Journal of Industrial Information Integration 17 (2020) 100124

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.elsevier.com/locate/jii

OCEAN: A multi agent system dedicated to knowledge management

Davy Monticolo[a,*], Inaya Lahoud[b], Pedro Chavez Barrios[a]

[a] University of Lorraine, ERPI laboratory, France
[b] Department of Computer Engineering, University of Galatasaray, Istanbul, Turkey

ARTICLE INFO

Keywords:
Agent-based modeling
Ontology
Semantic approach
Knowledge management

ABSTRACT

Emphasis on knowledge and information is one of the challenges of the 21st century to differentiate the intelligent business enterprises. Enterprises have to develop their organization in order to capture, manage, and use information in a context of continually changing technology. Indeed knowledge and information are completely distributed in the information network of the company. In addition, knowledge is by nature, heterogeneous since it is provided from different information sources like the software, the technical report, the meeting statements, etc. We present in this paper the architecture of a multi-agent system, which allows the capitalization of the distributed and heterogeneous knowledge. We then present how the agents help business experts to design ontologies in detailing this problematic and how the agents extract knowledge from different users' databases by using a semantic approach.

1. Introduction

The industrial interest in methodologies and tools enabling capitalization and management of distributed and heterogeneous knowledge grew stronger, especially in designing complex products within an extended enterprise. The development of these complex products involve multi-disciplinary team (mechanics, automation, designers, engineers and technicians methods, etc..) to work collaboratively to design, develop and industrialize this product. In these projects environment, Knowledge is heterogeneous, since it comes from different sources. Rahman [53] and Cheng et al. [13] highlight the importance to develop the best practice to manage data, information and knowledge. Knowledge is also distributed throughout the enterprise network since each professional actor uses his own software tool connected to the entire corporate network.

The establishment of knowledge management system is strategic for companies. Several research works [42,54] have shown that not managing knowledge is a loss of competitiveness. But the knowledge management of the company is quite complex because it does not consist only of collecting and disseminating knowledge through the application of new technologies [65]. Indeed, according to Ermine [1], it is a long-term program that starts from a strategic intent, which requires a good analysis of the nature of knowledge and know-how of the company and which led to the development of various and adapted tools. The confirmation made by Ermine proves that the mechanisms of knowledge management are complex [19].

In additional, inside an engineering project, the knowledge is provided from different information sources (multitude of professional software tools, several different databases, …) [20] which are distributed all over the network of the company [39,45]. Moreover, the population of the projects actors is, by nature, heterogeneous and distributed. In order to face of the complexity of a knowledge management approach in such environment we need to take into account of the social and cooperative aspects of the human actors when they create and share knowledge and we need also to have an efficient system which could manage the different complex information sources. Thus, we propose the use of the agent paradigm that has proved its effectiveness in solving complex problems in information environments where entities are by nature heterogeneous and distributed. However the multi agent system have also proved they relevance to integrate the social and cooperative aspect of the users [2,51,56].

In our previous researches [45], we have design a multi agent system to structure knowledge by using ontologies and the results of this first prototype allow us to identify four complex problems in our knowledge management approach; the knowledge bases definition by the experts, the knowledge identification and research inside the enterprise network, the knowledge broadcast and evaluation by the professional actors and the reuse of the knowledge during professional activities [10,30]. All this complex problems are treated by four different agents communities, which represent four parts of the knowledge management system.

The first part of our system is an ontology editor, which allows to

* Corresponding author.
E-mail addresses: davy.monticolo@univ-lorraine.fr (D. Monticolo), pedro.chavez@univ-lorraine.fr (P.C. Barrios).

https://doi.org/10.1016/j.jii.2019.100124
Received 14 May 2018; Received in revised form 11 September 2019; Accepted 13 December 2019
Available online 17 December 2019
2452-414X/

professional experts to create the structure of their knowledge base i.e. to define all the knowledge they want to capitalize and the relations between them. Since the creation of concepts in ontology is described in natural language, the same concept can be defined differently by each person. Indeed, we face here the problem that business expert can add a concept to the ontology which does not match exactly with the content of the database of the business application but it matches a synonym for instance [23]. For that reason we need a system that helps these professional experts in creating ontologies that fit with the existing databases.

This paper describes our work to help professional experts in creating their ontologies with our knowledge based system. This first part of the system is called OCEAN which is the ontology creation package. This paper is organized as follows: after a presentation of some works on the application of multi agents system (MAS) to develop a knowledge management system in Section 2, Section 3 introduces our MAS approach to support KM in OCEAN platform and the contribution of this system. At the end we describe the results of an experiment by using OCEAN in an industrial context.

2. Agents and ontologies to manage heterogeneous and distributed knowledge

OCEAN is a multiagent system aiming to support the knowledge management process with four functionalities; the knowledge definition, the knowledge extraction, the knowledge evaluation and the knowledge reuse. We present in this section the interest of using a MAS to support the knowledge management process.

2.1. MAS used in knowledge engineering

The aim of knowledge engineering is to gather, study, organize and represent knowledge. Multi-agent systems have already proved their efficiency to support such tasks. Klusch [35] made a list of the services that a multi-agent system can offer in a knowledge management approach:

Search, acquire, analyze and classify knowledge coming from various information sources.

Give information to human and computing networks once usable knowledge is ready to be consulted.

Negotiate on knowledge integration or exclusion into the system.

Give explanation to the quality and reliability related to the integrated knowledge.

Learn progressively all along the knowledge management process.

Such services are mostly implemented to create two MAS categories devoted to knowledge management. The first MAS type is based upon an agent cooperation to solve complicated problems related to knowledge types. Some of these MAS were created as complementary tools in information management (workflow, ontologies, information research systems and so on) to design platforms like FRODO [1], CoMMA [28], Edamok [6], or KRAFT [36]. All these works have focused on the 'Multi-Agent Information System' [68].

The second MAS type gathers management assistant agents depending on the actors' needs. In this range, agents are expected to be flexible, pro-active and reactive regarding the user requirements [11,16,35].

The new trend of using MAS in knowledge engineering is to associate agents to knowledge structures based on ontologies. This association allows MAS to support the knowledge management process but the issues of knowledge distribution and ontology consistency with MAS have not been solved yet [73]. The next section will present the related work concerning agent approaches using ontologies in knowledge engineering.

2.2. Ontologies to support the knowledge modeling

Ontology is an object of Artificial Intelligence that has become a mature powerful conceptual tool of Knowledge Modeling [5]. It provides a coherent base to build on, and a shared reference to align with, in the form of a consensual conceptual vocabulary on which one can build descriptions and communication acts.

Knowledge that is created in engineering projects needs to be defined precisely in order to be useful in an information system. Ontology provides a vocabulary and a semantic that enable the processing of knowledge related to a specific domain. Ontology is a set of items and their specific meanings. It gives definitions and indicates how concepts are connected to each other. These connections form a structure on the defined domain and clarify the possible meanings of the items [66]. Therefore, a domain ontology includes the specific concepts of a given domain. It describes the entities, properties and the way they can be related to each other. These ontologies are meant to be re-used in the same domain, in new but similar applications. These ontologies are said to be contextual [52] when the concept properties evolve according to the situation. We will use in our approach contextual ontologies in order to describe the context where the knowledge was captured.

2.3. Interests of the ontologies in MAS

The idea of using domain ontologies in an agent system aims at reusing pieces of the domain Knowledge to lead agents to share their information. Indeed in a MAS, several agents interact or work together to carry out common goals [63]. The coordination between agents depends on the process and knowledge they use to achieve their global goals. The domain ontology provides a section of the knowledge world that is essential for the agent to carry out its tasks [8].

Some research works like Buccafurri [7] and Wooldridge [71] use the ontology to give to the agents an internal representation of both interests and behavior of their associated human users. Other works use ontology to help agents to choose the most promising agents to be contacted for knowledge-sharing goals [5,8,72]. Generally, these systems have been designed to prevent the agents from having access to the ontologies of other agents; they ensure an individualistic view of agents' societies. This is the viewpoint of most of the so-called BDI (Beliefs Desires Intentions) approaches [31,55]. Another interesting approach that has been adopted to design MAS is related to the agent community, where agents automatically build their ontologies by observing the users' actions [4,67]. Indeed, the agents are able to automatically extract logical rules that represent the user behavior and/or causal implications among events due to the definition of the user interests described with the ontology.

In addition, Guerin and Pitt [37] and Singh [60] propose to design their MAS in adopting a "social" view of agent communities, where it is assumed that the ontology of each agent is, even partially, accessible for each other agent. In the next section, we will propose the architecture of a MAS using a common ontology for all agents [17]. Then, we will present the mechanism of knowledge distribution between agents based on a semantic analysis.

