A Semantic-based Data Service for Oil and Gas Engineering

Lina Jia[1,2], Changjun Hu[1,2], Yang Li[1,2], Xin Liu[1,2], Xin Cheng[1,2], Jianjun Zhang[3] and Junfeng Shi[3]
[1]*School of Computer & Communication Engineering, University of Science & Technology Beijing, No.30 Xueyuan Road, Haidian District, Beijing, China*
[2]*Beijing Key Laboratory of Knowledge Engineering for Materials Science, No.30 Xueyuan Road, Haidian District, Beijing, China*
[3]*Research Institute of Exploration & Development, CNPC, No.20 Xueyuan Road, Haidian District, Beijing, China*

Keywords: Semantic-based Data Integration, Ontology, Data Service, Data of Oil and Gas Engineering.

Abstract: For complex data sources of oil and gas engineering, this paper summarizes characteristics and semantic relationships of oil data, and presents a semantic-based data service for oil and gas engineering (SDSOge). The domain semantic data model is constructed using ontology technology, and semantic-based data integration is achieved by ontology extraction, ontology mapping, query translation, and data cleaning. With the semantic-based data query and sharing service, users can directly access distributed and heterogeneous data sources through the global semantic data model. SDSOge has been used by upper applications, and the results show that SDSOge is efficient in providing a comprehensive and real-time data service, saving energy, and improving production.

1 INTRODUCTION

With continuous expansion of the scale of petroleum exploration industry, the domain of oil and gas engineering has accumulated massive data resources, like production data, geological structures, equipment data, well structure data, etc. These data are large in scales, numerous in kinds, complex in relationships and various in characteristics:

1) Distribution: In oil fields, different types of data are stored in different specialized databases, such as production database, geological database, and equipment database. But applications of oil and gas engineering require various data from different databases.

2) Heterogeneity: Each specialized database has its own data organizing and naming convention, which results in system, syntax, structure, and semantic heterogeneity. (1) System heterogeneity: Different data have different operating environments, such as hardware configurations and operating systems. (2) Syntax heterogeneity: Different data are stored in different forms in the computers. Some are in relational databases, while some are in text files. (3) Structure heterogeneity: Similar data are represented in different data schemas. (4) Semantic heterogeneity: Similar data have different semantic understandings, or different data have the same meaning, which has traditionally been divided into homonyms and synonyms.

3) Complex Semantic Relationships: There are complex relationships between different data.

4) Real-time Performance: The data of oil and gas engineering is dynamic and instantly updated with high real-time demand.

The characteristics of data of oil and gas engineering bring unprecedented challenges for conventional data management. On the one hand, with the differences in data schemas of different oil fields and the shortage of data management and naming rules, it is necessary to shield heterogeneity of underlying data to establish a global semantic data model for the domain of oil and gas engineering, which can maintain the unification of rules and standards, and data management platform. On the other hand, applications of oil and gas engineering are typically data-intensive. Data are the source of these applications and various data from different specialized databases are needed, but databases of oil fields are highly autonomous, which makes data interacting and sharing more difficulty. Thus semantic-based data integration is urgently in need, which can provide a unified and semantic-based interface to access the underlying data sources directly and implement data sharing.

Jia L., Hu C., Li Y., Liu X., Cheng X., Zhang J. and Shi J..
A Semantic-based Data Service for Oil and Gas Engineering.
DOI: 10.5220/0004947601310136
In *Proceedings of the 10th International Conference on Web Information Systems and Technologies* (WEBIST-2014), pages 131-136
ISBN: 978-989-758-024-6

This paper presents a semantic-based data service for oil and gas engineering named SDSOge, which provides a rich semantic view of the underlying data and enables an advanced querying functionality. Users can enjoy a plug-and-play (Mezini and Lieberherr 1998) model and have direct access to the distributed and heterogeneous data resources anywhere. In addition, the data service offers a semantic reasoning functionality, which can reason implicit knowledge behind the complicated semantic relationships.

