A framework for development of integrated intelligent knowledge for management of telecommunication networks

Antonio Martín *, Carlos León, Joaquín Luque, Iñigo Monedero

Department of Electronics Technology, University of Seville, Escuela Superior de Ingeniería Informática, Avda. Reina Mercedes S/N, Spain

Keywords:

Intelligent agents
Expert system
GDMO
MIB
TMN
Artificial intelligent
Network management

ABSTRACT

The management of modern telecommunication networks is a complex and intensive task that requires the assimilation of vast amounts of information and knowledge management. The significance of management networks is growing, and more advanced techniques are needed to support these activities. It is necessary to develop new models that offer more possibilities. In this study, we present a new approach for distributed intelligent management networks. The goal of our study is the assignment and dispersed control of proper network resources, pertaining to hardware as well as software, to help operators manage their networks more effectively and also to promote reliability in network services. We propose a new paradigm where intelligent knowledge management is integrated into the conceptual repository of management information. This article presents a technique for the design and implementation of a distributed intelligent system that is designed through the normalization of knowledge management. Our study focuses on an intelligent framework and a language for formalizing knowledge management descriptions and combining them with an existing Open Systems Interconnection (OSI) management model. Further, this work outlines the development of an intelligent system named ScanEXP based on our proposed standard and describes the most important facets, advantages, and drawbacks that were found after prototyping our proposal.

1. Introduction

Management network has been growing in complexity due to the increased network versatility and numerous services. On the other hand, the expansion of networks has provided a new orientation to the classical way of using services. Traditional network management models are inadequate to support this large demand for services. In the traditional manager-agent network architecture, the agent is kept as simple as possible, only tasked with device status report and update, whereas the burden of management and data processing resides with the manager. The standard manager-agent architecture based on thin objects should be reviewed.

Distributing intelligence to management objects is an inevitable trend in network management and one that is critical for the success of future network management designs. We make use of this approach in our work. Therefore, it is definitely necessary to review the way network management architecture is designed, with the idea that we will have to use distributed intelligent entities to perform various tasks. The intelligent agent approach is adopted in this study and is viewed as an advanced solution by management networks. The aim of this work is to model an intelligent management network method that is capable of solving problems in an autonomous and distributed form, while utilizing concepts from software engineering, artificial intelligence, and multi-agent systems. Concepts from all these systems across distributed systems, aligned with a focus on the integration of linked systems, information, and services, can improve the development of new management network systems.

Nowadays, distributing intelligent management networks is viewed as one of the fastest growing areas of research and new application development in network management. There are other researchers for these topics, but none in terms of normalization of the knowledge of the intelligent agents. Typical work in related fields include intelligent agents such as (Yang & Chang, 2011), who present a system to collect information through the cooperation of intelligent agent software, in addition to providing warnings after analysis to monitor and predict some possible error indications among controlled objects in the network. This technique is derived from the ontology that accurately stores the operating information of network management in the backend database. This article (Vallejo, Albusac, Castro-Schez, Glez-Morcillo, & Jimenez, 2011) discusses the architecture devised to deploy intelligent surveillance systems by means of a set of autonomous agents who are responsible for the management of different

* Corresponding author. Address: Escuela Superior de Ingeniería Informática, Avda. Reina Mercedes S/N, Spain.

E-mail address: toni@us.es (A. Martín).

surveillance tasks and for cooperating to monitor complex environments. This multi-agent architecture is inspired by a normality-based formal model that is used to define the knowledge needed to analyze general-purpose surveillance concepts. Other examples of application of intelligent agents include (Chantaraskul & Cuthbert, 2008), who propose the application of intelligent agents to a service-level-based control in resource management, essentially for congestion management, and demonstrate the ability of intelligent agents to improve and maintain the quality of service (QoS) to meet the required service level. Current approaches to network management employ the client/server model, cooperative stationary agents, and/or non-intelligent mobile agents. A particularly novel aspect of this work is the use of learning (case-based reasoning—CBR) to predict the control strategies to be imposed. (Ray, Parameswaran, & Lewis, 2010) describe a novel approach for managing service-centric communications networks called *distributed autonomic management*. The model consists of communities of mobile and stationary intelligent agents in collaboration.

In addition, the advancing development of integrated intelligent management systems has motivated researchers to begin addressing the specific problem of integrating knowledge management. Many researchers have suggested that intelligent sensor network technologies could improve the effectiveness and efficiency of real-time management. The objective of Shin, Chin, Yoon, and Kwon (2011) is to develop a seamlessly integrated information management framework that can provide logistics information for decision making (Oztemel & Tekez, 2009). In this article, such a reference model for an intelligent integrated manufacturing system is introduced. This research identifies a need for an integrated manufacturing system that will handle all interactions and interrelationships which will then be affected by the changes and create maximum gain under limited resources. The system has a hierarchical architecture with several agents responsible for different manufacturing functions. To allow interaction among the agents for sharing their knowledge, a specific knowledge exchange protocol in a knowledge network is being developed. To support Just-In-Time delivery, it is necessary to build a framework that can facilitate the collection and sharing of information on construction components and material flow throughout the whole supply chain process. Tacconi, Miorandi, Carreras, Chiti, and Fantacci (2010)) propose an architecture that enables mobile nodes to query a largely deployed wireless sensor network in an intelligent transportation system scenario. This architecture covers situations in which sensors deployed in a parking place or along a road provide information to cars on the conditions of the surrounding environment.

In intelligent decision support systems, (Lei, He, & Zi, 2009) propose a new method for intelligent fault diagnosis of rotating machinery, and an experiment of rolling element bearings is carried out to test the performance of the proposed method. Song, He, Chu, and Gu (2008) describe a research project that aims at developing a Web-based intelligent diagnosis system for the hydro turbine and discusses the process of the fault diagnosis and the difficulties involved in developing the system. The paper also includes several practical issues related to the architectures of intelligent Web-based applications. This article illustrates that an integrated Web-based expert system of fault diagnosis has been developed for the hydro turbine of a power station. Moreover, it can be regarded as an advisory tool by those who have much technical experience and as a training tool for the less-experienced personnel who seek guidance and advice. The current study (Doukas, Patlitzianas, Iatropoulos, & Psarras, 2007) presents an intelligent decision support model that uses rule sets based on a typical building energy management system. In addition, the model's impact on the energy consumption and indoor quality of a typical office building in Greece is presented. The model can control how the building's operational data deviate from the settings, carry out a diagnosis of internal conditions, and also optimize the building's energy operation. In this context, the integrated decision support model can contribute to the management of the daily energy operations of a typical building that are related to the energy consumption.

In this study, we present a new paradigm where resources specifications and knowledge managed are integrated in a same language definition. In this article, we describe an intelligent agent who processes information collected by the agents and uses it to detect the network anomalies that typically precede a fault. Fault identification involves testing the hypothetical faulty components. Repair is achieved by performing intelligent corrective actions. The agents collect information about the network node through their management information base (MIB), which holds the management knowledge and a set of variables pertinent to that particular node (Stallings, 2000). Intelligent agents learn the normal behavior of each measurement variable and combine the intelligent knowledge for the management of the network resources. The objective is to improve insight and understanding of network management, and present an alternative distributed management network model. This article proposes a new evolution called *integrated intelligent management*.

