OPERATIONALISATION OF INDUSTRIAL GAS TURBINE ONTOLOGY

[1] F.Z. LAALLAM & [2] M. SELLAMI
[1] Department of math and data processing, Kasdi Merbah University, Ouargla, Algeria.
Tel.: 078.53.87.15. Email: laallam_fz@yahoo.fr

[2] Department of data processing, LRI, Badji Mokhtar University, Annaba, Algeria
Tel.: 038.87.29.91. Email: sellami@lri-annaba.net

ABSTRACT:

The activity of industrial supervision and control of the gas turbines is a very complex spot and requires a great experiment. This experiment is acquired with the passing of years what makes departure of an expert in retirement a great loss of the know-how. In the same way, these experts are not available constantly and in each site. The problem thus consists in capturing this know-how and allows experiment to be cumulate with an aim of the construction of a system of assistance to the diagnosis.

The work presented in this article relates to ontological engineering and more particularly the use of ontologies in the Knowledge-based systems*. After the stages of conceptualization and ontologisation [6], which allowed the capture of knowledge of the domain; we proceed in this article to the Operationalisation of conceived ontology. The modeling of the inference is based on the specification of the operational objective by scenario of use. The JessTab tool is used for the integration of the contexts of use in JESS. Thus we have obtained our system of assistance to the diagnosis "OntoTurb-Expert" of which the goal is not to solve the problem of diagnosis automatically, but to help the user by providing him suitable information and by leaving him the responsibility for a contextual evaluation of this information*

Key words: *Operationalisation, ontology, industrial gas turbine, scenario of use, diagnosis of breakdowns, JessTab.*

1. INTRODUCTION:

In IA, ontology is the specification of the objects, the concepts, the classes, the functions and the relations of a domain independently of a particular application like the semantic networks and the conceptual graphs. They are used by people, data bases and applications needing to share information relating to a domain. Ontology is thus the support of the acquisition of knowledge and it is also a useful tool to interface the software agents and the human agents. Many works were already undertaken concerning the construction of ontologies, but their practical use within the Knowledge-based systems was approached still little [3]. We try in this article to present a contribution concerning the Operationalisation of ontologies, starting from the axioms, concerning an industrial gas turbine, quoted in [6]. In [6], we have built manually industrial gas turbine ontology using middle out approach. After the stages of conceptualization (figure1) and ontologisation (figure 8), we obtained a formal ontology, edited in Protégée2000 [7], but not operational because it is not equipped with the structures of reasoning. The work presented in this article consists in integrating the ontology conceived in a Knowledge-based system of assistance to the diagnosis.

2. DETECTION OF A BREAKDOWN

The origin of a breakdown is detected by monitoring. Measuring instruments (thermocouples, thermostats…) are attached to the various components of the system allow measuring various parameters (temperature, pressure…). Following the detection of a breakdown, one or more alarms are announced. These detections make it possible to inform and attract the attention of the agents in charge of the monitoring of the industrial systems so that they can prevent the dysfunctions. Thus, the breakdowns are detected before their effects are propagated in the system and aggravate the situation. We note that our ontology covers all types of breakdowns.

3. THE DIAGNOSIS OF BREAKDOWNS:

In general, the contents of an alarm are not enough to identify a breakdown or to make a decision concerning the spots of repair because the breakdown is not inevitably in the component in which alarm was detected. Even if the breakdown is in the component where alarm was announced, its identification can be impossible without the observation of the symptoms on the neighbors components. The diagnosis of breakdowns become then a difficult spot and becomes a very important challenge. On the one hand, the process of the diagnosis must be able to consult the experience captured to facilitate the localization of the problem and on the other hand to deduce the propagations of underlying breakdowns.

4. CONSTRUCTION OF ONTOLOGIES :

Ontology consists of concepts, relations and the axioms. The concepts represent the primitives of the domain, the relations make it possible to bind these primitives and the axioms are assertions accepted like true in the considered domain.
To build an operational ontology, three stages are necessary:

- **Conceptualization:** consist in identifying the knowledge contained in a corpus (expressed in natural language and integrating all knowledge or raw data of the domain that we wish to formalize) representative of the domain;
- **Ontologisation:** consist in formalizing, as much as possible, the conceptual model obtained at the preceding stage.
- **Operationalisation:** consist with the transcription of ontology in a formal and operational language of representation of knowledge.