SDSOge firstly extracts local ontologies from schemas of data sources using ontology technology, and then establishes a completed global ontology which can support each local data source. Furthermore, an interface is set up to access underlying data sources, which can eliminate differences in data sources and provide a uniform and transparent semantic-based data query service. Finally, the cleaned standard data are returned to the upper applications.

This paper is organized as follows. Section 2 introduces related work while section 3 describes the architecture of SDSOge and its implementation in details. The usage of SDSOge system and its production application pointing out the advantages comparing to previously employed techniques are illustrated in section 4. Finally, the conclusion and directions for future work are given in section 5.

2 RELATED WORK

As the complexity of data brings more and more challenges, a new approach of data service is becoming increasingly necessary.

Carey et al. (2012) survey three kinds of popular data services, service-enabling data stores, integrated data services and cloud data service, respectively. But none of the three considers semantic association.

Doan et al. (2004) introduce the special issue on semantic integration. They point out that 60-80% of the resources in a data sharing project are spent on reconciling semantic heterogeneity. Halevy et al. (2005) describe successes, challenges and controversies of enterprise information integration. Kondylakis et al. (2009) review existing approaches for ontology/schema evolution and give the requirements for an ideal data integration system.

Bellatreche et al. (2006) propose the contribution of ontology-based data modeling to automatic integration of electronic catalogues within engineering databases, but this method assumes the data source itself does not have enough semantic information.

Ghawi and Cullot (2007) propose a semantic interoperability from relational database to ontology, but it only considers the case of one data source.

In order to make a more intuitive view of mapping, many mapping tools like COG, DartGrid, VisAVis, and MAPONTO, are developed. These tools need users to build mappings in an interactive way.

Data from different domains have different characteristics. These data are the basis of scientific research in the fields. Semantic–based data integration and data services for domain-oriented ontology are hotspots of current research. Establishment of semantic data models, and integration and application of semantic data in scientific fields are important aspects worthy of discussion and research.

3 SDSOGE ARCHITECTURE AND IMPLEMENTATION

3.1 System Architecture

SDSOge provides a global semantic data model and APIs for users and upper applications to send queries and receive desired data. Service consumers need not to know the source and original schema of data. Figure 1 shows the architecture of SDSOge.

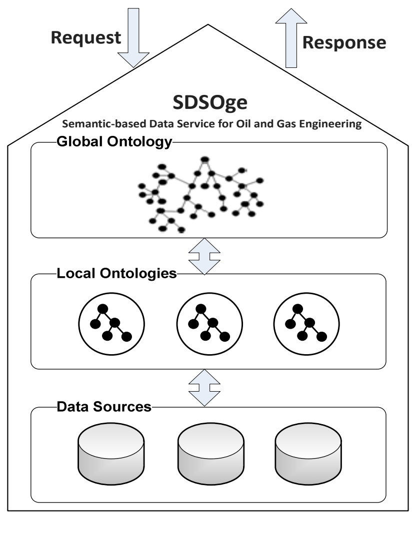

Figure 1: SDSOge Architecture.

3.2 Global Ontology Construction

There are four steps to establish the global ontology.

First of all, filter data of oil and gas engineering field and get entities that system needs and relationships between the entities. Next, extract schema information of databases to establish local ontoloties using ontology technology. Then, the global ontology can be built through standardizing names of properties with the synonym table, and further refining, improving and merging of local ontologies. Finally, adding semantic constraint rules and reasoning mechanisms to form a complete and semantically rich global ontology. The global ontology construction process is shown in Figure 2.

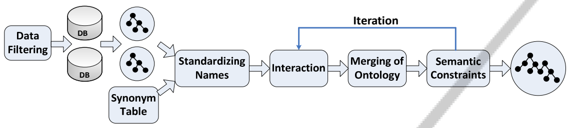

Figure 2: The global ontology construction process.

3.2.1 Data Filtering

In the field of petroleum exploration and development, data involve more than 20 professional aspects, and data of oil and gas engineering domain are just a part of them. So we should firstly define the basic scope of required data to form entities, attributes and relationships between entities referring to the data dictionary.

Take block data entity and sucker rod data entity as examples, the corresponding entity models are as follows.

