucía). The RIUS network is based on ATM technology (Asynchronous Transfer Mode) and fibre-optic links.

The test network is allocated in two buildings (building 1 and building 2), which are well equipped to interconnect different devices such as ATM switches, ATM/LAN switches (Local Area Network), LAN switches, routers, hubs, repeaters, etc. as figure 1 shows.

Four LANEs (LAN Emulated) are defined from the ATM switches, describing the logical topology of the test network: LANE for secretary's office, LANE for library, LANE for deparments and LANE for students.

3. EXPERT ALARM MANAGEMENT

3.1. Alarm Management in Communication Networks

Alarm management is responsible for receiving and processing alarms. These may be external alarms generated by resources and sent to the management platform, or internal alarms generated by other platform components.

Under normal network conditions, the operator controls periodically the list of active alarms looking for the presence of possible failures.

The occurrence of faults at one point in the network generally leads to a large flood of alarm messages being reported to the user, since faults propagate horizontally and vertically in the network and, correspondingly, alarms are generated at many points in the network.

In the case of a major failure in relevant network components, the large number of alarms reported to the management station represents a difficult problem for the network operators, who are usually overloaded and are not able to resolve these specific faulty situations.

The operator must hold a high level of qualification and experience in order to be able to identify the failures that can be origin of each alarm or group of alarms.

Thus, alarm filtering and correlation within the management platform are important prerequisites for effective alarm management.

3.2. Rule-Based Expert System

An expert system is a sophisticated computer program that applies human knowledge in a specific area of expertise to arrive at a solution to a problem from that area [1], [3], [4]. This solution is essentially the same as that concluded by a person knowledgeable about the domain of the problem when confronted with the same problem.

All expert systems have a knowledge base, which is the repository of the rules, facts and knowledge acquired from the human expert. Most expert systems are often referred to as rule-based systems. The knowledge is typically represented in the form of IF ... THEN type rules (premises and conclusions), facts and assumptions about the problem the system is designed to solve.

The other three components present in all experts systems are the inference engine, the justifier and the user interface. An expert system can be built using any conventional programming language on any computer but, nevertheless, most of them have been developed using shells, rather than languages. A shell is a special expert system development program that makes it possible to create an expert system without programming the inference engine, justifier and user interface. It is a reasoning system without the knowledge.

An expert system made with a shell is a very flexible system that may accommodate easily new knowledge in its knowledge base. The main problem with this approach is the difficult integration of the expert system into the management platform.

3.3. Rule Based Expert System for Alarm Management

In order to achieve an effective alarm management, the expert system, that is been developed by the authors by using the ART-Enterprise shell, acts collecting the management platform information and applying rules which contain the operators' knowledge. It is very difficult to extract the rules because of the lack of enough expert operators. These rules (about fifty) carry out the following tasks:

- Redundant alarms are filtered to avoid the saturation of the capacity of the operator. This filtering is not accomplished indiscriminately. Those alarms that show a service degradation or that can mask certain failures are not eliminated.
- System failures are detected using the information collected by the alarms.
- A severity index is created to give priority to the alarms.
- Some recommendations are shown to the network operators in order to solve failures or to improve the system operation.

4. INTEGRATION OF EXPERT KNOWLEDGE IN THE MANAGEMENT PLATFORM.

To be useful, a management expert system must be able to adapt for the network it works on, whose topology and components change continuously. It is easy to achieve this goal using an expert system shell, because a shell is a very powerful programming environment that lets us introduce any kind of rule needed. However this approach needs a very qualified programmer and exhibits integration difficulties.

Another approach may be to let the network operator modify the rules by easy dialog boxes, at the expense of reduced flexibility.

A third approach, which is the aim of this work, is the development of a methodology that lets us include the expert rules in the object structure integrated into the management platform. Thus, network operators can introduce the expert knowledge using the same tools as that the management platform provides to create and modify the managed objects.

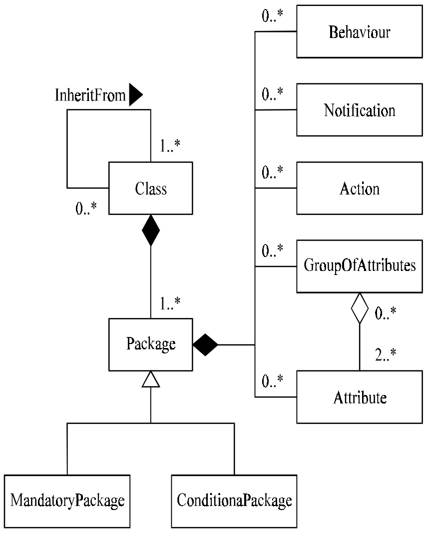

Figure 2: Information structure of a GDMO managed object class, described using UML

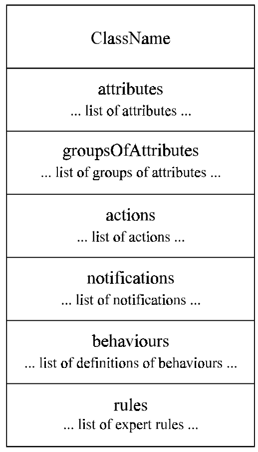

Figure 3: UML representation of a GDMO managed object class

The MIB (Management Information Base) would include the expert knowledge, so it is necessary to redesign its structure.

