# Building Reusable Knowledge Models for the Communication Networks Domain


**Marilza A. de Lemos**
mlemos@lsi.usp.br
LSI - EPUSP
University of São Paulo
Av. Prof. L. Gualberto 158
São Paulo - Brazil
05508 - 900

**Leliane N. de Barros**
leliane@ime.usp.br
IME - BCC
University of São Paulo
Rua do Matão, 1010
São Paulo-Brazil
04717-020

**Volnys Bernal**
volnys@lsi.usp.br
LSI-EPUSP
University of São Paulo
Av. Prof. L. Gualberto 158
São Paulo-Brazil
05508-900

**Jacques Wainer**
wainer@dcc.unicamp.br
Instituto de Computação
University of Campinas
Campinas - SP - Brazil



**Abstract**: The construction of Knowledge Based Systems (KBS) in complex domains typically involves the reuse of domain ontologies; selection and adaptation of Problem Solving Methods (PSMs). The development of such components is proposed in the field of Knowledge Engineering to allow sharing and reuse of knowledge, avoiding efforts in the construction of KBSs from scratch. This paper presents an experience of reuse of knowledge components from Benjamins' PSMs Library focusing on the construction of a fault diagnosis method to solve communication network management problems. The suitability criteria of methods proposed by the Benjamin's Library were used to guide the selection of methods. The work involved an intensive study of network management area in order to specify the domain features that can match the PSMs suitability criteria. The method proposed reasons over two network model types: fault-states causal models and topology model. Meta-knowledge about the domain ontology was also elicited in order to guide the diagnosis system for the manipulation of multiple domain models and for the dynamic selection of alternative methods. The development of this work is an initial step in the construction and reuse of ontologies, PSMs and domain models for automation of complex tasks in the network management application area.


## 1 INTRODUCTION

Network management tasks have been executed typically by network administrators with the support of management systems (NMS - Network Management Systems) which provide means to monitor and control network resources. However, administrators have been finding difficulty in the execution of complex tasks of managing, just as the diagnosis task, due to the great number and diversity of the resources to be managed [12] [20]. Those reasons have been driving the research in the area, towards the automation of complex tasks in network management. In that context, Artificial Intelligence (AI) techniques have been used to support development of Knowlege Based Systems (KBS) in the domain. In this paper the approach of knowlege modeling is used to propose a reusable solving method for the construction of KBSs that accomplishes the diagnosis task in the fault management domain in communication networks. An application example is presented for the problem of Traffic and Congestion Control. The modeling approach for knowledge acquisition results in a set of reusable modeling components, such as problem solving methods (PSMs), ontologies and domain models. The possibility of reusing the knowledge models has been driving researchers of Knowledge Engineering area to build libraries of reusable models with the purpose of supporting KBSs developers. Thus, some libraries are now already available in the literature such as: CommonKADS library [6], models for domain of Laws [18], models for prediction task [5], models for diagnosis task [3] and models for planning task [2]. The use of a PSM library makes the knowlege acquisition activity less complex because it gathers literature knowledge structured on a reusable way in the building of new knowledge systems.

The task of knowledge acquisition accomplished in this work is supported by Benjamins' Library which consists in a set of methods that may be used to build KBSs which are intended to perform diagnosis task for several domains. The library is organized in a task-method decomposition structure where the diagnosis task is decomposed recursively in sub-tasks and primitive inferences according to the methods set of the library. There are several methods proposed for each sub-task, which may solve it. Depending on the features of the domain in which one wants to execute a task, the choice of a method can be more appropriate than another. The selection of a method in the library is guided by method suitability criteria, which present the conditions demanded by the method for its use. Leaded by such criteria, the network management domain was researched and a set of assumptions was established during selection of methods.

A main assumption in the model-based approach for diagnosis refers to the types of models used to represent the domain. In this work, the method proposed reasons on two network model types: fault-states causal models and topology model. Meta-knowledge about models was also modelled, where units of the models were characterized according to domain and task to be accomplished (diagnosis). This knowledge type was introduced in the domain model which is manipulated by the proposed method.

The structure of the paper is as follows: the section 2 defines a set of assumptions on the domain in order to guarantee the suitability of the proposed method to the domain; the section 3 presents an ontology for network management domain to be used in the domain and task models; the section 4 presents proposed PSM. Finally, the section 5 presents the conclusions.

