A Combined Use between Rules, Ontology and Agents in BRMS Design: Application to SME in Algeria

Nawal Sad houari, and Noria Taghezout

***Abstract*—**The goal of this paper is to present an approach, based on a domain ontology and a multi-agents system, permitting to manage automatically the consistency of the business rules within a typical Small and Medium Enterprise.

Our approach permits to capitalize the business experts knowledge as business rules by using an agent-based collaborative platform. This latter offers facilities to the experts as standardization and auditability of the business rules. We propose to use a domain ontology in order to generate the business model corresponding to the enterprise.

***Keywords*—**Business rules, Business Rules Management System, Encryption, Inference engine, Multi-Agent Systems, Ontologies.

I. Introduction

FROM a historical point of view, the rule-based systems, or Business Rules Management System, are descended from the artificial intelligence and in particular from expert systems. The aim of expert systems is to reproduce the reasoning of an expert: such a system is used as a decision support system.

In computing science, a business rule is a high-level description that allows controlling and / or taking a decision, using enterprise specific concepts. Thus, the business rules describes what an expert needs to do to take a decision [1]. They capitalize the knowledge of an enterprise and translate its strategy by describing the actions to lead for a given process. They are generally written in a controlled natural language [2].

These rules can be defined in the form of simple rules (as IF <conditions> THEN <Actions>), decision tables, or decision trees.

These business rules should be maintained and modified easily to allow the enterprise to evolve according to its market and to take operational and strategic decisions that are required. Or the collection and integration of these rules to the information system of the enterprise remain a particularly difficult task.

A business rules management system establishes collaboration between business and IT. A BRMS is a software that manages and supports the business rules of an organization or an enterprise. The objective is to have clear rules, which means unambiguous and comprehensible by all, especially by non-IT. The method is to separate the business logic (rules) of the logic system (programs, development languages, databases, operating systems) from an application. So business logic can evolve separately from the application code.

The BRMS solutions automate operational decisions in business applications. They reduce the maintenance costs of these applications and improve the collaboration between business and IT teams. A BRMS allows:

1) To develop rules without programming,
2) To the business experts to participate directly in the management of rules,
3) To automate the decisions and manage frequent modifications of the rules,
4) A reduction of the development costs,
5) An improvement of the capacity to react and to adapt facing the changes.

The article is organized as follows, Section 2 presents some related works. We describe in Section 3 our approach in details. We illustrate our approach by an example in the section 4. Finally the section 5 presents our perspectives and concludes.

II. Related works

The business rules management systems are promising techniques to facilitate edition, creation, modification, and management of the business rules. They also, permit to improve the decision making in the organizations. However, in the literature, several works exist; we give in he following the most important:

A. Work N° 1

The work presented in [5] proposes a conception and an implementation of a multi-agents system that coordinate an expert system and a neural network to construct production orders to produce labels. The architecture of this model is presented in the figure 1.

Nawal Sad Houari, LIO laboratory, University of Oran, Algeria (E-mail ID - sad.houari.nawal@gmail.com).

Noria Taghezout, LIO laboratory, University of Oran, Algeria (E-mail ID - Taghezout.nora@gmail.com).

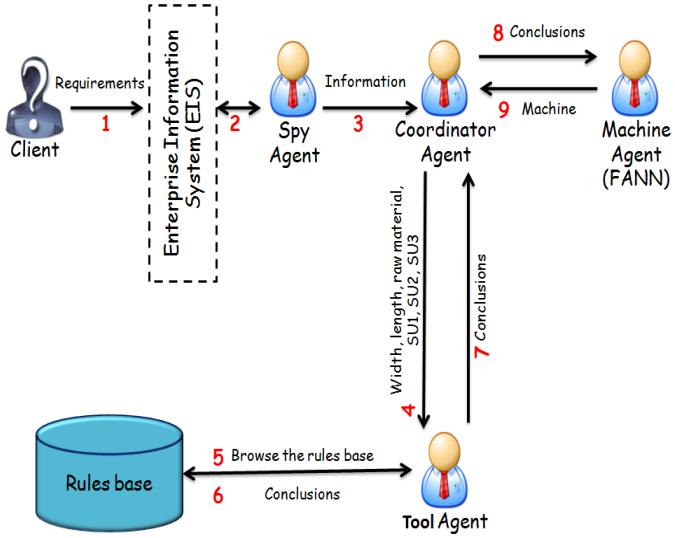


Fig. 1 Global architecture of the system [5]

