RAMOLI: A Generic Knowledge-based Systems Shell for Symbolic Data

Saoussen Bel Hadj Kacem[1], Amel Borgi[2], Moncef Tagina[1]
[1]National School of Computer Science
Manouba University, Tunisia
SOIE research laboratory, ISG
Tunis University, Tunisia
[2]Higher Institute of Computer Science
LIPAH research laboratory, FST
Tunis El Manar University, Tunisia
Saoussen.BHK@ieee.org, Amel.Borgi@insat.rnu.tn, Moncef.Tagina@ensi.rnu.tn

ABSTRACT: *Non classical logics were introduced to allow handling imperfect concepts in intelligent systems. One of the principal non classical logic is multi-valued logic that has the particularity to support symbolic data. We introduced in a previous work an approximate reasoning in the multi-valued framework based on linguistic modifiers that checks approximate reasoning axiomatics. This paper describes the development of software model for the treatment of imperfection with our approach of approximate reasoning. It is a knowledge-based systems shell for symbolic data called RAMOLI. This shell provides simple and interactive Graphical User Interface to introduce knowledge and to infer with our approximate reasoning.*

Keywords: Knowledge-based System, Symbolic Multivalued Logic, Approximate Reasoning, Linguistic Modifiers, Computing With Words

Received: 25 February 2013, Revised 31 March 2013, Accepted 5 April 2013

1. Introduction

In recent decades, researchers have emphasized the consideration of imperfect knowledge in intelligent systems. Indeed, knowledge manipulated by human mind can be vague, imprecise, uncertain, etc. Several non-classical logics were therefore born. The principal non classical logic is fuzzy logic introduced by Zadeh [1]. It is based on fuzzy set theory which is generally based on a numerical domain. However, according to some authors [2] [3] [4] fuzzy logic does not allow the representation of abstract terms from natural language whichdo not refer to numerical scales, such as ugly, beautiful, intelligent, etc. Symbolic multi-valued logic [2] [3] [5] is another non-classical logic that allows a symbolic representation of terms, it is based on multi-set theory.

After the phase of knowledge representation, is the phase of knowledge manipulation by means of intelligent systems. We focus in this work on Knowledge-Based Systems (KBS). A KBS is a computer software that uses artificial intelligence to solve problems that often require the additional expertise of a human authority [6] [7] [8] [9]. It is able to support explicit representation of knowledge in a specific area of expertise, and to exploit them through appropriate reasoning mechanisms. The goal is to provide a high level performance in problemsolving. KBS is one of the few industrial success of Artificial Intelligence.

In the literature, several fuzzy KBS shell have been proposed in fuzzy logic context like FuzzyShell [10], FuzzyClips [11], FuzzyJess [12], Fuzzy Prolog [13]. In the multi-valued framework of Akdag et al. [5], only one KBS shell was developed by Khoukhi [14] and called Nuance. However, we proved in [15] that approximate reasoning approach used in Nuance does not check approximate reasoning axiomatics proposed in the literature [16] [17], and we proposed a new approach of approximate reasoning that check this axiomatics.

Our goal in this work is the construction of a KBS shell for symbolic data. The role of this shell is to provide user interfaces that allow constructing a symbolic knowledge base. Moreover, it integrates an inference engine that deduces new facts from this knowledge base. Symbolic knowledge are represented and treated with multi-valued logic, context of our work. In particular, the inference engine of the system uses an approximate reasoning that we proposed and presented in previous work [15] [18] [19] [20]. We call our KBS shell *RAMOLI* (Raisonnement Approximatif base sur les MOdifi- cateurs LInguistiques).

This paper is organized as follows. In Section II, we focus on knowledge representation in symbolic way, and we briefly describe the basic concepts of multi-valued logic, the context of our work. Section III is devoted to the presentation of our approximate reasoning which is based on linguistic modifiers. In section IV, we describe the design and implementation of the KBS shell. Then, section V presents a description of the Graphical User Interfaces (GUIs) of the system, allowing one hand to introduce knowledge and secondly to deduce new knowledge. Finally, section VI concludes the work.

2. Knowledge Repersentation

Multi-valued logic introduces symbolic truth degrees which are intermediate between true and false [5]. According to this logic, every linguistic term is modeled by a multi-set. It generalizes classic set theory: the notion of belonging or not to a set is replaced by a partial belonging to a multi-set. The set of possible truth degrees is $\mathcal{L}_M = \{\tau_0,...,\tau_i,...,\tau_{M-1}\}$[1] with the total order relation: $\tau_i \leq \tau_j \Leftrightarrow i \leq j$, its smallest element is τ_0 (false) and the greatest is τ_{M-1} (true) [5], [21].

A possible list of truth-degrees used in [5] for $M = 7$ is $\mathcal{L}_7$ = {*not-at-all, very-little, little, moderately, enough, very, completely*}.

On the scale of truth degrees $\mathcal{L}_M$, operators can be defined to aggregate degrees as implications, T-norms and T-conorms. In multi-valued logic, the aggregation functions of Aukasiewicz are often preferred [5], [22]. In this context and with M truth-degrees, they are defined by:

$$T_L(\tau_\alpha, \tau_\beta) = \tau_{max(\alpha+\beta-M+1,\,0)} \tag{1}$$

$$S_L(\tau_\alpha, \tau_\beta) = \tau_{min(\alpha+\beta,\,M-1)} \tag{2}$$

$$\mathcal{I}_L(\tau_\alpha, \tau_\beta) = \tau_{min(M-1-\alpha+\beta,\,M-1)} \tag{3}$$

These qualitative degrees can be considered as membership degrees of multi-sets. Indeed, "X is $\upsilon_\alpha A$" means that υ_α is the degree to which X satisfies the multi-set A[2]. In other words, the predicate A is satisfiable to a certain degree expressed through the scalar adverb υ_α associated to the truth-degree τ_α of $\mathcal{L}_M$.

Multi-valued logic is based on the following interpretation:

$$\begin{aligned} X \text{ is } \upsilon_\alpha A &\Leftrightarrow \text{“}X \text{ is } \upsilon_\alpha A\text{” is true} \\ &\Leftrightarrow \text{“}X \text{ is } A\text{” } \tau_\alpha\text{-true} \end{aligned}$$

For example, the statement "*John is rather tall* " means that John satisfies the predicate tall with the degree rather.

---

[1] with M a positive integer not null, which represents the number of truthdegrees in the scale $\mathcal{L}_M$.

[2] Denoted mathematically by "$X \in_\alpha A$": the object X belongs with a degree α to the multi-set A.

3. Approximate Reasoning Based on Linguistic Modifiers

In order to manage imperfect knowledge in intelligent systems, Zadeh has introduced the concept of *Approximate Reasoning* [23]. It is based on a generalization of *Modus Ponens* (MP) known as *Generalized Modus Ponens* (GMP). This rule can be expressed in standard form as follows:

$$\frac{\begin{array}{ll}\text{If} & \text{“}X \text{ is } A\text{” then “}Y \text{ is } B\text{”} \\ & \text{“}X \text{ is } A'\text{”}\end{array}}{\text{“}Y \text{ is } B'\text{”}} \tag{4}$$

where X and Y are linguistic variables and A, A', B and B' are fuzzy sets. GMP serves to infer not only with an observation exactly equal to the rule premise (“X is A”), but also with an observation which is different but approximately equal to it (“X is A'”). This allows to handle imprecise knowledge in the inference process.