2.4. Ontology editors

There are many ontology editors which consist of creation and the manipulation of ontologies like Protégé 2000 [28] which is an open source framework from Stanford University. NeOn Toolkit [15] is another ontology editor which provides comprehensive support for the ontology engineering life-cycle [33]. It is especially suited for heavyweight projects (e.g., multi-modular ontologies, multi-lingual, ontology integration, etc.). OBO-Edit [7] is a java-based system developed by the Gene Ontology Consortium for editing biological ontologies and Anzo for Excel [3] is a system that generates ontologies from Excel spreadsheets. PoolParty [58] is a thesaurus management system and a SKOS editor but this system use DBpedia to assign categories to concepts and auto-populate the thesaurus based on data from DBpedia. Knoodl [20] is an ontology editor, wiki, and ontology registry. Supports creation of communities where members can collaboratively import, create, discuss, document and publish ontologies but all content in Knoodl is organized into communities so user must request permissions from administrator to be able to view or edit a vocabulary which is a combination of an OWL based ontology editor and a wiki [38]. Oherwise, HOZO [25] creates heavy-weight and well thought out ontologies and Swoop [18] is an OWL Ontology browser and editor from the University of Maryland, it is a web browser which means user must enter an URI of the ontology in an address bar to load the ontology and this is done inline with the HTML renderer [32].

As we work in industrial environment, we are obligate to make our system with friendly and easy GUI especially because the users of the system will be mechanical experts. Our ontology editor must be adapted to the mechanical domain. Moreover, our ontology editor must be able to manage several ontologies and deal with the alignment problems. We present in the next section a brief state of the art of the alignment technics.

2.5. Alignment technics

In our approach we have to calculate the similarity between two concepts in order to identify the similar concept or closed concept from a ontology to another. This work will allow us to optimize our knowledge research inside the enterprise information system [41]. Rahm and Bernstein [26] propose the following classification of different alignment technics:

Terminology methods: they compare the labels of the various entities. They are decomposed into:

Syntactic methods: they conduct the correspondence through the measures of dissimilarity of strings.

(Lexical methods: they perform the correspondence through lexical relations (synonymy, hyponymy …).

Structural methods: they are based on comparing the structures of entities. We distinguish:

The methods of comparison of internal structures: they are based on the internal structure of entities (cardinality of attributes, transitive properties…).

The methods of comparison of external structures: a comparison of similarity between two entities of two ontologies depends on the relationships of these entities with other entities. Types of relationships exist: taxonomic relationships, the mereological relationship, etc.

Semantic methods: they are based on the entities model used to validate and justify the alignment results. These are deductive methods. Among these techniques, we quote the propositional satisfiability which tends to find alignments by transforming ontologies and the alignment request in propositional formulas and then to verify its validity.

In this paper, we applied the terminology method to search the correspondences between concepts. We used the syntactic method “edit distance” formulated by Levenshtein. It measures the minimum number of token insertions, deletions, and substitutions required to transform one string into another using a dynamic programming algorithm.

We evaluated two formulas to compute the similarity between two concepts which are based on the levenshtein edit distance. These formulas compare two concepts C1 and C2. We analyzed these two formulas and we selected at the end the best for our case. We present below this analysis and the reason of our choice.

$$\textit{The first one } [31] \textit{ is } SM(C1,\ C2) = 1/1 + ED(C1,\ C2) \tag{1}$$

$$\textit{The second one } [23] \textit{ is } SM(C1,\ C2) = max(0,\ minLen(C1,\ C2) - ED(C1,\ C2)/minLen(C1,\ C2) \in [0,\ 1] \tag{2}$$

SM in formula (A) returns a degree of similarity between 0 and 1, where 1 stands for perfect match and zero for bad match. It considers the number of changes that must be made to change one string into the other and weighs the number of these changes against the length of the shortest string of these two.

We can conclude that we get 0 in formula 1 if the minimum length of the two concepts is less than the edit distance, which is not convenient. For that reason we chose the formula (B). The algorithm is applied by the multi agent system, which will manage the system. In the next section we present different multi agent systems used in Knowledge Management approaches.

3. Overview of our approach

Dastaviz in [15] highlights the interest to model the value of know-how and knowledge in an organizational perspective

In this domain, Maier in [39] presents a Knowledge Management System (KMS) as a platform which can support the functions of knowledge creation, construction, identification, capturing, acquisition, selection, valuation, organization, linking, structuring, formalization, visualization, distribution, retention, maintenance, refinement, evolution, accessing, search, and application. This KMS uses a variety of technologies designed to enhance knowledge storage and knowledge communication/transfer. Grundstein proposed a knowledge management life-cycle, where, according to him, "in any operation of knowledge capitalization, it is important to identify the strategic knowledge to be capitalized". His cycle is divided into four facets which are: identify, formalize, value and update [14].

We are inspired from these two systems and we created our own cycle of knowledge capitalization from four facets, which are: Definition, extraction, validation and reuse to fit with our objective. Thus the OCEAN platform aims to support our Knowledge Management process. The systems aims to helps professional actors to collaborate in order to share knowledge from different business units. Indeed, this platform allows business expert to describe his needs by creating an ontology. Afterwards, the multi-agent system transforms the ontology into queries in order to extract adapted information from databases of professional software tools (project management tools, calculus software, CAO tools, etc.). We have proposed a set of transformation rules to transform the ontologies into SQL queries without losing the semantic meaning of the knowledge base.

After extracting information resulting from the application of these queries on the professional databases, the system annotates and stores it in RDF base. The annotation of information is a key factor of the system since it provides the context where the information was created (information source) in which project, and from who.

Afterwards, business experts can consult a semantic wiki, which formats the information inside the RDF base in order to broadcast it inside wiki pages. Thus the professional actors can validate and/or evaluate this information by approving it, modifying it or rejecting it. The semantic wiki allows the knowledge diffusion and to obtain a feedback of business actors.

The problematic of creating an ontology by the business experts illustrated by the fact that each expert creates the concepts of ontology in the natural language. Thus the same concept can be described differently by different business expert and can be different from the data stored in the databases. For this reason this paper will describe our methodology to deal with this problematic and to help business experts to create their ontology. The expert must take into account the help of the system for better results. These results will be stored in RDF files. Information in these RDF files will be validated by experts using a semantic wiki in OCEAN platform to transform information into knowledge.

In the next section we will present the OCEAN platform which is composed by the three packages based on multi-agents system: knowledge definition, knowledge extraction, knowledge evaluation and knowledge reuse.

3.1. A MAS architecture to support our knowledge management approach

Knowledge agents are a part of cognitive and intelligent agents. They constitute a coupled network of agents that worked together to achieve the same objective i.e. to support the knowledge management process by providing full range of functionalities like extracting, annotating, storing, updating and sharing knowledge [49,50]. The MAS architecture is a structure of an agent network with different types of agents and different relationships between them [28,71]. The OCEAN architecture starts from the highest level of abstraction with the description of agent societies and goes down to the description of the roles, interactions and responsibilities of the agents.

The proposed approach to design a MAS is based on an organizational approach like the A.G.R model used in AALAADIN [22], OperettA [51] and methodologies like GAIA [71] or TROPOS [9] or RIOCC [46]. Thus the OCEAN architecture is tackled as a human society in terms or role, skill and relationships.

The main objective of the OCEAN system is to manage heterogeneous and distributed knowledge coming from different information sources and used by professional actors. The second objective is to permanently evaluate this knowledge in order to delete obsolete knowledge. The third objective is to broadcast the knowledge by using a semantic wiki. Considering these objectives we have defined three main functionalities for the system:

To allow users to describe their knowledge domain with a semantic approach (i.e. a characterization of concepts and their relations).

To extract knowledge from different information sources.

To broadcast, update and validate the knowledge base with the users in order to avoid broadcasting wrong information.

These three functionalities are implemented in three packages “Knowledge Definition”, “Knowledge Acquisition” and “Knowledge Diffusion”. The Knowledge Acquisition package aims to create ontologies by the professional experts in order to build knowledge bases. Afterwards the Knowledge Acquisition package uses these ontologies and transform it into queries by using a transformation rules [21] in order to extract data from the different databases or sources files of the software used by the project teams. Data resulting from these queries are annotated and stored in RDF files. These data becomes information since the system provides a context for each data by using the annotations. The next step consists in broadcasting and evaluating information through a semantic wiki. We identified three societies of agents (Fig. 1). The following agent types have been identified:

The Ontologist agents (OA): they manage the different ontologies (knowledge models) build by the users. The ontologies created by users are a set of concepts connected by relationships. The proposed interface looks like protégé 2000 but has in addition a representation of mechanical design patterns. The agents help the professional actors to build domain ontology by visualizing a mechanical product. We will detail this package in the next section with an experiment.

The Interpreter agents (IA): these agents “transform” the ontologies created into queries by applying several rules, which are detailed in section 6. The transformation from a ontological model to a query model is made by MDE (Model Driven Engineering) transformations. Data resulting from the process of extraction are then sent to agent to annotate these data which means put them in a specific context to become knowledge. This will be useful in the case of reuse of knowledge. Annotating data means adding information, details, comments and other semantics concerning data in order to contextualize it and transform it into Knowledge. The data annotated (knowledge) will be stocked in RDF files [27].

The Announcers agents (AA): in this phase agents had to validate knowledge and evolve the knowledge base. The knowledge is already stored in RDF files and structured in an organizational memory. Alhashem in [1] highlights the importance to build organizational memory in an innovation process. The organizational memory is composed by the information extracted from the enterprise databases according to the ontologies concepts defined by the experts and the annotations made by the agents which represent an organizational context (name of the actor, role of the actor, the project name, the software where was created the information, etc.) This organizational memory is accessible to business actors under the form of a semantic wiki. The semantic wiki allows: the structuration of knowledge, his diffusion and the obtaining of feed-back from business actors. Indeed, each actor can confirm, reject, or modify Knowledge using the semantic wiki. If some knowledge are rejected by the majority of actors then they will be deleted of the knowledge base.

