AN ONTOLOGY AND INDEXATION BASED MANAGEMENT OF SERVICES AND WORKFLOWS APPLICATION TO GEOLOGICAL MODELING

Nabil Belaid[1*], Stéphane Jean[1], Yamine Ait-Ameur[2] and Jean-François Rainaud[3]
[1]*Laboratoire d'Informatique Scientifique et Industrielle*
Ecole Nationale Supérieure de Mécanique et d'Aérotechnique
Avenue Clément Ader, Futuroscope (86961), France
[2]*Ecole Nationale Supérieure d'Electrotechnique, d'Electronique, d'Informatique, d'Hydraulique et des Télécommunications*
Rue Charles Camichel, Toulouse (31071), France
[3]*Institut Français du Pétrole Energies Nouvelles*
Avenue Bois Préau, Rueil-Malmaison (92852), France

ABSTRACT

Management of IT services in general and Web Services (WS) in particular currently relies on syntactic descriptions of service interfaces using languages such as WSDL (Web Service Description Language), BPEL (Business Process Execution Language) or XPDL (XML Process Definition Language). As a consequence, complex management tasks such as WS discovery, interoperability or adaptability can not be performed with the level of precision expected by users. As a solution, ontologies have been used in several approaches to provide a semantic description of WS. However, most of these approaches do not distinguish syntactic descriptions of WS, useful for their storage and execution, from their semantic description, useful for their discovery and composition. Moreover, these approaches do not provide a hierarchical classification of WS according to the semantic functions they perform. To solve these problems, we propose a semantic description of WS based on the definition of *ontologies of services* and *semantic indexations*. This approach consists in enriching WS with an abstraction level that will leverage the management capabilities of WS without overloading their descriptions on the one hand and provide a hierarchical relationship between the semantic descriptions of the services on the other hand. Our proposition relies on an ontology based database for the representation and persistence of WS and their semantic characterizations represented by the created ontologies of services. To validate our approach we show its application to the geological modeling field and the specific tools we have developed.

Keywords: Ontology, Meta-modeling, Semantic Service, Semantic Workflow, Semantic Indexation, Ontology based Databases, Geological Modeling

1. INTRODUCTION

Nowadays, computer services such as software or complex programs are used to perform multiple tasks such as data modeling, processing and storing. In particular, Web Services (WS) are designed to support interoperable interactions over a network. Currently, a huge amount of WS is available on the Web thanks to the progress of networks and distributed computing. These WS are designed for several application fields like electronic commerce, concurrent engineering, scientific computations or grid computing. The composition of WS enables even more complex and fruitful tasks.

Generally, in order to use WS and their compositions, several functionalities have to be enabled from which we quote:

1. discovering a service performing a stated function and/or manipulating a given concept;
2. defining correct services compositions by composing other available services through the "functions" they perform;
3. replacing a service by another one performing a similar function;
4. ensuring the quality of the discovered and/or composed service when it is invoked. By quality,

Corresponding author: nabil.belaid@ensma.fr

المنارة للاستشارات
www.manaraa.com

we mean the satisfaction of properties like availability, safety and response time.

Nevertheless, enabling the previously enumerated functionalities is limited within the current description languages and standards for WS in particular. Indeed, many languages have been proposed and/or standardized to describe WS and their composition. These languages, such as WSDL [19] for WS or BPEL [8] and XPDL [20] for compositions of WS, offer a syntactic oriented description of WS useful for their storage and execution. However, these languages have limitations. One of the most obvious is that these languages do not explicitly and precisely state the performed functions behind the described services and compositions. Indeed, these languages mainly describe service interfaces using names which do not carry any information, i.e. “explicit semantics”, about the accomplished tasks. Thus, the WS discovery process is based on services registries, such as UDDI (Universal Description, Discovery and Integration) [12], which are queried according to the syntactical definitions of WS and/or to the definitions of their input and output parameters usually defined by XML schemas. Usually, and in particular in our application domain, the geological modeling field, this syntactic discovery process does not provide the level of precision expected by users. The same observation can be done for other WS management tasks such as adaptability or composability of WS. The lack of explicit semantics in WS description limits the capability of WS management.

Recently, several approaches have been proposed to handle the lack of semantics in the description of WS [6,10,13,17]. Ontologies, which provide meaning using concept and relationships, are used in these approaches for semantic WS descriptions. These semantic descriptions lead to enhanced discovery algorithms that provide more precise results. However, these approaches raise two problems. Firstly they do not make a clear distinction between the syntactic description of a WS, used for its storage and execution, and its semantic description, used for tasks such as their discovery and composition. As a consequence, these approaches require a modification of existing syntactic descriptions of WS as well as corresponding tools. Secondly, these approaches do not provide a hierarchical classification of WS according to the semantic function they perform. This lack of subsumption relationships between semantic WS makes it hard to replace a service by another one performing a similar function.

Our proposition relies on the definition of *ontologies of services* and *semantic indexations* to address the two previous problems. We have introduced these two notions in [1]. In this article, we extend this work by describing more deeply these notions and showing the design and implementation of the approach we proposed for the management of IT services and workflows.

Ontologies of services provide semantic descriptions for IT services in general. Throughout this article though, we consider particularly the case of WS descriptions. Using semantic indexation, ontologies of services are an abstraction level that will enhance semantically WS without overloading their syntactic descriptions. We define the concepts of the ontologies of services as *semantic services.* They are abstract descriptions carrying the semantics i.e. the performable functions of WS. Moreover, ontologies of services organize their semantic services in hierarchies whose semantics is the subsumption. This hierarchical classification, according to the carried semantics, enables for instance replacing an IT service by another one performing an exact, a more general or a more precise function.. Finally, in order to validate this approach we have implemented it within an ontology-based database and applied it to the geological modeling field.

This article is structured as follows. Next section presents the geological modeling domain from which we draw our case studies. We describe how WS are currently managed in this field and the raised issues. Section 3 reviews the main related work aiming at improving the semantic description of services and workflows. In Section 4, we detail our approach. We show how we represent services and data they reference both at the syntactic and semantic levels with a clear separation between them. Section 5 overviews the implementation of our approach supported by an ontology based database (OBDB). To show the interest of our approach, we present several interesting queries enabled by our approach for defining adaptive services and for discovering services. Section 6 describes the software we have developed in order to hide the complexity of our approach. We show how it has been used to support the geological modeling tasks. Finally, we conclude and give some perspectives of our work in Section 7.

2. A MOTIVATING EXAMPLE

The geological modeling task is a part of the petroleum engineering. Its goal is to characterize petroleum reservoirs in terms of geometry, topology and properties for petroleum extraction and/or CO2 storage. Figure 1 shows the major steps of the geological modeling task, as it is planned out by geologists.

The geological modeling task is achieved by processing data on a particular prospect (zone containing potentially a reservoir) such as 3D-images obtained from a seismic exploration, well logs acquired from drilling trajectories or observations of

المنارة للاستشارات
www.manaraa.com

rock samples. Then, the geologists, with the aid of computer applications, perform a seismic interpretation on the seismic images in order to identify the main geological objects such as horizons and faults. The result of this step is a structural model. The interpretation of well logs (intersections between well trajectory and horizons) is useful to make adjustments in the identified positions. For the next steps, the geologists consider the geological configurations, the rock structures and combine them to model important properties such as porosity or permeability. Then, a simplified reservoir grid model is defined for faster calculations, etc. The resulting reservoir model will be used by reservoir engineers to compute realistic CO2 fluid migrations simulations.

