7th WSEAS Int. Conf. on MATHEMATICAL METHODS and COMPUTATIONAL TECHNIQUES IN ELECTRICAL ENGINEERING, Sofia, 27-29/10/05 (pp355-361)

A Declarative Language for Representation of Knowledge : Ibn Rochd

MOHAMED TOUAHRIA (*) — ABDALLAH KHABABA (*) — LOUIS FRECON (**)

(*) Computer Science Department, Farhat Abbas University, DZ 19000 Setif, Algeria.

(**) Computer Science Department, INSA of Lyon, F69621 Villeurbanne, France.

ABSTRACT. *This paper proposes a declarative language for knowledge representation (Ibn Rochd), and its environment of exploitation (DeGSE). This DeGSE system was designed and developed to facilitate Ibn Rochd writing applications. The system was tested on several knowledge bases by ascending complexity, culminating in a system for recognition of a plant or a tree, and advisors to purchase a car, for pedagogical and academic guidance, or for bank savings and credit. Finally, the limits of the language and research perspectives are stated.*

KEY-WORDS *:Knowledge representation, declarative language, Ibn Rochd, DeGSE, facets, cognitive approach.*

1 Introduction

A major field of the Artificial Intelligence (AI) is the design of declarative systems, commonly called Systems at Knowledge Bases (KBS) or Expert Systems (ES). These systems are characterized by a separation between knowledge necessary to solve a problem and the mechanisms exploiting this knowledge [1]. This separation makes it possible to describe knowledge independently of their later use. This facilitates on the one hand the modification and the addition of new knowledge at the base ; in addition, one can provide justifications and explanations of the behavior of the system.

Such systems are above all the software which simulates the reasoning of an expert. It thus draws its motivation and its justification in the human experiment [2]. And it is by it, that it must validate its semantics.

The artificial intelligence is thus at the crossroads of data processing and the "sciences cognitives", who aim at understanding how knowledge is born, be used, evolved and is transmitted. If the purpose of a program of AI is to imitate an intelligent behavior, it is necessary for him as well as an individual to have access to knowledge.

However, on their data-processing representation the effectiveness depends on the program [3]. This is why, the representation of knowledge is a key question of the AI. It consists in effectively formalizing the concepts of the real world in an exploitable form by "intelligent" programs [4].

One can leave the simple idea that knowledge results from a *correspondence* of the real world with a system symbolic ; system which makes it possible to reason [5]. The *cognitive step* then consists in ensuring this correspondence using given processes.

According to Vogel [6], knowledge relates to the still impregnated information of he vision of its enonciator. In the field of the artificial intelligence, knowledge relative to an expertise is usually divided

into various categories. Among those, most usual present themselves according to various criteria : *scientific knowledge* and *empirical knowledge*, *major knowledge* and *knowledge of surface*, *factual knowledge* and *procedural knowledge*, *expert knowledge* and *fortuitous knowledge*.
The definition of a formalism in the form of rules of production exploited by a general mechanism -the engine of inference- uses a paradigm of representation and exploitation of knowledge [7], true incarnation of the declarative programming.

The process of collection of expertise is based on a distinction between axis of the models and axis of the paradigms [8]. The problems of the acquisition of knowledge are centred on the level of knowledge, knowledge which it is necessary well to study for better formalizing it and structuring. One can postulate that the treatment of knowledge includes :

- a phase of factual comprehension, inductive and which leads to rules of representation of knowledge starting from the facts observed ;
- a deductive, cognitive phase of decisions and which allows to envisage consequent drivings according to a logic guided by a strategy aiming at to control the validity of the obtained conclusions.

Our main purpose is to show the interest and the contributions of the language of representation of knowledge (KRL) Ibn Rochd [9] to resolve problems of representation, and in second time achieving a Cognitive Genius Workshop, beginning by an environment of exploitation Expert Systems **DeGSE** and the KRL.

The work presented in this paper is situated in the context of an approach object, in the style of Sygemor [10] who uses the *Networks of Worlds*, (to see Shirka [11]). He joins within the framework of the centred models objects. We shall find two essential notions there : the *classes* and the *instances* or the *objects*, detailed in section 2.

