2016 IEEE First International Conference on Data Science in Cyberspace

OPSDS: a semantic data integration and service system based on domain ontology

Liu Xin
School of Computer and Communication Engineering
University of Science and Technology Beijing
Beijing, China
liuxin_ustb@163.com

Hu Chungjin
School of Computer and Communication Engineering
University of Science and Technology Beijing
Beijing, China
hucj0313345@163.com

Huang Jianyi
School of Computer and Communication Engineering
University of Science and Technology Beijing
Beijing, China
huangjianyi_ustb@163.com

Liu Feng
School of Computer and Communication Engineering
University of Science and Technology Beijing
Beijing, China
m17801003063@163.com

Abstract—**For the distributed, heterogeneous, relational complex data sources of petroleum engineering, we present an oil production engineering semantic-based data integration system (OPSDS). OPSDS establishes a semantic data integration and service system based on domain ontology on the premise of building a global semantic model and realizing the global semantic search. The global semantic data model applied to various oil fields is set up by ontology extraction, ontology evolution, ontology combination and semantic constraints. The domain-oriented data integration to provide the data access and shared service is realized by ontology mapping, query transformation, and data cleaning. Users and upper applications can have a direct access to underlying complex data sources in times of need through the global semantic data model, and the cleaned data can be returned in a unified format. OPSDS has been realized and got extensive use in many platforms of China National Petroleum Corporation(CNPC). It has been found that the method can not only provide the comprehensive and real-time data support for oil and gas wells, but also improve the production and recovery efficiency with good application.**

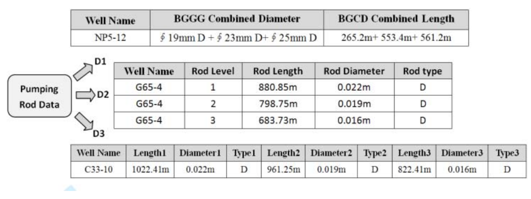
Keywords* — *petroleum information system; data service; domain data semantic integration; ontology technology of petroleum engineering; distributed data processing

I. Introduction

Oil is an important strategic resource of a country. But oil and gas production of many oil fields around the world is trending downward and more fields are transiting into decline each year. Reports indicate that a hybrid average production decline rate for oil fields worldwide is 6% more or less. Currently, various research methods to improve oil and gas production are booming. Production design, decision analysis, diagnosis and management of oil and gas wells are the keys to enhance productivity, reduce costs and increase profits. Optimal design of oil and gas wells involves large amounts of data, such as production data, well profiles, equipment data, geological structures, seismic data, reservoir data, etc. These data with huge value have drawn attentions of academia, industry and government. For the purpose of improving production and saving energy, it is a highly advocated idea to dig out the value to utilize the data more efficiently. We focus on the features of oil data firstly.

1) Distribution. Oil field is composed of a number of oil production plants, exploration institutes, geophysical research institutes and other units. Different units collect, collate, process, analysis and apply different kinds of data, and store corresponding data in their own databases; which results in that different types of data are stored in different professional databases.

2) Heterogeneity. Each database has its own specialized data organization and naming conventions, leading to system heterogeneity, syntax heterogeneity, structure heterogeneity and semantic heterogeneity[1]. System Heterogeneity means operating environments and hardware platforms of data are various in different oil companies. Syntax Heterogeneity indicates that oil companies take different storage methods for different types of data. For example, some data are stored in relational databases, and some are stored in forms of text files. Structure Heterogeneity intends that different oil fields represent the same type of data with different data schemas. A typical example is shown in Figure 1. Semantic Heterogeneity mainly refers to different words with the same meaning or the same words with different meanings.



Well Name	BGGG Combined Diameter	BGCD Combined Length
NP5-12	∮ 19mm D + ∮ 23mm D+ ∮ 25mm D	265.2m+ 553.4m+ 561.2m

Well Name	Rod Level	Rod Length	Rod Diameter	Rod type
G65-4	1	880.85m	0.022m	D
G65-4	2	798.75m	0.019m	D
G65-4	3	683.73m	0.016m	D

Well Name	Length1	Diameter1	Type1	Length2	Diameter2	Type2	Length3	Diameter3	Type3
C33-10	1022.41m	0.022m	D	961.25m	0.019m	D	822.41m	0.016m	D

Fig. 1. Sucker rod data with structure heterogeneity.

Figure 1 shows an instance of sucker rod structural data stored in relational database. A well requires a set of sucker rod data, which contain rod level, length, diameter and other information, and different rod levels correspond to different rod lengths. For a multilevel sucker rod, D1 combines the three-level rod length and saves as one field, D2 saves rod lengths into three rows according to different rod level, and D3 represents three fields in one row.

