

Semantic Hubs for Geological Projects

e-WOK_HUB consortium¹

Yamine Aït Ameer (LISI), Nabil Belaid (ENSMA/LISI), Mohammed Bennis (INRIA), Olivier Corby (INRIA), Rose Dieng-Kuntz (INRIA), Jérémie Doucy (EADS), Priscille Durville (INRIA), Chimène Fankam (ENSMA/LISI), Fabien Gandon (INRIA), Alain Giboin (INRIA), Patrick Giroux (EADS), Sandrine Grataloup (BRGM), Bruno Grilheres (EADS), Florian Husson (BRGM), Stéphane Jean (ENSMA/LISI), Joël Langlois (BRGM), Phuc Hiep Luong (INRIA), Laura Mastella (ENSMP), Olivier Morel (BRGM), Michel Perrin (ENSMP), Guy Pierra (ENSMA/LISI), Jean-François Rainaud (IFP), Idir Aït Sadoune (ENSMA/LISI), Eric Sardet (CRITT), François Tertre (BRGM), Joao Francisco Valiati (IFP)

¹ INRIA Sophia Antipolis, 2004 route des Lucioles - B.P. 93,
06902 Sophia-Antipolis Cedex, France
Rose.Dieng@sophia.inria.fr

Abstract. This paper describes a service-oriented architecture for accessing resources through semantically designed portals called hubs. The services are dedicated to: (a) ontology management, (b) annotation generation from texts based on linguistic or machine learning techniques, (c) persistent storage of ontologies and metadata, and (d) semantic search in annotation bases or ontological databases. These services are, themselves, semantically annotated in order to facilitate their identification and composition. The application of our methodology is carried out within the e-WOK_HUB project in the geological domain.

Keywords: Semantic web, semantic annotation, ontologies, Service Oriented Architecture, semantic web service, persistence, semantic query, generation, storage, retrieval and evolution of metadata.

1 Introduction

The intended end-users of e-WOK_HUB project are geologists / researchers in geology, carrying out CO₂ storage prospection projects. They need to use a large variety of available resources such as scientific articles offering geological knowledge, internal or external reports of past projects, etc. The number of such resources increases since, while performing their tasks, geologists can produce additional knowledge described in new resources useful for the current project or for another prospection project. This increasing amount of resources has to be managed.

To address this use case, one option is to integrate these heterogeneous resources in a global architecture where semantics play a central role with metadata exchanges and

ontology-based searches. The e-WOK_HUB project¹ aims at developing a set of communicating portals (called hubs), offering both: (a) web applications accessible to end-users through online interfaces, and (b) web services accessible to applications through programmatic interfaces. The e-WOK_HUB system relies on such a service-oriented architecture (SOA) where business applications are built on service composition and orchestration and where a hub can be considered as a warehouse of semantic business resources. So far, a first prototype that focuses on geographical purposes has been realized.

The aim of this article is to show how semantic technologies are adapted and can be integrated in a global semantic architecture in order to improve the use of the geological knowledge base while respecting the end-user constraints. We will first describe the global architecture (section 2). Then, we will present our original contributions to the ontology and annotation management (sections 3 and 4). Next we will describe users' searches based on such ontologies and annotations (section 5). Finally, we will detail the architecture implementation (section 6) and conclude with our further work.

2 Global architecture

As in a usual service-oriented architecture, each service (implementing one or more services interfaces) just provides one or several processing capabilities which will be driven by a service orchestrator to define the business processes.

The general architecture can be seen as a multi-layer architecture. From a bottom-up view, we can describe it with the following layers:

- The **infrastructure** layer includes the hardware platform, the networks, the operating system and any system software like virtual machines.
- The **data** layer includes repositories for the storage of documents, metadata or knowledge elements as ontologies.
- The **component** layer includes the software modules developed by the e-WOK_HUB project's partners (which will be discussed in this paper) and some open-source solutions. These modules deal with semantic data.
- The **service** layer is composed of the interfaces implemented by the different components and offered by the hub. Among these, some are dedicated to metadata management.
- The **middleware** layer provides the communication and the messages distribution between services with an Enterprise Service Bus (ESB) : **Petals** from EBM Websourcing/ OW2.
- The **process** layer acts as an intermediary between services in order to perform workflows or business processes. Orchestration scripts can be defined by an external graphical tool that generates process description in languages such as BPEL² or XPD¹. The orchestration layer embeds an execution engine (**Orchestra** from Bull/OW2) that is able to run such process descriptions.

¹ <http://www-sop.inria.fr/edelweiss/projects/ewok/>

² www.oasis-open.org/committees/wsbpel/

- Finally, the **access** layer provides user interfaces to resources and services through a portal. The **eXo platform** solution has been adopted. The architecture (see Fig. 1) is complemented by transverse components dedicated to security and technical management of the hub.