We present an analysis of corporate network management requirements and technologies, together with our implementation experience with the development of an integrated management system for a company network. We used the Guidelines for the Definition of Managed Object (GDMO) (Hebrawi, 1995) and also propose a new extension called Extended GDMO, or simply GDMO+, for the incorporation of the distributed knowledge management.

In the next sections, we review the experimentations using intelligent agents to solve the problems encountered by the classical network management approaches. First, we analyze the current management models, their evolution, and applications. We offer a general vision of traditional management, analyzing its deficiencies and discovering the needs that push us toward new distributed intelligent management paradigms. Then, we examine the design and development of the intelligent platform named *ScanEXP*. From there, we present the concept of the formulation of the system design proposal and a detailed description of the various modules. The next section summarizes the performance of the platform and the results of the research. Finally, we outline the conclusion and future works.

2. Information modeling

Basic network management models are a starting point for understanding our work. There are two main standards for controlling telecommunications equipment: Telecommunications Management Network (TMN) and Simple Network Management Protocol (SNMP) (Liao & Seret, 1991). The SNMP is the standard operations and maintenance protocol for the Internet. The SNMP is used to manage networked devices and is one of the key technologies that enabled the Internet's phenomenal growth. Developed by the Internet Engineering Task Force (IETF), SNMP standards are defined in a series of documents, called *request for comments* (RFCs), proposed by the Internet Engineering Task Force (IETF). In a private network environment, SNMP enjoys near-universal support; however, in a public environment, a more heterogeneous mix of de facto telecommunications industry standards has prevailed, with a move toward TMN support (ITU-T, 1996a; ITU-T, 1996b). The TMN was the first that started, as a part of its Open Systems Interconnection (OSI) program, the development of the architecture for network management. The TMN is introduced by the ITU-T (the former CCITT) (Zuidweg, 2002).

The SNMP and OSI network management standards define two different languages and information models for describing management information. These specific languages are developed within projects for specific purposes: GDMO for OSI and SNMP for the Internet (Doukas, Patlitzianas, Iatropoulos, & Psarras, 2007). The effective transfer of the management information between agent and manager processes is performed by using the Common Management Information Protocol (CMIP) and SNMP protocols. The main concept in network management is the managed object, which is an abstract view of a logical or physical resource to be managed in the network. Managed objects provide the necessary operations for the administration, thereby monitoring and controlling the telecommunications network. Each manager or agent maintains a local database of information that is relevant to network management, known as the *management information base* (MIB) (Morris, 2003). An MIB is the specification containing definitions of management information required by a management system. The MIB, which is specified by using the Structure Management Information (SMI), defines the actual objects to be managed (ISO/IEC & ITU-T, 1998). The relationship between the manager, the agent, and the MIB is shown in Fig. 1.

The MIB is generally described as a model of the total information that is reflected or results from the operation of the implementation on a system, or controls that implementation. The main focus of our article is on the information models themselves.

3. Intelligent agent structure

A managed object can be viewed as a mediator between the network management interface and the hardware. On the other hand, an intelligent agent is a managed object situated within and a part of an environment which senses that environment and acts on it, over time, in pursuit of its own knowledge agenda and so as to execute autonomous management actions. Our essential idea behind network management is the possibility for the intelligent agent to exist in different heterogeneous networks. In fact, we may consider an intelligent agent as a category of managed objects. Then, intelligent agents may be developed as if they would be always on the same resource network called *managed object*. Management information modeling plays a large part in this network management model. The languages that are the easiest to create an agent are specification languages which define the Managed Object Class (MOC). We suggest a new standard for the information management definition. Our approach is to explore how the same management information would be modeled within the OSI standard.

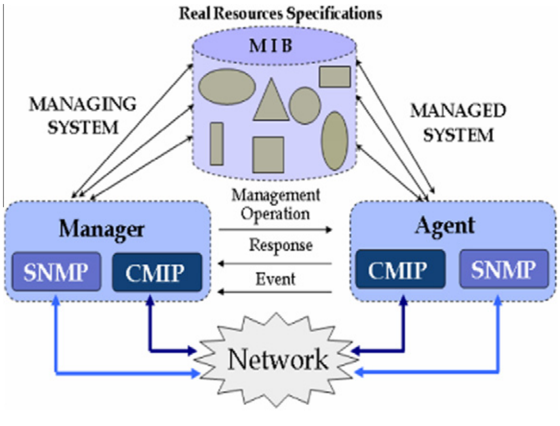


Fig. 1. Architecture of management models.

To formalize the main proposal of this article, we analyze necessary areas of requirements to undertake aspects related with the knowledge integration in objects and getting intelligent agents. These objects are defined according to the rules set out in the SMI. Practical experience with SMI shows that, from an intelligent point of view, the quality of current specifications is not satisfactory. Intelligent object specifications are incomplete in defining the management knowledge of a specific resource. It will be necessary to make changes in the template of the SMI standard (ITU-T., 1992). As consequence, a new element is necessary. To solve the current problem of undertaking an intelligent integrated management, we offer an original contribution that includes intelligent knowledge management in the specifications of the network features. We propose to extend the SMI language in the OSI network management model to accommodate the intelligent management requirements. The intelligent agent will maintain management information and knowledge related to its managed elements in the MIB+. We present an extension named *GDMO+* with the goal of facilitating the normalization and integration of the intelligent knowledge base into resources specifications. Fig. 2 shows the anatomy of our intelligent agent.

Intelligent agents can actively and dynamically cooperate for solving problems by using task- and domain-level protocols. An important goal is convergence on solutions despite incomplete or inconsistent knowledge or data. It stands on three essential properties:

- Autonomy or self-government independence: The intelligent agent decides when and under which condition he/she will perform what actions. The autonomy is explicitly required not only as property, but also as reactive, proactive, and cooperative behavior.
- Communication, which is the ability to speak with a peer, with humans (through an interface agent), or with devices.
- Cooperation agents are collaborative when they are able to work together.

There are experiments that replace simple SNMP or CMIP managed objects by intelligent agents such as Doukas et al. (2007), Waiman, Leung, and Tam (2005), Power and Bahri (2005), Akinyokun and Imianvan (2006), Barco, Díez, Wille, and Lázaro (2009), and Li (2007); however, without a global review of management principles, this cannot resolve the current issues. In our work, we propose a set of criteria for integrating the knowledge management into the network resource specifications. This goal has been obtained by using the GDMO+ method. Our intelligent agent approach provides advantages to the management networks: scalability, robustness, upgradability, and performances; with a shorter distance to be managed from the systems, the distributed intelligent management allows faster reactivity and even

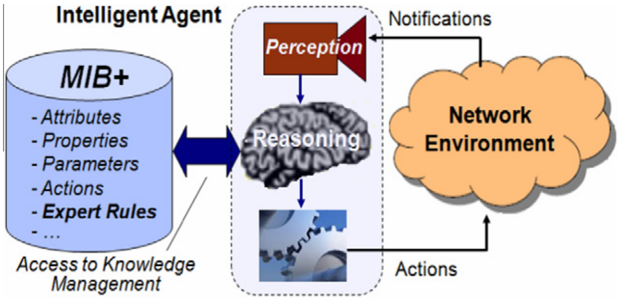


Fig. 2. Intelligent agent architecture.

pro-activity, and less resources consumption. We have chosen TMN as a target network standard to develop the next sections because of its more advanced modeling capabilities and structuring degree.