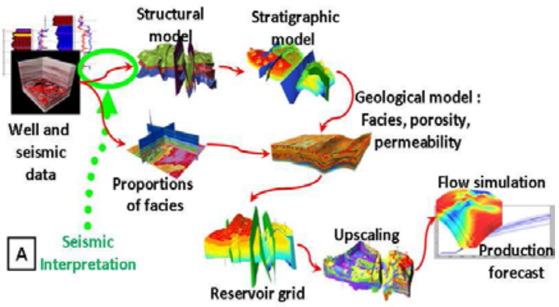


Figure 1: Major steps of the geological modeling task

Each arrow of Figure 1 corresponds to complex processes of functions. To illustrate our proposition, we detail in next subsections the seismic interpretation process (first part of the geological modeling task). We show how it is planned out by geologists and how its functions are implemented as WS and workflows.

2.1 Processes in Seismic Interpretation

The seismic interpretation aims at building a structural model from the basic geological objects. Figure 2 shows an example of a process performing a seismic interpretation as planned out by geologists.

In this figure, starting from a *Seismic Cube* representing seismic images of a particular reservoir, an extraction of reflectors is performed (*Reflector Extraction*) to extract the reflectors which are small visible parts of geological objects. Two functions are performed in parallel: combining reflectors in order to shape horizons (*Horizon Reflector Association*) and faults (*Fault Mirror Detection*). In parallel to this function, channels are extracted (*Channel Extraction*). Finally, the three geological objects are combined (*Structural Model Assembling*) to obtain a rough *Structural Model*. These processes are implemented as WS as shown in the next section.

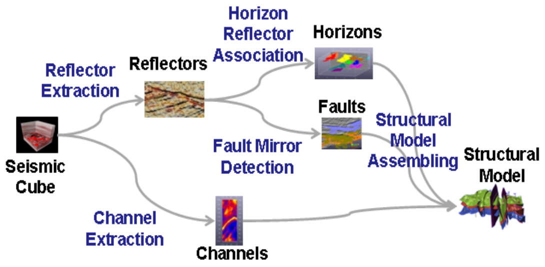


Figure 2: A seismic interpretation process designed by geologists

2.2 Processes Implementation and Management

The next step concerns the implementation of seismic interpretation process described in the previous section. This is achieved by implementing the functions that compose the process as IT services (WS, software, etc.). In order to achieve this goal, geologists work with IT developers to implement the target services and to establish data models representing the referenced concepts. Then, a workflow is designed as an orchestration of services and can thus be executed.

Figure 3 shows examples of software (interactive services) implementing the previously described functions and data models representing the referenced concepts. This implementation looses most of the semantics described in the previous section. For example, the *Reflector Extraction* function is performed by the *getRef* service and the concept of *Horizon* is represented by the *XYZFile* data model. It stores coordinates of points and is not specifically used to represent geological objects such as *Horizon*. Conversely, geological objects are not exclusively representable in *XYZFile*. The semantics carried by the data model *XYZFile* is indeed lost. Likewise, the term *getRef* does not give explicit information about the function this WS performs.

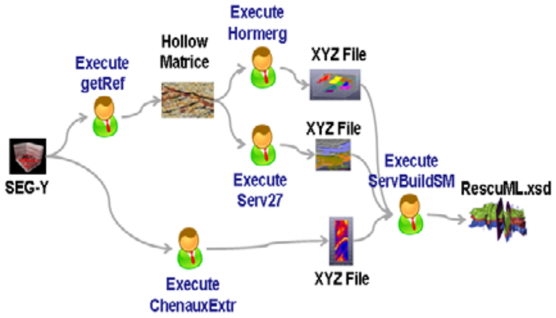


Figure 3: A seismic interpretation workflow

In the current systems used in geological modeling, only syntactic descriptions of services and workflows are stored. Yet, some of the most important issues in this field are to reuse these existing services and to build new workflows with limited technical knowledge. Currently, most of these tasks rely heavily on oral explanations, fuzzy

المنارة للاستشارات
www.manaraa.com

discovery or even redeveloping new services performing already implemented functions.

This example will be used throughout this article to illustrate our approach. The choice to draw case studies from the geological modeling field is motivated by the fact that it involves complex workflows and that we have access to precise information through the work of [18].

3. RELATED WORK

Several approaches have been proposed to attach explicit semantics to services and compositions of services in order to facilitate their management (discovering, composition, adaptability, etc.). Ontologies in this zone, as formal and consensual conceptualizations of domains, play a major role in associating explicit semantics to service and workflows descriptions. These approaches can be divided into two categories:

1. *annotation-based approaches* like SAWSDL [6] or SESMA [14]. They define an annotation layer over existing description languages. They annotate either WSDL or BPEL documents by semantic information issued from an ontology. It is an upper layer added to WSDL that allows defining semantic terms used in WSDL by referencing RDF-based ontologies. Another approach defined in [13] proposes extensions of UDDI registries in order to handle some semantic extensions;
2. *ontology-based approaches* like OWL-S [10] or WSMO [17]. They define ontologies describing services. An OWL-S ontology defines services as a composition of processes that may be atomic or composite. These processes have input and output data models that may refer to existing domain ontologies.

Thus, most of these approaches proceed by annotating or by extending web services description languages like WSDL or BPEL. They do not provide a clear separation between the syntactic description offered by these languages and the semantic description offered by their extension or annotation. As a consequence existing WS and tools managing them have to be adapted to these approaches. Another limitation is that they do not manage heterogeneous syntactic descriptions. Moreover, these approaches suffer from the absence of a subsumption relationship between semantic services which is useful for adaptability for instance.. Indeed, although WSMO does not refer to a specific service description language, like OWL-S, it does not define semantic web services organized in an ontology with a subsumption relationship nor enforces the use of such ontologies for services.

Several other workflow management systems have been proposed such as WDO-It! [15,16] and Pegasus [5]. WDO-It! enables the semantic characterization of functions but do not link this semantic layer to existing services and workflows. It enables the design of “abstract” workflows but its goal is not their execution. Pegasus starts from existing scientific workflows which are supposed to pre-exist. Then, its objective is to refine them in order to obtain “executable” workflows. Another approach [2,9] proposes the notion of community of services that gathers, in a structure named community, a set of WS which share the same semantics. However, they do not provide a subsumption relationship between WS nor a clear separation with existing syntactic descriptions of WS.

4. KOUR PROPOSED SEMANTIC MANAGEMENT OF SERVICES

The main idea behind our approach is to add a semantic layer to the syntactic descriptions without overloading them. This semantic enrichment leverages the discovery for instance of heterogeneous syntactic descriptions. In this section, we first present in more details our approach for indexing existing syntactic descriptions of IT services and workflows and then, we explain how, based on this work, new workflows can be built. Although our approach is generic, throughout this section, examples are given in the geological modeling field.

4.1 Indexing Existing Services and Workflows

In our approach, we propose to add a semantic layer by creating ontologies of services whose concepts are semantic services. They are used in order to semantically index (characterize) existing services and workflows descriptions and the existing engineering models they reference in their descriptions. Our approach (i) stores heterogeneous descriptions of IT services and workflows in a repository and (ii) adds a semantic layer through ontologies in order to explicit the semantics of stored services and workflows. Our work is complementary to [11] whose objective is to extract the semantics from data models and files in the geological modeling field.

Our approach is illustrated in Figure 4. It includes two parts (vertical decomposition) and two levels (horizontal decomposition) which we describe and illustrate in this sub-section.

4.1.1 Implementation Level (Figure 4 (A)) Engineering Models (Figure 4 (A.1))

Data are instances of engineering models whose types or formats are usually defined by XML-Schemas. They represent the input and output of software, Web services and workflows. For example, formatted files such as SEG-Y (usually

المنارة للاستشارات
www.manaraa.com

used to represent seismic images) or XYZ File (used to represent coordinate of points) are such complex engineering models.