3.2. How to facilitate the ontology building with OCEAN

We present in this section the Knowledge Definition Package of OCEAN lead by the Ontologist Agents. The expert identifies the knowledge needed for the project by creating an ontology. But the major problem here is that expert can enter a concept, which does not match with the structure of the database. To face this problem we propose to assist the expert to enrich his ontology with concepts, which are semantically closed from the concepts he has chosen. We describe below the Ontologist agents and their mechanisms to assist the expert during the ontology building process.

3.2.1. How to facilate the building of adapted ontologies: the mission of the ontologist agents

OCEAN is a multi agents system where different societies of agents communicate together to support the knowledge management process. Each agent has a role i.e. an abstraction of a behavior [22], a set of actions that the agent does inside a cooperative problem-solving situation. Each agent is able to play one of several roles. The intelligence of the agent is related with its capability to select the most appropriate role to play in a problem-solving state.

We define the identity of an agent by the name of its group (type), its roles, its interactions, its interface and its goals. Table 1 describes the Ontologist Agents identity:

The OA aims to help human experts to build adapted ontologies by proposing concepts which are close semantically with the concepts choose by the experts. The fact to enrich semantically the ontologies increases the quality of knowledge extraction in the information landscape.

As Fig. (2) shows, the ontology creation is supported by a six steps process:

Step 1: create an ontology in OCEAN once in the beginning which means give it a name. This ontology will be stored in the database of ontologies. The OCEAN interface (Fig. 2) allows building a concepts tree, to define the relationships between these concepts and to visualize the object that the experts want to model. As soon as a new ontology is created, a new Ontologist Agent is also created. It has the role of “ontology creator” at the beginning. This role allows it to build the ontology in the OWL format;

Step 2: create a concept “C1”. In the same time, the Ontologist agent takes the role of “Similarity Researcher” and proposes some other concepts after to have researched some semantic similarities in the WordNet ontology [44,64]. Indeed WordNet provides linguistic relationships between concepts like the synonymy (words that have similar meanings, e.g. happy and glad), the hypernymy (it refers to a hierarchical relationship between words. For example, furniture is a hypernym of chair since every chair is a piece of furniture (but not vice-versa)), the hyponymy (it is the opposite of hypernymy. Dog is a hyponym of canine since every dog is a canine) and the meronymy (it refers to a part/whole relationship. For example, paper is a meronym of book, since paper is a part of a book). For this reason the Ontologist Agent uses the lexical database "WordNet" to detect semantical relationships and to enrich the ontology. WordNet is a large lexical

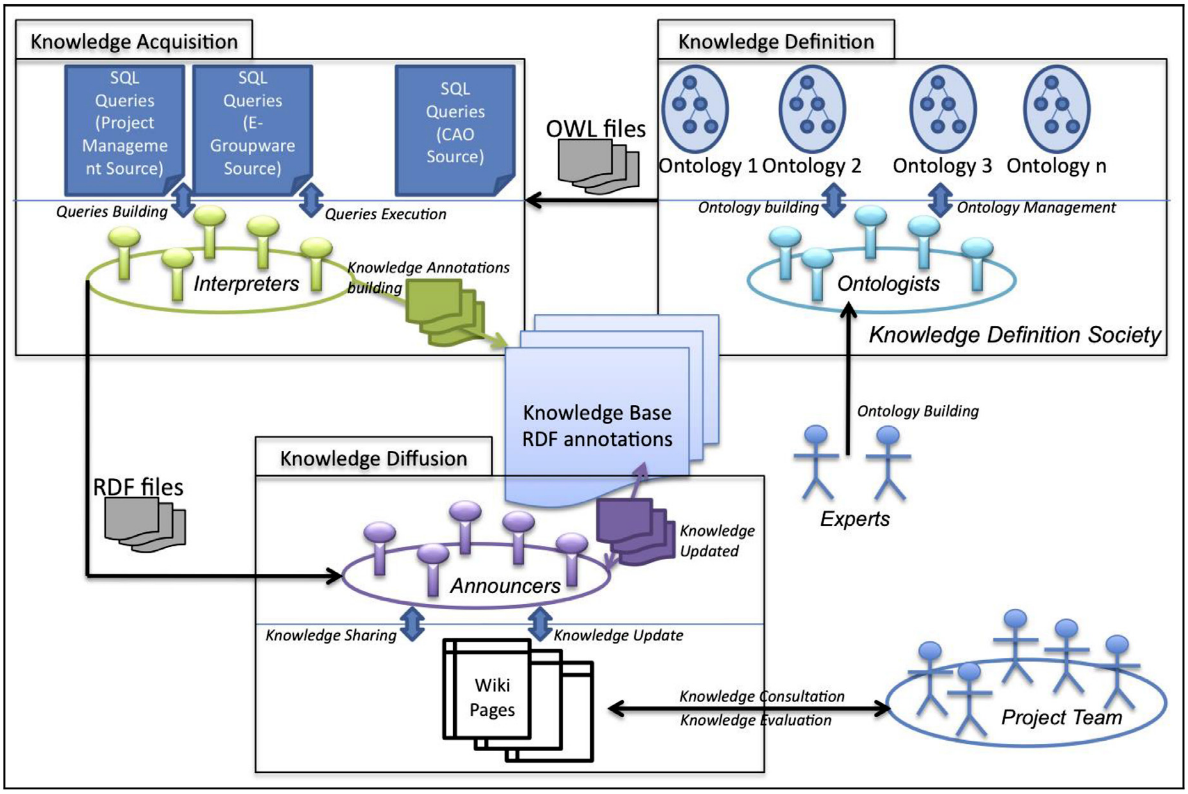

Fig. 1. OCEAN Agents architecture [48].

Table 1
Ontologist Agents identity.

Group	Ontologists (OA)
Role(s)	Ontology Creator, Similarity Researcher
Interactions	Ontologist Agents, WordNet Ontology, Human Expert, Interpreter Agents
Goals	To build reliable ontology to obtain the best results in knowledge extraction
Number	There is one OA for one ontology
Interface	Ontology builder

database of English, developed under the direction of Miller [44]. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations.

Step3: the OA of the new ontology communicates with all the OA i.e., with all the agents which are managed the others ontologies. They will verify if they are in their ontologies some similar concepts. In a case of a similarity the OA share all the concepts which have semantic relationships (hyperymy, meronymy, synonymy).

Step 4: the obtained results from WordNet and from the other ontologies are proposed to the expert. With these results he can chose to enrich his ontology semantically. Fig. 3 presents a set of proposals made by the agent OA to enrich semantically the ontology "CyclingOnto".

Step 5: the OA takes the role "Ontology creator" and builds the ontology with all the concepts and their relationships validated by the expert. We will describe in the Section 4, how the agents manage the research of similar concepts by using WordNet. The OA finalizes its actions by sending the ontologies to the Interpreter Agents in order to start the knowledge extraction from the different information sources (databases, XML files, …) of the enterprise network.3.2.2. How to extract knowledge for the user's databases: the mission of the Interpreter Agents

The Interpreters Agents (IA) aim to extract the heterogeneous and distributed information from different sources (project management software, CAD platforms, PLM platforms, etc.). They transform the ontologies provided by the Ontologist Agents into queries in order to send request to the different software databases. Afterwards, the IA annotated the obtained data, which are the results for the requests.

The annotations allow transforming data to information. Indeed, Weggeman [70] defines Information as a data structured according to a convention i.e., a data with a given context. Table 2 defines the identity of the Interpreters Agents.

As we shown in the Fig. 4, the knowledge extraction process lead by the IA is based on three steps:

Step 1: the first task of the IA is to transform their ontology into SQL queries. This mechanism used eleven transformation rules, which allow transforming a concept of the ontology to a table for the database and to covert semantic relationship into database relationships between tables. The role of the IA is "Query Builder" for this task. We describe in detail how the agent extract knowledge form the users' databases by using the ontologies in the Section 4.

Step 2: the IA take the role of "Data Extractor" and executes the SQL requests obtained after the ontology transformation into SQL queries. The IA executes the request in the different professional software tools databases. The results of this approach (Step 1 and step2) are exposed in our previous paper [45].

Step 3: the IA have the role of "Annotator" and annotates each results of the execution of the SQL queries. Thus it builds the context for each information. A context is described by the name of the software tool where the information was extracted, the name of the creator, the creation date, the name of the product and of the project. There are two parts in a annotation file, the context section and the knowledge section. The knowledge section is built according to the structure of the ontologies. Fig. 4 shows the example of a cycling knowledge base. The figure presents how the IA build the knowledge section according to a ontology. All the results and their annotations are stored in RDF files. Afterwards these files are shared with the Announcer Agents.

