Journal of Web Engineering, Vol. 6, No. 4 (2007) 309-336
© Rinton Press

A MODEL-DRIVEN APPROACH OF ONTOLOGICAL COMPONENTS FOR ON- LINE SEMANTIC WEB INFORMATION RETRIEVAL

HAJER BAAZAOUI ZGHAL[1], MARIE-AUDE AUFAURE[2, 3], NESRINE BEN MUSTAPHA[1]

[1]Riadi-GDL Laboratory, ENSI Campus Universitaire de la Manouba, Tunis, Tunisia
{hajer.baazaouizghal, nesrine.benmustapha}@riadi.rnu.tn

[2]Supelec, Computer Science Department, Plateau du Moulon, 91 192 Gif sur Yvette, France
marie-aude.aufaure@Supelec.fr

[3]Inria Paris-Rocquencourt, Axis project, Domaine de Voluceau, 78 153 Le Chesnay Cedex, France
marie-aude.aufaure@inria.fr

Received June 1, 2007
Revised December 1, 2007

With the development of Web engineering and the advent of the Semantic Web, adding a semantic dimension to the Web by the deployment of ontologies has contributed to solve many problems. Ontologies specify the knowledge shared by a community of a target domain in an explicit and formal way. The present work deals with the problem of ontology construction for the Semantic Web. The objective of the paper is to propose a solution for the representation of Web knowledge by means of ontologies as well as to define an approach to their semi-automatic construction. The implementation and the experimentation of the proposition were achieved by the development of a framework, called "OntoCoSemWeb" and an On-Line Information Retrieval tool using the generated ontologies. An evaluation and an analysis of the results of this approach were carried out.

Keywords: Semantic Web, Ontology building, Meta-ontology, ontology construction, ontology learning, on-line information retrieval

1 Introduction

Web engineering is a relatively new branch of software engineering, which addresses specific issues related to the design and development of large-scale Web applications. In particular, it focuses on the methodologies, techniques and tools that are the foundation of complex Web application development and that support their design, development, evolution, and evaluation.

The volume of available data on the Web is growing exponentially. Consequently, integration of heterogeneous data sources and information retrieval become more and more complex. Adding a semantic dimension to Web pages is a response to this problem and is known as the Semantic Web [1][2][3]. Ontologies can be seen as a fundamental part of the Semantic Web. They can be defined as an explicit, formal specification of a shared conceptualization [4]. Ontologies can be classified as [5]: lightweight ontologies gathering concepts and relations hierarchies which can be enriched by classical properties, called Axiom Schemata (algebraic properties and signatures of relations, abstraction of concepts, etc.) and heavyweight ontologies which add properties to the semantics of the conceptual primitives and are only expressible in Axiom Domain form. The axioms schemata describe the classical properties of concepts and relations (subsumption, disjunction of concepts, algebraic properties and cardinalities of the relations, etc.).

The domain axioms characterize domain properties expressible only in an axiom form. They specify the formal semantics constraining the conceptual primitive interpretation [6].

Indeed, their use facilitates Web information retrieval, domain knowledge sharing as well as knowledge integration. Building an ontology manually, on the other hand, is a long and tedious task. Many approaches have been proposed this last decade to ease this task.

Numerous approaches have been defined in order to develop ontologies [7]. Some of them describe how to build an ontology from scratch or to reuse other ontologies (Cyc approach, Uschold and King's proposal [8], METHONTOLOGY [9], On-To-Knowledge [10], etc.). Other approaches describe how to build an ontology by means of the transformation of other ontologies, for example, using reengineering [4]. Ontological reengineering [11] is the process of retrieving and mapping a conceptual model of an implemented ontology to another, more suitable, conceptual model which is re-implemented.

Starting from the fact that ontology is a shared and common understanding of a domain, emphasis is put on consensus about the content of ontologies, in the sense that a group of people agrees on the formal specification of the concepts, relations, attributes and axioms that the ontology provides. However, the problems of how to jointly construct ontology (with a group a people) and how to commonly construct ontology (with people at different locations) are still unsolved. [12] identified the following problems concerning collaborative construction of ontologies: management of the interaction and the communication between people; data access control; recognition of a moral right concerning knowledge (attribution); detection and management of errors; and concurrent management and modification of the data. There are a few detailed proposals about how to build ontologies in a collaborative way. Nevertheless, for several years now, the main proposals have been: (1) CO4, for collaborative construction of Knowledge Bases at INRIA; and (2) the approach used in ontologies building at the Knowledge Annotation Initiative of the Knowledge Acquisition Community, also know as $(KA)^2$ initiative.

Acquiring domain knowledge for building ontologies requires a lot of time and many resources. In this sense, we can define ontology learning as the set of methods and techniques used for building ontology, and enriching, or adapting existing ontology in a semi-automatic way using several sources. Other terms are also used to refer to the semi-automatic construction of ontologies, e.g., ontology generation, ontology mining, ontology extraction, etc. Several approaches exist for the partial automation of the knowledge acquisition process. Ontology learning approaches are classified according to the type of input data: texts, dictionaries, knowledge bases, semi-structured schemata and relational schemata [13]. Web engineering focuses on methodologies, techniques and tools that are the foundation of Web application development and which support their design, development, evolution, and evaluation but lack the management and generation of metadata.

Ontology development and maintenance is a prerequisite for the Semantic Web. Many methods and methodologies have been defined but without any consensus. Our aim is to capitalize how the techniques are used for ontology construction in a meta-model called a meta-ontology. This model allows us to keep a trace about the way an ontology was built. We also think that ontology about domain application document structure and available services are strongly correlated. This is the reason why we define three ontologies: a domain ontology, a structure ontology and a service ontology. All these components are linked to the meta-ontology. In order to validate our conceptual approach, we have designed an on-line information retrieval system which uses the domain and the service ontologies.

This paper is organised as follows. Section 2 presents related works in the field of ontology building and learning methodologies to develop semantically annotated Web pages. Section 3 describes the proposed model-driven architecture to represent domain knowledge on the Web by means of ontologies.

In Section 4 our approach for semi-automatic building of domain ontology from Web content is described. The development of the framework supporting this approach is presented in Section 5. A case study is then detailed in Section 6. Finally, we conclude and give some perspectives for this research work.

2 Related Work

Web Engineering focuses on the systematic and cost efficient development and evolution of Web applications. The proposed ontology building methodologies in a context of Web engineering consist in ontology extraction from texts by applying linguistic techniques, statistic techniques, conceptual clustering and association rules. Figure 1 shows a classification of methodologies for ontology building.

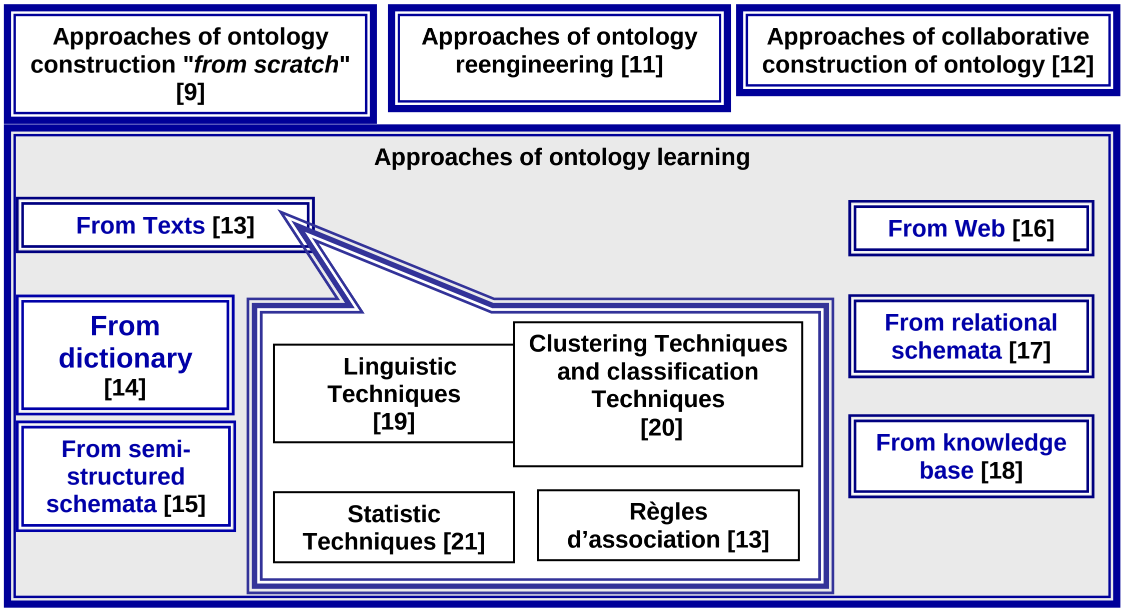

Approaches of ontology construction "*from scratch*" [9]
Approaches of ontology reengineering [11]
Approaches of collaborative construction of ontology [12]
Approaches of ontology learning
From Texts [13]
From Web [16]
From dictionary [14]
From semi-structured schemata [15]
Linguistic Techniques [19]
Clustering Techniques and classification Techniques [20]
Statistic Techniques [21]
Règles d'association [13]
From relational schemata [17]
From knowledge base [18]

Fig. 1. Classification of ontological engineering approaches.

Methodologies for ontology building can be classified according to the use or non-use of a priori knowledge (such as thesaurus, existing ontologies, etc.) and according to learning methods. In the following section, we focus on ontology learning from texts. Sub-section 2.2 deals with ontology learning from Web pages. Sub-section 2.3 analyses ontology building tools. A synthesis is outlined in Sub-section 2.4.

2.1 Learning Ontologies from Texts

The proposed approaches can be classified according to the technique used, namely linguistic, lexico-syntactic patterns extraction, clustering or classification, and hybrid ones. The input data is constituted by linguistic resources such as a list of terms and relationships. The huge volume of these data, their quality and relevance has to be taken into account by filtering methods. These methods can be guided by an expert (knowledge acquisition from texts) or can be automatic (text mining, learning).

Linguistic-based techniques include lexical, syntactic and semantic texts analysis. The objective is to extract a conceptual model of a domain. We find two main methodologies defined by [19, 22]. The methodology defined by [19] intends to extract knowledge from technical documents. Two hypotheses are made by the authors: the first one supposes that the ontology designer has a good knowledge of the application domain and can determine the relevant terms, while the second one assumes that the ontology designer also has a precise idea of ontology usage. This methodology analyses a corpus with tools appropriate for natural language automatic processing and linguistic techniques, and extracts terms and relationships. The normalization step concerns mapping between natural language and a formal language. The semantic interpretation of the texts relies on usage and expertise. Semantic relationships are obtained from lexical and syntactic relationships, from semantic patterns like causal and consequence relations and from rules given by domain experts. The concepts' hierarchy is built using the semantic relationships. The formalization step automatically translates the ontology into a given language, e.g., RDF, OWL, etc.

In these linguistic approaches, lexico-syntactic patterns are manually defined by linguists. Some research work has been proposed to extract lexico-syntactic patterns automatically. [23] starts from an existing ontology and extracts a set of pairs of concepts linked by relationships, in order to learn hyponymy relationships and produce lexico-syntactic patterns. The latter are used to discover other relationships between the concepts of the existing ontology and are based on the learned patterns. This approach is used to extend an existing lexical ontology. The KAT system (Knowledge acquisition from Texts) [24] includes four steps: learning new concepts, classification, learning relationships and ontology evaluation. Concept classification consists in analyzing words that appear in the expression associated with a candidate concept: [word, C], where "word" can be a noun or an adjective and C is a known concept. This classification states that the concept [word, C] subsumes C. This is equivalent to adding a hyponymy relationship between [word, C] and C (i.e., C is the concept "hotel" and the word is "luxurius").

These techniques, based on lexico-syntactic patterns learning, lead to good results when it comes to learning hyponymy relationships. At the same time, however, we find problems such as terms polysemy or errors that depend on the corpus. The use of classification techniques such as hierarchical or conceptual clustering is a way to solve these problems. The methodology proposed by [1] consists in classifying documents into collections related to words sense, using a labelled corpus and Wordnet [25]. For each collection, the relative frequencies are then extracted and compared to the other collections. Topic signatures are computed and compared to discover shared words. This methodology is used to enrich concepts of existing ontologies by analyzing Web texts. Other methods [26, 20] combine linguistic techniques and clustering to build or extend an ontology.