## 2 ASSUMPTIONS ON THE DOMAIN

The specification, at level of knowledge, of the diagnosis task consists of a domain description and a set of methods that can manipulate such descriptions. This dependence among the domain representation and criteria demanded by the methods implies in the establishment of restrictions and assumptions to the domain. This procedure is necessary for some reasons, such as the complex nature of the problems that KBSs intend to solve, difficulty of modeling the domain in details, efficiency and tractability of the system [9]. This section describes assumptions has been assumed in the domain of network management, in order to guide the selection of PSMs of the library to compose the proposed PSM.

To develop a KBS to accomplish a task in a domain, it is necessary to identify the dependences among the several types of tasks found in that domain [7]. That need originates from the fact that the execution of a task can depend on the previous accomplishment of another task types. In the domain of network management, the diagnosis task needs that a monitoring task is previously accomplished, in order to obtain and to accumulate data regarding the managed resources. Another management application, denominated here as event_generator, should be able of summarizing the monitored information in order to express the occurrence of network fault states, which are typically denominated as events in the domain of network management. In the same way, so that a repair task can be executed, it is necessary that a diagnosis task is previously executed indicating the diagnosis. Therefore, the input and output dependences of these tasks can be presented by the serie:

$$\text{monitoring} \rightarrow \text{event\_ generator} \rightarrow \text{diagnosis} \rightarrow \text{repair}$$

Once efforts have been spent for standardization and observability of network resources by standardization organizations (ISO/OSI, ITU-T, IETF), the monitoring and event_generator tasks are well supported by NMSs (Network Management Systems). Such systems interact with NEs (Network Elements) to obtain data about its behavior so that actions can be taken by network administrators to maintain the network in an acceptable operational level. A management system is a distributed application presenting two main entities: the manager and the agent. In management basic architecture, the manager is the central entity able to sending requests to the agents that are located in network elements. Each agent obtains the requested information accessing its managed resources and sending them to the manager. The agent can still send to the manager notifications known as *traps*.

To guarantee interoperability among heterogeneous hardware and software that constitute the networks, special protocols were developed to allow network management (interaction between manager and agents) such as SNMP (Simple Network Management Protocol) [14] for Internet network architecture and CMIP (Common Management Information Protocol) for OSI (Open Systems Interconnection) network architecture [8]. Besides, network resources that are common to several network configurations have been selected to compose standardized groups of important resources to the network management. Standardized notations were also established to represent such resources which when are modeled like this, are denominated managed objects. The managed objects sets constitute so called MIBs (Management Information Base).

To build the PSM for diagnosis task in network domain we have considered the existence of data generated by other management applications such as monitoring and event_generator applications. Events happen when attributes values of managed objects reach established thresholds and they represent the occurrence of relevant states in the network. The existence of network functionality indicators (managed objects and its attributes), which allow the creation of the events base, facilitates the construction of causal models with fault states to represent the network behaviour. Besides, the explicit representation of cause and effect relationships among states in a causal model is very suitable to the diagnosis task where a cause can represent a diagnosis and an effect can represent a symptom. Thus, a second assumption that has been done to the domain is the existence of causal models with fault states, which represent the network fault behavior in the context of focused management. A third assumption that has been done in the domain is the existence of the network topology model. The introduction of this knowledge in a diagnosis system, allows to divide the space of network fault-states, in such a way that the method is able to go through such a space in an organized way, supported in the network topology. The fourth assumption that has been done to the domain refers to the number of faults that may be diagnosticated by the method during a diagnosis session. Each fault that happens in the network generates a symptoms set. The diagnosis application should identify the cause (the fault) for each symptoms set that has been observed during a diagnosis session. The start of diagnosis sessions can be typically controlled through time windows, that is, symptoms sets are analyzed in a same session since they have happened in the same interval of time. This control avoids that faults are accumulated generating multiple faults situations. Multiple faults are more difficult be identified, they demand more complex methods and larger time of processing than single fault situation. The method proposed in this work takes into account single fault assumption for network management domain since the domain offers ways to obtain temporal information on the occurrence of events in the netwok. A fifth assumption we have been done on the domain is related with generated hypotheses by the method. Since the network must be represented by causal models with fault states, hypotheses are also fault states linked by cause-effect relations. Therefore the causal dependence among states of the model also establishes the dependence among hypotheses.In summary, the assumptions established in the network management domain for the configuration of the proposed method are:

1. occurrence of relevant network states are registered in a data base (denominated here as *event_log*) and it is available for manipulation by method, in order to detect symptoms and to obtain additional information to test hypotheses. The event_log results of performing the monitoring and event_generator applications previously mentioned;

2. existence of causal models with fault states of network elements (NEs);
3. existence of topology model of the management network;
4. single fault;
5. dependent hypotheses.

