

TOWARD A SEMANTIC MANAGEMENT OF GEOLOGICAL MODELING WORKFLOWS

Nabil Belaid^{1,2}, Yamine Ait-Ameur², Stéphane Jean² and Jean-François Rainaud¹

¹*IFP Energies nouvelles/R114, 1 & 4 av. de Bois-Préau, 92852 Rueil-Malmaison, France*

²*LISI/ENSMA-UP, 1 av. Clément Ader, 86961 Futuroscope, France*

Keywords: Semantic workflow, Semantic service, Ontology, Geological modeling, Meta-model.

Abstract: Today, the petroleum industry demonstrates a greater concern toward ecology. Actions are undertaken in order to reduce the CO₂ emissions in particular. Storing the CO₂ in depleted petroleum reservoirs represents one of the ways of controlling the CO₂ emissions. However, a prior modeling of the reservoirs has to be done for a secure storage. This geological modeling task follows series of complex workflows (business processes) of data processing services. These workflows are built without any methodological rule. It is difficult for geologists to execute workflows or even services that they have not designed. Moreover, it is not possible to build new workflows without having a precise knowledge on the compounding services and sub-workflows. We claim that in order to solve the two stated problems, explicit semantics has to be applied to describe the workflows and the services that compose them.

In this article, we first explain how geologists operate today. Then, we show how we enrich geological modeling workflows, the services that compose them and the data they manipulate with semantic indexations through ontology-based characterizations (Geological Data and Services Ontologies). Finally, we explain how our approach supports reuse of existing workflows and build new ones.

1 INTRODUCTION

In the petroleum industry, controlling the CO₂ emissions has become a serious concern. One of the solutions is to re-inject the produced CO₂ in depleted petroleum reservoirs. However, a prior modeling of the target reservoirs is necessary for a secure storage.

In order to model the target reservoirs, data which are obtained thanks to measure campaigns have to be analyzed in order to build the geological structures of the target petroleum reservoir. For example, images on reservoirs, called seismic cubes, can be acquired by seismic explorations. They can be interpreted by geophysicists and geologists so that geological structures can be identified. Several other types of data representations are defined to store and exchange the huge amount of created data, from which we quote the seismic, the geological and the structural information. These data are then processed leading to a fine reservoir description. This makes it ultimately possible to realize a precise CO₂ flow simulation forecast in a target reservoir and answer to the question "can we store securely CO₂ in the given reservoir?".

This whole reservoir modeling task consists in

complex workflows¹ or processes of services. These workflows, their composing services, their compositions into more complex workflows are not formally modeled in this exploration and production part of petroleum industry. They are built without any methodological rule, executed and stored without attaching to them any information regarding what they actually do. Consequently, no resource, except their names may give information about their meanings and their roles. Indeed, what would "workflow21_geoReflect_15042009.wsdl" mean for a geologist? It is then problematic for geologists to execute services or workflows they have not created because no semantics is explicitly attached to them. Moreover, it is not possible to build new workflows from scratch without having a precise knowledge of how to compose their compounding processing services. We claim that associating semantic descriptions to services and workflows will help the geologists in their daily work.

In order to achieve these objectives, we have established an approach based on the definition of on-

¹Throughout this article, we designate the atomic data processing services or software by *services* and the composition of data processing services by *workflows*.

tologies². These ontologies, that we have contributed to build, characterize the services, the workflows and the data they consume and produce in the geological modeling field. Then, we have enriched the services, the workflows and the data with a semantic layer by linking them to their ontological characterizations. This leads to, on the one hand, the easy retrieval of existing services and workflows, and on the other hand, the building of new workflows.

The remainder of this article is structured as follows. Section 2 describes a case study in geological modeling: the seismic interpretation workflows. We present, in section 3, our ontology-based approach describing services and workflows semantics. Then, in section 4, we outline the description of the ontology-based database OntoDB which we use as a semantic repository to store the services, the workflows and their ontological descriptions. We show, in section 5, the implementation of the case study in OntoDB. In section 6, we describe the *Semantic Workflow* concept that allows building new workflows. Finally, we conclude and discuss some perspectives.

2 SEISMIC INTERPRETATION

The goal of the seismic interpretation is to build a structural model from basic geological objects. It is a complex process and can be modeled by multiple complex workflows. Figure 1 shows a candidate seismic interpretation workflow³ as it is modeled by experts in geology.