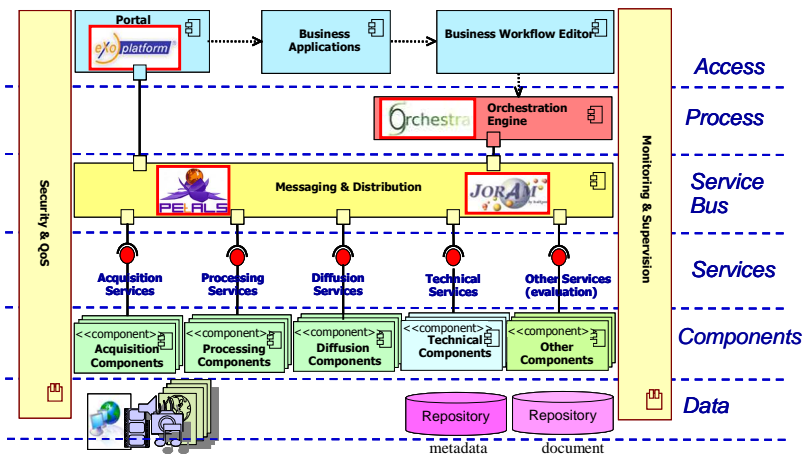


Fig. 1 : Global architecture overview

3 Ontology management

In such a semantic web architecture, services use domain models, in our case ontologies represented in RDF(S)/OWL [1][2]. These ontologies have to be developed prior to service runtime. At runtime, these ontologies have to be updated while the domain models evolve. So the different steps we have to deal with in such an architecture are: first, creation and edition of the ontology, then install process/deployment of the ontology in the semantic repository and last, evolution of the ontology (i.e. when an ontology must be modified, it is copied back from the semantic repository into the semantic development repository, the old version is still used as long as the new one is not deployed) and so on. This ontology life cycle requires dedicated services in the global architecture.

3.1 A collaborative editor for ontology creation and evolution: ECCO

An ontology dedicated service will allow users to create or adapt the ontologies in order to keep them up to date. ECCO (a contextual and collaborative ontology web editor) provides such a service. ECCO was developed by the EDELWEISS team of INRIA for e-WOK_HUB project. It was used at ontology creation time to develop models from domain texts in a collaborative way between domain experts

¹ http://www.wfmc.org/standards/docs.htm#XPDL_Spec_Final

(geologists), ontologists and developers. By extracting significant words in texts, manually via its graphical user interface (or semi-automatically using external embedded NLP¹ tools) ECCO users create a common vocabulary which can be organized into a hierarchy and then can be refined to describe semantic characteristics with OWL Lite properties. On each item (word of the vocabulary, or class/property/individual of the ontology) users can add status tags or textual comments to improve the collaborative work. The results of this collaborative process are kept in memory as RDF [3] annotations and can be requested by a semantic search engine. For now, ECCO is the only one existing ontology editor to provide such a way of collaborative ontology creation in the context of semantic web languages: Protégé² provides ontology versioning and collaborative modification of concepts/properties; WebODE³ does not keep links to the texts that are sources of the ontology; OntoLT⁴ plugin for Protégé and TexttoOnto⁵ tool suite above KAON⁶ offer ontology learning from texts but without collaborative aspects.

ECCO will also be used at evolution step. At that time, the ontologies to modify are edited with this tool. In this case, annotations relying on the currently modified ontology have already been created and used. The modifications have possibly an impact on these existing annotations. In order to send back the ontology modifications on ontologies to annotations, ECCO generates log files describing the current modifications in RDF syntax. These metadata are related to a meta-ontology about ontology modifications [4]. With such metadata, we will see in section 4.4 that annotation management services are able to update impacted annotations in order to keep the knowledge base consistent. This metadata generation on ontology modifications in RDF syntax is an original aspect of ECCO editor with respect to the state of the art. KAON offers ontology evolution capabilities but without solving the problems of propagation to related annotations.

3.2 Persistence of ontologies with OntoDB

Some of the ontologies used in the e-WOK_HUB project are related to the geographical domain. More precisely, these ontologies are used to describe geographical referenced objects by specifying their spatial coordinates defining a geometry object (polygon, point ...) in a geographical information system. One of the repositories used within the e-WOK_HUB approach is the OntoDB [5] database. It is associated with an editor called PLIB-Editor⁷ which enables the manipulation of stored ontologies and the associated data through a Java API and a graphical interface. Initially, OntoDB was not designed to support spatial information. Thus, we have extended the datatype system of its underlying ontology model (PLIB) to handle geometry types. Consequently, PLIB-Editor has also been extended with an

¹ Natural Language Processing

² <http://protege.stanford.edu/>

³ http://webode.dia.fi.upm.es/WebODEWeb/webode_home2.0.html

⁴ <http://olp.dfki.de/OntoLT/OntoLT.htm>

⁵ <http://sourceforge.net/projects/texttoonto>

⁶ <http://kaon.semanticweb.org/>

⁷ <http://www.plib.ensma.fr/plib/demos/ontodb/index.html>

appropriate end-user interface and a Java API. To our knowledge, PLIB-Editor is the only ontology editor allowing the definition and manipulation of geometry types. Below, we briefly describe the two major steps followed to implement geometry types in OntoDB and PLIB-Editor.