Services and Workflows (Figure 4 (A.2))

Services and workflows are described in languages such as WSDL for WS and BPEL and XPDL for compositions of WS. In geological modeling, an example of WS is a process that extracts a reflector. This WS is named *getRef*. It takes as input data representing a seismic cube and produces as output data representing reflectors. These input data are represented in a specific format like *XYZ File*.

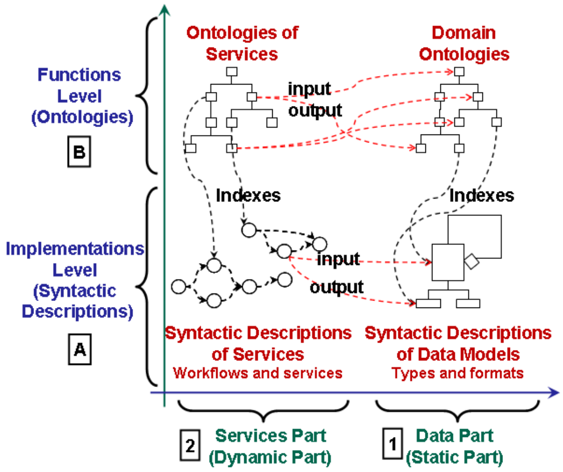


Figure 4: Our approach for managing of services and workflows

4.1.2 Implementation Level (Figure 4 (A))

Domain Ontology (Figure 4 (B.1))

Domain ontologies give explicit semantics to the data and engineering models. In geology and more specifically in the geological modeling domain, several domain ontologies exist. They define concepts such as *Horizon*, *Fault*, *Reflectors* and *SeismicCube*. In our approach, these concepts index the engineering models enriching them with explicit semantics through an independent layer without overloading them. Thus, these concepts hide the technical details of engineering models such as coordinate of points or seismic images.

Notice that a single ontology concept may index multiple engineering models. Conversely, the same model can be indexed by multiple ontology concepts. For example, h*orizons* can be represented in both *XYZ files* and *IJK files* whereas *XYZ files* can represent both *horizons* and *faults*.

In Figure 5, we show a fragment of a domain ontology defined for the seismic interpretation field.

The main concepts of this ontology related to our example are the following. *GeoConcept* is the root concept. This concept has two subclasses: Well and *SeismicSurface*. A *SeismicSurface* is part of *SeismicCube*. A *SeismicSurface* can be a Channel, a Fault or an Horizon. A Horizon is composed by Reflector which are also *SeismicSurface.*

Services Ontology (Figure 4 (B.2))

Ontologies of services are semantic descriptions of services and workflows in a given domain. They define hierarchies and compositions of services. We call a concept of an ontology of services a *semantic service.* A semantic service represents a performable function. Such a concept is independent of a specific implementation. For example, the semantic service that indexes the WS *getRef* is named *ReflectorExtraction.* It is a specialization of an *Extraction* function. It takes as input an instance of the concept *SeismicCube* defined in the ontology depicted Figure 5 and returns a *Reflector* defined in the same ontology.

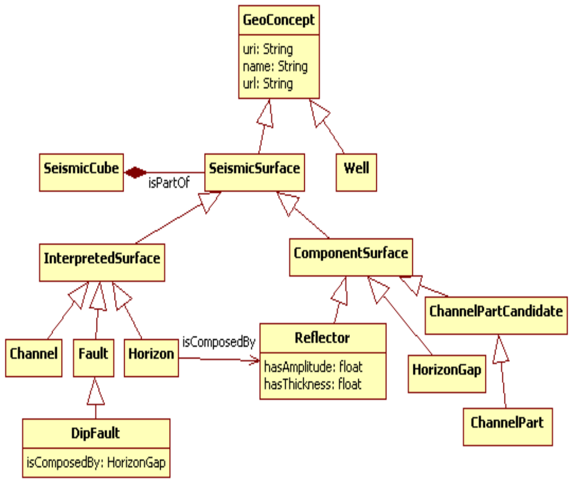


Figure 5: Extract of a domain ontology for the seismic interpretation field

Notice that the same semantic service may index multiple workflows or services when it proposes multiple implementations for the same function. Conversely, two semantic services may index the same generic service or workflow.

Ontologies of services are different from other ontologies as they characterize particular elements: services. Indeed, the services have input, preconditions, etc. Thus, these ontologies are designed following a specific model of ontologies described in Section 5.

In Figure 6, we present an extract of the ontology of services we have defined with the help of geologists for the seismic interpretation functions. *GeoService* is the root concept that subsumes the seismic interpretation functions. *Extraction*, *Association* and *Merging* are derived from it to represent the services and workflows for respectively extracting geological objects, associating multiple objects and merging them.

www.manaraa.com

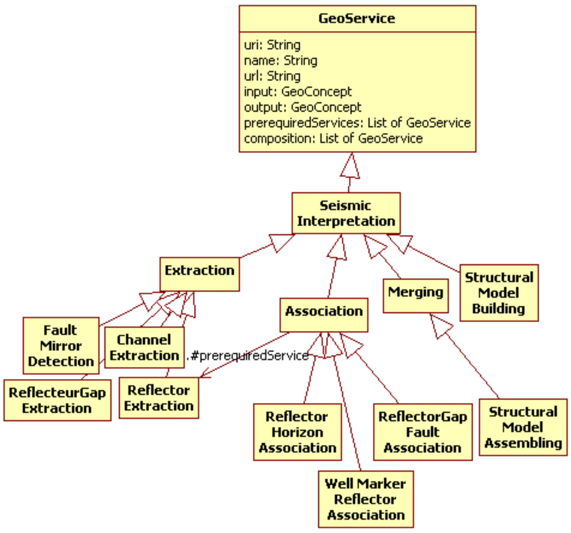


Figure 6: Extract of an ontology of services for the seismic interpretation field

An ontology of services represents a model that characterizes semantically the WS and workflows of a given domain. However, some constraints have to be added in order to ensure its consistency such as the covariance relationship [3]. It is defined as follows.

If the semantic service SS_i defined as follows (DO stands for *domain ontology concept*):

SS_i: $DO_{i,1} \times\ldots\times DO_{i,n} \rightarrow DO'_{i,1} \times\ldots\times DO'_{i,n}$

subsumes SS_j defined as follows

SS_j: $DO_{j,1} \times\ldots\times DO_{j,n} \rightarrow DO'_{j,1} \times\ldots\times DO'_{j,n}$

then

$DO_{i,1} \subseteq DO_{j,1}, \ldots, DO_{i,n} \subseteq DO_{j,n}$

and

$DO'_{i,1} \subseteq DO'_{j,1}, \ldots, DO'_{i,n} \subseteq DO'_{j,n}$

In Figure 7, we illustrate the covariance relationship. The semantic service *Extraction* takes as input the *SeismicCube* data ontology concept and produces a *SeismicSurface*. The covariance relationship implies that its sub semantic service *ReflectorExtaction* takes as input a *SeismicCube* and produces a *Reflector*.