Block data entity:

E(BlockInfo)={block_name, oil_density, permeability, reservoir_depth, ……}

Sucker rod data entity:

E(SuckerRodInfo)={sucker_rod_id, diameter, length,……}

3.2.2 From Relational Database to Local Ontology

Based on the features of tables and constraints between tables in the specialized databases, rules from relational database to local ontology are defined as follows.

Rule1: Convert each table T into a class or a subclass C_T (OWL: Class or OWL: Subclass).

Rule2: Convert C_{Tj} into a subclass of C_{Ti}, if the foreign key of table T_i corresponds to the primary key of table T_j (OWL: Subclass).

Rule3: Convert the foreign key of table T into object property OP_T (OWL: ObjectProperty).

Rule4: Convert the primary key of table T into the datatype property with functional property DP_T (OWL: DatatypeProperty).

Rule5: Convert other columns of table T into data properties DP_T (OWL: DatatypeProperty).

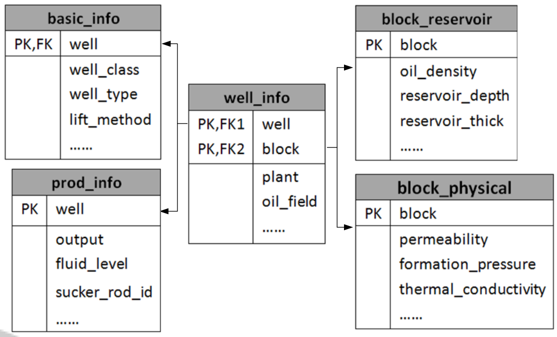

Figure 3: Tables in production database (partial).

Figure 3 shows the schema of a few tables in production database. According to the mapping rules above, the local ontology can be generated automatically. The relationships between classes are foreign key constraints in the database, as shown in Figure 4.

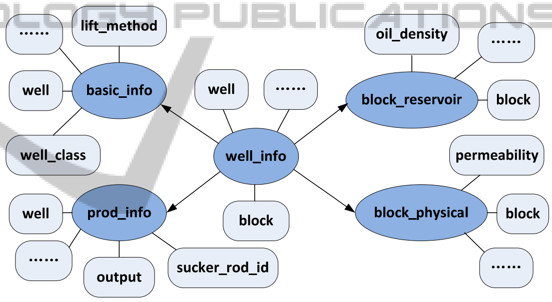

Figure 4: Local ontology of production database (partial classes).

3.2.3 From Local Ontologies to Global Ontology

The process of local ontologies to global ontology is divided into three steps, renaming of properties, merging of classes, and combination of local ontologies.

Renaming of properties, comparing names of ontology properties with the corresponding terms in the synonym table, aims at ensuring consistency of domain terminologies and reusing the semantic data model in the field. The synonym table, which is constructed by domain experts and DBAs referring to exploration-development database handbooks, can solve problems of semantic heterogeneity. The names of terms with synonymous semantic relations in the handbooks are stored in a same collection in the synonym table. The collection name is unified into the corresponding name of the attribute in the entity, which is defined in 3.2.1.

If the name of ontology property is in the synonym table, rename the ontology property to the corresponding collection name in the synonym table. If it is not in the synonym table, user is required to complete the property renaming task through the GUI, and then add the property into the synonym table. If one property name of local ontology corresponds to multiple collection names in the synonym table, which is semantic heterogeneity of the same vocabulary expressing different meanings in different data sources, the GUI is also needed.

We propose a merging algorithm in the stage of classes merging. Comparing local ontology properties with the entity attributes constructed in the step of data filtering, the scope of ontology datatype properties of a class must be consistent with the corresponding attributes range of the entity, and the class name must be same with the corresponding entity name. If properties of two or more ontology classes correspond to one entity, merge the two or more classes into one class named the corresponding entity name.

The classes merging algorithm is detailed as follows.

Step1: Create an ontology class C_i, whose name is the name of entity E(i).