Some research work has been done to study the distribution of words in texts to improve concepts clustering using new similarity measures. DOODLE II, an extension of DOODLE, is an environment for the rapid development of domain ontologies. It is based on the analysis of lexical co-occurrences and the construction of a multidimensional space of words [27]. This approach extracts taxonomic relationships using Wordnet and learns non taxonomic relationships by searching association rules and extracting pairs of similar concepts using the words multidimensional space. In order to detect relations between terms, those appearing in a window with a size of four words around important terms are extracted. The, terms that frequently co-occur are candidates to be linked in the ontology. Verbs that are in the context of such terms are proposed to label the relation.

2.2 Web-based Ontology Learning

Our main objective is to define an approach to build ontologies for the Semantic Web. This kind of ontology, closely linked to the Web usage, has to integrate the dynamic aspects of the Web. In this section, we present some approaches defined specifically for the Web. Propositions have been made to enrich an existing ontology using Web documents [1, 28]. However, these approaches are not specifically intended for Web knowledge extraction. The approach proposed by [29] attempts to reduce the terminological and conceptual confusion between members of a virtual community. Concepts and relationships are learned from a set of Web sites using the Onto-learn tool. The main steps are: the terminology extraction from Web sites and a Web documents data warehouse, the semantic interpretation of terms and the identification of taxonomic relationships. Some approaches transform html pages into hierarchical semantic structures encoded in XML, taking into account html regularities [30].

Finally, we can also point out some approaches dedicated only to ontology construction from Web pages without using any a priori knowledge. The approach described in [16] is based on the following steps: (1) extract some keywords representative of the domain, (2) find a collection of Web sites related to the previous keywords (using for example Google), (3) analyze exhaustively each Web site, (4) search the initial keywords in a Web site and find the preceding and following words; these words are candidates to become a concept, (5) performed a statistical analysis of each selected concept based on the number of occurrences of this word in the Web sites and at last, (6) define a new keyword for each concept extracted using a window around the initial keyword. The algorithm iterates recursively.

In [31], a method is proposed to extract domain ontology from Web sites without using a priori knowledge. This approach takes advantage of the Web pages structure and defines a contextual hierarchy. The data pre-processing is an important step to define the more relevant terms to classify. Weights are associated to the terms according to their position in this conceptual hierarchy. These terms are then automatically classified and concepts are extracted.

2.3 Ontology Building Tools

In this sub-section, we will provide a broad overview of some of the available tools and environments that can be used to build ontologies, either from scratch or reusing other existing ontologies. We distinguish two categories: tools for conceptualization and tools for ontologization. Apart from the technical characteristics (software architecture, versions; etc.), the criteria which are used for the evaluation of an engineering ontology environment, concern reasoning capabilities, interoperability with other tools, and the use of graphical interfaces. Table 1 shows a synthesis of the most used tools, focusing on the above mentioned evaluation criteria. The tools use mostly the frame model enriched by formulas in first order logic (e.g., OntoEdit [32], ONTOLINGUA [33], Protégé2000 [34], WebOnto [35], WebODE [36]), or by description logic (e.g., OilEd [37], TERMINAE [38]).

Table 1. Synthesis of main environments of ontology building.

	OILED	ONTOEDIT	TEXT-TO-ONTO	ONTOLINGUA	PROTEGE	WEBODE	WEBONTO	DOE
Knowledge model	DL (DAML + OIL)	Frames +FOIL	RDF + Rdfs	Frames + FOIL	Frames + FOIL	Frames + FOIL	Frames + FOIL	Entity / Association
Axiom language	Yes	Yes (FLogic)	No	Yes (KIF)	No (PAL)	Yes (WAP)	Yes (OCML)	No
Methodological support	No	Yes	Yes	No	No	Yes	No	Yes
Inferring engine	FACT	OntoBroker	No	ATP	PAL	Prolog	Yes	No
Coherence Test	Yes	Yes	Yes	No	Yes	Yes	Yes	No
Graphical hierarchy	No	No	Yes	Yes	Yes	Yes	Yes	Yes

On one hand, only KAON and DOE offer ontology edition using the Entity/Association model. However, the latter doesn't allow the specification of the semantics related to the domain and doesn't offer any reasoning or evaluation functionalities. On the other hand, many tools enable the user to specify many types of axioms but do not allow the editing of axioms on the conceptual level and do not provide operational constraints. This is due to the fact that the user specifies axioms in order to validate the created ontology. The constraints related to testing knowledge bases are then specified by the PAL language used in "Protégé". These constraints cannot be applied to deduce new knowledge. Moreover, the inference engine in these tools is used to test the ontology coherence and doesn't permit other type of reasoning tasks applied to knowledge bases. Besides, the existing tools don't offer the functionalities which would allow the construction of heavyweight ontologies.

2.4 Synthesis

A study of ontology building methodologies has been done according to the use or non-use of a-priori knowledge (such as thesaurus, existing ontologies, etc.) and according to learning methods. Ontology learning from texts and from Web has been detailed. Different criteria like: *learning sources*, *type of ontology to be built*, *techniques used to extract concepts*, *relationships and axioms*, and *existing tools*; allows distinction between studied approaches and techniques. The most recent methodologies generally use a priori knowledge such as thesaurus, minimal ontology, other existing ontologies, etc. Each one proposes techniques to extract concepts and relationships, but does not propose axioms. Axioms can represent constraints but also inferential domain knowledge. As is the case with instance extraction, we can also find techniques based on first order logic [39], on Bayesian learning [40], etc. It is important to capitalize the obtained results by the different methods and to characterize existing techniques, their properties and to determine how we can combine them. From the state of art we can resume requirements of learning ontology process by: (1) *Knowledge sources preparation (textual corpus, collection of Web documents),* eventually using a priori knowledge (ontology with a high-level abstraction, taxonomy, thesaurus, etc.), (2) *Knowledge sources preprocessing*, (3) *Concepts and relationships learning* and finally, (3) *Ontology evaluation and validation* (generally done by experts).

The ontology is built according to the following dimensions: *Input type* (data sources, possible a priori knowledge, etc.), *tasks involved when preprocessing* (simple text linguistic analysis, document classification, text tagging using lexico-syntactic patterns, disambiguating, etc.), *learned elements* (concepts, relationships, axioms, instances, thematic roles), *learning methods characteristics* (supervised or not, classification, clustering, rules, linguistic, hybrid), *automation level* (manual, semi-automatic), automatic, cooperative), *characteristics of the ontology to build (*structure, representation language, coverage) and *the usage of the ontology and users' needs* [19]. In the next section ontological building approaches towards the Semantic Web is detailed.

3 Ontological Building Approaches Towards the Semantic Web

Generally, a Web document lays out contents, services and a structure. Thus, to specify knowledge related to a specific domain in the Web, we distinguish three types of semantics: *the semantic related to a domain*, *the semantic of services* of the domain and *the semantic of the structure* of the Web sites. Analyzing the Web content is a difficult task relative to redundancies and incoherencies of Web structures and information.

The proposed approach is considered as semi-automatic one. Functionalities related to the initialization phase and the step of the Metaontology alimentation of the incremental phase of training are the principal automated steps of the proposed approach. In fact, metaontologies existed implicitly in the proposal of new representation languages. They are used during the conceptualization or the validation and the evaluation of ontologies [41]. The contribution of this work appears principally in the use of a Metaontology for the construction of ontologies.

Proposing an approach to build automatically ontologies is crucial. Our approach is based on the cyclic relation between Web mining, Semantic Web and ontology building as stated in [42] and is based on the following statements: (1) *satisfy the fact that the ontology is useful to specify and extract knowledge from the Web*, (2) *link the semantic content within the Web documents structure,* and (3) *combine linguistic and learning techniques taking into account the scalability and the evolution of the ontology*. Our knowledge base is produced using Web mining techniques.

We mainly focus on Web content and Web structure mining. Building such ontology imposes the resolution of two main problems. The first one is relative to the heterogeneity of Web documents structure while the second one concerns technical choices to extract concepts, relationships and axioms (formulated in SWRL language which supports first order logic) as well as the selection of learning sources and scalability. We propose here an architecture of ontological components to represent the domain knowledge, the Web sites structure and a set of services.

3.1 Ontological Architecture for Semantic Web

Learning ontologies from Web sites is a complex task because Web pages can contain more images, hypertext and frames than text. Learning concepts is a task that needs texts able to explicitly specify the properties of a particular domain. Starting from the state of the art, we can say that no available learning method to extract concepts and relationships is better than another. On the Web, we can identify three types of knowledge concerning a target domain: *general and common knowledge, knowledge related to services of domain* and *knowledge relates to structure of Web sites*.

Relations which exist between domain, structure and services knowledge are not explicitly formulated. This is due to the use of HTML language. In fact, it is not possible to identify automatically from a HTML document its semantic structure or services that it offers. General semantic related to a domain is communicated by the textual resources on the Web. Besides, the semantic behind the design of a Web site is not the result of a compromise between the designer of Web site and surfers. Unrolling of domain services is not clearly distinguished from the other knowledge and cannot be extracted in a formal way. In this architecture, we propose a set of interdependent ontologies to build a knowledge base of a particular domain, constituted by a set of Web documents, and their structure and associated services, as depicted in Figure 2. Thus, we distinguish three ontologies, namely a *generic ontology of Web sites structures*, *domain ontology* and *ontology of domain services*.

Domain ontology is a set of concepts, relations and axioms that specify shared knowledge concerning a target domain. Ontology of domain services specifies for each service, *its provider*, *its interested users*, *possible process of its unrolling*, *main activities and tasks composing this service*. This ontology contains axioms specifying the relations between domain services and precise main domain concepts which identify each service.

The ontology of Web sites structure contains a set of concepts and relationships allowing a common structure description of HTML, XML and DTD Web pages. This ontology enables to learn axioms that specify the semantic of Web documents patterns. The main objective is to ease structure Web mining knowing that the results can help to populate the domain ontology. These ontologies will be detailed in respective 3.1.2.1, 3.1.2.2 and 3.1.2.3 subsections. These ontologies are generated by a Metaontology which is the main component of the proposed architecture.

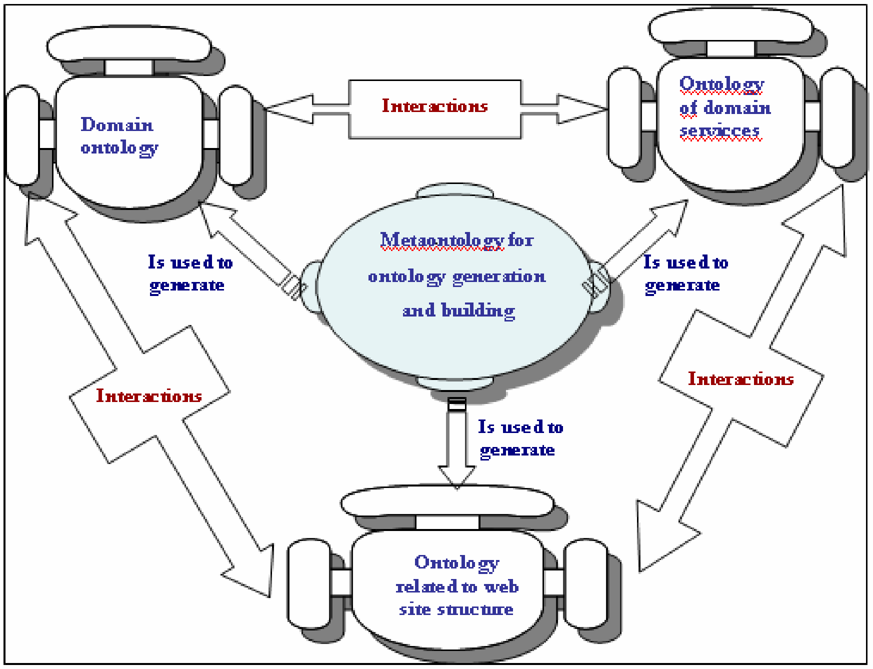

Domain ontology
Interactions
Ontology of domain servicces
Is used to generate
Metaontology for ontology generation and building
Is used to generate
Interactions
Interactions
Is used to generate
Ontology related to web site structure

Fig. 2. Ontological architecture for the Semantic Web.

The Metaontology is a specification of metamodels of domain ontology, ontology of domain services and ontology related to structure of Web site. Besides, knowledge concerning the semi-automatic construction of domain ontology is also specified by this Metaontology. A more detailed description is presented in the following paragraph.

3.1.1 Metaontology Description

Starting from the fact that the proposed architecture is composed of three ontologies, the Metaontology contains: knowledge representation related to each ontology, required knowledge for ontology construction, knowledge representation specified by the metamodel related to these three ontologies.