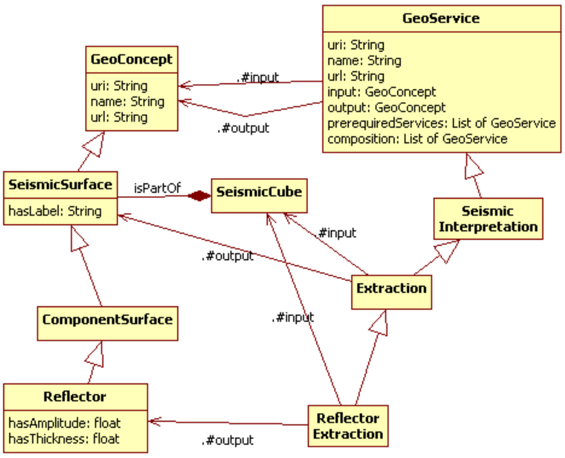


Figure 7: A fragment of a service ontology respecting the covariance relationship

In this section, we have shown that our approach clearly separates the syntactic descriptions of WS and workflows from their semantic description. Syntactic descriptions of WS and worklows use specific data models to describe their inputs and outputs. Semantic descriptions are defined as ontologies of services that use domain ontologies to characterize input and output of a semantic service. The link between the syntactic and semantics layers is done through indexation. In next sections, we show the interest of our approach for managing WS and workflows.

4.2 Building New Workflows

Geologists involved in a geology modeling activity have to compose different functions to achieve a specific task. However, the workflow corresponding to a given task is not predefined. Geologists have to try different compositions. Currently, each workflow has to be defined by the IT service which limits and slows down the work of geologists. Thus, an objective of our work is to enable the creation of new workflows with limited technical knowledge. Thanks to the previously presented work, it becomes possible to achieve this task by building semantic workflows first. We call semantic workflow, the composition of semantic services (concepts of ontologies of services). Since semantic services represent the performable functions and not syntactic descriptions, composing them is natural for ”experts in a given domain” and more complex functions are described. After having designed a semantic workflow, a workflow can be build by selecting a WS for each semantic service that composes the global semantic workflow as it is illustrated in Figure 8. Next sub sections give the details of this process

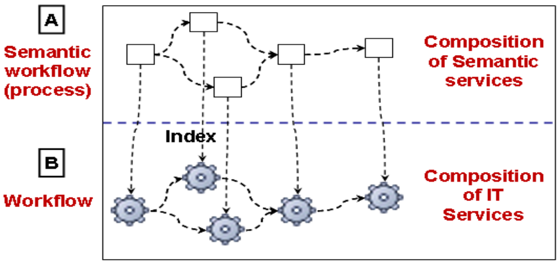

A
Semantic workflow (process)
Composition of Semantic services
Index
B
Workflow
Composition of IT Services

Figure 8: Semantic and IT workflows

4.2.1 Description of Semantic Workflows

Designing semantic workflows requires defining a description language and operators in order to compose semantic services. Figure 9 presents the simple semantic workflow model we have established inspired by classical syntactic description models. A semantic workflow can be derived in a composite semantic service (semantic sequence, a semantic parallel or a semantic loop) or a basic semantic service. A semantic loop is composed by

www.manaraa.com

one semantic workflow while a semantic sequence or parallel can be composed of multiple semantic workflows. A new designed semantic workflow can be used to enrich the ontology of services and thus behave as a new semantic service.

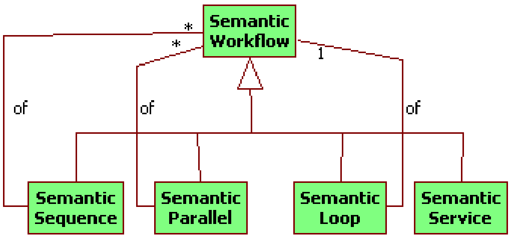


Figure 9: A semantic workflow description model

4.2.2 Building a Processing Workflow

When a semantic workflow has been defined, a workflow can be built or derived by selecting a WS for each semantic service that composes the global semantic workflow. For example, starting from a semantic workflow which is similar to the process described in Figure 2, an IT workflow is obtained and shown in Figure 10. In this figure, the process starts from a *SEG-Y File* representing a 3D-image. Then, the *getRef WS* (described in WSDL) is executed to obtain a *Hollow Matrix file*. Next, the IT workflow *Hormerg* (described in BPEL) and the service *Serv27* (described in WSDL) are performed in parallel to obtain two *XYZ Files* which represent respectively the *Horizons* and *Faults* concepts.

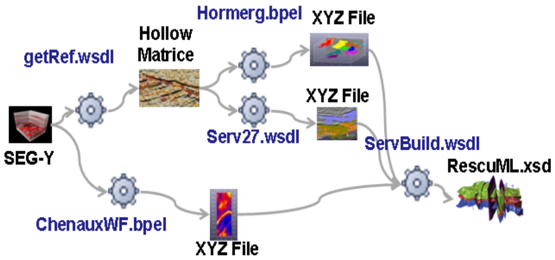


Figure 10: A seismic interpretation IT workflow

Although the semantic services of a semantic workflow compose, it may not be the case for the obtained IT workflow (necessary condition). In the previous example, if the *ChenauxWF* workflow has an *IJK File* as output, it can not be composed with the following service since they take an *XYZ File* as input. In this case, an *adapting WS* that converts *IJK Files* to *XYZ Files* is searched in the database and can be automatically added to the workflow if found.

This section has shown the basics of our approach and an example of a complex management task enabled by our approach: the creation of new workflows with limited technical knowledge. In the next two sections we present the operational validation of our approach we have performed. It consists of the implementation of this approach using a particular database structure and the development of a tool applied to the geological modeling domain.

5. IMPLEMENTATION

The implementation and validation of our approach require first to store the ontology characterizations and semantic indexations of services and workflows and secondly to show the interesting queries enabled by our approach.

For the persistence of all the data necessary we opt for a database to get a scalable storage. However since we have to store three modeling levels: (i) meta-models or model of ontologies, (ii) models and ontologies and (iii) instances; and two parts: (1) data and (2) services, we need a database architecture that provide a semantic based storage and search. Indeed, classical database architectures are limited since they do not have built-in storage structure and query language to manipulate ontologies.

In the last decade, several approaches and systems have been proposed in order to store in the same database data and ontologies describing their meanings. These approaches introduce the notion of ontology based databases (OBDB) and some of these database architectures support meta-modeling facilities which are required in our approach to have a specific ontology model for ontology of services. Among them, we have chosen to use the OntoDB [4] OBDB equipped with the OntoQL [7] exploitation language that has been developed and is available in our laboratory.

In this section, we first present OntoDB and OntoQL. Then, we show the extension of OntoDB we have done to manage ontologies of services. Finally, we show interesting queries for the management of services and workflows enabled by our work.

5.1 OntoDB: The Semantic Repository

OntoDB is composed of four parts (see Figure 11). Part 1 (meta-base) and part 2 (ontology instances) are the traditional parts available in all database management systems. They respectively contain instance data and the system catalog. Part 3 (ontology) and part 4 (meta-schema) are specific to OntoDB. The ontology part allows the system to represent ontologies in the database. The meta-schema of part 4 records the ontology model into a reflexive meta-model. It contains two main building components: *ENTITY* and *ATTRIBUTE* that respectively store the entities of the ontology model (e.g. classes, properties, etc.) and attributes of the entities (e.g. names, labels, comments, etc.). For the ontology part, the meta-schema plays the same role as the one played by the system catalog in traditional databases. But, contrary to a system catalog of a DB, this part can also be enriched to define new entities

المنارة للاستشارات
www.manaraa.com

describing ontology model extensions. The interest of the OntoDB architecture lies in this capability of extending the ontology model. Indeed, the ontology model can be manipulated, and thus extended, by manipulating the meta-schema through the OntoQL querying language.