To determine the inference conclusion (“Y is B'”), a set of axioms is taken into account in order to have a logical and coherent result in concordance with humain reasoning [16], [17]. In [15], we have proposed the generalization (5) of criteria appeared in [17]:

$A' \succ$ A means that A' is a reinforcement of A,
$A' \prec$ A means that A' is a weakening of A.

C I	$A' = A \Rightarrow B' = B$
C II-1	$A' \succ A \Rightarrow$ the more $A' \succ A$, the more $B' \succ B$
C II-2	$A' \succ A \Rightarrow B' = B$
C III	$A' \succ A \Rightarrow$ the more $A' \prec A$ the more $B' \prec B$
C IV-1	$A' = \overline{A} \Rightarrow B'$ is indefinite
C IV-2	$A' = A \Rightarrow B' = \overline{B}$

(5)

Existing works in multi-valued framework of Akdag et al. [5] does not respect these axioms (see [15]). We introduced in a previous work [15] [18] an approximate reasoning that checks this axiomatics and it is based on linguistic modifiers. A linguistic modifier is a fonction that allows expressing the modification that a predicate must undergo to become another predicate. In the multi-valued framework, modification of predicates is performed by dilation or erosion of the scales, and/or increasing or decreasing of the truth degrees. Akdag and al. [24] introduced linguistic modifiers in the multi-valued context and called them *Generalized Symbolic Modifiers* (see table 1).

The GMP of our approximate reasoning based on linguistic modifiers is the following:

$$\frac{\begin{array}{l}\text{If “}X \text{ is } \upsilon_\alpha A\text{” then “}Y \text{ is } \upsilon_\beta B\text{”} \\ \text{“}X \text{ is } m(\upsilon_\alpha A)\text{”}\end{array}}{\text{“}Y \text{ is } m(\upsilon_\beta B)\text{”}} \tag{6}$$

where X and Y linguistic variables, A and B multi-sets and υ_α and υ_β linguistic degrees associated to the truth degrees τ_α and τ_β in $\mathcal{L}_M$. For the GMP (6), the observation is modeled by a modification of the rule premise $m(\upsilon_\alpha A)$, where m represents a linguistic modifier among those of table 1 [24]. m is determined by the following algorithm:

Algorithm 1: **det_mod (multi-set** $\upsilon_\alpha A$, multi-set $\upsilon_{\alpha'} A'$): **modifier** m

M_A and $M_{A'}$ are the basis of A and A' respectively.

- If $M_A = M_{A'}$ and $\alpha = \alpha'$ then $m = CC$.
- If $M_A = M_{A'}$ and $\alpha \neq \alpha'$, m is equal to the operator $m_{\rho 1}$, defined below, which allows to modify the degree τ_α.
- If $M_A \neq M_{A'}$ and $\alpha = \alpha'$, m is equal to the operator $m_{\rho 2}$, defined below, which allows to modify the base $\mathcal{L}_{M_A}$.
- If $M_A \neq M_{A'}$ and $\alpha \neq \alpha'$, m is equal to the composition of the two operators: $m_{\rho 1} \circ m_{\rho 2}$, with m1 the operator allowing to modify the degree and $m_{\rho 2}$ the one allowing to modify the base.

• Determination of the operator $m_{\rho 1}$ acting on the degree: the radius of the operator is $\rho_1 = |\alpha - \alpha'|$;

– If $\alpha < \alpha'$ then the operator is $CR_{\rho 1}$.
– If $\alpha > \alpha'$ then the operator is $CW_{\rho 1}$.

• Determination of the operator $m_{\rho 2}$ acting on the base: the radius of the operator is $\rho_2 = |M_A - M_{A'}|$;

– If $M_A < M_{A'}$ then the operator is $DW_{\rho 2}$.
– If $M_A > M_{A'}$ then the operator is $ER_{\rho 2}$.

End of algorithm

Example 1: Given the list of truth-degrees $\mathcal{L}_7$ = {not-at-all, very-little, little, moderately, enough, very, completely}, and the following knowledge (rule and observation), one obtains:

MODE / NATURE	Weakening	Reinforcing	Central
Erosion	$\tau_{i'} = \tau_{max(0,\, i-\rho)}$, $\mathcal{L}_{M'} = \mathcal{L}_{max(2,\, M-\rho)}$ $\boxed{EW_\rho}$	$\tau_{i'} = \tau_i$, $\mathcal{L}_{M'} = \mathcal{L}_{max(i+1,\, M-\rho)}$ $\boxed{ER_\rho}$ $\tau_{i'} = \tau_{max(1,\, min(i+\rho)\, M-\rho-1)}$, $\mathcal{L}_{M'} = \mathcal{L}_{max(2,\, M-\rho)}$ $\boxed{ER'_\rho}$	$\tau_{i'} = \tau_{max(\lfloor \frac{i}{\rho} \rfloor,\, 1)}$, $\mathcal{L}_{M'} = \mathcal{L}_{max(\lfloor \frac{M}{\rho} \rfloor + 1,\, 2)}$ $\boxed{EC_\rho *}$
Dilation	$\tau_{i'} = \tau_{i+\rho}$, $\mathcal{L}_{M'} = \mathcal{L}_{M+\rho}$ $\boxed{DW_\rho}$ $\tau_{i'} = \tau_{max(0,\, i-\rho)}$, $\mathcal{L}_{M'} = \mathcal{L}_{M+\rho}$ $\boxed{DW'_\rho}$	$\tau_{i'} = \tau_{i+\rho}$, $\mathcal{L}_{M'} = \mathcal{L}_{M+\rho}$ $\boxed{DR_\rho}$	$\tau_{i'} = \tau_{i\rho}$, $\mathcal{L}_{M'} = \mathcal{L}_{M\rho-\rho+1}$ $\boxed{DC_\rho}$
Conservation	$\tau_{i'} = \tau_{max(0,\, i-\rho)}$, $\mathcal{L}_{M'} = \mathcal{L}_M$ $\boxed{CW_\rho}$	$\tau_{i'} = \tau_{min(i+\rho,\, M-1)}$, $\mathcal{L}_{M'} = \mathcal{L}_M$ $\boxed{CR_\rho}$	$\tau_{i'} = \tau_i$, $\mathcal{L}_{M'} = \mathcal{L}_M$ $\boxed{CC}$

$*$ $\lfloor . \rfloor$ is the floor function.