Management of spatial information within OntoDB

The PLIB datatype system has been extended with geometry types defined by the OpenGIS Consortium [6]. First, the OpenGIS geometry type hierarchy has been translated into an EXPRESS data model (the underlying data modeling language of PLIB). Then, starting from this EXPRESS data model, the corresponding persistency structures (database tables) have been automatically generated in OntoDB using a model transformation.

To be effectively processed by OntoDB, spatial information has to be supported by the underlying database management system (DBMS). Most DBMS currently propose a geometry extension. OntoDB is implemented on top of Postgres which can be extended with PostGIS¹ to enable spatial information (geometry types and functions) support. We have used it to make persistent the geometry type instances.

Definition of an access API and end-user interface for PLIB-Editor

The Java API for geometry types has been automatically generated from the EXPRESS data model of OpenGIS types previously defined. Each class of this API encodes the mapping between an OpenGIS geometry object and the corresponding PostGIS geometry object. Finally, an end-user interface was developed to allow the manipulation of geometry types within PLIB-Editor. At the creation of a geometry property, its coordinates dimension and the spatial reference system which defines the origin of the coordinates in the space must be specified. Currently, PLIB-Editor allows us to visualize a geometry property value in the text format (WKT).

3.3 Domain ontologies developed

For meeting e-WOK_HUB needs, we have defined:

- an **ontology of geographical terms**, which both rests on administrative nomenclature and on spatial (polygonal) area definition,
- an ontology for defining and managing **geological ages**,
- ontologies for describing **basic geology, geological units, geological boundaries, geological properties and geological processes**.

We relied on various sources of information. A set of representative text documents was first selected by domain experts that selected manually the vocabulary contained in these documents that was relevant to the geographical and geological aspects of the CO₂ storage subject. This vocabulary was tentatively classified. The results of this classification have then been compared with the classical terminologies already available in the geological domain (NADM, Geoscience ML). Considering these two sources of information and relying on the expertise of the e-WOK_HUB

¹ <http://www.postgis.fr/>

consortium members concerned (BRGM, IFP, ENSMP), we have defined domain ontologies adapted to our needs [7]. Fig. 2 provides an overview of the defined ontologies. Each grey box is a specific ontology. The basic geological ontology is zoomed in the central part of the figure showing relations between its main concepts and other ontologies.

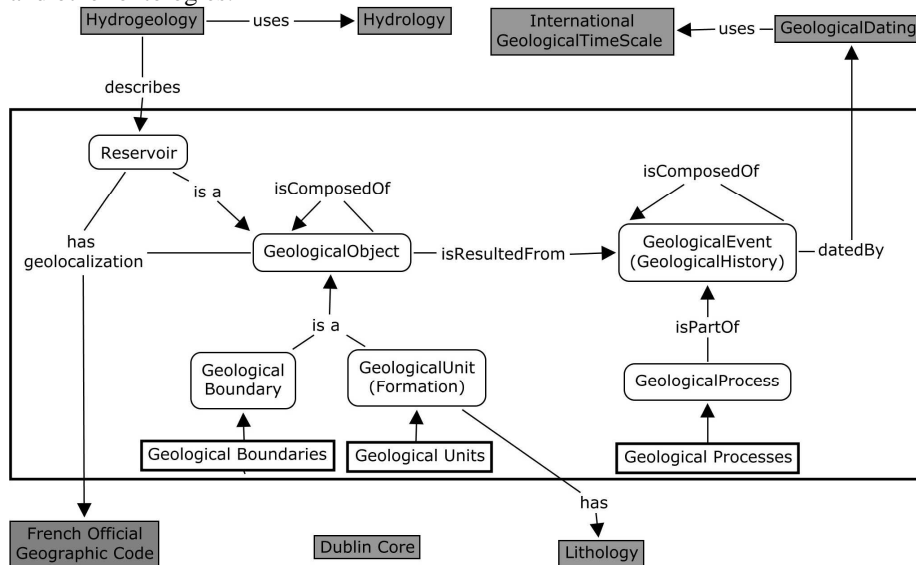


Fig. 2: Overview of e-WOK_HUB Domain Ontologies

4 Annotation management services

Documents entering the data base are formatted into the e-WOK_HUB exchange syntax and annotated in a semi-automatic way. The annotation process requires different successive services. Each of them uses answers of one or more of the previous chained services. So far, these annotation services have been tested on geographical purposes. But they are all generic and only rely on the domain ontologies used by the system. These services are (by running order): a language identification service that creates Dublin Core annotation¹, a syntactic analysis service (4.1), a KCRF annotation service (4.2) and a semantic annotation service (4.3). Next, all these generated annotations are extracted from the exchanged document and stored into the knowledge base (4.5). In this process chain, we will describe below the different annotation services and the storage characteristics.

