Integrating Agents into a Collaborative Knowledge-based System for Business Rules Consistency Management

Nawal SAD HOUARI and Noria TAGHEZOUT

Laboratoire d'Informatique Oran (LIO), Département d'Informatique, University of Oran1 Ahmed BenBella, BP 1524 EL Mnaouer Oran, Algeria

***Abstract* — Capitalization and reuse of expert knowledge are very important for the survival of an enterprise. This paper presents a collaborative approach that utilizes domain ontology and agents. Thanks to our knowledge formalizing process, we give to domain expert an opportunity to store different forms of retrieved knowledge from experiences, design rules, business rules, decision processes, etc. The ontology is built to support business rules management. The global architecture is mainly composed of agents such as Expert agent, Evaluator agent, Translator agent, Security agent and Supervisor agent. The Evaluator agent is at the heart of our functional architecture, its role is to detect the problems that may arise in the consistency management module and provides a solution to these problems in order to validate the accuracy of business rules. In addition, a Security agent is defined to handle both security aspects in rules modeling and multi-agent system. The proposed approach is different from the others in terms of the number of rule's inconsistencies which are detected and treated like contradiction, redundancy, invalid rules, domain violation and rules never applicable, the collaboration that is initiated among business experts and the guarantee of security of the business rules and all the agents which constitute our system. The developed collaborative system is applied in an industrial case study.**

***Keywords* — Business rules (BR), Collaboration, Consistency Management, Knowledge-based System, Multi-agents Systems (MAS), Ontologies.**

I. Introduction

Small and medium enterprises (SME) are facing a high-level market competition. Despite the modernization of manufacturing processes, the use of experts (local and foreign) is costly. It becomes crucial to invest in knowledge and knowledge management systems. Especially, when the decision-making can be improved by using knowledge-based systems as mentioned in [1].

Companies tend to pay more and more attention on the subject of knowledge, thereby moving the improvement of the business strategy from product quality to the design of systems for knowledge.

The experts' knowledge capitalization has become a major goal of companies. Consulting an expert for a simple advice or an opinion on a critical situation in an organization has become necessary to guarantee the efficiency of decision making.

As mentioned in [2], Knowledge Management (KM) is a process of managing knowledge, which aims to manage existing and acquired knowledge assents to meet needs for now and develop opportunities in the future. Nowadays, Knowledge Management has been a focus in both academic and industrial fields. Thus, in knowledge-based engineering, many approaches are used to capture and reuse knowledge [3][4].

Tacit knowledge (experiences, competencies, etc.) usually resides in the people's brain, the effective and efficient way to utilize this type of knowledge is consulting the expert. However, domain experts could be unavailable when they are needed to participate in knowledge management system. This can be essentially due to the complexity of expertise and knowledge needs [5]. It has been well acknowledged that the difficulty of knowledge sharing lies in the sharing of tacit knowledge, especially when decision-makers come from very different backgrounds. Furthermore, they sometimes face confusing terms (such as communities, groupware), knowledge management and knowledge network are being used simultaneously [6].

Senior managers of SME enterprise are investing massively in the conception and implementation of knowledge bases to improve business processes, management and sharing of knowledge and retaining expertise, even after the employees' departure from the organization [7].

With regard of new technologies, the formal representation of knowledge has evolved with the tools of artificial intelligence. In a formal representation, knowledge is represented by logical objects linked by properties, axioms and rules.

As described in the literature, agent paradigm can be used in order to effectively model the dynamic aspect of the organizational environment. Agents are, by definition, autonomous entities, proactive and capable of social interaction in dynamic environments [8]. Additionally, they are thus capable of handling this issue when they are considered as components of agents groups with the given possibility to interact together (cooperation, collaboration) for the purpose of achieving their common goals.

We get inspired by the definition given in [9], the main idea is that the concept of knowledge can be included within the agents organizations, this will give more power to the proposed model : ''An organization of agents is a community of knowledge sharing in which agents collaborate and exchange knowledge to carry out their activities.''

Our paper aims to transfer knowledge from SMEs' experts to formal representations, which allow systems to reason with such knowledge. The main idea behind this study is to join the agent-based modeling and ontology-based approach, in order to take benefit of the advantages of the both.

The major objective is to automatically manage the consistency of business rules introduced by the experts during the capitalization of the business rules process as part of a collaborative system. Our proposal is mainly based on agents such as Expert agent, Evaluator agent, Translator agent, Supervisor agent and Security agent.

The remainder of this paper is organized as follows. Following the introduction, a state of art highlighting the problems associated with knowledge management and security is proposed in Section 2. In this section, we also describe some related works and our contribution.

DOI: 10.9781/ijimai.2016.4210

Section 3 presents our system architecture. To illustrate the feasibility of our proposed approach, the experimentations results are given in Section 4. Finally, Section 5 provides the conclusion of this paper, including possible limitations of our approach and potential direction for future research.

II. Background

Generally speaking, the business rules management has given birth to the Business Rules Management Systems technology or BRMS, which has rapidly become the best solution to the problem of effectively maintaining business rules. In this section, we will introduce the foundational concepts and briefly review related works on the domains that influence our work: Business rules management systems, the agent-based modeling and security techniques.

A. Literature review on Business Rules Management Systems

The work presented in [10] tackles the rule acquisition problem, which is crucial and critical for the development of BRMS. The proposed approach assumes that regulations written in natural language are an important and essential source of knowledge, but turning them into formal statements is a complex task that cannot be completely automated. The authors propose to decompose the acquisition process into two principal phases: the translation of natural language statements into controlled language and their formalization into an operational rule base. The authors focus on the normalization phase. As a limitation of this work, the rules acquisition from natural language is a complex task and cannot enumerate all the possible cases.

The business rules and procedures are usually provided in the text, the proposed method in [11] allows the acquisition of business rules from texts. Thus, they construct and operate a documented rules model, an "index" structure which connects the source text, the ontology that defines the conceptual domain vocabulary and the rules drawn from the text.

The paper in [12] describes a simple formalism designed to encode lexicalized ontologies and shows how it is used in a business rule management platform of the automotive domain.

In [13][14], the authors show two prototypes based on the BRMS WODM (WebSphere Operational Decision Management). The first prototype allows the creation and execution of business rules over OWL ontologies. The second prototype detects the inconsistencies that may be caused by the ontology evolution and proposes solutions to solve them.

The aim of the paper presented in [15] was to indicate possible applications of rule-based approach in production planning and scheduling. Besides, many solutions described in the study can be implemented also in the computer decision support systems for iron cast manufacturers.

B. Motivation of using agent-based modeling in support of knowledge management

We behold the continuous incorporation of agent technology into knowledge management systems. We believe that there are a lot of related works on integrating agent technology into knowledge management systems. For example, the research reported in [16] is aimed at conceiving an adaptable data-sourcing service in order to deploy business rules effectively in supply chain management. The authors propose an agent-based mechanism that dynamically maps business terms in business rules to the data objects in the enterprise data model.

A knowledge management platform for marketing decision making is proposed in [17]. The authors utilize an agent technology with fuzzy logic and fuzzy Analytical Hierarchy Process (AHP).

The author in [18] proposes a heterogeneous and distributed knowledge management system, called OCEAN that is based on ontologies and multi-agents system. The OCEAN system is based on a knowledge life cycle composed of four steps that are: identification, extraction, validation and knowledge reuse. Each step is the goal of an organization of agents.

In addition, another work has been done in this area, it consists of a methodology named DOCK, Girondo et al. [19] have designed an intelligent knowledge-based system in order to support the knowledge management process. The main objective of the MAS is to support the decision-making process within design projects and allow engineers to capitalize, share and reuse the knowledge generated throughout decision-making steps and more generally throughout design projects in order to gain in efficiency.

