Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2014 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

2014

A Semantic-Based Knowledge Management Platform

Yi Liu

Deli Yang

Yu Wang

A Semantic-Based Knowledge Management Platform

Abstract

We describe the development of a semantic-based knowledge management platform for web-enabled environments featuring intelligence and insight capabilities. The main objective of the platform is to semantically search, analyze and present information retrieved from the web (or any other type of document) as well as allows domain ontology to evolve periodically. This is achieved through the use of Multi-Agent Systems and ontologies, one for building distributed systems and the other for knowledge representation. The most important feature of the SKMP lie in that the information retrieved from the web is the source of ontology evolving, while the periodical ontology evolving that will enrich domain ontologies by adding more semantics in return significantly improves efficiency of the semantic retrieval, i.e., the two are mutually reinforcing relationship. We test and verify the feature through three domain ontology from different domains.

Keywords: *knowledge management, domain ontology, semantic analysis, ontology evolving, multi-agent system.*

1 INTRODUCTION

In the era of knowledge economy, knowledge is a major determinant of competiveness and knowledge sharing is beneficial to improve the company performance and industrial practices[1]. Knowledge Management (KM) is a concept of systematically managing knowledge assets of an organization to improve performance, gain competitive advantage, transfer lessons learnt and develop collaborative practices[2].

Engineers typically assess the evolution of their disciplines by reading journals, attending conferences or, quite often, by hearsay. Instead, the web (as well as other information resources) offers scattered and distributed information that is impossible to analyze manually. Today, the Web has a large amount of information and becomes an important expedient for communication between people and enterprises[3]. In addition, organizations have intranets amounting to several million pages. The large majority of these documents are weakly structured. These repositories are usually searched by means of the keywords specified by the user. Unfortunately, these keywords are generally too common and abundant[4]. Documents downloaded from the web are indexed according to their contents, and only those matching the query (according to some metric) are returned to the user. The results of this type of search usually suffer from two problems derived from the nature of the query and the lack of structure in the documents: some of the retrieved documents are irrelevant, and some of the relevant documents may not have been retrieved (low precision and recall ratios).

The performance of a search engine can be improved by the use of an ontology. In its conventional form an ontology accounts for the representation of shared concepts in a domain by specifying a hierarchy of terms facilitating communication among people (collaboration) and applications systems (integration of tools), i.e., facilitating knowledge sharing and reuse.

Building and evolving an ontology are difficult and complex tasks[5]: they involve numerous entities (terms, concepts, relations), the environment of the ontology is dynamic (addition of new documents, ontologist's actions) and we cannot predict all ontology evolution possibilities. That is why a unique entity to solve these problems cannot list all the possible situations to which it can be confronted as well as the actions it has to take in such situations. The multi-agent system distribute the problems on autonomous entities that have a local perception of each situation that can arise during the system functioning and that have simple, generic and local behaviors in order to self-adapt to these situations.

In this paper, we propose an Semantic-based Knowledge Management Platform named SKMP that is based on Multi-Agent System (MAS) that can automatically perform the semantic retrieving and analyzing of pages form the web and periodically evolving of the domain ontology. In SKMP, the agents can collect information from the web more effectively through query ontology (i.e. semantic query) by taking advantage of semantic annotations in a document that are machine processing and add structure and/or semantics to the document (meta-information). The contents of these web-pages then are semantically analyzed, which is capable of understanding words synonymy and ambiguity in a text, detecting and resolving sense conflicts between different parts of text, acquiring additional implicit information and working with inter-phrasal context. In the end, the retrieved pages are presented to the user in decreasing order according to their relevance to the user query on semantics, thus ensuring higher precision and recall ratios. Also, the semantics contained in the content of the retrieved web-pages is the source of the ontology evolving. The periodically upgraded knowledge is the basis of the knowledge reasoning, the data mining, decision making, etc.

In the next section we present the proposed architecture for a knowledge management platform. We then explain in detail the procedure of the main modules. Finally, we discuss the conclusions and the further work.

