Semantic interoperability of heterogeneous semantic resources

Catarina Ferreira Da Silva, Lionel Médini, Samer Abdul Ghafour, Patrick Hoffmann, Parisa Ghodous [1]

Lyon Research Center for Images and Intelligent Information Systems
Claude Bernard Lyon 1 University,
Villeurbanne, France

Abstract

This paper presents a three-step approach for interoperabilising heterogeneous semantic resources. Firstly, we construct homogeneous representations of these resources in a pivot format, namely OWL DL, with respect to the semantics expressed by the original representation languages. Secondly, mappings are established between concepts of these homogenised resources and stored in a so-called articulation ontology. Thirdly, an approach for ranking those mappings is suggested in order to best fit users' needs. This approach is currently being implemented in the Semantic Resources Interoperabilisation and Linking System (SRILS). The mapping results as well as the work to be done are discussed.

Key words: heterogeneous semantic resources, ontologies, interoperability, homogenisation, mapping, ranking, description logics, subsumption algorithms, OWL.

1 Introduction

The growing use of the Web for collaborative and distributed work, associated to the standardisation of the languages and tools related to the Semantic Web, have led to the availability of multiple terminological and syntactical resources in numerous professional domains using the Internet. Different types of resources such as taxonomies, vocabularies, thesauri or ontologies, have been elaborated according to different existing standards. They treat different topics and approach them from different viewpoints, techniques and objectives. The existence, availability and complementariness of these resources on the Web initiated new usages which could benefit from their simultaneous use,

[1] {cferreir,lmedini, sabdulgh, phoffman, ghodous}@liris.cnrs.fr

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

for as various applications as electronic commerce, information retrieval or collaborative design.

The simultaneous use of those resources requires them to be interoperable. Interoperability – i.e. "the ability to exchange information and use the information which has been exchanged" [2] – can be treated at several levels such as technical and semantic. Basically, achieving technical interoperability is a matter of enabling distributed applications to work, taking into account syntactical and structural heterogeneity issues between different resources. However, this can cause serious "misinterpretations" of data and lead to misunderstandings between users, calculation errors, or even system failure, depending on the regarded application. At an upper level, semantic interoperability consists in preventing these problems from happening, while taking into account the semantics associated to the data, and ensuring exchanged information share the same meaning. To achieve both syntactical and semantic interoperability, we herein suggest an "interoperabilisation" approach for heterogeneous semantic resources (SR), based on three steps. Firstly, we homogenise the SR representation formats, considering the expressiveness of the source and target languages to preserve the meanings of the resources for the further steps. Secondly we align previously homogenised resources by mapping SR entities. Thirdly, to rank by relevance the mappings obtained in the previous stage, we suggest a personalized and contextualized measure of these mappings.

This paper first presents a state of the art of different existing interoperability approaches between heterogeneous SR. Then, our approach is presented and its different stages are detailed. The architecture and implementation of the SRILS system, which partially implements this approach, are described. An application example taken from the construction field is presented. Finally, the mapping results, as well as the work to be done are discussed.

2 Existing interoperability approaches between semantic resources

Heterogeneity of SR can be considered at different levels, such as representation format level (syntax), data organization level (structure) and different points of view level (semantics). Some authors have treated these levels separately, even if they are not easy to split. Chaudri et al [9] search solutions for resolving the semantic ambiguity at syntax level. Mitra et al [22] describe several types of structural semantics ambiguity. Bowers et al [6] show that using different syntax or structure formalisations is the source of different errors when transforming a representation into another. In the rest of this section, we summarise background work regarding homogenisation of representation formats and alignment of SR.

[2] European Union Software Directive, see Council Directive 91/250/EEC

2.1 Homogenisation of representation formats

In order to make heterogeneous SR interoperable, Kalfoglou et al [17] consider that homogenisation by translation of resources in a common representation language is often a necessary preliminary stage. The choice of the target language is essential because it should be expressive enough in order to represent explicit and precise knowledge. On the other hand, it is important to compromise between expressiveness and complexity. Jannink et al [16] propose an algebra based on mapping rules in order to achieve that normalization. The OntoMorph system [8] enables to specify syntactic transformations using syntax rewriting mechanisms, rules based on pattern matching and models describing how to apply those syntactic rules. Normalisation treats the syntactic heterogeneity without loosing the expressiveness of the representation languages.