The GDMO is organized into archetypes that are standard formats used in the definition of a particular aspect of the object. A complete agent definition is a combination of a relationship between a class of managed objects, package, attribute, group of attributes, action, notification, parameter, connection of name, and behavior (ISO/IEC DIS 10165–4 / ITU-T., 1993). The elements that currently form the GDMO standard do not make a reference to the management knowledge base. Until now, the agents have not been able to use the knowledge that the base of knowledge provides which collects the management operations and controls a management domain. We observe the need to define new structures for those cases in which it is necessary to express the knowledge. To solve this problem, we suggest a refinement of the MOC template. Thus, the description of certain aspects of managed object knowledge, for example, the definition of expert rules, can be supported. We proposed adding a new element in the GDMO standard named *RULE*, as shown in Fig. 3.

This attribute will define all the aspects related to management knowledge in a specific intelligent agent. In this way, we implemented a standard agent with knowledge characteristics that can generate a new type of intelligent agent, requiring only the management knowledge specification of its principal classes.

3.1. Managed object class archetype

In our study, the resources being monitored and controlled are modeled as intelligent agents. An MOC template is the base of the formal definition of an intelligent agent. This template is used to define the different kinds of agents that exist in the system. Classes describe what information and services they provide each intelligent agent, and GDMO defines the format for this information. In this way, we ensure that the intelligent agents defined in system A can be easily interpreted in system B.

The set of MOCs and instances under the control of an intelligent agent is an MIB, the abstraction of network resources properties and states for management purpose (ITU-T, 1996a; ITU-T, 1996b). The attribute values for an instance are accessed by issuing set and get requests to change or retrieve the attribute values, respectively. Thus, intelligent agents need a standard structure template on each managed object where they need to be hosted. Each intelligent object class may be seen as the integration of the following basic components: packages, name bindings, and behavior characteristics.

```
<IA-label> MOC
   [DERIVED FROM
      <IA-label> [,<IA-label>]*;]
    [CHARACTERIZED BY
      <IA_properties-label>[,<IA_properties-label>]*;]
    [CONDITIONAL PACKAGES
      <IA_properties-label> PRESENT IF condition;
      <IA_properties-label>] PRESENT IF condition]*;]
REGISTERED AS object-identifier;
```

The management scheme just cited clearly encompasses object-oriented concepts and features. The ‘‘DERIVED FROM” clause plays a very important role in determining the relations of inheritance that make 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 that are defined. The intelligent agent will be characterized by one or more mandatory packages while they may also comprise conditional ones. The ‘‘CHARACTERIZED BY” construct identifies the specific peculiarities of the class that characterizes it with regard to its superclass. This clause defines a package or packages that are always present when the managed object is included in the system, and it is used to specialize an intelligent agent by adding new characteristics. The ‘‘CONDITIONAL PACKAGES” clause is used to identify those packages that may or may not be included each time the managed object of this class is instantiated. Managing functions may request that particular conditional packages be present when they create a managed object instance. Finally, the ‘‘REGISTERED AS” clause identifies the location of the MOC on the OSI registration tree.

3.2. Package archetype

The smallest re-usable entity of management specification is not the object class, as is the case in other object-oriented frameworks, but the package. The package template is used to specify the characteristics that represent a consistent set of specifications about a network resource. The package template is a combination of behavior definitions, attributes, attributes groups, operations, notifications, and parameters. One purpose of the package is to provide a set of reusable definitions that can be used in several intelligent agent class specifications. We add a new feature to the current template package that has been adapted in the GDMO standard. In addition to the properties just indicated, we suggest the incorporation of a new property called *RULES* and its associated

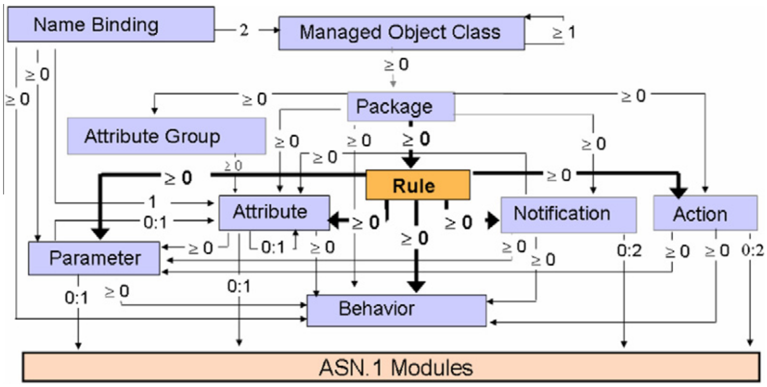


Fig. 3. Template relations in GDMO+ methodology.

archetype called *RULE*, which contains all the specifications of the intelligent knowledge management base. The next definition shows the elements of a package template, in which it is possible to observe the new property RULES.

```
<IA-properties-label> PACKAGE
  [BEHAVIOUR <behaviour-label> [,<behaviour-label>]*;]
  [ATTRIBUTES
    <attribute-label> propertylist [,<parameter-label>]*
    [,<attribute-label> propertylist [,<parameter-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;
```

The expert rule is a piece of code that can be seamlessly loaded into an agent to supply it with more new capabilities. These new properties have all the management knowledge about a particular resource. Each intelligent agent has two types of knowledge: local network management for individual problem solving and community and local network management for coordination of associated node activities. The agent's knowledge base defined in the MIB+ consists of local problem-solving competence and community interaction for coordinating actions among them. Similar to the other properties in an intelligent agent, the property RULES need a template.

3.3. The expert rule template structure

Management knowledge will be transported between intelligent agents and should be described by using the prescriptions of the structure management information language using the CMIP protocol and Common Management Information Services (CMIS). The CMIS defines a system of network management information services and supports knowledge exchange between network management applications and management agents. There are different knowledge representation techniques for structuring knowledge in an intelligent system. The three most widely used techniques are expert rules, semantic nets, and frames. In our work, we are going to represent the knowledge management in production rules or simply rules (Brachman & Levesque, 2004). Rules specify the actions for the inference engine to take when the premise or conditions in the rule are true. Rules are expressed as "IF-THEN" statements, which are relatively simple, very powerful, and very natural to represent expert knowledge. Production rules typically represent management logic, which is the network knowledge management used to decide what to do with the various patterns of data values that exist in the domain model and which procedural logic statements to execute when. A major feature of a rule-based system is its modularity and modifiability that allow for incremental improvement and fine tuning of the system with virtually no degradation of performance.