The fundamental goal of the multi-agent System (MAS) is to construct a manufacture order. Authors proposed to decompose the decision process carried by the human responsible in a set of tasks that are executed by the software agents. The JADE platform [8] was used for the interaction of the agents. The MAS possesses the following agents:

1) The tool agent: it possesses a system of rules to determine the good tooling.
2) The machine agent: incorporates a neural network, and detect the suitable machine.
3) The coordinator agent: is responsible to maintain the consistency of the data during the process by the control of messages.
4) The spy agent: must read the Enterprise Information System to acquire the orders of the customers, and send information to the coordinator.
5) The scheduler agent: as soon as the machine, the tools and the other data are established, it assigns a priority to the manufacture order.

The scheduler agent employs a FIFO politics, even though it is the simplest ordering/queuing mechanism, therefore it is necessary to develop a strong scheduling politics to provide more realistic plans.

B. Work N° 2

A methodology is proposed in [6] to help the business experts and the developers to keep the business rules at the business level in line with the rules that are implemented at the system level.

In order to manage the business rules for an entire organization and to establish the link between its business and the Information System (IS), several activities are required. Besides the activities that are dedicated to managing rules during IS development, there are also the activities that have to be performed at the business level.

At the business level, the most important activity that is required for BRM (Business Rule Management) is to identify and document the elements that may act as a source, motivation, or explanation for business rules. There are several such elements: business goals, problems, policy, regulations, business processes, etc. In the BRME project (Business Rule Management in Enterprises), enterprise modelling (EM) was identified as a promising technique that may lead into formalisation of business environment to the extent required for BRM.

As shown in Figure 2, the approach recognizes five sub-models:

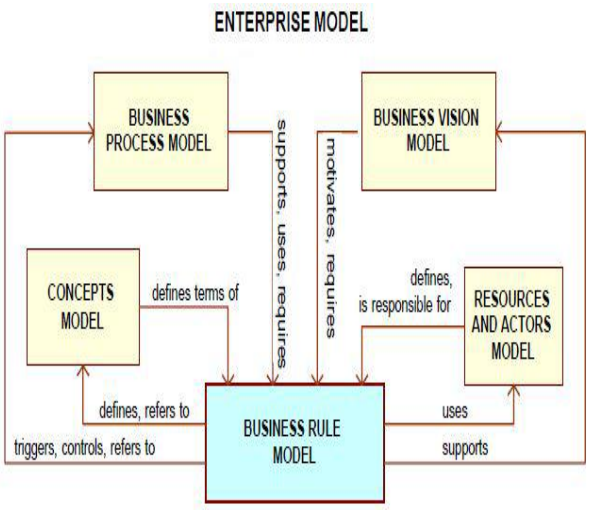


Fig. 2 Enterprise model and its sub-models [6]

C. Work N° 3

The article presented in [7] describes an approach that permits to manage the consistency of the business rules published from OWL ontologies at the time of their evolutions. Two methods are presented; the first permits the edition of the business rules, in controlled natural language, from OWL ontologies and the second permits to manage the consistency of these rules at the time of the evolution of the corresponding ontology.

