Knowledge-Based Systems for Power System Control Centres: Validation and Verification

Zita A. Vale

University of Porto
Faculty of Engineering
Department of Electrical and Computer Engineering
Rua dos Bragas - 4099 Porto Codex
Portugal
Fax: 351-2-2041820 Phone: 351-2-2041820
Email: zav@fe.up.pt

Carlos Ramos
Jorge Santos

Polytechnic Institute of Porto
Institute of Engineering
Department of Computer Engineering
Rua de S. Tomé - 4200 Porto
Portugal
Fax: 351-2-821159 Phone: 351-2-8300922
Email: {csr|jsantos}@ dei.isep.ipp.pt

M. Fernanda Fernandes
Couto Rosado
Albino Marques

REN – Portuguese Transmission Network - (EDP Group)
Rua Cidade de Goa, 4 – 2685 Sacavém
Portugal
Fax: 351-1-9411318 Phone: 351-1-9411262

Abstract:

The activity of electrical network Control Centres is supported by a large number of computer applications. However, Control Centre operators usually have to face critical situations without the support of efficient decision support tools. During the last years, electrical utilities began to install intelligent applications in their Control Centres. These applications are mainly intended to provide operators with assistance especially during incident situations. The successful use of such applications is very dependent from the ability to guarantee that its performance is correct and efficient, even under incident conditions. This only can be guarantied after a Verification and Validation phase.

This paper addresses the Verification and Validation of Knowledge-Based Systems. The paper deals with the Verification and Validation of SPARSE, a Knowledge-Based System to assist operators of Portuguese Transmission Control Centres in incident analysis and power restoration. A specific tool (VERITAS) has been developed to verify SPARSE Knowledge Base.

VERITAS is a generic, domain independent KBS verification tool, it performs Knowledge Base structural analysis in order to detect knowledge anomalies. It has been successfully applied not only to SPARSE but also to other KBS.

Keywords: **Control Centres, Knowledge-Based Systems, Real time, Validation, Verification**

1 Introduction

Nowadays, Control Centres (CC) are of high importance for the operation of electrical networks. These Centres receive real-time information about the state of the network and Control Centre operators must take decisions according to this information.

Under incident conditions, a huge volume of information may arrive to these Centres, making its correct and efficient interpretation by a human operator almost impossible. In order to solve this problem, some years ago, electrical utilities began to install intelligent applications in their Control Centres. These applications are usually Knowledge-Based Systems and are mainly intended to provide operators with assistance, especially in critical situations.

The correct and efficient performance of such applications must be guarantied through Verification and Validation (V&V). V&V of Knowledge-Based Systems are not as usual as desirable and are usually undertaken in a non-systematic way.

The systematic use of formal V&V techniques is a key for making end-users more confident about Knowledge-Based Systems, especially when critical applications are considered.

This paper, addresses the Validation and Verification of Knowledge-Based Systems in general and in particular of SPARSE (a Knowledge-Based System to assist operators of Portuguese Transmission Control Centres in incident analysis and power restoration), namely the questions involved in V&V and the several methods used. The V&V SPARSE is analysed, being stressed the importance of applying formal verification methods.

Section 2 focus the main aspects related to acquisition, representation and maintenance of knowledge in KBS and its interaction with V&V stages.

Section 3 will describe the Validation and Verification stages of SPARSE development, specially the field tests and the need of applying formal methods in SPARSE V&V.

Section 4 presents VERITAS, a verification tool based in well known formal methods pointing the advantages and disadvantages of adopted techniques. This tool has been successful applied to different KBS: SPARSE; ARCA, an expert system applied to Cardiology diseases diagnosis; and another expert system created to assist in Otology diseases diagnosis and therapy. Finally, section 5 presents some conclusions.

2 Knowledge as the main problem of KBS

KBS are recognised as having many advantages when compared to traditional computer applications but their most appreciated characteristics are the following:

- KBS can include knowledge acquired from experts and use it in order to react to a situation as the expert would react
- KBS provide a clear separation between reasoning and knowledge, allowing to change the Knowledge Base without intervention in the inference mechanism.