## 3 MODELLING THE ONTOLOGY OF THE DOMAIN

Once defined the types of models that will represent the network, the units that compose each one of the models and its relationships were modeled according to the domain and the task to be accomplished (diagnosis). Like this, fault-states are the units that compose the state model and network elements (NE) are the units that compose the topological model. Typically, the NE play one or more functions in the network. Therefore, each NE was represented through a model of states in agreement with the function played by the element in the network. The table 1 relates some types of state models that can be built in the domain and MIBs [13] [15] [19] that supply important variables for this.

| NE function | Model type | MIB |
|---|---|---|
| Network_monitor | Network_monitor fault-state model | RFC 1757 (RMON) |
| Router | Router fault-state model | RFC 1213 (MIB-II) |
| Workstation | Workstation fault-state model | RFC 1213 (MIB-II) |
| Mail_server | Mail_server fault-state model | RFC 1566 |
| DNS_server | DNS_server fault-state model | RFC 1611 |

**Table 1** Model types to Internet network management according to managed function

As the network elements interact with each other in a network, there will be situations in that the method should analyze more than one model of states to find the diagnosis. Thus, a classification for the fault-states of the models it was identified in order to allow that PSM navigates among the models of states searching by the diagnostic:

- *initial fault-states*: states which are observed by the method to detect symptoms;
- *local fault-states*: states which can exist in the network element observed [16];
- *remote fault-states*: states which can exist in remote network elements (the other elements belonging to the network) [16];
- *cause fault-states*: states that originated the other observed states. Such states, when found by the method in a diagnostic session, they represent the diagnostic.

Figure 1 shows a state model labelled with ontological domain knowledge. The figure 2 shows the domain ontology obtained for the fault management domain in networks. For the traffic management that is target of a first KBS using the proposed method in our laboratory, a state model was built for NEs have workstation generic function. The figure 3 shows such model.

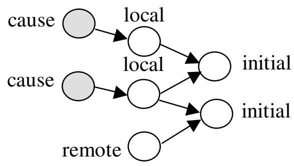


Figura 1 Causal model with four types of fault-states: initial, local, remote and cause