It is mostly based on three generic concepts: "*Metaconcept*", "*Metarelation*", and "*Metarxiom*". The class of "*Metaconcept*" is divided into three subclasses which represent respectively the domain metaconcept, the metaconcept of domain services and the metaconcept of element related to Web structure. Besides, the class "*Metarelation*" and the class "*Metaaxiom*" are designed in the same way. Figure 3 shows a conceptual model of the Metaontology.

The metaonology is, consequently, made up of three homogeneous knowledge fields. The first field is a conceptualization of knowledge related to learning concepts, relations and axioms related to a target domain. Besides, for each instance of the class "*Domain_Concept*" and the class "*Domain_Relation*", the technique leading to its discovery is specified. The second field is based on the design model related to the ontology of domain services. The third one concerns specific knowledge related to Web site structure. These elements can contribute to discover semantic relations between domain concepts and domain services. It should thus be possible to adjust the presentation of elements in a Web page according to semantic relations between domain concepts. In addition to these generic concepts, the Metaontology is based on further generic relations and axioms. Generic relations specify the relation between the three ontologies of the architecture.

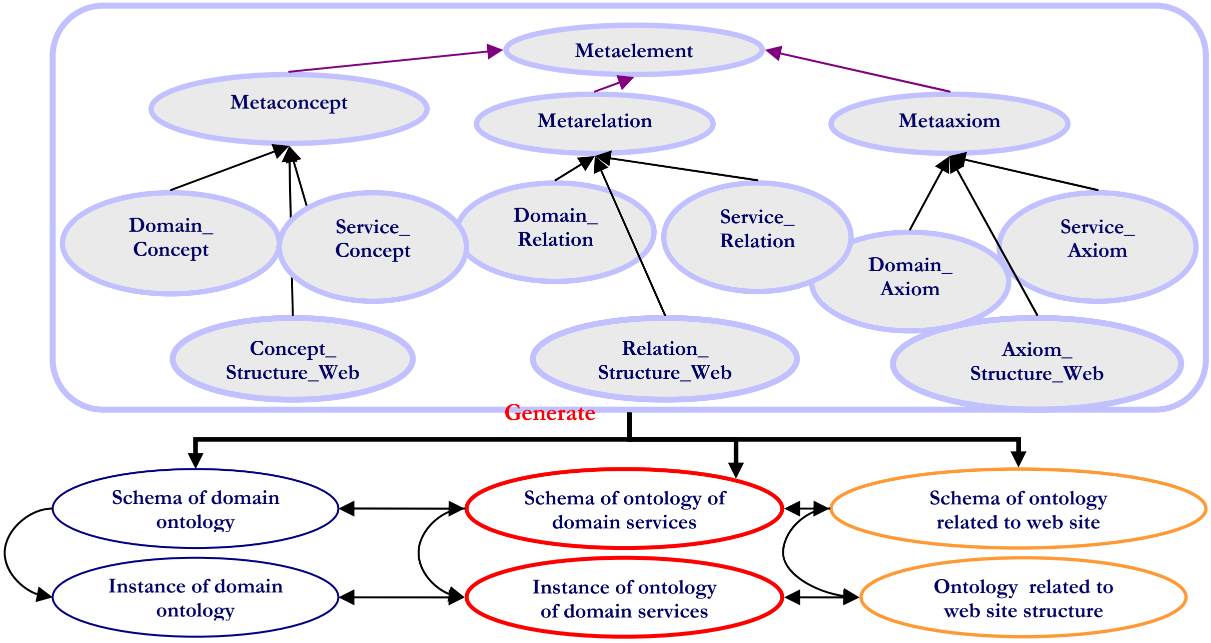


Fig. 3. General description of Metaontology.

Axioms are formulated in first order logic and they are edited in the following form: *"IF <condition> then <Consequence>"*. Conditions and consequences are formed by using Metaconcepts and Metarelations. These generic axioms specify constraints linked to the existing relations between Ontologies. In the following, we describe the Ontologies generated with the help of the Metaontology.

3.1.2 Ontologies Generated with the Metaontology

3.1.2.1 Domain Ontology

Concepts related to domain ontology definition and learning, were identified according to the survey elaborated from the main approaches related to ontology learning from texts and from Web pages. Thus, in addition to the metamodel of the three ontologies, the required information dealing with concepts, relations and axioms learning is also specified in the Metaontology. These elements are domain independent. Concepts and relations of domain ontology are designed respectively as the instances of the generic concepts "*Domain-concept*" and "*Relation-domain*". This makes it possible to separate the ontology design from the ontology operationalization. Figure 4 show the relation between the elements of the domain ontology and the technique involved in their discovery which are maintained and capitalized in the Metaontology.

The domain ontology schema generated by the Metaontology is constituted by a set of five elements: (1) *two disjoint sets of concepts and relations* designated respectively "*C*" *et "R",* (2) *a concept taxonomy* "*HC*" where HC⊆ C x C. HC (C1, C2) means that C1 is hyponym of C, (3) *a relation hierarchy "HR*" which defines non taxonomic relations where HR⊆ R x R. HR (R1, R2) means that R1 is subclass of R2 and (4) *three functions* are associated with this hierarchy (The function "*Relation*": R → C x C where relation (R) = (C1, C2), the function "*Domain*": R → C where domain (R):= C1 and the function "Range": R → C where range (R):= C2), finally, (5) *axioms set*.

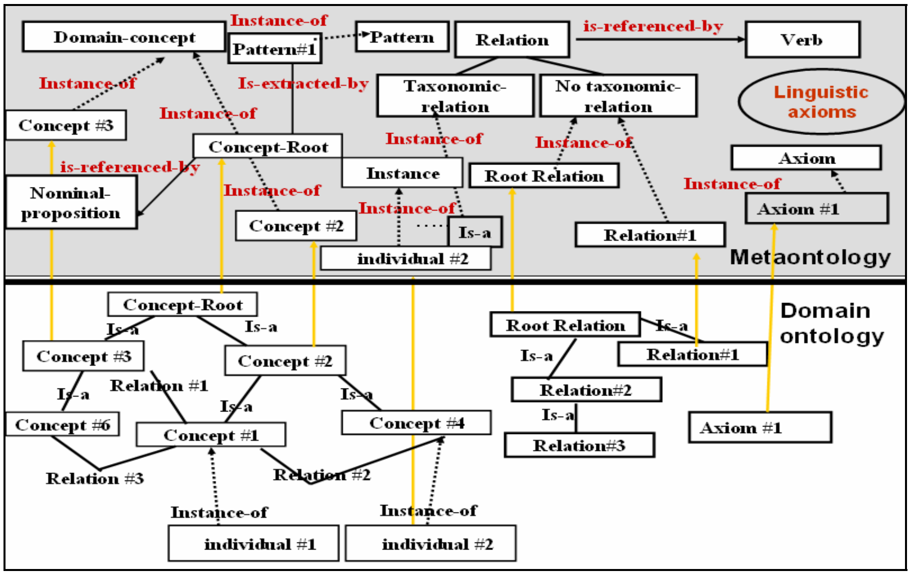


Fig. 4. Domain ontology.

Two types of axioms are distinguished: generic axioms and specific axioms. The generic ones specify restrictions concerning the relation between the concepts related to domain. The specific axioms deal with constraints concerning relation instances or concept instances.

3.1.2.2 Ontology of Domain Services

Our desire to define an ontology related to domain services comes from studying the task ontology. In a Web context, the most important problem is to find easily and fast relevant knowledge satisfying a solicitation of a target service.

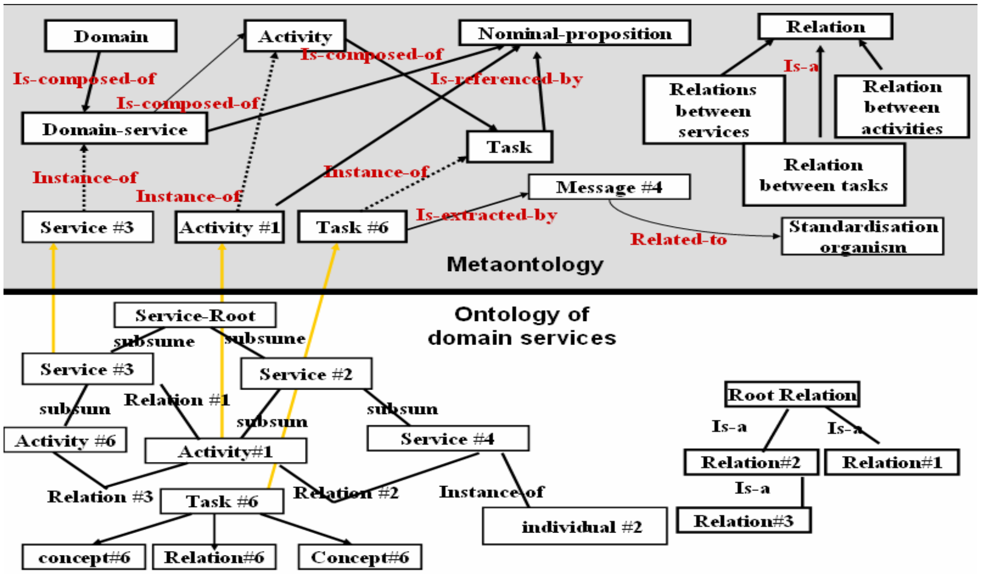


Fig. 5. Ontology of domain services.

Defining an ontology of domain services requires more than classical research based on keywords. In order to ease this task, an ontology related to domain services has to be defined, as shown in Figure 5, in order to specify: *a set of the domain services*; *the main activities* involved in the user's queries or in the provider's services; *the tasks* related to these activities; *the relations existing between services*, *subservices, activities and tasks*; *the management of rules related to the execution of services*. Besides, this ontology can be a good support to express users' queries with a structured vocabulary. It consists in a set of predicates modeling the services of interest.

3.1.2.3 Ontology Related to Web Site Structure

The third component of the proposed architecture is complex as it deals with Web site structure. It is a conceptualization of the semantic related to Web sites structure. The structure recognition of a Web site cannot be an automatic task; consequently, much of the related work deals with the recognition of structured patterns. The complexity of this task comes from the fact that the logical structure is not explicitly expressed by the *HTML* language.

We propose to specify these patterns and the related axioms which allow the extraction of semantic structure of sites. This ontology is useful to specify the patterns extracted by applying structure mining techniques; restructure the Web site according to the ontology of Web site structure and to generate this ontology from the Metaontology. Axioms which enable us to reason on the basis of structural elements of a Web site generate the semantic structure of each Web page. We can extract knowledge from these Web pages. Such a semantic structure will then be specified in the Web structure ontology.

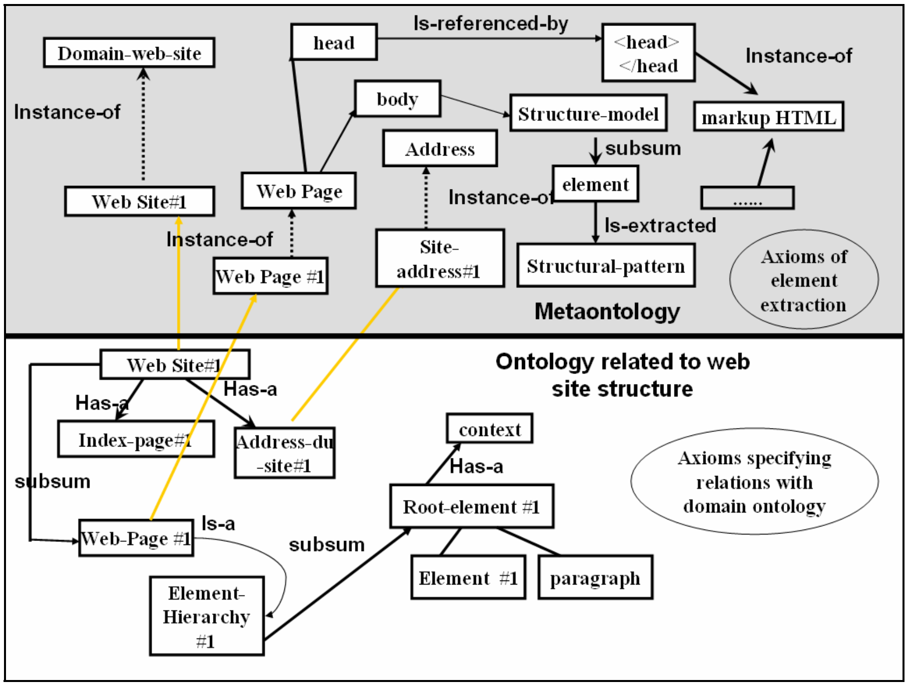


Fig. 6. Ontology of the Web structure site.