As mentioned in [20], Multi-agent systems can be considered as a good solution because of its distributed nature, autonomy and solidarity. Moreover, they ensure the coordination and cooperation between agents which have some capabilities of reasoning and facilitating the achievement of overall objectives. Multi-agent systems are considered as autonomous platforms that perform their actions to achieve a predefined set of objectives. They are able to provide an infrastructure that facilitates the construction of complex and composite tasks. More precisely, the work tackled the problem of Globus Resource Allocation Manager (GRAM) and proposed a model on the basis of a multi-agent system between the user and grid client. Furthermore, they used the hidden Markov models for the matchmaker process and Telecom Italia Lab approach for developing their system. This latter has been implemented by using the Java Agent Development Framework (JADE).

Regarding the role that multi-agents systems play in enterprise management in general, we can distinguish the work developed in [21]. In this paper, authors have envisaged a combination of Software agents and Web services within a platform of an enterprise software. In addition, they considered a contract net protocol among the agents and invoked the contract net from a web service. This work can be added to the direct applications of SMA in enterprise projects.

In [22], authors have proposed an automated multi-agent negotiation framework for decision making in the construction domain. The presented Architecture allowed software agents to mimic human behaviors and styles in building an automated negotiation system. In this work, BDI (Belief-Desire-Intention) agents have been chosen in order to take benefit from their intelligence in elaborating efficient decisions.

C. Security

We firstly consider the security integration in computer systems and particularly the modeling of business rules then in a second time, we will focus on the security in MAS. Finally, we end this section by expressing our motivation to use a security agent to guarantee this role in the two cases.

1) Security in business rules modeling

Capitalization of experts' knowledge is essential to the decision-making and strategic management of businesses. Facing with abundant information and continuous interactions, One question may arise: how to secure the knowledge and expertise of businesses? We cannot talk about the acquisition and capitalization of experts' knowledge in companies without addressing the crucial issue of security. To improve the knowledge safety and accessibility by the experts, the notion of cryptography has emerged as a basic concept in the business rules modeling.

2) Security aspect in multi-agents systems

As mentioned in many studies, security is a decisive factor to multi-agent technology to gain widespread use and provide viable solutions on a wider scale for commercial applications [23][24][25].

More precisely, Foner [26] provided a general discussion about the need for security in Multi-agent systems; here, the author discussed the threats and possible attacks that may occur in the system, He focused on the privacy of the users' information.

As we know, agents and multi-agent systems are distinguished from other software applications by a certain number of properties or characteristics. Such distinctions can make them vulnerable to attacks, intrusions and malicious programs.

As a first propriety which can be a subject of serious security concerns; is the autonomy of the agents. Generally, these latter are recognized to perform several actions in order to achieve their goals without human assistance.

Due to this autonomy, agents could be forced to commit malicious tasks that will conduct to a failure in the general functioning of the system; it will be possible for them to use the powerful features for malicious or suspicious purposes if they are not well controlled.

In multi-agent systems, the system security does not only depend on data confidentiality and integrity, but also on communications in message exchanges. Generally, agents must interact by cooperating, collaborating or negotiating their goals. One can say that problems deriving from social ability (in terms of interactions) are difficult to solve.

Agents, as any other software system, when distributed over a network, they must be prepared to face the classic security problems, such as breach of confidentiality, integrity attacks, and attempts to block the availability of a resource or the whole system [27].

In MAS, the communication is very important as we know agents are capable of executing any task to achieve their goals, and this implies the ability to communicate and interact with other parties (other agents, non-agents, humans) that constitute the global environment.

All these agents use social ability through cooperation, collaboration, coordination and negotiation once launched.

As argued by [28], in MAS especially when agents communicate with each other, they are subject of threats as well as conventional computer networks. The authors affirm that they will be considered as passive if they want to hear the communication otherwise (means active) when they try to intercept and modify the data related to the exchanges.

Several authors have discussed in many works the security issue in multi-agents systems providing some techniques or proposals. They sometimes add a security feature to the MAS and sometimes they try to solve this problem by integrating tools in the developed architecture.

D. Our contribution

Our suggested system allows the detection and management of inconsistencies in the business rules, following a rigorous control strategy implying the expert opinion in most situations. To do so, we develop a domain ontology that represents the enterprise's business model and supports checking for inconsistencies.

We can summarize our contribution in the following key-points:

- Knowledge acquisition from business experts through a user-friendly and ergonomic web editor.
- Design and development of domain ontology to generate the company's business model.
- Design and implementation of an agent-based architecture, where the Evaluator agent plays an important role.

This paper focuses on the decisions generation, this can be facilitated by executing the suggested rules in an automatic manner. Our approach makes it possible for businesses to respond rapidly to changes in business needs, thus enhancing the overall business efficiency.

In table 1, we present a comparison between some related works and ours. We just sort the items by using ' + ' to indicate that the option exists and ' - ' to indicate that the option doesn't exist. We put our approach in the bottom of table 1. Some indications are given with the table1 in order to clarify some criteria that are used in the comparison:

A: Use of Agents

B: Use of Ontology

C: Business rules consistency types

C1: The contradiction

C2: The rules never applicable

C3: The domain violation

C4: The invalid rules

C5: The redundancy

C6: The equivalence

D: Security aspect

E: Collaboration or cooperation between experts

TABLE I

Comparison Between Some Related Works and Our Approach

Works	A	B	C						D	E
			C1	C2	C3	C4	C5	C6		
Guissé et al. [10]	-	+	-	-	-	-	-	-	-	-
Guissé et al. [11]	-	+	-	-	-	-	-	-	-	-
Omrane et al. [12]	-	+	-	-	-	-	-	-	-	-
Chniti et al, [14]	-	+	-	+	+	-	-	-	-	-
Ram and Liu, [16]	+	+	-	-	-	-	-	-	-	+
Moradi et al., [17]	+	-	-	-	-	-	-	-	-	+
Our approach	+	+	+	+	+	+	+	-	+	+

As the Table 1 shows, our proposed approach has major advantages related to past approaches. We note that the security aspect is not taken into account by any of the competitor works as well as the collaboration between experts except some works that describe agents in cooperation. We can also notice that our approach treats more cases of rules consistency compared to other approaches (it treats 5/6 of the possible cases).

In the following section, we will describe in details our agent-based approach with more attention given to the management of business rules consistency.

III. The suggested Model

Our approach permits to acquire and capitalize the business experts' knowledge as business rules through an agent-based platform. Domain ontology is used in order to generate the business model corresponding to the enterprise and check the consistency of business rules.

The knowledge-based system is composed of several components as described in Fig. 1 [29]. We can cite, for example: Development

environment that defines the business model and the rules model (we use the standard format IF <Conditions> Then <Actions> because it offers more possibility to describe complex conditions and actions) as well as some necessary functions for the application development, a collaborative Interface and Knowledge management system.

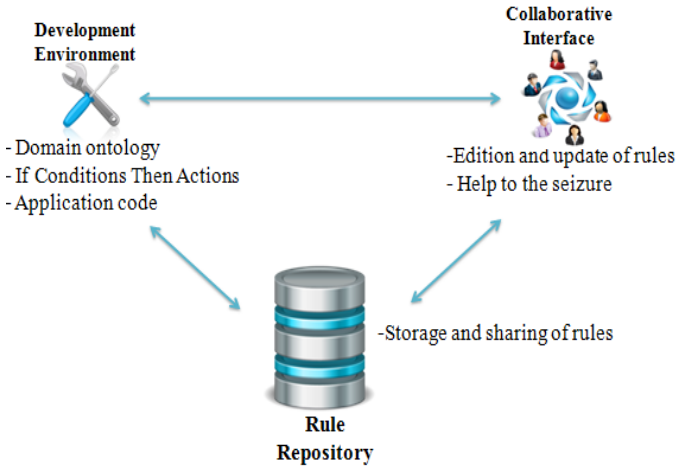

Fig. 1. The proposed model.

A. Domain ontology

It is necessary to provide an intuitive rule language easy to handle using the business vocabulary, in order to enable business experts to implement and maintain up to date their rule-based systems. Business language should allow representing the entities of the business domain, the actions that can be taken and the strategies to follow. To do this, our BRMS is based on domain ontology, representing all business entities and their properties and relations.