Template RULE permits the normalized definition of the specifications of the expert rule to which it is related:

```
<IA_rule-label> RULE
  [PRIORITY <priority>;]
  [BEHAVIOR <behavior-label> [,<behaviour-label>]*;]
  [IF occurred-event-pattern [,occurred-event-pattern]*]
   [THEN sentence [, sentence]*;]
REGISTERED AS object-identifier;
```

This template allows an intelligent agent class to have properties that provide normalized knowledge of a management dominion. The first element in a definition is headed. It consists of the name of the management expert rule <rule-label> and a key word that indicates the type of template RULE. After the head, the following elements compose a normalized definition of the management knowledge:

- BEHAVIOR: This construct is used to extend the semantics of previously defined templates. It describes the behavior of the rule. This element is common to the other templates of the GDMO standard.
- PRIORITY: Each production rule in a rule base is a single piece of knowledge that has the capacity to draw a certain conclusion from some evidence. If there are two sources of evidence for some hypothesis, then this value represents the priority of the rule, that is, the order in which competing management actions will be executed by intelligent agents.
- IF: We can add a logical condition that will be applied to the events that have occurred or their parameters. Multiple conditions are joined by logical operators such as AND or OR, and the premise evaluates to true if all or at least one of the conditions evaluates to true for AND and OR, respectively. Rules fire as soon as events are matched with the premises or antecedents of these rules. The premise of a rule examines parameter or slot values, and once the condition evaluates to true, then the action part is executed. Those events should be defined in the notification archetype.
- THEN: An agent's repertoire of tasks represents its capabilities or methods. These are actions and diagnoses that the management platform makes as an answer to network events that have occurred. These actions can consist of setting other parameter or slot values or invoking methods on an instance or a class. Each task can have its procedural "how to do" component represented as expert rules. Those operations should be previously defined in the action template.
- REGISTERED AS is an object-identifier. This clause identifies the location of the expert rule on the OSI registration tree. The identifier is compulsory.

Based on this GDMO+ structure, we have defined a large number of kinds of production rules. In the next section, we give two examples of expert rules.

4. Example of a management expert rule

This section shows a complete example of expert rules integration in the GDMO+ proposed standard. It defines an intelligent agent named *radio transceiver RT21* corresponding to a real device in the network. *RadioTransceiver* intelligent agent includes the compulsory *TransceiverPackage*, which contains all the specifications corresponding to the device. This is a device that both sends and receives radio signals. Its primary purpose is to broadcast the signals. The transmitter and the receiver share a common circuitry into a single housing such as transponders, transverters, and repeaters. These units typically offer the convenience of multiple functions such as establishing radio channels, controlling signals, monitoring stations, monitoring alarm conditions, controlling logic to activate operations in response to commands received over said communications network, and so on. The type of alarm is modeled for both network elements. We model the effect of the microwave link alarms between these network hosts.

```
radioTransceiver_RT21 MOC
  DERIVED FROM radioTransceptor;
  CHARACTERIZED BY transceiverPackage;
REGISTERED AS {nm-MobjectClass 1};
```

Two typical examples of expert rules used in our GDMO+ specification are *transmissionError* and *receptionError*. These expert rules have been associated with the defined class by means of the RULES clause, and they are defined by using the RULE template. The expert rules are used to capture and detect anomalies or defects of operations produced in the transceiver device and suggest the necessary measures for solving the problem.

```
transceiverPackage PACKAGE
  ATTRIBUTES
    reception Power   GET,
    sense         GET,
    speedTransmission   GET, . . .;
  NOTIFICATIONS
    damageFeeding, inferiorLimit, repairAction;
  RULES
    transmissionError, receptionError;
REGISTERED AS {nm-package 1};
```

When a mistake occurs, these rules go to the agenda system. These rules are fired when the conditions are right. The first rule *transmissionError* is devoted to the detection of errors in the data transmission module of the radio transceiver RT21. The second rule *receptionError* is in charge of detecting failures in the power supply of the transceiver RT21. Both rules provide recommendations on how to solve the failures.

```
transmissionError RULE
  PRIORITY 4;
  BEHAVIOR transmissionErrorBehavior;
  IF (?date ?time1 ?st1 SPU_1_AIS_TX ?st2 ALARM)
    (?date ?time2 ?st1 SPU_1_AIS_TX ?st2 ALARM &
    : (<(ABS(? ?time1 ?time2)) 1.00))
  THEN ("Severity:" PRIORITY), ("Diagnostic:
    "It damages in the modulate transmission between", ?st1,
   "and" ?st2),
    ("Recommendation "Revision transceiver");
REGISTERED AS {nm-rule 1);
receptionError RULE
  PRIORITY 3;
  BEHAVIOR receptionErrorBehavior;
  IF (?date ?time1 ?st1 SPU_1_AIS_RX ?st2 ALARM)
    (?date ?time2 ?st1 SPU_1_AIS_RX ?st2 ALARM &
    : (<(ABS(? ?time1 ?time2)) 1.00))
  THEN ("Severity:" PRIORITY), ("Diagnostic:
    "It damages in the modulate reception between",
?st1, "and" ?st2),
    ("Recommendation "Revision remote transceiver");
REGISTERED AS {nm-rule 2);
```

This element definition and knowledge management are placed on an MIB module, written by following standard GDMO+. The information management network has been stored in the MIB tables. The MIB modules are implemented in intelligent agents, and their instances are maintained in the MIB caches. Several CMIS operations can be applied to the intelligent agents such as creating/deleting object instances, getting/setting attribute values, executing management actions, and so on through CMIP messages, which work at the application level in an OSI protocol stack (Clemm, 2006).

5. A diagnostic model for a private microware network case

In order to validate our approach, we have developed a production system based in intelligent agents that integrates the management knowledge into the resources specification of a network. The following system exemplifies how network topology information, resources properties, and management information may be used to dynamically develop an intelligent diagnostic when errors occur. For this purpose, we have used the GDMO+ specifications. The primary objective of our system is to automate the network management as much as possible. Our intelligent system should meet the following requirements (Maggiora, Elliott, Pavone, Phelps, & Thompson, 2000):

(1) It should perform non-intrusively. The management activity should not interfere with normal operations of the network. It should only intervene when necessary.
(2) It should be robust. Management applications should be able to perform even when the network is not fully operational, as management is mostly needed in abnormal situations, for example, when connections are broken.
(3) It should scale up well to growing networks.

We are going to study the diagnosing microwave link problems in a private network. We study an example of alarm detection and resolution of incident concerning a private network. For this purpose, we used a telecommunications network that belongs to a company in the electrical sector (SEVILLANA-ENDESA's a Spanish power utility) (León, Mejias, Luque, & Gonzalo, 1999). Part of long-distance traffic in this net is controlled by a wireless intelligent system distributed throughout this private network, as seen in Fig. 4.

Nowadays, there is an expert system that controls company networks. In this section, we will design and develop a new version of this expert system. This expert system named *SCANEXP* is developed according to our proposal GDMO+. In the next sections, we will study the main elements of the system.