2 The Semantic-based Knowledge Management Architecture

We proposed a distributed architecture to perform the semantic retrieval and analysis of information from the web and the evolving of domain ontologies. As shown in Fig. 1, the system has six main parts: the *user interface*, the *search module*, the *semantic text analysis module*, the *ontology evolving module*, the *ontology merging module* and the *knowledge management module*.

In the next sections we describe in more detail this architecture and its main features.

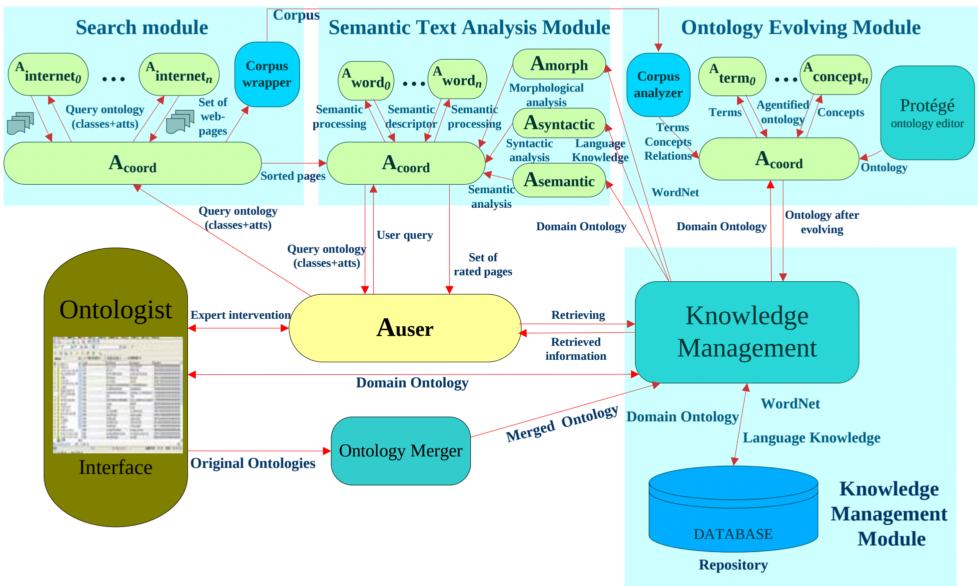

Figure 1. *The Semantic-based Knowledge Management System*

2.1 The User Interface

The user interface consists of two parts: a *Graphical Interface* and a *User Agent*. The first one allows the user to specify the actions to be performed by the system, and to visualize the search results. Additionally, the user may retrieve ontology from a repository.

The second part, the *User Agent* (A_{user}), represents a user of the system. It stores all personal data, such as user preferences or profiles.

2.2 The search module

This module has an agent-based system whose main goal is to retrieve a collection of relevant web pages by means of semantic annotations in a document.

In a Multi-Agent System, each agent is an autonomous entity with its own beliefs and goals. In the proposed module we define two types of agents:

- *Coordinator Agent* (A_{coord}). The search process has to be coordinated around some information agents. The coordinator has two tasks: sending the semantic query (i.e. query ontology) to *Internet Agents* and sorting the retrieved web-pages that are the input of the semantic text analysis module.
- *Internet Agents* ($A_{internet}$). These agents encapsulate the analysis of pages from the web with a given criterion.

The *Corpus Wrapper* is in charge of packaging concepts, terms (annotations in a document) and their relations in a corpus as the input of the ontology evolving module.

This module is explained in more detail in §3.

2.3 The semantic text analysis module

As an output of the previous module, we have a set of web pages for rearranging them to *User Agent*. The main objective of the semantic analysis module is to completing the semantic analysis of sorted pages and transmitting them to user in decreasing order according to their relevance to the user query.

This module is composed of three types of agents:

- *Coordinator Agent* (Acoord). The coordinator has two tasks: assigning each word to *Word Agent* and rearranging the retrieved web pages.
- *Word Agents* (Aword) . These agents perform the semantic analysis of text with a given criterion.
- *Morphological analysis Agent* (Amorph), *Syntactic analysis Agent* (Asyntactic) and *Semantic analysis Agent* (Asemantic). The three agents provide the support of three stages of semantic analysis respectively.