For the first method, in order to edit the business rules, they used the business rules management system "IBM WebSphere ILOG JRules". JRules offers an infrastructure to publish the business rules, in controlled natural language. The idea is to exploit this infrastructure and to adapt it to the OWL ontologies. This adaptation consists in doing an automatic transformation of the OWL model toward the BOM of JRules. It permits to import ontologies written in OWL and to generate the corresponding BOM for every ontology. Once the BOM generated, all functionalities of JRules, as the edition and the execution of the rules, can be used without any change. To do the transformation of the OWL model toward the BOM, the authors took the method described by Kalyanpur and al, 2004 as a basis with some changes according to their need.

In the second method, the approach adopted for the management of the consistency of a set of rules at the time of the evolution of the corresponding ontology permits the management of the ontology's consistency at the time of its evolution. They used an approach based on PATRONS named PATRONS of Management of Change who consist in three

categories of PATRONS:

1) PATRONS of change: permit to modeling the changes;
2) PATRONS of incoherence: permit to detect the incoherence caused by a change;
3) PATRONS of repair: permit to repair the incoherence.

III. Proposed Approach

Our approach permits to capitalize the business experts knowledge as business rules by using an agent-based collaborative platform. This latter offers facilities to the experts as standardization and auditability of the business rules. We propose to use a domain ontology in order to generate the business model corresponding to the enterprise.

The business rules management system (BRMS) is composed of several components as described in Fig. 3.

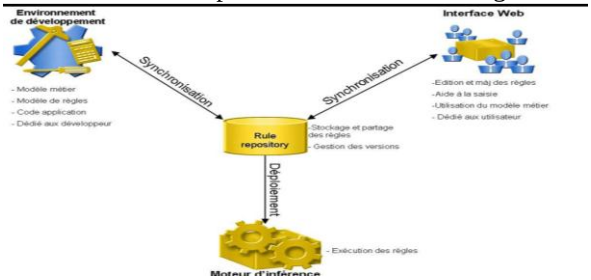


Fig. 3 The classical architecture of a BRMS [7]

A. Development Environment

The development environment encourages the collaboration between the developers and the business experts. This step consists in defining the business model and the rules model as well as some functions necessary to the development of the application.

1) Definition of the business language: In order to allow the business experts to implement and maintain up to date their rules-based systems, it is necessary to provide a language of rules intuitive and easy to manipulate that uses the vocabulary of the business. A business language must permit to represent the entities of the business domain therefore, the actions to lead and the strategies to follow. For that to make, our BRMS found on domain ontology, representing all entities of the business as well as their properties and relations. To construct our ontology, we are going to be based on the collection and the capitalization of the processes business rules with the experts, the UML diagram that represents the enterprise, the questionnaires, and the interviews with the directors of the enterprise.
2) Definition of the rules model: Several types of rules exist: The IF - THEN format, table of decision, and tree of decision. In our business rules management system, we use the standard format IF <Conditions> Then <Actions>, because it is the format that offers more possibility to describe the conditions and the complex actions. Nevertheless, the suppleness and the flexibility of this format can entail the mistakes that return the system unusable, therefore to avoid this type of problem, we provided a help to the seizure to guide the experts at the time of the rules edition, so another solution to avoid these mistakes would be to define models of rules that permit to freeze some parts.

B. Web Interface

This component is dedicated to the business experts in order to create and to update their business rules. The interface must be convivial and most ergonomic possible to provide the different functionalities to the business experts.

1) Rules Edition: Thanks to the generated business language, the business expert can write the rules in an autonomous manner. A rule is composed of a condition part and an action part; therefore the expert must specify the two parts. The process needs to pass through several steps, until the final storage in the rules base. We can summarize these steps in the following:

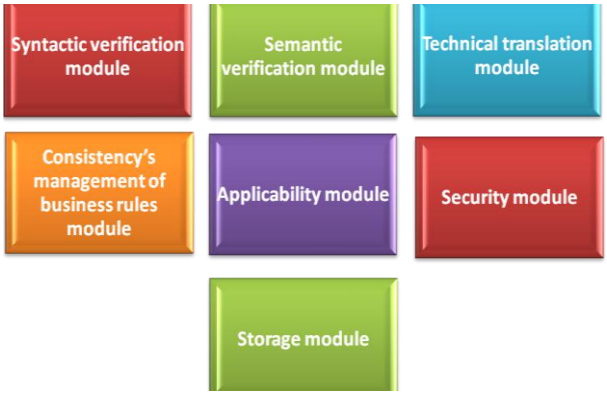


Fig. 4 The different services to add a rule

a. Syntactic verification module: this module provides a syntactic analysis to the experts in order to avoid the mistakes that return the system unusable, for example:

> **R1:** If an employee's salary exceeds the salary of his director then mark this employee as having a special status ➔rule syntactically correct.
>
> **R2:** If an employee's salary exceeds the salary of his director then mark this employee as having a special status ➔rule syntactically incorrect.

To solve this problem, our system handles these errors at the beginning.

b. Semantic verification module: This service is composed of two sub-modules, which are :

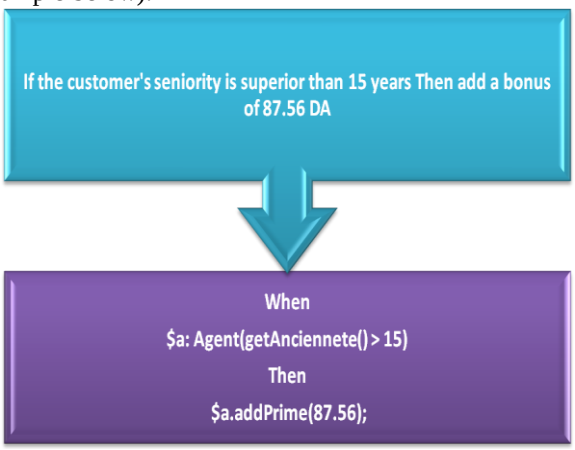
The synonym verification: by using the domain ontology to translate the business rule, for example:

R1: If an employee's salary exceeds the salary of his director then mark this employee as having a special status.

R2: If an employee's salary is superior of his director's salary then the status of the employee is special.

The verification of the rule's validity: this module detects if the rule is valid or not.

c. Technical translation module: Once introduced by the expert, the rule must be translated into technical rule capable to be executed by the inference engine (see the example below).



d. Consistency management module: the BRMS will provide a consistency management of the rules to avoid some ambiguity. In our case, we treat contradiction, redundancy, equivalence, invalid rules (in the semantic verification module) and rules ever applicable.

e. Security module: the business rules must be accessible that for the company's business experts. However, the access to business rules must be highly secure. The establishment of a security system requires considering the following issues:

Authentication: the identification of a given expert is through an authentication process. The technological evolution offers a variety of tools (code PIN, login, banking card, badge, fingerprint, retinal scan, vocal recognition).

Encryption: consists to transform a business rule "in clear" in an indecipherable rule in order to ensure the system security. The inverse operation is decryption.

f. Applicability module: a business rule can be: syntactically correct, semantically correct and consistent, but it is not exact, so to solve this problem of accurateness we send the rule to a meta-expert to judge if it is correct or not.

g. Storage module: if the meta-expert judges that the business rule is correct, then it will be stored in the final rule base.

2) Multi agents system: To achieve all functionalities, we use a multi agents system that is composed of five agents:

- Expert Agent: is responsible for the recuperation of the rules seized by the expert. This agent saves the rules and transmits them to Translator agent.
- Supervisor Agent: performs all control tasks in the system.
- Translator Agent: is the heart of our system, it retrieves the rule from the Expert agent, browses the domain ontology and extracts the set of concepts that correspond to the introduced rule. Finally, it sends the technical rule to Evaluator agent.
- Evaluator Agent: is responsible to assessing the consistency of the business rules. It recovers the rule translated from the Translator agent, and accesses to the rules repository to test if this rule poses a problem with another rule, if it is the case then the Evaluator agent send a message to the Expert agent, otherwise it validates the rule.
- Security Agent: is responsible to encrypting and decrypting the business rule.