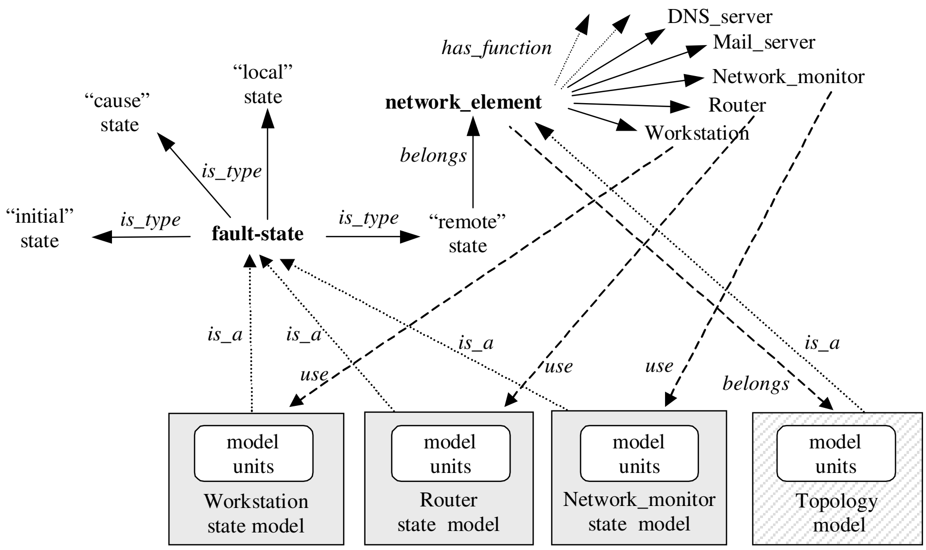


Figura 2 Domain ontology proposed to network management domain

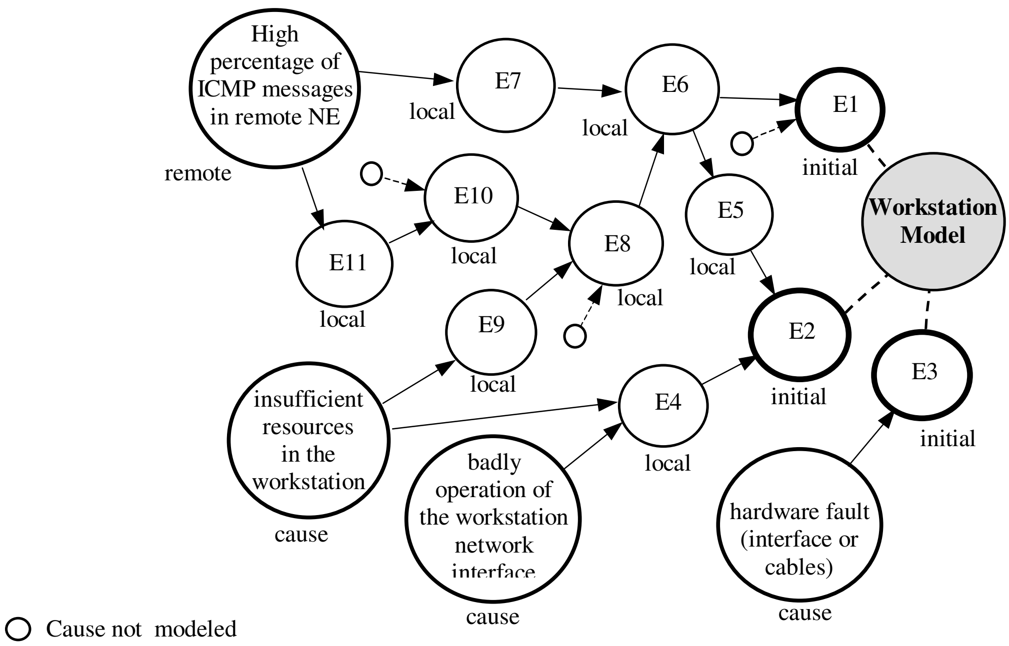


Figura 3 - Workstation fault-state model

The table 2 shows the attributes (variables) of MIBs [15] [19] that should be monitored and related in order to represent a certain state.

| Fault States in the Workstation | | Attributes (MIB-II e MIB-RMon) |
|---|---|---|
| E1 | high percentage of ICMP messages transmitted and received by the workstation | icmpInMsgs<br>icmpOutMsgs |
| E2 | high rate of utilization of the network by the workstation | hostInOctets<br>hostOutOctets |
| E3 | high rate of errors transmitted by the workstation | hostOutErrors |
| E4 | high rate of packages discarded in the workstation | hostInDiscards<br>hostOutDiscards |
| E5 | high rate of TCP segments | tcpInSegs<br>tcpOutSegs |
| E6 | high rate of retransmitted segments | tcpRetransSegs |
| E7 | high rate of segments with errors received | tcpInErrs |
| E8 | high rate of fault TCP connections | tcpAttemptFails |
| E9 | high rate of discarded datagramas | ipInDiscards<br>ipOutDiscards |
| E10 | excess of reassembly faults | ipReasmFails |
| E11 | high percentage of timeout messages | icmpOutTimeExcds |

Tabela 2 - Monitored attributes for observation of states in workstations

The network domain is very wide and its management can be focused under several aspects, such as [12] [16]:

- Operating systems
  - Resources
  - Network subsystems
- Communication interfaces
  - Ethernet
  - FDDI
  - Frame Relay
  - Others
- Communication Equipments
  - Hubs
  - Switches
  - Routers
  - PCs
  - Workstations
  - Modems
  - Printers
  - Others
- Network Services
  - DNS
  - Web
  - Mail
  - FTP
  - Others

Thus, a causal-models library (L) may be built to represent the network elements and its function in the network. Such library can be expanded and reused by the whole network community since the models can be built with standardized variables (attributes of managed objects). Other model types can be still built to be manipulated by different PSMs. In [1][4] it can be seen a PSM that diagnoses communication faults manipulating a network configuration model and network path model. The next section presents the proposed PSM that is able to diagnose faults manipulating fault-state models.