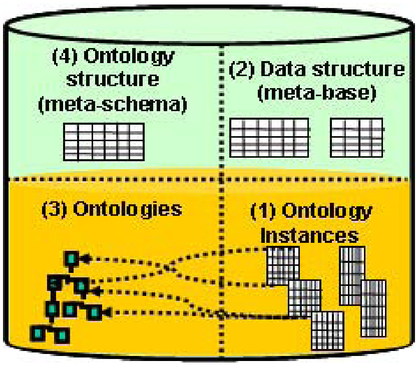

(4) Ontology structure (meta-schema)
(2) Data structure (meta-base)
(3) Ontologies
(1) Ontology Instances

Figure 11: OntoDB

5.2 OntoQL: The Querying Language

The OntoQL language is the exploitation language defined for managing and querying databases that embed meta-schemas like the OntoDB database. Regarding parts 1 and 2, OntoQL behaves like the traditional SQL language. But, it also allows a user to query the ontology of part 3 and its instances stored in part 2. Finally, OntoQL also supports the exploitation of part 4. In that case, all the manipulated elements are prefixed by the # symbol.

To illustrate OntoQL, let's see several examples. The following statement creation an ontology concept:

```
CREATE #Class Person (

DESCRIPTOR (#name[FR]= 'Personne')
PROPERTIES (uri STRING, name STRING));
```

In this example, we have created the ontology concept *Person* as a class (#class) with the properties *uri* and name. Real world persons can be added as instances to this concept as shown in the following example:

```
INSERT INTO Person (uri, name) VALUES (1729, Belaid);
```

As previously stated, OntoDB is based on a core ontology model (part 4) that is extendable through OntoQL statements. As an example, let's consider the following OntoQL statement:

```
CREATE ENTITY #OWLAllValuesFrom UNDER #Class (
#onProperty REF(#Property),
#allValuesFrom REF(#Class) )
```

The previous OntoQL statement adds the new entity *OWLAllValuesFrom* (a constructor of the OWL ontology model) to the core ontology model as a specialization of the Class entity. It has two attributes *onProperty* and *allValuesFrom*.

The classical architecture of the OntoDB ontology model supports the description, indexation and storage of domain ontologies and their instances. It needs to be extended in order to support the description, indexation and storage of semantic services and the WS they index. Figure 12 shows the main elements of the extended ontology model to handle ontologies of services.

5.3 OntoDB Extension for Managing Semantic Services

The classical architecture of the OntoDB ontology model supports the description, indexation and storage of domain ontologies and their instances. It needs to be extended in order to support the description, indexation and storage of semantic services and the WS they index. Figure 12 shows the main elements of the extended ontology model to handle ontologies of services.

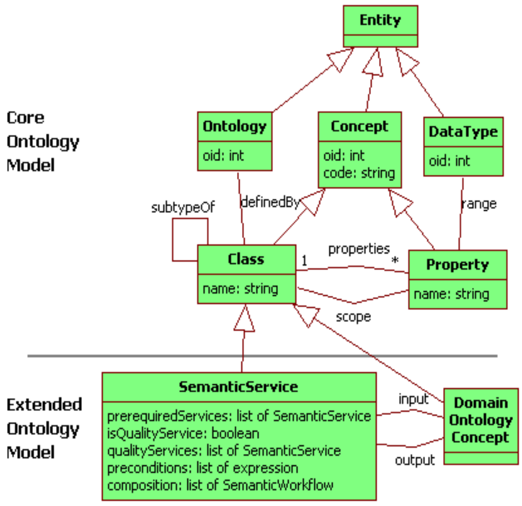


Figure 12: Extract of the extended ontology model for handling semantic services

The SemanticService entity has been added to create semantic services (concepts of ontologies of services). The creation of this entity has been done with the following OntoQL statement:

```
CREATE ENTITY #SemanticService UNDER #Class (
      #inputs REF (#DomainOntologyConcept) Array,
      # inputs REF (#DomainOntologyConcept) Array,
      #prerequiredServ REF (#SemanticService) Array,
      #qualityServices REF (#SemanticService) Array,
      #preconditions REF (#Expression) Array,
      #composition REF (#SemanticWorkflow) Array)
```

A semantic service is a specific class of an ontology of service. Thus SemanticService is defined as a sub entity of Class (UNDER #Class). Compared to a class a semantic service has new attributes such

المنارة للاستشارات
www.manaraa.com

its input (#inputs), outputs (#inputs) or previous services (#prerequiredServ).

5.4 The Creation of Ontologies of Services

With the ontology model extension we have described in the previous section, OntoDB supports the definition of ontologies of services. As an example, we have created the ontology of services that characterizes the seismic interpretation services and workflows shown in Figure 6. This ontology of services is created by an OntoQL script in OntoDB. Let’s see one statement of this script:

```
CREATE #SemanticService ReflectorExtraction
UNDER Extraction(
DESCRIPTOR (
#input = ARRAY(
SELECT doc.#oid
FROM #DomainOntologyConcept doc
WHERE doc.#name='SeismicCube')
#output = ARRAY(
SELECT doc.#oid
FROM #DomainOntologyConcept doc
WHERE doc.#name='Reflector'))
```

This OntoQL statement creates the *ReflectorExtraction* semantic service. *ReflectorExtraction* is defined as a specialization of Extraction (UNDER Extraction). Thus, this semantic service inherits the properties of its super semantic service and thus, the properties of the root semantic service. It has as input (#input) the ontology concept *SeismicCube* and as output (#output) the ontology concept Reflector. These concepts are specified using OntoQL queries.

Once a concept has been defined, a table needs to be created to store its instances. The following statement creates a table corresponding to the semantic service *ReflectorExtraction* with a sub-set of its properties as column.

CREATE EXTENT OF *ReflectorExtraction*
(URI, description, URL, *inputsModels, outputsModels*)

This statement creates a table to store instances of *ReflectorExtraction*. The columns of this table correspond to the properties URI, description, URL, *inputsModels* and *outputModels*.

5.5 The Web Services

Once the semantic services have been created, they can be used to index WS and worflows. In our approach, this indexation is done using the instanciation mechanism. Thus, a WS or workflow instance of a semantic service is indexed by this service. The following OntoQL statement shows the indexation of the WS *refGet* by the semantic services *ReflectorExtraction*:

```
INSERT INTO ReflectorExtraction
(URI, description, URL, inputsModels, outputsModels)
VALUES ('uri2', 'WSDL', 'http://lisi/refGet.wsdl',
(SELECT   oid   FROM   GeoConcept   WHERE   URL
= '/ifp.fr/SEG-Y.txt'),
(SELECT   oid   FROM   GeoConcept   WHERE   URL
= '/ifp.fr/XYZFile.txt'))
```

The WSDL WS refGet is created as an instance of the ReflectorExtraction semantic service to index it. Its input (inputModels) and output (#outputsModels) are defined by OntoQL queries searching for corresponding formats (SEG-Y and XYZ File) in the stored data model (instances of GeoConcept).

5.6 Usage of the Repository

The interest of the defined semantic repository is the capability to support queries that exploit the semantic indexations and in particular that retrieves WS and workflows indexed by semantic services. The following queries are representative examples of OntoQL queries that support the WS and workflows discovery process based on their semantic descriptions.

Query 1: retrieve the WS and workflows that perform an extraction of reflectors and that take as input the data model *SEG-Y File.*

Classical Approaches. It is only possible to retrieve the WS and workflows that reference a given data model. Indeed, the semantic function performed by the service is not described.

The Corresponding OntoQL Query:

```
SELECT RES.uri, RES.url
FROM ReflectorExtraction RES, GeoConcept GC
WHERE GC.oid IN RES.inputsModels
AND GC.name = 'SEG-Y File'
```

This query retrieves the WS indexed by the ReflectorExtraction semantic service. Its input (inputsModels) is an instance of a data model and thus of GeoConcept.

Query 2: retrieve the semantic services (functions) that follow the *ReflectorExtraction* semantic service:

Classical Approaches. It is not possible to find a successor for a service in classical approaches since it is not defined explicitly and not agreed on. We rely on ontologies of services to address this issue.