In Figure 6, we have presented some concepts of the ontology of Web structure and the Metaontology. The semantics of each markup are formulated with a relevance level. For example, the word "*presentation*" found near the markup "*<P>*" does not have the same relevance level as the word "*hotel*" associated of one of the following markup *(<EM> , <B>, <STRONG>, <DFN>, <SAMP>).* In fact, the word hotel represents a more relevant concept than the word “presentation” for our application domain. Thus, axioms reasoning on this level of relevance based on markup are also specified to enrich domain ontology with relevant concepts. An example of such an axiom is: $\forall$ *comboboxlist (C, x), c* $\in$ *Concepts (C)* $\rightarrow$ *x* $\in$ *instances(C).*This axiom instantiates the concept C referenced by the label of the combobox with the elements x of its list.

In this section, we have presented the particularities and the role of each ontology proposed in our ontological architecture. This architecture enables the representation of domain knowledge to be used on the Web, the relations between the ontologies which make up the architecture add more formal structure to the Web, Metaontology as a fundamental part of our architecture. This one is used to generate three ontologies, besides it plays an important role in the construction of domain ontology from Web pages. So, it has a particular life cycle which is describes in the following section.

3.1.3 Life Cycle of the Metaontology: Conceptualization, Ontologization and Operationalization

Our contribution lies in specifying knowledge related to the representation and the construction of the domain ontology, the ontology of domain services and the ontology related to Web site structure. However, this specification cannot be qualified with consistence and completion. In fact, the Metaontology is built in an incremental way as it is not possible to define all the required and sufficient knowledge for ontology construction instantaneously. New techniques could be discovered and other new approaches could be proposed in the field of ontology engineering. That is why Metaontology enrichment and maintenance are needed. The Metaontology is usable only after its instantiation by a data source and its adjustment by the ontology engineer. Figure 7 shows the life cycle of the Metaontology.

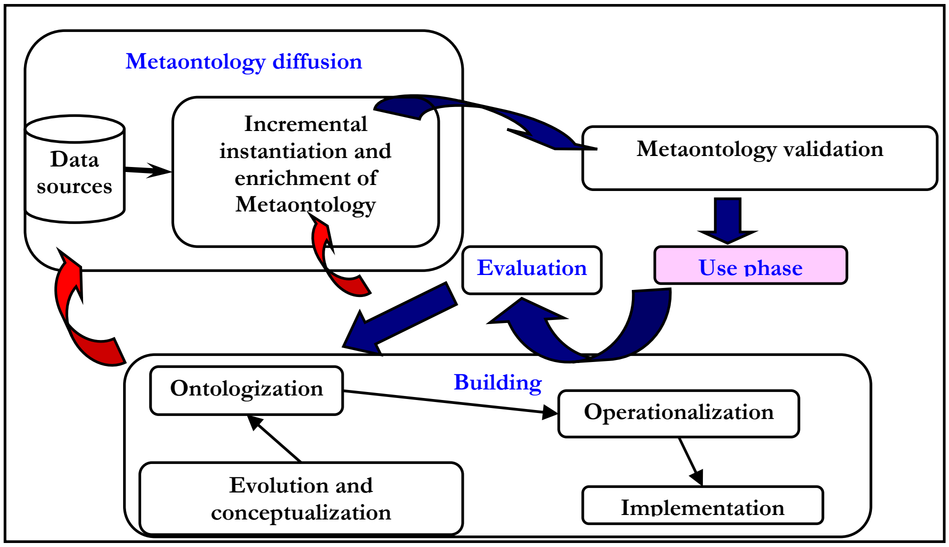


Fig. 7. Metaontology life cycle.

Metaontology instantiation is carried out with real instances of the metaconcepts found in free texts. Its *validation* is an important step, especially in the context of an incremental instantiation with many sources. The main purpose of this phase of *Metaontology validation* is to maintain the state of concepts and relations discovered as candidate elements in the Metaontology (“*valid*”, “*invalid*”, “*candidate*”, “*excluded*”). The phase related to using Metaontology consists in using of a Metaontology instance in order to generate ontology schemata. *The evaluation phase* is important when updating knowledge specified by the Metaontology. It is also based on the evaluation of the generated ontology schema. Maintaining the conceptualization of the Metaontology, as well as its *maintenance*, *ontologization* and *operationalization* is triggered in order to update the original Metaontology. The cycle can then start again.

3.2 Incremental Approach of Domain Ontology Building: "LEO-By-LEMO"

In this section, we describe, at first, the incremental and semi-automatic process of ontology building from Web content according to "LEO-By-LEMO" approach. Then, phases that make up this process are detailed in the following subsection.

3.2.1 Incremental And Semi-Automatic Process Of Ontology Building From Web Content

Our approach, namely LEO-By-LEMO ("*LEarning Ontology BY LEarning MetaOntology*") is based on learning rules of ontology extraction from texts in order to build ontologies. It suggests a process illustrated by Figure 8 and based on three main phases:

Initialization phase;

Incremental phase of learning ontology;

Result analysis phase.

The initialization phase is dedicated to data source cleaning. The input of this phase is constituted by a minimal ontology, the Metaontology, the terminological resource "*Wordnet*" and a set of Web sites classified by domain services. The second phase is a learning iterative process. Each one of the iterations is made up of two main steps. The first one is the Metaontology enrichment and the second one enables us to apply the Metaontology axioms related to the learning of ontology. The last phase is useful to verify the Metaontology coherence by analyzing learning results. In the following section, we will describe each phase.

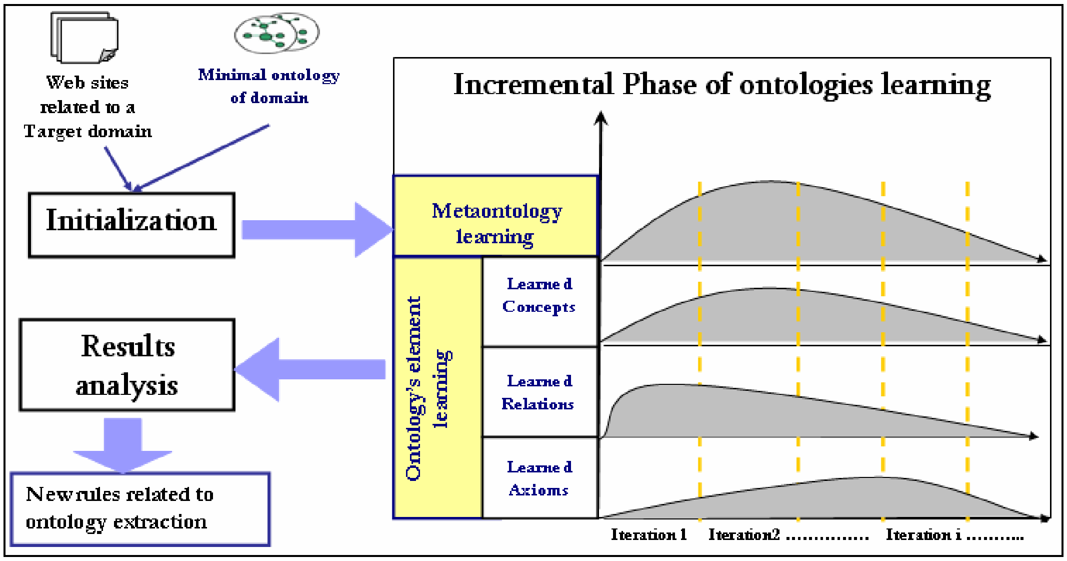

Web sites related to a Target domain
Minimal ontology of domain
Initialization
Results analysis
New rules related to ontology extraction
Incremental Phase of ontologies learning
Metaontology learning
Ontology's element learning
Learned Concepts
Learned Relations
Learned Axioms
Iteration 1
Iteration2 ....
Iteration i ....

Fig. 8. Incremental approach of semi-automatic ontology building.

3.2.2 *Description of the Phases of "LEO-By-LEMO" Approach*

3.2.2.1 *Initialization Phase*

A minimal ontology is designed and built to be enriched in the second phase of the approach. It is called "*minimal domain ontology*" as the number of concepts and relations are reduced. Consequently, data source preparation consists in: *searching Web documents related to the domain corresponding to a query based on concepts describing a target service* (these concepts are obtained from the projection of the corresponding service specified in the ontology of domain services), *selecting a dozen Web sites provided by a Meta search engine tool* (The number of chosen sites is limited because analyzing an important number of Web pages requires very important execution time), *classifying Web pages according to domain services*, finally, *cleaning Web pages* by eliminating markup elements and images, text segmentation and tagging in order to obtain a tagged textual corpus. One hypothesis is that we deal with Web documents written in a target language. The Metaontology adjustment is thus done according to linguistic knowledge related to the target language.

3.2.2.2 *Incremental Phase of Domain Ontology Learning*

Starting from the fact that domain ontology learning causes numerous problems (semantic disambiguation, selecting referent terms for concepts, taxonomic relation extraction, and textual corpora extraction), an iteration is not sufficient to learn a coherent and complete ontology. For this reason, we have designed an incremental phase where the textual corpora is updated to enrich the ontology and to validate the discovered ontological elements. These iterations, consequently, enable the ontology engineer to obtain a better semantic coverage of the target domain, to learn new extraction rules which are not dependent from a corpus and to correct errors related to the techniques used for ontology learning.

An iteration is processed in two steps. In the first step, techniques identified by our approach survey are applied to the corpus. In this context, we have adapted the construction of a word space [5] by applying the N-Gram analysis instead of a 4-gram analysis. We have also proposed a disambiguation algorithm shown in Figure 9. It aims to determine the right sense of a lexical unit. This algorithm is based on the study of term co-occurrence in the text and the selection of the adequate sense. Besides, we propose to use many similarity measures to build the similarity matrix which describes the contextual similarity between concepts. The application of these techniques causes the extraction of conceptual elements from texts which instantiate abstract concepts of the Metaontology. This step is called "*Metaontology alimentation*".

A) *Metaontology Alimentation*

In the following, we will describe the implemented tasks related to alimentation of Metaontology. Firstly, for each concept "C" of the minimal domain ontology related to the current iteration, the involved tasks are the following: The first one is *the Extraction of nominal clauses which contain terms that refer to the same concept from textual corpus:* Metaontology is enriched by these nominal propositions associated to their occurrence probability. A nominal clause is a group of terms that can act as subject, direct or indirect object, subject complement or object complement in a sentence. The occurrence probability is the division of frequency of nominal propositions in the corpus by frequency of referent terms in the corpus.

The second one is *the extraction of synonyms, hyponyms and hyperonyms using Wordnet in the case where the term is monosemic*: Synonyms are added to Metaontology as instances of metaconcept "Term". If the referent of the target concept has many senses according to Wordnet, the task of semantic disambiguation starts.

Then, *the extraction of Topic signature of the concept "C"* has to be done when a topic signature or semantic signature is defined as "*a set of concepts which appear frequently in most sentences or Web pages*". Terms having a frequency superior to empirical threshold are selected as belonging to the semantic signature of concept "*C*". This threshold is fixed as:

$$\sum_{i=1}^{Number\_of\_concepts} \text{frequency (Concept) / Total number of concepts making up the semantic signature.}$$

The following task is the semantic disambiguation which is done according to the following algorithm showed by Figure 9. The last task is Metaontology enrichment by synonyms, hyponyms and semantic signature extracted below.

```
Suppose L= a set of terms which refer to concepts making up semantic signature of a concept C

Suppose LConcept = a set of terms which refer to concepts of minimal domain ontology.
For each sense "i" identified in wordNet do
  {   create_List_synonyms(Lsyn, i);
      create_List_Hyponyms(Lhypo, i);
      create_List_Hypernyms(Lhyper, i);
      create_List Def(Ldef, i) ; // a set of terms appearing in its definition from WordNet
      ContexteSense (C, i ) = Lsyn(i) ∪ Lhypo (i) ∪ Lhyper (i) ∪ List Def(Ldef, i).
      Intersection (T,i)= ContexteSense (T, i) ∩ L
  }
Select a set of sens j where Intersection (T, j) had a maximum of elements
If more than one sense is selected than
     Select sense j where List Def (j) ∩ LConcept (j) has maximum of elements
Else
     Concept admits a disambiguation state in Metaontology.
```

Fig. 9. Semantic disambiguation algorithm.