Databases and SR are both knowledge representations. However, databases use fewer modelling primitives. Databases researchers face the problem of finding correspondences between different schemas, just as with SR. In relational databases a large number of embedded SR were developed since the 80's. The logic model of databases allows to organise vast data and formally express relationships among data. The semantics of these resources are implied from the tables structure and integrity constraints. However, a semantically well-defined format representing all integrity constraints in the context of an automatic transformation does not exist. Consequently, a conversion from a database to an OWL ontology is still an *ad hoc* transformation, dependent of each database schema [4]. Calvanese et al [7] suggested an approach to transform a database into a set of description logics (DL) axioms. This work is a preliminary stage for developing a generic method to convert a relational database into an OWL ontology.

2.2 Alignment of semantic resources

An alignment can be defined [5] as a set of mappings expressing a correspondence[3] between two entities of different ontologies. Schema matching [25] as well as SR alignment require finding related entities, (i.e. concepts and properties), in two or more knowledge structures [23]. For this, several methods have been used, among which terminological, structural, extensional (i.e. based on instances) or semantic methods. Those methods come from different disciplines such as data analysis, machine learning, language engineering, statistics or knowledge engineering. Their applicability both depends on the type of SR features (e.g. labels, structures, instances, semantics) to be compared and on the expected types of results. Techniques to find inter-ontology relations lean most frequently on instances and schema matching, concept matching and matching of structural elements of the source ontologies [13].

[3] A correspondence is constituted of a relation and a trust assessment.

Kalfoglou and Schorlemmer [17] survey a set of frameworks, methods and tools related to ontology mapping, such as the ones shown in the next paragraphs.

In the machine-learning discipline, GLUE [11] searches mappings over instances of classes. Given two ontologies, for each concept in one ontology, GLUE finds the most similar concept in the other ontology using probabilistic definitions of several practical similarity measures. It exploits information about the concept instances – such as the words frequency in the text value of instances, the instance names, the value formats, or the characteristics of value distributions – and the taxonomic structure of the ontologies. Nevertheless, machine-learning techniques work better when large sets of instances are available.

The OBSERVER [21] system helps users define mappings between concepts of two ontologies by finding pairs of related concepts. It uses the data structures underlying the domain-specific ontologies and the synonymy, hyponymy and hyperonymy relations to detect linguistic matches between concepts. Once the mappings are defined, users ask DL-formatted queries about terms of one of the ontologies, and the system expands the query to the terms of the other ontology. For this, users have to be familiar with DL constructors.

Euzenat [12] proposes an alignment API that supplies OWL-compliant functions for helping programmers automatically map ontologies. This API currently uses term-matching techniques that cannot guarantee valid alignments. For instance, matched terms can be homonyms (and have different meanings) or be semantically close without being complete synonyms.

2.3 Semantic methods for ontologies alignement

Notwithstanding these efforts, we think that these approaches could benefit from techniques that take into account the semantics associated to concepts definitions. DL-based techniques [2] are appropriate for that, since they rely on the explicit and formal semantics represented by ontologies. When used for comparing ontologies, they ensure the original semantics of the SR entities is preserved and provide an explicit and formal interpretations of both entities being compared and the produced relations. We present in this section a basic DL algorithm and a tool currently used in these methods. More details about the method we used in this work are provided in section 3.2.

Standard DL techniques apply subsumption algorithms to establish relationships between concepts. The *tableau* algorithms are a class of subsumption algorithms that firstly expand each ontology: each occurrence of a name of concept on the right side of a definition is replaced by the concept it stands for. This recursive process of dependency-eliminating substitutions (known as *unfolding*) is done until no cycle in the set of definitions exist. Thus an *unfoldable* ontology implies that all axioms are unique and acyclic definitions. Even if the expanded ontology size can increase exponentially compared to its original size, the unfolding process enables to reduce the reasoning process to the computation of subsumption and satisfiability[4].

A *tableau* is a graph which represents a model, with nodes corresponding to individuals (elements of the domain of interpretation $\triangle^{\mathcal{I}}$). Tableau algorithms try to prove the satisfiability of a concept D by constructing an interpretation model $\mathcal{I}$ in which $D^{\mathcal{I}}$ is not empty (i.e. constructing a *tableau* starting with a single individual and then inferring the existence of additional individuals or additional constraints on individuals). This kind of reasoning uses a refutation-style proof [15]: C is subsumed by D if it can be shown that the existence of an individual x that is an instance of C ($x \in C^{\mathcal{I}}$) but is not an instance of D ($x \notin D^{\mathcal{I}}$) is logically inconsistent. This corresponds to testing the logical (un)satisfiability of the concept $C \sqcap \neg D$ (i.e. $C \sqsubseteq D \iff C \sqcap \neg D$ is not satisfiable). The inference mechanism consists in applying a set of consistency-preserving transformation rules for $\mathcal{ALC}$, known as $\sqcap$-rule, $\sqcup$-rule, $\exists$-rule and $\forall$-rule until no more rules apply; see [15] for details. The algorithm terminates when the graph is complete (no further inference is possible) or when contradictions have been revealed. A concept is unsatisfiable if the graph obtained this way contains a contradiction (called a "clash") and satisfiable (the graph is consistent) otherwise.