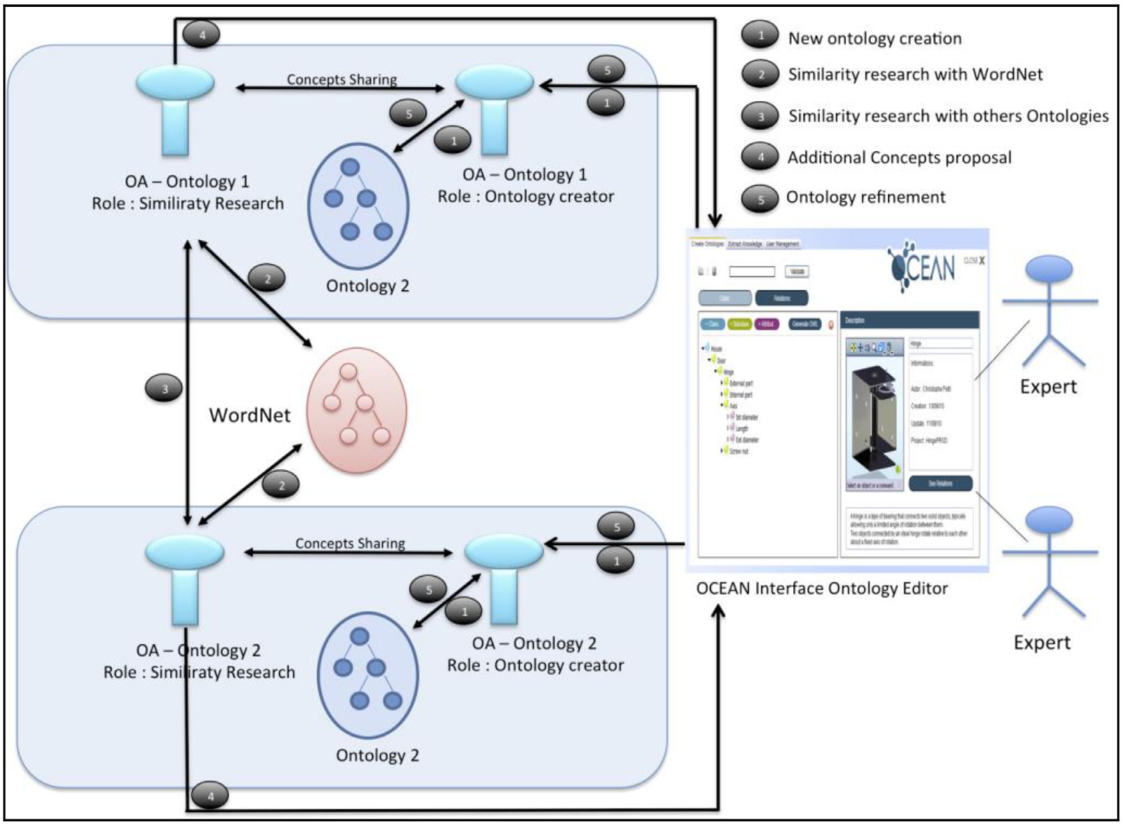

Fig. 2. The ontology building lead by the Ontologist Agents.

3.2.3. How to broadcast and share knowledge; the mission of the announcers agents

We present in this section the Knowledge Diffusion package leads by the Annoucer Agents (AA). The AA aim to allow professional users to validate, broadcast, evaluate and update knowledge. To achieve these objectives, they manage a semantic wiki called Wiki-K. Indeed, a semantic wiki seems to allow to broadcast knowledge by publishing articles in wiki pages, to validate and update knowledge by allowing users to add evolution for the wiki pages. Table 3 presents the identity card of these agents dedicated to manage the semantic wiki (Fig. 5).

As Fig. (6) shows, the knowledge broadcast is supported by a four steps process:

Step 1: the AA agent has the role of " WikiPages creator". It generates Wiki pages from the information extracted by the Interpreters agents and stored inside RDF annotations files. The work of the AA is based on the work of Jovanovic [24] and Souzis [62] in order to build the XSLT processor allowing to use the RDF files content and to transform them into web pages. The semantic relationships described in the RDF files are used by the Announcers Agent to generate relevant hypertext links to navigate for a wiki page to another. For example the

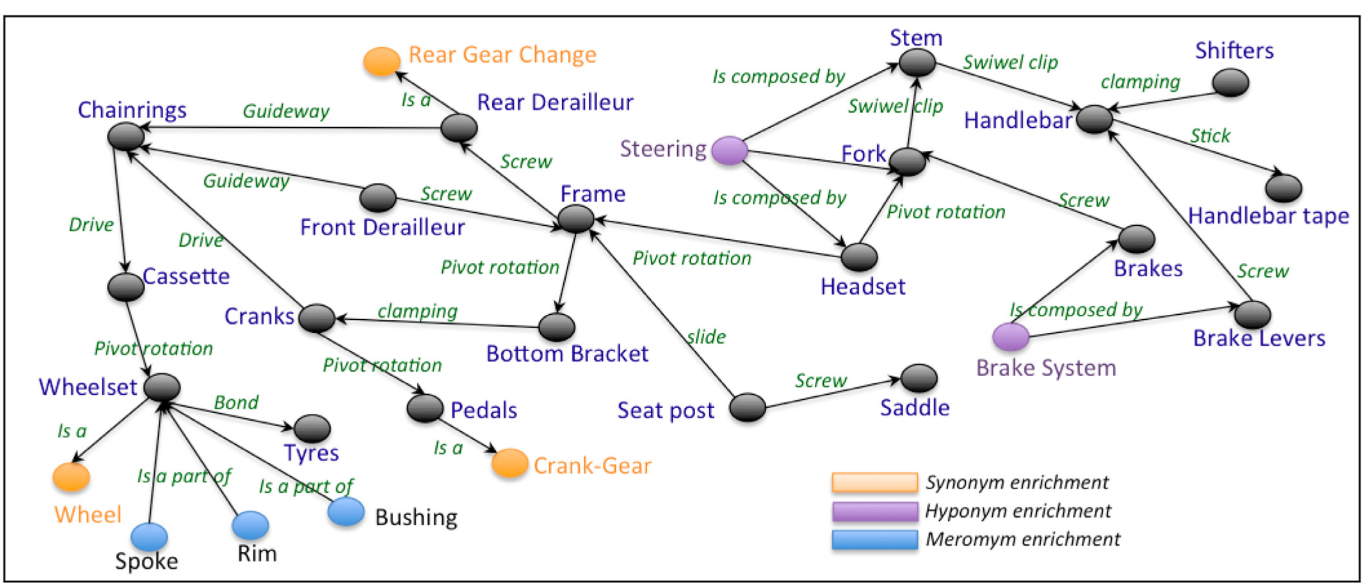

Fig. 3. The ontology «CyclingOnto» enriched by the agent OA.

Table 2
Interpreters agents identity.

Group	Interpreters (IA)
Role(s)	Query builder, Data Extractor, Information annotator
Interactions	Ontologist Agents, Entreprise Dataabases Softwares, Announcer Agents
Goals	To extract and annotate knowledge from the different Information sources in the entreprise network.
Number	There is one IA for one ontology i. e. for each knowledge structure.
Interface	None

AA automatically build links between "bicycle", "bike" and "cycle" from the "CyclingOnto" ontology. Then, thanks to the ontology, the bicycle wiki page has to links with the bike and the cycle wiki pages.

Step 2: a Professional Actor can search information by using Wiki-K. The AA help users to find the Knowledge they need. In this situation, they have the role of "Knowledge Researcher". The research is made in the Wiki-K interface. A search page view is shown in Fig. 7. From keywords the users request the knowledge. The AA build SPARQL queries from the users request. SPARQL [59] is a W3C recommendation language to query RDF. The list of wiki pages is generated in the same page. Each result corresponds to a knowledge described in a wiki page. Fig. 7 shows the Wiki-K interface with a result of a research for the keyword "Hood".

Step 3: the AA follow the action of the users inside the Wiki-K, they have the role of "Knowledge Updater". The Professional Actor has the possibility to simply read it, or to update it or to disapprove it. The AA add for each wiki page an evaluation according to its maturity (number of stars describing the number of evaluations) and its percent of positive evaluation. Indeed when a professional actor approves or modifies an article, he assigns a positive evaluation for this article. Moreover when he disapproves, the article obtains a negative evaluation. AA calculate the knowledge maturity by positioning a percentage of positive evaluation and a number of stars. Thus knowledge which has just been created has one hundred percent of positive evaluation. Progressively with the evaluations attributed by users, the percentage can decrease if the article obtains negative evaluations. In addition wiki page which has a score in lower than twenty five percent of positive evaluation, is deleted in the knowledge base. Indeed the AA are able to automatically

Table 3
Announcers agents identity.

Group	Annoucers (AA)
Role(s)	WikiPages Creator, Knowledge Researcher, Knowledge Updater
Interactions	Human Actors, Interpreter Agents
Goals	To validate, broadcast, update and evaluate knowledge
Number	There is one AA for one ontology
Interface	Semantic Wiki

delete knowledge which are become obsolete or are not a consensus inside the community of experts.

In the next section we describe in details how the Interpreters agents extract knowledge from different users' databases (Tables 4 and 5).

3.3. Interaction between the different agent groups

The multi-agent system OCEAN is composed by three groups of agents: the Ontologists, the Announcers and the Interpreters. Each group of agents has its own objective such as "to assist the ontology building process", "to collect knowledge" and "to broadcast and share knowledge". To reach their goals each group of agents need different types of information. The group of Ontologist Agents is always active. When it builds new domain ontology, it sends the description of this ontology via OWL files to the group of Interpreters agents. This message activates the Interpreters. Then the Interpreters agents collect the knowledge from the different databases of the enterprise. When this objective is reached they send the collected knowledge to the Announcers agents. The Announcers builds the knowledge base composed by RDF files and share the knowledge with the professional actor by using Wiki-K. Each time ta professional actor makes a research with an unknown concept, the Announcers send the concept to the Ontologists. These agents will propose the new concept to the expert in order to associate it to an domain ontology or to delete it. Fig. 8 describes the interactions between the different groups of agents.