Secondly, for each non taxonomic relation specified in the minimal domain ontology, Metaontology is alimented by:

Lexico-syntactic patterns associated to its frequency in the corpus;

Patterns of sentences related to the verb that refers this relation;

Syntactic frames related to the verb refering the current relation.

A *syntactic Frame* of a relation labeled "*Travel*" is illustrated by this example: "*<To travel><subject: human><by: vehicle>*". According to this syntactic frame, many classes and instances can be extracted such as the following ones:

<travel> (<subject : Jean>) (<by : voiture>);

<travel> (<subject : David>) (<by : train>).

Then, the construction of concepts space is implemented. It represents a multidimensional space which associates a dimension to each concept. Then, each concept is represented as a vector of the other concepts. This space is useful to calculate similarity matrix. To obtain this concept space, a co occurrence matrix is built for each pair of concepts *C1* and *C2*. The co occurrence between concepts is determined by calculating frequency between corresponding nominal propositions, synonyms and hyponyms of the first concept *C1* with the corresponding ones of the second concept *C2*.

Then, a context vector id determined for the concept *C1* and *C2*. A context vector of concept *C1* is obtained by adding co-occurrence vectors related to *C1*. Each context vector is then divided by the maximum of occurrence of the two concepts to obtain concept space vector of each concept. From this concept space, we can determine the semantic distance between concepts by building similarity matrix for each similarity measure. In our case, we have implemented two measures among many others specified by Metaontology to test it. These measures belong to the LP norm family, we distinguish:

Euclidian Distance : $D(T_j, T_r) \equiv \sum_i^{Nbr\_of\_document} \sqrt{(f_i(T_j) - f_i(T_r))^2}$;

Manhattan distance (City-block) : $D(T_j, T_r) \equiv \sum_i^{Nbr\_of\_document} \left| f_i(T_j) - f_i(T_r) \right|$.

B) Ontology Elements Learning

Ontology element learning is an iterative step. Based on the new instances created at Metaontology alimentation, Metaontology axioms are then applied to deduct new candidate concepts and relations in each iteration. The corpus is then updated, and lexico-syntactic patterns and syntactic frames obtained within Metaontology alimentation are applied in order to learn new elements and to modify the state of ontology elements discovered as instances of metaconcepts and metarelations of the Metaontology. The concepts and relations having a valid state are then generated to obtain an ontology schema which is now the input of the following iteration as shown in Figure 10, and is realized by implementing the following tasks: (1) *Learning new concepts and relations from nominal expressions*; (2) *Learning non taxonomic relations by clustering syntactic frames specified in the Metaontology;* (3) *Searching pairs of terms by applying lexico-syntactic patterns specified in the metaontolog and* (4) *Extracting Lexico-syntactic pattern from updated corpus between most similar concepts according to similarity matrix.*

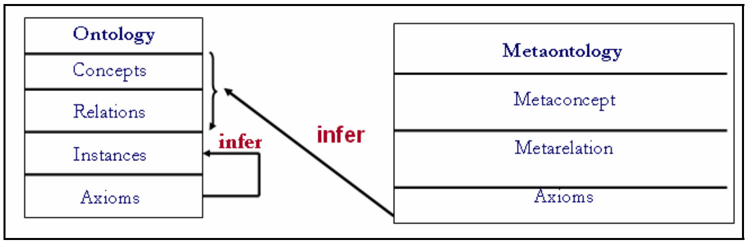


Fig. 10. Use of Metaontology in deducing candidate concepts and relations of domain ontology.

After all the required iterations, the last phase is carried out by the ontology engineer. This phase is the result analysis one and is described in the next subsection.

3.2.2.2 Results Analysis Phase

The phase of results analysis is the last phase of the proposed process. It represents the only phase that really needs the intervention of the ontology engineer. In fact, the ontology engineer consults both the domain ontology generated and the Metaontology to detect any conceptual mistakes. Next, based on the results specified in the Metaontology, it is possible to deduce the techniques responsible for wrongly discovered elements in the generated ontology. The ontology engineer may thus maintain the extraction rules specified by the Metaontology. The adjustment of rules in the Metaontology analysis will improve the ontology construction within further execution of the proposed process. In order to study the feasibility of the proposed approach a development of the framework "*OntoCoSemWeb*" (*ONTOlogy COmponent learning tool for SEMantic WEB*) has been done.

4 OntoCoSemWeb Development

The architecture of OntoCoSemWeb shown in Figure 11 supports the proposed approach. It is composed of: (1) *a pre-treatment module of data sources* which perpetrates textual corpus, its POS (Part-of-Speech) tagging and importation of terminological and conceptual resources (minimal ontology, Ontology of domain services and terminological resource "*Wordnet*"); (2) *an editing module of the Metaontology* which allows concepts and ontology axioms update by integrating the Plug-in of Protégé-OWL tool; (3) *a module of domain ontology* generation; (4) *an alimentation module of the Metaontology* which consists to project conceptual elements in the Metaontology from text and implements the first step of the second phase (i.e., Metaontology alimentation step related to incremental phase of *domain ontology learning*) of the incremental process of ontology domain construction proposed by the "*LEO-By-LEMO*" approach and (5) finally, *a module of domain ontology learning* which is the result of association and development of a set of learning techniques of concepts and relations. This module collaborates with an inference engine to add ontological elements by specified axioms and Metaontology to enrich knowledge database of ontology extraction from Web pages.

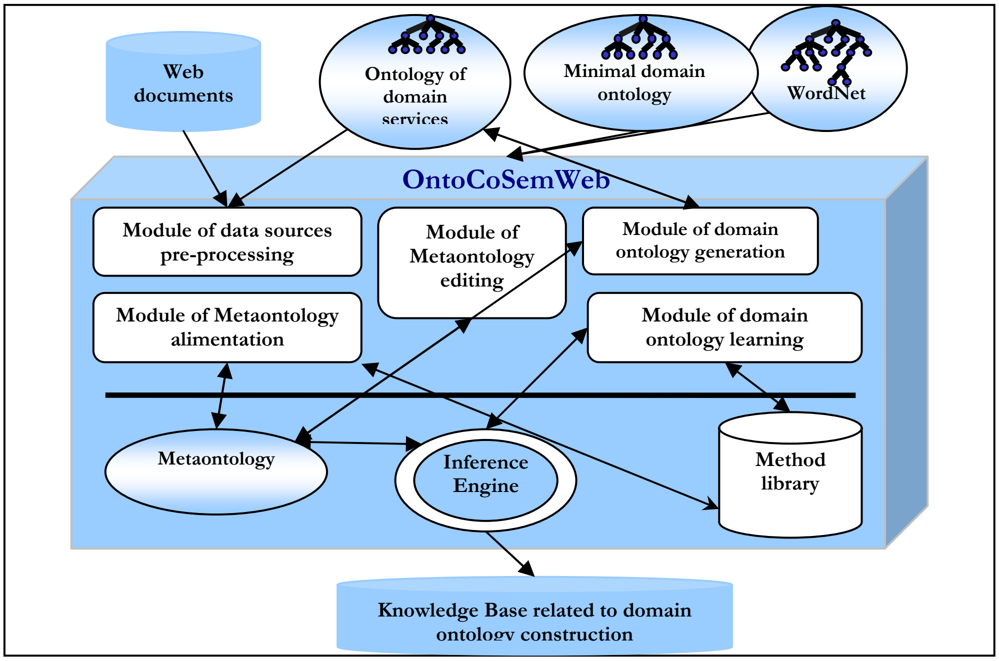

Web documents
Ontology of domain services
Minimal domain ontology
WordNet
OntoCoSemWeb
Module of data sources pre-processing
Module of Metaontology editing
Module of domain ontology generation
Module of Metaontology alimentation
Module of domain ontology learning
Metaontology
Inference Engine
Method library
Knowledge Base related to domain ontology construction

Fig. 11. OntoCoSemWeb architecture.

The functionalities of each identified phase of the proposed method are implemented by using tools as: (1) Protégé2000 for tourism minimal ontology edition; (2) TreeTagger for tagging the terms; (3) Copernic to collect Web sites in tourism domain; (4) Teleport Pro to crawl collected Web sites; (5) JbuilderX to develop the main programs of the proposed approach. The projects JENA, API DOM (*Document Object Model*), the plug-in SWRL, the inference engine Bossam, the source code of Protege-OWL, have been used.

5 Case Study and Experimentations

In the last sections we have presented an ontological architecture (composed of three ontologies and a Metaontology) and an incremental approach construction of the first component (the domain ontology) in conformity with the ontological architecture. A first experimentation of these propositions consists in validating the Metaontology and the incremental construction of ontologies. So, the framework OntoCoSemWeb (*ONTOlogy COmponent learning tool for SEMantic WEB*) is developed and will be used in Semantic Web context and more precisely for Semantic Information Retrieval. Then, a second experimentation demonstrates the usability of the proposed architecture for the on-line Semantic Information Retrieval.

5.1 OntoCoSemWeb Experimentation Scenario in the Tourism Domain

According to the proposed framework, the experiment scenario is carried out in a semi-automatic way. In fact, the construction of ontological components for the Semantic Web could not be completely automated because we have to deal with a semantic level which involves human validation. For this reason, we use OntoCoSemWeb for the implemented task related to the incremental phase of the proposed approach. Results analysis is performed by experts in ontology engineering, domain knowledge and Semantic Web technologies fields, who are able to maintain the conceptual model and axioms of the Metaontology. In the following, we illustrate the different phases of our approach in the tourism domain.

5.1.1 Initialization Phase

The initialization phase starts with the construction of a minimal ontology for the field of tourism. We take the lodging service as an application example. The set of documents related to this service is pre-processed in order to obtain a tagged textual corpus. The minimal ontology construction is carried out by using the ProtégéOWL tool [34]. Figure 2 shows the minimal ontology which contains eight concepts (“*Concept-Lodging*”, “*Concept-hotel*”, “*Concept-person*”, “*Concept-tourist*”, “*Concept-Transport-mean”*, “*Concept-boat”*, “*Concept-train*” and “*Concept-bus*”) and six relations (“*has_an*”, “*has_name*”, “*has_star*“,“*travel_by_train*”, “*travel_by planes*”, “*travel_by_ship*”).

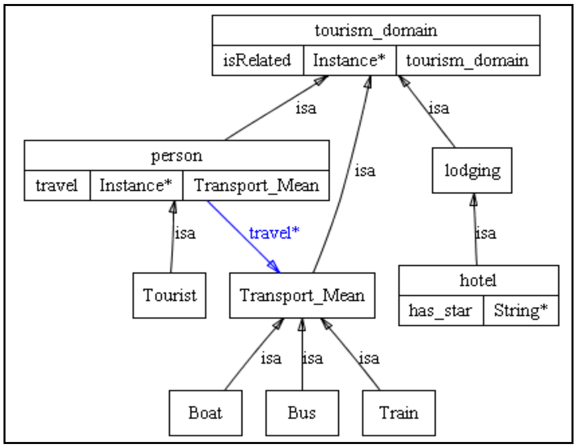


Fig. 12. Concept hierarchies and relations of the minimal ontology.

The linguistic axioms are built using the rules editor integrated into the Protégé2000 environment. This editor allows us to formalize the rules in first order logic using SWRL (language resulting from the combination of “*RuleML*” and “*OWL*”). Data sources used during the experimentation are obtained by searching on the Web using Copernic. The query is composed of the following keywords: tourism, lodging, accommodation, housing, Paris, service. The designed framework generates a tagged textual corpus from the results obtained with Copernic. During this step, the user can edit linguistic axioms or formulate new extraction rules. Next, we start the first iteration of the incremental phase by enriching the Metaontology. The concept “*lodging*” is chosen by the user to begin the incremental learning.

5.1.2 Extraction of Nominal Expressions and Semantic Signature Related to “lodging” Concept

The Metaontology enrichment starts by searching synonyms, “part-of” relations and nominal expressions referring to the concept “lodging”. The extraction of each nominal expression is done by searching the nominal expression containing the *lodging concept* or one of its synonyms (for example “*housing*” or “*living accommodation*”).

A lexico-syntactic pattern and its occurrences are associated with each nominal expression. OntoCoSemWeb allows the insertion of a nominal expression related to the concept lodging. The following step allows the enrichment of the Metaontology with the nominal expressions of the given concept, the occurrence of this concept in the corpus and the occurrences of each nominal expression as shown in Figure 13.