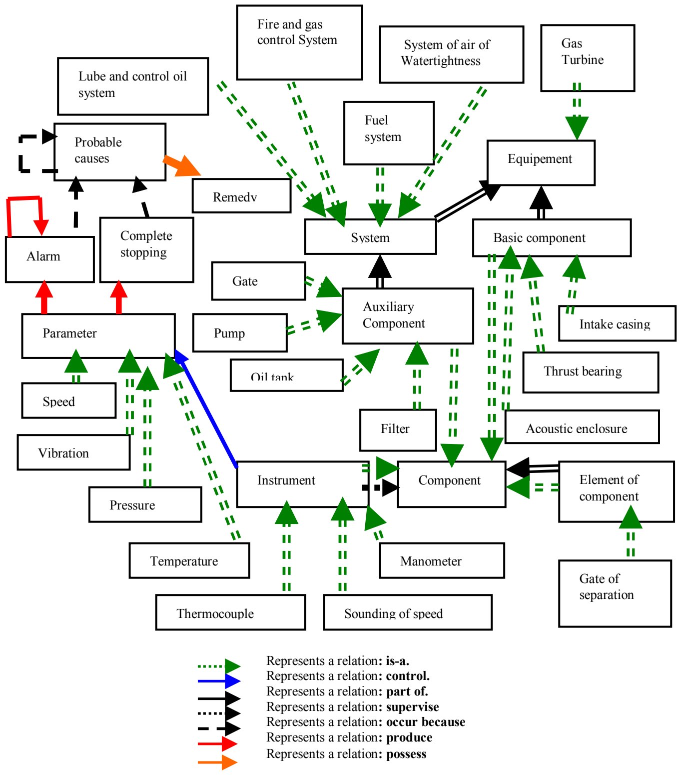


Figure 1: An extract of our ontology representation model [6]

5. THE OPERATIONALISATION :

To integrate ontology within a Knowledge-based system, it is advisable to specify the semantics of manipulation of the axioms, which is related to the application considered (i.e. with a well defined operational use). Indeed, the representation of terminological knowledge of the domain does not depend on the multiple possible applicative contexts. At the terminological level, the representation of a concept or a relation is made by a term. This representation does not vary according to the operational objective of the system. Only the representations of the semantics of the domain must be adapted to the objective of the considered application [4].

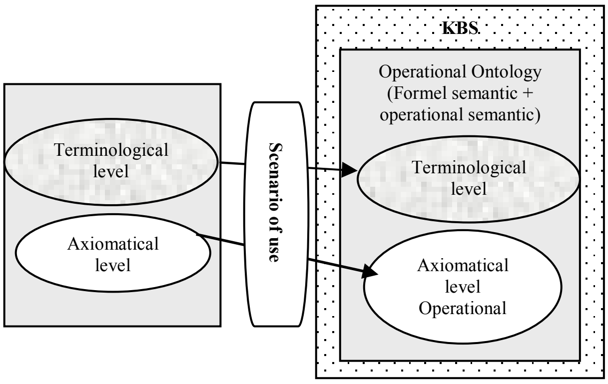


Figure 2: Operationalisation of an ontology

5.1- THE AXIOMS:

The following sentences represent axioms listed for the Operationalisation of our ontology (figure1) [6]:

1. An **instrument** is a **component.**
2. A **gas turbine** is a **equipment**
3. If a **parameter** (temperature, pressure, vibration…) given by an **instrument** assembled on a **component** is lower or higher than a normal value then an **alarm** is announced or a **complete stopping** of the **equipment** is carried out.
4. A **system** belongs to **equipment.**
5. An **auxiliary component** is a **component** belonging to a **system.**
6. A **basic component** is a **component** belonging to the **equipment.**
7. An **element of component** is a **component** belonging to a **component.**
8. Each **instrument** monitors a determined **component.**
9. Each **instrument** gives values of the **parameter** of control (temperature, pressure, vibration…) on the monitored component.
10. An **Alarm** can be produced because of other **Alarm.**
11. Each **Alarm** is produced because of one or several **probable causes**.
12. Each **complete stop** is produced because of one or several **probable causes.**
13. Each **probable cause** has one or more **remedies.**
14. A **probable cause** can be produced because of others **probable causes**.