The interaction among the different agents of the system is shown by the diagram of sequence represented in the Fig. 5.

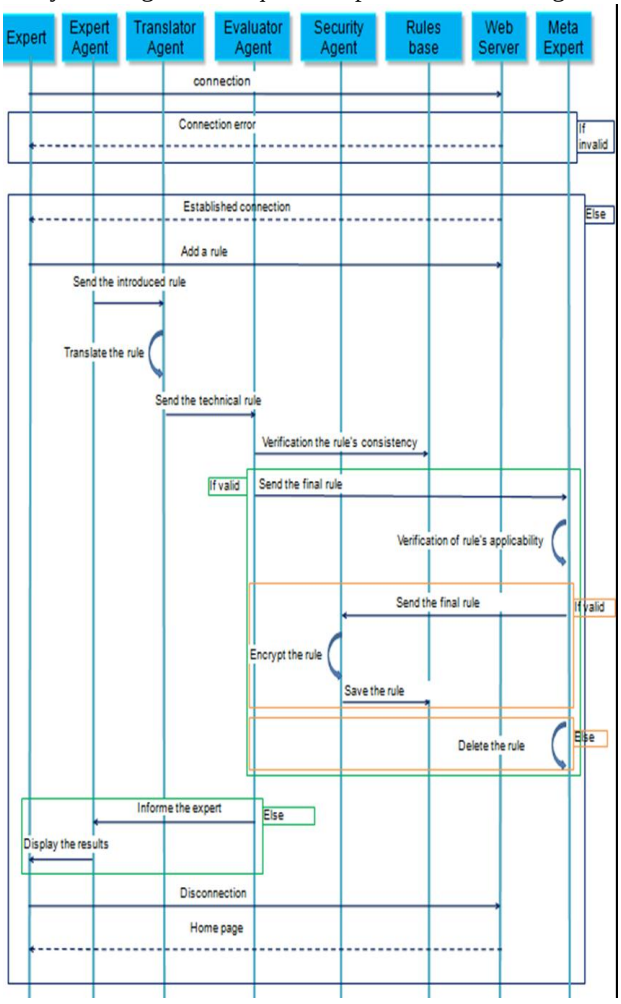


Fig. 5 Sequence diagram of our system

Figure 6 gives an overview of the agent-based architecture of the system. Once the expert authenticates himself, he access to the BRMS editor, if he wants to introduce a rule, the Expert agent saves the rule and contact the Supervisor agent to know if the Translator agent exists, if this last doesn't exist the Supervisor agent generates him and answers the Expert agent.

After the Expert agent receives the answer from the Supervisor agent, it sends the rule introduced by the expert to the Translator agent. The Translator agent harvests the data on behalf of the Expert agent and reaches to the ontology to translate the rule, after the translation is made, this agent contacts the Supervisor to know if the Evaluator agent exists, after receiving the answer it sends the technical rule to the Evaluator agent. The Evaluator Agent reaches the rules repository and verifies the consistency of this rule with the others, if it doesn't pose any problem therefore the rule will be sent to the meta-expert for the applicability verification, otherwise this agent sends a message to the Expert agent. Then, if the meta-expert validates the rule, this latter will be sent to the security agent to encrypt. After the step of encrypting the rule will be stored in the rule repository.