This module is explained in more detail in §4.

2.4 The ontology evolving module

This module focuses on ontology evolving from texts.

First, the *Corpus Analyzer* identifies relevant candidate terms as well as relevant lexical relations in corpus made by *Corpus Wrapper*. The *Coordinator Agent* (A_{coord}) then assign terms and domain concepts to the *Term Agents* (A_{term}) and the *Concept Agents* ($A_{concept}$) respectively. After ontology evolving, the agentified ontology is transformed into domain ontology by using ontology editor Protégé. Finally, the *Coordinator Agent* (A_{coord}) transmit to the *Knowledge Management Module* the domain ontology and the set of retrieved pages in which every page link to at least one concept in ontology that will allows user to retrieve information directly from the local repository as well as make a significant contribution to knowledge reasoning, data mining and decision making.

This module is explained in more detail in §5.

2.5 The ontology merging module

We present the ontology merging module for the purpose of further improving the efficiency of ontology evolving by way of emerging the domain ontology in repository with other ontologies from the same domain. Ontology merging in the same domain is the most effective method in ontology integration.

For this reason, a multiple-mapping-based ontology merging system is proposed, which can fully improve the efficiency of merging with the advantages of the combination of three relative mature mapping: Grammar mapping, Structure-based mapping and Machine Learning-based mapping.

This module is explained in more detail in §6.

2.6 The knowledge management module

Knowledge should be presented to the user in the form most suitable to its comprehension. The main goal of the knowledge management module is to store and retrieve knowledge, improve collaboration, locate knowledge source, and further provide the support of the knowledge reasoning, data mining and decision making.

The *Knowledge Management* is implemented to provide user with the retrieval of domain ontologies, transmit knowledge between the modules described above and the repository such as WordNet, Language Knowledge and domain ontologies, ensure the security and integrity of data, and so on.

3 THE SEARCH PROCESS

As shown in Fig. 1, the search module is composed by a *Coordinator Agent* and a set of *Internet Agents*. The search is monitored by the *Coordinator Agent* (A_{coord}) that receives requests from the *User Agent* (which sends a *query ontology* to A_{coord}).

The *domain ontology* received by the coordinator is a hierarchy of classes and properties that is split in several parts; each part, that can contain one or more classes, is sent to an *Internet Agent*. The *Internet Agents* perform the search and retrieve the more interesting pages related to a concept of the *query ontology* by taking advantage of semantic annotations in a document within a deadline specified by the user. When the deadline is reached, all *Internet Agents* will return the results to the coordinator. Then, all these results will be removed repeated and similar elements and preprocessed, which will be the core input to the semantic analysis module.

4 THE SEMANTIC ANALYSIS PROCESS

The essence of the semantic analysis process is that each word of the text is assigned an autonomous agent capable of negotiating with other similar agents about the meaning of each word in the sentence and its general meaning on the basis of domain ontology.

During negotiations the word agents can speculate about the possible word meanings and their semantic relations, find and resolve meaning conflicts, detect implicit information on the basis of domain knowledge, take into account the context of the word usage within one sentence and inter-phrasal context thus connecting the words of various sentences into one semantic network composed of descriptors that contain formal monosemantic description of the initial text meaning and simple to compare similarity of information they contain on the base of the ontology.

The algorithm of comparing semantic descriptors was developed for semantic analysis. This algorithm is based on finding in one of the descriptors the sub-network which is close to the network of the other descriptor as much as possible. Similarity degree of two sub-networks is defined as similarity degree of their respective pairs of nodes. Similarity degree of two nodes depends on relative position of corresponding concepts in the ontology and on the values of attributes connected with nodes under comparison.