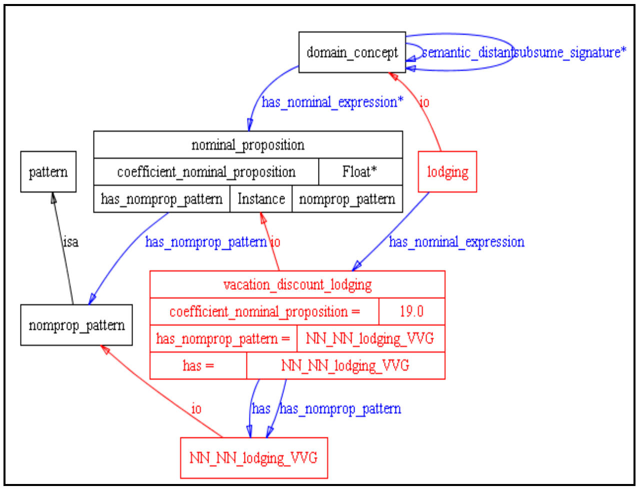
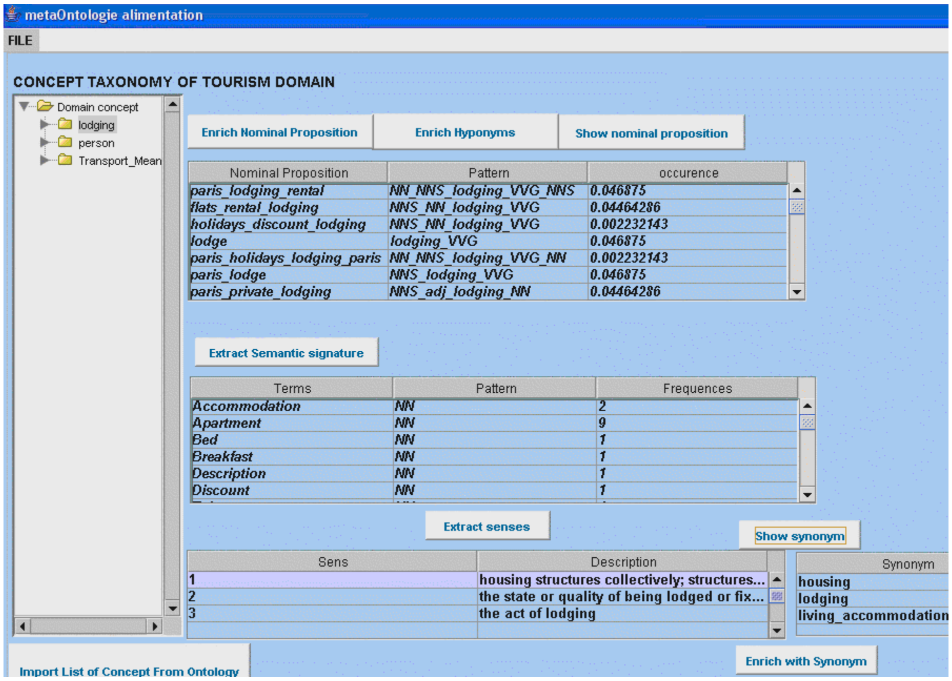


Fig. 13. Sample of a nominal expression inserted in the Metaontology.

The existing generic linguistic axioms stored in the Metaontology are used to deduce new concepts or instances (from nominal expression patterns or other lexico-syntactic patterns). Nominal expressions are labelled with lexical roles such as noun, adverb, etc. These labels become lexical patterns of the nominal expressions. The concepts which are similar to the concept “*lodging*” (*palace* for example) are extracted.

Indeed, with OntoCoSemWeb, Figure 14 shows that the user can select a concept (for example: “*lodging*”) and apply the functionality “*Extract senses*”, which allows the visualization of the meanings of the word “*lodging*”, given by Wordnet. The “lodging” concept is referenced by nominal propositions such as “holidays_discount-lodging” tagged with the corresponding lexical pattern “noun (NN), plural noun (NNS), verb (VV)” as depicted in Figure 14.

Table 2 show terms that represent the concept characteristics which contain adjectives (“*adj*”) as “*short term*” and "*private*", prepositional expressions of the name "*lodging*" as possessive expressions (example: "*Lodging's place*" thus “has place" is a new candidate relation) and the terms appearing after the verbs “*to be*” and “*to have*” in sentences containing the term “*lodging*”. In this table, “NN” stands for noun, “NNS” for plural noun and “adj” for adjective. When a term is composed by several words, the lexical pattern associated is composed by the concatenation of the lexical patterns corresponding to each word (i.e., the pattern associated to "Holidays discount" is "NNS_NN").


metaOntologie alimentation
FILE
CONCEPT TAXONOMY OF TOURISM DOMAIN
Domain concept
lodging
person
Transport_Mean
Enrich Nominal Proposition
Enrich Hyponyms
Show nominal proposition
Extract Semantic signature
Extract senses
Show synonym
Import List of Concept From Ontology
Enrich with Synonym

Nominal Proposition	Pattern	occurence
paris_lodging_rental	NN_NNS_lodging_VVG_NNS	0.046875
flats_rental_lodging	NNS_NN_lodging_VVG	0.04464286
holidays_discount_lodging	NNS_NN_lodging_VVG	0.002232143
lodge	lodging_VVG	0.046875
paris_holidays_lodging_paris	NN_NNS_lodging_VVG_NN	0.002232143
paris_lodge	NNS_lodging_VVG	0.046875
paris_private_lodging	NNS_adj_lodging_NN	0.04464286

Terms	Pattern	Frequences
Accommodation	NN	2
Apartment	NN	9
Bed	NN	1
Breakfast	NN	1
Description	NN	1
Discount	NN	1

Sens	Description
1	housing structures collectively; structures...
2	the state or quality of being lodged or fix...
3	the act of lodging

Synonym
housing
lodging
living_accommodation

Fig. 14. Screenshot of Metaontology enrichment in OntoCoSemWeb.

Table 2. Lodging characteristics.

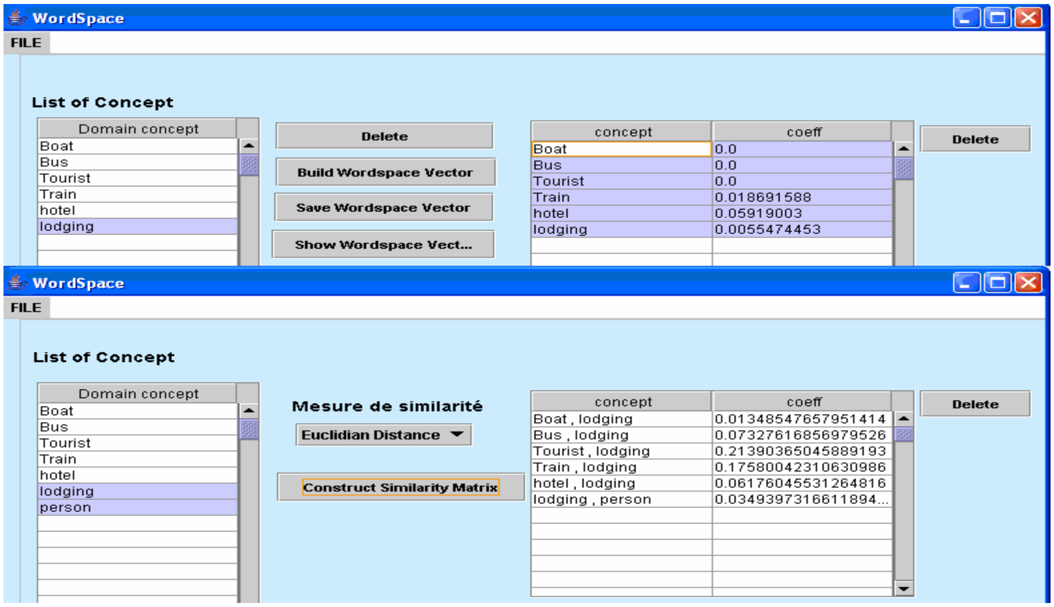
Frequent terms in extracted nominal expressions	Lexical patterns	Occurrence Probability
Holidays discount	NNS_NN	0.045454547
Holidays	NNS	0.045454547
Rental	NN	0.90909094
Vacation	NN	0.8636364
short_term	adj	0.90909094
Private	adj	0.90909094
Vacation discount	NN_NN	0.8636364

The second step allows for the location of the terms that co-occur with "lodging" in the context of sentence. The semantic signatures represent close concepts. Some of them have no taxonomic relation in the domain ontology. We then have to verify the existence of a taxonomic or a non taxonomic relation, in order to filter the semantic signatures list related to a given concept. In the case of Hotel/restaurant, no existing pattern identifying a relation has been discovered during the first step. The concept "restaurant" doesn't occur in the domain ontology but appears in the semantic signatures list. The existence of a same pair in Wordnet allows us to identify the new concepts "restaurant" and "building" and a taxonomic relation.

5.1.3 Semantic Disambiguation of « Lodging » Concept

The semantic signature contributes to realize the disambiguation algorithm. Indeed, each sense of the term "lodging" is described and presented by a set of *synonyms*, *hyponyms* and *hypernyms*. We note that: *ContextSens1 ("Lodging") ∩ Semantic_Signature ("Lodging") = {appartment, living accommodation, flat, housing}*. A synonyms reduction is done by using WorldNet to obtain the following set {(flat, apartment), living accommodation}. In fact, the intersection of the sets ContextSens and Semantic_Signature is a subset of synonyms, hyponyms and/or hyperonyms related to a sens i. In which case a learning of a new concept tagged by lexical unit "flat" and its synonym "apartment" is added and The hyponymy relation between lodging and flat is done. The disambiguation step allows for the enrichment of the Metaontology with the hyponyms extracted from *Wordnet*. After the Metaontology enrichment with possible referent of the lodging concept and its candidate hyponyms, the same steps must be repeated for the other concepts of minimal ontology. The construction of concepts space referring to all the other concepts of minimal ontology is done. In fact, in the Metaontology, a distinction is made between concepts and their referent terms. This multidimensional space is specified to represent concepts and not terms. So it is rather called concepts space than *word space*.

5.1.4 Construction of Concepts Space and Similarity Matrix



Domain concept
Boat
Bus
Tourist
Train
hotel
lodging

concept	coeff
Boat	0.0
Bus	0.0
Tourist	0.0
Train	0.018691588
hotel	0.05919003
lodging	0.0055474453

Domain concept
Boat
Bus
Tourist
Train
hotel
lodging
person

concept	coeff
Boat , lodging	0.01348547657951414
Bus , lodging	0.07327616856979526
Tourist , lodging	0.21390365045889193
Train , lodging	0.17580042310630986
hotel , lodging	0.06176045531264816
lodging , person	0.0349397316611894...

Fig. 15. Screenshot of Construction of the word space and similarity matrix for concept lodging.

Similarity matrix and word space building are done according to text mining techniques allowing for the definition of taxonomic or non taxonomic relations between two concepts of the ontology. The word space is based on the construction of the co-occurrence matrix related to terms. According to the built Concept Space, the user can choose the similarity measure. A similarity matrix between concepts is computed and added to the Metaontology as shown in Figure 15. As an illustration of this step, we propose to compute the similarity vector of the concept "*lodging*", which is added to the Metaontology as an instance of the metaconcept "*similarity vector*" tagged as "*Lodging Vector*". We used Euclidian distance as similarity measure. Thus, in the example of the similarity between the concept "*Lodging*" and the concept "*hotel"*, the co-occurrence of the word lodging with the word "*hotel*", is about 5.919 %.

5.1.5 Learning Domain Ontology Components

The second learning step of our first iteration consists in applying axioms of the Metaontology to the information extracted in the previous step. We start by looking for a relation between the concepts of the minimal ontology and the concept's semantic signature on one hand, and learning new candidate concepts, on the other hand. Consequently, we added to the meta-ontology: 13 concepts with the mention "*non valid candidate*", 13 hyponym candidate relationships and 3 referent terms for the *lodging* concept (e.g., *Housing, living, accommodations*).

These learned concepts are maintained by their states because the ontology engineer intervenes only in the phase of results analysis and after more than one iteration. Then a candidate concept discovered for the first time is "*candidate*". If it is learned by another method, it will be "*candidate not yet validated*". Then, if in the following iteration, an axiom in the Metaontology confirms that it is a good candidate, its state is changed to "*validated*". So, not all the concepts that are discovered in the first time are generated to be in the domain ontology. Only the validate ones are concerned by *Metaontology generation*.