The FaCT[5] inference engine [14] applies DL techniques and allows reasoning over concepts, roles and attributes descriptions, as well as managing concepts hierarchies based on the subsumption relation. FaCT uses a very expressive DL language $\mathcal{SHIQ} = \{\mathcal{ALC} + \text{number restriction} + \text{role hierarchy} + \text{inverse role} + \text{transitive role}\}$[6] able to capture a large part of the semantics of knowledge models. This language is equipped with effective reasoning techniques that are sound and complete with respect to the semantics.

The ONDIL system [18] reuses the inference mechanisms of FaCT, and also supports the management of several Tboxes[7].

ONDIL includes three modules, namely *ontology management*, *mediator*, and *inference engine*. The latter uses two kinds of subsumption algorithms: *tableau-based algorithms* for standard inferences and *structural algorithms* for non-standard inferences [28]. The inference engine uses pairs of ontologies to deduce new knowledge that essentially consists in relations between ontological concepts.

[4] A concept C is *satisfiable* with respect to a Tbox, if it exists an interpretation model $D^{\mathcal{I}}$ of the Tbox such that $C^{\mathcal{I}}$ is nonempty

[5] Fast Classification of Terminologies (FaCT) has been developed to assess the feasibility of using optimal subsumption algorithms based on *tableau technique* [2].

[6] $\mathcal{AL}$ description logic language, ($\mathcal{AL}$ stands for *Attribute Language*), has been introduced by Schmidt-Schaub et al [28]. The other languages of this family are extensions of AL. The $\mathcal{ALC}$ language allows to express the negation of concepts [2].

[7] A Tbox or Terminology is a finite set of definitions, if for every atomic concept C there is at most one axiom in the Tbox whose left side of the definition is C.

3 Interoperabilisation approach

This stage consists in converting SR to a common format and is prior to any other processing stage. The next stage is to identify correspondences enabling to align the SR. For this, we suggest a hybrid approach combining mapping and contextualisation of relations between entities (concepts and roles).

3.1 Homogenisation

Homogenisation of the representation formats enables to solve syntactical issues, as well as part of structural heterogeneities and issues that come from different expressivenesses in the encoding languages. The purpose of this step is to homogenise dissimilar formats of SR representations, while maintaining their semantics. We mainly deal with three types of SR: taxonomies, non-tree based graphs, and ontologies. The conversion procedure is based on an explicit distinction between different levels of knowledge representation, namely meta-model, model, and data. A meta-model specifies the structure of a knowledge representation language and clarifies its expressiveness. The model level represents the data structure for each application. This level contains classes, attributes attached to classes, and possible relations between various classes. The instance level gathers the data.

In order not to develop specific tools for each of the different representation formats as well as to express the mapping rules in an appropriate one, we chose to use a common representation language, sufficiently expressive to represent all types of given SR. We chose OWL[8], and more precisely its sub-language based on Description Logics, OWL DL, for two main reasons. Firstly, it is part of the W3C recommendations related to the Semantic Web. It takes advantages of previous work on XML syntax, RDF and RDF Schema (RDFS) formal semantics. Secondly, OWL DL allows maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will end in finite time) [10]. An OWL (Lite or DL) ontology corresponds to a DL Tbox together with a role hierarchy, describing the domain in terms of *classes* – corresponding to concepts – and *properties* – corresponding to roles [3].

The homogenisation process aims at converting different SR into OWL while maintaining their intrinsic semantics. It consists in (1) elaborating a meta-model representing the constructions and expressiveness of the pivot language[9]; (2) designing meta-models of the available SR representation languages[10], defined as restrictions of the OWL DL meta-model; (3) converting the SR contents into OWL DL. The latter step is described in the following paragraphs for the three general SR categories considered herein.

[8] Ontology Web Language : http://www.w3.org/2004/OWL/

[9] The OWL meta-model is represented in two UML classes schemas: a view of the classes and one of the properties. It is presented in [1]

[10] Schemas of these meta-models are as well available in [1]

Taxonomies (**XML**): (1) Concepts of a taxonomy are considered as OWL classes (syntax: *owl:Class*). (2) The attributes attached to these concepts are treated as properties in the OWL ontology (*owl:Property*). The values of the attributes can either be literals (represented with *rdfs:Literal*) or resources (represented as OWL classes). The latter is tied to the previously defined property by another particular property (*rdfs:range*). (3) The subsumption relationship between concepts is represented in OWL using the *rdfs:subClassOf* constructor.