Table 1. Definitions of Weakening, Reinforcing and Central Modifiers

If a tomato is moderately red then it is enough ripe
$(\tau_\alpha = \tau_3, \tau_\beta = \tau_4)$
This tomato is little red
$(\tau_{\alpha'} = \tau_2)$

This tomato is moderately ripe
$(\tau_{\beta'} = \tau_3)$

Indeed, the modifier m transforms “*moderately*” to “*little*”, so we go from the truth-degree τ_3 to τ_2. Thus, $m = CW_1$ since the operator CW decreases truth-degrees. We apply this modifier to the rule conclusion.

$$CW_1 \,(\textit{enough ripe}) = \textit{moderately ripe}.$$

For the following fact one obtains:

If a tomato is moderately red then it is enough ripe
$(\tau_\alpha = \tau_3, \tau_\beta = \tau_4)$
This tomato is little red
$(\tau_{\alpha'} = \tau_4)$

This tomato is Very ripe
$(\tau_{\beta'} = \tau_5)$

The modifier which transform "*moderately*" to "*enough*" is CR_1. We apply this modifier to the rule conclusion:

$$CR_1(\textit{enough ripe}) = \textit{very ripe}.$$

We extended this approximate reasoning in [19] to handle with heterogeneous knowledge. We mean by this heterogeneity that the multi-set in the observation is not necessarily the same as that of the rule premise, and/or the multi-set in the inferred conclusion is not necessarily the same as that of the rule conclusion. This offers more flexibility in the inference process. The corresponding GMP is the following:

$$\begin{array}{ll} \text{Rule} & \text{If } X \text{ is } \upsilon_\alpha A \text{ then } Y \text{ is } \upsilon_\beta A \\ \text{Observation} & X \text{ is } \upsilon_{\alpha'} A' \\ \hline & \qquad\qquad Y \text{ is } \upsilon_{\beta'} B' \end{array} \tag{7}$$

Linguistic terms of A' and A (respectively B and B') are synonymous. However, A' and B' don't necessarily have the same basis than A and B.

Inference with data whose bases are different leads to the consideration of the proportion of mutli-sets in order to achieve a coherent inference. We propose to normalize all the multi-sets A, A', B and B', i.e. transform them to obtain homogeneous propositions. Now, approximate reasoning of the previous section can be applied. Finally, the inference conclusion, which is in the normalized form, is converted itself to have the desired conclusion form $\upsilon_{\beta'} B'$. The conversion are made by central modifiers that are DC and EC. The following proposition gives this procedure:

Proposition 1: The symbolic degree $\tau_{\beta'}$ in the base $\mathcal{L}_{M_{B'}}$ of the inference conclusion of schema (7) is determined as follows:

M_A, M_B, $M_{A'}$ and $M_{B'}$ are the basis sizes of respectively the multi-sets A, B, A' and B'. Given $M = LCM\,(M_A - 1, M_B - 1, M_{A'} - 1,$ $M_{B'} - 1) + 1$, and the modifier $m = det\_mod\,(DC_{\frac{M-1}{M_A-1}}(\upsilon_\alpha A),\ (DC_{\frac{M-1}{M_{A'}-1}}(\upsilon_{\alpha'} A')$, we have: $\upsilon_{\beta'} B' = EC_{\frac{M-1}{M_{B'}-1}}(m\,(D_{\frac{M-1}{M_{B'}-1}}C\,(\upsilon_\beta B)))$

Example 2: Given the lists of truth-degrees:

$\mathcal{L}_9$ = {not-at-all, very-little, little, somewhat, moderately, more or less, enough, very, completely},

$\mathcal{L}_7$ = {not-at-all, very-little, little, moderately, enough, very, completely},

$\mathcal{L}_5$ = {not-at-all, little, moderately, enough, completely},

$\mathcal{L}_3$ = {not-at-all, moderately, completely},

and the rule:

"*If a tomato is more or less red then it is enough ripe*".

with *red* and *ripe* being multi-sets in the basis $\mathcal{L}_9$ and $\mathcal{L}_7$ respectively (so $\tau_\alpha = \tau_5$ and $\tau_\beta = \tau_4$). The user gives the fact:

"*This tomato is moderately scarlet* ".

with *scarlet* a multi-set considered in the system equivalent to *red*, but on the basis $\mathcal{L}_5$ (so $\tau_{\alpha'} = \tau_2$). The user wants the inference conclusion to be represented with the multi-set *cultivatable*, which is considered equivalent to *ripe* but on the base $\mathcal{L}_3$. The result is:

$$\begin{array}{l} \text{If a tomato is more or less red then it is enough ripe} \\ \hfill (\tau_\alpha = \tau_5, \tau_\beta = \tau_4) \\ \text{This tomato is moderately scarlet} \\ (\tau_{\alpha'} = \tau_2) \\ \hline \hfill \text{This tomato is moderately cultivatable} \\ \hfill (\tau_{\beta'} = \tau_1) \end{array}$$

Indeed, if we consider the multi-set A which will be used in the normalization of the multi-sets of this problem, the base size of A is:

$$M = LCM\,(2, 4, 6, 8) + 1 = 25$$

The modifier which transforms the rule premise to the observation is:

$$m = det\_mod\,(DC_{\frac{24}{8}}(\tau_5 \text{ red}), (DC_{\frac{24}{4}}(\tau_2 \text{ scarlet })) = det\_mod\,(\tau_{15}A, \tau_{12}A) = CW_3$$

So, the inference conclusion is determined as follows:

$$\upsilon_{\beta'} \text{ cultivatable} = EC_{\frac{24}{2}}(m\,(DC_{\frac{24}{6}}(\tau_4\,ripe), (EC_{12}(CW_3(\tau_4\,ripe)) = \tau_1 \text{ cultivatable} = \text{moderately cultivatable}$$

Sometimes expert knowledge must be modeled by complex rules, i.e. rules whose premises are conjunction or disjunction of propositions. For this reason, we improved our approximate reasoning in [20] to deal with complex rules. We introduced for that a new operator that aggregates linguistic modifiers. This aggregator verifies logical connectives properties.

4. Development of RAMOLI

A Knowledge-Based Systems shell is a generic tool that allows the construction of Knowledge-Based Systems. It provides a software platform for building a knowledge base, and provides a generic inference engine that allows the deduction of new knowledge. The knowledge base contains knowledge provided by the expert. It contains operational knowledge or expertise (rules), and factual knowledge which are information on the current state (facts).

In this work we build a Knowledge-Based Systems shell for symbolic multi-valued knowledge, we call it RAMOLI. This shell is a generic tool that can be used in any field. Obviously, in our context the introduced knowledge in the knowledge base must be multi-valued. Domain expertise is represented by multi-valued production rules, and facts by multi-valued predicates. The inference engine implements exact reasoning and approximate reasoning. We have integrated our approximate reasoning based on linguistic modifiers that we had proposed in [15] [18] (cf. section III). Figure 1 show the architecture of RAMOLI.