2 Definition of the Language of Representation of Knowledge

A language of representation of knowledge wants a support of a reasoning with which it supplies links: his power is that of a programming language, but it is less directive in its form [9].

The popular aspects are :

- *descriptive capacity*, knowledge of a domain, laws and states of a system… ;
- *capacity heuristics*, knowledge useful for the resolution of problems, operational concepts, strategies: rough-hewing, focalisation, planning ;
- *granularity*, assured by very autonomous elementary constructions, facilitating incrémental development and evolutionary maintenance [9].

A Knowledge base is the projection of an expertise in a Langage for Representation of Knowledge.

2.1 Representation of the knowledge in Ibn Rochd

In Ibn Rochd, exists two types of knowledge: the *descriptive knowledge* and the *knowledge operating*.

2.1.1 Descriptive knowledge

Describe the frame of the problem (knowledge), model (classes) and objects. When several objects, possess a structure and a common behavior, it is very useful to group together them in a common mould : a **class**.

♦ **The Classes**

A class is a general model from which will be genered *objects* : intances of class.

Principes : A class groups together similar objects. The classes :

- offer a hierarchical organization of the knowledge ;
- federate a very big number of information which will as well serve for describing the abstract model, that to use these knowledge ;

- represent generic information, and realize the descriptive model of the knowledge.

The *inheritance of properties* allows to group together the similar objects in a subclass and the more general objects in *a super-class (or class-mother)*.

A class represents a general model with a set of *attributes.* All classes will be build up :

- in a *autonomous* way, all the attributes being defined at its level ;
- by *refinement* of a class-mother, whose new class resumes the attributes, to which she adds her appropriate characteristics : we speak then about *simple inheritance*.

♦ **The Attributes and the facets**

The *Attributes* possess a name, and a set of facets, statements (declarations) or characteristics (sometimes optional) of these attributes. *Facets* represent all the characteristics for every attribute, such as: *type*, *Cardinality*, *Interval*, *Domain*, *Question*, *Defect*, If_Need, *Validity* and Advice.

Syntax :

CLASS : <name of class>
[**INHERIT_OF** : {<name of class>}*]
[**ATTRIBUTE** :
{<name of attribute> :
TYPE :integer|real|string|[name-of-class]
[**DOMAIN** : (value_1, …, value_n)]
[**CARDINALITY** : (simple | multiple)]
[**INTERVAL** :[limit_inf, limit_sup]]
[**IF_NEED** : [call of procedure]
[**VALUE**: expression |
QUESTION : "character string" |
DEFECT : value_i] }*]

{}*: possibility of having zero or several cases (occurrences).

Semantics :

- DOMAIN and INTERVAL are of use to restreiendre the type ;
- VALUE introduces a calculable attribute, which aligns itself automatically with the attributes on which it depends, so assuring the coherence of the object ;
- QUESTION activates (starts) a dialogue ;
- DEFECT allows to introduce a useful typical value for lack of know specific value ;
- TYPE : *name of class* clarifies an interweaving. The imbricated object will be inevitably or possiblement present. Example :

CONTENTS : PALETTE, contens are a palette.

CONTENTS : [PALETTE], the possible contents are a palette.

- Genericity : we can define macro, hyper-classes in the style below, where #object indicate a substitutable symbol.

CLASS LIST (#object)

first : [CONTAINER(#object)]

last : [CONTAINER(#object)]

cardinal : VALUE (if first = NOTHING then 0 else first.nb) END LIST.

♦ **Objects**

The copies which we can make from a class are instances or *objects* of this class. Objects have common properties defined by their class, in the form of lists of couples (**attribute, value**) where *attribute* indicates an attribute of the class or the super-class, and where *value* is compatible with the facets of the attribute.