5.1. The Communication Supervisory System

Detection mechanisms are implemented in real time in our prototype and have been embedded within the network elements,

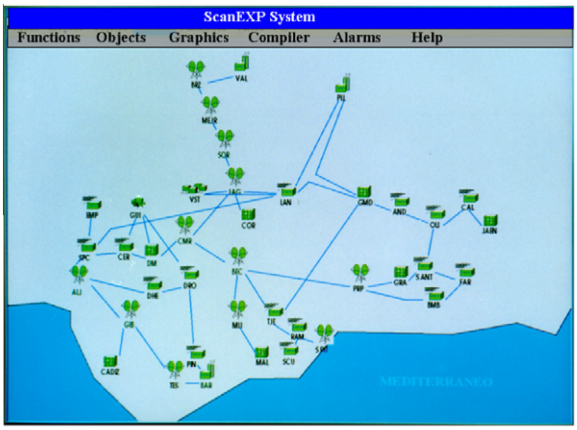


Fig. 4. Power company network map in Andalucía.

network protocols, and devices. Intelligent agents need to communicate with operators. SCANEXP operations use a supervision system called *Communication Supervisory System* (CSS). The CSS takes care of interfacing a set of remote Terminal Units (RTU) to the traditional switched network. Nowadays, current trends in power system controls lead to intelligent Remote Terminal Units (RTUs) that can support order functionality. 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), placed on the main company building, and the RTUs are installed at different stations. The use of an SCADA system is due to the management limitations of network communication equipment.

Further, the intelligent agent is able to communicate and negotiate with the other intelligent agents. He is deliberative and may coordinate his actions with the others. Collaborative intelligent agents are useful, especially when a task involves several systems on the network. If coordination occurs without collaboration, then it needs negotiation. For these purposes, we have the CSS. The CSS listener is the component responsible for the monitoring of the RTU's message received by the network resources. It receives the RTU's messages and verifies whether the messages should be further redirected to the integrated expert system (real invoked service) or to the network operator. The CSS 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 CSS in order to solve the failures that could appear or to send a technician to repair the stations equipment. Fig. 5 shows a hierarchical architecture that represents the whole network, the RTU objects, and the network elements. The SCADA hierarchical architecture fits into the managers of agents. This shows how the communication between the expert agents or interfaces, utilizing RTUs as an inbox for messages received and as a destination for messages sent from each agent in the system, separates the correspondences by the destination agent of each message.

Collaborative management actions between intelligent agents are achieved through the CSS. RTUs receive the different management actions from the control center. These operations are the answer to the management's request in the network resources. The CSS has the capability of selecting the intelligent agent that is best suited for satisfying the client's requirement, without the client being aware of the details about the agent. The management agent has the responsibility of executing the agents of the multi-agent community, finalizing the execution of the agents of the community, and running checks by soliciting the user in specified agents. Agents use CMIP standard protocol for communication purpose, so interconnectivity is guaranteed. If we want to integrate a new RTU in an environment, then we will have to develop a new intelligent agent and its corresponding MIB module.

5.2. System operations

Intelligent agents are responsible for the wireless traffic in their cell area. These stations are typically connected via microwave links. The network topology is logically a star and physically a tree where the traffic to several base stations is distributed over a chain of microwave links and leased lines. When a link fails, a number of alarms is generated and passed to the CSS. One has to reason based on the network topology to correlate the alarms and detect the root cause of the failure. Since the topology is subject to frequent changes, the intelligent model should be easy to adapt to the changed configuration. Fig. 6 shows the conceptual architecture of our system.

Our system has two major components: an inference engine and a knowledge base.

- The inference engine is an integrated collection of problem-solving algorithms that combine and apply relevant data, facts, and rules in the knowledge base to achieve a goal or draw a conclusion. In order to do this, it first reads the agenda to see what high-level instructions have been defined for the application (Ruiz-Mezcua, Garcia-Crespo, Lopez-Cuadrado, & Gonzalez-Carrasco, 2011). It then determines which management logic structures, or rules, it needs to execute in order to accomplish the agenda instructions. These execute the inference engine notes about which procedural logic structures, methods, or functions should be executed and proceed to do so. During the entire process, the inference engine may find a need to retrieve, update, or otherwise manage internal or external data. The inference engine processes rules can infer new knowledge from already known knowledge and depending on the particular problem-solving situation. If there are no data matching the premise of the identified rule present in the knowledge base, then further rules are identified with matching actions to resolve these new goals (sub-goals). These are added to the goal list. As soon as the premise of a rule that the inference engine is testing becomes true, then the rule executes and yields the data necessary for the next rule on the goal list to fire.

In our demonstrations, we have used ART*Enterprise 2.0, a commercially available expert system shell made available by Brightware's. ART*Enterprise 2.0 is a set of programming paradigms

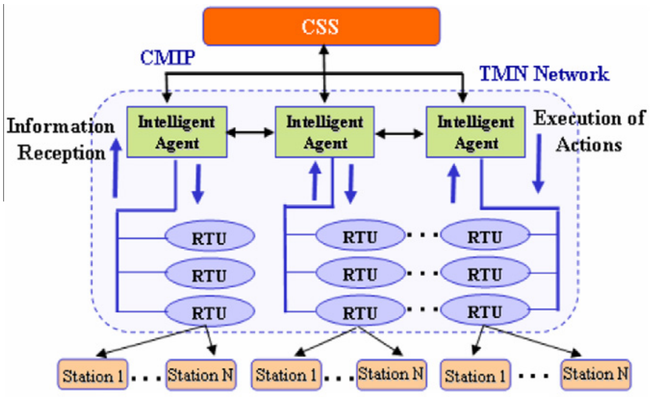


Fig. 5. SCANEXP system features.

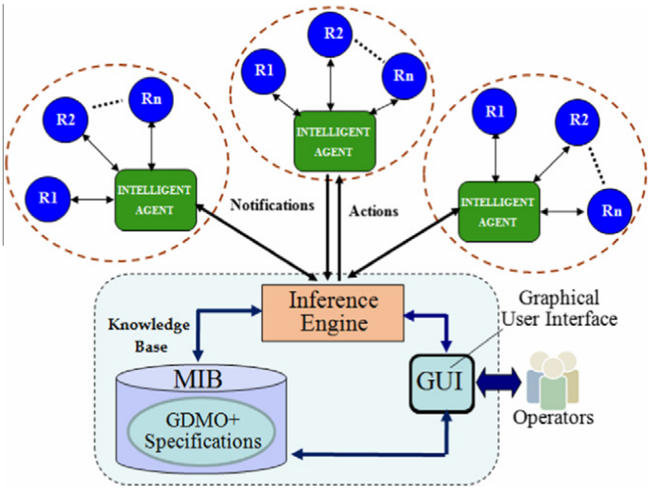


Fig. 6. Communication Supervisory System (CSS).

and tools that are focused on the development of efficient, flexible, and commercially deployable knowledge-based systems. It provides a user interface and a programming library to create a reasoning system with discrete random variables and also supports Value-of-Information computations and cost-based decision theoretic troubleshooting. By using an existing general purpose tool, we were able to build a standard and extensible platform with proven performance and quality.

- The knowledge base: This is the core of the system and is a collection of facts and if-then production rules that represent stored knowledge about the problem domain (Negnevitsky, 2002). The first aim of this approach has been to make a classification of management knowledge that we met during our study and integrate this intelligence in the managed objects in order to obtain intelligent agents. The knowledge base comprises expert rules and facts expressed in the ARTScript programming language ART*Enterprise representing the management knowledge. The knowledge base contains both static and dynamic information and knowledge about different network resources and common failures. The resultant knowledge source has about 600 rules about network resources. This initial knowledge has been acquired from the experts in the management domain. The knowledge base of our system can be extended by adding new higher level rules and facts through the graphic user interface (GUI).