Step2: $\forall DP_T \in C_T$, if $DP_T \in E(i) \wedge DP_T \notin C_i$, add DP_T into class C_i, and delete DP_T from class C_T. If $DP_T \in E(i) \wedge DP_T \in C_i$, delete DP_T from class C_T, and do not add DP_T into class C_i.

Step3: If $\forall DP_T \notin C_T$, delete class C_T, the C_T' constraint relationships convert into C_i'.

Step4: Traverse other classes C_T of local ontology, loop through Step 2 and 3.

Step5: Select other entities E(i), and loop through Step 1-4 until all the entities have been traversed.

Figure 5 shows the normalized local ontology of production database after properties renaming and classes merging. Take class BlockInfo in Figure 5 as an example to illustrate the classes merging steps. Create a new class named BlockInfo firstly. In Figure 4, the names of datatype properties of class block_reservoir are in the entity BlockInfo, which is defined in the step of data filtering, so add the datatype properties into the new class BlockInfo, and delete the datatype properties from class block_reservoir. If all the datatype properties in class block_reservoir are deleted, delete class block_reservoir, and the constraint relationships of class block_reservoir are turned into class BlockInfo'. Similarly, traverse other classes. Here, we also add the datatype properties of block_physical into the new class BlockInfo.

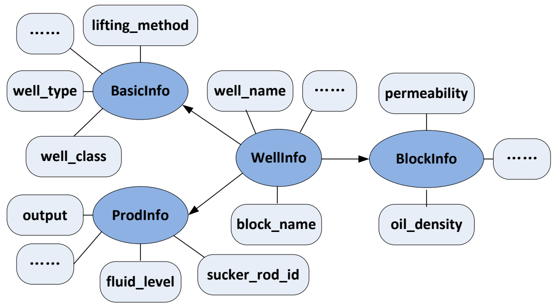

Figure 5: Normalized local ontology of production database (partial classes).

Next is combining local ontologies generated from different specialized databases into a global ontology. Starting to traverse the root classes of two local ontologies, if the two classes have the same datatype property, bridge the two classes by a foreign key constraint relationship. The class with functional property is converted into the subclass of the other class without functional property. Two local ontologies can be linked in this way. And then other local ontologies can be combined.

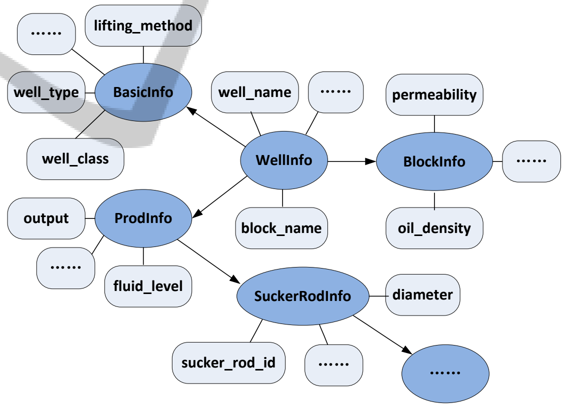

Figure 6: Global Ontology (partial classes).

Figure 6 shows a global ontology, which is a result of the combination of production database ontology and equipment database ontology. Sucker_rod_id is not only the primary key of table sucker_rod in equipment database, but also a property of table prod_info in production database, so bridge the two classes via sucker_rod_id by a foreign key constraint relationship.

Local ontologies can be converted into a global ontology after properties renaming, classes merging, and local ontologies combining.

3.2.4 Adding Semantic Constraint Rules

Semantic constraint rules are added to strengthen the hierarchical relationships between concepts. Reasoning engine can use the constraint rules to reclassify and reorganize concepts of the global ontology, achieve a certain reasoning function, and obtain the implicit knowledge.

3.3 Semantic Query

According to the global semantic view, users can submit SPARQL statements to query the global ontology. SPARQL statements are converted into SQL to access the underlying data sources. Finally, the query results are presented to users in a uniform format after cleaning.

The semantic query implementation steps are as follows.

Step1: Get the query request, and generate the global query statement Q_G, which is described by SPARQL.