The *ranking* process is one of the major steps of ranking the retrieved web-pages, which ensures the sequence list of the pages sent to the user in strict accordance with their relevance to the user query on semantics. The rate of a web page **p**, shown in Eq.1, is a function, **rate**(**p**), that depends on comparing algorithm of semantic descriptors (**D**) from semantic network, domain information (**O**) from the ontology, parametric information (**U**) from the user personal data (preferences, profile), and global information (**PR**) from previous results of retrieved pages.

$$\mathrm{rate}(p) = f(D, O, U, PR) \qquad (1$$

5 THE ONTOLOGY EVOLVING PROCESS

The ontology evolving module is composed of a *Corpus Analyzer*, a *MAS* and a *Coordinator Agent*. The input of this module is a corpus of documents, and the output is an ontology after evolving. The ontologist interface allows the ontologist to interact with the MAS.

The goal of the *Corpus Analyzer* is to identify relevant candidate terms as well as relevant lexical relations. It includes a *terms extractor* and a *lexical relations generator*. In this module, we are interested in four types of lexical relations: (*I*) *Hypernym* expresses a generic-specific relation between terms. This may lead to define a class-subclass (*is_a*) relations between the concepts denoted by these terms; (*II*) *Meronymy* may lead to define a *part of* relation between concepts; (*III*) *Synonymy* relates semantically close terms that may denote the same concept; (*IV*) Other relations are any other kinds of lexical relations that will lead to a specific set of semantic relations, such as *causes, leads to, etc.*. The *Corpus Analyzer* generates triples $< T_i$, Rel, $T_j >$ where T_i and T_j are candidate terms or terms (if the term is present in the ontology) and Rel is a lexical relation. Each triples has a confidence (Q, I) where Q is the quality of the relation (value between 1 and 10) and I is the number of instances of the relations in the corpus.

The **multi-agent system** has, as input, the triples returned by the *Corpus Analyzer*. The MAS consists of (*i*) *term* agents that represent the terminological component of the ontology and (*ii*) *concept* agents that represent the conceptual part of the ontology. The creation of agents and their communications are managed by the *Coordinator Agent*.

Term Agent Behaviors

A *term agent* has a status (*term* or *candidate term*) indicating if it is in the ontology or not yet. A *term* agent is connected by a lexical relation to other *term agents*. It must also be connected to one *concept agent*. Each relation between *term agents* is tagged by the confidence of the triples $< T_i$, Rel, $T_j >$. A *term agent* has three objectives:

(1) In order to **denote a *concept* agent**, a *term agent* asks for the creation of a *concept agent* to the *coordinator*. This creation is done if, in the current MAS, a *concept agent* having

the same label does not exist. The *coordinator* transmits thereafter the identifier of this new *concept agent* to the *term agent*. Then, the *term agent* sends to the *concept agent* a request for establishing a denotation relation. This request is always accepted by the *concept agent*. The confidence of the denotation link is equal to the greatest value of the lexical relations of the *term agent*.

(2) A *term agent* **processes its outgoing lexical relations**. A lexical relation has a confidence and a status (*not treated, treated or refused*). A *term agent* processes its relations from the most relevant to the less relevant. To do this, a *term agent* sends a request to its *concept agent* in order to transform the lexical relation. A *concept agent* processes the request, and then notifies the *term agent* by a message of acceptance or refusal. The *term agent* updates the status of the processed relation. If the relation is refused, the *concept agent* sends a "refuse" message and the status of the lexical relation will be refused. A *term agent* can request again, later, to process the refused relation if its confidence increases. When a *term agent* asks for the management of a synonymy relation, its *concept agent* sends a denotation request to the target *term agent* of this relation. If the confidence of the request is greater than the current denotation link of the target *term agent*, this latter accepts the request, changes its denotation link and notifies the *concept agent* by message of acceptation. The target *term agent* refuses the request in the contrary case. The initial *term agent* is thereafter notified.