Non-tree based graphs (**RDF**): Graphs represent knowledge by linking concepts with complex relations. RDF is a graph-based language based on statements (Subject, Predicate, Object). The conversion of a graph into OWL is achieved as follows: (1) the subject is represented in OWL by defining it as a class. (2) An object can either be represented as a class or a literal element, depending on its type. (3) A predicate is represented as an object property (*owl:ObjectProperty*) if the related object is a resource, or as data type property (*owl:DatatypeProperty*) when its related object is a literal. *rdfs:domain* and *rdfs:range* are used to specify a property. The former specifies the subject, and the latter defines the object of the statement.

Ontologies (**RDFS and DAML + OIL**): Ontologies extend graph based formalisms by providing high level constructions that make the semantics of the graphs explicit (for instance, *owl:disjointWith* specifies two disjoint classes). As OWL is an extension of RDFS, it inherits characteristics from RDFS, and any document which is valid according to an RDFS schema is also valid as an OWL Full ontology. However, it could not be seen as an OWL DL document. Consequently, converting an RDFS document to OWL DL requires to distinguish the differences between OWL Full and OWL DL [24]. As OWL, DAML+OIL is an ontology language built on RDF and RDFS and it is based on Description Logics, compared to the three sublanguages of OWL, DAML+OIL semantics is the closest to the OWL DL semantics.

3.2 Alignment

This section explains how we apply a semantic method based on DL techniques to discover mappings between concepts of the different homogenised ontologies. For this, we use the ONDIL system that can process several ontologies at a time, as well as axioms[11] between their respective concepts.

Firstly, in order to ensure that ontologies are consistent, they are separately unfolded. This is done by using the ONDIL standard inference services based on a *tableau* algorithm and on the $\mathcal{ALC}$ language. The mapping search process takes as inputs expanded concepts definitions.

Let o and o' be a pair of ontologies and $A(x)$ a set of axioms given as inputs. The ontology reasoning services of ONDIL use a tableau algorithm to

[11] Axioms are previously defined relationships between entities of the two ontologies, that the inference engine can use during the satisfiability test step.

identify subsumption relationships between concepts of o and o', as shown in the following generic example. Let $C_1 := \forall r.A \sqcap \exists r.B$ be a concept of o, and $C_2 := \exists r.B$ be a concept of o'. We are now going to test if $C_1 \sqsubseteq C_2 \iff C_1 \sqcap \neg C_2 \sqsubseteq \bot$.

$$\begin{array}{rl} C_1 \sqcap \neg C_2 \equiv & \forall r.A \sqcap \exists r.B \sqcap \neg\exists r.B \\ \text{applying the De Morgan's law} & \neg\exists r.B \iff (\forall r.\neg B) \\ C_1 \sqcap \neg C_2 \equiv & \forall r.A \sqcap \exists r.B \sqcap \forall r.\neg B \\ C_1 \sqcap \neg C_2 \equiv & \forall r.(A \sqcap \neg B) \sqcap \exists r.B \end{array}$$

ONDIL was modified to accept as inputs two ontologies and the set of axioms. These inputs constitute the knowledge processed by the inference engine module of ONDIL to construct the graph of the above definition. This is done by applying transformation rules as follows.

Let the graph be a direct graph in which each node x is labelled with a set of concepts ($\mathcal{L}(x) = \{C, \ldots, C^n\}$) and each edge (x, y) is labelled with a role ($\mathcal{L}(x, y) = r$). When a concept C is in the label of a node x ($C \in \mathcal{L}(x)$), it represents a model in which the individual corresponding to x is in the interpretation of C. When an edge (x, y) is labelled r, it represents a model in which the tuple corresponding with (x, y) is in the interpretation of r.

From the $\sqcap$-rule we add to the graph the instances of the concepts A and B that compose the definition $\forall r.(A \sqcap \neg B)$, i.e. $A(x)$ and $\neg B(x)$. From the $\exists$-rule we add the following instances to the graph: $B(x)$, $r(x, y)$. We do not need to further apply the rules because a clash is found: $\{B(x), \neg B(x)\} \subseteq \mathcal{L}(x)$ results in a contradiction. Thus $C_1 \sqcap \neg C_2 \sqsubseteq \bot$ (this means that $C_1 \sqcap \neg C_2$ is not satisfiable) and $C_1 \sqsubseteq C_2$. So a subsumption relation was detected between the concepts C_1 and C_2.