A specific object already having a name can be directly appointed by this name, or indicated anonymously by $, followed by the name of a class of the object : example : $bird indicate *some* bird. To compare objects by couples, triplets etc. Supposes to arrange so many different variables by rules (a figure discriminating in suffix, offering in a rule up to ten variables sweeping simultaneously the same class, is amply enough.

2.1.2 Knowledge operating

They express the expertise in a declarative way (rules). The rules of production represent the formalism of representation of the knowledge by far the most used in expert systems. Expressed as a couple Premises/Consequents, they express knowledge Operating whose implemented consists in looking for for a given context the rules the part of which *Premises* is verified, then to activate them, that is to execute the part *Consequent* which will have the effect of modifying the context. The rules of production:

- represent the dynamic part of the knowledge base (KoB) ;
- offer legibility and ease of writing ;
- offer the possibility of balancing the knowledge.

Syntax :

R <number.number>
{**IF** <premise>}$^{\oplus}$
THEN
{<action>}$^{\oplus}$
BECAUSE <justification>
{}$^{\oplus}$: expressed at least once.

♦ **Premises**

The *premises* compare *terms* : attribute, constant, instance of a class, specific or anonymous. They use for it *comparative operators*, relational operators among " =, <,>, <=, >=, IN and EXCEPT (for the consequentive list), DIVIDE and MUTIPLE ".

♦ **Actions**

In this part we can make operations of additions, modifications or retreats of information in the base of objects.

- Operation of affectation : = impose a new exclusiv value.
- Operation of addition + = adds a value to a multiple attribute.
- Operation of abolition (deletion) - = removes a value (or some) from a multiple attribute : the new value is the ensembliste difference between the ancient (former) value and the modifier.

Ibn Rochd offers a dynamic management of the operations on the instances of a given class and allow of :

- create an object : the action CREATE creates a new instance of the indicated class ;
- remove an object : the action KILL (name of object) inverse operation CREATE, removes all values from this object.

2.1.3 Operations of Communication

They aim at acquiring data or to post (show) results.

- **READ**

This operation allows to read an entity (value of an attribute) of the base.

Examples:

```
READ<object-designation>.<attribute-
          name>
READ "continuation of characters"
+$<class-name>.<attribute-name>
CLASS MENU
ATTRIBUTES :
  POST : LIST (CHAIN)
  SHOW : METHOD
  {    SCREEN.CLEAN
       POST.WRITE
   }
  CHOICE : TYPE INTEGER
       VALUE :
     {  *.SHOW
        WRITE « YOUR CHOICE ? »
        READ CHOICE
      }
END MENU
```

- **WRITE**

This action shows texts, information and/or some entity and/or one (or some) values of a relation associated to a given entity.

- **Management of Screen**

For a bigger legibility of the applications, Ibn Rochd offers the possibility to the knowledge engineer to manage its screen. Before every operation of writing, the construction IN(line_number, column_number) positions the cursor in the place specified by the indicated coordinates. RAZ SCREEN : Clean screen.

2.1.4 Piloting and Meta-operations

Ibn Rochd offers to the knowledge engineer the possibilities of intervention on order of examination of rules.

- INSPECT {number of rule}*), the priority evaluation of the mentioned rules provokes.
- FREEZE ({number of rule}*): inverse operation INSPECT, makes inactive

rules, whatever is the order to their place in the " list of the conflicts".

2.2 Comments

Ibn Rochd use three types of comments:

- comments without effects, beginning with the symbol '/* ' ;
- comments used by the engine in the writing of the track of the reasoning (explanation), beginning with the symbol '!' ;
- BECAUSE : justification.

3 The environment / Workshop DeGSE

DeGSE workshop is an environment of Development of Big Expert Systems in Ibn Rochd.

This workshop :

- Support the declarative language Ibn Rochd, on base of rules of production and objects ;
- analysis, code, archives and manages knowledge bases described in this language ;
- proposes a Editor of Knowledge (KoE) to write them and modify them;
- work in deductive mode (natural deduction), with possibility of track of its functioning.

This workshop is dedicated :

- in the Development : experts and knowledge engineer use KoE and Executive to produce and finalize a KoB ;
- in the Exploitation : the end users use the Executive coupled with the KoB (specific of the application).