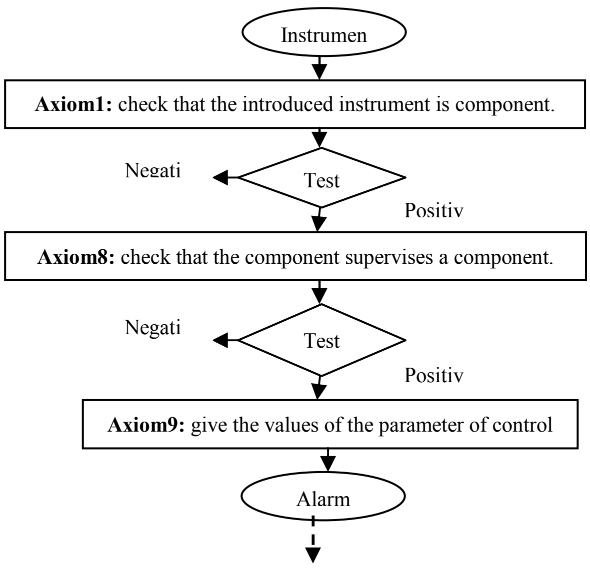
5.2- SCENARIO OF USE AND THE OPERATIONALISATION OF THE AXIOMS:

The Operationalisation of ontology is conceived only for one well defined operational use, thus based on the specification of a scenario of use which will determine the operational form of ontology i.e. the choice, for each axiom, of a context of use which specifies to what will serve the axiom and how it will be implemented. Thus, an axiom can be used to infer new knowledge or to validate the adequacy of a knowledge compared to the semantics of the considered domain. For example, the axiom "Each Alarm is produced because of one or several probable causes." can be used for infer the causes of a given alarm.

The various contexts of use are:

- The **inferential and explicit** context of use where the user starts the application of the axiom on a basis of facts to produce new assertions;
- The **inferential and implicit** context of use where the axiom is applied by the system on a basis of facts to produce new assertions;
- The context of use of **explicit validation** where the application of the axiom is started by the user to control the semantic conformity of the facts of a base compared to the domain;
- The context of use of **implicit validation** where the axiom is applied by the system to control the semantic conformity of the facts of a base compared to the domain [4].

A scenario of use thus consists of a set of selected contexts of use for each axiom present in ontology. It thus describes at which ends the knowledge specified in ontology will be used, i.e. to what will serve the axioms of ontology.



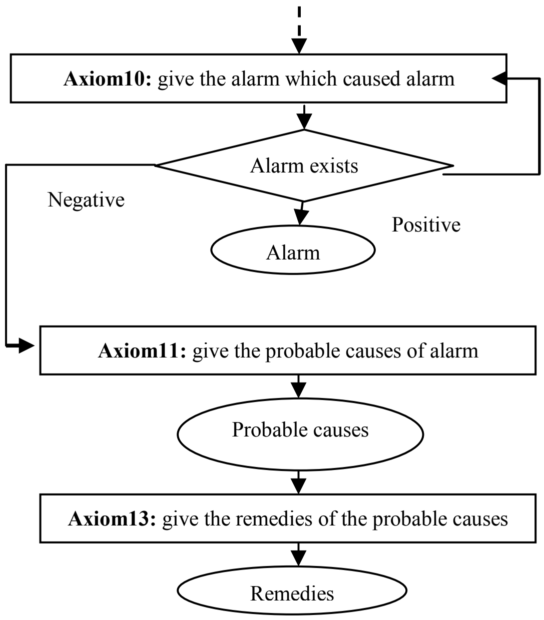


Figure 3 : the scénario of use

By detecting an alarm announced in the monitoring system, the user consults the system of diagnosis by introducing the instrument announcing alarm.
After a succession of inférentiel and implicit contexts of uses, the system gives the causes of announced alarm and the corresponding remedies.