## 4 The PSM for the Network Management Domain

The method here proposed is supported by research made in the Benjamins' Library, in the assumptions established in section 2 and in the ontological knowledge shown in section 3. Figure 4 presents the task decomposition obtained to diagnostic task represented in a task-method structure. The control structures of the *symptom_detection*, *hypothesis_generation* and *hypothesis_discrimination* are respectively described below.

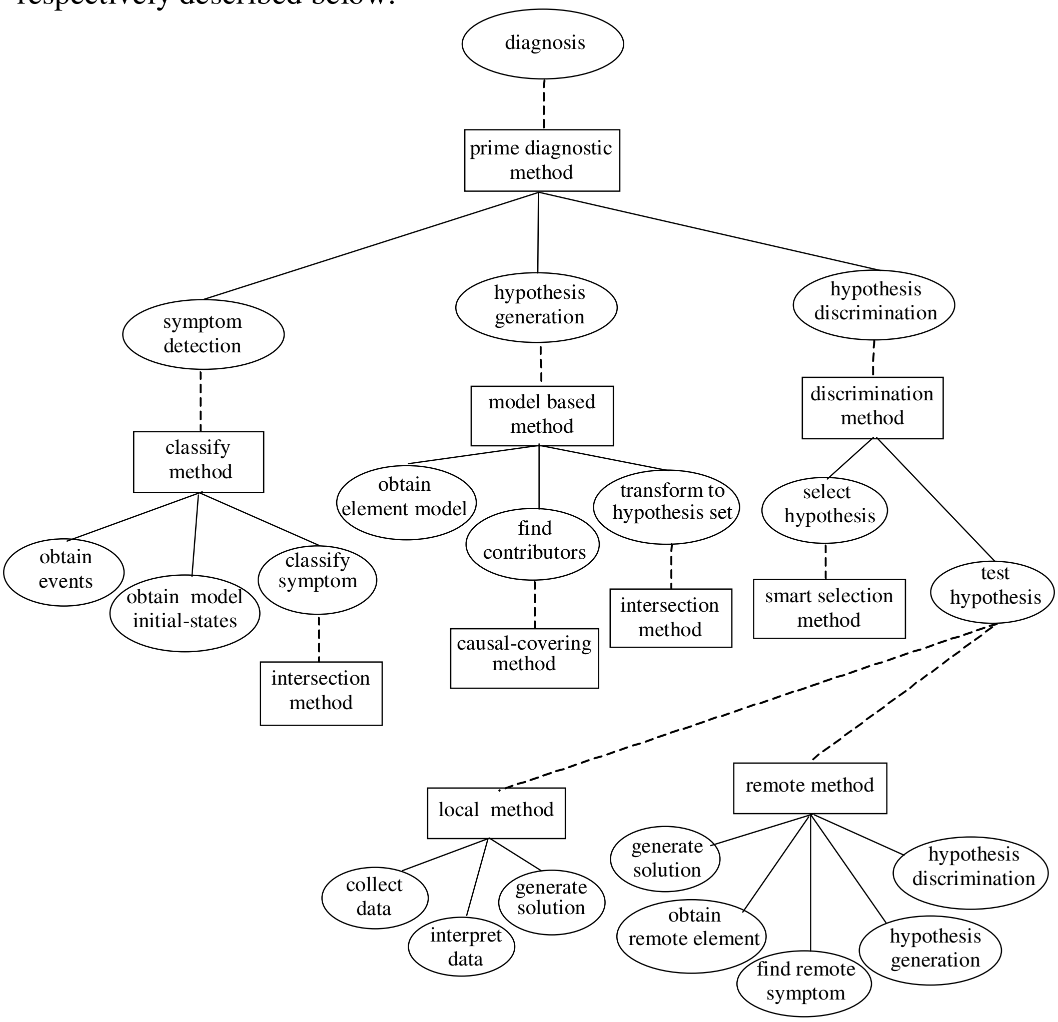


Figura 4 - The task-method decomposition structure. Tasks are represented by elipses and the method by a rectangule. Dashed lines indicate that methods are alternatives for achieving the task goal. Solid lines decompose a method into its sub-tasks.