Step2: Reasoning engine converts names of classes/properties of Q_G in global ontology into the names in relative local ontologies based on the information of synonym table.

Step3: Divide the global query Q_G into sub queries $\{Q_{L1}, Q_{L2}, \ldots\ldots, Q_{Ln}\}$ for local ontologies.

Step4: Rewrite sub queries $\{Q_{L1}, Q_{L2}, \ldots\ldots, Q_{Ln}\}$ as local sub queries $\{Q_{D1}, Q_{D2}, \ldots\ldots, Q_{Dn}\}$ for each data source. Local sub queries are described by SQL.

Step5: Execute local sub queries and return the results $\{R_{D1}, R_{D2}, \ldots\ldots, R_{Dn}\}$ in unified formats.

Step6: Combine the results $\{R_{D1}, R_{D2}, \ldots\ldots, R_{Dn}\}$, and return the final query response after data cleaning and converting.

4 APPLICATION OF SDSOGE

Due to the demand of oil and gas engineering domain, we develop the SDSOge system, which is implemented based on JAVA technology. SDSOge parses the global ontology and related local ontologies using Jena and makes the reasoning function into effect. Meanwhile, SDSOge implements the extraction of schemas of data sources and the data searching process using JDBC data access interfaces. SDSOge makes the use of data more profound and efficient.

Oil and gas engineering optimization design and assisted management system (OGEA) is a typical example of industrial application of SDSOge. OGEA is widely used in oil and gas engineering field. It could implement the production design and decision-making process with the support of specialized databases, thus increase the production and recovery ratio.

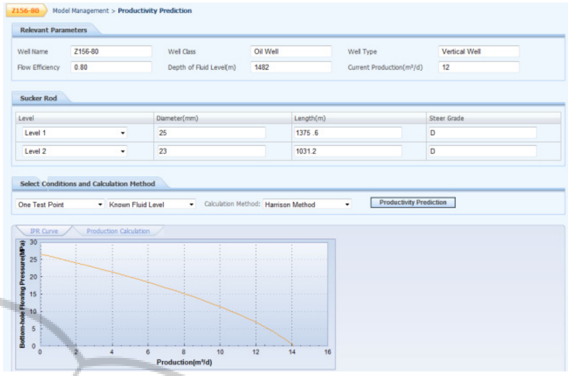

Figure 7: Interface of productivity prediction module.

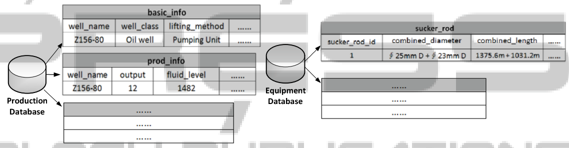

Figure 8: Corresponding data sources of productivity prediction module.

Figure 7 shows the interface of productivity prediction module of OGEA. The corresponding data sources of the module are shown in Figure 8. In Figure 7, the relevant parameters, such as depth of fluid level and current production, are collected from production database, while sucker rod data are collected from equipment database; which implements the integration of distributed data. The structure of sucker rod in Figure 7 is stored differently in databases from that in Figure 8. SDSOge shields the structural heterogeneity and presents sucker rod data to the upper level in the same format. The lower part of Figure 7 is the result of productivity prediction using the data in the upper portion. The application shown in Figure 7 is for multiple fields, but names of the same type of needed information are not identical in the databases of different oil fields. SDSOge can shield this semantic heterogeneity and map into the corresponding individuals by reasoning engine.

The OGEA system equipped with SDSOge has been put into production in oil fields of Daqing, Jilin, Huabei, and Dagang. Currently, SDSOge, which has measured effect evaluation for 28985 wells, could provide an entire and real-time data service of production monitoring and perform well in real applications.

After application of OGEA system with SDSOge in five oil production plants in Huabei Oil Field, the average efficiency has increased by 3.6%, while the average pump inspection period has increased by 83 days, and total oil production has increased by 9054 tons. The cost of manpower and material resources has been saved, and the efficiency of management has been improved. Moreover, the average system efficiency has improved 3.75% and the average pump inspection period has increased by 75 days after the SDSOge applied in six oil production plants of Dagang Oil Field, which makes a lot of sense in extending pump inspection period, saving energy and raising production.