5.3- THE PROCESS OF OPERATIONALISATION:

Before proceeding to the Operationalisation of ontology, the choice of an operational language is necessary.
The operational language to choose must allow:

1. to represent the concepts, the relations and the axioms,
2. to manipulate represented knowledge (reasoning),
3. the use of the same paradigm as that used to structure ontology to avoid the transcription from a paradigm to another in the process of Operationalisation,
4. to manipulate a representation of the axioms of the domain in agreement with the scenario of use (adaptation to the domain of knowledge).

Each language offers a set of diagrams of preset axioms, such as subsumption between conceptual primitives. Choose a language offering a panel of diagrams of axioms, which cover as much as possible of axioms of the domain, facilitates the Operationalisation.

The process of Operationalisation adopted is represented in the following figure:

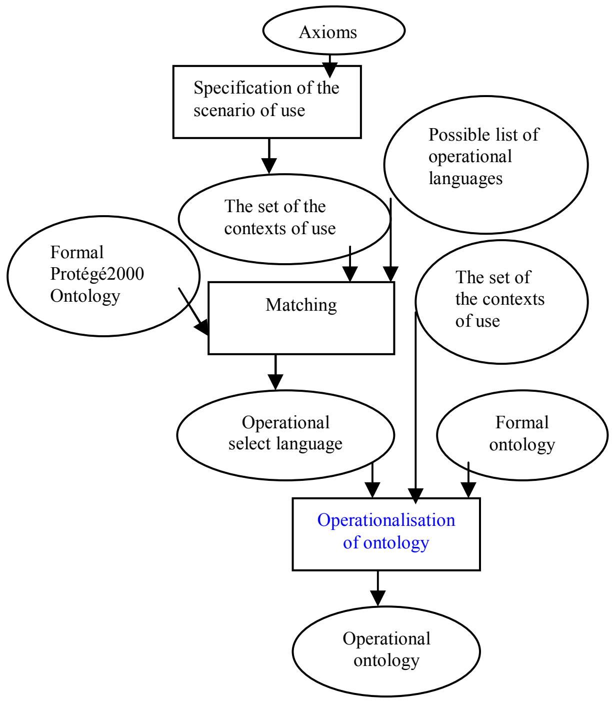


Figure 4: process of Operationalisation

The Protégé2000 [7] editor does not allow making reasoning on conceived ontology. Another tool called Jess was coupled with protégé2000 to give a new tool called JessTab. Thus we have obtained an efficient tool which allows the development of our system of diagnosis.

5.4- REPRESENTATION OF THE AXIOMS IN LOGIC:

For the ontological representations of the axioms, we use first order logic because it is the formalism used in Jess. Axioms 1, 8, 9, 10, 11 and 13 (figure 3) can be translated into logic as follows:
Axiom 1 : ∀ I, Instrument (I) => Composant (I).
Axiom 8 : ∀ I, ∃ J, Instrument (I), Composant (J) => surveille (I, J).
Axiom 9 : ∀ I, ∃ P, Instrument (I), Paramètre (P) => contrôle (I, P).
Axiom 10 : ∃ A, ∃ B, Alarme (A), Alarme (B) => produire-à-cause (A, B).
Axiom 11 : ∀ A, Alarme (A), ∃ C, cause-probables (C) => produire-à-cause (A,C)
Axiom 13 : ∀ C, cause-probables (C), ∃ R, remède (R) => possèder (C,R).

5.5- CONSTRUCTION OF THE BASE OF THE FACTS:

The construction of the base of the facts is carried out by transferring the ontology represented in protégé2000 [6] in a representation JESS [5].

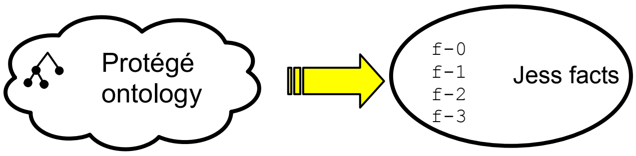


Figure 5: transfer of ontology from Protégé2000 representation in JESS representation [Eri03]

To transform the instances of a class to facts JESS, the function (mapclass) is used. Example :(mapclass component) transforms the instances of the class component to facts JESS. To transform all the instances of the hierarchy at the same time, we use the function (mapclass) and we specify like parameter the root THING (mapclass: THING).

OntoTurb ontology was already defines in protégé2000 (figure 8), it is then enough to type the order "(mapclasse: THING)" to transform all the instances of the hierarchy into facts JESS.