Retrieved correspondences can be equivalences and non-equivalences. When and the definitions of and are equivalent. Equivalences enable to state that the interpretation of two concepts from two different SR is 100% equal. We name non-equivalences "semantic proximities". These refer to mappings in which only a part of the concepts of the SR is common. This is the case of subsumption and conjunction. Conjunction mappings are consequences of the subsumption ones. Therefore, from input ontologies o and o', the axioms $C \sqsubseteq C'$ and $C \sqsubseteq C_1$ with $C, C_1 \in o$ and $C' \in o'$ allow to state: $(C \sqcap C') \sqsubseteq (C_1 \sqcap C')$.

3.3 Ranking

Identified mappings are delivered to users through client applications. Users are specialised in specific activities, and their requests may concern only a limited field of knowledge and particular tasks: their different contexts of work involve different intentions and needs [27]. A measure that would consistently interpret the mappings according to the context of work should improve the reliability of mappings ranking relevance. Though this work is not yet implemented, we present it as it is part of our approach.

In order to take context into account when comparing mappings, we need reliable information about the users' works and environments as well as information about why SR have been developed, and what fields they cover. We intend to take advantage of a context modelling for representing domains, tasks, etc. It will serve as reference for situating all considered SR as well as users' profiles.

We define a fragment of a SR as a concept of this resource and all the concepts of this resource it subsumes. Each fragment is associated with zero, one or more domains, and is allotted as many "contextual vectors" (CV): these are sets of normalized weights depending on the relevance of the SR fragment content for the domain-specific criteria and tasks.

Similarly, we associate to a user as many "user domain vectors" (UDV) as there are domains she/he is interested in. These vectors she/he instanciates once depending on the significance she/he assigns to each criterion, and on the importance of each task in her/his activity.

Let a user submit a query on a concept C to retrieve all the concepts semantically related. Each concept is included in a SR fragment and is being attached its CV. We compare the CV of C with every other CV by applying a specific measure for each criterion or task that appears in both CV, and storing the result in a "fragment comparison vector" (FCV).

Then we interpret these FCV according to the user's domains: we ponder them by calculating all concepts "User Domain Interpretations" (UDI) constituted of the UDV-pondered FCV and of a computed relevance of the concept for the corresponding domain. Each concept UDI is then ranked depending on these valuations. Concepts are sorted depending on each relevant interpretation of C. Outputs are the sorted list of these relevant interpretations with the corresponding concept rankings, and the remaining concepts for which no accepted interpretation held.

4 The SRILS system

The development of SRILS has been motivated by an industrial need expressed by the Centre Scientifique et Technique du Bâtiment (CSTB) located in Sophia-Antipolis (France). We have used three different SR from the building and construction (B&C) domain. bcXML is a multilingual taxonomy of concepts, products and services, developed in the eConstruct project [20]. This resource holds 3000 terms, in 6 different european languages. The Edibatec[12] dictionary covers several B&C domains, such as electrical or ventilation equipments. The e-Cognos ontology [19], developed at CSTB, contains 17 000 concepts and relations and covers several parts of the B&C domain.