Our aim is to extend the GDMO standard (Guidelines for the Definition of Managed Objects) of ISO in order to achieve this goal in a standardized way [5]. We will use the UML notation (Unified Modelling Language), graphic language widely extended in the object-oriented community [7]. This notation will help us to express the way we can include expert-knowledge in the object definition. As GDMO states, managed objects with the same features and similar behaviour, are regarded as belonging to the same managed object class.

As figure 2 shows in UML notation, GDMO object classes are organized in an inheritance hierarchy. A managed object class inherits properties from one or more parent classes (superclasses) and refines them. Every object class at least inherits from Top class. And, similarly, an object class may be related to several subclasses that inherit behaviour and attributes from it and refine them.

Moreover, the definition of a managed object class exhibits a package-based structure. We can differentiate between mandatory packages and conditional packages. Every object has to include one mandatory package at least, whereas conditional packages are optional.

An object class instance incorporates all the properties and functions that form the mandatory packages. When a managed object is instantiated, the decision is taken as to whether a conditional package should take effect, and hence become an integral part of the managed object.

A package includes information about the following elements: attributes, groups of attributes, actions that can be performed on the managed object as a whole, notifications that can be emitted by the managed object and behaviours exhibited by the managed objects.

Figure 3 shows the UML representation of a GDMO managed object class. It can be seen that the package concept is not explicitly present in this representation. However, UML supports enough methods (constraints, stereotypes, etc.) so that the semantics related to the concepts of mandatory and conditional packages can be expressed properly.

Once we have shown how to define, by using UML, the managed object classes recommended in the GDMO standard, let us now consider how to integrate the expert knowledge (expert rules) into the very object class definition.

First of all, it is important to analyse the different rule types we can find in the managed objects, related to the scope of application of such rules.

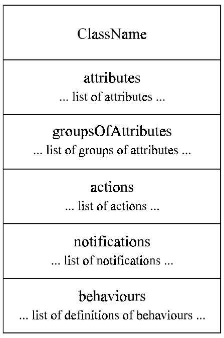

Figure 4: UML representation of a GDMO managed object class that includes expert knowledge

From this point of view, we can distinguish the following types of rules:

- Object scope rules: Those rules that apply to one object in an individualized way.
- Class scope rules: Those rules that apply to a group of objects belonging to the same class.
- Wide scope rules: Those rules that apply to a group of objects belonging to different classes.

By extending the previous definition of the managed object class, we will manage to integrate all kind of rules. So, let us consider a new list zone, called rules. It will include, in the adequate syntax, the expert system rules that apply to the objects. In order to distinguish among the three rule types, second type rules will use the same syntax but underlined, as UML does. Third type rules will be characterized by placing a dollar sign ($) in front of the rule definition. This approach avoids the definition of different classes of rules, because our aim is to include the rules into the standardized object classes of the existing MIBs.

Figure 4 shows the definition of the managed object class including the rule definition, whereas figure 5 shows the consequential structure after including the expert rule definition in the information structure of the managed object class proposed by GDMO.

5. CONCLUSIONS

Today we are accomplishing the evaluation process of the system.The resulting product is an integrated management platform with basic management capabilities, that provides an infrastructure in which different management applications may be embedded. It is particularly

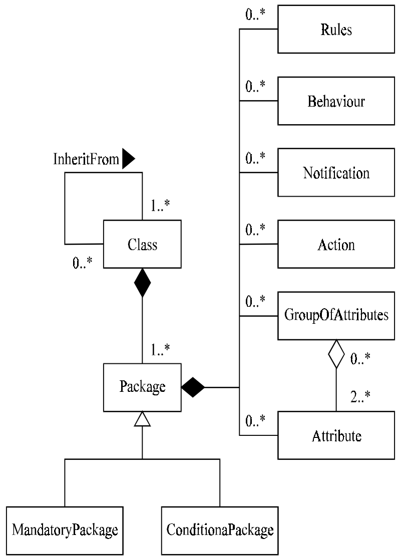

Figure 5:Information structure of a GDMO managed object class that includes expert knowledge, described using UML

interesting the fact that the network manager could easily include expert system functionalities in the platform by using the same standardized environment employed to define the network managed objects.

6. ACKNOWLEDGEMENTS

The work described in this paper has been co-funded by the Comisión Interministerial de Ciencia y Tegnología (CICYT) and the European Commission within the framework of the Fondo Europeo de Desarrollo Regional (FEDER) through the project reference number 1FD97-0595.

7. REFERENCES

[1] Awad E. M., 1996. Building expert systems. West publishing company.
[2] Black, U.D., 1995. Network Management Standards. 2nd ed. McGraw Hill.
[3] Durkin, J, 1996. Expert Systems: a view of the field. IEEE Expert, April 1996, 56-63.
[4] Gonzalez, A.J., Dankel D.D., 1993. The engineering of knowledge-based systems. Prentice-Hall.
[5] Hebrawi, B., 1995. GDMO, object modelling and definition for network management. Technology appraisals
[6] Hegering, H.G. and Abeck, S., 1994. Integrated network and system management. Addison-Wesley.
[7] Rumbaugh, J., Booch, G. and Jacobson, I., 1999. The Unified Modelling Language reference manual. Addison-Wesley.