The Corresponding OntoQL Query:

```
SELECT SS2.#name
FROM #SemanticService SS1, #SemanticService SS2
WHERE SS1.#name = 'ReflectorExtraction'
AND SS1.#oid IN SS2.#prerequiredServices
```

In this query, SS1 represents the ReflectorExtraction semantic service and SS2 is used

www.manaraa.com

to retrieve the semantic service following this WS, i.e. the semantic services that have ReflectorExtraction as a previous service (`prerequieredServices`).

Query 3: retrieve the services that perform an equivalent function as the one performed by the WS or workflow whose uri is *uri15*. Notice that this query provides a solution for services adaptability.

Classical Approaches. In classical approaches, services are not linked to each other according to the functions they perform. Thus, it is not possible to find a service performing a similar task. Furthermore, it is not possible to replace a service by another one performing a more general or precise function because no hierarchical relationship is defined between services.

The Corresponding OntoQL Query:

```
SELECT SS.#name, S1.uri, S1.url
FROM GeoService S1, GeoService S2, #SemanticService SS
WHERE (TYPEOF(S1).#oid = TYPEOF(S2).#oid)
AND (S2.uri = 'uri15')
AND NOT (S1.oid = S2.oid)
AND (SS.#oid = TYPEOF(S2).#oid)
```

In this query `S2` represents the `uri15` service or workflow and `S1` the services that perform an equivalent function as `S2`. If they perform an equivalent function, `S1` and `S2` must be indexed by the same semantic service represented by `SS`. The `TYPEOF` function is used to check that `S1` and `S2` are instances (and thus indexed) by `SS`.

As this section shows, our approach has been fully implemented using the OntoDB database and the OntoQL language. Using this language all the data can be created in the database and interesting queries can be executed to retrieve WS and worflows from their semantic descriptions. However, writing such queries and statements requires an IT technical knowledge. To address this problem we have developed a specific tool to support our approach hiding to the geologists all the IT specific manipulations.

6. THE GEOLOGICAL WORKFLOW EDITOR (GWE)

The GWE is a software we have developed using the Eclipse environment to support the approach we propose and in particular for supporting the geological modeling task. It is connected to the OntoDB database which is used as a semantic repository for data models, services, workflows and their semantic descriptions. This database connection is enabled through the API JOBDBC and OntoAPI [7]. They extend the standard JDBC API in order to execute OntoQL queries over an OntoDB database from Java programs. The objective of the GWE is to enable users with little technical knowledge to adopt our approach thanks to its graphical user interface that hides the complexity of OntoQL queries.

To support the use cases of geologists we have identified *profiles* that are attached to particular tasks:

1. *domain experts*: they discuss and define the ontologies of domain and services;
2. *technical experts*: they index existing and new developed WS, workflows and data models by their ontological characterizations. They contribute to the creation of workflows from semantic ones when the generation is not fully automatable;
3. *managers of semantic workflows*: complementary to the domain experts, they design new processes (semantic workflows) and with the help of domain experts, they enrich the ontology of services. Based on the definition of semantic workflows, technical experts create IT workflows;
4. *final users*: they search for the semantic services corresponding to the function they want to perform. Then, they select a WS or workflow and execute it.

We present in this section two scenarios that show the improvement brought by our approach to the management of IT services and workflows.

6.1 Reusing Existing Services and Workflows

Geological modeling services and workflows are specified by actors and experts of geological modeling and created by IT developers. However, the methodology, the initial goals and the actual performed tasks are not explicitly stated in their syntactic descriptions. As a consequence, geologists and other actors of geological modeling have difficulties to re-use existing IT services and workflows.

The GWE enables the semantic indexation of already developed services and workflows through semantic services to facilitate their management (discovery, adaptability, etc.). For this purpose, ontologies of services in geological modeling have to be created by experts in this field. They make an inventory of the performable functions, their mutual relationships and the geological concepts they manipulate in order to design the target ontologies of services. Then, these ontologies are represented in the GWE (Figure 13 (1)) through its graphical interface and are automatically stored in the OntoDB semantic repository. The obtained ontologies can be enriched continuously. The semantic services that compose them are used to index WS and workflows. This indexing activity has to be performed by a service and workflow management system administrator (Figure 13 (2)). The obtained indexes are also automatically stored in OntoDB and used to help the

المنارة للاستشارات
www.manaraa.com

end-user geologists in their management such as discovering the IT services and workflows that meet their requirements and execute them (Figure 13 (3, 4 and 5)).

We now show the different zones of the GWE involved in this scenario in Figure 14:

- In zone (A) of the figure, a view of the domain and services ontology is displayed;

- In zone (B), data models, WS and workflows indexed by the ontology concepts and the semantic services are displayed;

- In zone (C), an editor supports the creation of new ontology concepts and semantic services in order to enrich the domain and services ontologies, and enable the indexation of data models, WS and workflows by ontology concepts and semantic services;

- In zone (D), the properties of the syntactic descriptions are displayed.

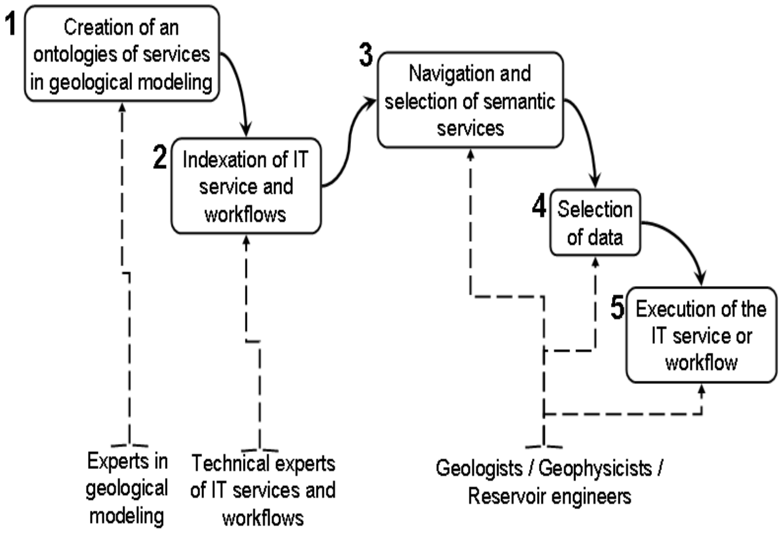

1
Creation of an ontologies of services in geological modeling
2
Indexation of IT service and workflows
3
Navigation and selection of semantic services
4
Selection of data
5
Execution of the IT service or workflow
Experts in geological modeling
Technical experts of IT services and workflows
Geologists / Geophysicists / Reservoir engineers

Figure 13: Reusing existing WS and workflows

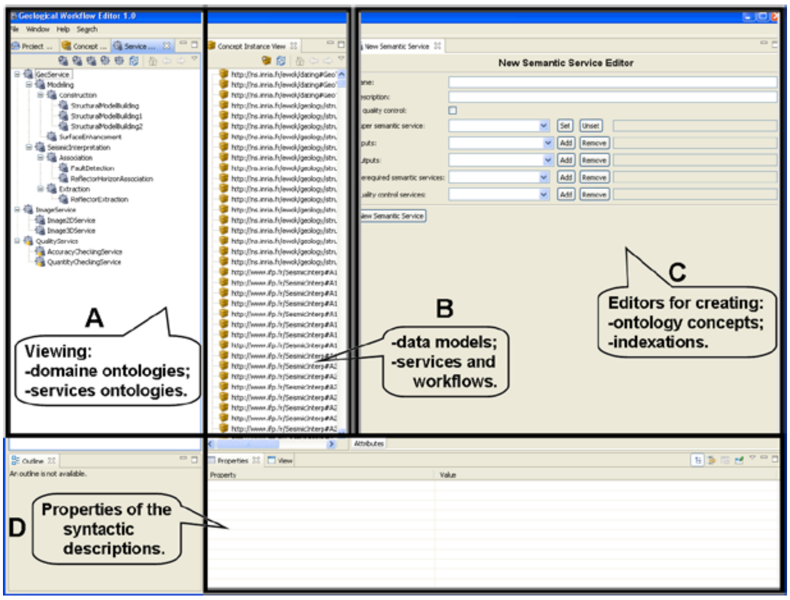


Figure 14: A view of the GWE for indexing existing WS and workflows

6.2 Building New Workflows

As an example of a task enabled by our approach we have previously described the creation of new workflows. Based on the creation of ontologies of services and the indexation of IT services and workflows, we show in this section how this task is supported by the GWE.