Domain experts may have difficulties expressing their knowledge in formalized logic languages because they are not business rules experts. One of our goals is to support them in their management of the knowledge needed to write these rules.

The use of different vocabularies for the same things by the experts is one of the difficulties with business rules modeling, so they cannot understand each other immediately [12]. In order to represent the business vocabulary used to express the rules, we propose using domain ontology as a unified model. This later will help the experts to express their rules more efficiently, to reduce misunderstandings and guarantee that people are discussing the same thing.

Our domain ontology is developed by acquiring knowledge from documents, collection and capitalization of business rules processes with domain experts and the interviews with company managers. Currently, we have implemented our ontology in Protégé 4.0.2. Fig. 2 depicts an overview of the main concepts and classes of the domain ontology that is applied to an e-business enterprise.

We have created our ontology according to the method described in what follow:

- Our ontology is built manually,
- Ontology Enrichment: we use the TERMINAE method which is a method and platform that assists users in designing terminological and ontological resources from texts [30].

The main objective is to save the different terms that are related to the conceptual vocabulary. We use the SKOS-based approach [12] to save the various linguistic units that denote a concept, instance or role.

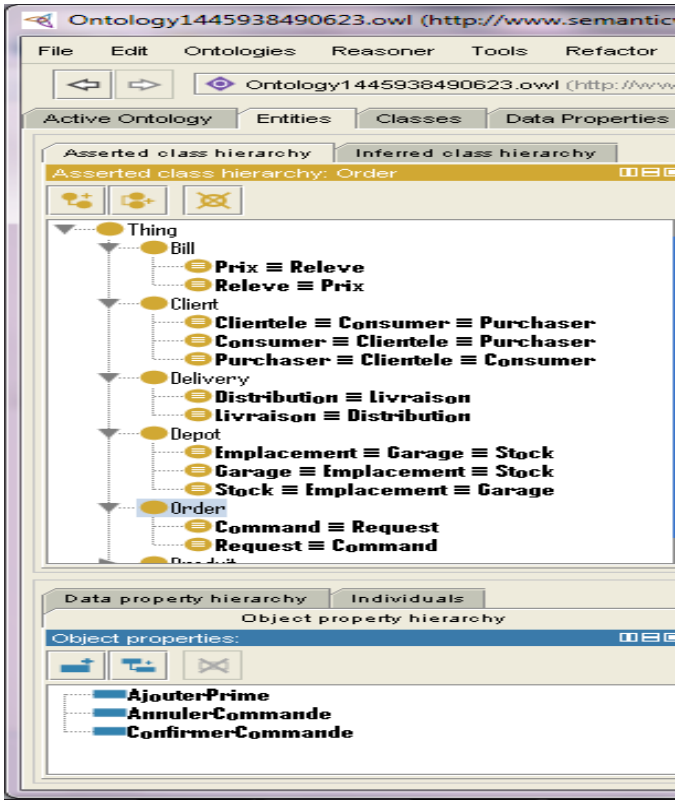

Fig. 2. A snapshot of our domain ontology.

B. Agent-based modeling

To achieve all functionalities, we use a multi-agents system that is composed of several agents (see Fig. 3):

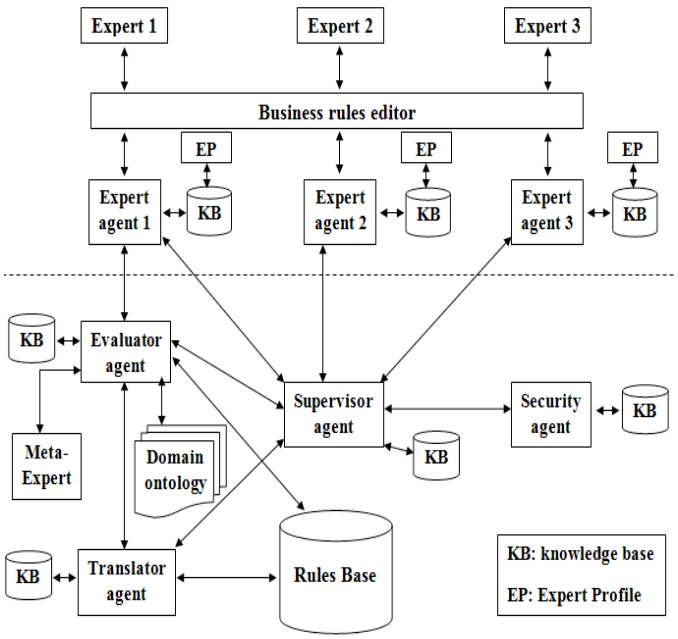

Fig. 3. An agent-based architecture.

1) Expert agent

This agent is responsible for the recuperation of the rules seized by the expert. This agent saves the rules and transmits them to the Evaluator agent. In the case of consistency problem, it receives a message from the Evaluator agent and displays the notifications and

recommendations on the rule editor. The Expert agent stores its data in a knowledge base that contains all the information about the rule being introduced (See Fig. 4).

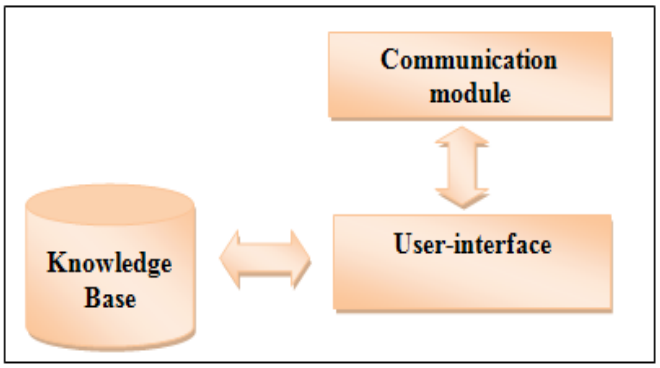

Fig. 4. Expert Agent Structure..

The communication module is responsible for the agent's interaction with the other agents in the system to get, send and receive some information.

The user-interface permits the interaction with the business expert to allow him to interact with our system.

In the knowledge base, all knowledge of the agent is stored:

- The information transmitted by the Evaluator agent,
- Profiles of business experts,
- The profile of the current expert.

2) *Supervisor agent*

This agent performs all control tasks in the system, that means it verifies if there is no failure or blockage in the whole system. This agent monitors and controls the accessibility and the sequence of the running of all the agents. Furthermore, it keeps the history of the agents and their tasks during the system execution.

The Supervisor agent plays a very important role within the MAS. Indeed, all interactions between the different agents of the MAS pass through it. In other words, if an agent A wants to send a message M to an agent B, then the agent A must firstly contact the Supervisor agent to know if agent B exists or not. In this case, the Supervisor agent checks whether the agent B exists or not, if yes then an ACK message will be sent to the agent A, otherwise it regenerates the agent B and sends the ACK to the agent A.

3) *Evaluator agent*

This agent is responsible for checking the consistency of the business rules. It recovers the rule from the Expert agent, browses the domain ontology to extract the set of concepts that correspond to the introduced rule 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 sends a message to the Expert agent and the Translator agent, otherwise it sends the rule to the Meta-expert.

We propose to use our Ontology to have benefit of the semantic similarity between the concepts as it offers a structured and unambiguous representation of knowledge in the form of conceptualizations interconnected by means of semantic pointers.

The Evaluator agent is composed of several modules such as (see Fig. 5):

- Consistency module: manages the consistency of business rules.
- Similarity Module: permits to value the similarity and dissimilarity between the concepts of the rule and the ontology. This latter is representing a very reliable and structured knowledge source used to compute semantic likeness.
- Communication Module: allows the interaction with the other agents in the system.
- Knowledge Base: contains all the information recorded by the Evaluator agent.
- Rules Base: contains all the consistent business rules of the enterprise.
- Domain ontology: permits to define a set of knowledge in a given domain.