The nomenclature used to describe the method follows the following pattern:

**task:** [task name]
**goal:** [goal to be reached by the task]
**input_data:** [input data]
**output_data:** [output data]
**internal_data:** [internal data manipulated for the task]
**algorithm:** [description of the task control]

**PSM:**

**task-1:** [***symptom_detection***]
**goal:** [detect occurring of initial fault-states in network element]
**input_data:** [event_log, ER: network element set, L: models library]
**output_data:** [S: symptoms set]
**internal_data:** [Eo: ocurring states set, Ei: initial states set, function, element_model]
**algorithm:**

```
symptom_detection (ER, event_log, L → S) =
        do
            for each network_element ∈ ER
            obtain_events (event_log, network_element → Eo)
            obtain_model_initial_state (L, network_element → Ei) =
                obtain_function (network_element → function)
                select_model (function, L → element_model)
                obtain_initial_states (element_model → Ei)
            classify_symptoms(Eo, Ei → S) =
                intersection (Eo, Ei → S)
        until  S != 0
```

**task-2:** [***hypothesis_generation***]
**goal:** [generate hypothesis that explain the symptoms set]
**input_data:** [S: symptoms set, L: models library, network_element]
**output_data:** [H: hypothesis set]
**internal_data:** [element_model, c: contributors set, C: set of contributors set]
**algorithm:**

```
hypothesis_generation (S, L, network_element → H) =
        obtain_element_model (network_element, L → element_model)
        for each symptom ∈ S
            find contributors (symptom, element_model → c) =
                    causal_covering (symptom, element_model → c)
            C ← c
        end
        transform_to_hypothesis_set (C → H) =
                intersection (C → H)
```

**task-3:** [***hypothesis_discrimination***]
**goal:** [find hypothesis set that explain symptoms and additional observations]
**input_data:** [H: hypothesis_set, network_element, topological_model, event_log]
**output_data:** [diagnosis_set: network states that explain the symptoms set]
**internal_data**:[E: occurred states set in the network_element, ERR: remote network_element set, hypothesis_selected, remote_symptom, element_model, data, H_remote: remote hypothesis set, remote_diagnosis_set]
**algorithm:**

*hypothesis_discrimination* (H, network_element, topological_model, event_log → diagnosis_set)**=**

```
do
    select_ hypothesis (H → hypothesis_selected)
    if type (hypothesis_selected , "local")
    then
        collect_data (network_element, hypothesis_selected, event_log→data) =
            obtain_events (network_element, event_log → E)
            intersection (E, hypothesis_selected → data)
        interpret_data (data, hypothesis_selected, H → H)
    if type (hypothesis_selected, "cause")
    then
        generate_solution (hypothesis_selected, H → diagnosis_set)
        return (diagnosis_set)
while ∃ hypothesis to test ∈ H ⊥ ¬ (type (hypothesis_selected, "remote"))

if type (hypothesis_selected, "remote")
then
    generate_solution (hypothesis_selected, H → diagnosis_set)
    obtain_remote_element (network_element, topological_model → ERR)
    for each remote_element ∈ ERR
        find_remote_symptom (remote_element, event_log, hypothesis_selected →
                                                        remote_symptom)
        if (∃ remote_symptom)
        then
            hypothesis_generation (remote_element, L, remote_symptom → H_remoto)
            hypothesis_discrimination (H_remote, network_element, topological_model,
                                            event_log → remote_diagnosis_set)
            diagnosis_set ← (diagnosis_set + remote_diagnosis_set)
    until ERR == 0
return (remote_diagnosis_set)
```

The execution below shows in short the reasoning method:

**step-1:** *symptom_detection*

Starting with a network element, the symptom_detection task verifies if there are incidents of events of this element based on the event_log. The function of the element, wich is the management aim, and its respective type of model are identified. The states of the model that are *initial* type and correspond to occured events are admitted as symptoms.

**step-2:** *hypothesis_generation*

Hypothesis are generated going throught the state model starting of the symptoms (states of the initial type). Hypothesis correspond to states of the *local*, *remote* or *cause* type that have a causal relation (direct or indirect) to the symptoms.

**step-3:** *hypothesis_discrimination*

Here, additional observations are obtained to test hypothesis. At this point the introduction of ontological knowledge in the system enables the dinamic selection of methods to carry out the test_hypothesis task. If the hypothesis is a *local* state, then it belongs to network element under analysis, therefore the *local method* is selected. If the hypothesis is a *remote* state, then it belongs to the other network elements. Therefore, the *remote method* is selected. The *remote method* consults the topological model of the network in order to identify the next network element to be investigated. The network elements where the *remote state* is detected are analysed. In this stage, the processing goes back to step-2 in order to generate new hypothesis. The diagnostic session finishes when a hypothesis correspond to a *cause* state or when there are no more elements to be investigated and the last hypothesis resulted in remote. In this last case, the primary cause is found outside of the managed domain.