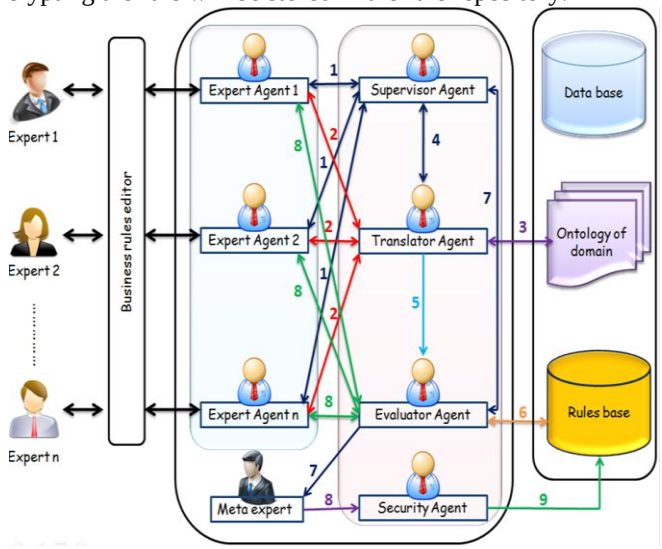


Fig. 6 An agent-based architecture

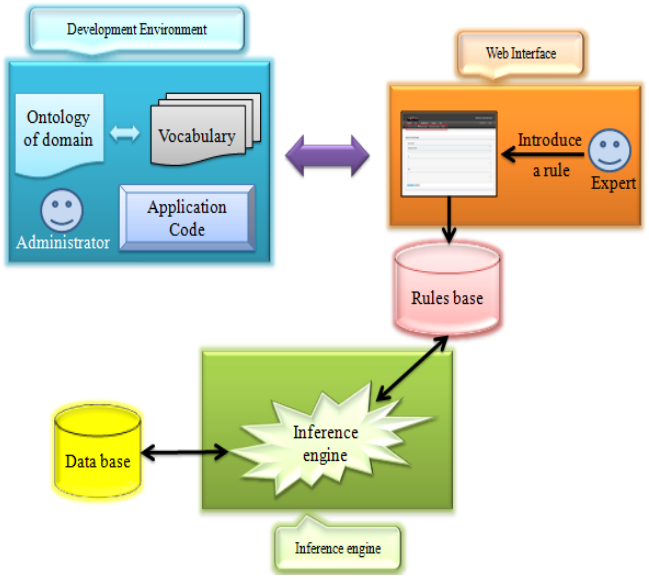


Fig. 7 The proposed system

A. Inference engine

According to the presented diagram in Figure 3, an inference engine is generally constituted of three parts; the "Rule set" component contains the set of rules represented in a technical language, the "Working memory" component contains the set of objects (named also the facts) of the application that will permit to trigger the execution of the rules and the "diary" component that contain the set of the eligible rules on a set of working memory object. The global architecture of the system is presented in the Figure 7.

IV. Implementation

In this section, we are going to mention the different tools used in the development of our business rules management system, as well as a script of execution.

We describe in this section the major tools we used in the development of our BRMS.

A. Development Tools:

1) JADE: (Java Agent DEvelopement framework) is a software framework to develop agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems [9].
2) Netbeans: To implement and develop our approach, we used Java as programming language oriented object and NetBeans 7.0 like an recent execution environment Open source. NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an application platform framework for Java desktop applications and others [10].
3) Protégé 4.0.2: is a free, open source ontology editor and a knowledge acquisition system. Protégé provides a graphic user interface to define ontologies. It also includes deductive classifiers to validate that models are consistent and to infer new information based on the analysis of ontology [11].
4) JENA API: is an open source Semantic Web framework for Java. It provides an API to extract data from and write to RDF graphs [12].
5) Apache Tomcat: is an open source web server and servlet container developed by the Apache Software Foundation (ASF) [13].
6) WampServer: is a platform for Web development under Windows for dynamic Web applications using the Apache 2 server, PHP scripting language and MySQL database [14].

B. Execution Scenario

We consider here a simple example to illustrate our approach:

Miss. Nawal is an expert in the company. She accesses to the developped platform to capitalize her knowledge and her experiences in a particular domain (see Fig. 8).

Fig. 8 Home page

She must follow some steps:

1) Introduction of rules

Miss. Nawal wants to introduce the following business rule (see Fig. 9):

Fig. 9 A business rule in a action business language to program a discount

Once the access is guaranteed (like it is shown in Fig. 10), the business rule editor will be visualized as described in Fig. 11.