5.3. Graphic user interface

In our system, we implemented a graphical user interface (GUI) written in Java running on a Server to control the whole embedded system. It is a set of I/O handling routines for managing the system and permits the management of the system by the user. The user interface is easy to use. To access the interface only requires a Web browser such as Explorer o Mozilla. The GUI controls the inference engine and manages system input and output.

The management system in normal operation generates different notifications and alarms. An alarm is an event asynchronously generated whenever the value of some quality indicator crosses a predefined threshold either positively or negatively. Those alarms are caused when an incident occurs. When a microwave link fails and the connection to the agent is broken, then an intelligent agent can send alarms such as *incoming_signal_missing*, *D_channel_failure*, *BER_over_limit*, *PCM_failure*, *LAPD_failure*, *BCCH_missing*, and so on. These events are accompanied by parameters that show different aspects of the events. Further, some of the alarms may be missing or additional alarms may be included that constitute noise. The GUI is enabled to receive the warnings and events. The received data is then passed to *agendaPacket*() to be saved into the buffer. In the given example, the network elements send various types of alarms on detecting abnormal behavior:

```
F1 (31/01 1100.200 sevilla.us.es 7_TX_C2 tajo.us.es ALARM)
F2 (31/01 1103.168 sevilla.us.es 7_TX_C2 tajo.us.es ALARM)
```

Each alarm contains main information about circumstances that caused the incident. In a previous relation, taking as an example the first fact, the following information is obtained:

- Implied equipments: sevilla.us.es and tajo.us.es
- Device Kind: Radio Transceiver
- Alarm kind: 7_TX_C2
- Date and time alarm: 31/01 1100.200

These events or notifications used have been previously defined by using the corresponding notification template and are included in the same class of intelligent agents in which the expert rule acts. The application called *Network Alarm Monitor* provides the ability to collect alarms. This element displays the alarms in a different order, displays the alarm history, filters the alarms based on certain criteria, and so on, as seen in Fig. 7.

These alarms indicate problems that require corrective actions. The management system analyzes and checks the rules that match these conditions. When a rule is ready to fire, then it means that since the antecedent is satisfied, the consequent can be executed. If the antecedent of some rule is satisfied, then this rule is ready to fire and it is placed in the agenda of the system. The results generated by the management system are as follows:

```
FIRE 1: transmissionError f-2
  Severity 4
  Diagnostic: It damages in the modulate transmission
   between sevilla.us.es and tajo.us.es
Recommendation “Revision transceiver”
  1 rules fired.
  Run time is 0.074 seconds, 27.0270 Rules/Sec.
```

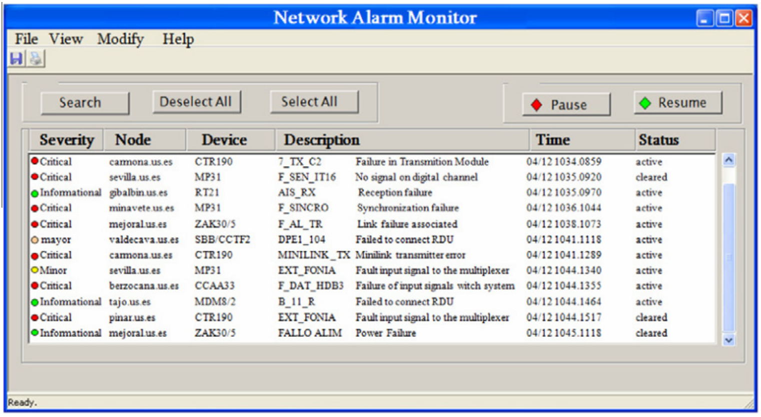
On the other hand, from a developer’s perspective, creating an MIB database from a higher-level set of abstractions has become difficult by using traditional database technology. The MIB modeling technique generates an information model comprising many highly interrelated objects. To facilitate the management of the MIB, our GUI provides a variety of tools for the creation, editing, and updating of MIBs and an MIB Browser. The MIB Browser is a tool that allows the administrator to interactively inspect the definitions of management object classes. This tool browses the inheritance architecture classes generated, as seen in Fig. 8.

The MIB Browser simplifies the task of defining managed object definitions. The MIB Browser may also perform the following functions: retrieve and display MIB data in a human readable form, allow it to look smarter by compiling more MIB definitions, set MIB variables, create, modify, or delete conceptual table rows basic alarm (trap) management, and view the text file in a graphical manner by usually featuring an MIB tree. In addition to pulling out data, when new knowledge is uncovered, it will need to be incorporated into the system to keep it updated. The user interface GDMO Template Editor and Parser allows modifying or including new expert management rules in the managed objects definition.

6. System evaluation and testing

The system developed under GDMO+ specifications is quite different from the traditional expert system, despite the fact that the scenario testing and rules underlying the analysis are similar. As we have already seen in previous sections, ScanEXP provides a lot more than just a detection of the faults, which is all that the traditional expert system is capable of at the moment. In the first place, our system is able to automatically resolve most of the faults produced in the communications network. On the other hand, GDMO+ intelligent agents can decentralize processing and control and improve management efficiency. In order to check the improvements of the GDMO+ intelligent agents with a real application in this section, we have compared ScanEXP with a traditional expert system. For this purpose, ScanEXP has been tested with regard to the following aspects: system validation using test cases, validation by case studies, and validation against human experts.

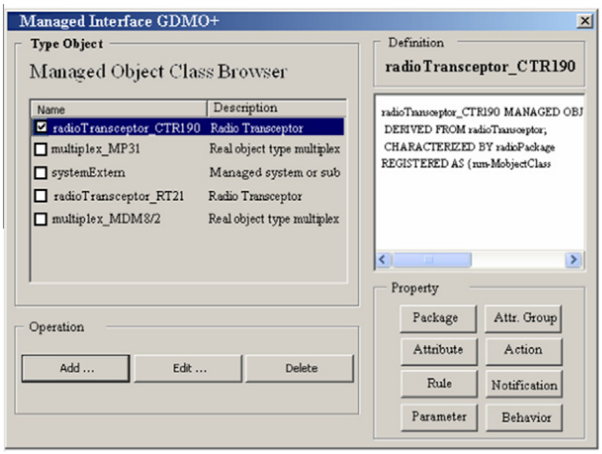
In this section, we compare intelligent agent performance with the traditional managed object one by evaluating the traffic around the management agent and response time in retrieving CMIP variables and knowledge management. An important aspect of the design and implementation of an intelligent system is determination



Severity	Node	Device	Description		Time	Status
● Critical	carmona.us.es	CTR190	7_TX_C2	Failure in Transmition Module	04/12 1034.0859	active
● Critical	sevilla.us.es	MP31	F_SEN_IT16	No signal on digital channel	04/12 1035.0920	cleared
● Informational	gibalbin.us.es	RT21	AIS_RX	Reception failure	04/12 1035.0970	active
● Critical	minavete.us.es	MP31	F_SINCRO	Synchronization failure	04/12 1036.1044	active
● Critical	mejoral.us.es	ZAK30/5	F_AL_TR	Link failure associated	04/12 1038.1073	active
○ mayor	valdecava.us.es	SBB/CCTF2	DPE1_104	Failed to connect RDU	04/12 1041.1118	active
● Critical	carmona.us.es	CTR190	MINILINK_TX	Minilink transmitter error	04/12 1041.1289	active
○ Minor	sevilla.us.es	MP31	EXT_FONIA	Fault input signal to the multiplexer	04/12 1044.1340	active
● Critical	berzocana.us.es	CCAA33	F_DAT_HDB3	Failure of input signals witch system	04/12 1044.1355	active
● Informational	tajo.us.es	MDM8/2	B_11_R	Failed to connect RDU	04/12 1044.1464	active
● Critical	pinar.us.es	CTR190	EXT_FONIA	Fault input signal to the multiplexer	04/12 1044.1517	cleared
● Informational	mejoral.us.es	ZAK30/5	FALLO ALIM	Power Failure	04/12 1045.1118	cleared

Ready.