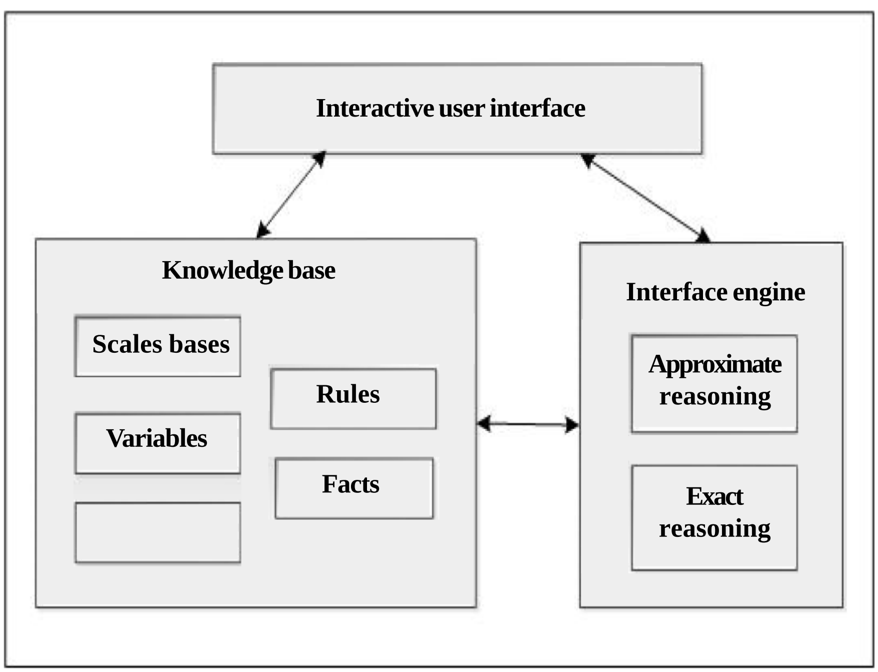

Figure 1. Architecture of RAMOLI

RAMOLI was developed in Java programming language. Thus it provides platform portability, extensibility and easy integration

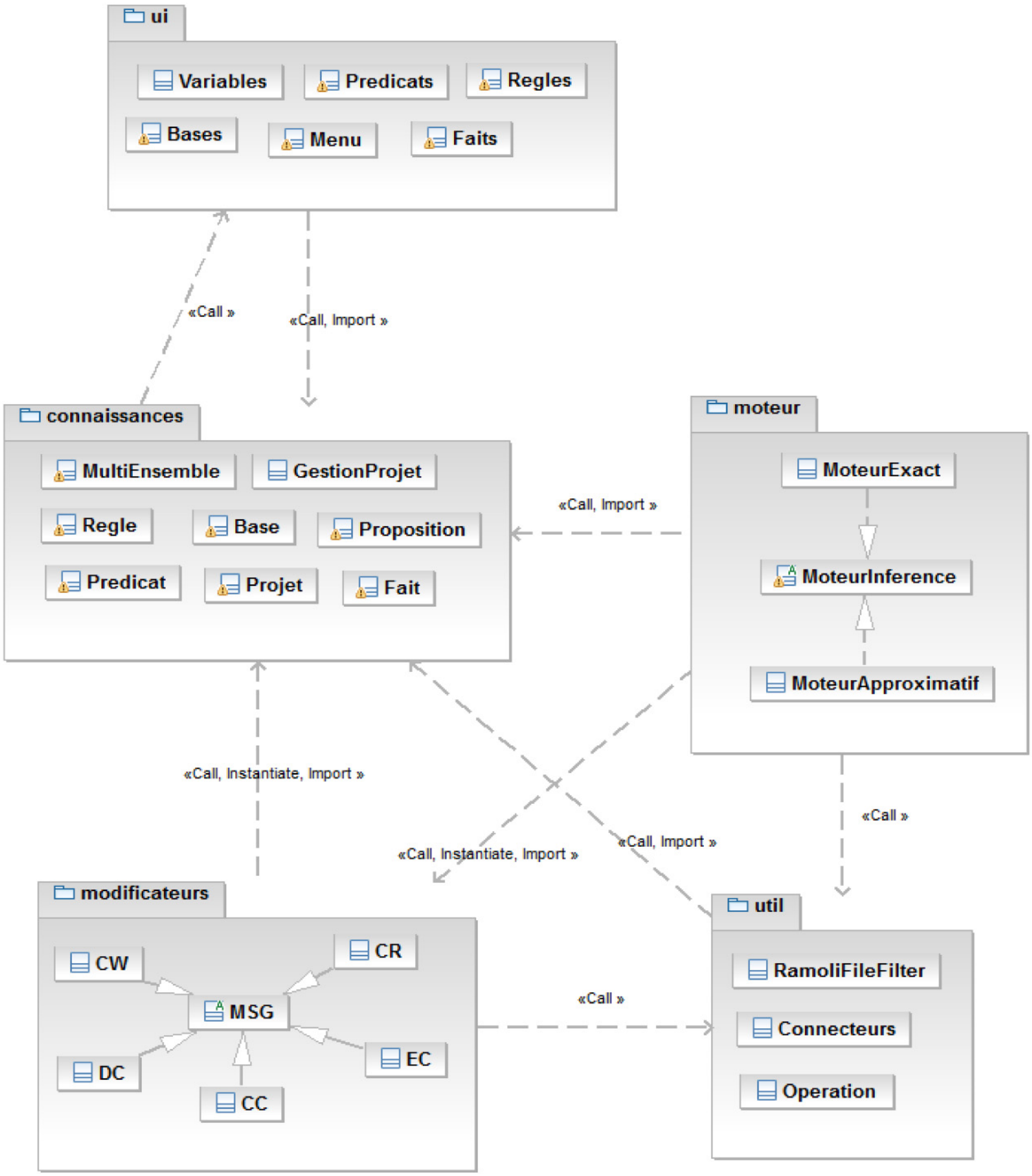

Figure 2. Packages diagram of RAMOLI

with other Java code or applications. The packages diagram of the system is presented in figure 2. The constructed packages in the system are:

• **Package ui:** contains the GUIs of the application.

• **Package Knowledge:** meets the necessary classes for constructing knowledge of the KBS and their management.

• **Package modifiers:** includes linguistic modifiers and their transformation functions.

• **Package engine:** allows to infer with a registered knowledge base.

• **Package util:** includes additional features required for application execution.

In knowledge-based systems, the purpose of the inference engine is to simulate a human inference from a set of facts and a set of rules. Its principle is execute cycles with following phases:

• **Filtering** determines the set of rules which can be triggered.

• **Selection** select a rule to infer among the triggerable rules. For this strategy of conflicts resolution, heuristics and meta-rules are used.

• **Run** The inference engine executes selected rule to produce a new fact.

• **Update** the new fact is added to the facts list and the executed rule is removed from the set of triggerable rules.

We choose in this work to use forward chaining. The principle of forward chaining is to perform, using a state of the fact base, all logically possible deductions, until the saturation of the fact base or until a goal is reached.

5. Demonstartion of RAMOLI

We describe in this section the user interfaces and the working of the KBS shell.

5.1 Projects management

RAMOLI is based on the notion of project. When a user wants to create a KBS, he creates a project with the ability to save and reuse it. Our system stores the data for each project in a binary file. This gives the possibility to the user to move the project to any computer. The projects management menu is given in Figure 3. When the item new is selected, a window will open allowing to choose the name and path of the new project (Figure 4).