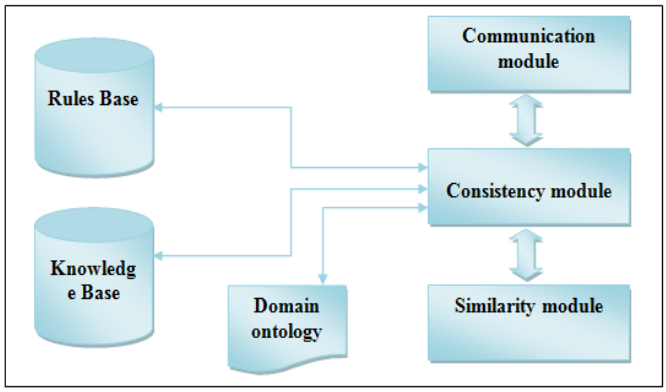

Fig. 5. Evaluator Agent Structure.

4) *Translator agent*

The Translator agent retrieves the rule from the Evaluator agent, and translates the introduced rule into a technical rule. The internal database of the Translator agent contains all mapping information for translating the rule.

To translate the business rules, we adopted the approach presented in [16], with some changes according to our needs.

The mechanism of translation consists of three parts: a business rules model, an enterprise data model, and a Translator agent linking these two models. The enterprise data model defines the data objects about which business rules are expressed, the business rules model captures semantics of the business rules and the Translator agent maps business terms to data objects usually attributes in the enterprise data model.

The Translator agent establishes a link between the business rules and the enterprise data model. To connect business rules with the enterprise data model, the agent performs the following steps:

- It identifies business terms in the business rule.
- It maps the identified business terms to data objects in the enterprise data model.

Translator agent is composed of several modules as described in Fig. 6:

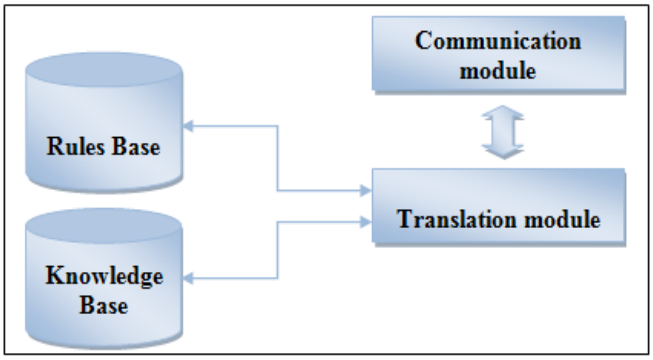

Fig. 6. Translator Agent Structure.

- Translation module: translates the business rules into technical rules.
- Communication Module: allows the interaction with the other agents in the system.
- Knowledge Base: stores all the knowledge that is manipulated by this agent.
- Rules Base: contains all the consistent business rules related to the enterprise.

5) Security agent

This agent maintains the security in the system; its role is to maintain the encryption and decryption of the business rule. For security reasons, exchanges between agents are encrypted by a hybrid system; the keys are provided by the Security agent.

This agent applies an encryption/decryption algorithm to allow confidentiality, authenticity and integrity of the most important rules. So, to increase the rules security, we use the well-known hybrid or mixed cryptography (see Fig. 7) in order to take benefit of the advantages of symmetric and asymmetric algorithms. The secret key exchange is done thanks to the RSA public key algorithm. The communication that follows is encoded by using the AES secret key algorithm. In [31], we have demonstrated through several sets of experiments the efficiency of the AES algorithm in terms of response time, space memory and security.

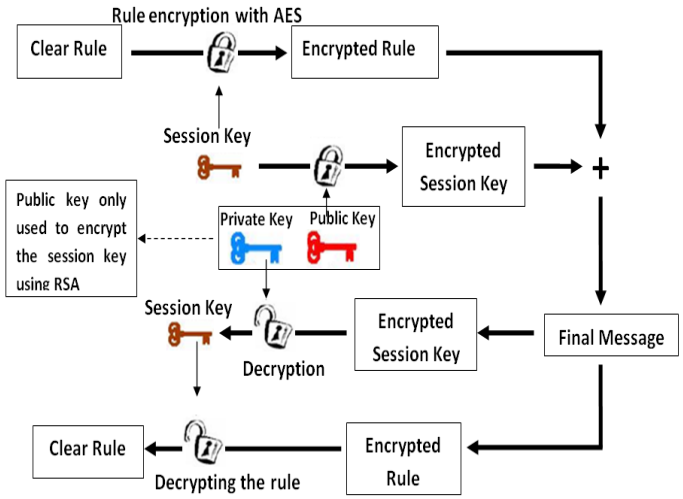

Fig. 7. Process of encryption and decryption.

C. Communication between agents

The interaction among the different agents of the system is shown by the sequence diagram that is expressed in AUML and represented in Fig. 8. There are several possibilities with knowledge management in MAS:

- Every agent has knowledge about its problem domain.
- Each of the agents has its own knowledge representation.
- If new information exists, an agent sends a message to all agents which might be interested.

D. Business rules consistency management

Currently, the effectiveness and correctness of business rules defined by the experts have always been a challenging problem. Our system should ensure that the complete set of business rules includes only the rules that are consistent and do not conflict among themselves. A possible way is to group the rules by the objects they constrain or actions they trigger and to check if there are any conflicts [32].

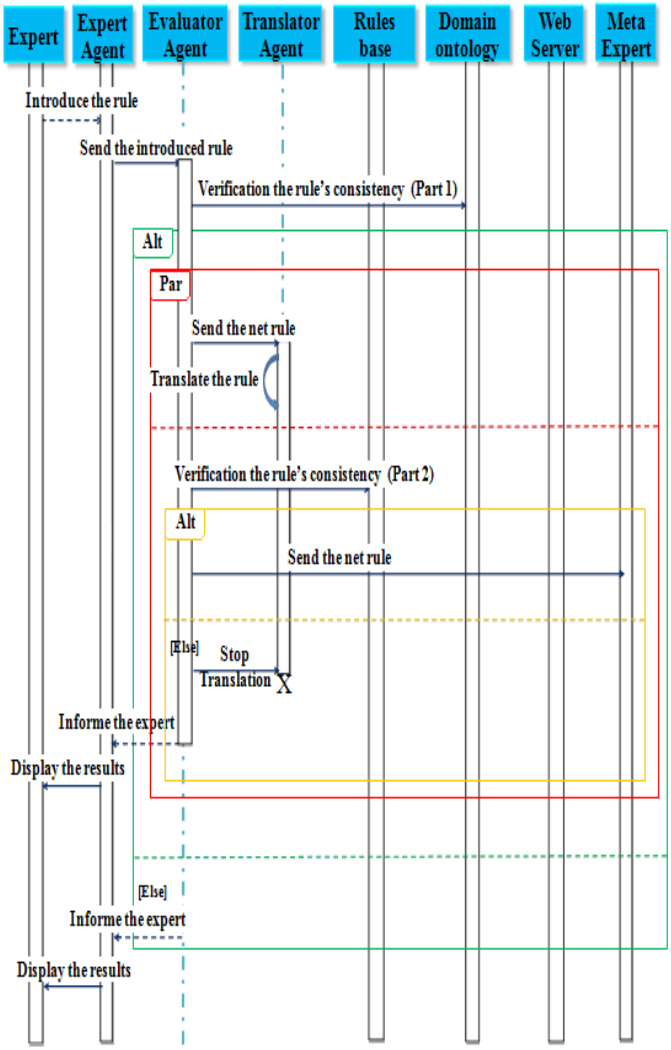

Fig. 8. Sequence diagram of our system,

In what follows, we will present the different types of inconsistencies that may impact a set of rules, and the solution brought to each type of inconsistency.

E. Business rules consistency management

Currently, the effectiveness and correctness of business rules defined by the experts have always been a challenging problem. Our system should ensure that the complete set of business rules includes only the rules that are consistent and do not conflict among themselves. A possible way is to group the rules by the objects they constrain or actions they trigger and to check if there are any conflicts [32].

In what follows, we will present the different types of inconsistencies that may impact a set of rules, and the solution brought to each type of inconsistency.

1) The contradiction

A contradiction is detected in a set of rules R, if this set contains, at least, two rules that have the same condition part and assign two different values to a same attribute in the action part.