3) Instantaneity. Oil production engineering data are dynamic, updated in real time, and in critical instant need. Each oil field has not only production data every day, but also constantly updated basic data and regularly updated equipment data. So it is vital to ensure the real-time of data for upper applications.

4) Complex semantic associations. It mainly refers to the complex associations between different data. For example, we regard the well whose deviation angle is less than 5 degrees as a vertical well, the well whose deviation angle is greater than 75 degrees as a horizontal well, and the well whose deviation angle is between 5 degrees and 75 degrees as a inclined well.

Focusing on the characteristics of the data leads us to discuss the challenge of traditional data management. On one hand, data of oil fields are scatteredly stored, the logical organization lacks of 'soul', and the data schemas are various without naming rules and management methods. Thus it is urgent to establish a global semantic data model which is suitable for multiple oil fields to achieve the unification of data management platform. On the other hand, data of oil companies are considerable autonomy, which increas the difficulty of data exchange and sharing. But data from different professional databases are increasingly need to work together to support upper applications of the domain. So semantic data integration and building uniform interfaces directly accessing to the underlying data resources is of great significance.

In this paper, a petroleum-engineering semantic-based data service (OPSDS) is presented to achieve a semantic data integration and service system based on domain ontology. The system provides a semantically richer global ontology and query-based access to the distributed and heterogeneous data. OPSDS shields the complexity and sources of data to enable users and upper applications to take full advantage of data resources in a pay-as-you-go approach everywhere. Besides, a reasoning function is available for inferring the hidden information behind the semantic associations.

II. Related work

As the complexity of data leads to a raising challenge for traditional data management, it is of utmost importance to generate a new way of data service. Data services provide access to data drawn from one or more underlying information sources[2]. Service-enabling data stores, integrated data services and cloud data services inthe enterprise world are introduced in detail, but semantic relationships are not considered. And Dong et al.[3] propose a framework for scientific data services. Data integration is a pervasive challenge faced in applications that need to query data residing at multiple autonomous and heterogeneous data sources [4]. And new features of data integration are concluded[5]. Bernstein and Haas[6] say that information integration combines information from different sources into a unified format, and they specify the complicacy of integration after investigating the tools and technologies of data integration in the enterprise. They also indicate that every step of the integration process requires a good deal of manual intervention and more automation is surely possible.

Wang et al.[7] present a collaborative environment called distributed interoperable manufacturing platform, in which the STEP-NC data model is built to promote data exchange among heterogeneous systems. Bender et al.[8] propose a service-oriented framework for integration of domain-specific data models in scientific workflows, which links the data sources and upper applications. However, the data model is built by domain experts, which is subjectivity and lacking in semantic relations between the data elements. Mapping of data with association relationships are constructed[9], but a certain inaccuracy exists. Zdravković et al.[10] present three existing enterprise ontologies with different levels of expressivity. Apparently, their work is not for specific domain, especially for the oil field.

III. Architecture of OPSDS

A. OPSDS architecture

OPSDS provides a rich semantic view of the underlying data and interfaces enabling users and upper applications to access data. The architecture of OPSDS is shown in Figure 2.

The bottom of the architecture is data sources storing in different databases, such as Oracle, SQL Server, etc. The middle layer is local ontologies extracting from the data sources below. And then, the global ontology is formed as the result of combining local ontologies. Users and upper applications can access the data resources easily. The only thing service consumers need to do is to send queries according to the global ontology, and then the desired data can be received.

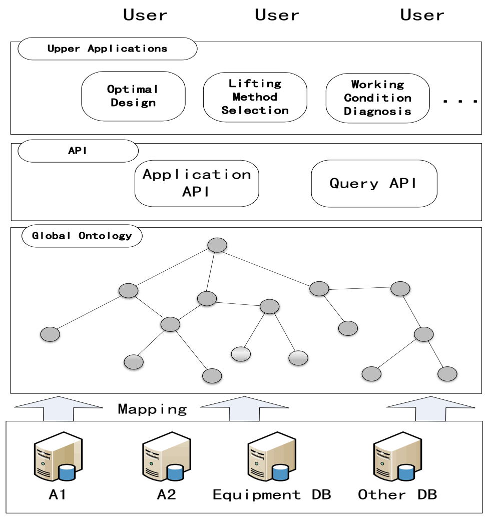


Fig. 2. Architecture of OPSDS.