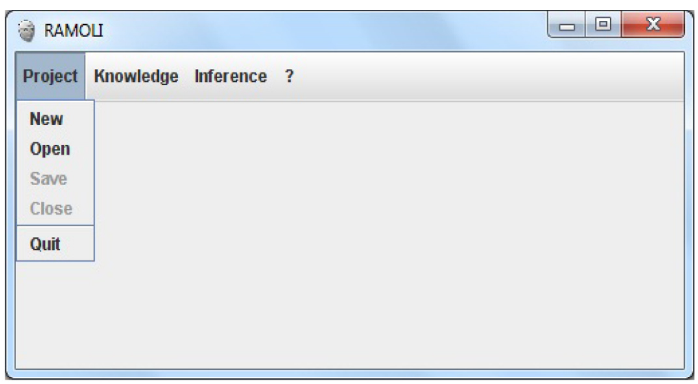

Figure 3. Menu of projects management

5.2 Knowledge management

The second menu allows knowledge management, i.e. enables the user to add facts and rules. In symbolic multivalued logic, a proposition is of the form: "X is $\upsilon_\alpha A$". In this expression, X is a linguistic variable. υ_α represents the linguistic expression associated to a symbolic degree τ_α belonging to a degrees scale $\mathcal{L}_M$. It expresses the membership degree of the linguistic variable X to the multi-set A. For example in the proposition "*tomato is very red* ", the term "*very*" refers to the linguistic degree to which the symbolic variable "*tomato*" satisfies the multi-set "*red* ".

Therefore, before adding facts and rules, the user must provide: linguistic variables, predicates and their scales bases that will be used in the representation of knowledge. Figure 5 shows menu items of knowledge management.

5.2.1 Scales bases management

The first step in representing multi-valued knowledge is to specify the symbolic scales bases $\mathcal{L}_M$. It is possible to work with several bases in a single project, however each base must have a unique name as an identifier. The user must also specify the

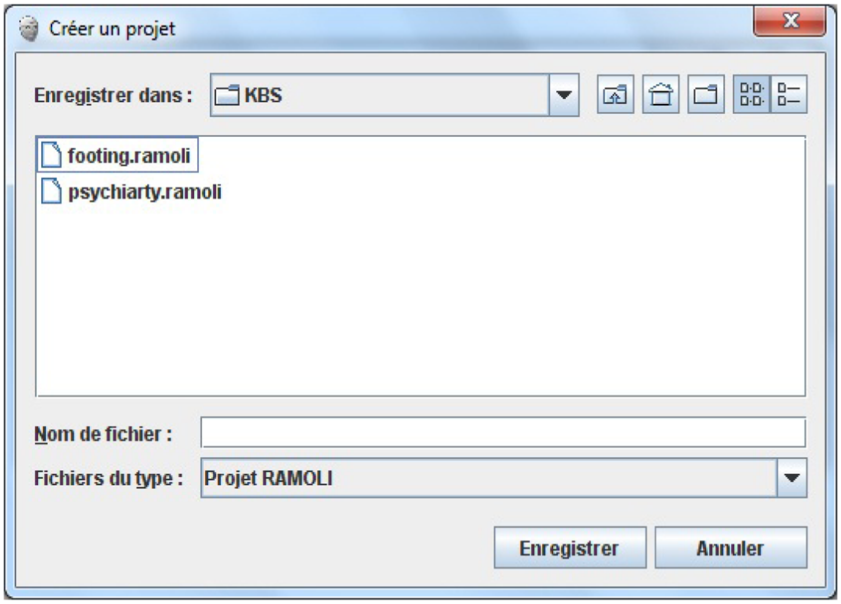

Figure 4. Dialog of new project

base size, i.e. the number of symbolic degrees of this base. He can provide a scale of corresponding linguistic terms. If not, thesystem will automatically name the degrees of the base by υ_i ($i \in [0, M-1]$). When creating a new project, the system automatically inserts some bases: $\mathcal{L}_2$, $\mathcal{L}_3$, $\mathcal{L}_5$, $\mathcal{L}_7$ and $\mathcal{L}_9$. The window of scales bases management is given in Figure 6.

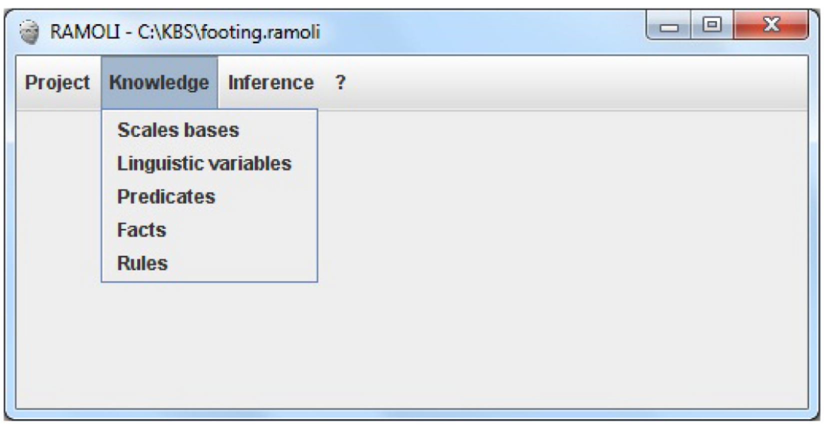

Figure 5. Menu of knowledge management

5.2.2 Linguistic variables management

This functionality allows managing linguistic variables that will be used in the KBS. A linguistic variable is characterized and identified by its name. Figure 7 shows dedicated window to the management of linguistic variables.

5.2.3 Predicates management

Predicates management allows specifying predicates that will be used in knowledge management. For each new predicate, it is necessary to assign a name and a scale base among the scales bases already stored in the project (by the menu item of scales bases management). Window of predicates management is shown in Figure 8.

5.2.4 Facts management

After adding scales bases, linguistic variables and predicates, it is possible to add facts and rules in the knowledge base. The lists of linguistic variables, predicates and their associate degrees are automatically generated in the dropdown lists. Thus, the user has only to choose the elements needed to build his fact. Figure 9 presents the window of facts management.