¹ <http://dublincore.org/documents/2008/01/14/dcmi-terms/>

4.1 Linguistic annotation service

The linguistic annotation service performs a syntactic analysis on a text in order to get word grammatical classes (like noun, verb, adverb...) and then generates annotations containing this part-of-speech (POS) and lemma information. To perform this analysis, it uses the Dublin Core dc:language metadata generated before. The generated metadata of this linguistic service will be used by the next two services described below. For the syntactic analysis part, we need a usual linguistic pipeline with a tokenization process, a sentence splitting process and word syntactic analysis process. For that linguistic pipeline, we use the Gate platform [8], embedded as a web service, with the Tokenizer and the SentenceSplitter plugins (provided by the Gate platform) and the TreeTagger¹ tool [9] embedded as a Gate plugin.

For the annotation generation part, we have developed a lightweight ontology² to abstract part-of-speech forms in order to be able to exchange linguistic metadata independent from languages and from existing POS tools between services. This ontology captures the Penn-Treebank tags set [10] and adds subsumption relations between classes. We have augmented the number of classes taking into account more detailed linguistic forms. For example: tenses for verbs, gender and quantity for nouns and pronouns, language specificities etc.

The returned Gate document contains metadata describing part-of-speech and lemma information that uses a refinement of Penn-Treebank tags set for texts in English and a French specific tags set³ for texts in French. We have developed an XSL transformation [11] that takes a Gate document as input and generates RDF annotations relying on our linguistic ontology as output. The generated annotation on the word “is” in the sentence “*Chemical stratigraphy is less developed in shelf environments than in basin ones*” for example is shown below:

```
1 <rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
2   xmlns:w1='http://model.core.weblab.eads.com/'
3   xmlns:ling='http://ns.inria.fr/penn_treebank_pos#' >
4   <w1:Resource rdf:about="weblab://myWS/myDocument#2_4">
5     <ling:lemma>be</ling:lemma>
6     <ling:linguistic-desc>
7       <rdf:Description>
8         <rdf:type rdf:resource="http://ns.inria.fr/penn_treebank_pos#V"/>
9         <ling:tense>
10          <rdf:Description>
11            <rdf:type rdf:resource="http://ns.inria.fr/penn_treebank_pos#Present"/>
12          </rdf:Description>
13        </ling:tense>
14      </rdf:Description>
15    </ling:linguistic-desc>
16  </w1:Resource>
17 </rdf:RDF>
```

Fig. 3: Example of a linguistic metadata generated

4.2 KCRF annotation service

The task of automatic annotation of documents in a new domain such as CO2 capture may be very difficult. Indeed, this task requires the acquisition of implicit knowledge

¹ <http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>

² <http://www-sop.inria.fr/edelweiss/projects/ewok/penn-tree-bank-pos.rdfs>

³ <http://www.ims.uni-stuttgart.de/~schmid/french-tagset.html>

of experts through the annotation they make on documents. In order to accelerate this acquisition process, we propose to use supervised learning mechanisms to learn how to annotate documents on the CO2 capture fields. We propose to use Conditional Random Fields [12]. They are conditional graphical models enabling to model the conditional law of a sequence of labels (Y) conditionally to a sequence of observations (X). In the e-WOK_HUB domain, the sequence of labels consists of a sequence of semantic annotations of the considered domain (geological annotation, geographical annotation) and the sequence of observations, a sequence of words or sentences of documents.

Lafferty et al. [13] proposed a way to include textual kernels into the Conditional Random Fields framework. Kernels [14] are a way to represent similarity between documents. They enable to describe any classification or annotation problem using the same kernels and to use the same classification and annotation algorithm problems on a variety of tasks. Moreover kernels allow the introduction of a-priori knowledge (stop-words, linguistic annotations, general concepts, etc.). One of the tasks that is carried out within the e-WOK_HUB project is to build smarter kernel (smarter representation of the similarity between texts using a priori knowledge provided by other partners).

Learning the Kernelized Conditional Random Fields (KCRF) is done by an algorithm of parsimonious selection of kernels that maximizes the likelihood of the training set [15]. Annotation is carried out using a modified Viterbi algorithm. Another important issue concerning KCRF within e-WOK_HUB is to scale the learning algorithm to enable the user to learn on bigger amounts of data.

Last but not least, if early user tests (in the business intelligence domain) have validated the attractiveness of the approach, e-WOK_HUB will enable us to go further in our usability tests with non computer engineers by involving geologists.