## 4.1 Pre-requirements for validation of the proposed PSM

As mentioned in item 2, the implementation of a diagnosis system makes the assumption of the existence of input data supplied by other management applications. This paper does not aim at specifying such applications, but to supply some guidelines for such specification. Four basic applications are necessary to provide information for a diagnosis system that implements the proposed method: monitoring application; event generator; network topology management application and causal-model management application:

### Monitoring application

This application is responsible for periodically collecting and storing values of network resources (objects and attributes). The pre-requirements for its implementation are:

- Definition of the network elements to be managed;
- Definition of the function to be managed in the network element;
- Definition of the faults group that involves the function;
- Definition of the objects and attributes set that allows observing the function state.

**Event generator**
This application takes the objects and attributes values obtained by the monitoring application, it generates events (states faults or abnormal) and it stores them in a database (log). The pre-requirements for its implementation are:

- Output data of the monitoring application;
- Definition of expressions among managed resources (objects and attributes) capable to express fault-states;
- Definition of relationships between managed states and pre-defined faults (creation of causal models);
- Identification and study of the network typical behavior (normal behavior) to support definition of thresholds for generation of events.

**Network topology management application**
This application should allow the creation, modification and storage of the network topological model, generating a complete and consistent model to be consulted by the diagnostic system.

**Causal model management application**
This application should allow the creation, modification and storage of causal models of network elements generating complete and consistent models to be used by the diagnostic system.

## 5 CONCLUSIONS

In this paper we present an analysis of the network management domain that allowed us to select and adapt a tasks and methods subset from the Benjamins' Library, resulting in the PSM proposed for automation of the diagnosis task in this domain. Althought we have designed a system at such high abstration level, the so called "knowledge level", the work was accomplished for a real and complex domain. Differently of a toy domain the study of the network management environment requires a great effort and considerable amount of time. Moreover, the size and variety of information present in wide and complex network environments make it difficult to identify the relevance of the information for the task to be automated. Based on that, we have shown that a description at knowleledge level is a necessary step to be adressed before the implementation level to be reached.

The problem solving methods of the expertises of this domain are not always well understood or yet, they do not exist for particular problems. Therefore, it has not been possible to structure knowledge (control and domain knowledge) unless under the rule-based approach. In this sense, the support of the Benjamins'Library was very important for the configuration of the proposed method since it contains well known methods from literature, in a structured and reusable way, to build automatic diagnostic systems. Moreover, the suitability criteria of the library guided the knowledge acquisition task, resulting in the choice of model types and ontological description of the domain. However, it is important to point out that due to domain-specific requirements, tasks and PSMs from a library need to be adapted to suit the application. The network domain presents some features not covered by the Benjamins' Library: the distributed feature of the networks and the dynamic feature of the domain that involve issues about time of events, symptoms and faults. Such features are being current target of our research and details can be seen in [4]. A modification example introduced in the selected methods of the Library was the adaptation introduced in the *test_hypothesis* task. In this task we implemented an alternative method (called *remote_method*) besides the method suggested by the Library (called *local_method*) due to distributed feature of the networks and its management. In the adaptation processes of tasks and methods a knowledge engineer needs to make various sorts of decisions in order to reuse library components in a proper way. The Benjamins' Library supplied us valuable support for selection of methods, but not for adaptation processes. Other experiences related with the use of the Benjamins' Library can be seen in [10] [11] [17]. Another important feature to be considered in KBS projects for network management is the dynamic characteristic of the users requirements. They reflect changes in the network topology and services that running on the network may imply changes in network management applications. The description of the method at the knowledge level allows the manipulation of the domain elements in an independently way, p.e., changes in the network do not necessarily reflect changes in the method embedded in the system. In other words, if a new network element is inserted, all that is necessary is to represent it in the topological model and to make it available its causal model. We consider the development of this work as an initial step for the construction and reuse of ontologies, PSMs and domain models for automation of complex tasks in the network management domain. The efforts in this direction will result in the future, in a collection of reusable components to support the design of complex and sound network management systems.

## Acknowledgements

The research reported here was developed within the GSAD project (High Performance Systems Management Project). The partners in this project are Laboratory of Integrated Systems (University of São Paulo, Brazil) and ABC Bull S.A. Telematic.