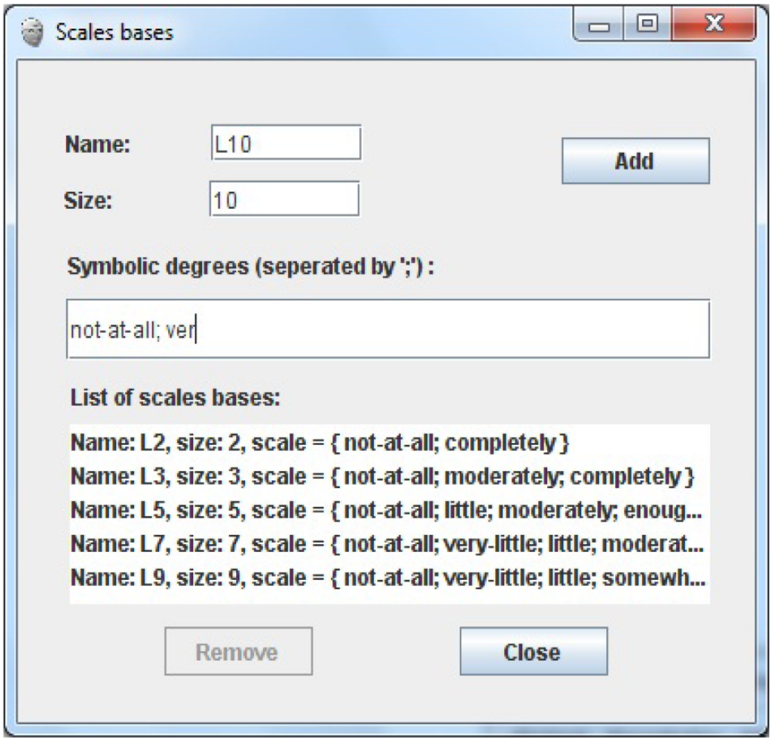

Figure 6. Window of scales bases management

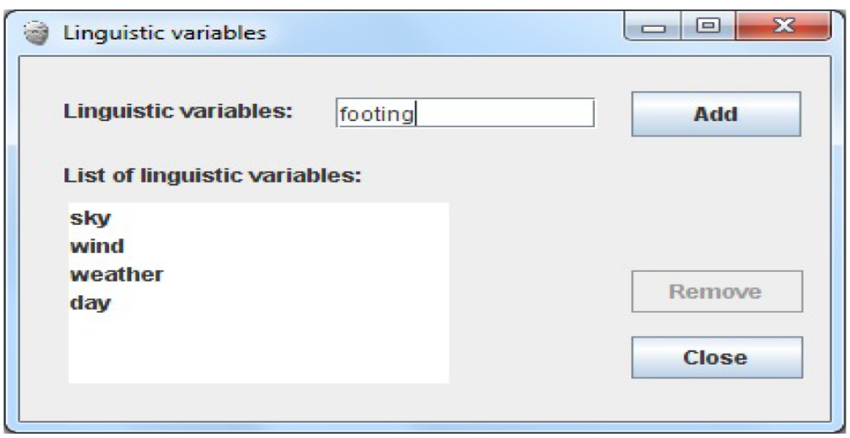

Figure 7. Window of linguistic variables management

5.2.5 Rules management

This module allows to manage rules base. Two types of rules are considered in this system: conjunctive rules and disjunctive rules. A conjunctive rule is of the form:

If X_1 is $\upsilon_{\alpha_1} A_1$ and ... and X_n is $\upsilon_{\alpha_n} A_n$ then Y is $\upsilon_\beta B$ and a disjunctive rule is of the form:

If X_1 is $\upsilon_{\alpha_1} A_1$ or ... or X_n is $\upsilon_{\alpha_n} A_n$ then Y is υ_β B with X_i (for i from 1 to n) and Y linguistic variables, A_i and B predicates and υ_{α_i} and υ_β symbolic degrees belonging to the scales bases of A_i and B.

The representation of a rule includes two steps: the construction of its premise and the construction of its conclusion. While the premise can be multiple, it is necessary to specify the propositions constituting it one by one. This is facilitated by dropdown lists. Moreover, the user must also choose their connector (conjunction AND or disjunction OR). Finally, it remains to construct the conclusion of the rule, this is also done by dropdown lists. Figure 10 shows the rules management window.

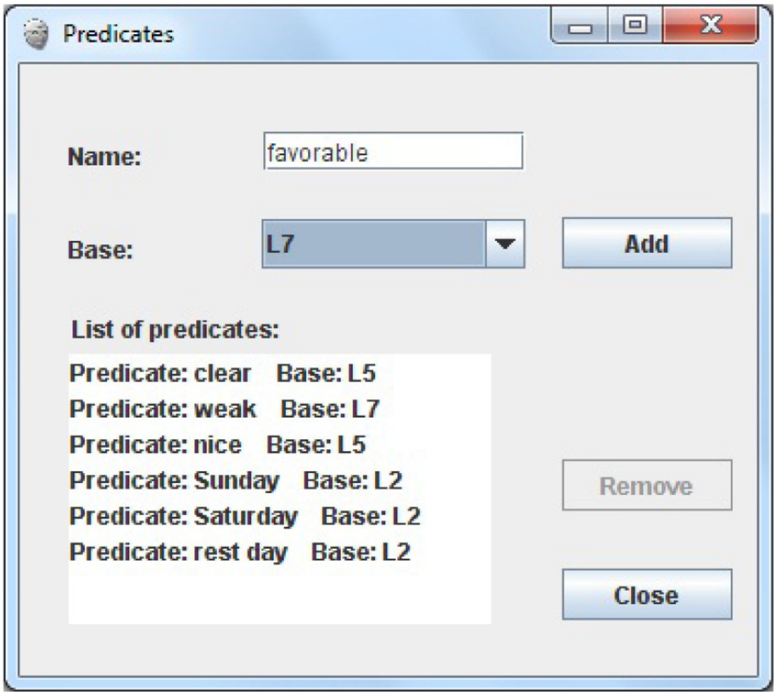

Figure 8. Window of predicates management

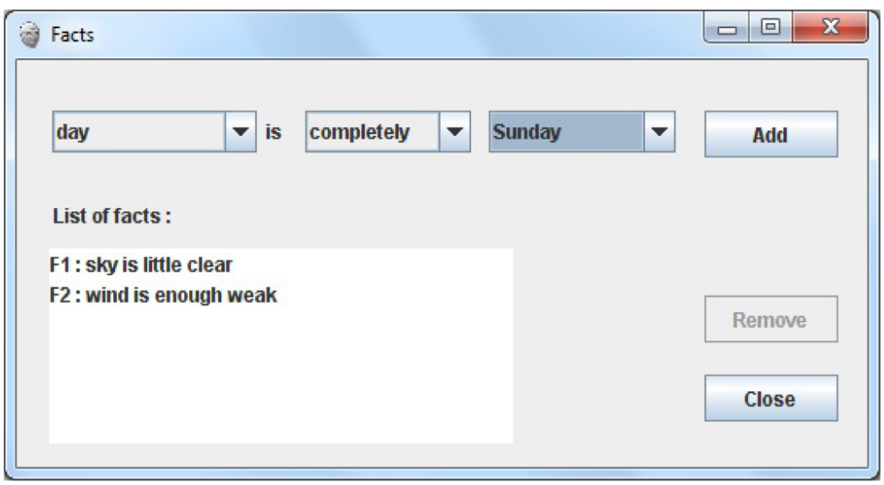

Figure 9. Window of facts management

5.3 Inference engine

The inference menu includes the functionality of knowledge processing, namely the implementation of reasoning to infer new facts. Two types of inference are available: exact inference and approximate inference. Figure 11 shows the different modules of the menu.

5.3.1 Classical reasoning

By choosing the classical inference, the system performs a classical forward chaining, it is based on exact Modus Ponens. After the execution, new facts are deducted and added to the knowledge base. Figure 12 shows a result example.

5.3.2 Approximate reasoning

This feature allows a forward chaining, but here with Generalized Modus Ponens. This inference uses our approximate reasoning based on linguistic modifiers. After the inference, new facts can be added to the knowledge base. Some others are updated. Figure 13 shows an example of results obtained by approximate inference.