4.3 Semantic annotation service

The semantic annotation service analyzes texts regarding a set of ontologies and already existing annotations. The aim of that service is to point out some themes evoked by texts. In our first prototype, the theme we are interested in is the geographical zone studied by documents. For that purpose, we re-use an existing ontology: INSEE's COG¹, describing the administrative French geographic area boundaries and the annotations representing all these areas. We also use an extension we have developed of these metadata describing geological well known areas.

The analysis process, adapted from an existing term extractor used in [16], tries to find mappings between words (or groups of words) in the text and the existing annotations. The mapping blueprint is represented by a SPARQL [17] query. In our geographical case, the query tries to map words to the existing values of the COG property named *http://rdf.insee.fr/geo/nom* describing geographical areas names.

```
PREFIX geo: <http://rdf.insee.fr/geo/>
SELECT ?x WHERE { ?x geo:nom ?n }
```

¹ Code Officiel Géographique, <http://rdf.insee.fr/geo/>

This query is an input of the semantic service which is thus independent from it. The service uses a semantic search engine called CORESE¹ [18] to answer the query. CORESE implements the RDF graph-based knowledge representation language and the SPARQL query language extended with the ability to use custom user functions in filter clauses. These external SPARQL functions are developed by users as Java methods. We use this CORESE functionality to customize the input query by adding a similarity function to check if a potential mapping is right or wrong. This similarity function we have developed uses the Jaro Winkler algorithm [19]. So, in the geographical case, the final internal query is the one below²:

```
PREFIX sim:
<function://fr.inria.annotator.nlp.CoreseSimilarityFunction>
PREFIX geo: <http://rdf.insee.fr/geo/>
SELECT ?x WHERE {
?x geo:nom ?n .
FILTER (sim:test(?n, 'a word') > 0.9) }
```

The results sent by CORESE are returned as RDF annotations on the text.

4.4 Evolution of annotations after evolution of ontologies

Since the ontologies used by the system can evolve, in parallel we studied thoroughly the problem of semantic annotation evolution caused by the changes of their reference ontologies [4][20]. The evolution of the reference ontology often leads to inconsistencies in the related semantic annotations.

We distinguish two cases of ontology evolution which can influence the consistency state of semantic annotations: (i) ontology evolution with trace and (ii) ontology evolution without trace of the changes which were carried out between two versions of the ontology.

In the first case, all the executed changes as well as the results of operations between two versions of the ontology are preserved in a log of changes. For each ontological change, we propose evolution strategies for restoring the consistent state of the influenced semantic annotations [20].

In the second case, we have proposed a process comprising two main steps:

- *Annotation inconsistency detection*: We apply inconsistency detection rules for detecting the actual inconsistent annotations that violate the consistency constraints defined for the annotation. In this phase, we use CORESE semantic search engine, for querying the annotation base taking into account the concept hierarchy and the relation hierarchy defined in the ontologies.
- *Annotation inconsistency resolution*: Once determined, the inconsistent annotations, will be repaired by applying inconsistency correction rules. We have established possible solutions for solving the propagation of ontological changes (related to concepts, properties, domain, range and datatypes) to their semantic annotations in order to keep consistency status.

¹ <http://www-sop.inria.fr/teams/edelweiss/wiki/wakka.php?wiki=Corese>

² The slack can be customized so that the similarity can be adjusted or relaxed.

These propositions were implemented and validated in the CoSWEM tool which facilitates evolution management. It enables to carry out some tasks automatically or semi-automatically: comparison of different ontologies, inconsistency detection and correction of the semantic annotations, etc. This tool was partially experimented in the framework of e-WOK_HUB with the INSEE's COG ontology and it will be later integrated in the e-WOK_HUB prototype. This approach for annotation evolution based on inconsistency detection rules and on inconsistency correction rules is original in comparison with related work [21][22].

4.5 Annotation storage in a database

In order to establish the link between a resource and associated metadata, we have introduced an annotation table in OntoDB. For each resource created, one or several annotations can be added to link the resource with an instance of an ontology concept. This annotation table establishes a many-to-many relationship between resources and instances of ontologies. It describes an annotation predicate or relationship between resources and instances of ontology concepts. An example of such a predicate is "geolocalized by" to express that a document *d* is geolocalized by *URI*, where *URI* describes an ontological concept instance. It includes columns for the following information:

- Concept_ID : the identifier of the concept (its URI for example)
- Resource_ID : the identifier of the resource (its URI for example)

Using an annotation table, we separate metadata description from concepts description. Therefore, metadata are defined and managed independently using a separate model for resources.

4.6 Conclusions

Linguistic and KCRF annotation services enable to generate annotations on textual documents. The evolution of such annotations after evolution of the corresponding ontologies can be tackled through CoSWEM. Annotation persistence can be offered through OntoDB. Such a combination of annotation services is quite original.

5 Ontology-based Search

We rely on SPARQL for offering semantic search services both on the RDF semantic annotation base and on the OntoDB database.