[12] EDIBATEC dictionary, see http://www.edibatec.org/Accueil/Default4.htm

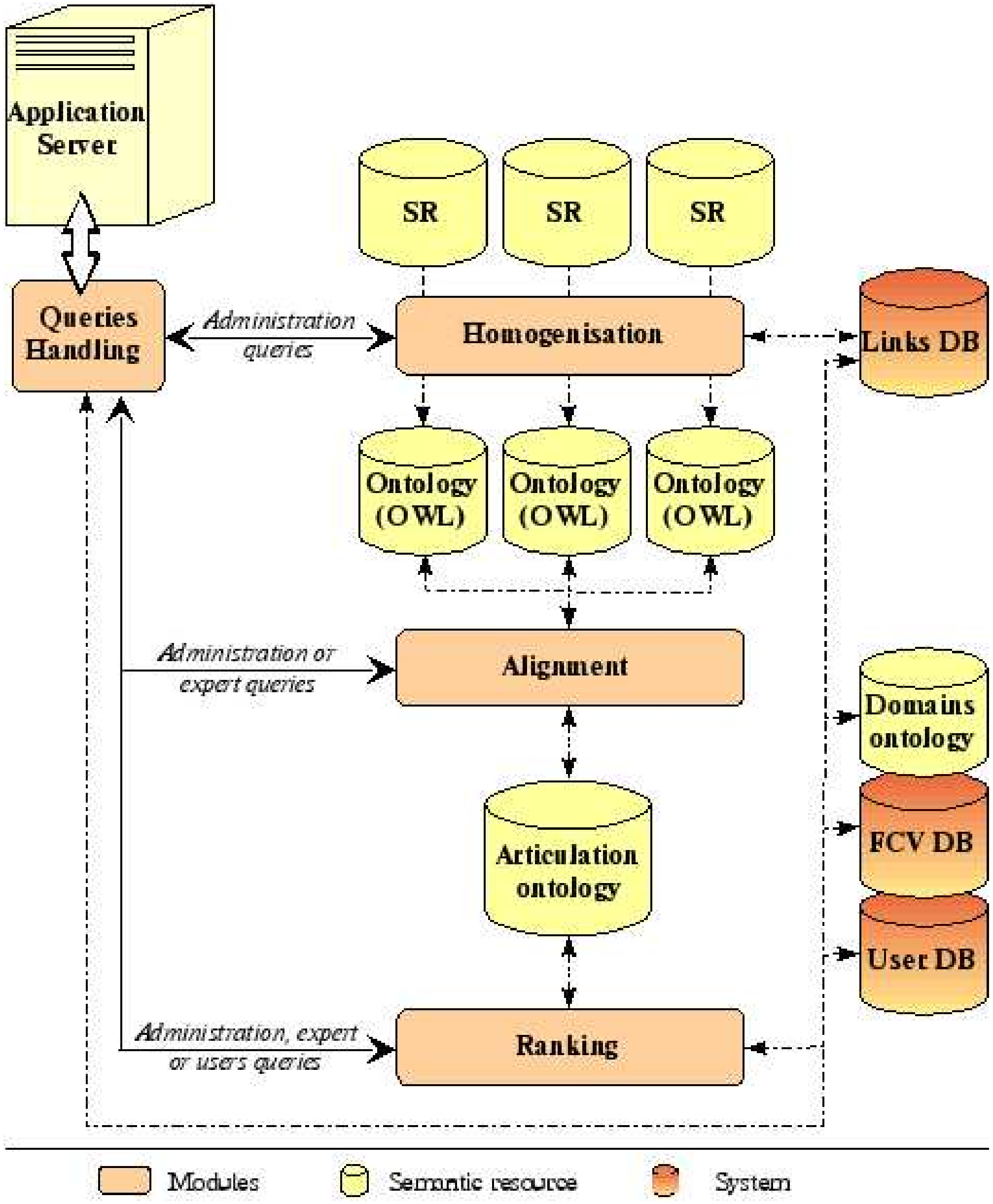

Fig. 1. Architecture of the SRILS system

CSTB has also developed an ontology server, named e-COSer[13] [19], that processes queries regarding concepts and relations of different SR and supplies high-level services to different users. In the context of the SRILS system, we consider e-COSer as the client application.

SRILS relies on four modules and several types of resources (see Fig. 1). The external interface of the system is provided by the queries processing module and targets the integration within a services-oriented architecture. The conversion module is in charge of homogenisation. The alignment module performs mapping search. The contextual ranking module, still in develop-

[13] eCOSer – eCognos Ontology Server, http://195.83.41.67/eCOSer/Login.jsp

ment, ranks the mappings by relevance. The modular architecture of SRILS enables to emulate non-developed modules in order to supply the expected services to the upper layer. The different kinds of SR used in SRILS are the ones containing data to be "interoperabilised" (original and converted SR), an "articulation ontology", where mappings are stored and the specific resources needed for the contextual ranking stage. Detailed descriptions of the different modules are not in the scope of this paper. The next section presents an example of mappings search in the B&C SR presented above.

5 Alignment tests using the mapping search module

This section presents the first tests of the mapping search module, using the inference services of ONDIL (Sect. 3.2). The search is performed between two ontologies at a time. Retrieved correspondences can be equivalence relations or semantic proximities (mainly subsumption and conjunction, but also transitivity, which is implied by the subsumption relation). As inference is a time-consuming task that can take several minutes when we deal with large ontologies, mappings search is carried out *a priori* in order to optimise processing time. After mappings validation by domain experts, the mappings constitute the articulation ontology, which is queried by the queries handling module, and will not change, unless SR are modified and mappings search reprocessed.

In order to first prove the correctness of the mapping method, we mapped each SR with itself. Obviously, mapping a SR with itself produces equivalences between the same concepts and only that kind of equivalences. In addition, results for subsumption and conjunction are also obtained, but giving only redundant information, since if A is equivalent to B, $A \sqsubseteq B$ and $B \sqsubseteq A$.