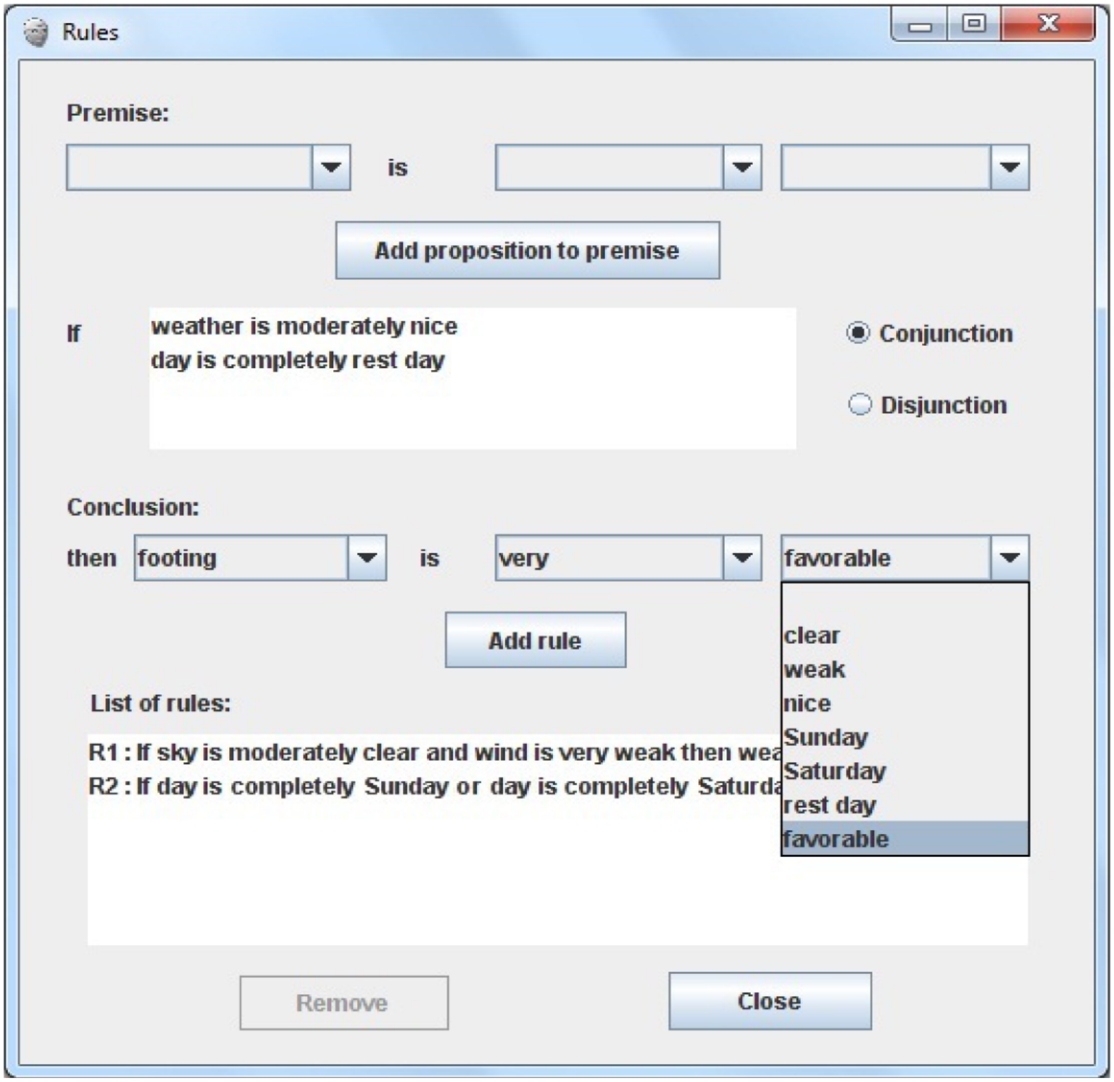

Figure 10. Window of rules management

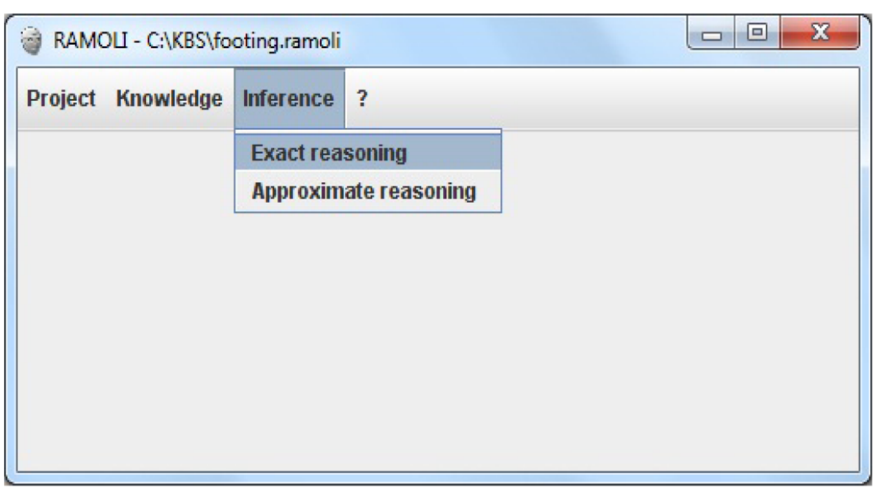

Figure 11. Menu of Inference engine

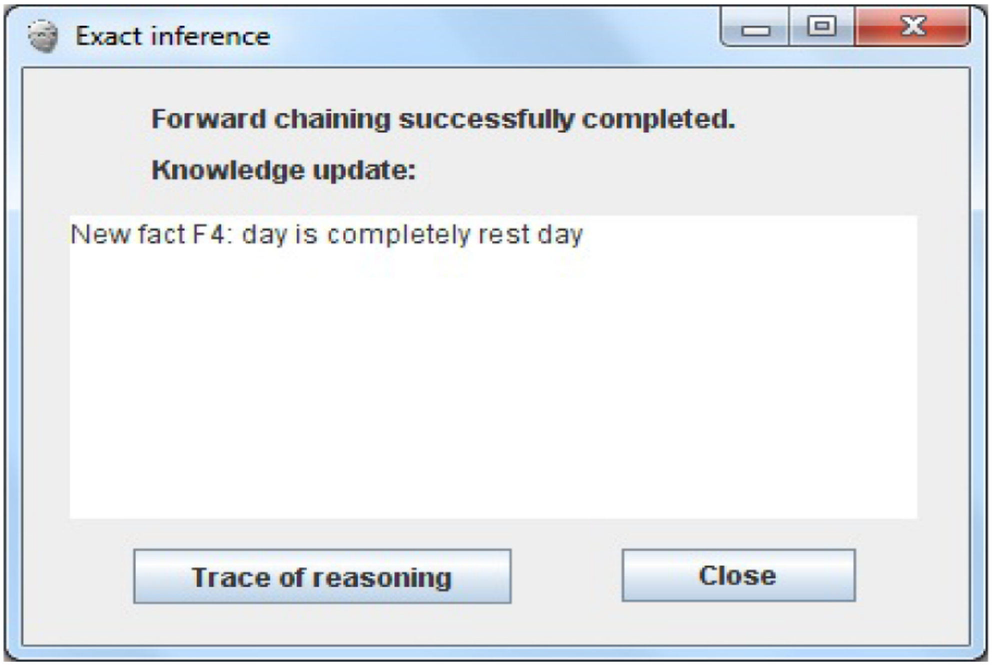

Figure 12. Window of classical inference result

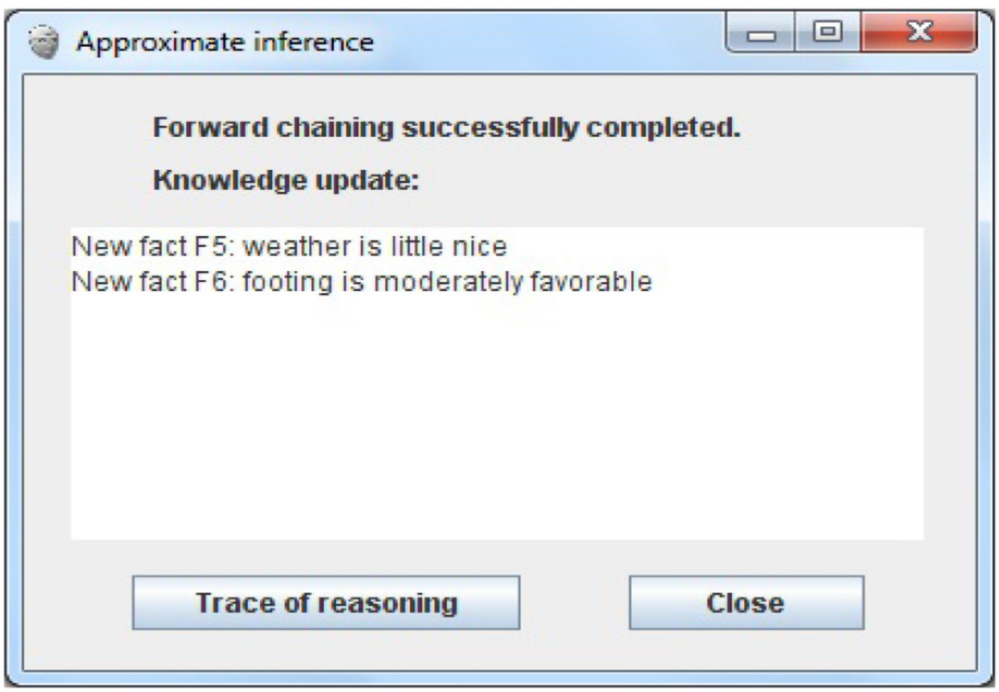

Figure 13. Window of approximate inference result

6. Conclusion

We have presented in this paper a Knowledge Based System shell for symbolic data called RAMOLI. This system provides GUIs for manipulating multi-valued knowledge base. It also includes a multi-valued inference engine that implements exact reasoning as well as approximate reasoning. The used approximate reasoning in RAMOLI is the one that we have proposed in previous works [15] [19]. In order to use RAMOLI in a practical application, it is necessary to have a knowledge base of a specific domain, with is constructed by knowledge acquisition phase. We are in the process of developing a KBS using RAMOLI for psychiatric diagnosis.

References

[1] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8 (3) 338–353.

[2] De Glas, M. (1989). Knowledge representation in a fuzzy setting. report 89-48, LAFORIA, University of Paris VI.

[3] Pacholczyk, D. (1992). Contribution au traitement logico-symbolique de la connaissance. PhD thesis, University of Paris VI.

[4] Chung, H. T., Schwartz, D. G. (1995). A resolution-based system for symbolic approximate reasoning. *Int. J. Approx. Reasoning* 13 (3) 201–246.

[5] Akdag, H., De Glas, M., Pacholczyk, D. (1992). A qualitative theory of uncertainty. *Fundam. Inform*. 17 (4) 333–362.

[6] Schaffer, J., McGee, D. (1997). Knowledge Based Programming For Music Research. The Computer Music and Digital Audio Series. A-R Editions.

[7] Tasso, C., Oliveira, E. (1998). Development of Knowledge-Based Systems for Engineering. Cism Courses and Lectures. Springer.

[8] Kendal, S. S. L., Creen, M. (2007). An Introduction to Knowledge Engineering. Springer-Verlag London.

[9] Akerkar, R., Sajja, P. (2009). Knowledge-Based Systems. 1st edn. Jones and Bartlett Publishers, Inc., USA.

[10] Pan, J., DeSouza, G. N., , Kak, A.C. (1998). Fuzzyshell: a large-scale expert system shell using fuzzy logic for uncertainty reasoning. *IEEE Transactions on Fuzzy Systems*, 8 (6) 563–581.

[11] Orchard, B. (2004). Fuzzyclips user’s guide. National Research Council Canada.