The creation of new workflows requires knowledge about the precise performed functions of the services and sub-workflows to be composed. It requires also technical knowledge on these services and sub-workflows. Thanks to the semantic indexation of existing WS and workflows developed in the geological modeling field, it becomes possible

المنارة للاستشارات
www.manaraa.com

to overstep these limitations by designing semantic workflows first and then creating IT workflows. First, the GWE is used to design semantic workflows by composing semantic services in a similar way to the one used to design classical workflows. This activity is performed by experts in geological modeling workflows (Figure 15 (1)). They are subject to a partial validation such as checking that the order of semantic services is coherent through the attribute *prerequiredServices* for instance (Figure 15 (2)). Then, the obtained semantic workflow is used to enrich the ontology of services either by creating a new semantic service or by modifying the definition of one of the semantic services (Figure 15 (3)). In the next step, a technical expert in services and workflows selects an IT service or a workflow for each semantic service composing the global semantic workflow (Figure 15 (4)). Although the semantic services are interoperable in the global semantic workflows, the selected IT service may not compose. In this case, adapting IT services are searched and used if found to ensure the interoperability. Finally the last step is the design of a new IT workflow (Figure 15 (5)) that can be executed and/or indexed (Figure 15 (6)).

We now show the different zones of the GWE involved in this scenario in Figure 16:

- In zone (A) of the figure, a view of the ontology of services is displayed;
- In zone (B), WS and workflows indexed by semantic services are displayed;
- In zone (C), an editor supports the creation of semantic workflows by composing semantic services;
- In zone (D), a tree view of the designed semantic workflow is displayed (Figure 17 zooms on the tree view);

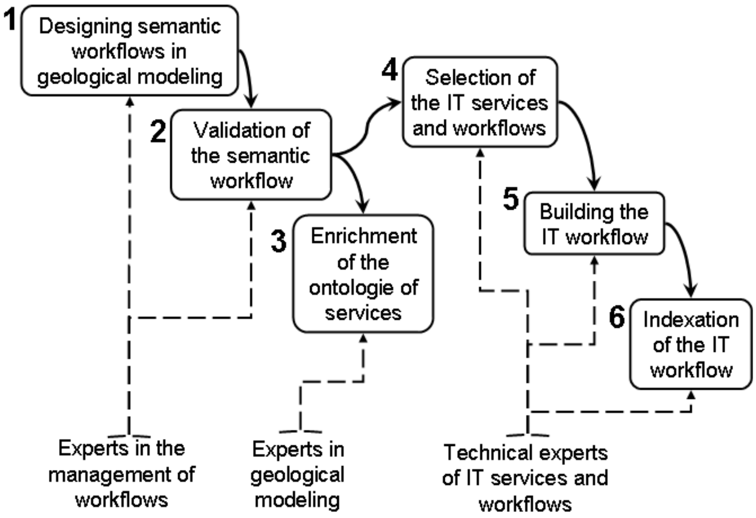

1
Designing semantic workflows in geological modeling
2
Validation of the semantic workflow
3
Enrichment of the ontologie of services
4
Selection of the IT services and workflows
5
Building the IT workflow
6
Indexation of the IT workflow
Experts in the management of workflows
Experts in geological modeling
Technical experts of IT services and workflows

Figure 15: Creating new IT workflows

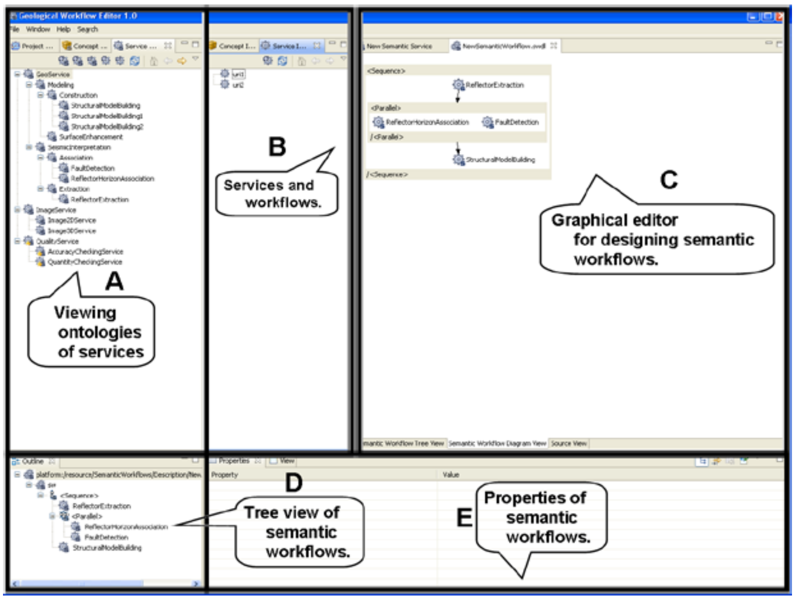
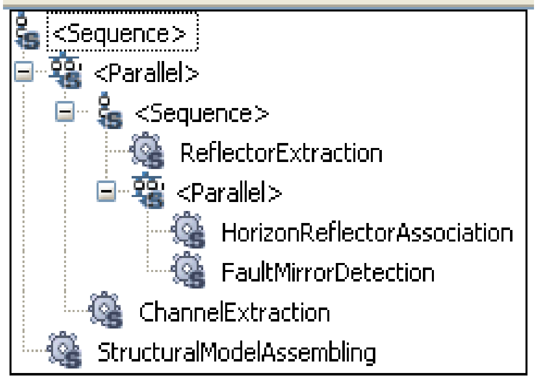

Geological Workflow Editor 1.0
A
Viewing ontologies of services
B
Services and workflows.
C
Graphical editor for designing semantic workflows.
D
Tree view of semantic workflows.
E
Properties of semantic workflows.
<Sequence>
ReflectorExtraction
<Parallel>
ReflectorHorizonAssociation
FaultDetection
</Parallel>
StructuralModelBuilding
</Sequence>

Figure 16: A view of the GWE for creating new workflows

www.manaraa.com

- In zone (E) of Figure 16, the properties of the designed semantic workflows are displayed.

This section has shown how we have applied our approach to the geological modeling field using the GWE tool we have developed. Let's now summarize the main contributions of our proposition and see outlooks of this work.



```
<Sequence>
<Parallel>
    <Sequence>
        ReflectorExtraction
        <Parallel>
            HorizonReflectorAssociation
            FaultMirrorDetection
    ChannelExtraction
StructuralModelAssembling
```

Figure 17: Extract of the GWE for a tree view of a semantic workflow

7. CONCLUSION AND OUTLOOKS

WS and workflows assist engineers to achieve tasks such as data modeling, processing and storing. However, current descriptions of WS and workflows are mostly syntactic which leads to (1) the need of IT technical knowledge to manipulate them and (2) imprecise results for complex management tasks such as composition or adaptability of WS. As a solution, several approaches have proposed to enrich semantically the description of WS using ontologies. However, these approaches propose either annotations of the existing description language or extensions of these languages. Thus, they do not clearly separate the syntactic description of WS from their semantic description. As a consequence, existing services and tools have to be adapted to these approaches. Moreover these approaches do not provide a classification of services according to the performed functions leading to difficulties to solve problems such as the adaptability of WS.