B. Construction of global ontology

We adopt a hybrid strategy to construct the global ontology. On the one hand, a top-down approach is used to filter the demand data. Entities, attributes and relationships. For Peer Review Only between entities can be got by classifying and organizing the data. On the other hand, take a bottom-up method to build local ontologies, which are results of extracting schemas of databases and items of synonym list. And then the global ontology is established according to ontology evolution, ontology mapping and imposed semantic constraints. Figure 3 shows the construction process of global ontology.

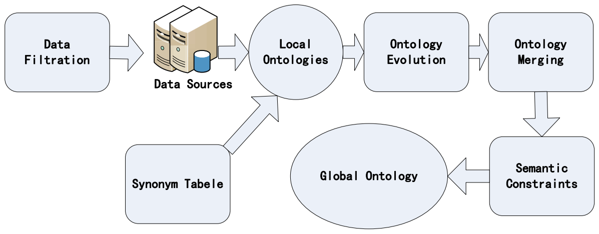


Fig. 3. The construction process of a global ontology.

Data of petroleum exploration and development vary in many aspects, such as exploration, production, geology, seismology, well logging, well drilling, etc, while data of petroleum engineering are just a part of them. Thus, firstly, we filter data to define the basic requirements, and classify, organize and aggregate the data to form entities, attributes and labels. Then identify the relationships between entities referring to the data dictionary.

Take production data entity and equipment data entity as examples. The entity models are as follows.

Production data entity: Production (oil_output, gas_output, flow_pressure……)

Equipment data entity: Equipment (pumping_unit, sucker_rod, defueling_pump……)

Since the majority of petroleum engineering data are stored in relational databases, we are here to study mapping from Relational Database to OWL ontology. A relational database is composed of a set of relational schemas, including basic table structures and integrity constraints. An OWL ontology consists of classes, properties, individuals and axioms. As we aim at providing mappings between data models and ontologies, classes and properties are considered in this step.

Because of individuals are widely exist in underlying database, individuals are not taken into consideration. Axioms are covered later in this paper. The synonym list of petroleum engineering is built by domain experts and DBAs by reference to exploration and development handbooks of oil fields. The synonymous items with different names and same meaning in the handbooks are gathered together in the synonym list to solve the phenomena of semantic heterogeneity.

Based on the schemas of tables in the specialized databases, we analyze characteristics of tables and constraints between tables, and then define an oil production engineering data source ontology (OPDSOnto), which maps synonyms in the synonym list and schemas of tables to classes and properties in the ontology. The local ontology can be generated automatically through the program. Getting innovations from Relational.OWL, OWL-RDBO and Pro/Innovator, we design OPDSOnto to describe tables, columns relations of tables and synonymy. Then extraction rules are defined as follows.

Rule 1. Convert tables and columns in databases to classes OPDSOnto: Table or OPDSOnto: Column (owl: Class), which express main concepts of the domain.

Rule 2. Hierarchical relationships between tables and columns are presented by OPDSOnto: hasParent and OPDSOnto: hasChild (owl: ObjectProperty) with owl: inverseOf constructs. OPDSOnto: hasChild has a direction from domain Table to range Column, while OPDSOnto: hasParent has an opposite direction.

Rule 3. Relationships between columns in one table are presented by OPDSOnto: hasBrother, which is defined in owl: ObjectProperty.

Rule 4. If a column C in table A is the foreign key to table B, OPDSOnto: hasChild represents the foreign key constraint, from domain column C to range Table B, while OPDSOnto: hasParent is the reverse semantic association.

Rule 5. Datatype Properties of classes are defined, such as OPDSOnto: isPK, OPDSOnto: isFK, OPDSOnto: isNullable, OPDSOnto: dataType, to describe the primary key, the foreign key, nullable and data type of the individual.

Rule 6. Extract the items which express the same meaning from the synonym list to convert into classes, and the relationships between classes are defined as OPDSOnto: hasSynonymy, which is built in owl: ObjectProperty. Semantic Relationship is defined as shown below.

DEFINITION(Semantic Relationship). $\forall$ $x_1,x_2\ldots\ldots x_n,z$, if x_1->z, x_2->z, …… , x_n->z, then $x_1\#x_2\#\ldots\ldots\#x_n$, where -> indicates the relation between two classes and # identifies the semantic relationships between classes.

According to the definition, the relationships that are hasParent, hasChild, hasBrother and hasSynonymy defined above among classes can be enriched.