[12] Orchard, R. (2001). Fuzzy reasoning in jess: The fuzzyj toolkit and fuzzyjess. *In*: ICEIS. 533–542.

[13] Guadarrama, S., MuÜnoz-Hern´andez, S., Vaucheret, C. (2004). Fuzzy prolog: a new approach using soft constraints propagation. *Fuzzy Sets and Systems*, 144 (1) 127–150.

[14] Khoukhi, F. (1996). Approche logico-symbolique dans le traitement des connaissances incertaines et impr´ecises dans les systemes a base de connaissances. PhD thesis, Université de Reims, France.

[15] Kacem, S. B. H., Borgi, A., Ghedira, K. (2008). Generalized modus ponens based on linguistic modifiers in a symbolic multi-valued framework. *In*: Proceeding of the 38th IEEE International Symposium on Multiple- Valued Logic, Dallas, USA,150–155.

[16] Baldwin, J., Pilsworth, B. (1980). Axiomatic approach to implication for approximate reasoning with fuzzy logic. Fuzzy Sets and Systems 3 (2) 193–219.

[17] Fukami, S., Mizumoto, M., Tanaka, K. (1980). Some considerations of fuzzy conditional inference. *Fuzzy Sets and Systems*, 4 (3) 243–273.

[18] Borgi, A., Kacem, S. B. H., Gh´edira, K. (2008). Approximate reasoning in a symbolic multi-valued framework. *In*: Computer and Information Science. V. 131 of Studies in Computational Intelligence. Springer, 203–217.

[19] Kacem, S. B. H., Borgi, A., Tagina, M. (2009). On some properties of generalized symbolic modifiers and their role in symbolic approximate reasoning. *In*: ICIC (2). V. 5755 of Lecture Notes in Computer Science., Springer 190–208.

[20] Kacem, S. B. H., Borgi, A., Tagina, M. (2013). Extended symbolic approximate reasoning based on linguistic modifiers. To be published in *Knowledge and Information Systems Journal*.

[21] Ginsberg, M. L. (1988). Multivalued logics: a uniform approach to reasoning in artificial intelligence. Computational Intelligence 4 (3) 265–316.

[22] Bartusek, T., Navara, M. (2001). Conjunctions of many-valued criteria. *In*: Proceedings of the International Conference Uncertainty Modelling, Bratislava, Slovakia, 67–77.

[23] Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning - i - ii - iii. *Information Sciences* 8:199–249, 8:301–357, 9:43–80.

[24] Akdag, H., Truck, I., Borgi, A., Mellouli, N. (2001). Linguistic modifiers in a symbolic framework. *International Journal of Uncertainty*, *Fuzziness and Knowledge-Based Systems*, 9 (Supplement) 49–61.