The extraction of a frame for the "*travel*" relation and, building concept space and similarity matrix enable the learning of domain elements ontology during the second step this phase (learning domain ontology elements); for instance, the extraction of the characteristics of the term "*lodging*" (see Table 1) which consists in adjectives, adverbs, sentences and the semantic signature, leads to discovery of new candidate concepts (hyponyms of *lodging*) such as *Block, Camp, Condominium, dwelling, home, domicile, abode, habitation, dwelling house, hostel, youth hostel, student lodging*, etc.); the extraction of taxonomic relations from WordNet, and non taxonomic relations such as "*has_place*", which link the concept lodging to a new candidate concept "*town*".

5.2 Ontological Architecture Experimentation for On-Line Semantic Search on The Web: Development and Tourism Application

In order to demonstrate the utility of the proposed ontologies in Figure 2, we propose an on-line IR system using two ontologies of the proposed architecture (domain ontology and service ontology) and WordNet. The main idea is that, for a particular domain, it could be useful to associate a set of available services to specific domain ontology (tourism, medicine, cultural heritage, etc.). The service ontology is related to tasks, such as, in the tourism domain, hotel booking, car reservation, etc. In the future, this system will be integrated to the prototype OntoCoSemWeb.

The proposed system SIRO [43], is composed of three main modules: query processing and enrichment, search and document processing and finally a module for service classification. The query processing module reformulates the user query using the concept and relation of the domain ontology and WorldNet. The expanded query is submitted to a Web search engine. In a further step, documents will be classified by services using the service's classification module. This module guides the user to build a second query using posted services.

These modules cooperate together. The main originality of this system is the classification by service of the results of a query which can then be used to perform a search based on the corresponding services. Document classification is a process that affects a document in one class or subclass. It is expected that a result classified by category makes the search of information easier. Our classification module will classify, by service, the results of a user query as explained below.

The reformulated query contains a set of concepts which belong to the domain ontology. These concepts are linked to a set of services belonging to the service ontology, more precisely each concept is related to one or more services. For example, the "*Restaurant*" concept is associated to the service

"*Restauration*". This relation is manually built during the construction of the service ontology. Given the concepts added to the query and the link between the domain ontology and the service ontology, we can extract all the services related to these concepts as shown in Figure 16.

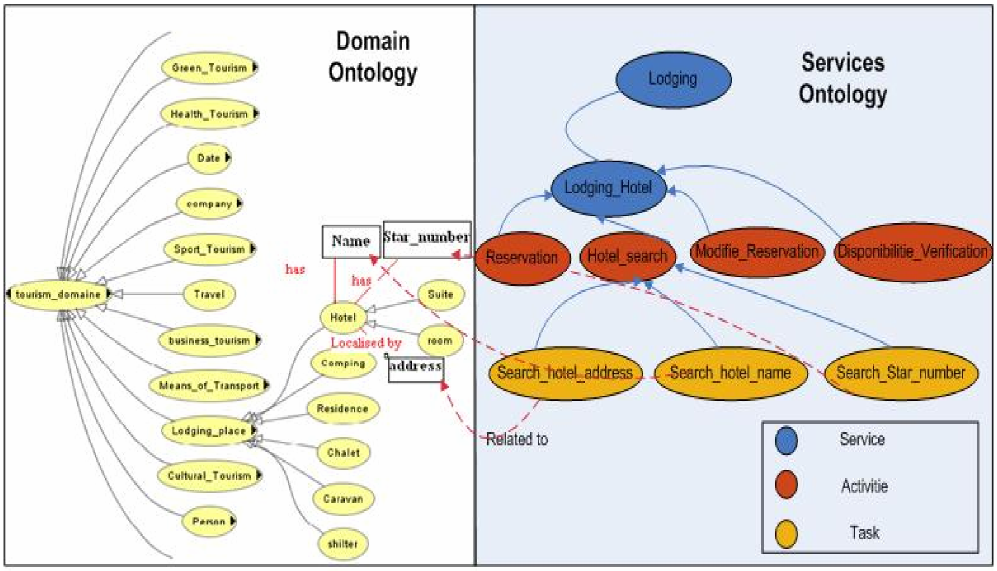
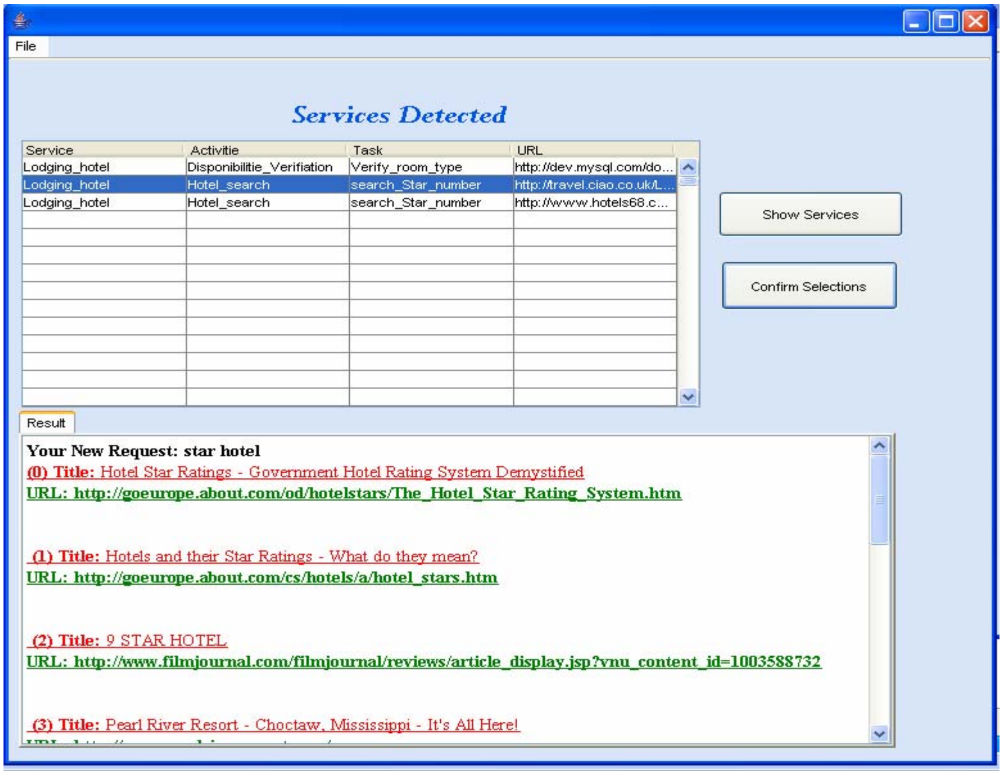


Fig. 16. Relation between the domain ontology and the service ontology.



Service	Activitie	Task	URL
Lodging_hotel	Disponibilitie_Verifiation	Verify_room_type	http://dev.mysql.com/do...
Lodging_hotel	Hotel_search	search_Star_number	http://travel.ciao.co.uk/L...
Lodging_hotel	Hotel_search	search_Star_number	http://www.hotels68.c...

Your New Request: star hotel
(0) Title: Hotel Star Ratings - Government Hotel Rating System Demystified
URL: http://goeurope.about.com/od/hotelstars/The_Hotel_Star_Rating_System.htm

(1) Title: Hotels and their Star Ratings - What do they mean?
URL: http://goeurope.about.com/cs/hotels/a/hotel_stars.htm

(2) Title: 9 STAR HOTEL
URL: http://www.filmjournal.com/filmjournal/reviews/article_display.jsp?vnu_content_id=1003588732

(3) Title: Pearl River Resort - Choctaw, Mississippi - It's All Here!

Fig. 17. Classification by service of the documents.

The vector model is used again to represent a service by a vector: $Serv_i = (C_1, C_2, \ldots\ldots\ \ldots\ldots, C_N)$ with N is the number of concepts related to a service. For each selected document, we compute its similarity with the services by using the cosine formula. The document is then assigned to the more similar service to the document.

Each domain is characterized by a set of services, activities and tasks as depicted in Figure 16. Services are linked to domain ontology concepts (for instance, "lodging_Hotel" service is associated to "Hotel" concept). Each domain concept can be associated to one or more services. This relation between services and domain concepts is exploited to refine user search and to help the user to discover his needs.

Finally, the documents are displayed by the service. For example, an initial query was "*find hotel*". The services, activities and tasks detected from the concepts in the service ontology are: *Service of Lodging_Hotel*, following *activities*: *Reservation*, *Hotel_Search*, *Modify_Reservation*, *Disponibility_Verification*, and following *Tasks*: *Verify_RoomNumber*, *Verify_RoomType*, Search_HotelName, Search_HotelAdress. Each document having a similarity greater than 0, is attributed to a service, an activity and a task. In this context, we have built a domain ontology and service ontology of tourism domain. We analyse the relations between the two ontologies as shown Figure 17 according to the proposed architecture. Indeed, the user can perform a new query based on services according to document classification by services.

6 Analysis of Results

6.1 ONTOCOSEMWEB Result Analysis

After three iterations of the second phase of the proposed approach, the analysis phase allows us to show how the Metaontology can help the ontology engineer to evaluate the construction of rules specified in the Metaontology. In fact, the state of the enriched Metaontology is used at the end of the second phase to make conclusions related to the applied techniques. Comparisons have been done to evaluate the contributions of OntoCoSemWeb. The comparison was done between the results obtained by our prototype OntoCoSemWeb and the common knowledge of the ontology engineers resulting from their observation. Furthermore, the number of concepts learned is much more important comparing to other semi-automatic ontology building tools. The non valid rules are corrected according to the obtained conclusions. In fact, the analysis of the obtained results shown in Figure 18 confirms that using lexico-syntactic patterns is not the right technique to discover taxonomic relations between concepts. In the Metaontology, the lexico-syntactic patterns deleted by the ontology engineer are consequently marked as "*excluded*" and will not be considered in the next iteration. The framework is thus more flexible. It enables easy maintenance of errors which had appeared by updating axioms of the Metaontology.

The main objective of the experiment presented in the paper was to show the feasibility of our approach to ontology construction for the Semantic Web by applying techniques of knowledge extraction. These techniques are mainly text mining techniques: extraction of lexical patterns, construction of word space and construction of matrix similarity.

The originality of the suggested approach consists in capitalizing knowledge about ontology learning. Thus, during the last iteration, the iterative characteristic of the approach makes it possible to obtain association rules such as: If "*hotel*" near to "*sea*" then it is expensive or 80% of hotels of "*Paris 2*" are classified *hotels 2 stars* and less. Such rules represent knowledge which is able to contribute to the subsequent decision-making.

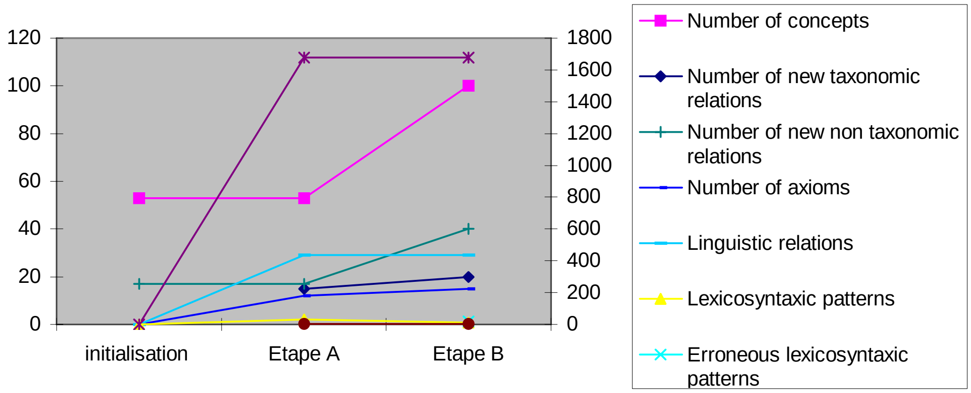


Fig. 18. Results analysis.

6.2 SIRO Comparison with other IR System

In order to evaluate the contribution of the ontological architecture and relations between ontologies, we use the classical precision and recall measures.

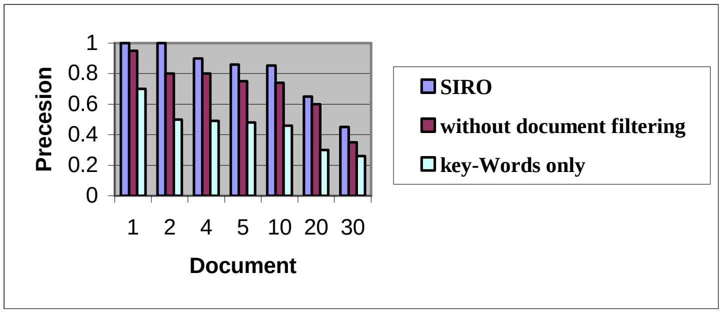


Fig. 19. SIRO comparison with other IR systems.