Although these characteristics seem very appealing, it is interesting to consider the most commonly referred disadvantages of KBS:

1. Difficulties in knowledge acquisition - The knowledge acquisition phase is usually very time consuming. On the other hand, it poses some problems related with the communication between the expert(s) and the knowledge engineer. The participation of several experts is also appointed as problematic.
2. Difficulties in rule updating - When the Rule Base requires some changes, this process is considered rather complex. On the other hand, changes in the Rule Base may question the consistency of the new Rule Base.
3. Lack of methods to validate and verify KBS - As it is assumed that there are not commonly accepted methodologies to validate and verify KBS, informal (and usually incomplete) procedures are taken making difficult to guarantee the correctness and efficiency of the KBS. This problem is even more acute when a KBS is already deployed and is subjected to changes.
4. Difficulties in maintaining the knowledge - Even when the modification of rules is not necessary, the Fact Base requires maintenance in order to keep the KBS operational. As KBS usually require more information than the other applications, this maintenance is not usually automatic and operators tend to forget it.
5. KBS are less portable than other applications - As special languages and tools are used to develop KBS, portability is not guaranteed.
6. Real time performance - Some KBS have difficulties in providing real-time performance, especially in case of emergencies.

It is interesting to note that the two first presented disadvantages are directly related with the most appreciated characteristics of KBS. In fact, knowledge acquisition has always been one of the serious weak points of KBS and especially of Expert Systems. Although there exist some tools commercially available to support knowledge Acquisition, it does not seem useful to use a general tool for a specific application.

Establishing methodologies and developing support tools are important factors to ease the process. On the other hand, the establishment of good relations between the expert(s) and the knowledge engineer is of fundamental importance. When the experts are strongly involved, believe that the project is really important and trust the knowledge engineer, knowledge acquisition problems are considerably reduced.

The second disadvantage that is pointed out is in fact the negation of the second advantage presented. Is it really easy to include modifications in the Rule Base of a KBS or not? Changing rules poses no problems because a quasi-natural language is usually used allowing rules to be changed even by people not familiar with Artificial Intelligence.

What constitutes a real problem is:

- To acquire the knowledge required to undertake the changes. This aspect is in fact a knowledge acquisition problem and the use of machine learning techniques for suggesting changes in the Rule Base would be useful.
- To guarantee that the modified Rule Base remains consistent and that its use will not make the KBS incorrect or inefficient.

This last point leads us to the third disadvantage: problems of Validation and Verification (V&V). In fact, V&V is usually not considered as seriously as desirable in the case of KBS.

Although there seems to be difficult even to find in the specialised literature total agreement about the meaning of these terms, let us provide our personal definitions:

- Validation - Allows to assure that the KBS provides solutions that present a confidence level as high as the ones provided by the expert(s). Validation is then based on tests, desirably in the real environment and under real circumstances. During these tests, the KBS is considered as a "black box" and only the input and the output are really considered important.
- Verification - Allows to assure that the KBS has been correctly conceived and implemented and does not contain technical errors. Verification is intended to examine the interior of the KBS and find any possible errors.

For a great number of KBS, only validation is undertaken. Although this can guarantee, when the system is deployed, that its performance is correct, the existing problems may arise when there is a need to change the Rule Base.

Verification is more difficult because it relies on formal methods and requires the development of tools to implement these methods. Although there are already some available tools in the market, specific needs of Power System applications usually require the development of specific tools for this purpose.

As formal methods of verification rely on mathematical foundations, they are able to detect a large number of possible problems. In this way, it is possible to guarantee that a KBS that has passed trough a verification phase is correct and efficient. Moreover, it is possible to assure that it will provide correct performance with examples that have not been considered in the validation phase.

The next disadvantage pointed out refers to the difficulties encountered in the maintenance of the Knowledge Base. In fact, the maintenance of the Rule Base has already been considered but the maintenance of the Fact Base also puts some problems.

Some pointed disadvantages of KBS are really difficult to solve as the problem in knowledge acquisition. However, most of the problems can be completely solved or, at least, become less serious if all the required tools are provided. Figure 1 shows what we consider the complete architecture of a KBS for a Control Centre.

This architecture includes, besides the modules that are always included in a KBS, the following modules:

- Knowledge acquisition assistant

- Knowledge update assistant
- V&V assistant.

The inclusion of these modules allows to provide the users of the KBS with assistance to solve the most commonly pointed problems.