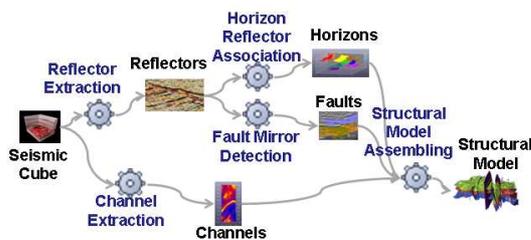


Figure 1: Simplified fragment of a potential seismic interpretation workflow.

In this figure, we start from a *SeismicCube* which represents the seismic images of a given reservoir. Then, we apply a *Reflector Extraction* in order to extract the *Reflectors* which are small visible parts of geological elements. Two services *Horizon Reflector Association* and *Fault Mirror Detection* are per-

²An ontology is a *formal* and *shared* representation of the knowledge of a given domain by a set of concepts and the relationships between those concepts.

³This workflow has been simplified for illustration.

formed in parallel to obtain respectively *Horizons* and *Faults* etc.

The choice for this case study is motivated by the fact that it defines a complex chain involving different compositions of services. Moreover, precise information on this part through the work of (Verney et al., 2008) can be accessed.

3 FOR A NEW MANAGEMENT OF WORKFLOWS

The main idea behind our approach (Ait-Ameur, 2009) is to create ontologies in order to index semantically existing workflows and services descriptions and the engineering models they reference. Here lays the difference with the other approaches that start from specific existing workflows and services description languages such as ontology-based approaches (OWL-S (Martin et al., 2004), WSMO (Roman et al., 2005) etc.) that intend to define the semantics of the terms related to the services and workflows or the annotation-based approaches (SAWSDL (Farrell and Lausen, 2007), WSDL-S (Miller et al., 2004) etc.) that intend to enrich services and workflows descriptions. Our approach allows to: (i) store heterogeneous descriptions of workflows and services in a repository and (ii) add a semantic layer through ontologies in order to bring semantics to these stored workflows and services and enable their complex semantic management.

First, consider the two horizontal layers (a) and (b) of Figure 2. The principle consists in building models that represent the data and the services and workflows (Figure 2.(a)). Then, create the ontologies that characterize them (Figure 2.(b)). Finally, link the models to the ontologies level through indexations.

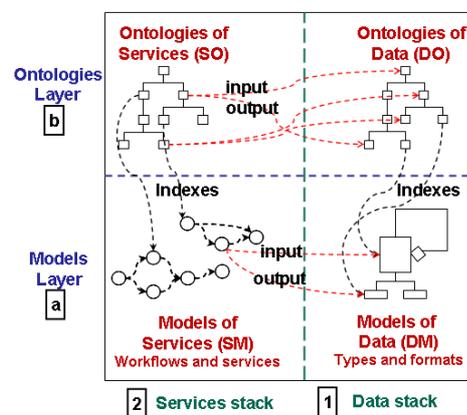


Figure 2: An approach for semantic handling of workflows.

3.1 Data Stack

Engineering Models. Data are instances of engineering models whose types or formats are usually defined in XML-Schemas. For example, integer and string are simple data models and formatted files such as *SEG-Y* (representing seismic images) are complex engineering models.

Domain Ontology. A domain ontology gives explicit semantics to the concepts of a given domain. For example, there are ontologies that characterize the medical field or the mechanical parts. The domain ontology concepts index the engineering models enriching them semantically through an independent layer without overloading them.

3.2 Services Stack

Services Model. The service models are the descriptions of workflows and services. In our case study, we have the example of a Web service that extracts *Reflectors* (produces a *Hollow Matrix File* from a *SEG-Y* file).

Services Ontology. The ontologies of services are semantic descriptions of services and workflows in a given domain. They define hierarchies and compositions of services. A concept of an ontology of services is a *semantic service* and is independent of specific implementations. Figure 3 shows a fragment of the ontology that characterize the seismic interpretation services.

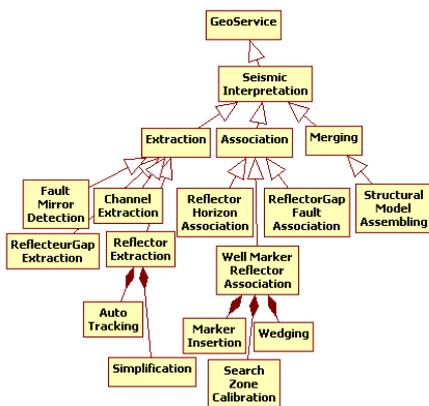


Figure 3: Fragment of the seismic interpretation services ontology and the referenced domain ontologies.