As is, DeGSE is operational under Windows XP / 98/ ; it contains 4 modules including approximately 20 thousand lines of C++.

4 Bases and Validation

The validation is a check of the product with regard to the initial specifications. The at present formed train of tests is organized by increasing complexity.

This benchmark allowed to validate the language and the first model of DeGSE 1. This task is frequently situated on the most critical road of the project and constitutes one of the main elements of increase of details, for the definitive fulfillment of the work.

In fact, this validation is two floors : does DeGSE support suitably Ibn Rochd ? Is Ibn Rochd a KRL corresponding to our ambitions ?

In this respect, the validation of expert systems differs substantially from that of the other computer programs as far as :

- the specifications of expert systems are called to evolve, sometimes in a very sensitive way, throughout the cycle of development ; therefore, the validation of the expert systems cannot consist only of a check of the adequacy of the product to its specifications ;
- most of the time, there are no objective criteria to decide if the found result is indisputably the best.

The train of current test is formed by knowledge bases ordered by increasing complexity : search (research) for a Perfect number, CORUS : educational guidance and univesitaire, detailed in [12], Diagnosis of an Anaemia, Recognition of a Tree and a Financial Advice. Seen the number of bases developed knowledge, we shall present only an extreme (Advice for the Purchase of a Vehicle).

4.1 Advice for the Purchase of a car

♦ **Problem**

To Develop an Expert System of Advice for the Purchase of Vehicles.

♦ **Analyze**

At first, the composed question is that to realize a huge expert system, realizing the operations on thousands of objects. Eventually, it will be a question of making the modelling of a knowledge base on hundred of objects.

Seen the number importing **countries** and **groups** (Psa, Vag, Bmw, Fiat) builders (manufacturers) of cars and *type* of cars that every country can produce or groups, we limit ourselves to four big producing countries represented in Algeria : France for Renault, Germany for the Group Volkswagen (VW, Audi, Skoda, Seat), Japan for Nissan and the USA for General Motors (Opel, Daewoo)[1].

Modelling a knowledge base for such application revealed a delicate task.

♦ **Realization**

Was structured and coded according to the agreements of writing of the KRL Ibn Rochd, this knowledge base contains at present :

- 5 classes : *car*, *technique*, *buyer*, *performance*, *equipment* ;
- more than 400 *objects* (models) : 135 for Renault, 126 for Volkswagen and 95 for Nissan ;
- hundred of *rules*.

The class *buyer* allows for one person to formulate a choice. An object deserves here a certain attention, the object *asks*, at first initialized *vacuous* then gradually filled (as a form) by a continuation (suite) of questions put by the system to the buyer.

♦ **Tested Aspects :**

After interpretation of the base above, the system detects and indicates all the abnormalities in the writing of the base to the user. After correction of the base, the system creates the structures of data necessary for the evaluation of the base. All the made essays (attempts) showed themselves correct.

A validation from the editor of knowledge invites the user in a multitude of options ; and the choice of some of them can lead to vague situations :

- If the knowledge base contains only a single attribute, its abolition (deletion) returns its reference class without attribute, what presents a flimsiness in the modelling ;
- the existence of a single attribute with *facets* reduced to the only *type* facets and cardinality, shows a lack of expressiveness ;
- we noted the lack of operations reflexes (call of procedures), to boost the exploitation in case when the attribute value does not possess value.

[1] Magazines Automobile, Auto journal n°625, 2003.

5 Conclusions

The purpose of this paper was to show the contributions of the language of representation of knowledge Ibn Rochd. The realized works were applied to several expertise. These experiments confirmed the interest of this language, and bring us to postulate that the treatment of the knowledge includes :

- a phase of factual, inductive understanding and which leads to rules of representation of knowledge from the observed facts ;
- a deductive, cognitive phase of decisions and which allows to envisage consequent drivings (behaviors) according to a logic guided by a strategy aiming at to control the validity of the obtained conclusions.