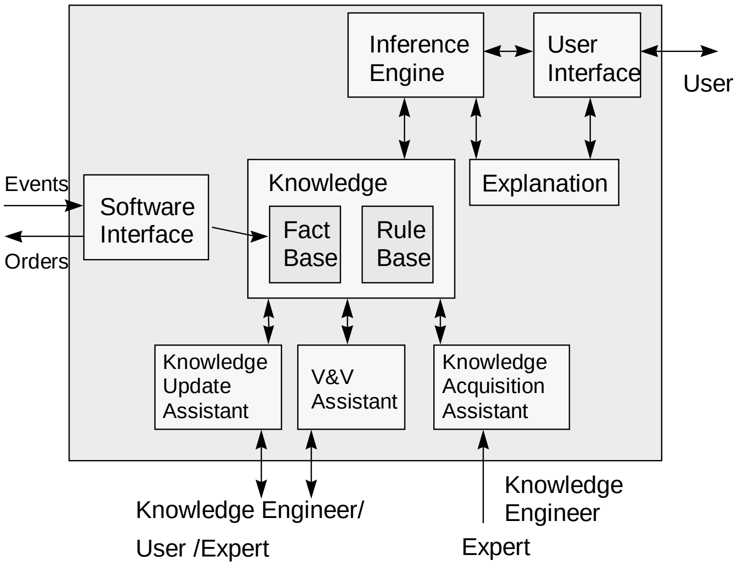

Figure 1 - Architecture of a KBS

In fact, several complaints about the KBS are due to the use of an incomplete KBS, lacking one or more of these modules. It is not fair to consider an incomplete system and judge it as a complete one.

3 SPARSE

SPARSE is a KBS developed for the Control Centres of the Portuguese Transmission network, owned and operated by REN. This KBS assists Control Centre operator in incident analysis and power restoration [Vale-93][Vale-94][Vale-96a][Vale-96b].

SPARSE has been developed using PROLOG and C language and is installed in a DECstation 5000/240 under ULTRIX operating system. This machine is connected, trough a Local Area Network Ethernet of duplicate configuration with the two MicroVAX II machines that support SCADA functions in the Control Centre.

SPARSE has passed trough a validation phase and is presently installed in one of the two Control Centres of REN - Vermoim Control Centre, providing real-time assistance to operators.

This section presents some interesting aspects of SPARSE project, especially important for the success of the final application.

3.1 Validation

The process of Verification and Validation should start as early as possible during the development of the application. In the case of SPARSE, V&V has been considered since the very beginning and special arrangements have been made in order to provide conditions for it.

The project team aimed to perform the validation of SPARSE using examples as close as possible to the ones that the application should face in the real environment. According to this, it was considered that validation should be based mainly on real information about the network.

Another important aspect that has been considered since an early stage of development is the software required to interface SPARSE with SCADA applications used in the Control Centre. In fact, it was realised that some limitations imposed by SCADA should be considered since the very beginning in order to allow to take them into account during the development of the prototype, namely during the knowledge acquisition phase.

When integration issues are not addressed in an early phase of the project, the changes that are required when the system is integrated in the real environment may be very significant and impose almost a complete rebuilding of the system. These issues should, namely, be considered during the knowledge acquisition phase by the experts.

The staff of REN has developed an application named TTLOGW [Rosado-93] to acquire real-time information from SCADA and to send it to SPARSE. This application has been installed in the SCADA machines (microVAX II) and has been used in order to obtain material to validate SPARSE.

This application acquires the information related to the state of the equipment of the electrical network. This information includes, namely:

- state of breakers and disconnectors (open, closed, changing);
- mode of operation of substations (local or remote and automatic or manual).

When an incident takes place, TTLOGW enter in “incident” mode and sends to SPARSE not only this kind of information but also all the alarms that are generated by the SCADA applications. In the early stage of the project, this application was used to create a file containing this information. As SPARSE was not installed in the Control Centre, the detection of the end of the incident had to be done manually. Control Centre operators were in charged of changing the mode of operation of TTLOGW back to “normal” as soon as the incident in the electrical network was cleared.

In this way, files concerning real incidents have been obtained and have been used in order to validate SPARSE conclusions. Experts involved in the project commented these conclusions and corrections in the Knowledge Base were made whenever necessary.