Fig. 7. Network alarm monitor.



Name	Description
☑ radioTransceptor_CTR190	Radio Transceptor
□ multiplex_MP31	Real object type multiplex
□ systemExtern	Managed system or sub
□ radioTransceptor_RT21	Radio Transceptor
□ multiplex_MDM8/2	Real object type multiplex

radioTransceptor_CTR190 MANAGED OBJ
DERIVED FROM radioTransceptor;
CHARACTERIZED BY radioPackage
REGISTERED AS {nm-MobjectClass

Fig. 8. System object classes dialog box.

Table 1
Comparison traditional expert system & ScanEXP.

	Traditional ES		ScanEXP	
	Proceeding time (s)	Rules/s	Proceeding time (s)	Rules/s.
A1	1.525	109.120	1.250	124,0000
A2	6.561	45.773	5.249	55,8202
A3	3.213	69.156	2.975	85,3782
A4	21.758	17.125	17.982	19,2415
A5	0.142	388.983	0.118	432,2034

of the degree of speed in the answer that the network provides. We will discuss the issue of response time for five agents associated to transceiver resources. Every intelligent agent is assigned a particular resource repair task. We test the model by inserting some alarms into the system. We compared our results with those we had obtained with the traditional system. Note that the response time would vary depending on both the agent and the fault type. Fig. 9 shows a sample plot of these parameters that was collected as a part of the experiment, which shows that the speed of the

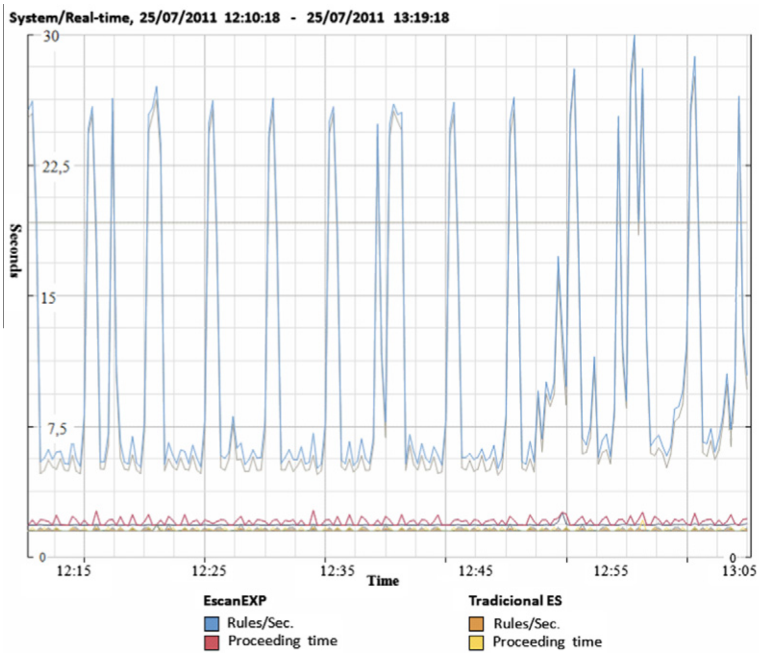


Fig. 9. Performance EscanEXP & traditional ES.

Table 2
Performance management events and alarms.

Alarms number	Autonomous resolution	Autonomous resolution %	Managed actions executed	Operator warnings
100	99	99	151	1
200	190	95	202	6
400	369	92.25	501	16
600	562	93.66	793	16
800	745	93.12	994	23
1000	946	946.1	1528	49

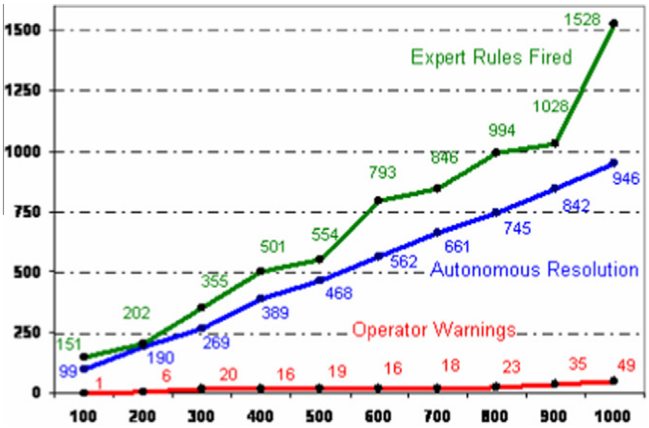


Fig. 10. Filtration process effectiveness.

ScanEXP system improves the proceeding time and the average of the traditional expert system.

In Table 1, we present the average set-up time for some measurements. As the table shows, the results for ScanEXP are 15.1% better than proceeding time and 19.5% better than executing time rules/sec in the traditional expert system.

Another test of significance is the analysis of the number of alarms that have been automatically resolved by ScanEXP and the warnings received by the system operator. As noted in the next Table 2, ScanEXP performs satisfactorily with about a 94.6% rate of success in real cases. It is also noted that the performance of ScanEXP may depend considerably on the facts stated. The more information is input, the better the chance of diagnosing the likely causes of the problems in the network.

From these results, we can establish that our system with more than 600 integrated expert rules has produced excellent results, and we can establish the following conclusions:

- Intelligent agents solved nine of the ten faults studied. We note that ScanEXP performs satisfactorily with about a 95% rate of success in real cases. We considered a fault automatically resolved if the probability of abnormal network variable behavior is less than 0.1, and resolution is greater than 0.9. These efforts should reduce the number of false alarms to the operator.
- Filtration process effectiveness is very high after extensive field testing proved to be capable of filtering 90% of produced alarms with a precision of 95% in locating them. This has the advantage of a decreasing percentage in the amount of indications presented to the operator, as seen in Fig. 10.
- The speed of the system improves the proceeding time and the average.

7. Conclusions

Current networks are very complex and demand ever-increasing levels of quality, thus making their management a very important aspect to be taken 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. The use of intelligent agents in network supervision can help the administrator in using the maximum capabilities of the network management platform. To improve the techniques of intelligent management in a communications network, we propose the possibility of integrating, normalizing, and distributing the knowledge management within the actual definition of the managed objects.

In this article, we showed possibilities to apply and integrate artificial intelligence techniques in network management and supervision by using OSI. In fact, these kinds of applications underline the power of GDMO and CMIS as both simple and powerful knowledge modeling language, thus offering new possibilities that traditional management do not offer. The problems of the actual network management paradigms let one think that the properties of the resources together with intelligence are a good recipe to deal with a more hierarchical and decentralized management. 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.