If this type of problem is detected, the system sends a notification to the expert who introduced this rule with a very detailed report on the problem description.

The expert must respond to this notification and provide a solution to this problem as soon as possible, either by deleting or changing the rule.

2) The rules never applicable

A rule is never applicable if its condition part can never be verified.

The system sends a notification to the expert if this type of problem is detected with a very detailed description on the part of the rule which causes the problem as well as the range and interval of values requested for this attribute.

The expert must respond to this notification in the briefest delays and modifies the rule according to the sent report. In the case where the expert wants to modify the values or the properties of an attribute, he should contact the administrator to update the vocabulary of the domain according to new market changes.

3) The domain violation

A domain violation is detected in a set of rules, if this later contains, at least, one rule with a particular statement: it means in the action part, the assigned value to the attribute is out of its domain.

If this problem is detected, the business expert would receive a notification with recommendations on the editor with an explanation of the problem encountered as well as the values and properties allowed for this attribute. The expert must answer this notification in the briefest delays and modifies the rule according to the sent report.

4) The invalid rules

A rule is invalid if it uses in its premise or conclusion part, a concept or a property that does not belong to the ontology from which the rules were edited. To check if the rule is valid or not, the system accesses the domain ontology and follows these steps:

1. Search if the concept exists, if the concept does not exist then the system sends a notification message to the business expert and delete the rule, otherwise go to 2.
2. Search if the property exists, if it is not the case then the system sends a notification to the business expert and delete the rule, otherwise the system assumes that the rule is valid.

5) The redundancy

Two rules are called redundant if they have the same condition and produce the same action.

If this type of problem is detected, the system deletes the rule and sends notifications to the expert to inform him that the rule creates a redundancy problem and consequently it has been deleted.

6) The equivalence

Two rules are equivalent if the condition of the first one is included in the condition of the second and both rules produce the same action. To solve this problem, the system sends a message containing the equivalent rules to the expert and asked him to integrate and merge all equivalent rules into a single coherent rule.

The main idea of our work is the integration of ontologies and business rules in order to enable experts representing knowledge that describes the constraints and the business decisions. This integration creates a dependency between ontologies and rules due to the fact that the rules are edited from concepts and properties of the ontology. The domain ontology can be modified in two cases, which are:

1. In the case of the detection of some rules inconsistencies such as invalid rule, domain violation or rule never applicable, the expert is asked to correct the rule, three alternatives are possible: change the rule by using one of the stored values that exist in the ontology, delete completely the rule or update the vocabulary. To do this, the expert must work with the administrator in order to update the domain ontology.
2. To meet the new needs and the market changes, the ontology must be modified by the administrator in collaboration with the domain experts. This modification may cause some inconsistencies in business rules. Actually, we are studying the impact of the ontology evolution on the business rules consistency.

We present in the following the pseudo code of business rules consistency management (see Algorithm 1).

Algorithm 1: Checking the consistency of business rules

```
1.  Input : a business rule
2.  Output : the rule is consistent or not
3.  Eliminate the empty words
4.  Extract the Property, Concept, Test, Value and Action of the business rule
5.  Access to the domain ontology
6.  // The verification of the invalid rules
7.  for all concepts of the ontology do
8.      if (ConceptOnto.equals(Concept)) then
        // check whether the concept exists
9.        The Concept exists
10.           if (Concept is a term) then
11.             Recover the concept of the
        term
12.           end if
13.       Get all the properties of this concept
14.     for all properties of this concept do //
        check if the property exists
15.           if (propertyOnto.equals(Property))
        then
16.             The Property exists
17.               Get the main property if
        exists
18.             // The verification of not applicable
        rules
19.             // The case where the property contains predefined
        values
20.             try {
21.             Get the predefined property values and their types
22.                 if (ValueProp.contains(Value))
        then
23.                     The rule is
        applicable
24.                 else
25.                     The rule is not
        applicable
26.                 end if
27.             } catch (Exception e) {
28.             // The case where the property has a
        type
29.             Get the type of the property
30.             Parse the value according to the property
        type
31.                 if (Property Type.
        equals("ValueType")) then
```

```
32.                                The rule is applicable
33.                        else
34.                                The rule is not applicable
35.                        end if
36.                }
37.                else
38.                        The rule is invalid // The concept exists but the property does not exist
39.                end if
40.        end for
41.        else
42.                The rule is invalid // The concept does not exist
43.        end if
44. end for
45. Connection to the Rules Base
46. for all number of rules do
47.    // The redundancy verification
48.        if (the same condition and the same action exist) then
49.                The rule is redundant
50.        else
51.                The rule is not redundant
52.        end if
53.    // Verification of the contradiction
54.    Recover contradictory actions with this action
55.        if (the same condition exists and at least one contradictory action exists) then
56.                The rule is contradictory
57.        else
58.                The rule is not contradictory
59.        end if
60. end for
```

F. The collaboration between experts in the case of inconsistencies

In the case of inconsistencies, the business expert receives a notification on the editor to correct its introduced rule according to the sent report. If the expert doesn't answer before XX days (the number of days is defined by the enterprise), the system sends to him an e-mail and SMS to correct the rule. If the expert doesn't answer after XX days, then the system sends the rule as well as the assessment report to the other experts and to the meta-expert to ask for their help and their opinion.

Each expert can answer by «Yes» or «No». In the case of «Yes», the expert should send the new modified and corrected rule.

After harvesting responses from business experts, a voting system is launched to decide. If all experts vote «No», therefore the rule will be definitely deleted and the system sends a notification message to the expert who introduced this rule, otherwise the experts can correct the wrong rule and this later pass through the reparation process (see Fig. 9).

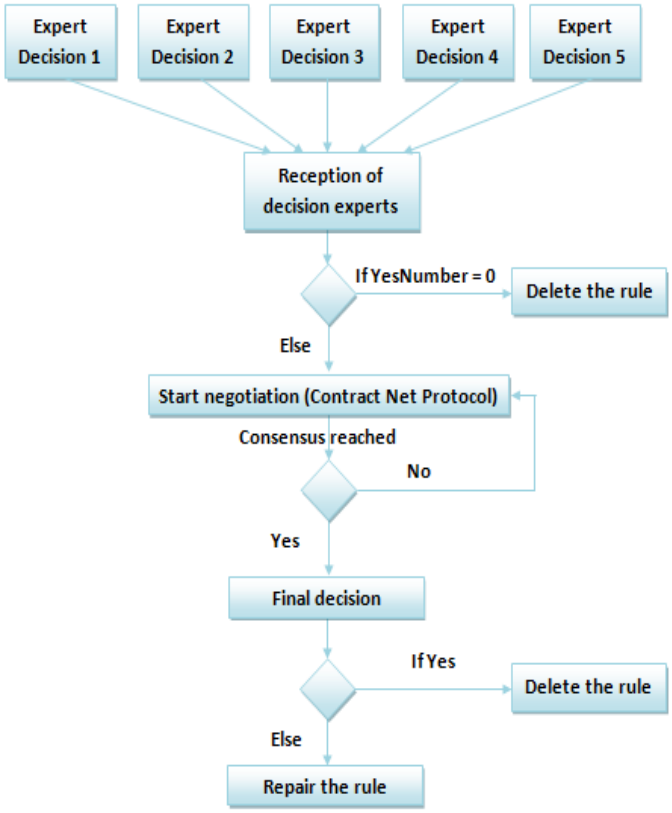

Fig. 9. Collaboration between experts in the case of inconsistencies.

The negotiation mechanism among the experts is based on the well-known Contract Net Protocol which is a model for which only the manager emits propositions. The contractors can only make an offer, but not counter-propositions. In our case, we propose to extend the CNP to consider the opinion of contractors (experts) in order to find more quickly a common accepted solution. In our approach, we use the Contract Net Protocol for its advantage to be a dynamic and easy to implement as an algorithm. The suggested negotiation model operates on a business rule, thus other tasks are added to the manager. This will eventually influence its decision- making.