We compare these measures with two other systems; the first one, *Lucene*, is a traditional information retrieval system based on keywords search. The second one is based on a query reformulation, but without using the vector model. Figure 19 presents the results in terms of precision. The test was realized on 15 queries in the tourism domain and evaluated by 10 users. Our system is better than the two other systems in terms of precision.

7 Conclusion and Future Work

The methodologies for building ontologies described in this paper are fairly complementary. Indeed, methodologies defined to build ontologies from scratch are oriented towards ontological engineering and ontology life-cycle. They are also based on information systems development methodologies. Learning methodologies try to give a response to the time-consuming manual ontology building task, and will provide the Semantic Web more rapidly. Learning techniques can be either numeric or symbolic. They have been exploited to semi-automate certain foundation tasks such as concept hierarchy building, taxonomic relationships extraction, non taxonomic relationships learning, etc. All this research work constitutes a methodological toolbox which can be used to semi-automate ontology construction. We have to take into account previous experiments concerning Metaontology generated for different domains and different data sources. They constitute a knowledge base containing frames, patterns, similarity measures, etc.

Our perspective is to use Semantic Web mining techniques and to restructure Web pages in order to implement an adaptive Web based on the semantic structure, content and services. Our framework presented in this paper is based on ontological components architecture. We first focus on the automation of the domain ontology construction. We will then implement the other ontological components. The service ontology is first designed manually, and then enriched with our tool SIRO when the user does not find relevant services for his query. In this case, the user can manually add services to the ontology. We want to automate the incremental enrichment of the service ontology in parallel with the domain ontology enrichment. The final goal is to build a knowledge Web base on a specific domain. This approach is not domain-dependant.

A case study is described in this paper. From this case study, we can conclude that we must take into account various learning sources (like on-line dictionaries) and structure regularities in Web sites to go further in the implementation.

In the state of art we concluded in [44] that methodologies for building ontologies are fairly complementary. An ontological architecture based on self-learning ontological components is presented. It defines Web content semantics, i.e., the domain ontology; Web structure semantics, i.e., the Web structure ontology and domain Web services semantics, i.e., the Web services ontology. The Metaontology is reusable and can be applied to other domains (from a corpus with a target language). Axioms to extract concepts, relationships and instances are learned incrementally.

OntoCoSemWeb framework has been designed to support the approach described in this paper. An application to the tourism domain was presented which allows a semi-automatic ontology construction by combining three techniques: *lexico-syntactic patterns*; *syntactic frames* and *multidimensional word space*. The proposed approach is based on extracted information weighted by its frequency of appearance in the corpus and updated, from iteration to another. This is useful to revise the information and the associated weightings. These weightings are put according to the technique applied for discovering ontology elements.

The framework is now being enriched to be used on line. In fact, the user can now compose the query which is reformulated using concepts and relations related to the ontology and sent to a search engine. Our future work aims at the construction of a semantic search engine based on the ontological architecture proposed in the present paper. This semantic search engine will have the capacity to enrich its base of ontologies from Web, in a autonomous and on-line way by adopting the present FrameWork. Another perspective concerns the spatial visualization of the search results of this engine according to the distances of similarity between concepts and the representation of the profiles of users according to fuzzy ontologies.

References

1. E. Agirre, O. Ansa, E. Hovy and D. Martinez (2000), *Enriching very large ontologies using the WWW*, in Workshop on Ontology Construction of the European Conference of AI, pp.1-4.
2. T. Berners-Lee, J. Hendler and O. Lassila (2001), The Semantic Web, Scientific American, Vol. 284(R), pp. 34-43.

3 M.A. Aufaure, B. LeGrand, M. Soto, N. Bennacer (2006), *Metadata- and Ontology- Based Semantic Web Mining*, in Web Semantics and Ontology, D. Taniar & J. Wenny Rahayu eds., Idea Group Publishing, Chapter 9, pp. 259-296.

4. T. Gruber (1993), *Toward principles for the design of ontologies used for knowledge sharing*, in International Journal of Human-Computer Studies, 43(5/6), pp. 907-928.

5. R. Sekiuchi, C. Aoki, M. Kurematsu and T. Yamaguchi (1998), *DODDLE: A Domain Ontology Rapid Development Environment* -5th Pacific Rim International Conference on Artificial Singapore (PRICAI98), pp. 194-204.
6. F. Fürst and F. Trichet (2005), *Axiom-based ontology matching: a method and an experiment* -3rd international conference on Knowledge capture, pp. 195-196.
7. Gomez-Perez, A. and Fernandez-Lopez (2003), *Ontological Engineering – advanced information and knowledge processing,* Springer, pp. 107-196.
8. M. Uschold and M. King (1995), *Towards a Methodology for Building Ontologies*, in Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI 1995), pp. 1-13.
9. M. Fernandez, A. Gòmez-Pérez, A. Pazos-Sierra and J.Pazos-Sierra (1999), *Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment*, in IEEE Intelligent Systems & their applications, 14(1), pp. 37-46.
10. S. Staab, H.P. Schnurr, R. Studer and Y. Sure (2001), *Knowledge Processes and Ontologies*, IEEE Intelligent Systems, Vol. 16(1), pp. 26-34.
11. A. Gòmez-Pérez and M.D. Rojas (1999), *Ontological Reengineering and Reuse*, in European Workshop on Knowledge Acquisition, Modeling and Management (EKAW 1999), Lecture Notes in Artificial Intelligence LNAI 1621, Springer-Verlag, pp. 139-156.
12. J. Euzenat (1995), *Building consensual knowledge bases, context and architecture* -2nd international conference on building and sharing very large-scale knowledge bases (KBKS 1995), IOS press, pp. 143-155.
13. A. Maedche and S. Staab (2000), *Semi-automatic engineering of ontologies from text* -12th International Conference on Software Engineering and Knowledge Engineering (SEKE 2000), pp.231-239.
14. J. Jannink and W. Gio (1999), *Thesaurus Entry Extraction from an On-line Dictionary,* in Fusion 1999, Sunnyvale CA, pp. 110-138.
15. A. Deitel, C. Faron and R. Dieng (2001), *Learning ontologies from RDF annotations*, in IJCAI 2001 Workshop on Ontology Learning (OL 2001).
16. D. Sanchez and A. Moreno (2004), *Automatic generation of taxonomies from the WWW* -5th International Conference on Practical Aspects of Knowledge Management (PAKM 2004). LNAI, Vol. 3336, pp. 208-219.
17. D.L. Rubin, M. Hewett, D.E. Oliver, T.E. Klein and R.B. Altman (2002), *Automatic data acquisition into ontologies from pharmacogenetics relational data sources using declarative object definitions and XML*, in the Pacific Symposium on Biology, pp.88-99.
18. H. Suryanto and P. Compton (2001), *Discovery of Ontologies from Knowledge Base*s, in First International Conference on Knowledge Capture, The Association for Computing Machinery, pp171-178.
19. N. Aussenac-Gilles, B. Biébow and S. Szulman (2002), *Revisiting Ontology Design: A Methodology Based on Corpus Analysis* -12th International Conference in Knowledge Engineering and Knowledge Management (EKAW 2000), Springer-Verlag, pp. 172-188.
20. D. Faure and C. Nedellec (1998), *A corpus-based conceptual clustering method for verb frames and ontology acquisition*, in workshop of the 1st International Conference on Language resources and Evaluation (LREC), pp. 1-8.
21. N. Sugiura, K. Masaki, F. Naoki, I. Noriaki and Y. Takahira, (2003), *A Domain Ontology Engineering Tool with General Ontologies and Text Corpus* -2nd Workshop on Evaluation of Ontology based Tools, pp.71-82.
22. J. Nobécourt (2000), *A method to build formal ontologies from text* -12th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000), pp. 21-27.
23. M.A. Hearst (1998), *Automated Discovery of WordNet Relations*. “Wordnet An Electronic Lexical Database”, C Fellbaum ed. MIT Press, Cambridge, MA, pp. 132-152.
24. D. I Moldovan and R. Girju (2000), *Domain-Specic Knowledge Acquisition and Classification using WordNet* -13th international FLAIRS 2000 Conference, pp. 224-228.
25. G.A. Miller (1995), *WORDNET: A Lexical Database for English*, Communications of ACM, pp. 39-41.

26. L. Khan and F. Luo (2002), *Ontology Construction for Information Selection* -14th IEEE International Conference on Tools with Artificial Intelligence, pp. 122-127.
27. M. Hearst and H. Schütze (1993), *Customizing a Lexicon to Better Suit a Computational Task*, in Workshop on Extracting Lexical Knowledge, pp. 77-96.
28. A. Faatz and R. Steinmetz (2002), *Ontology enrichment with texts from the WWW,* in the First International Workshop on Semantic Web Mining, European Conference on Machine Learning (ECML/PKDD 2002), pp. 20-34.
29. R Navigli and P.Velardi (2004), *Learning Domain Ontologies from Document Warehouses and* Dedicated Web Sites, Computational Linguistics, MIT press, pp. 151-179.
30. H. Davulcu, S. Vadrevu and S. Nagarajan (2003), *OntoMiner: Bootstrapping and Populating Ontologies from Domain Specific Websites,* In IEEE Intelligent Systems, Vol. 18, pp. 24-33.
31. L. Karoui, MA. Aufaure and N. Bennacer (2006), *Context-based Hierachical Clustering for the Ontology Learning*, in International Conference on Web Intelligence, pp.420-427.
32. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke (2002), *OntoEdit: Collaborative ontology development for the Semantic Web,* In 1st International Semantic Web Conference (ISWC 2002), Vol. 2342, Springer, pp. 221-235.
33. A. Farquhar, R. Fikes, and J. Rice (1997), *The Ontolingua Server: a Tool for Collaborative OntologyConstruction*, in International Journal of Human-Computer Studies 46, pp. 707-727.
34. H. Knublauch, RW. Fergerson, NF. Noy, and MA. Musen (2004), *The Protege OWL Plugin: An open development environment for semantic Web applications* -3rd International Semantic Web Conference (ISWC 2004), pp. 229-243.
35. J. Domingue (1998), *Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web* -11th Workshop on Knowledge Acquisition for Knowledge-Based Systems (KAW 1998).
36. J. Arpirez, O. Corcho, M. Fernandez-Lopez and A. Gomez-Perez (2001), *WebODE: a Workbench for Ontological Engineering*, in International Conference on Knowledge Capture, pp. 6-13.
37. S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens (2001), *OilEd: a reason-able ontology editor for the semantic web,* In Austrian Conference on Artificial Intelligence (KI 2001), Springer-Verlag, pp. 396-408.
38. B. Biebow and S. Szulman (1999), *TERMINAE: A linguistics-based tool for the building of a domain ontology* -11th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW 1999), Springer, pp. 49-66.
39. M. Junker, M. Sintek and M. Rinck (1999), *Learning for Text Categorization and Information Extraction with ILP*, in 1st Workshop on Learning Language in Logic, pp. 84-93.
40. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery (2000), *Learning to construct knowledge bases from the World Wide Web*, Artificial Intelligence, pp. 69-113.
41. A. Gangemi, C. Catenacci, C. Ciaramita and J. Lehmann (2005), *A theoretical framework for ontology evaluation and validation* -2nd Italian Semantic Web Workshop in Semantic Web Applications and Perspectives (SWAP 2005), pp. 9-26.
42. B. Berendt, A. Hotho and G. Stumme (2002), *Towards semantic Web mining*, in International Semantic Web Conference, Vol. 2342 of Lecture Notes in Computer Science, Springer, pp. 264-278.
43. MA. Aufaure, R. Soussi, H. Baazaoui Zghal and H. Ben Ghezala (2007), *SIRO: On- Line Semantic Information Retrieval using Ontologies*, in Second International Conference on Digital Information Management (ICDIM 07), pp. 28-31.
44. N. Ben Mustapha, MA. Aufaure and H. Baazaoui Zghal (2006), *Towards an Architecture of Ontological Components for the Semantic Web*, in Web Information Systems Modeling Workhop (WISM) -18th Conference on Advanced Information Systems Engineering (CAISE 2006), pp. 22-35.