The ontology of services shown in Figure 3 has *GeoService* as a root concept and more specifically the concept *SeismicInterpretation*. From this concept,

three concepts are derived: *Extraction*, *Association* and *Merging* which respectively characterize the services for elements extractions, services that associate elements and services for elements merging etc.

The concepts of this ontology of services have properties that reference domain ontologies. For example, the concept *Reflector Extraction* has as input a *Seismic Cube*. Notice that if a semantic service subsumes a second semantic service, the input and output of the first one subsume respectively the input and the output of the second one (co-variance relationship). Moreover, these ontologies are different from other ontologies since they characterize particular elements which have input, preconditions etc. Thus, they are designed following a specific model of ontologies.

4 ONTODB: A SEMANTIC REPOSITORY FOR SERVICES

Our approach cuts through three modeling levels: (i) meta-models (model of ontologies), (ii) models and ontologies and (iii) instances; and two stacks: (1) data and (2) services. For a full exploitation of the proposed approach descriptive power, we need a database architecture that enables a semantic based storage and search. Classical database architectures are limited in the way that they neither enable meta-modeling representations nor offer a query language for querying simultaneously the different database levels.

Recently, several approaches and systems were proposed to store in the same database data and the ontologies describing their meanings. We call these databases, ontology-based databases (OBDB). OntoDB is one of these approaches (Dehainsala et al., 2007). It stores explicitly in the database not only the data, but also the conceptual model defining the structure of data and the ontology representing the semantics of data (see Figure 4).

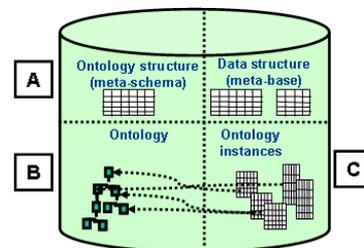


Figure 4: The OntoDB architecture.

Our approach requires the definition of the concept of *semantic service* not available in the classical ontology models. Since OntoDB allows accessing

and enriching the meta-model level, we have chosen this OBDB as a persistence model for our workflow descriptions. Moreover, OntoDB offers the possibility of querying the databases at the ontology level thanks to the query language OntoQL (Jean et al., 2008). This language is also used to create and/or manipulate concepts of meta-models (models of models or ontologies) using OntoQL statements (Figure 4.A) of the form:

CREATE ENTITY #[name of the concept of ontology model] ([list of attributes definitions]).

Example: **CREATE ENTITY** #Class (#definition STRING).

In the example, we enriched the ontology model by adding the attribute "#definition" to the entity "#class". Once the model of ontologies is created and enriched, it is possible to create ontologies concepts (valuating the new attribute) using OntoQL statements (Figure 4.B) of the form:

CREATE #[name of the concept of ontology model] (DESCRIPTOR [name of the ontology concept] PROPERTIES [list of attributes valuations] [list of properties definitions]).

Example: **CREATE** #Class Person (DESCRIPTOR (#definition='Humain being') PROPERTIES (uri STRING, name STRING)).

In the example, we have created the ontology concept "Person" with the properties "uri" and "name". It is finally possible to create instances of ontologies concepts using OntoQL statements (Figure 4.C) of the form:

INSERT INTO [name of the ontology concept] [list of properties to be valuated] VALUES [list of properties valuations].

Example: **INSERT INTO** Person (uri, name) VALUES (1728, Belaid).

In this example, the "Person" with the name "Belaid" is instantiated. An example of each type of these queries is given in next section and applied to our case study.

5 IMPLEMENTATION

We present the application of our approach to the seismic interpretation. The implementation of the data stack is outlined in the work of (Mastella et al., 2008). Here, we show how we have implemented the services stack. We recall that the services ontologies are particular characterizations. Thus, these ontologies are designed following a specific model of ontologies which we will describe with its implementation. Then, we present the implementation of the ontology of services shown in Figure 3 according to the specified model. Next, we show examples of services and workflows corresponding to a particular semantic ser-

vice and how the indexation is done. Finally, we explain how queries can be made on OntoDB in order to perform semantic searches.

5.1 Ontology Model of GeoServices

Semantic services are characterized by different attributes such as their inputs or their preconditions. These attributes are not available in usual ontology models as they are specific to services. As a consequence, we have specialized the class constructor of the usual ontology model to represent services. This extension is depicted in Figure 5 and is stored in the A part OntoDB (Figure 4.A).