IV. Implementation and Discussion

This section is divided into two parts; the first one is dedicated to the collaborative graphical platform that allows the experts to introduce, update their rules and launch simple or advanced research by using some keywords. In the second part, some experiments are done to test the business rules consistency management and evaluate the agents' performance. The system was implemented with Java language and JADE (Java Agent Development) interface. The knowledge base and rules base was implemented by using WampServer. We used Apache Tomcat as an open source web server and servlet container. The open source ontology editor Protégé 4.0.2 was used for the creation and visualization of the ontology. We have developed our application and launched the simulations on a computer Intel (R) Core (TM) i5-3230M CPU at a speed of 2.60 GHz, equipped with a Memory capacity of 4.00 GB of RAM on Windows 7.

A. How to edit the rules

Once authenticated, the expert can access to the rules editor, here he will capitalize his knowledge and experience in a particular domain. For example, thanks to the editor, the expert can introduce the following rule (See Fig. 10):

If the customer's state is MIN and the customer's category is GOLD Then Position the discount of the order to 10%.

The rule used here is extracted from a detailed case study of the work done in [33]. We prefer using the known rule base for our scenario in order to compare the most relevant results.

Our editor offers other features like advanced research mainly based on some keywords and criteria, update and deletion of the rules. The editor permits to the experts to interact with each other.

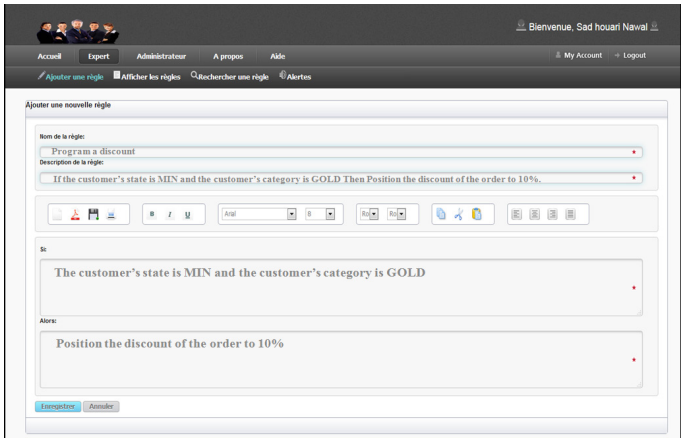

Fig. 10. Business rules editor interface.

B. Evaluation of the platform usability

An evaluation of system usability using a questionnaire was conducted to show the effectiveness of the proposed platform. We get inspired by the questionnaire given in [34] and we develop our questions regarding the system we suggested.

The questionnaire was distributed to 10 users consisting of experienced and un-experienced users, including business experts from the enterprise.

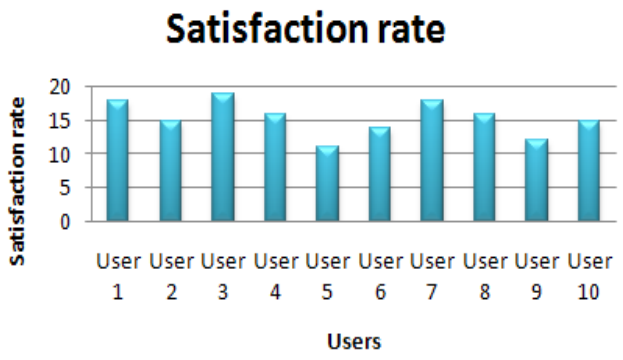

Fig. 11. Evaluation of the platform usability.

Fig. 11 shows the usability results obtained from the questionnaire, in which some questions are followed by a reversed question to reveal opposing facts. For some questions, we assigned a weight. At the end, we count the sum of weight to find the user's satisfaction level according to the formula 1. We note that some questions are not included in the calculation. These questions just give us information, conclusions and directions for future improvement of the platform.

$$Satisfaction\_rate_j = \sum_{i=0}^{n} Expert\_Response_i \quad (1)$$

Where :
j : the expert number j
n : Number of questions

Expert_Response$_i$: the expert response on the question number i. We note that: Expert_Response$_i$ $\leq$ Weight$_i$ where Weight$_i$ = the weight assigned to the question number i.

Some questions of the questionnaire are presented in what follow:

- If you have three words to describe the platform, what are these words?
- Which part of the platform did you find most interesting?
- Which part of the platform do you want to discuss or treat in more depth? And why?
- Would you like to add anything else? Do you have any other comments?
- On a scale of 0-5, how would you value the performance of the platform (response time and access)?
- On a scale of 0 to 5, how much do you value the ergonomics / the ease of use of the tool?
- On a scale of 0 to 5, how much do you value the overall tool?
- There were inconsistencies in the navigation? Yes Or No
- The interaction components (buttons, menus, text fields, drop-down lists, etc.) can be easily understood? Yes Or No
- Do you need to learn many things before using the platform? Yes Or No
- Is it easy and effective to share your experiences in the platform? Yes Or No

TABLE II
BUSINESS RULES SET

Inconsistency type	Rules number	Number of rules detected
Invalid rules	12	12
Domain violation	7	7
Rules never applicable	11	11
Redundancy	8	8
Contradiction	12	12
Equivalence	10	0

From the questionnaire results, one could argue that the usability/ readability of the platform is reasonably high.

C. Experimentations

An important factor for an acceptance of our system is its runtime performance. This section provides various scenarios to evaluate our prototype and test the feasibility of our approach on several business rules. It also serves to confirm the claims that were made through the paper.

1) Experiment 1: Test of our consistency management module

In order to test our system, we launched 60 business rules which contain: 12 contradictory rules, 8 redundant rules, 10 equivalent rules, 12 invalid rules, 11 rules never applicable and 7 rules that pose a domain violation problem (see Table 2). We note that our rule base already contains 30 consistent rules.

The obtained results are very encouraging and show that the developed system treats the problem of contradiction, redundancy, domain violation, invalid rules and rules never applicable but does not treat the equivalence problem.

2) Experiment 2: Agents performance

Table 3 presents the execution time of the system agents in milliseconds with 8 business rules. The execution time of the Evaluator agent depends greatly on both: the ontology size and rules base size.

TABLE III
RESPONSE TIME OF THE SYSTEM AGENTS

A \ R	R1	R2	R3	R4	R5	R6	R7	R8
EA	315	388	484	328	367	312	434	420
TA	801	760	749	750	845	716	758	791
EvA	860	760	663	822	768	795	915	760
SA	227	211	222	210	230	207	220	230
SuA	218	225	218	302	323	360	277	334
Total	2421	2344	2336	2412	2533	2390	2604	2535

A: Agents
R: Rules
EA: Expert Agent
TA: Translator Agent
AvA: Evaluator Agent
SA: Security Agent
SuA: Supervisor Agent

In Table 4, we calculate the average response time of each agent.

TABLE IV
AVERAGE RESPONSE TIME OF THE SYSTEM AGENTS

Agents \ Rules	Average response time
Expert Agent	381
Translator Agent	771.25
Evaluator Agent	792.875
Security Agent	219.625
Supervisor Agent	282.125
Total	2446.875

Table 5 presents the space memory, in bytes, occupied by each agent when introducing the business rules.

TABLE V
SPACE MEMORY OCCUPIED BY THE SYSTEM AGENTS

R \ A	EA	*TA*	*AvA*	*SA*	*SuA*	Total
R1	5,217,000	12,888,168	5,045,912	8,465,032	9,089,840	40,705,952
R2	4,882,304	13,843,472	4,994,128	8,057,272	8,455,584	40,232,760
R3	3,845,216	11,084,144	5,010,560	8,143,848	8,169,928	36,253,696
R4	4,083,520	12,610,440	4,981,352	8,073,504	10,518,416	40,267,232
R5	3,393,840	13,430,088	5,142,112	8,183,336	7,659,904	37,809,280
R6	2,653,456	14,204,656	5,062,320	8,192,048	12,227,992	42,340,472
R7	3,981,064	14,131,048	5,020,184	8,083,992	9,343,208	40,559,496
R8	3,735,376	13,035,096	5,012,224	7,921,296	11,764,176	41,468,168

A: Agents
R: Rules
EA: Expert Agent
TA: Translator Agent
AvA: Evaluator Agent
SA: Security Agent
SuA: Supervisor Agent

In Table 6, we calculate the average space memory of each agent.