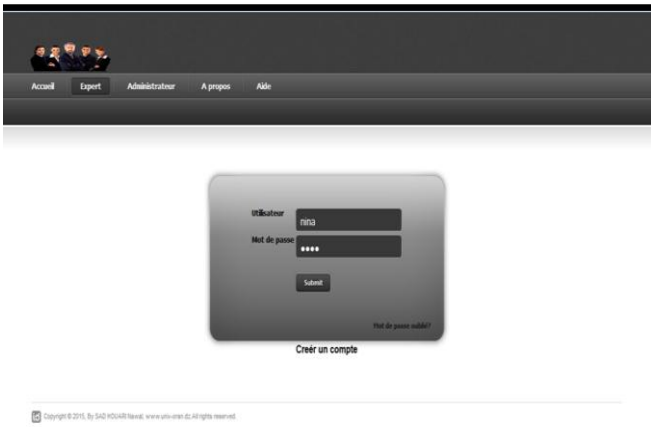


Fig. 10 Authentication of the experts

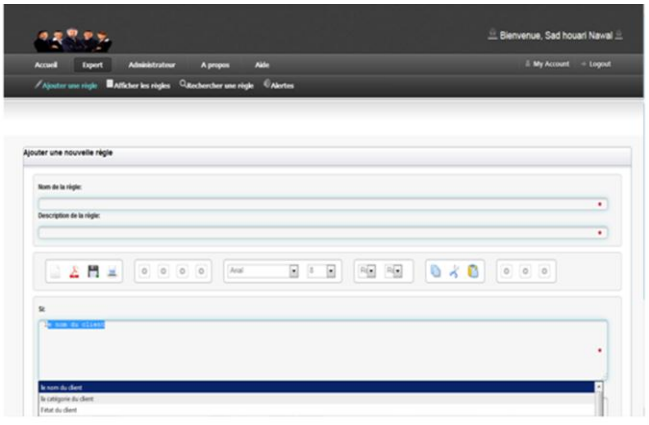


Fig. 11 Business rules editor

2) Research rules :

Miss. Nawal can search for a rule (see Fig. 12), the developped platform offers to the business experts' an advanced research, based essentially on some keywords or criteria.

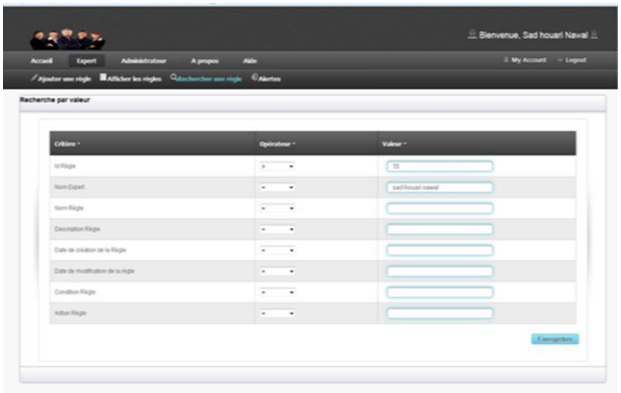


Fig. 12 Search a rule

3) Updating rules:

By using the platform, the expert will be able to update the rule a shown in Fig. 13.

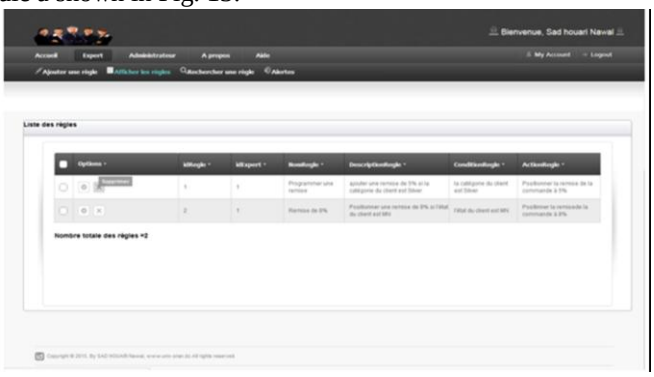


Fig. 13 Modify/delete a rule

C. Communication between agents

The addition of a rule requires a messages exchange among the different agents (Expert Agent, Translator Agent, Evaluator Agent, Supervisor Agent and Security Agent). The Sniffer Agent of JADE allows us to observe these exchanges through its graphical interface as it appears in Fig. 14.