We were so able to :

- validate Ibn Rochd1 and DeGSE1 by means of small bases ;
- then model vaster and more varied knowledge bases, seekers of improvements ;
- combine finally gradually software engineering and needs of the cognitive methodologies, with the aim of a cognitive genius.

This postulate allowed us :

- to conceive Ibn Rochd as a declarative centred language object for the representation of the knowledge ;
- to conceive his DeGSE1 environment, which codes bases, lists the classes, the attributes, the objects, the rules and in the execution, supplies a formed track.

5.1 Limits of Ibn Rochd 1

The prototype DeGSE 1 was realized to test the legitimacy of the basic principles of the KRL Ibn Rochd1. The objective of this prototype for the developer of knowledge bases was to supply a editor of knowledge for the preparation, and an executive (which an interference engine) for the execution, to reveal the incapacities and remedy it as a consequence.

In view of the tests made thanks to DeGSE1, certain problems bound to the language Ibn Rochd1 supported appeared. The problem is now to supply the satisfactoriest possible KRL.

Except bugs, the defects of the KRL can be two orders :

- absence of certain facets « reflexes » that must avoid to the user a forced guide, towards a not always convincing solution ;
- rules with static priority, allowing to distinguish the likely of the truth, but not allowing the evolution by learning.

5.2 Specifications of Ibn Rochd 2 and Perspectives

To treat these problems, it is not only to improve the KRL, but also it is also to relieve gradually the methodology to be spread upstream to the coding, and the objections that we can make to the system. Where from the necessity of :

- introduce a revisable priority (heuristics) by meta-rules according to the obtained results (learning) ;
- allow a coupling of DeGSE with the relational data bases: consider DeGSE as a pit cooperating with data bases ;
- integrate features of control of coherence, to pass from a editor to a publisher / tax auditor (anticipating better the number of conditions) ;
- integrate an online help ;
- introduce a function of compilation (translation C++ or in Java) applications focusing.

So strengthened, DeGSE 2 could be of use to the development of expert systems getting more varied domains : the behavior of the new system should show itself rather supple to allow an artificial intelligence to express, to represent and finally to exploit easily its knowledge, without having to for all that undergo useless computer rigidity.

References:

[1] Tayar N, *Management of the versions for the construction incrémentale and shared by knowledge bases*, Doctoral thesis of state, Joseph Fourier university, Grenoble, 1995.

[2] Touahria M, Frécon L, *Workshop of Cognitive Genius for Ibn-Rochd 1/2*, I.N.S.A of Lyon, Computer Science Department, 2002.

[3] Hamon Y, *Integration of Techniques of Artificial intelligence in the Workshops of Software engineering, application to the workshop Softpen: SAIGL*, Doctoral thesis in Data processing Lyon 1, 1993.

[4] Massini G, Napoli A, Colnet D, Léonard D, Tombre K, *object-oriented languages, languages of classes, languages of frames, actors languages*, Inter-Editions, Paris, 1989.

[5] Decourbe B, *factual Understanding and cognitive decisions, Seen again(Revised) by Artificial intelligence*, Flight 1, n°2, 1987.

[6] Vogel C, C*ognitive Ge*nius, Masson, 1988.

[7] Caplat G, C*ognitive Modelling and resolution of problems*, PPUR, Lausanne, 2002.

[8] Vogel C, *KOD : the implemented*, Masson, 1990.

[9] Touahria M, Frécon L., *Workshop of Cognitive Genius for Ibn-Rochd 1*, I.N.S.A of Lyon, Computer Science Department, 1999.

[10] Sohbi Y, *Study and Realization of SYGEMOR: environment for Systems Multi Experts Centred O*bjet, Doctoral thesis of the INSA of Lyon, 1990.

[11]ReichenmannF,*SHIRKA : management system of centred knowledge bases object*, Manuel of reference, INRIA / ARTEMIS, Grenoble, 1988.

[12] Khababa A., Touahria M., Frécon L., cognitive Modelling a system for pedagogical and academic guidance CORUS, Computer Science Department, Sétif, (DZ), 2004.