Concept Agent Behaviors

A *concept* agent has a status (*concept* or *candidate concept*) indicating if the agent is in the ontology or not yet. A *concept* agent is connected by conceptual relations to other *concept* agents and connected by denotation links to other *term* agents. Every relation can have the status (*not treated, treated or refused*). A *concept* agent has three objectives:

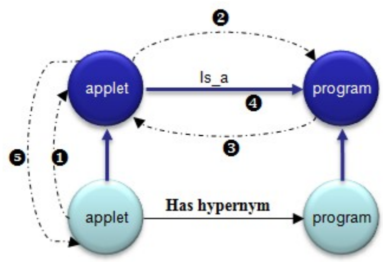

Fig. 2 *Interaction between term agent and concept agent to establish an is_a relation*

(1) A *concept* agent receives requests coming from *term* agents for **processing lexical relations** ❶ (in Fig. 2). In order to process a request coming from a *term* agent, the *concept* agent gets the *concept* agent denoted by the *term* agent that is target of this lexical relation. Then it sends a request for establishing a conceptual relation with this *concept* agent ❷. When a *concept* agent receives a request to create a conceptual relation, it can accept or refuse the relation by sending a notification ❸ (it will refuse if it has a stronger conceptual relation). When a *concept* agent receives a notification, it updates the status of the concerned relation and its links with the other *concept* agents ❹. A *concept* agent, can propose later a "refused" conceptual relation if its confidence evolves. The initial term agent of the lexical relation is notified thereafter by the *concept* agent that established the conceptual relation ❺.

(2) A *concept* agent must **have a preferred label**. This label is the label of the *term*

agent that is connected to it and has the greatest confidence value. This label can evolve if new *term* agents denote the *concept* agent or if the confidence of the denotation links has evolved.

The processing of synonymy relations involves the move of a *term* agent that denotes a *concept* agent towards another *concept* agent. If a *concept* agent is not connected to any agent, it cannot receive any request and then cannot reach any of its objectives: it considers itself as useless in the MAS and disappears.

6 THE ONTOLOGY MERGING PROCESS

The multi-mapping-based ontology merging framework is as bellow (Fig. 3).

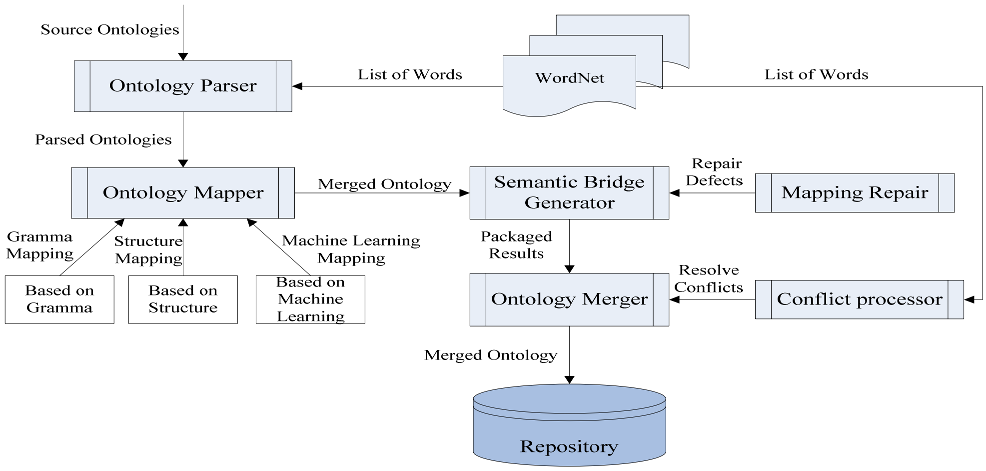

Figure 3. *Ontology Merging Framework*

Ontology Parser

The ontology parser of this framework is used for analyzing and identifying the source ontology. In this framework, the common tool of Jena is utilized.

Ontology Mapper

The ontology mapper is used to discover the semantic association among different ontologies. In the proposed framework, several relative mature mapping methods are utilized, which of them have different emphasis. Hence, for sake of the mapping results, the comprehensive evaluation method is employed in the analysis and comparison, and then the final results are sequentially gotten.

Semantic Bridge Generator

For the purpose of the merging usage preparation, the semantic bridge generator is used to package all the information of ontology mapping and mapping results. The generated results after mapping according to the synonyms and similarity relations provided by WordNet are class mapping, property mapping and relationship mapping.

Ontology Merger

Based on the mapping results, the ontologies can be merged through making related merging rules, as shown in Fig. 4.