A typical usage scenario is the following: a user submits a product-centred query about a concept of the bcXML taxonomy, and wants to retrieve documents related to that product. The articulation ontology is queried by the system, since it stores retrieved and validated mappings between the bcBuildingDefiniton taxonomy (where the products are really defined) and the e-Cognos ontology (where the concepts that represent the products and that are used to index the documents are defined). We present hereafter a fragment of the articulation ontology showing three examples of subsumption mappings retrieved by ONDIL between eCognos and bcXML SR.

```
<owl:SRILS-ArticulationOntology rdf:about="">
<rdfs:label> mappings between eCognos and bcXML<rdfs:label>
<owl:imports rdf:resource="http://195.83.41.67/eCognos"/>
<owl:imports rdf:resource=
         "http://195.83.41.68:8080/bcBuildingDefiniton"/>
</owl:SRILS-ArticulationOntology>
<owl:Class rdf:about="&bcBuildingDefiniton;Gas">
 <rdfs:subClassOf
```

```
  rdf:resource="&eCognos;energy_sources_and_distribution"/>
</owl:Class>
<owl:Class rdf:about="&eCognos;Highrise_building">
 <rdfs:subClassOf
  rdf:resource="&bcBuildingDefiniton;Building"/>
</owl:Class>
<owl:Class rdf:about="&bcBuildingDefiniton;Wall">
 <rdfs:subClassOf
  rdf:resource="&eCognos;Wall_and_Cladding"/>
</owl:Class>
```

Subsumption mappings are more numerous than equivalences. It is worth noticing that subsumption mappings can depend on the ontology order of mapping calculation. This means that the subsumption mappings of (O_1, O_2) may be different from the subsumption mappings of (O_2, O_1). In other words, this difference comes from the asymmetry of the subsumption relationship between two concepts. More precisely, a subsumption mapping $C_1 \sqsubseteq C_2$ (where $C_1 \in O_1$ and $C_2 \in O_2$) belongs to the set of mapping of (O_1, O_2) while the set of mapping of (O_2, O_1) may not contain the subsumption mapping $C_2 \sqsupseteq C_1$. However, equivalence mappings are preserved.

6 Conclusion

This paper presents an approach to facilitate interoperability between heterogeneous SR, based on three heterogeneity levels: syntactic, structural and semantic. We apply different “interoperabilisation” approaches to tackle these heterogeneity levels. This approach is partially implemented in the SRILS middleware system: the two former levels are automatically processed. This system is likely to convert taxonomies, graphs and ontologies into OWL DL format, keeping the semantic and expressive power of the original encoding languages. Semantic “interoperabilisation” of SR is done by retrieving mappings between entities of the produced ontologies, using an inference engine and description Logics-based techniques.

We briefly present an approach of contextualisation of the retrieved mappings in order to establish a ranking of the mappings relevant to the users. This last stage is being implemented. The use of SRILS is showed by an application in the building and construction domain. We also consider testing other methods for discovering semantic alignments, by using linguistic corpus to help find new mappings. Regarding measuring semantic proximity, we consider using fuzzy logic and probabilistic methods.

Acknowledgement

The authors wish to acknowledge that part of the work presented here is largely due to an intense teamwork, carried out in the *eContent* European FUNSIEC project, that is a Feasibility study for a UNified Semantic Infrastructure in the European Construction sector, see www.funsiec.org. Special thanks to Celson Lima and Chan Le Duc at CSTB.

References

[1] Abdul Ghafour, S., "Méthodes et outils pour l'intégration des ontologies," Training report of Master Documents multimédia, Images et Systèmes d'Informations Communicants, LIRIS Laboratory, Lyon, France, 2004.

[2] Baader, F., A. Borgida, R. J. Brachman *et al*, "The Description Logic Handbook, Theory, Implementation and Applications," Baader, F., Calvanese D., McGuinness D., Nardi D. and Patel-Schneider P. (Eds), Cambridge University Press, 2003.

[3] Baader, F., I. Horrocks and U. Sattler, Description Logics, "Handbook on Ontologies," Staab S., Studer S. (Eds), Springer-Verlag Berlin Heidelberg, ISBN 3-540-40834-7, 2004, 3–28.

[4] Barrasa, J., O. Corcho and A. Gómez-Pérez, Fund Finder: A case study of database-to-ontology mapping, ISWC2003 Workshop on Semantic Integration, Sanibel Island, Florida, 2003, 6pp.

[5] Bouquet, P., J. Euzenat, F. Franconi, L. Serafini, G. Stamou and S. Tessaris, "D2.2.1 Specification of a common framework for characterizing alignment – Knowledge Web project," realizing the semantic web, IST-2004-507482 Programme of the Commission of the European Communities, 2004.

[6] Bowers, S., and L. Delcambre, Representing and transforming model-based information, First Workshop on the Semantic Web at the Fourth European Conference on Digital Libraries, Lisbon, Portugal, 2000, 18–20.

[7] Calvanese, D., G. De Giacomo and M. Lenzerini, A framework for ontology integration, Proceedings of the 1st Internationally Semantic Web Working Symposium (SWWS), 2001, 303–317.

