Semantic Annotation of Oilfield Models

Laura S. Mastella

École des Mines de Paris / LISI-ENSMA Poitiers, France
laura.mastella@ensmp.fr

1 Research Problem

Engineering models are computer-based models used to run simulations over technical data. They are represented in logical models that are close to low-level computer representations, and the data is issued from different sources. In this context, integration task is a challenge. The classical solution for model integration in Software Engineering is to describe an upper-level meta-model and to define every datamodel as an instance of the upper meta-model. However, the models do not interoperate to each other, because there is no semantic added to their objects and the expression of model mappings is only hard coded. Therefore, in this work is to analyse how semantic-based techniques can be used to handle the interoperability problem of a specific engineering domain.

Several engineering domains rely on engineering models, such as civil engineering, aeronautics, environment or agriculture. Our work is conducted in the domain of petroleum reservoir engineering, in particular, the activity of *reservoir characterization*. Considering a typical reservoir characterization workflow, geoscientists rely on three-dimensional representations of the earth underground (called *reservoir models* or *oilfield models*) to take important decisions about oil-reservoir operations. The end-users of this community aim to be able to retrieve and re-use information that are created in the various areas of expertise within reservoir characterization and represented in diverse oilfields models. A recent W3C Use Case report from Chevron company presents a survey of the main applications of Semantic Web Technology for petroleum industry [1] and claims that "*in order to deal with the flood of information, as well as the heterogeneous data formats of the data* [in petroleum industry]*, we need a new approach for information search and access*".

The proposal of this work for addressing this issue is an approach based on *semantic annotation of engineering models*. We envisage the use of semantic annotation for: (*i*) making the expert knowledge explicit in the model and (*ii*) interrogating raw data using semantic concepts. Proceeding this way, end-users does not need to know the internal data structure of the model to search for information, they can use the semantic concepts from their area of expertise.

2 Background

At the moment, there are several frameworks and tools that allow to create semantic annotations over resources (web pages, textual documents, multimedia files). From a comparative analysis of several semantic annotation projects, available in [2], we understand that most of those frameworks and tools (such as SHOE Knowledge Annotator[1]), still rely on knowledge in HTML pages, XML documents or in other textual resources. Tools like Vannotea[2] also propose to annotate multimedia resources. However, despite the significant number of tools and frameworks that provide ontology-based annotation, none of the annotation tools proposed so far enable the annotation of *engineering models* (or, more generally, computer-based models). Concretely, there is no technique allowing to complete those models by formal comments or explanations, or to attach more semantics to the technical data produced by the modeling tools. Considering that those models keep the expert's interpretation about data, and that each expert can have a different opinion, it is expected that the annotations of those models can record the different interpretations raised by different experts.

3 Proposed Approach

In order to explicit expert knowledge in engineering models, we propose to annotate them by domain ontologies. This *model annotation* approach must be able to represent the following elements: (*i*) ontologies and their instances; (*ii*) engineering models and their data and (*iii*) annotations of the engineering models.

(*i*) The knowledge about geoscience fields was acquired with experts and represented as *domain ontologies* in RDFS/OWL. The geological ontologies were persisted in an ontology-based database.

(*ii*) We applied, then, meta-modeling techniques to represent the engineering model's data *as instance of its meta-data*. We represented the actual data schemes and their access information (such as file names) as meta-data. This way, we can address the problem of retrieving the real data artefacts. In order to persist the access information of engineering models, we need to provide a meta-model and store these information in the same database as the ontologies. But it is not desirable to represent the engineering meta-data using constructs for ontologies, since we do not expect to have, for engineering meta-data, the same features that are currently proposed to ontologies, such as, subsumption between concepts. The constructs of engineering meta-data are different from the constructs used to define ontologies (such as `owl:Class` in OWL language), because those entities have different nature. For these reasons, in order to persist engineering meta-data along with ontologies, we decided to increase the original set of constructs for building ontologies with *Engineering Meta-model* constructs. The Engineering Meta-model is actually the minimum necessary set of the features that allows an uniform description of those models (file name, identificator, main composite objects, etc.). The main constructs for building engineering meta-data are `#DataElement` and `#DataAttribute`.

(*iii*) Finally, we have to provide a means of linking engineering meta-models to the concepts of ontologies. As explained in section 2, in this context, each

[1] `http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html`
[2] `http://www.itee.uq.edu.au/~eresearch/projects/vannotea/`

end-user will furnish his or her own interpretation about the data. It means that for the same dataset, we will probably have different annotations expressing each user's opinion, and that must be uniquely identified. Another requirement is that one user can annotate several data elements with one ontology concept, and vice-versa. We need, then, an N-to-N annotation elements. Therefore, in this approach, the annotation becomes a top-level entity, separated of the ontological concept and from the entity being annotated. The annotation entity have also its own attributes, such as creation date, author name and version information.

It follows that we also defined a *Meta-model for Annotation* of engineering models. The construct `#Annotation`, creates a link between the entity used for defining ontology concepts (varies depending on the ontology model) and the entity `#DataElement`, by means of the relations `#annotates` and `#isAnnotatedBy`. The meta-models are illustrated in Fig. 1 as UML class diagrams[3].