5.1 Creation of a semantic request from a geographic input

Geological CO₂ storage domain implies that we deal with geolocalized data. The two possible ways to express geolocalization in documents are the following. First, the actual coordinates of the given area (these coordinates are expressed in different

projection systems accordingly to the data source) are provided. This is the case of exploration wells for example. The second one is an indirect one. In this case, a reference denoting geographic coordinates is provided. This is the case of BRGM public reports which are geolocalized by cities or other administrative part names.

In this huge set of internal and external documents containing texts with geographic references, a geologist wants to find those that rely on its geographic interest area. From a user point of view, the best way to do that is to point such an area on a map. Hence, a cartographic functionality is embedded accordingly in user interfaces in e-WOK hubs. The render uses the Web Map Service (WMS) interoperability standards: the client (based on free MapBuilder and OpenLayer software) gathers and manages images sent by different servers. Adding the recent Web Processing Service (WPS) technology, users can now interact with the map to produce geometries that will be translated into global positioning coordinates. With this tool users are able to directly request geometric databases. However metadata does not always contain the coordinates for positioning. So, we also need requests relying on names of places. Therefore input geometries are transformed into a list of administrative divisions intersecting those geometries. After that, the administrative divisions list can be inserted in a query processed by a semantic engine. Consequently, from the semantic engine point of view, geometry aspects are hidden.

In the near future, we aim at finding a solution to directly use coordinates when they are known. The system should route and transform requests between semantic and geographic levels in an automatic manner. Semantics will have to help in decision of translation.

5.2 Semantic search service 1 : SPARQL syntax directly processed

In order to answer SPARQL queries, each hub needs a semantic search service and repository. The first one relies on CORESE [18] which enables the processing of RDF and OWL Schemas and RDF statements relying on conceptual graph formalism¹. It can perform queries in SPARQL syntax and rules over the base of RDF (stored as XML or N3 syntax). In our architecture, it is used at different levels. First, it is used at a domain level for answering users searches (on geographical requests for example, as shown in section 5.1). It is also used at a software level for domain reasoning. We have developed rules for geological time scales matching for example in order to be able to work with metadata coming from different sources referring to different time scales. At a lower level, it is used by the global architecture to find services and, in a future prototype version, to find chains of services.

5.3 Semantic search service 2 : by translation from SPARQL to OntoQL

The second repository used in the e-WOK_HUB approach is the OntoDB database. This database is equipped with the exploitation language OntoQL [23] enabling to define, manipulate and query data and/or ontologies stored in this repository.

¹ see the note from T. Berners-Lee on the subject: <http://www.w3.org/DesignIssues/CG.html>

However, in order to provide an access to data for any repository, a common access interface is required. We have chosen to use SPARQL since this language plays the role of a standard in the Semantic Web area. This section briefly describes our implementation of SPARQL on top of OntoDB.

Instead of implementing SPARQL from scratch, we have chosen to interpret its constructs by OntoQL constructs. More precisely, a sequence of OntoQL algebra operators (named OntoAlgebra [24]) calls is executed as an interpretation of a SPARQL query. As a consequence, we get benefits from the OntoQL queries implementation and optimization. Currently, due to the indexation possibilities offered by the PLIB typing system, solely a subset of SPARQL translatable into OntoQL and efficiently processed on OntoDB is implemented. This subset is characterized by two templates: one for queries on data and one for queries on ontologies.

SPARQL queries with a WHERE clause defined as:

$?id_1 \text{ type } C_1 [OPTIONAL] ?id_1 p_{11} ?v_{11} [OPTIONAL] ?id_1 p_{n1} ?v_{n1} [FILTER()] \theta \dots$ $?id_2 \text{ type } C_2 [OPTIONAL] ?id_2 p_{12} ?v_{12} [OPTIONAL] ?id_2 p_{n2} ?v_{n2} [FILTER()] \theta \dots$ \dots $?id_n \text{ type } C_n [OPTIONAL] ?id_n p_{1n} ?v_{1n} [OPTIONAL] ?id_n p_{nn} ?v_{nn} [FILTER()]$

can be executed. In this template, C_i and p_i are classes and properties of ontologies, brackets are used for optional elements and θ denotes one of the three SPARQL operators (\cdot , *OPTIONAL* or *UNION*).

To be efficiently processed on OntoDB, queries conform to this template must also satisfy the following rules:

- *Typing Rule.* For each triple $(?id, p, ?v)$ a triple $(?id \text{ type } C)$ must be defined.
- *Each property has a domain.* For each triple $(?id, p, ?v)$ the property p must be defined on the class C of the corresponding $(?id \text{ type } C)$ triple.

These typing rules avoid unnecessary accesses to ontologies while processing queries.