## References

[1] Barros, L. N.; Lemos, M. A.; Bernal, V.; Wainer, J. ***Model Based Diagnosis for Network Communication Faults***. In Proceedings of the Third International Workshop on Artificial Intelligence for Distributed Information Networking (AiDIN'99), Editors: Steven Willmott and Sue Abu-Hakima. Workshop held at the AAAI'99 conference in Orlando, Florida, July 1999.

[2] Barros, L. N.; Hendler, J.; Benjamins, V. R. ***Par-KAP: a Knowledge Acquisition Tool for Building Practical Planning Systems***. In: IJCAI, 1997. *Proceedings*, 1997.

[3] Benjamins, V. R. ***Problem Solving Methods for Diagnosis***. Amsterdam, 1993. 172p. Thesis (PhD), University of Amsterdam.

[4] Bernal, V.; Barros, L. N.; Lemos, M. A.; Wainer, J. ***Fault Diagnosis for Local Area Network Environments***. Latin American Network Operations and Management Symposium (LANOMS'99). Rio de Janeiro, RJ, December, 03-04,1999.

[5] Bredeweg, Bert. ***Expertise in Qualitative Prediction of Behaviour***. Amsterdam, 1992. Thesis (PhD), University of Amsterdam.

[6] Breuker, J.; van de Velde, W. ***CommonKADS Library for Expertise Modelling Reusable problem solving components***. IOS Press, Amsterdam, 1994.

[7] Breuker, J. ***Components of Problem Solving and Types of Problems***. In: EKAW - European Knowledge Acquisition Workshop, 8th, Belgium, Sep 1994, *Proceedings*, p. 118-136. (Lecture Notes in Artificial Intelligence).

[8] Carvalho, T. C. M. B. ***Arquiteturas de Redes de Computadores OSI e TCP/IP***. BRISA, Makron Books do Brasil, 1994.

[9] Fensel, D.; Straatman, R. ***Problem Solving Methods: Making Assumptions for Efficiency Reasons***. In: EKAW'96 - European Knowledge Acquisition Workshop, 9th, Nottingham, United Kingdom, May, 1996. *Proceedings*, p. 17-32. (Lecture Notes in Artificial Intelligence, v. 1076).

[10] Orsvärn, K. ***Case-study in Knowledge Modelling with Benjamins' Library of Diagnosis Methods***. Proceedings of Knowledge Engineering Forum, GMD Sankt Augustin, Germany, March 1995.

[11] Orsvärn, K. ***Principles for Libraries of Task Decomposition Methods - Conclusions from a Case-study***. In Proceedings of European Knowledge Acquisition Workshop - EKAW'96, Lecture Notes in Artificial Intelligence 1076, pp. 48-65, Springer-Verlag, May 1996.

[12] Pras, A. ***Network Management Architectures***, Enschede, The Netherlands, 1995, series n. 95-02, 191p. Thesis (PhD). University of Twente.

[13] ***RFC 1155***. SMI – Structure and Identification of Management Information for TCP/IP based Internets, 1990.

[14] ***RFC 1157***. SNMP – A Simple Network Management Protocol, 1990.

[15] ***RFC 1213***. Management Information Base for Network Management of TCP/IP-based Internets. (MIBII). 1991.

[16] Rose, M. T.; McCloghrie, K. ***How to Manage Your Network Using SNMP: the networking management practicum***. Prentice Hall, New Jersey, 1995.

[17] Speel, P.; Aben, M. ***Applying a Library of Problem Solving Methods on a Real-Life Task***, In proceedings of KAW'96, (Banff), 1996.

[18] Valente, A**.** ***Legal Knowledge Engineering*****:** ***A modelling approach***. Amsterdam, 1995. Thesis (PhD), University of Amsterdam.

[19] Waldbusser, S. MIB RMON: Remote Network Monitoring Management Information Base. ***Request for Comments 1757***, Carnegie Mellon University. Feb, 1995. FTP anônimo em nic.ddn.mil, arquivo rfc/rfc1757.txt.

[20] Yemini, Y. ***Network Management: Problems, Technologies, Standards - The State of The Art***. In: SBRC - Simpósio Brasileiro de Redes de Computadores, 14º , Fortaleza, 20-23 maio, 1996. ***Anais***. Fortaleza, SBC, 1996.