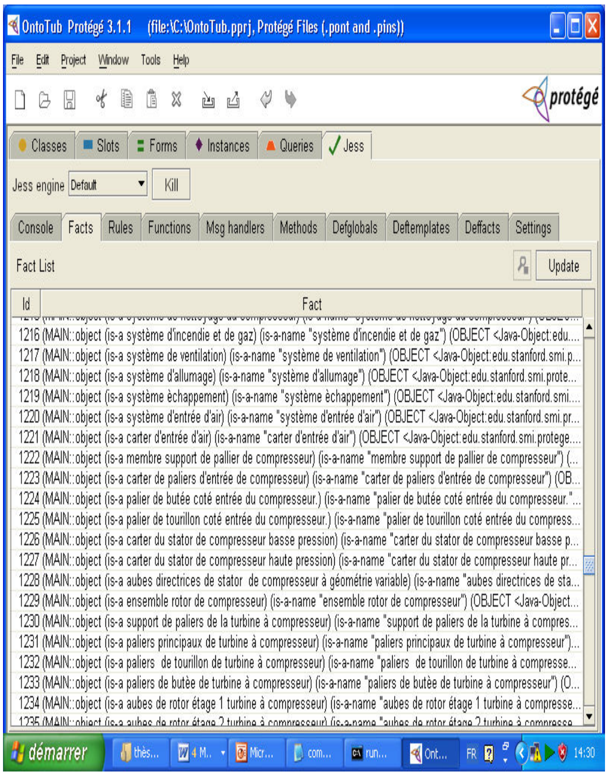


Figure 6: the base of the facts de « OntoTurb-expert »

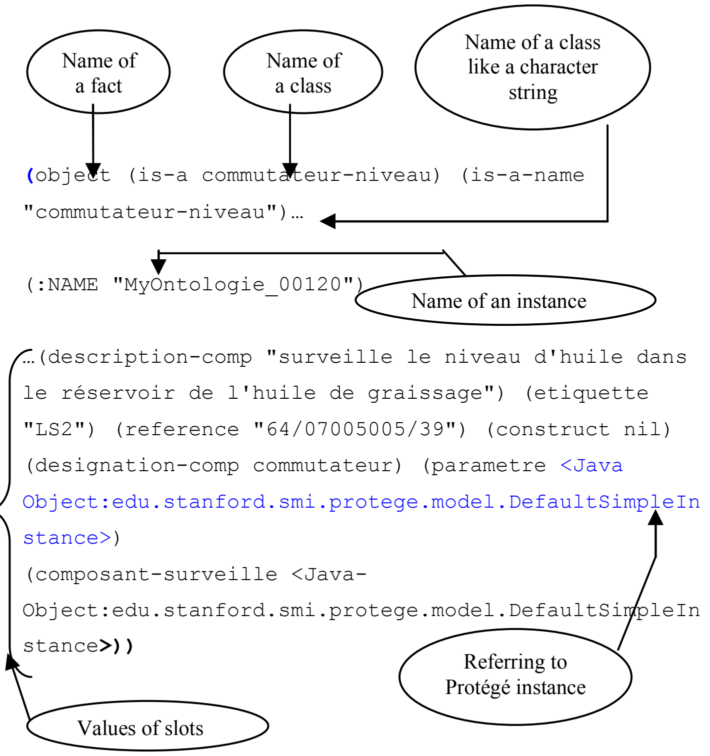


Figure 7: Model of the facts corresponds to Protégé2000 instances

5.6- SPECIFICATION OF THE AXIOMS IN JESSTAB:

The specification of the axioms in JessTab [2] is translated by the rules. To specify a rule in JessTab, it is necessary to specify: the name of the rule, its premises and its conclusions. The definition of a rule is done thanks to the function « **defrule** ».

(**defrule** name of the rule
``commentary''
premise1, premise2…, premiseN
=> action1, action2…, actionP)

The rule will start if each premise can be unified with a fact, or if a condition is checked.

For the translation of the axioms in JessTab, the left part in an axiom represented in logic represents the premises and the right part represents the conclusions.