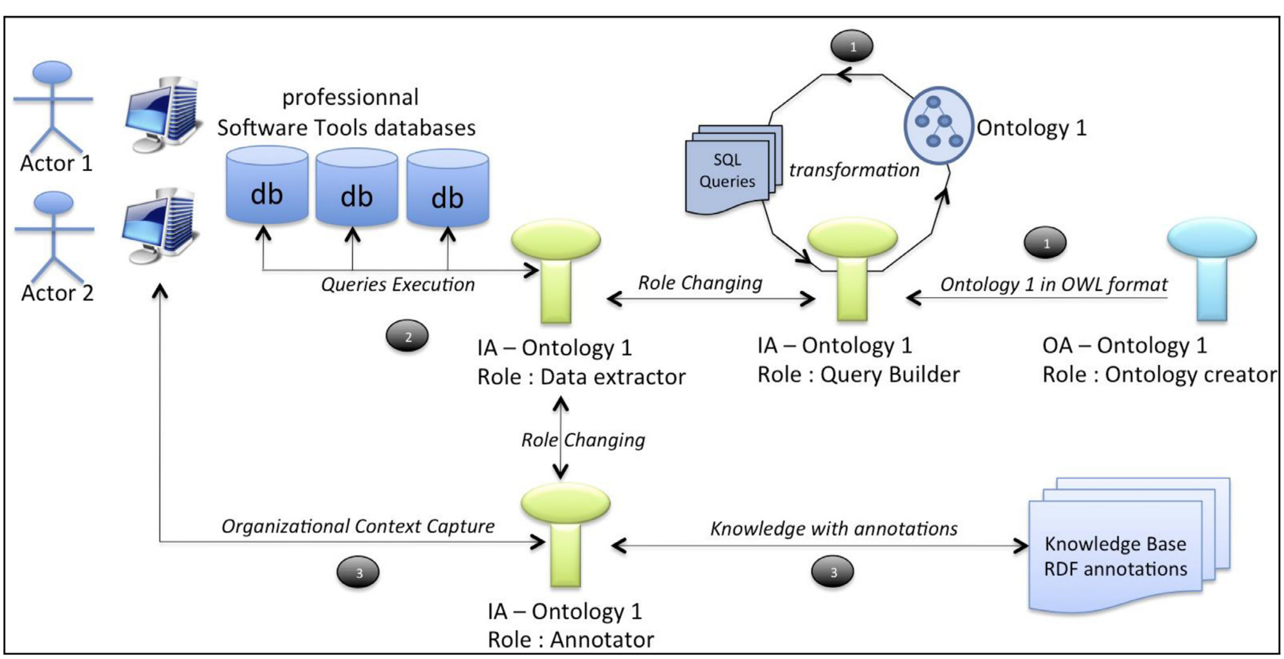

Fig. 4. Knowledge extraction and annotation lead by the Interpreter agents.

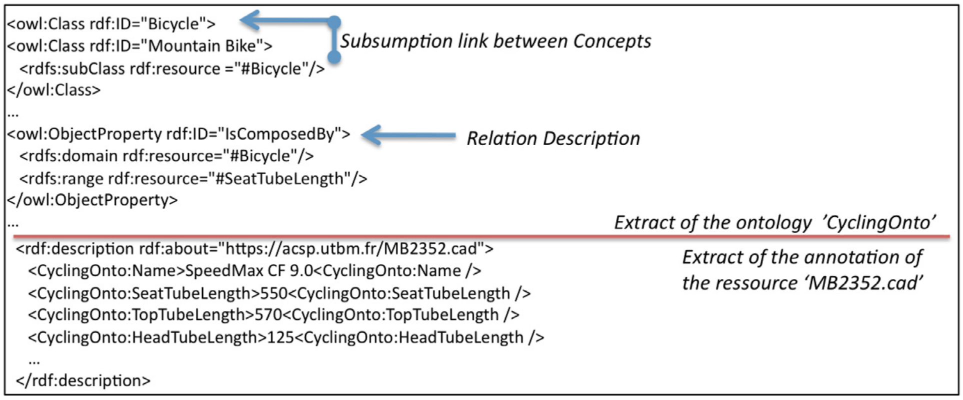

Fig. 5. An example of an annotation built from the ontology 'CyclingOnto'.

4. Our approach to extract knowledge from the users' databases

4.1. The knowledge extraction approach

To extract knowledge from the database of business applications the Interpreters Agent have to apply an algorithm to transform OWL ontologies in SQL queries.

The transformations between models was used recently in the knowledge management research fields to build ontologies from relational databases [12,57] or to transform ontologies in SQL queries [21,61]. Our approach is based on the result of these research works, and aims to build SQL queries from the ontologies stored by the Ontologists agents. The Interpreters agents apply some corresponding rules to detect similarity between the structure of the databases tables and the ontologies. Tables 1 and 2 present the correspondence between database components/properties and ontology concepts/relationships.

We have experimented transformation rules with four different information sources (an e-Groupware, a project management system, a risk analysis system and a CAO platform). The Interpreters apply 11 transformation rules to go from the ontology to the SQL model. These rules are described below:

R1 A class is a table

R2 If there is an association which is surrounded by the cardinality* on both sides, we search the primary keys that correspond to these 2 tables. Once we found it, we store the name of this table in the list of tables and the primary-foreign key in the relationship table.

R3 If there is an association which is surrounded by a * on the first side and a 0 or 1 on the other side we look for the primary key of the table next to *. This key is added as foreign key in the table next to 0 or 1

R4 Store the name of tables in a table

R5 Search the primary and foreign keys of the tables in the list (the relations between them) and store these relations in a table.

R6 Forming conditions: we consider the range of each Data type Property

a. If it is an attribute positive Integer then the condition is > 0
b. If it is a Data Range which includes a list, so attribute in [value1,

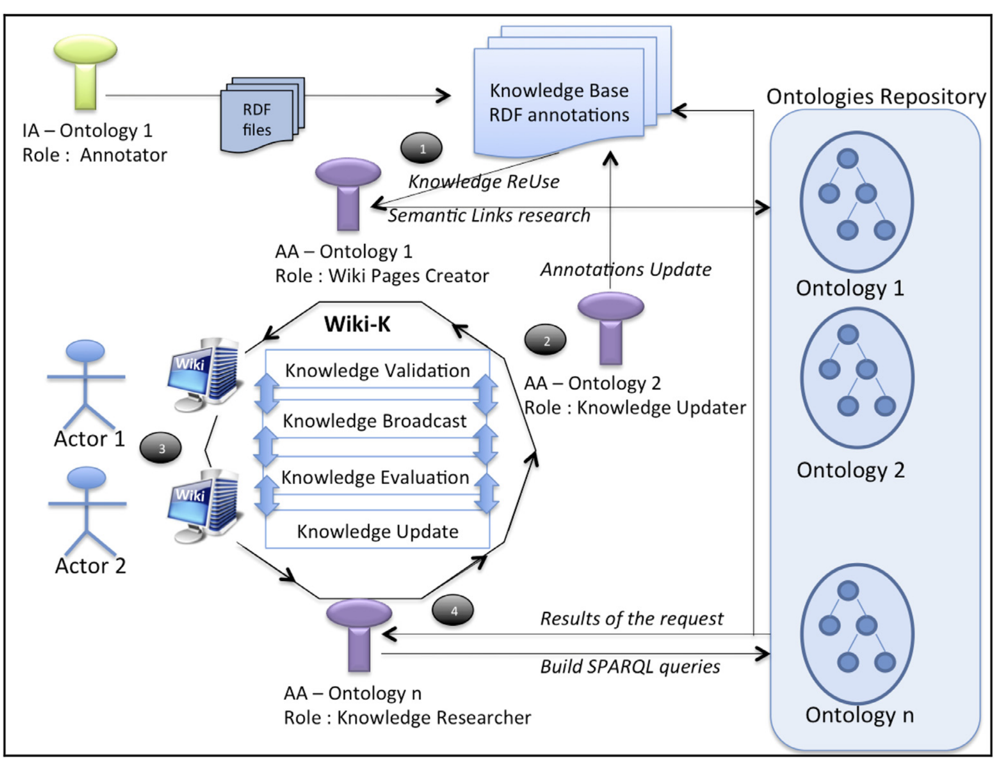

Fig. 6. Knowledge validation, broadcast, evaluation and update lead by the Interpreter agents.