When SPARSE was installed in the Control Centre, receiving real-time information sent by TTLOGW, new techniques of validation had to be applied. The validation of SPARSE considering real-time information was very important due to several reasons:

- Temporal reasoning should be tested under real situations in order to assure its correction
- Consideration of multiple faults is an important aspect of SPARSE performance that is very dependent from the way in what information flows
- Processing times should be tested in order to guarantee real-time performance, even under incident conditions.

As nowadays electrical networks are very reliable it was not possible to completely validate SPARSE with real incidents. A large number of different types of incidents should be simulated to allow validation. As this simulation should be as accurate as possible, two different techniques have been used:

- Simulation of incidents by operators located in chosen substations
- Simulation of incidents using a programmable impulse generator and a Remote Terminal Unit (RTU).

Each of these two techniques complements the other one, allowing a complete validation.

The simulation of incidents by operators allowed to obtain real-time information that was forced to be generated but presenting exactly the same characteristics as the information obtained during a real incident. During these tests, the behaviour of the protection equipment was simulated by operators, making the whole system act as if a real incident was taking place. In this way, the information, used by SPARSE was generated as it would be under a real incident.

Due to the difficulties of co-ordinating operators in several substations, the simulation is not always correct and the whole process may have to be repeated several times in order to obtain a good test.

In spite of all the difficulties and costs involved, this kind of tests has been considered absolutely essential for the validation of SPARSE, allowing to increase the confidence in its real-time behaviour.

In order to undertake a complete set of tests without the extremely high costs required by this technique, a different technique of test has also been used. This technique involves the use of a Remote Terminal Unit (RTU) and of a programmable impulse generator (PIG). The alarm messages generated by the SCADA are simulated by the impulses generated by the PIG.

This technique of test may be used in a chosen substation, being used the most convenient one, namely a substation close to the Control Centre. In this way, the operator that is manipulating the PIG is not far away from the team involved in the analysis of SPARSE results and may participate in co-ordination meetings between two consecutive tests.

This technique has allowed to simulate a wide set of incidents in order to validate SPARSE Knowledge Base as completely as possible.

In order to make validation tests faster, a time simulator has been developed. This simulator allows to use incident files saved by SPARSE and replay those situations off-line. In this way, after carrying on a set of tests, the files that are generated may be used by the simulator to provide SPARSE with the corresponding information.

These methods of validation have been considered sufficient to put SPARSE in service, without the need to undertake formal verification of SPARSE Knowledge Base.

When a Knowledge-Based System, as SPARSE, is in continuous use, the necessity to make changes in the Rule Base arises sooner or later. In the case of the Portuguese Transmission network, the introduction of new substations, with different types of operation or layout, has already imposed some modifications.

Under these circumstances, it is not possible to accept the need to undertake complete validation tests, as the ones described before. Even if the costs are acceptable, the required time would oblige the Knowledge-Based System to be either out of service or in service without a validated Rule Base for more time than desirable.

This problem must be addressed through the use of a verification tool using formal methods. The use of these tools, to detect possible problems in the Rule Base, after making changes, allows reducing the time required to V&V the new Rule Base.

3.3. Verification

There were two major reasons to start SPARSE verification work. First, the SPARSE team carried out a set of tests (see Validation) in order to assure the quality of the answers of SPARSE to a set of real and/or simulated cases. Although, considering the expected reliability and confidence of the tools to be applied in power systems area, it was decided to develop a verification tool to perform anomalies detection in SPARSE KB assuring the consistency of the represented knowledge. On the other hand, the tests applied in Validation phase, namely field tests, are very expensive because during their execution it was necessary to allocate a lot of technical personnel and physical resources, like lines and substations. It seems obvious that it is impossible to carry out those tests after each knowledge updating so the developed verification tool offers an easy and inexpensive way to assure the knowledge quality maintenance.

Due to SPARSE verification work, a tool (VERITAS) [Santos-97][Vale-97] was developed allowing to perform the structural analysis in order to detect knowledge anomalies.