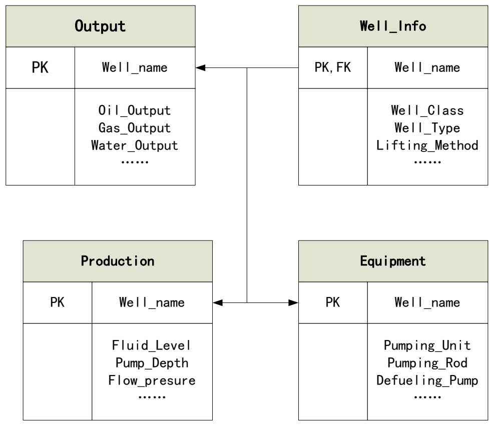


Fig. 4. Tables from production database (partial).

Figure 4 shows schemas of four tables from production database. Take Table well_info, Column well_name from Table well_info and Table output as examples to specify the process of ontology extraction, which is shown in Figure 5.

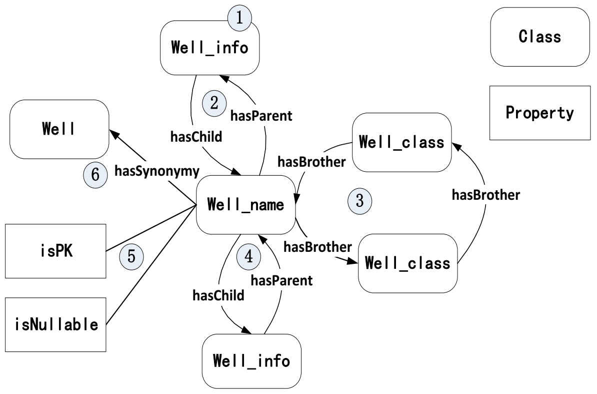


Fig. 5. The steps of local ontology extraction.

The number in Figure 5 corresponds to the rule number. No.1 indicates that convert table names well_info and column name well_name into classes well_info. No.2 means the relational schema of well_info and well_name is turned to a parent-child relationship in the local ontology. No.3 is converting the two columns well_name and well_class from the same table into a hasBrother relation. No.4 represents that the foreign key constraint of well_name. Table well_info and table output is converted into a parent-child relationship. Rule 5 defines datatype properties of class well_name, while Rule 6 extracts synonyms of well_name from synonym list and defines relationships between synonyms as hasSynonymy.

Figure 6 shows the local ontology after extracting schemas of production database and synonym list. The arrows in Figure 6 represent the relation of hasChild. The classes with synonymy are circled in one node.

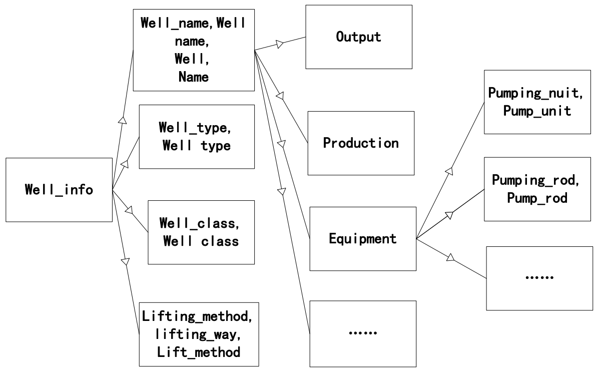


Fig. 6. Local ontology of production database (partial classes).

There are two steps from local ontologies to global ontology, which are evolution of local ontology and combination of local ontologies. Due to data storage characteristics of specialized databases in the petroleum engineering field, evolution of local ontology means if two classes with different parent nodes describe the same kind of information, that is, the two classes correspond to different attributes of one entity formed in the step of data filtering, the two classes evolve to a relation of hasBrother, parents of the two classes evolve to a relation of hasSynonymy.

When it comes to combination of local ontologies, definite relationships must exist between the local ontologies. The global ontology can be built by mapping relevant local ontologies. After analyzing schemas of different databases in the domain, the relationship between two local ontologies, which have the same class, can be established as a foreign key constraint. The parent node of the class with isPK property is mapped to the subclass of the class without isPK property. For example, equip_info is a table of production database, and pump_parameters is a table of equipment database.

The schemas of the two tables are as follows:

equip_info (well_name, pumping_unit, pumping_rod……) ;

pump_parameters (pumping_unit, manufacturer, power……).

Well_name is the primary key of table equip_info, and pumping_unit is the primary key of table pump_parameters. In our method, we regard pumping_unit as a foreign key of table equip_info linking to table pump_parameters. Thus, production ontology and equipment ontology are combined into one ontology. Figure 7 shows the combined ontology.