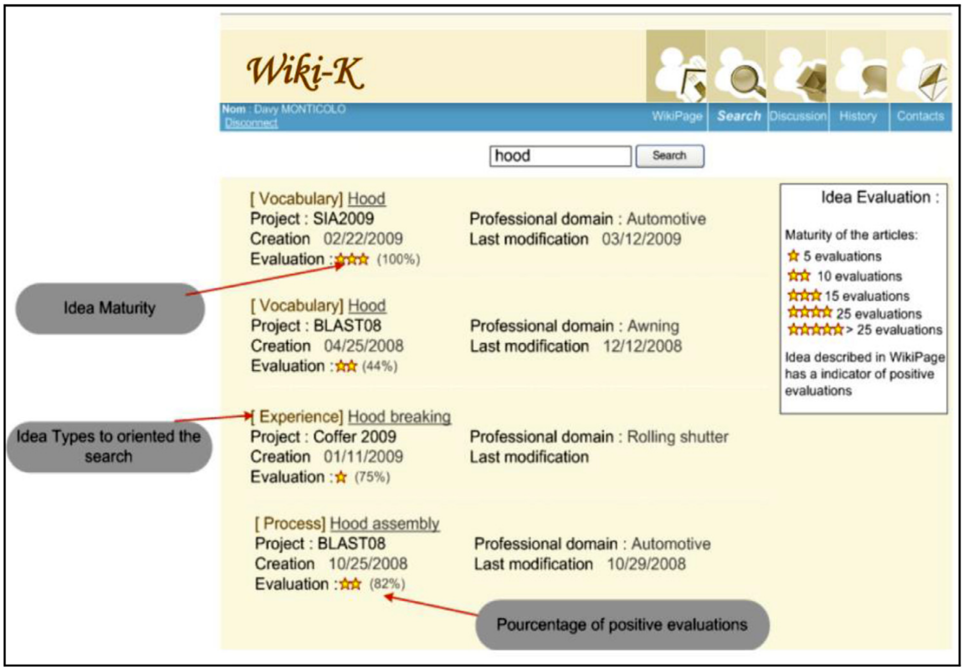

Fig. 7. An example of an annotation built from the ontology 'CyclingOnto'.

Table 4
Component/concept correspondence.

Database component	OWL concept
Table	Class
Column	Functional Propoerty
Row	OWL individual
Column Metadata:	OWL property restriction:
Data Type	-AllValues From restriction
Mandatory/Not nullable Nullable	-Cardinality () Restriction
	-maxCardinality() Restriction

Table 5
Property/relationship correspondence.

Database property	OWL relationship
NOT NULL	owl:minCardinality rdf:datatype="& xsd:Int"1/
UNIQUE	owl:InversFunctionalProperty
CHECK	owl:hasValue
FOREIGN KEY	owl:objectProperty

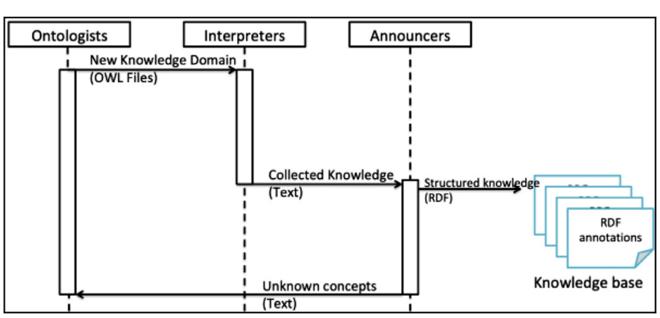

Fig. 8. Interaction between groups of agents.

value2 ...]
c. If there are restrictions on DatatypeProperty and it is not a cardinality restriction as Has Value, so the condition is: check attribute = value

R7 Inverse functional property is the "Distinct" constraint

R8 Required property mean that the fields are not null (not 0 or not = "")

R9 Symmetric property is a recursive table R10 Store conditions in a table.

R11 Build the query

Select all DatatypeProperty by the first letter of the domain table. e.g.: p.nom, and followed by a comma

From all tables stored in the table (R4) followed by the first letter as naming the table by a letter

Where all the relationships stored in a table to make the connections between the tables + all conditions stored in the table separated by "and".

We illustrate the mechanism used by the interpreters for the knowledge extraction with an ontology concerning the cycling domain. The cycling ontology describes the cycling world. The ontology defines a vocabulary and a semantic to structure, organize, detail all the characteristics of a bike, all the roles of the professional actors during the development project of a new bike and all the processes used to develop and industrialize a bicycle.

This ontology is transformed into an SQL query which will return information from business applications. For example if business experts want to know how to design a 'SpeedMax CF' bike size M so the *Interpreters* agents transform the cycling ontology (Fig. 9) in specifying these details. The cycling ontology will be transformed into an SQL query such as the example below. The information returned with this query becomes knowledge and stored in RDF files as shown in Fig. 1 in the annotation part.

By using those rules, we have succeeded in extracting 58% of the concepts defined in the ontologies. It is more than previous work like in [4] but we have to more improve this result. Indeed it depends on the structure of databases. If the database has not relevant relations we obtain a multitude of results which are not relevant.

The loss of semantics is due to the fact that some of the concepts and relations defined in the ontology have no equivalence in the database (table or relations). In the SemKnow platform, the professional actors build their own ontology from their professional expertise. We observed that when actors had a good acquaintance with the professional software tool he/she used, then he/she properly defined ontology with

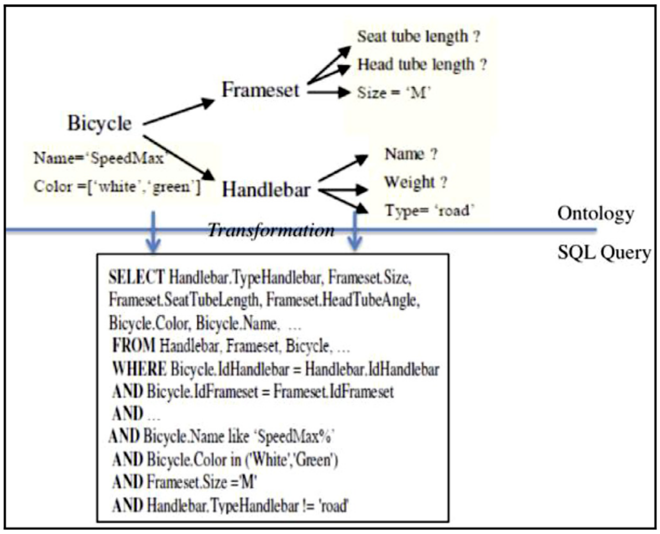

Fig. 9. Example of SQL Query built from the Cycling.

concepts close from the database structure and the agents obtained good knowledge extraction results.

4.2. The industrial experiment and results

We have experimented our approach with bike designers. They have created the cycling ontology presented below with the OCEAN interface (Fig. 10).

We have made our experiments by requesting three existing databases of collaborative platforms, which manage product data related to bicycle. The Interpreters agents ensure the connection with the database and the knowledge extraction by executing the SQL queries.

The results of the executed queries on the three databases are not always relevant. It depends on the structure of database. The results are different if the database is relational (all foreign keys constraints exist), or if the database is non-relational (the relationships are lacking).

We present thereafter in Fig. 11 the results obtained if the extraction is made on a relational database and a no relational database. The example shows that, on the one hand, the number of results obtained and the number of conditions used are inversely proportional. On the other hand, we can notice the explosion in the number of results obtained when the database is not relational compared to the relational database.

In our example the difference between the relational database and not relational database is the absence of the relation Bicycle.IdHandlebar = Handlebar.IdHandlebar in the last one.

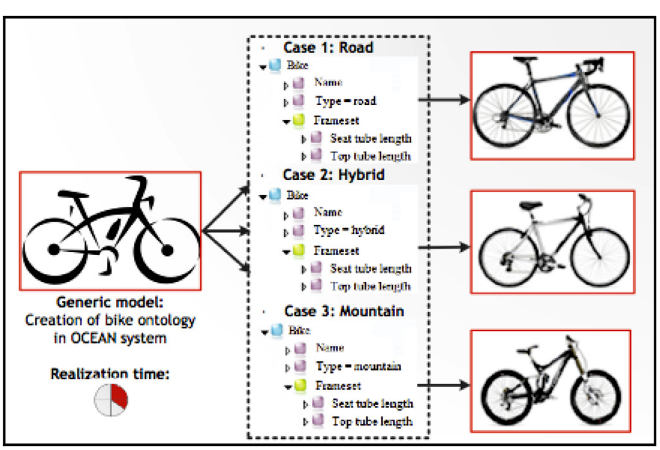

Fig. 10. The Interface to create the Cycling Ontology.

Number of results		
Conditions	Relational DB	DB NOT Relational
A,B,C,D	2	15
A,B,C	2	20
A,B,D	3	20
A,B	3	25
A,D	4	25
B	3	25
D	4	44
A	5	55

Good results Wrong results

List of Conditions	
A	Frameset.HeadTubeAngle >0
B	Frameset.Size ='M'
C	P.Type != 'road'
D	Bicycle.Color in ('White','Green')

Fig. 11. Results of queries on relational and not relational databases.

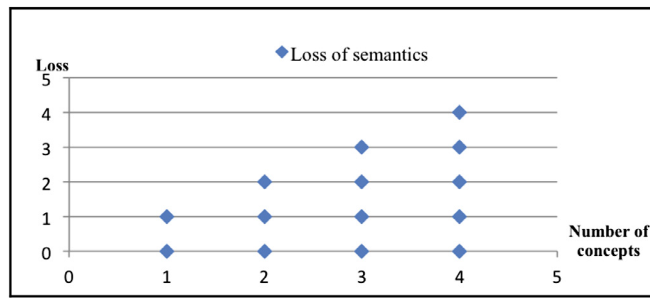

Fig. 12. Loss of semantics in our transformation models.

Thus, Fig. 12 illustrates the difference between the results obtained in both cases. The goal of our future researches is to find a method to optimize query to return a good result even if there is lacking in the relationships between tables.