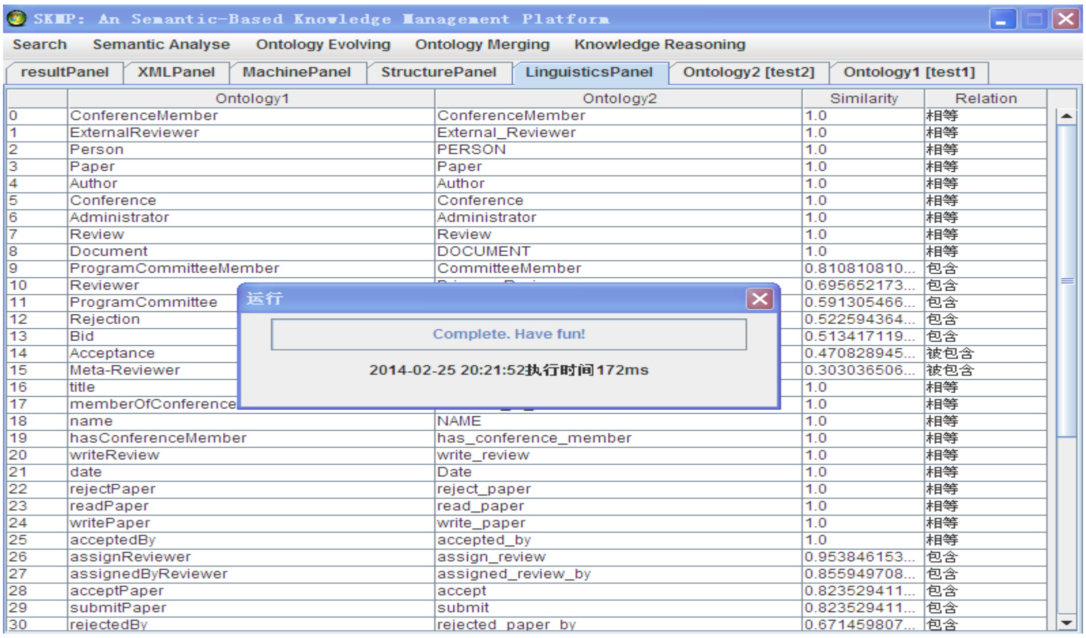

Figure 4. *The Ontology Merging Result*

We tested our SKMP on 3 ontologies from different domain. The results obtained are quite encouraging that the more the times of ontology evolving is, the higher the Precision and Recall ratio of retrieval are.

7 CONCLUSIONS AND FUTURE WORK

This paper describes a distributed knowledge management system. The main objective of the platform is to improve the results of retrieval, generate and periodically upgrade knowledge that is the basis of the knowledge reasoning, data mining and decision making. The retrieval and semantic analysis of information from the web and the evolving of ontology are achieved through the use of multi-agent systems and ontologies.

In the next phase of the project, we are planning to study on the knowledge reasoning. A first version of the ontology reasoning system has been implemented. Also, we are currently working on the local behaviors of agents in order to improve the efficiency of semantic analysis and ontology evolving.

References

[1] X.H. Wang, T. N. Wong and Z.P. Fan (2013). Ontology-based supply chain decision support for steel manufacturers in China. Expert syst. appl. 40 (18), 7519-7533.
[2] P. Moonseo, K.W. Lee, Kyung-won and H.S. Lee (2013). Ontology-based Construction Knowledge Retrieval System. KSCE Journal of Civil Engineering, 17(7), 1654-1663.
[3] C. Ikkyu, R. Seungmin and K. Minkoo(2013). Semi-automatic construction of domain ontology for agent reason. Personal and Ubiquitous Computing, 17(8), 1721-1729.
[4] K.Marat, M. Igor and R. George (2007). Multi-agent meta-search engine based on domain ontology. in: 2nd International Workshop on Autonomous Intelligent System - Agents and Data Mining, vol. 4476, p. 269-274.
[5] S. Zied and C. Valerie (2012). Evaluation of a Multi-Agent System for the Evolving of Domain Ontologies from Texts. in: 10th International Conference on Practical Applications of Agents and Multi-Agent Systems, vol. 155, p. 169-179.