Our approach aims at solving the previous problems by providing another abstraction level independent of the syntactic descriptions of WS and workflows. This abstraction level is composed of ontologies of services that provide hierarchical characterization of semantic services. This hierarchy uses a subsumption relationship based on the covariance semantics. Existing WS and workflows can be indexed by semantic services using the instantiation mechanism. We have shown that this semantic description of WS and workflow is useful for complex management tasks such as the creation of new workflows starting from existing WS and workflows.

Our approach has been implemented on the OntoDB database using OntoQL, its associated language. We have shown that this implementation supports the definition of interesting queries related to the complex management of WS and workflows. Finally, to help users with limited technical knowledge to manage WS and workflows, we have developed a tool named the Geological Workflow Editors (GWE). It has been customized and applied to the geological domain on a real case study.

For future work, we plan first to extend the model of ontologies of services to support more functional properties such as conditional required services and non functional properties such as performance or security of the function. In the same way we need to extend the indexation part with non functional requirements such as preferences, the availability or the quality of services. Another interesting perspective is to support the generation of executable workflows from semantic workflows. From the GWE point of view, we intend to identify other templates of queries needed to perform geologist's tasks. We are also currently scaling up our approach to a high number of WS and workflows descriptions in order to evaluate the access time. Finally, we are interested in connecting the GWE to orchestrators for supporting the execution of heterogeneous WS and workflows descriptions.

REFERENCES

1. Ait-Ameur, Y., 2009, "A semantic repository for adaptative services," *IEEE Congress on Services, Part I (SERVICES'09), IEEE Computer Society*, pp. 211-218.
2. Belaid, N., Ait-Ameur, Y. and Rainaud, J. F., 2009, "A semantic handling of geological modeling workflows," *International ACM Conference on Management of Emergent Digital EcoSystems (MEDES'09)*, pp. 83-90.
3. Bentahar, J., Maamar, Z, Wan, W., Benslimane, D., Thiran, P. and Subramanian, S., 2008, "Agent-based communities of web services: an argumentation-driven approach," *Journal of Service Oriented Computing and Applications*, Vol. 2, No. 4: pp. 219-238.
4. Castagna, G., Ghelli, G. and Longo, G., 1995, "A calculus for overloaded functions with subtyping," *Journal of Information Computation*, Vol. 117, No. 1, pp. 115-135.
5. Dehainsala, H., Pierra, G. and Bellatreche, L., 2007, "An ontology-based database for data intensive applications," *In the 12th International Conference on Database Systems for Advanced Applications (DASFAA'07)*, pp. 497-508.

المنارة للاستشارات
www.manaraa.com

6. Deelman, E., Singh, G., Su M.-H., M., Blythe, J., Gil, A., Kesselman, C., Mehta, G., Vahi, K., Berriman, G. B., Good, J., Laity, A. and Jacob, J. C., 2005, "Pegasus: A framework for mapping complex scientific workflows onto distributed systems," *Journal of Scientific Programming*, Vol. 13, No. 3, pp. 219-237.
7. Farrell, J. and Lausen, H., 2007, "Semantic annotations for WSDL and XML schema (SAWSDL)," *W3C Recommendation.*
8. Jean, S., Pierra, G. and Ait-Ameur, Y., 2006, "Querying ontology based database using OntoQL," *International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE'06),* pp. 704-721.
9. Jordan, D. and Evdemon, J., 2007. "WS-BPEL 2.0," *Technical report, OASIS Standard.*
10. Maamar, Z., sheng, Q. Z., Tata, S., Benslimane, D. and Sellami, M., 2009, "Towards an approach to sustain Web services high-availability using communities of Web services," *International Journal of Web Information Systems*, Vol. 5, No. 1, pp. 32-55.
11. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N. and Sycara, K., 2004, "OWL-S: Semantic markup for Web services," *W3C Recommendation*.
12. Mastella, L. S., Ait-Ameur, Y., Perrin, M. and Rainaud, J. F., 2008, "Ontology-based model annotation of heterogeneous geological representations," *The 4th International Conference on Web Information Systems and Technologies (WEBIST'11)*, pp. 290-283
13. Paolucci, M., Kawamura, T., Payne, T. R. and Sycara, K. P., 2002, "Importing the semantic web in UDDI," *Web Services, E-Business, and the Semantic Web (WES'02), International Conference on Advanced Information Systems Engineering (CAiSE'02), Lecture Notes in Computer Science*, pp. 225-236.
14. Peer, J., 2005, "SEmantic service and MArkup (SESMA)," *Web Service Semantics Workshop (WSS'05) at the 14th International World Wide Web Conference (WWW'05)*, pp. 100-116.
15. Pinheiro Da Silva, P., Salayandia, L., Gándara, A. and Gates, A. Q., 2009, "CI-Miner: semantically enhancing scientific processes," *Journal of Earth Science Informatics*, Vol. 2, No. 4, pp. 249-269.
16. Pinheiro Da Silva, P., Salayandia, L., and Gates, A. Q., 2007, "Using WDO-It to build a geoscience ontology," *In Geoinformatics*, pp. 54-56.
17. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C. and Fensel, D., 2005. "WSMO," *Journal of Applied Ontology*, Vol. 1, No. 1, pp. 77-106.
18. Verney, P., Perrin, M., Thonnat, M. and Rainaud, J. F., 2008, "An approach to seismic interpretation based on cognitive vision," *In the 70th European Association of Geoscientists & Engineers Conference (EAGE'08).*
19. W3C, 2004, *Web Service Definition Language (WSDL 1.1), Technical Report.*
20. Workflow Management Coalition, 2008, "XML process definition language (XPDL)," *Workflow Management Coalition.*

ABOUT THE AUTHORS

Nabil Belaid is a Ph.D student in the data engineering research team of the laboratory of computing (LISI) at the National School of Engineers in Mechanics and Aeronautics (ENSMA). He has worked for both the IFP Energies Nouvelles (IFPEN) and the LISI/ENSMA on the semantic indexation of workflows through ontologies of services and has developed the GWE software that intends to help geologists in their management of geological modeling workflows.

Stephane Jean is an assistant professor at the University of Poitiers. He is a member of the data engineering research team of the laboratory of computing (LISI) at the National School of Engineers in Mechanics and Aeronautics (ENSMA). He conducts research in the ontologies management area especially in the context of databases. During his Ph.D thesis he has designed the OntoQL exploitation language for databases that store ontologies and the data they describe. This language is currently used in several projects in the engineering area.

Yamine Ait-Ameur is full professor at the National Poltechnique Institue in Toulouse. Formal methods, ontology based data modeling and database are his main topics of interest. He has written several research papers and supervised several Ph.D thesis. His research results have been applied in various areas like aeronautics, embedded systems and petroleum industry.

Jean-François Rainaud is a research project manager at the IFP Energies Nouvelles (IFPEN) in France. He received his Ph.D degree at UMPC Paris VI University (France) in 1979. After twenty years in the industry (Beicip Franlab), he joined IFPEN to develop knowledge based geological modeling research projects. He is also teaching IT project management at the University of Poitiers (France). His research interests are geological modeling, ontology based knowledge modeling and workflow.

(Received September 2010, revised January 2011, accepted April 2011)

المنارة للاستشارات

www.manaraa.com

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

www.manaraa.com