We will have some loss of semantic concepts during the transformation as we realize it between two different models: ontology and SQL. The first is related to knowledge that involves not only syntax but also semantics and inferences, and the second is a language related to databases.

We firstly recognized that we have a loss of structure because not all concepts of ontology can have equivalences in SQL and therefore the quality of transformation should be analyzed. Second, we have done until now a certain number of transformation rules and we will continue to find more. Hence, we can conclude that we have really a loss of semantic if the ontology created contains concepts that are not taken into account in the transformation rules.

Fig. 11 shows the loss of semantics that we can have on a sample of four concepts. For example, we have four concepts in the ontology and which have equivalences in SQL managed by our rules. If the ontology created by the user includes four concepts and all these concepts have equivalent then there is no loss. Else, if two of the four concepts have equivalences and two not so we lost two semantic and so on.

So the ideal is to have an ontology that includes equivalent concepts to avoid losses. For these reasons we limit the user at the moment to create his ontology in OCEAN using just the concepts managed by our rules and in this way we neglect the losses. At the same time we continue our research on the transformation rules to allow user to use all the concepts of the ontology.

5. Discussion and conclusion

We presented in this paper our approach of multi-agents system to help business experts in the creation of their ontologies in OCEAN platform. The creation of ontology is the first step in our cycle of knowledge management. The goal of our KMS is to allow business experts to capitalize their heterogeneous and distributed knowledge by using a semantic and an agent approach.

The OCEAN multi-agent system helps business experts to create their domain ontologies which describe their needs as explained in this paper. MAS will transform then the ontology into queries in order to extract adapted information from databases. This information will be annotated and stored in RDF base. Afterwards, professional actors can consult a semantic wiki and validate and/or evaluate these information by approving them, modifying them or rejecting them. Our knowledge extraction process described in this paper use a semantic approach based on two steps, the semantic enrichment of the ontologies created by the experts and the use of transformation rules to transform a ontology into SQL queries. This approach allows us to extract approximately sixty percent of the good information in the users' database.

This research work allows us to identify several innovations. The first is about the ontology builder composed by the ontologist agents. This package is very appreciated by the professional experts. They can define and share their knowledge concepts by using the product editor and by enumerating every components of each product. The multi agent system "AdSiF" [43] also uses agents to generate an ontological view of a knowledge domain. This system does not allow the expert to associate a product view to a knowledge domain [29].

The second innovation is about to use a Wiki to share and to update knowledge. Our approach shows that the Wiki is very efficient to update and enrich the knowledge base from the professional actors' comments. Others research works [69,74] use Wikis to improve the knowledge management process but do not use the Wiki in an integrated platform. The advantage of out approach is that OCEAN is an integrated platform composed by three packages, which ensure the whole knowledge process [40]. However the extraction knowledge package has to be improved. The obtained results with the non-rational databases are not enough relevant. Some new research works use ontology-to-database mapping approaches [34,47]. We will complete our knowledge extraction process by adapting ontology-to-database approaches and query refinement approaches to increase the number of relevant information results from the extraction in order to make the OCEAN knowledge management system more efficient.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Alhashem, A. Shaqrah, Exploring the relationship between organizational memory and business innovation, IJKBO 2 (2012) 32–46.
[2] A. Abecker, A. Bernardi, L. van Elst, Agent technology for distributed organizational memories, Proceedings of the Fifth International Conference On Enterprise Information Systems, 2 2003, pp. 3–10.
[3] S. Abar, T. Abe, T. Kinoshita, A next generation knowledge management system architecture, Proceedings of the Eighteenth International Conference on Advanced Information Networking and Applications, 2004. AINA 2004, 2 2004, pp. 191–195 Vol.2.
[4] I. Astrova, N. Korda, A. Kalja, Storing OWL Ontologies in SQL Relational Databases, World Academy of Science, Engineering and Technology, 2007, p. 29.
[5] G. Beydoun, N. Tran, G. Low, B. Henderson-Sellers, Foundations of ontology-based mas methodologies, Agent-Oriented Information Systems III 3529 SpringerLink, 2006.
[6] M. Bonifacio, P. Bouquet, P. Traverso, Enabling distributed knowledge management. Managerial and technological implications, Nov. Inf./Inf. 3 (1) (2002) 23–29.
[7] F. Buccafurri, D. Rosaci, G.M.L. Sarne´, D. Ursino, An agent-based hierarchical clustering approach for e-commerce environments, Proceedings of the Third E-Commerce and Web Technologies (EC-Web 2002), Aix-en-Provence, France, Lecture Notes in Computer Science, Springer, Berlin, 2002, pp. 115–118.
[8] M.C. Bravo, J. Pérez, V.J. Sosa, A. Montes, G. Reyes, Ontology support for communicating agents in negotiation process, Proceedings of the International Conference on Hybrid Intelligent System, 2005, p. 12.
[9] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, J. Mylopoulos, Tropos: an agent-oriented software development methodology, Autonomous Agents and Multi-Agent Systems 8 Springer, 2004, pp. 203–236.
[10] I. Brigui-Chtioui, I. Saad, A multiagent approach for collective decision making in knowledge management, Group Decis. Negot. 20 (1) (2010) 19–37.
[11] P.A. Champin, Y. Prié, A. Mille, MUSETTE: modelling USEs and tasks for tracing experience, Proceedings of the from structured cases to unstructured problem solving episodes - WS 5 of ICCBR'03, Trondheim (NO), NTNU, Trondheim (NO), 2003, pp. 279–286.
[12] A. Chen, L. Liu, J. Shang, A hybrid strategy to construct scientific instrument ontology from relational database model, Proceedings of the 2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM). Presented at the 2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM), 2012, pp. 25–33.
[13] Y. Cheng, K. Chen, H. Sun, Y. Zhang, F. Tao, Data and knowledge mining with big data towards smart production, J. Ind. Inf. Integr. 9 (2018) 1–13.
[14] K.H. Chow, K.L. Choy, W.B. Lee, On the design of a real-time knowledge-based system for managing logistics operations, Intell. Syst. Acc. Financ. Manag. 14 (1–2) (2006) 3–25.
[15] A.H. Dastaviz, Integration of competing values and knowledge organisational activities in a new model, Int. J. Knowl.-Based Org. (IJKBO) 6 (4) (2016) 50–61.
[16] Y. Demazeau, From interactions to collective behaviour in agent-based systems, Proceedings of the First European Conference on Cognitive Science, Saint-Malo, 1995, pp. 117–132.
[17] R. Dieng-Kuntz, N. Matta, et al., Knowledge Management and Organizational Memories, Springer, 2002.
[18] J. Day-Richter, M.A. Harris, M. Haendel, S. Lewis, OBO-edit – an ontology editor for biologists, Bioinformatics 23 (16) (2007) 2198–2200.
[19] J.-.L. Ermine, Knowledge crash and knowledge management, Int. J. Knowl. Syst. Sci. 1 (4) (2010) 79–95 34.
[20] A. Ekambaram, A. Sørensen, H. Bull-Berg, O.E. Olsson N, The role of big data and knowledge management in improving projects and project-based organizations, Proc. Comput. Sci. 138 (2018) 851–858.
[21] C. Fabkam, L. Bellatreche, H. Dehainsala, Y. Ait Ameur, G. Pierra, SISRO, conception de bases de données à partir d'ontologies de domaine, Tech. Sci. Inf. 28 (2009) 1233–1261.
[22] J. Ferber, O. Gutknecht, A meta-model for the analysis and design of organizations in multi-agent systems, Proceedings of the Third International Conference on Multi Agent Systems (ICMAS98), Paris, France, July 1998, pp. 3–7.
[23] R. Fu, Z. Xin, Research on electronic commerce kms based on agent and ontology, Proceedings of the First International Workshop on Knowledge Discovery and Data Mining, Washington, DC, USA, 2008, pp. 190–195.
[24] J. Jovanovic, D. Gasevic, Achieving knowledge interoperability: an XML/XSLT approach, Exp. Syst. Appl. 29 (2005) 535–553.
[25] The protégé ontology editor and knowledge acquisition system. [Online]. Available: http://protege.stanford.edu/. [Accessed: 23-sept-2012].
[26] P. Haase, H. Lewen, R. Studer, M. Erdmann, O. Gmbh et al., The neon ontology engineering toolkit.
[27] Anzo express – Cambridge Semantics. [Online]. Available:http://www.cambridgesemantics.com/products/anzo-express. [Accessed: 23-sept-2012].
[28] F. Gandon, L. Berthelot, Dieng-Kuntz R., A multi-agent platform for a corporate semantic web, in: Proceedings of the AAMAS 2002, Sixth International Conference on Autonomous Agents, Fifth International Conference on Multi-Agents Systems, Ninth International Workshop on Agent Theories Architectures and Languages Eds Castelfranchi C., Johnson W.L., p. 1025–1032, July 15-19, 2002, Bologna, Italy.
[29] M. Giannakis, M. Louis, A multi-agent based framework for supply chain risk management, J. Purchas. Suppl. Manag. 17 (1) (2011) 23–31.
[30] M. Grundstein, J.P. Barthès, An industrial view of the process of capitalizing knowledge, knowledge management: organization, Comp. Methodol. (1996).
[31] N. Griffiths, M. Luck, Cooperative plan selection through trust, Proceedings of the Ninth European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99), 1647 Springer, Berlin, 1999, pp. 162–174.
[32] K. Jee, J.-.J. Yang, et al., Knowledge description model for mas utilizing distributed ontology repositories, in: Z.-Z. Shi, R. Sadananda, et al. (Eds.), Knowledge description model for mas utilizing distributed ontology repositories, Agent Computing and Multi-Agent Systems 4088 (2006) 773–780.
[33] A. Kalyanpur, E. Sirin, B. Parsia, J. Hendler, et al., Hypermedia inspired ontology engineering environment: swoop, Proceedings of the International Semantic Web Conference (ISWC), 2004, pp. 7–11.
[34] K. Munir, M.S. Anjum, The use of ontologies for effective knowledge modelling and information retrieval, Appl. Comput. Inf. 14 (2) (2018) 116–126.
[35] Intelligent Information Agents: Agent-based Information Discovery and Management in the Internet, in: M. Klusch (Ed.), Intelligent Information Agents: Agent-based Information Discovery and Management in the Internet, Springer, 1999.
[36] J. Kyengwhan, J.-J. Yang, Knowledge Description Model for Mas Utilizing Distributed Ontology Repositories, in Agent Computing and Multi-agent Systems 4088 SpringerLink, 2006.
[37] F. Guerin, J. Pitt, Denotational semantics for agent communication language, AGENTS '01: Proceedings of the Fifth International Conference on Autonomous Agents, ACM Press, 2001, pp. 497–504.
[38] Knoodl. [Online]. Available:http://www.knoodl.com/ui/home.html;jsessionid=5FA3B9D6B13A71F3D2832C75D1FCB1FA. [Accessed: 23-sept-2012].
[39] S. Lakhmani, P. Oppold, A. Rupp M., L. Szalma J., P.A. Hancock, Heterogeneous knowledge distribution in MMO player behavior: using domain knowledge to