TABLE VI
AVERAGE SPACE MEMORY OCCUPIED BY THE SYSTEM AGENTS

Agents \ Rules	Average space memory
Expert Agent	3,973,972
Translator Agent	13,153,389
Evaluator Agent	5,033,599
Security Agent	8,140,041
Supervisor Agent	9,653,631
Total	39,954,632

3) *Experiment 3: Supervisor agent performance*

Table 7 shows the response time required to regenerate an agent which breaks down in milliseconds. We launched several simulations and calculate the average response time at a detected fault. In what follows, we will explain the results obtained by the Expert agent: when the Expert agent is planted for the first time, the required period to resolve the problem is 1 ms, in the second time, the system took 3 ms in order to create it again. For the third and the fifth time, 2 ms are required to be active in the system. Concerning the fourth time, we observed 6 ms for starting. As a final result, we can estimate the average response time required to generate the Expert agent in the case of its failure (blockage or failure) to 2.8 ms.

TABLE VII
RESPONSE TIME EVALUATED IN CASE OF AGENT FAILURE

Agents \ Test	1	2	3	4	5	Average response time
Expert Agent	1	3	2	4	2	2.8
Evaluator Agent	2	1	5	10	1	3.8
Translator Agent	1	11	1	1	2	3.2
Security Agent	1	8	2	9	1	4.2

4) *Experiment 4: Communication between agents under JADE*

The JADE print screen will show the agents' communication as it appears in Fig. 12. In this work, Sniffer Agent provided by JADE was employed to monitor the communication among agents on the agent platform. After getting authenticated, the human expert accessed to the rules editor, and the system automatically generates an Expert agent. If the expert wants to introduce a new rule, so the Expert agent saves the rule that is introduced in its knowledge base. In what follow, we will explain each message which is exchanged between agents:

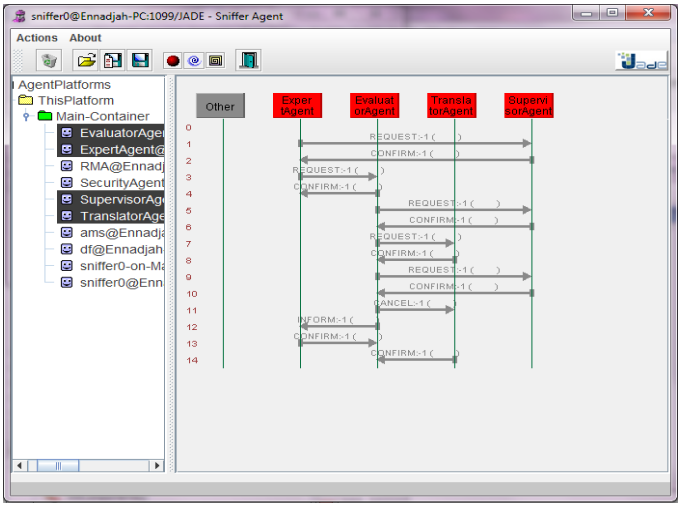

Fig. 12. Agents communicating with JADE platform.

Message 1 (Request): The Expert agent sends a request message to the Supervisor agent in order to ask whether the Evaluator agent exists or not.

Message 2 (Confirm): The Supervisor agent answers with a confirmation indicating the existence of Evaluator agent (if this later does not exist, the Supervisor agent will recreate it again).

Message 3 (Request): The Expert agent sends the rule that is introduced to the Evaluator agent.

Message 4 (Confirm): An acknowledgment of receipt is sent to the Expert agent once the Evaluator agent receives the rule.

Message 5 (Request): The Evaluator agent asks the Supervisor agent if the Translator agent exists.

Message 6 (Confirm): The Supervisor agent sends a confirmation in order to indicate that the Translator agent is ready to accomplish its tasks.

Message 7 (Request): The Evaluator agent accesses to the domain ontology for the first consistency check (verification of the validity, applicability and domain violation of the rule). At this stage, if there is no consistency problem then the Evaluator agent sends the net rule to the Translator agent for the technical translation and accesses to the rules base for the second step of the consistency checking (verify whether the rule is redundant or contradictory).

Message 8 (Confirm): Once the Translator agent receives the rule to translate, it responds with a confirmation message.

Message 9 (Request): The Evaluator agent sends a request asking the Supervisor agent whether the Translator and the Expert agents exist.

Message 10 (Confirm): The Supervisor agent confirm positively.

Message 11 (Cancel): The Evaluator agent sends a message to the Evaluator agent in order to stop the translation of this rule because it is inconsistent.

Message 12 (Inform): The Evaluator agent sends a message to the Expert agent to notify the human expert that the rule is inconsistent (here we take an example of a redundancy rule).

Message 13 (Confirm): The Expert agent sends a confirmation message to the Evaluator agent and displays the results on the rules editor.

Message 14 (Confirm): The Translator agent stops the translation and sends an acknowledgment of receipt to the Evaluator agent.

V. Conclusion

Distributed environments, artificial intelligence tools and collaborative technologies are factors that considerably influence our current research work.

This paper proposes a novel collaborative approach for business rules consistency management in a typical small and medium enterprise. The suggested approach is utilizing domain ontology within an agent-based architecture. Compared to other major business rules modeling approaches, our works aim to take well into account the organizational aspects as well as the security one in both rules level and agents level.

The modeling is based on agents to increase the execution speed of processes and effective response. In our system, all agents are autonomous and cognitive. They can aid experts and have the ability to explain their actions to other agents. Furthermore, they can adapt their actions to environmental changes.

The Expert agent can interact with human (expert) directly. Other agents communicate with each other to fulfill their common goal under the control of the Supervisor agent. In order to represent the business vocabulary used to express the rules, we propose using domain ontology as a formal model. On the one hand, the ontology gives the vocabulary used in expressing the rules and on the other hand, it provides a structured vocabulary that encodes relationships between concepts and supports checking for inconsistencies.

Our work is dedicated to the knowledge capitalization of industry experts. The knowledge management process involves two phases: first, the experts have an editor for entering their knowledge, all the update processing, and safeguard are insured. The consistency management of knowledge (rules) is also provided, and the case of redundancy, contradiction, invalid rules, domain violation and rules never applicable are investigated and verified by the system. According to the obtained result, we notice that our approach can detect all types of inconsistency except the equivalence problem. From the results of Table 1, we can say that our approach provides better results reports to the competing approaches, and it can detect almost any possible case of inconsistencies. Second, in the case of incoherent rule, a collaboration session is initiated between experts; this step can lead to a negotiation among participants. This model enables effective collaborative decision-making and also facilitates exchanges and knowledge sharing between the different actors in safety. Our collaborative system provides:

- An editor easy to use,
- It offers features that stimulate business users and developers to use it,
- Automate precise, highly variable decisions,
- Easy, safe and predictable rule management.

As a future work, we envisage to test and evaluate our prototype in other SME enterprises with a large number of rules and handle the problem of equivalence rules.

References

[1] H. F. Ho, and S. T. Li, "Using mutually validated memories of experts for case-based knowledge systems," Knowledge-Based Systems, vol. 86, pp. 102-115, 2015.

[2] W. Wang, X. Guo, Y. Fan, and J. Bi, "Extended Access Control and Recommendation Methods for Enterprise Knowledge Management System," in FIE 2014: International Conference on Future Information Engineering, IERI Procedia, Beijing, China, pp. 224–230, 2014.

[3] D. Monticolo, J. Badin, S. Gomes, E. Bonjour, and D. Chamoret, "A meta-model for knowledge configuration management to support collaborative engineering," Computers in Industry, vol. 66, pp. 11–20, 2015.

[4] S. Kumar, "A knowledge based reliability engineering approach to manage product safety and recalls," Expert Systems with Applications, vol. 41, no. 11, pp. 5323-5339, 2014.