4 VERITAS, a verification tool

Structural analysis technique was adopted as main technique for anomaly detection after considering several aspects, where the main advantages are:

- Anomaly detection is one of most used in the development of Verification tools and the one that receives greater acceptance in V&V community;

- A pragmatic approach allows to obtain practical results in a short time, bringing as immediately consequence a solidification of V&V knowledge, essential to development of more complex and adequate V&V tools;
- The referred technique is quite adjusted to verify KBS with monotonic characteristics, like ARCA system and ES for Otology diseases diagnosis and therapy.

While the main disadvantage is the anomaly detection isn't the most adequate technique to perform the verification of nonmonotonic KBS. First, because there is a set of anomalies specifically related with this type of systems [Antoniou-97] which detection was not yet addressed. Second, VERITAS will report some (false) anomalies that corresponds to environments that not exists due to knowledge represented in meta-rules used to rule selection mechanism.

VERITAS system is independent of knowledge domain and of the grammar used to represent the rules.

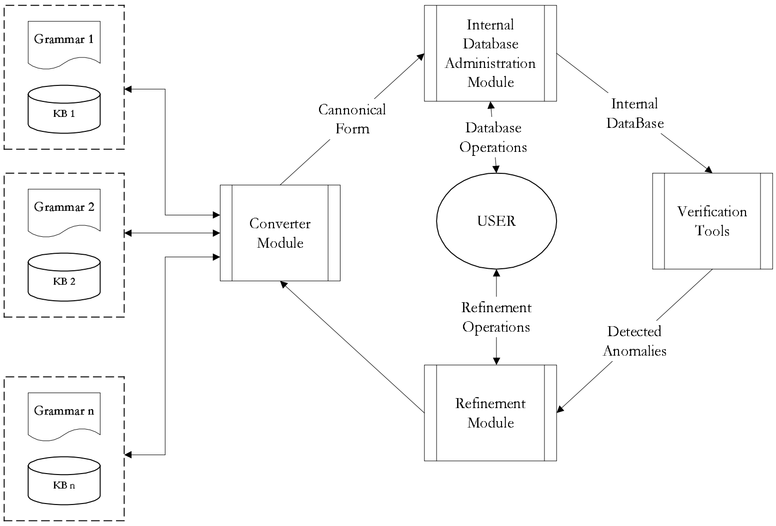

Figure 2 – VERITAS Architecture

VERITAS has been developed with an open and modular architecture (Figure 1) allowing interacting with the user along all the verification process. Since the tool is independent of KB grammar, in principle any rule base KB can be analysed by VERITAS.

The Converter module allows to represent the external rules in an internal canonical form that is recognised by the other modules. Notice that this module works in two directions. It can also convert the external KB into canonical form and allows to generate new rules during knowledge updating, after anomalies detection, using an external grammar.

The Internal DB Administration module is responsible for the extraction and classification of all the information needed the during anomalies detection phase from the internal rules (previous obtained). In the first step all literals extracted from rules are classified according to the following classification:

- Fact – if it just appears in rule antecedent (LHS);
- Conclusion – if it just appears in rule consequent (RHS);
- Hypotheses – if it appears in both sides of the rules.

Notice that this classification is domain independent and just makes sense for verification procedures. The described classification main advantages are a more compact knowledge representation and the reduction of the complexity of the rule expansion generating process. As it will be described later, this process corresponds to the analytical calculation of all possible inference chains.

In the second step, the Internal DB Administration module generates useful information about existing relations between literals (previously obtained). That information will be used not just to make the expansions generating process faster but also in the automatic detection of Single Value Constraints. Notice that VERITAS considers some type of constraints already described in literature [Zlatareva-94]. Considered constraints could be divided in the following way:

- Semantic Constraints – this type of impermissible set is formed by literals that cannot be present at the same time in KB. It has to be introduced by user.
- Logical Constraints – there are just two logical constraints:
  - A and not(A) (A means an literal)
  - A and notPhysical(A); this designation is obtained by analogy with logical negation and allows to represent the constraint defined by a literal and its retraction from KB. This operation is very common in non-monotonic systems like SPARSE.
- Single Value Constraints – this type of impermissible set is formed by just one literal but considering different values of its parameters. Notice that those potential constraints are automatically detected. After this the constraint can be either confirmed or changed by the user.

The anomalies detection module (Verification Tools) works in an autonomous way with no user interaction. It allows to be run in batch mode.