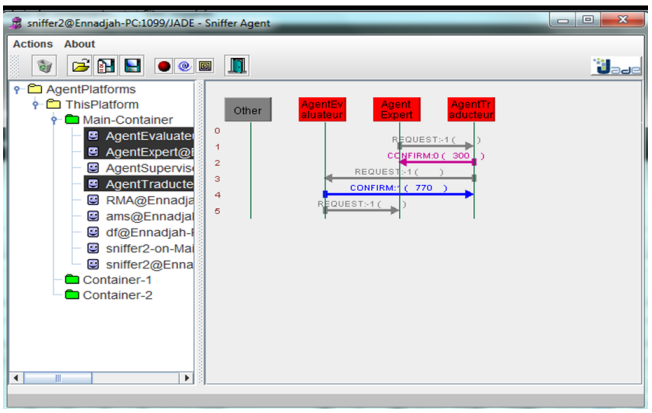


Fig. 14 Communication between agents under JADE

V. Conclusion

This work constitutes the first developed phase of a collaborative platform integrated into the BRMS and dedicated to an expert profile in a given domain. Knowledge is represented in the form of rules and ontologies. The reasoning will be made in two manners, by inference on rules and on ontologies.

We apply our system to the non-woven industry with two Algerian experts. The modeling is based on agents to increase the execution speed of processes and effective response.

The prototype is in progress and test at the ergonomists for an ergonomic evaluation of the editor and at the company experts for the executions evaluation of the rules: the RÊTE algorithm is intended to be used to reach this goal.

References

[1] V. Legendre, G. Petitjean and T. Lepatre, Gestion des règles «métier », Génie logiciel , N° 9 2, 43–52 , 2010.
[2] A. Chniti, P. Albert and J. Charlet, Gestion de la cohérence des règles métier éditées à partir d'ontologies OWL, dans le proceding IC2011.
[3] M. Diouf, S. Maabout and K. Musumbu, Génération automatique de règles métier par enrichissement sémantique de modèles : http://liris.cnrs.fr/inforsid/sites/default/files/a661c1Ungr88gvmII.pdf
[4] J. Ghattas, P. Soffer and M. Peleg, Improving Business Process Decision Making Based on Past Experience, Decision Support Systems (2013), doi: 10.1016/j.dss.2013.10.009. http://dx.doi.org/10.1016/j.dss.2013.10.009
[5] O. Lopez-Ortega and I.l Villar-Medina, A multi-agent system to construct production orders by employing an expert system and a neural network, dans le journal Expert Systems with Applications, numéro 36, année 2009, pages 2937–2946.
[6] M. Bajec and M. Krisper, A methodology and tool support for managing business rules in organizations, dans le journal Information Systems, numéro 30, année 2005, pages 423–443.
[7] A. Chniti, P. Albert and J. Charlet, Gestion des dépendances et des interactions entre Ontologies et Règles Métier, thése de dotorat soutenue à l'Université Pierre et Marie Curie - Paris VI, année 2013.
[8] Jade : http://jade.tilab.com/.
[9] F. Bellifemine, A. Poggi and G. Rimassa, JADE – A FIPA-compliant agent framework: http://sharon.cselt.it/projects/jade/papers/PAAM.pdf
[10] http://en.wikipedia.org/wiki/NetBeans
[11] http://en.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_(software)
[12] http://en.wikipedia.org/wiki/Jena_(framework)
[13] http://en.wikipedia.org/wiki/Apache_Tomcat
[14] http://fr.wikipedia.org/wiki/WampServer