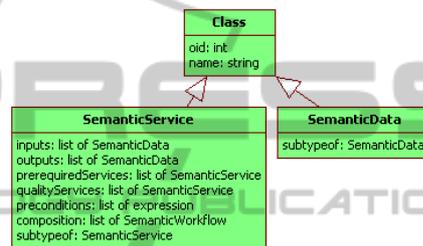


Figure 5: Fragment of the model of services ontologies.

The following statement implements this extension on OntoDB:

```
CREATE ENTITY #SemanticService UNDER #Class (
  #input REF (#SemanticData) Array,
  ...
  #subtypeof REF (#SemanticService))
```

We enrich the ontology model by creating a new entity that enables the definition of semantic services and their attributes.

5.2 Ontologies of GeoServices

The characterization of the services and workflows in the seismic interpretation field leads us to the ontology of services shown in Figure 3. The semantic services that compose the ontology of services is stored in the B part OntoDB (Figure 4.B). The following OntoQL statement represents an example of the creation of a semantic service *Reflector Extraction* as a concept of the ontology of services:

```
CREATE #SemanticService ReflectorExtraction (
  DESCRIPTOR ( #input = 'SeismicCube',#output = 'Reflector')
  PROPERTIES ( uri STRING,description STRING,url STRING,
inputM DataType,outputM DataType)
```

We create a new semantic service *ReflectorExtraction* and valuate its attributes. We also attach to it

some properties such as its description language, the input data type etc.

5.3 Geological Modeling Services

The services and workflows are indexed by defining them as instances of semantic services. They are stored in the C part OntoDB (Figure 4.C). Table 1 shows an example of the definition of some services and workflows corresponding to the semantic service *Reflector Extraction*.

Table 1: Some services and workflows descriptions.

uri	descr.	url	inputM	outputM
uri1	BPEL	../RMark.bpel	SEG-Y	WITSML
uri2	WSDL	../getRef.wsdl	SEG-Y	XYZFile
...

Each service is described by values of the properties defined on the semantic service it belongs to. As an example, the service *getRef* in the second row of Table 1 is a Web service described in WSDL and located in *../getRef.wsdl*. It takes as input a *SEG-Y File* and produces an *XYZ File* as output. The following OntoQL statement represents the indexation of this service by instantiating the semantic service *Reflector Extraction*.

```
INSERT INTO ReflectorExtraction (uri, description, url, inputM, outputM)
VALUES ('uri1', 'WSDL', '../getRef.wsdl', SEG-Y Format Type, XYZFile
Format Type)
```

We index a processing service by instantiating the semantic service *ReflectorExtraction* and valuating its properties. The service *getRef* is described in WSDL, takes as an input data file a *SEG-Y File* etc.

5.4 A Semantic Repository for the Handling of Geological Services

Our implementation consists in extending the ontology model definition and then storing services, workflow and data in a single repository: OntoDB. Storing all these data in a single repository enables complex semantic queries. For example, the following OntoQL query retrieves services that perform an "Extraction", and produces as output a file in an "XYZ File" format:

```
SELECT SE.uri, SE.url FROM Extraction SE
WHERE SE.outputM = Typeof(XYZFile Format Type)
```

This query uses the semantic service subsumption relationship. Indeed, services which are instances of "Extraction" such as "ReflectorExtraction" and "ReflectorGapExtraction" will be retrieved.

An interesting query consists in retrieving data processing services that may replace another one. For example, the following query retrieves the services that could replace the service whose uri is uri25 (GS2). This query consists in searching all services GS1 instances of the same semantic service as GS2 and that are not GS2:

```
SELECT GS1.uri, GS1.url FROM GeoService GS1, GeoService GS2
WHERE (TYPEOF(GS1).#oid = TYPEOF(GS2).#oid)
AND (GS2.uri LIKE 'uri25') AND NOT (GS1.oid = GS2.oid)
```

The complexity of the previous queries lies in the fact that they query the database at different levels (meta-model or model of ontologies), different stacks (data and services) and different conceptual representations (models and ontologies). These queries are hidden to the geologists thanks to a well adapted user interface.

6 SEMANTIC WORKFLOWS

Building semantic workflows can be done by composing semantic services. Such a task should be natural for "experts in the given domain" since semantic services represent explicitly the semantics of services and composing semantic services into semantic workflows is only a matter of "knowing" what is the order of execution of simple tasks in order to perform a more complex task.