[8] Chalupsky, H., OntoMorph – A Translation System for Symbolic Knowledge, Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning, Breckenridge, Colorado, USA, 2000, 471–482.

[9] Chaudhri, V. K., A. Farquhar, R. Fikes, P. D. Karp and J. P. Rice, OKBC – A programmatic foundation for knowledge base interoperability, Proceedings of the 15th National Conference on Artificial Intelligence and of the 10th Conference on Innovative Applications of Artificial Intelligence, AAAI Press, 1998, 600–607.

[10] Dean, M., and G. Schreiber (Eds), "OWL Web Ontology Language Reference, W3C Recommendation," 2004. URL: http://www.w3.org/TR/owl-ref/#appD.

[11] Doan, A., J. Madhavan, P. Domingos and A. Halevy, Learning to map between ontologies on the semantic web, Proceedings of the 11th International World Wide Web Conference (WWW'02), ACM-Press, 2002, 662–673.

[12] Euzenat, J., *An API for ontology alignment*, Frank van Harmelen, Sheila McIlraith, Dimitris Plexousakis (eds.), Proceedings of the 3rd International Semantic Web Conference (ISWC), Hiroshima (JP), Lecture notes in computer science **3298** (2004), 698–712.

[13] Euzenat, J., T. Le Bach, J. Barrasa, P. Bouquet, J. De Bo, R. Dieng *et al*, "D2.2.3: State of the art on ontology alignment – Knowledge Web project," realizing the semantic web, IST-2004-507482 Programme of the Commission of the European Communities, 2004.

[14] Horrocks, I., "Optimising Tableaux Decision Procedures for Description Logics," PhD thesis, University of Manchester, 1997.

[15] Horrocks, I., Implementation and Optimization Techniques,"The Description Logic Handbook, Theory, Implementation and Applications," Baader, F., Calvanese, D., McGuinness, D., Nardi, D. and Patel-Schneider, P. (Eds), Cambridge University Press, 2003, 306–346.

[16] Jannink, J., S. Pichai, D. Verheijen and G. Wiederhold, Encapsulation and composition of ontologies, Proceedings of AAAI'98 Workshop on Information Integration, 1998, 43–51.

[17] Kalfoglou, Y., and M. Schorlemmer, *Ontology mapping: the state of the art*, The Knowledge Engineering Review, **18** (2003) (1) 1–31.

[18] Le Duc, C., "Transformation d'ontologies basées sur la Logique de Description – Application dans le Commerce Electronique," PhD Thesis, Université de Nice, France, 2004.

[19] Lima, C. P., B. Fiès, G. Lefrancois and T. E. Diraby, The challenge of using a domain Ontology in KM solutions: the e-COGNOS experience, International Conference on Concurrent Engineering: Research and Applications 10TH ISPE 2003, Funchal, Portugal, 2003, 771–778.

[20] Lima, C. P., J. Stephens and M. Bohms, *The bcXML: Supporting eCommerce and Knowledge Management in the construction industry*, Itcon Journal, **8** (2003), 293–308.

[21] Mena, E., A. Illarramendi, V. Kashyap and A. Sheth, *OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies*, Distributed and Parallel Database – An International Journal, **8** (2000) (2), 223–271.

[22] Mitra, P., G. Wiederhold and J. Jannink, Semi-automatic integration of Knowledge sources, Proceedings of the Fusion'99 Conference, Sunnyvale, CA, USA, 1999, 9pp.

[23] Noy, N., Tools for Mapping and Merging Ontologies, "Handbook on Ontologies," Staab S., Studer S. (Eds), Springer-Verlag Berlin Heidelberg, ISBN 3-540-40834-7, 2004, 365–384.

[24] Patel-Schneider, P. F., P. Hayes and I. Horrocks (Eds), "OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation," 2004. URL: http://www.w3.org/TR/owl-semantics/.

[25] Rahm, E., and P. A. Bernstein, *A survey of approaches to automatic schema matching*, VLDB Journal: Very Large Data Bases, **10** (2001) (4), 334–350.

[26] Resnik, P., and M. Diab, Measuring verb similarity, Proceedings of the 22nd Annual meeting of the Cognitive Science Society, Philadelphia, 2000, 399–404.

[27] Rodriguez, M. A., and M. J. Egenhofer, *Putting similarity assessments into context: matching functions with the user's intended operations*, Proceedings of the 2nd international and interdisciplinary conference on modeling and using context CONTEXT'99, Lecture Notes in Computer Science, Springer-Verlag **1688** (1999), 310–323.

[28] Schmidt-Schaub, M., and G. Smolka, *Attributive concept descriptions with complements*, Artificial Intelligence, **48** (1991) (1), 1–26.