Some examples of axioms translated into JessTab:

1. **axiom8:** ∀ I, ∃ J, Instrument (I), Composant (J) => surveille (I, J). is translated in JessTab as follows:

(defrule axiome8 **(object (is-a** commutateur-niveau) (**:NAME** ?n))

=> **(bind** ?e **(slot-get** ?n composant-surveillé)) **)**

2. **axiom9 :** ∀ I, ∃ P, Instrument (I), Paramètre (P) => contrôle (I, P)
   is translated in JessTab as follows:

(defrule axiome9 **(object(is-a** commutateur-niveau)(**:NAME** ?n))

=> (**bind** ?e (**slot-get** ?n parameter)))**)**

3. **axiom11** ∀ A, Alarme (A), ∃ C, cause-probables (C) => produire-à-cause (A,C)
   is translated in JessTab as follows:

(defrule axiome11 **(object (is-a** alarme) (causes $?c)
(**:NAME ?n**)) => (**foreach** ?e (**slot-get** ?n causes)
(printout t (slot-get ?e description) crlf)))

4. **axiom13** : ∀ C, cause-probables (C), ∃ R, remède (R) => possèder (C,R).
   is translated in JessTab as follows:

(defrule axiome13 **(object (is-a** causes-probables)
(:NAME ?m)) => (**foreach** ?f
(**slot- get** ?m remèdes)
printout t (slot-get ?f designation)
crlf)))

6. EXAMPLES EXTRACTED FROM THE APPLICATION:

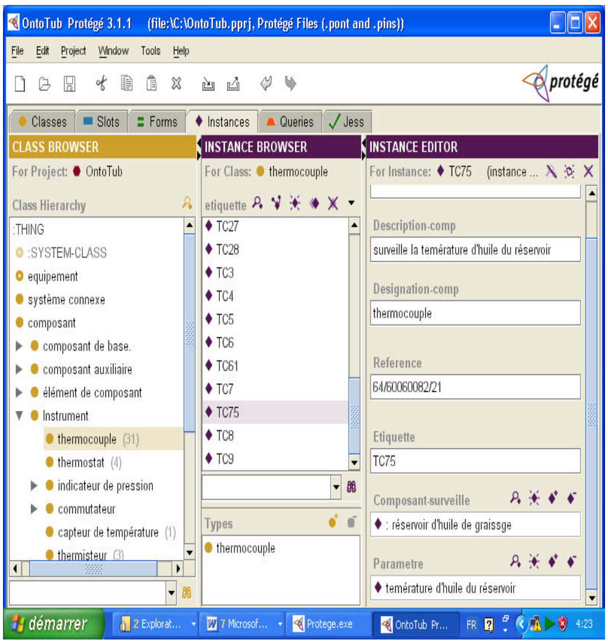


Figure 8: Part of industrial Gas turbine ontology [Lal 07]