[5] M. Li, L. Liu, and C.B. Li, "An approach to expert recommendation based on fuzzy linguistic method and fuzzy text classification in knowledge management systems," Expert Systems with Applications, vol. 38, no. 7, pp. 8586–8596, 2011.

[6] P. Zaraté, and S. Liu, "A new trend for knowledge-based decision support systems design," International Journal of Information and Decision Sciences, 2015.

[7] V. Sugumaran, "An agent-based knowledge management framework for the ECommerce environment," Journal of Computer Information Systems, suppl. Special Issue vol. 42, no. 5, pp. 63–73, 2002.

[8] N.R. Jennings, M.J. Wooldridge, "Agent Technology Foundations, Applications, and Markets," Springer, New York, 1998.

[9] R. S. S. Guizzardi, "Agent-oriented Constructivist Knowledge Management," PhD thesis, University of Twente, the Netherlands, 2006.

[10] A. Guissé, F. Lévy, and A. Nazarenko, "From regulatory texts to BRMS: How to guide the acquisition of business rules?," in The 6th International Symposium on Rules: Research Based and Industry Focused, Montpellier, France, pp. 77-91, 2012.

[11] A. Guissé, F. Lévy, N. Omrane, A. Nazarenko, and S. Szulman, " Une architecture pour la construction de modèles de règles métiers documentés," Paper Presented at IC 2012, 2012.

[12] N. Omrane, A. Nazarenko, P. Rosina, S. Szulman, and C. Westphal, "Lexicalized ontology for the management of business rules An industrial experiment," in Workshop "Ontology and lexicon" of the 9th International Conference on Terminology and Artificial Intelligence, Paris, France, 2011.

[13] A. Chniti, P. Albert, and J. Charlet, "A loose coupling approach for combining owl ontologies and business rules," in C. W. proceedings, Ed., RuleML2012@ECAI Challenge, at the 6th International Symposium on Rules Research Based and Industry Focused 2012, vol.874, pp. 103–110, 2012.

[14] A. Chniti, P. Albert, and J. Charlet, "Gestion des dépendances et des interactions entre Ontologies et Règles Métier," PHD thesis, Pierre et Marie Curie University, Paris VI, 2013.

[15] J. Duda, and A. Stawowy, "A possibility of Business Rules application in production planning," ARCHIVES OF FOUNDRY ENGINEERING

Vol.10, ISSN (1897-3310), Issue 2/2010, pp. 27-32, 2010.

[16] S. Ram, and J. Liu, "An Agent-Based Approach for Sourcing Business Rules in Supply Chain Management," International Journal of Intelligent Information Technologies, vol. 1, no. 1, 2005.

[17] M. Moradi, A. Aghaie, and M. Hosseini, "Knowledge-collector agents: Applying intelligent agents in marketing decisions with knowledge management approach," Knowledge-Based Systems, vol. 52, pp. 181–193, 2013.

[18] I. Lahoud, " Un système multi-agents pour la gestion des connaissances hétérogènes et distribuées," PhD thesis ,Computers and Society. Université de Technologie de Belfort-Montbeliard, 2013.

[19] J. Girodon, D. Monticolo, E. Bonjour, and M. Perrieet, "An organizational approach to designing an intelligent knowledge-based system: Application to the decision-making process in design projects," Advanced Engineering Informatics, vol. 29, no. 3, pp. 696–713, 2015.

[20] H.B. Baron, M. M. Rojas, R.G. Crespo, and O.S. Martinez, "A multi-agent matchmaker based on hidden markov model for decentralized grid scheduling," in Intelligent Networking and Collaborative Systems (INCoS), 2012 4th International Conference on, Bucharest, pp. 62-69, IEEE, 2012.

[21] J.P.P. Grau, A.C. Sanz, and R.G. Crespo, "An Evaluation of Integration Technologies to Expose Agent Actions as Web Services," Practical Applications of Intelligent Systems, pp. 259-270, Springer Berlin Heidelberg, 2014.

[22] M.A. Mahmoud, M.S. Ahmad, M.Z.M. Yusoff, and A. Idrus, "An Automated Negotiation-based Framework via Multi-Agent System for the Construction Domain," International Journal of Artificial Intelligence and Interactive Multimedia, vol. 3, no. 5, pp. 23-27, 2015.

[23] N. Borselius, "Security in multi-agent systems," International conference on security and management (SAM 02), Las Vegas: CSREA Press, pp. 31–36, 2002.

[24] G. Beydoun, G. Low, H. Mouraditis, and B. Henderson-Sellers, "A security aware metamodel for multi-agent systems," Information and software technology, vol. 51, no. 5, pp. 832–845, 2009.

[25] P. Novák, M. Rollo, J. Hodík, and T. Vlček, "Communication security in multiagent systems," Multi-Agent Systems and Applications III (Lecture Notes in Computer Science), vol. 2691, pp. 454–463, Berlin Heidelberg: Springer-Verlag, 2003.

[26] L.N. Foner, "A security architecture for multi-agent matchmaking," in Proceedings of the 2nd international conference on multi-agent systems, pp. 80–86, 1996.

[27] R.C. Cavalcante, I.I. Bittencourt, A.P.D. Silva, M. Silva, E. Costa, and R. Santos, "A survey of security in multi-agent systems," Expert Systems with Applications, vol. 39, no. 5, pp. 4835–4846, 2012.

[28] F. Lu, and M. Huang, "Research and design of security in multi-agent systems," in ICWMMN 2006: Proceedings of International Conference on Wireless, Mobile and Multimedia Networks, hangzhou, China, pp. 1 – 4, 2006.

[29] N. Sad-Houari, and N. Taghezout, "Towards a new agent based approach for modeling the business rules processes in small and medium enterprise in Algeria," in ICNAS'2015: Proceedings of the 2nd International Conference on Networking and Advanced Systems, Annaba, Algeria, pp. 111 – 116, 2015.

[30] ONTORULE Project. (2011). TERMINAE: from texts to ontologies. [Online]. Available: http://ontorule-project.eu/news/news/terminae.html (Accessed 24 June 2015)

[31] N. Sad-Houari, and N. Taghezout, "An agent based approach for security integration in Business Rules Management System," in IPAC'2015: Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication, Batna, Algeria, 2015.

[32] M. Bajec, and M. Krisper, "A methodology and tool support for managing business rules in organisations," Information Systems, vol. 30, no. 6, pp. 423–443, 2005.

[33] B. Stineman, (2009) Pourquoi des règles métier ? : Un cas pour les utilisateurs métier de l'informatique, [Online]. Available: ftp://public.dhe.ibm.com/software/fr/ilog/IBM_ILOG_-_Pourquoi_des_regles_metier.pdf, (Accessed 11 January 2014).

[34] C.J. Su, and C.W. Peng, "Multi-agent ontology based Web 2.0 platform for medical rehabilitation," Expert Systems with Applications, vol. 39, no. 12, 2011.

N. SAD HOUARI received her Master degree in Information System Technologies at University of Oran1 Ahmed BenBella, Algeria in 2013. She is a member of the EWG-DSS (Euro Working Group on Decision Support Systems) since 2016. Currently, she is a Ph.D. student in Computer Science Department at the same University. Her research interests include business rules modeling, Artificial Intelligence, Security, multi-agents system and knowledge management.

N. TAGHEZOUT is an assistant professor at University of Oran1 Ahmed BenBella, Algeria. She holds her doctorate thesis in MITT at PAUL SABATIER UNIVERSITY in France in 2011. She also received another doctorate thesis in Distributed Artificial Intelligence from University of Oran1 Ahmed BenBella in 2008. She holds a Master degree in Simulation and Computer aided-design. She conducts her research at the LIO laboratory as a chief of the research group in Modeling of enterprise process by using agents and WEB technologies. Since she studied in UPS Toulouse, she became a member of the EWG-DSS (Euro Working Group on Decision Support Systems). She is currently lecturing Collaborative decision making, Enterprise management and Interface human machine design. Her seminars, publications and regular involvement in Conferences, journals and industry projects highlight her main research interests in Artificial Intelligence.