The anomalies detection basically relies in the rules and constraint analysis. This method has already been used by some well known V&V tools, as COVER [Preece-90] and KB-REDUCER [Ginsberg-87]. The rule expansion generation can be done in two different modes: normal or exhaustive.

As example, let us consider the following KB:

```
r1: t(X) and r(a) → s(a)
r2: f(a) → t(a)
r3: f(b) → t(b)
r4: h(a) → r(a)
r5: j(a) → r(a)
```

Table 1 shows the expansions that will be generated in each mode:

Table 1

Normal Mode	Exhaustive Mode
`t(X) and h(a) → s(a)`	`t(a) and h(a) → s(a)`
`t(X) and j(a) → s(a)`	`t(a) and j(a) → s(a)`
	`t(b) and h(a) → s(a)`
	`t(b) and j(a) → s(a)`

It is possible to notice that normal mode generates fewer expansions but, on the other hand, the information obtained after anomalies detection is more useful. The exhaustive mode wastes a lot of time generating the rules expansions implying also more wasted time to analyse them, but in principle it will be possible to detect more potential errors.

The detected anomalies could be grouped in three major classes: redundancy, circularity and inconsistency, according to Figure 3.

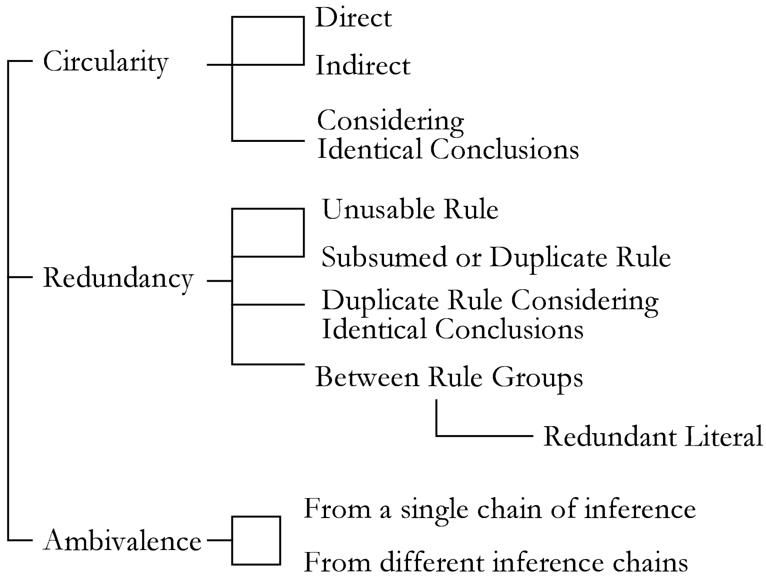

Figure 3 – Detected Anomalies

Notice that used classification is similar to Preece classification [Preece-94] with some differences. First, the matching values are considered in rule analysis, it means that a new set of anomalies will arise. Let us consider the following circular rules:

```
r1: t(a) and r(X) → s(a)
r2: s(a) → r(a)
```

for `X=a` some inference engines could start an infinite loop.

Another situation concerns to redundancy between groups of rules. In the following example:

```
r1: a and b and c → z
r2: not a and c → z
r3: not b and c → z
```

the rules `r1`, `r2` and `r3` could be replaced by the next rule:

```
rx: a and b and c or not a and c or not b and c → z
```

Applying logical simplifications to rule `rx`, it is possible to obtain the following rule:

```
rx': c → z
```

The described situation can represent an undesirable redundancy and it is a generalisation of the unused literal situation already studied by Preece [Preece-94]. Notice that this type of redundancy could be desirable. Rule bases are drawn as a result of a knowledge analysis/elicitation process, including, for example, interviews with experts or the study of documents such as codes of practice and legal texts, or analysis of typical sample cases. The rulebase should reflect the nature of this process, meaning that if documentary sources are used the rule base should reflect knowledge sources. VERITAS can detect these situations using an improved Quine-McCluskey method for logical expressions simplification.

5 Conclusions

This paper dealt with some important aspects for the practical use of KBS in Control Centres. These aspects include knowledge maintenance, portability, Verification, and Validation.