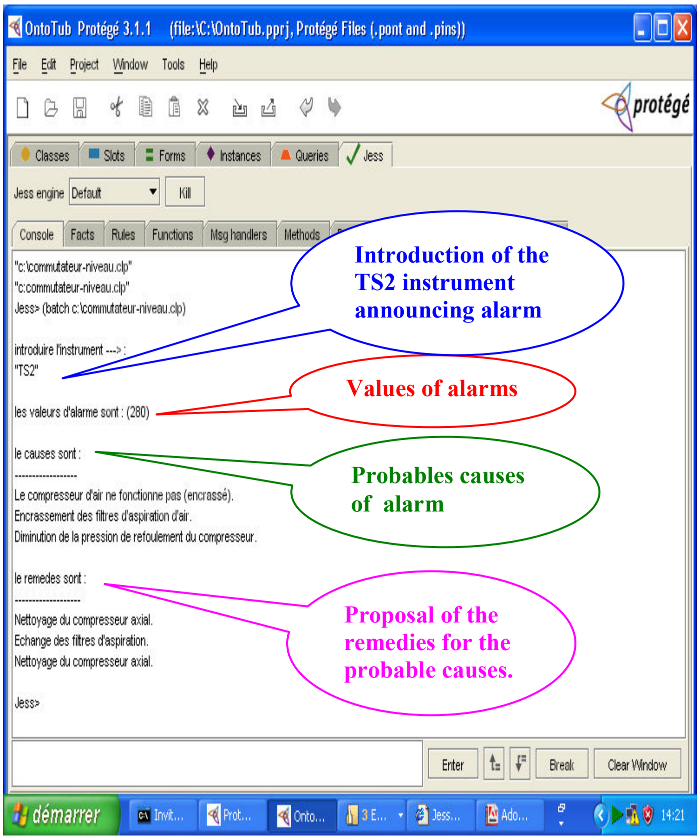


Figure 9: Example of diagnosis concerning the instrument liquid level switch TS2

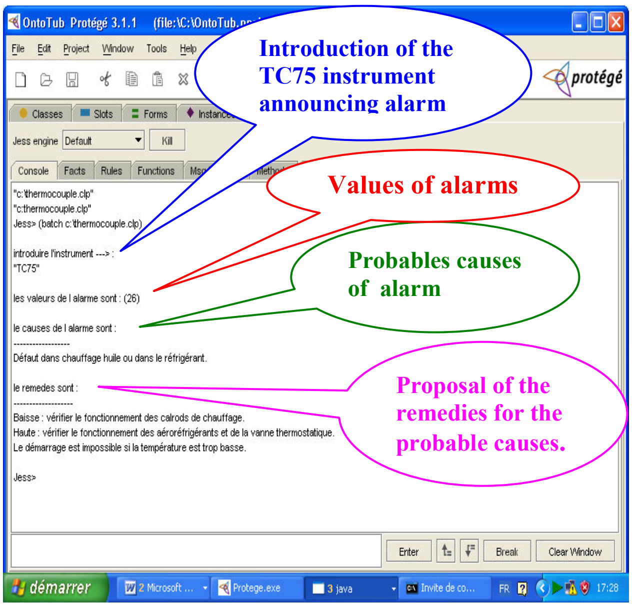


Figure 10: Example of diagnosis concerning the instrument thermocouple TC75

7. CONCLUSION:

The work presented in this article is inscribed in the frame of ontological engineering, and more particularly treats the integration of ontologies of domain within the Knowledge-based systems with operational ends.

We tried in this work to present our contribution concerning the Operationalisation of ontologies. After a stage of conceptualization and ontologisation that we published in [6], we tried in this work to integrate the ontology conceived within a Knowledge-based system by using the JessTab tool. The goal of the system is to help the agent of maintenance of the industrial gas turbines in its spots of diagnosis while presenting to him, following an alarm detected by an instrument, a list of probable causes and instructions.

The ontology conceived in [6] is already represented in a comprehensible form by the machine using Protégé2000. For the Operationalisation of ontology, we have given rules allowing passing from an ontological representation of knowledge to an operational form. These rules adopted for the modeling of the inference were based on the specification of the operational objective by scenario of use. We thus validated ontology gas turbine designed in a particular applicative environment. Our operational ontology can be used in several manufacturing. It can be used in the domain of production of electricity (like in SONELGAZ) or in the domain of the production and transport of gas and petrol (like in SONATRACH). We have validated our ontology with the experts of the domain and we hope that it will be really applied in the concerned manufacturing.

Bibliography:

[1] Eriksson H. "JessTab Tutorial". http://www.ida.liu.se/~her/JessTab/tutorial06/ .

[2] Eriksson H. "JessTab Manual, Integration of Protégé and Jess". Janvier 2004. http://www.ida.liu.se/~her/JessTab/JessTab.pdf

[3] *Furs*t *F.* " *L'operationalisation des ontologies* " . http://www.afia.polytechnique.fr/plateforme-2003/Articles/RJCIA/rjcia-03-Furst.ps

[4] Furst F., *Contribution à l'ingénierie des ontologies : une méthode et un outil d'opérationnalisation*. Thèse de doctorat d'université 2004, Université de Nantes. P184.

[5] http://herzberg.ca.sandia.gov/jess.

[6] Laallam F.Z, Sellami M. "Gas Turbine Ontology for the Industrial Processes". Journal of Computer Science 3 (2):.ISSN 1549-3636. pp.113-118, 2007.

[7] Protégé2000 "Protégé Ontology Editor and Knowledge Acquisition System", http://protege.stanford.edu

[8*] Solari J.J. 'OWL Web Ontology Langage Use Cases and Requirements'.*] http://www.yoyodesign.org/doc/w3c/webont-req-20040210/