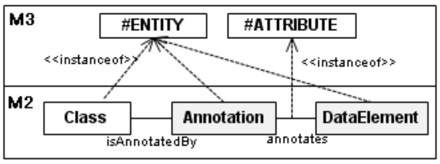

Fig. 1. Constructs of Engineering and Annotation Meta-models

Since data can now be annotated with ontological concepts, it is possible to formulate queries using semantic terms. In the following section we illustrate how we validate this approach by implementing the case study in oilfield modelling.

4 Case Study: Annotating Oilfield Models

In order to implement this case study (annotation of oilfield models), we need to represent (*i*) how data is actually structured by the computer applications, (*ii*) how data is identified using knowledge-level concepts, (*iii*) the annotations made by users over the data. Subsequently, we are interested in storing the whole data and knowledge manipulated by engineers in a persistent infrastructure. For this purpose, we use *ontology-based databases*.

4.1 Ontology-Based Databases (OBDBs)

Ontology-Based Databases are database architectures that deal with the problem of the persistence of ontologies while taking advantage of the characteristics of databases (scalability, safety, capability to manage a huge amount of data, etc.).

[3] The proposed meta-models are represented here as the M2 layer in the OMG's Meta Object Facility (MOF) four-layer architecture (`http://www.omg.org/mof/`).

The *OntoDB* architecture [3] makes use of metamodeling techniques and propose separation of modeling layers. OntoDB allows, thus, to represent the different constructors of existing ontology models (e.g, RDF, OWL, PLIB), which enables to store ontologies specified in different ontology languages, and to separate the instances, from their data structure and from their meta-model. The support of evolution of the OBDB meta-model is important, since we need to extend it to represent other meta-models than the ontology meta-model. As a consequence, we have chosen the OntoDB architecture for the persistence of data and ontologies in this approach.

4.2 Implementation

The first step was to extend the original meta-model of OntoDB architecture (which included constructs for building ontologies, such as `#Class` and `#Property`) with the new meta-models for handling both data elements issued from engineering models and annotations. After that, we declared the meta-data of oilfield models using the new constructs for Engineering Meta-Models, as exemplified in the expression in Turtle notation below, which creates a data-file element named `XYZFile` with a `filename` attribute:

```
PREFIX ma: <http://ex.org/modelannotation#>
CONSTRUCT ( ?x ma:type ma:DataElement .
            ?x ma:name ‘XYZFile’ .
            ?x ma:DataAttribute ‘filename’ . )
```

Subsequently, we added in OntoDB's repository all the oilfield meta-data and their instances (the values of the actual data artifacts).

We defined, then, with the help of the end-user, annotations that represent the expert's interpretation about field data. For example, the data contained in an `XYZFile` artefact is interpreted by a geologist as being a *Seismic Reflector*, which is a term from one of the oilfield specific domains (called GeoSeismic), and is represented as the concept `Reflector`. Therefore, we created an annotation-type that annotates element of type `XYZFile` with concepts of type `Reflector`.

```
CONSTRUCT ( ?x ma:type ma:Annotation .
            ?x ma:annotates ?data .
            ?x ma:isAnnotatedBy ?concept . )
WHERE ( ?data ma:type ma:XYZFile .
        ?concept rdf:type ma:Reflector. )
```

In the instance level, this annotation will make reference to an instance of the meta-data `XYZFile` and an instance of the ontology concept `Reflector`.

5 Results

At this point of the work, it is possible to formulate queries over field data using the concepts from the domain ontologies, instead of resorting to the internal format of data. We present here how to use the extended OntoDB's set of constructs to interrogate the database to find the *name of the data element that is annotated by the* `Reflector` *ontology concept*:

```
SELECT ?namedata
WHERE ( ?ann ma:type ma:Annotation .
        ?ann ma:isAnnotatedBy ?concept .
        ?concept rdf:type ma:Reflector .
        ?ann ma:annotates ?data .
        ?data ma:name ?namedata)
```

This query is formulated in the meta-data level (M1 layer, in OMG's MOF architecture, as explained in section 3). We have as answer the *type* of data element that is annotated by the concept `Reflector` (in this case, the data-element `XYZFile`). We are also able to formulate queries in the instance level, once we have the URIs of the ontological individuals that corresponds to the data set we are interested in.

6 Conclusions and future work

Thanks to the meta-models proposed, we are able to integrate engineering models in a knowledge level, by means of the annotation link between engineering meta-models and domain ontologies. This way, the semantic concerning the interpretation of field data, which is usually just in the head of the engineer that builds the model, can be added into the database. This approach enables to formulate queries that use the vocabulary that is significant for the domain professionals, instead of obliging them to understand how data is organised within the database. As future work, we intend to explore the multidisciplinary aspect of this domain. We aim to correlate data issued from different fields of expertise, by means of ontology mapping and subsumption relations.

Acknowledgments. This work is sponsored by The CAPES Foundation, Ministry of Education of Brazil (process no. 4232/05-4).

References

1. Chum, F.: Ontology-driven information integration and delivery: A survey of semantic web technology in the oil and gas industry. W3C Semantic Web Use Cases (2007) URL: `http://www.w3.org/2001/sw/sweo/public/UseCases/Chevron/`.
2. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web **4**(1) (2006) 14–28
3. Dehainsala, H., Pierra, G., Bellatreche, L.: OntoDB: An ontology-based database for data intensive applications". In: Proceedings of Database Systems for Advanced Applications, 12th International Conference (DASFAA07), Springer (2007)