The systematic use of Verification and Validation methods is very important for the acceptance of Knowledge-Based Systems by their end-users. The use of Verification tools, based on formal methods, increases the confidence of the users and eases the process of introducing changes in the Rule Base.

This paper described the SPARSE V&V process. SPARSE is a real-time knowledge-based system to assist operators of Portuguese Transmission Control Centres.

For the verification of SPARSE we decided to implement a tool using a formal verification method. This tool, being based on mathematical foundations, assures the detection of abnormal situations in the KB.

The method used for the verification of SPARSE KB lays in the fact that an abnormal situation is, most of times, related to conception and development errors.

As SPARSE KB presents features that are not usually considered in verification methods, we have considered an extension of CTMS-theory. These features include the consideration of disjunction in LHS, multiple conclusions, test of conditions, temporal operators and meta-rules.

Presently, the use of machine learning techniques is being considered. The idea is to use a measure to the success in the case of failure of rules, allowing suggesting modifications in the Rule Base.

6 References

[Antoniou-97] G.Antoniou – "Verification and Correctness Issues for Nonmonotonic Knowledge Bases", International Journal of Intelligent Systems, Vol.12, N°10, pp.725-738, 1997

[Cigre-93] CIGRE - Working Group 38.06.03, "Expert Systems: Development Experience and User Requirements", Electra, n. 146, pp. 29-67, February 1993

[Durkin-96] J. Durkin, "Expert Systems: A View of the Field", IEEE Expert, February 1996

[Ginsberg-87] A.Ginsberg – "A new aproach to checking knowledge bases for inconsistency and redundancy", In procedings 3rd Annual Expert Systems in Government Conference, IEEE Computer Society, Washington, D.C., pp.102-111, October, 1987

[Preece-90] A.Preece – "Towards a methodology for evaluating expert systems", Expert Systems (UK), Vol.7, N°4, pp.215-223, November, 1990

[Preece-94] A.Preece and R.Shinghal – "Foundation and Application of Knowledge Base Verification", Intelligence Systems, N°9, pp.683-701, 1994

[Rosado-93] C.Rosado, "Process TTLOGW", EDP Report RESP/SCDS 20/93, July 1993

[Santos-97] J.Santos –Verificação e Validação de Sistemas Baseados em Conhecimento – VERITAS, uma Ferramenta de Verificação", MSc Thesis, Faculdade de Engenharia do Porto, December 1997

[Vale-93] Z.Vale and A.Moura, "An Expert System with Temporal Reasoning for Alarm Processing in Power System Control Centers", IEEE Transactions on Power Systems, vol. 8, No. 3, pp. 1307-1314, August 1993

[Vale-94] Z.Vale, A..Moura, M.Fernandes and A.Marques, "SPARSE - An Expert System for Alarm Processing and Operator Assistance in Substations Control Centers", Applied Computing Review, vol. 2, no2, pp. 18-26, ACM (Association for Computing Machinery) Press, December, 1994

[Vale-96a] Z.Vale, L.Faria, C.Ramos, M.Fernandes and A.Marques, "Towards More Intelligent and Adaptive User Interfaces for Control Center Applications", International Conference on Intelligent Systems Applications to Power Systems (ISAP'96), Orlando, Florida, U.S.A., pp. 2-6, 28 January - 2 February, 1996

[Vale-96b] Z.Vale, M.Fernandes, C.Rosado, A.Marques, C.Ramos and L.Faria, "Better KBS for Real-time Applications in Power System Control Centres: What can be learned by experience?", First International Conference on Successes and Failures of Knowledge-Based Systems in Real-World Applications, Bangkok, Thailand, 28-30 October, 1996

[Vale-97] Z.Vale, J.Santos, C.Ramos - "SPARSE - A Prolog Based Application for the Portuguese Transmission Network: Verification and Validation", PAP'97 - Practical Application in Prolog, Londres, Inglaterra, April, 1997

[Wollenberg-86] B.Wollenberg, "Feasibility Study for an Energy Management System Intelligent Alarm Processor", IEEE Transactions on Power Systems, vol.1, n.2, pp. 241-247, May 1986

[Zlatareva-94] N.Zlatareva and A.Preece – "An Effective Logical Framework for Knowledge-Based Systems Verification", International Journal of Expert Systems, Vol.7, N°3, pp.239-260, 1994