Queries on ontologies, that can be executed, have a similar form as queries on data:

$?id_1 \text{ type } E_1 [OPTIONAL] ?id_1 a_{11} ?v_{11} [OPTIONAL] ?id_1 a_{n1} ?v_{n1} [FILTER()] \theta \dots$ $?id_2 \text{ type } E_2 [OPTIONAL] ?id_2 a_{12} ?v_{12} [OPTIONAL] ?id_2 a_{n2} ?v_{n2} [FILTER()] \theta \dots$ \dots $?id_n \text{ type } E_n [OPTIONAL] ?id_n a_{1n} ?v_{1n} [OPTIONAL] ?id_n a_{nn} ?v_{nn} [FILTER()]$

E_i are entities of RDF-Schema (*rdfs:Class*, *rdf:Property*, etc.), a_i are attributes of RDF-Schema (*rdfs:label*, *rdfs:range*, etc.).

6 Implementation and Integration of e-WOK_HUB services

6.1 Implementation of the service-oriented platform : the exchange model

All the services described in the previous sections have been implemented separately by the different partners of e-WOK_HUB project. As these services need to collaborate through the e-WOK_HUB workflows, a reference exchange model has been designed in order to specify a common semantics for input and output

parameters. This model was formalized in UML to ease understanding and communication. Then it was transformed into an XML Schema according to the MDA approach. In this way, the schema defines the data types to be exchanged and can be imported in the WSDL description of each service.

Thus the processing services could be easily chained: a producer service will encode its data and offer them to a consumer service which will decode the data and then process them. The workflow will then be rationalized since it does not need to develop specific interfaces between each service. The reference model allows us to reduce the computational effort on data processing. The service chaining will also be simplified and the introduction of new services will not need too much adaptation.

The data model allows us to describe the various subclasses of the e-WOK_HUB Resource class, the generic structure of a Resource and an RDF-based annotation system to be used by services to add information on any Resource.

An e-WOK_HUB resource can be a multimedia document, an element used to compose a document (text, image, audio or video part), a segment of document (a sentence in a text, an image area, a sound sequence, a video frame, etc.), an ontology, a query, a service or a resource collection. Any such resource can be annotated.

6.2 Web services as resources

In this architecture, web services are considered as common resources like documents or annotations themselves. And thus, they can also be annotated. Consequently, the semantic search engine can be used as a web service directory that can be requested to find possible web service chains for example. Based on this point of view, we are exploring the service annotation process. We rely on the SAWSDL syntax [25] in order to express semantic information of the service inputs and outputs, the service operations and the services themselves. We have developed an XSL transformation to extract this semantic information from the SAWSDL description of a service into RDF data which can be stored and used by the semantic search engine.

The next step will be to use this metadata and the semantic engine to generate domain processes based on BPEL activities that will be deployed in the orchestration engine Orchestra chosen for e-WOK_HUB.

6.3 Description of the orchestration mechanism

Workflow management and service orchestration allow us to focus on the business process (the “what”) and ignore the technical implementation of services (the “how”). The chain of process can be designed graphically with a Business Process Modeler which generates a program in an execution language (BPEL in e-WOK_HUB project). Then, the orchestration engine will run the program and invoke the services according to the process specification.

The orchestration engine is embedded in the ESB as a JBI component. In this way, all the services known by the bus can be invoked in a specified process and the whole process can be exposed on the bus and made available to clients.

As further work, we will focus on the dynamic orchestration capabilities and on the discovery and the selection of the services to be invoked in a process by using a semantic description of the involved elements.

7 Conclusions

This paper presented the first results of the e-WOK_HUB project: based on a service-oriented architecture, from ontology viewpoint, we developed new ontologies in the domain of geology, as well as an original tool for collaborative building of an ontology (ECCO) and we extended an editor for persistence of ontologies (PLIB Editor), with the capability of management of spatial information in an ontological database. For metadata generation, our system offers (a) a linguistic annotation service, (b) original learning-based techniques for semi-automatic annotation of texts. It also provides metadata storage in databases. For metadata evolution, it offers rule-based techniques for propagation of ontology of modifications towards their related annotations; it also supplies semantic search services, either based on SPARQL directly or obtained by translation from SPARQL to OntoQL. In addition to the originality of each of these services, their integration in this global architecture and their application in geosciences domain constitute an originality of the e-WOK_HUB system.

The first prototype enabling to execute a sequence of such services was tested on a geographical scenario. Based on several reports including geographical information, document management services are involved first. The scenario then use the linguistic and KCRF annotation services to generate and store annotations on these reports content. In a second step, a semantic query is generated from a map by users and sent to the semantic search engine to retrieve documents relative to the given area.

As further work, we will also develop the applicative services for integration of results of technological watch in the semantic repository and for querying one or several project memories. In addition to our already existing evaluation protocols for some services, we will perform an evaluation for the global system.