distinguish membership in a community of practice, Comput. Hum. Behav. 55 (Part A) (2016) 455–467.
[40] R. Maier, T. Hädrich, Knowledge and Process Management, Wiley Online Library, 2006.
[41] A. Maedche, S. Staab, et al., Measuring similarity between ontologies, Proceedings of the Thirteenth International Conference on Knowledge Engineering and Knowledge Management, Ontologies and the Semantic Web, London, UK, UK, 2002, pp. 251–263.
[42] A. Mardani, S. Nikoosokhan, M. Moradi, M. Doustar, The relationship between knowledge management and innovation performance, J. High Technol. Manag. Res. 29 (1) (2018) 12–26.
[43] Mehmet F. Hocaoğlu, AdSiF: agent driven simulation framework paradigm and ontological view, Sci. Comput. Program. 167 (2018) 70–90.
[44] G.A. Miller, Ed. wordnet: an on-line lexical database, Int. J. Lexicogr. 3 (4) (1990) 235–312.
[45] D. Monticolo, I. Lahoud, E. Bonjour, Distributed knowledge extracted by a MAS using ontology alignment methods, Proceedings of the IEEE International Conference on Computer and Information Sciences, Kuala Lumpur, Malaysia, 2012.
[46] D. Monticolo, V. Hilaire, S. Gomes, A. Koukam, A multi agents systems for building project memories to facilitate design process, Int. J. Integr. Comput. Aided Eng. 15 (2008) 3–20.
[47] Zina Nakhla, Kaouther nouira, automatic approach to enrich databases using ontology: application in medical domain, Proc. Comput. Sci. 112 (2017) 387–396.
[48] I. Lahoud, D. Monticolo, V. Hilaire, S. Gomes, A metamodeling and transformation approach for knowledge extraction, Proceedings of the Fourth International Conference on Networked Digital Technologies, Dubai, UAE, 2012.
[49] E. Leclercq, M. Savonnet, P. Naubourg, Traitement des variabilités métier dans les systèmes d'Information biologiques, Proceedings of the XXXeme Congrès INFORSID, Montpellier, France, 2012, pp. 173–188.
[50] M.R. Lee, T.T. Chen, Revealing research themes and trends in knowledge management: from 1995 to 2010, Knowledge-Based Systems, (2011).
[51] D. Okouya, V. Dignum, OperettA: a prototype tool for the design, analysis and development of multi-agent organizations, Proceedings of the Seventh international joint conference on Autonomous agents and multiagent systems (AAMAS 08): demo papers, EstorilPortugal, May 2008, pp. 1677–1678.
[52] P. Pouquet, C-owl: contextualizing ontologies, Proceedings of the Second International Semantic Web Conference (ISWC2003), 20–23 October 2003, Sundial Resort, Sanibel Island, Florida, USA, Oct 2003.
[53] N. Rahman, Enterprise data warehouse governance best practices, Int. J. Knowl.-Based Org. (IJKBO) 6 (2) (2016) 17.
[54] I. Runar Edvardsson, S. Durst, The benefits of knowledge management in small and medium-sized enterprises, Proc. – Soc. Behav. Sci. 81 (2013) 351–354.
[55] A. Rao, M. Georgeff, Decision procedures of BDI logics, J. Logic Comput. 8 (3) (1998).
[56] R.M. René Peinl, SimKnowledge – analyzing Impact of knowledge management measures on team organizations with multi agent-based simulation, Inf. Syst. Front. 13 (2011) 621–636.
[57] H.A. Santoso, S.-.C. Haw, Z.T. Abdul-Mehdi, Ontology extraction from relational database: concept hierarchy as background knowledge, Knowl. Based Syst. 24 (2011) 457–464.
[58] T. Schandl, A. Blumauer, et al., PoolParty: skos thesaurus management utilizing linked data, in: L. Aroyo, G. Antoniou, E. Hyvönen, A. Ten Teije, H. Stuckenschmidt, L. Cabral, et al. (Eds.), The Semantic Web: Research and Applications, 6089 Springer, Berlin/Heidelberg, 2010, pp. 421–425.
[59] A. Seaborne & E. Prud'hommeaux (2006), SPARQL query language for rdf. Technical Reporthttp://www.w3.org/TR/2006/CR-rdf-sparql-query-20060406/, W3C.
[60] B. Singh. Interconnected roles (IR): a coordination model. Technical Report CT-084-92, MCC, (1992).
[61] E.F. Souza, L.E. Oliveira, R.A. Falbo, L. Vijaykumar, Using ontologies to build a database to obtain strategic information in decision making, in: A. Malucelli, M. Bax (Eds.), ONTOBRAS-MOST, CEUR Workshop Proceedings. CEUR-WS.org, 2012, pp. 200–205.
[62] A. Souzis, Building a semantic Wiki, IEEE Intell. Syst. 20 (5) (2005) 87–91.
[63] K. Sycara, In-Context information management through adaptive collaboration of intelligent agents, Intelligent Information Agent: Agent-Based Information Discovery and Management on the Internet, Springer, Matthias Klusch, 1999, pp. 53–77.
[64] M.M. Stark, R.F. Riesenfeld, WordNet: an electronic lexical database, Proceedings of Eleventh Eurographics Workshop on Rendering, 1998.
[65] M. Shujahat, M.J. Sousa, S. Hussain, F. Nawaz, M. Wang, M. Umer, Translating the impact of knowledge management processes into knowledge-based innovation: the neglected and mediating role of knowledge-worker productivity, J. Bus. Res. 94 (2019) 442–450.
[66] M. Uschold, Building ontologies: towards an unified methodology, Proceedings of the Sixteenth conference of the British Computer Society Specialist Group and Expert Systems, Cambridge UK, 1996.
[67] A. Vizcaino, J.P. Soto, J. Portillo, M. Piattini, et al., A multi-agent model to develop knowledge management systems, Proceedings of the Fortieth Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007, 2007, p. 203b.
[68] Y. Wang, W. Liu, D. Bell, et al., Combining uncertain outputs from multiple ontology matchers, Proceedings of the First International Conference on Scalable Uncertainty Management, Berlin, Heidelberg, 2007, pp. 201–214.
[69] Nan Wang, Yongqiang Sun, Xiao-Liang Shen, Xi Zhang, A value-justice model of knowledge integration in wikis: the moderating role of knowledge equivocality, Int. J. Inf. Manag. 43 (2018) 64–75.
[70] M. Weggeman, Knowledge management: the modus operandi for a learning organization, in: J.F. Schreinemakers (Ed.), Knowledge Management: Organization, Competence and Methodology, Proceedings of ISMICK'96, Rotterdam, the Netherlands, Wurzburg:Ergon Verlag, Advances in Knowledge Management, 1 1996, pp. 175–187.
[71] M. Wooldridge, Reasoning About Rational Agents, MIT Press, Cambridge, MA, 2000.
[72] Y. Zhang, G.Q. Huang, T. Qu, O. Ho, et al., Agent-based workflow management for RFID-enabled real-time reconfigurable manufacturing, Int. J. Comput. Integr. Manuf. 23 (2) (2010) 101–112.
[73] X. Zhang, W. Liu, F. Jin, Intelligent agent for knowledge managemnt in E-commerce, Proceedings of the 2006 First International Symposium on Pervasive Computing and Applications, 2006, pp. 455–466.
[74] M. Zapp, M. Hoffmeister, A. Verl, Methodology to apply semantic wikis as lean knowledge management systems on the shop floor, Proc. CIRP 12 (2013) 444–449.