To show the feasibility of the GDMO+ standard, a reliable monitoring experiment was presented, where agents specialize their capabilities by acquiring different skills. The experiment we presented revealed the great advantage of using expert rules to implement management intelligent functions. One of the goals of the experiments was to demonstrate that the distributed intelligent perspective can provide more efficient resource utilization, by supporting control functionality and intelligence that are not feasible otherwise. We conclude by pointing out an important aspect of the obtained integration: the solution not only masks possible faults but also optimizes the management functions and efficiency of the distributed services and their resources by using an artificial intelligent strategy, while ensuring a high degree of functionality.

Our work of knowledge integration on the intelligent agents can be viewed as a first step toward automated management by using intelligent agents. One direction of our future work is to improve the agent's performance. We will also study in greater depth how to better incorporate the previous knowledge available at a network node. To that end, we plan to further investigate the feasibility and limitations of other knowledge representation techniques such as semantic networks, Bayesian networks, Ontology Engineering, and so on. These methods increase computational complexity, something we did not want in our initial investigation. In addition to the fault detection functional area, we would also like to expand the scope of our current work to other functional areas such as accounting management, configuration management, performance management, and security management. Further, the intelligent agent can be generalized to different types of network nodes with minimal network-specific information required beforehand. We are studying the way to integrate the expert knowledge in the management Internet model. The Internet management model does not use the Object Oriented Programming such as is used by the OSI model. This is one of the reasons for the Internet model's simplicity. The definitions contain objects, specified with ASN.1 macros. The resources specifications can only be groups of scalar variables and cells tables; in spite of not being an Object Oriented Programming model, we can use the tables as classes where the attributes are the table columns, and every file contains an instance of the class.

References

Akinyokun, O.C., Imianvan, A.A. (2006). Mobile agent system for computer network management Proceedings from the International Conference on Advances in, Engineering and Technology, p.796–808.

Barco, R., Díez, L., Wille, V., & Lázaro, P. (2009). Automatic diagnosis of mobile communication networks under imprecise parameters. *Expert Systems with Applications, 36*(1), 489–500.

Brachman, R. J., & Levesque, H. J. (2004). *Representation and reasoning*. San Francisco, CA: Elsevier/Morgan Kaufmann.

Chantaraskul, S., & Cuthbert, L. (2008). An intelligent-agent approach for congestion management in 3G networks. *Engineering Applications of Artificial Intelligence, 21*(4), 619–632.

Clemm, A. (2006). Network Management Fundamentals. Cisco Press.

Doukas, H., Patlitzianas, K. D., Iatropoulos, K., & Psarras, J. (2007). Intelligent building energy management system using rule sets. *Building and Environment, 42*(10), 3562–3569.

Hebrawi, N. (1995). GDMO, Object modelling and definition for network management. Technology appraisals.

ISO/IEC DIS 10165–4 / ITU-T. (1993). Recommendation X.722, Information Technology - Open Systems Interconnection - Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objects (GDMO), International Organization for Standardization and International Electrotechnical Committee.

ISO/IEC and ITU-T. (1998), Information Processing Systems - Open Systems Interconnection - Systems Management Overview. Standard 10040–2, Recommendation X.701.

ISO/IEC and ITU-T. (1998). Information Processing Systems - Open Systems Interconnection - Systems Management Overview. Standard 10040–2, Recommendation X.701.

ITU-T. (1992). Recommendation X.700, Management Framework for Open Systems Interconnection (OSI). CCITT Applications.

ITU-T. (1996). Rec. M.3010, Principles for a Telecommunications Management Network (TMN). Study Group IV.

ITU-T. (1996). Recommendation M.3400, TMN Management Functions. Study Group IV.

Lei, Y., He, Z., & Zi, Y. (2009). Application of an intelligent classification method to mechanical fault diagnosis. *Expert Systems with Applications, 36*(6), 9941–9948.

León, C., Mejias, M., Luque, J., & Gonzalo, F. (1999). Expert System for the Integrated Management of a Power Utility's Communication System. *IEEE Trans on Power Delivery, 14*(4), 1208–1212.

Li, S. (2007). AgentStra: an Internet-based multi-agent intelligent system for strategic decision-making. *Expert Systems with Applications, 33*(3), 565–571.

Liao, T., & Seret, D. (1991). Network management: interoperability and information model. *Computer Communications, 14*(10), 588–597.

Maggiora, Paul L. Della, Elliott, Christopher E., Pavone, Robert L., Phelps, Kent J., & Thompson, James M. (2000). *Performance and Fault Management*. Cisco Press.

Morris, Stephen B. (2003). *Network Management MIBs and MPLS: Principles Design and Implementation*. Addison Wesley.

Negnevitsky, Michael. (2002). *Artificial intelligence. a guide to intelligent systems*. New York: Addison Wesley.

Oztemel, E., & Tekez, E. (2009). A general framework of a Reference Model for Intelligent Integrated Manufacturing Systems (REMIMS) Engineering Applications of Artificial Intelligence. *Artificial Intelligence Techniques for Supply Chain Management, 22*(6), 855–864.

Power, Y., & Bahri, P. A. (2005). *Integration techniques in intelligent operational management: a review Knowledge-Based Systems, 18*(2-3), 89–97.

Ray, P., Parameswaran, N., & Lewis, L. (2010). Distributed autonomic management: An approach and experiment towards managing service-centric networks, Journal of Network and Computer Applications. *Advances on Agent-based Network Management, 33*(6), 653–660.

Ruiz-Mezcua, B., Garcia-Crespo, A., Lopez-Cuadrado, J. L., & Gonzalez-Carrasco, I. (2011). An expert system development tool for non AI experts. *Expert Systems with Applications, 38*(1), 597–609.

Shin, T., Chin, S., Yoon, S. & Kwon, S. (2011). A service-oriented integrated information framework for RFID/WSN-based intelligent construction supply chain management, Automation in Construction, In Press, Corrected Proof, Available online.

Song, G., He, Y., Chu, F., & Gu, Y. (2008). HYDES: A Web-based hydro turbine fault diagnosis system. *Expert Systems with Applications, 34*(1), 764–772.

Stallings, W. (2000). *SNMP, SNMPv2, and CMIP: the practical guide to network*. Reading Mass: Addison-Wesley.

Tacconi, D., Miorandi, D., Carreras, I., Chiti, F., & Fantacci, R. (2010). Using wireless sensor networks to support intelligent transportation systems. *Ad Hoc Networks, Vehicular Networks, 8*(5), 462–473.

Vallejo, D., Albusac Castro-Schez, J. J., Glez-Morcillo, C., & Jimenez, C. (2011). A multi-agent architecture for supporting distributed normality-based intelligent surveillance. *Engineering Applications of Artificial Intelligence, 24*(2), 325–340.

Waiman, C., Leung, L. C., & Tam, P. C. F. (2005). An intelligent decision support system for service network planning. *Decision Support Systems, 39*(3), 415–428.

Yang, S., & Chang, Y. (2011). An active and intelligent network management system with ontology-based and multi-agent techniques. *Expert Systems with Applications, 38*(8), 10320–10342.

Zuidweg, J. (2002). *Next generation intelligent networks*. Boston: Artech House.