Based on the distributed and heterogeneous databases of oil fields, SDSOge shields the heterogeneity of underlying databases, builds the global semantic data model, provides the semantic searching function based on domain terminologies, and makes the searching results available for upper applications. SDSOge enables the value of data improved.

5 CONCLUSIONS AND FUTURE WORK

The current researches and applications mainly focus on solving semantic heterogeneity between data sources using ontology, data integration based on semantic methods, and data services for upper applications.

The semantic-based data service mentioned in this paper connects distributed, heterogeneous and complicated data seamlessly, which makes upper applications moving smoothly on SDSOge platform. SDSOge, which makes data shared and reused, builds a semantic-abundant global ontology in the domain of oil and gas engineering, implements data query transformations based on semantic methods, and provides a data service for upper applications. SDSOge could shield the heterogeneity of underlying data sources and allow users to access the standard data everywhere directly, thus provide effective data supports for production. SDSOge combines industrial production and scientific research tightly and is a great example that science promotes the progress of industry.

In the future, we would add more reasoning mechanisms to provide better semantic-based data services, and introduce SDSOge into more oil fields.

ACKNOWLEDGEMENTS

This work is supported by the R&D Infrastructure and Facility Development Program under Grant No. 2005DKA32800, the Key Science-Technology Plan of the National ‘Twelfth Five-Year-Plan’ of China under Grant No. 2011BAK08B04, the 2012 Ladder Plan Project of Beijing Key Laboratory of Knowledge Engineering for Materials Science under Grant No. Z121101002812005, the National Key Basic Research and Development Program (973 Program) under Grant No. 2013CB329606, and the Fundamental Research Funds for the Central Universities under Grant No. FRF-MP-12-007A.

REFERENCES

Bellatreche, L., Dung, N. X., Pierra, G., Dehainsala, H., 2006. Contribution of ontology-based data modelling to automatic integration of electronic catalogues within engineering databases. In *Computers in Industry*, 57(8-9), 711-724.

Carey, M. J., Onose, N., Petropoulos, M., 2012. Data Services. *Communications of the ACM*, 55(6), 86-97.

Doan, A., Noy, N., Halevy, A., 2004. Introduction to the special issue on semantic integration. In *ACM SIGMOD Record*, 33(4), 11-13.

Ghawi, R., Cullot, N., 2007. Database-to-Ontology Mapping Generation for Semantic Interoperability. In *VLDB ’07*, Vienna, Austria.

Halevy, A.Y., Ashish, N., Bitton, D., et al, 2005.Enterprise information integration: Successes, Challenges and Controversies. In *SIGMOD Conference*, 778-187.

Hu, C., Tong, Z., et al, 2001. Research on Constructing of Object-Oriented Petroleum Common Data Model. *Journal of Software*, 12(3), 427-434.

Kondylakis, H., Flouris, G., Plexousakis, D., 2009. Ontology and Schema Evolution in Data Integration: Review and Assessment. In *Meeraman, R., Dillon, T., Herrero, P. (eds.) OTM 2009. LNCS,* 5871, 932-947. Springer, Heidelberg (2009).

Kondylakis, H., Plexousakis, D., 2011. Exlixis: Evolving Ontology-Based Data Integration System. In *SIGMOD’11*, 1283-1286.

Ludäscher, B., Lin, K., Bowers S., et al, 2006. Managing Scientific Data: From Data Integration to Scientific Workflows. *Geological Society of America Special Paper on GeoInformatics*, 109-129.

Mezini, M., Lieberherr, K., 1998. Adaptive Plug-and-Play Components for Evolutionary Software Development. In: *Proceedings OOPSLA’98*, ACM, Vancouver, British Columbia, Canada, 97-116.

Ye, Y., Yang, D., Jiang, Z., Tong, L., 2008. Ontology-based semantic models for supply chain management. In *The International Journal of Advanced Manufacturing Technology,* 37(11-12), 1250-1260.