In what follows, we show how we describe semantic workflows from semantic services. We then explain how a processing workflow can be derived from the constructed semantic workflow.

6.1 Description of Semantic Workflows

The definitions of semantic workflows require different operators to compose semantic services. Figure 6 presents the semantic workflow model we have established.

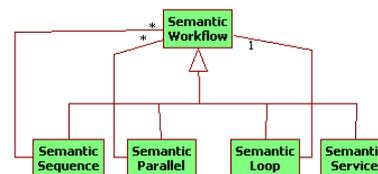


Figure 6: Description model of semantic workflows.

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 one semantic

workflow while a semantic sequence or parallel can be composed of multiple semantic workflows etc.

6.2 Building a Processing Workflow

When a semantic workflow has been built, it is possible to "deduce" a processing workflow by selecting a processing service for each semantic service that composes the semantic workflow. For example, starting from the semantic workflow described in Figure 1, Figure 7 represents a processing workflow obtained when selecting a processing service for each semantic service that composes the whole semantic workflow.

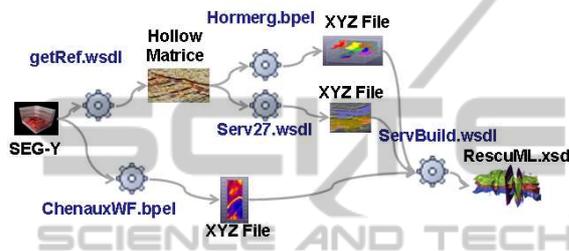


Figure 7: A potential seismic interpretation processing workflow.

In this figure, we start from a *SEG-Y* file which represents a 3D-image. Then, the service *getRef.wSDL* is executed to obtain a *Hollow Matrice* file. Next, the workflow *Hormerg.bpel* and the service *Serv27.wSDL* are performed in parallel to obtain two *XYZ Files* which represent respectively *Horizons* and *Faults* etc.

Although the semantic services of a semantic workflow compose, it may not be the same for the obtained processing workflow. Indeed, if in the previous example, the workflow *ChenauWF.bpel* had as output an *IJK File*, it would not be composable with the following service which take as an input an *XYZ File*. In this case, an adapting processing service that converts the *IJK File* to an *XYZ File* is searched in the database and is automatically added to the workflow.

7 CONCLUSIONS

Throughout this article, we have described an approach based on ontologies of services for the semantic handling of services and workflows. We have indexed the services, the workflows and the manipulated data through ontologies concepts. On the one hand, this semantic enrichment enables semantic queries over existing geological modeling services and workflows. On the other hand, it enables the design of new workflows based on semantic workflows first. This methodology is more natural for experts

in geology. This work represents a significant step toward the tool⁴ we are currently designing and that intends to assist geologists in searching, building and executing their geological modeling workflows.

In future work, we plan to identify templates of queries that are needed to perform geologists tasks. These queries will be parameterized, integrated in the graphical-user interface and thus hidden to geologists. We will also extend the definition of semantic services to take into account more detailed and crucial information when referencing services, particularly non functional requirements such as preferences, availability or quality of service.

REFERENCES

- Ait-Ameur, Y. (2009). A semantic repository for adaptative services. In *Int. conf. SHWS'09*.
- Dehainsala, H., Pierra, G., and Bellatreche, L. (2007). Ontodb: An ontology-based database for data intensive applications. In *DASFAA'07*.
- Farrell, J. and Lausen, H. (2007). *SawSDL*. Technical report, W3C Recommendation.
- Jean, S., Pierra, G., and Ait-Ameur, Y. (2008). Querying ontology based database using ontoql. In *ODBASE'06*.
- 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*.
- Mastella, L. S., Ait-Ameur, Y., Perrin, M., and Rainaud, J.-F. (2008). Ontology-based model annotation of heterogeneous geological representations. In *4th Int. Conf. WEBIST. Madeira, Portugal*.
- Miller, J., Verma, K., Rajasekaran, P., Sheth, A., Aggarwal, R., and Sivashanmugam, K. (2004). *WSDL-s*. <http://www.w3.org/Submission/WSDL-S/>.
- Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., and Fensel, D. (2005). *Wsmo*. *IOS Press*.
- Verney, P., Perrin, M., Thonnat, M., and Rainaud, J.-F. (2008). An approach to seismic interpretation based on cognitive vision. In *70th EAGE, Rome*.

⁴GWE (Geological Knowledge Editor)