References

1. Brickley, D., Guha, R.V., McBride, B.: RDF Vocabulary Description Language 1.0 : RDF Schema. W3C Recommendation (2004) w3.org/TR/2004/REC-rdf-schema-20040210/.
2. McGuinness, D. L., Harmelen F. v.: OWL Web Ontology Language. Technical report, W3C Recommendation (2004) w3.org/TR/2004/REC-owl-features-20040210/.
3. Manola, F., Miller, E., McBride, B.: RDF primer. Technical report, W3C Recommendation (2004) w3.org/TR/2004/REC-rdf-primer-20040210/.
4. Luong, P.-H. : Gestion de l'évolution d'un Web sémantique d'entreprise, PhD Thesis, Ecole des Mines de Paris, December 2007.
5. Dehainsala, H., Pierra, G., Bellatreche, L.: OntoDB: An Ontology-Based Database for Data Intensive Applications. In Proc. of the 12th Int. Conf. on Database Systems for Advanced Applications (DASFAA'07). LNCS. Springer (2007) 497-508. ISBN 978-3-540-71702-7

6. OpenGIS Simple Features Specification for SQL - Revision 1.1 Open GIS Consortium, Inc. http://portal.opengeospatial.org/files/index.php?artifact_id=829
7. Mastella, L., Perrin, M., Ait Ameer, Y., Abel, M., Rainaud, J.-F.: Formalising geological knowledge through ontologies and semantic annotation, 70th EAGE Conference and Exhibition, Rome, 9-12 June 2008.
8. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proc. of the 40th Anniv. Meeting of the Assoc. for Computational Linguistics (ACL'02) (July 2002)
9. Schmid, H. : Probabilistic Part-of-Speech tagging using decision trees. In: International Conference on New Methods in Language Processing, Manchester, UK, 1994
10. Marcus, M.P., Santorini, B., Ann Marcinkiewicz, M.: Building a Large Annotated Corpus of English: The Penn Treebank. In Computational Linguistics, Volume 19, Number 2 (June 1993), pp. 313--330 (Special Issue on Using Large Corpora)
11. Berglund, A.: Extensible Stylesheet Language. Technical report, W3C Recommendation (2006) www.w3.org/TR/2006/REC-xsl11-20061205/.
12. Lafferty, J., McCallum, A., and Pereira, F. Conditional random field : Probabilistic models for segmenting and labeling sequence data. In Proc. 18th International Conf. on Machine Learning, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.
13. Lafferty, J., Zhu, X., and Liu, Y.: Kernel conditional random fields : representation and clique selection. In Proc. of the 21st Int. Conf. on Machine Learning (ICML 2004), 2004
14. Renders, J. M. : Application des méthodes à noyaux à la fouille de données textuelles. In Acte de CIFT, 2004.
15. Grilheres, B., Canu, S., Brunessaux, S.: Patent FR 07 700 Semi-automatic annotation of multimedia documents.
16. Khelif, K., Dieng-Kuntz, R., Barbry, P.: An ontology-based approach to support text mining and information retrieval in the biological domain, Special Issue on Ontologies and their Appl., Journal of Universal Computer Science (JUCS), Vol. 13, No. 12, pp. 1881-1907
17. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for rdf. Technical report, W3C Recommendation (2008) www.w3.org/TR/rdf-sparql-query/.
18. Corby, O., Dieng-Kuntz, R., Faron-Zucker, C.: Querying the semantic web with the core search engine. In Proc. of the 16th Eur. Conf. on AI (ECAI'04), IOS Press (2004), 705–709
19. Winkler, W. E.: The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04. <http://www.census.gov/srd/www/byname.html>. 1999.
20. Luong, P.-H. and Dieng-Kuntz, R.: A Rule-based Approach for Semantic Annotation Evolution. The Computational Intelligence Journal, 23(3):320-338. Blackwell Publishing, Malden, MA 02148, USA.
21. Rogozan, D. and Paquette, G.: Managing Ontology Changes on the Semantic Web. Proc. of the 2005 IEEE/WIC/ACM Intl Conf. on Web Intelligence (WI'05), pp. 430-433.
22. Stojanovic, L., Stojanovic, N. and Handschuh, S.: Evolution of the Metadata in the Ontology-based Knowledge Management Systems. 1st German Workshop on Experience Management: Sharing Experiences about the Sharing of Experience, 2002, pp. 65-77.
23. Jean, S., Ait-Ameer, Y., Pierra, G.: Querying Ontology Based Database Using OntoQL (an Ontology Query Language). In: Proceedings of Ontologies, Databases, and Applications of Semantics (ODBASE'06). (2006) 704-721
24. Jean, S., Ait-Ameer, Y., Pierra, G.: An Object-Oriented Based Algebra for Ontologies and their Instances. In: Proceedings of the 11th East European Conference in Advances in Databases and Information Systems (ADBIS'07). LCNS 4690, Springer (2007) 141-156
25. Farrell, J., Lausen, H.: Semantic Annotations for WSDL and XML Schema. Technical report, W3C Recommendation (2007) <http://www.w3.org/TR/2007/REC-sawSDL-20070828/>.