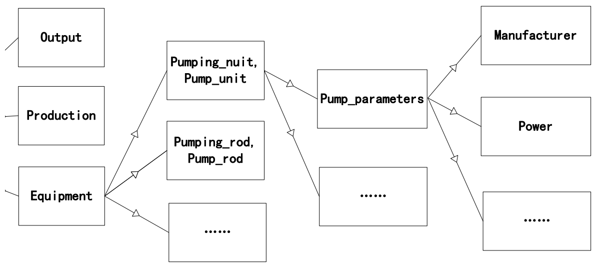


Fig. 7. Global ontology (partial classes).

In Figure 7, the node (output, production) is the evolutionary result of class output and class production. The relationship between output and production is evolved into hasSynonymy, and the relations of subclasses of output and production are hasBrother.

After local ontology evolution and local ontologies combination, local ontologies can be converted into a global ontology.

We add some semantic constraints to strengthen the relations of terminology, and inference engine can reason and deduce the global ontology to reorganize the concepts using the constraints. Thus the implied semantic information can be got and value-added services can be provided to users. Semantic constraints are defined as follows.

[Rule1: (?x OPDSOnto: has Child ?y) ， (?y OPDSOnto: hasSynonymy ?z) -> (?x OPDSOnto: has Child ?z)]

[Rule2: (?x OPDSOnto: hasSynonymy ?y) ， (?y OPDSOnto: hasBrother ?z) -> (?x OPDSOnto: hasBrother ?z)]

[Rule3: (?x OPDSOnto: hasSynonymy ?y) , (?y OPDSOnto: hasParent ?z) -> (?x OPDSOnto: hasParent ?z)]

C. The process of semantic query

Users and upper applications can submit query requests according to the global ontology, and OPSDS converts the requests to SPARQL statements to query the global ontology. Then the SPARQL statements can be rewrited as SQL statements to access underlying data sources. Finally, the query results are returned in a uniform format after data cleaning and converting. Figure 8 shows the process of semantic query.

Rewrite of SPARQL statements indicates that query requests based on the global ontology are converted into SQL statements to access underlying data sources. Mapping from global ontology to data sources is divided into one-to-one and one-to-many, which include the following three types.

1) Required data are from one table of a database.

2) Required data are from two or more tables of a database.

3) Query requests need to access more tables from multiple databases.

The process of semantic query implementation is as follows:

Step 1. Get the query request from service consumers to generate $Query_G(Q_G)$ to query the global ontology. Q_G is described by SPARQL.

Step 2. Inference engine reasons names of classes and properties of Q_G to the names in the relevant ontologies.

Step 3. Query resolver decomposes Q_G into $SubQuery_L$ (SQ_L) to query each local ontology. $SQ_L = \{SQ_{L1} , SQ_{L2} , \ldots\ldots, SQ_{Ln} \}$, where n is the number of local ontologies.

Step 4. Query rewriter converts SQ L into $SubQuery_D$ (SQ_D) to query underlying database. $SQ_D = \{SQ_{D1} , SQ_{D2} , \ldots\ldots, SQ_{Dn} \}$, and SQ_D is described by SQL.

Step 5. Execute SQ_D and return $SubResult_D$ (SR_D). $SR_D = \{SR_{D1} , SR_{D2} , \ldots\ldots, SR_{Dn} \}$.

Step 6. Result transformer cleans the query results SR_D referring to rules, and the normalized results can be got.

Step 7. Result combiner combines the normalized results, and returns the final query result.

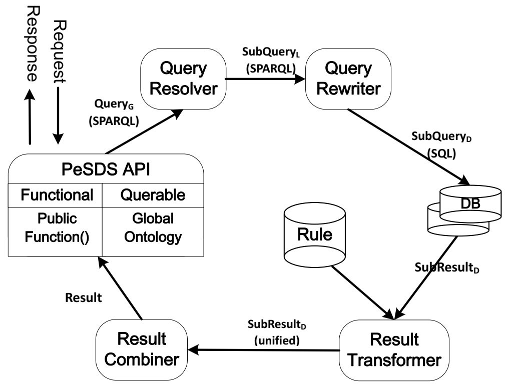


Fig. 8. shows the process of semantic query.

IV. CONCLUSION AND FUTURE WORK

Semantic-based data integration in specific area is becoming a key research currently. Build the global semantic data model for distributed, heterogeneous and complex semantic correlation data and provide comprehensive and real-time data services using ontology technology are efficient and feasible.

For data intensive industries, establishment of a global semantic data model based on domain ontology and realization of semantic-based data integration to serve for upper applications can get good results. Oil production engineering is a typical data intensive field, and OPSDS, which has realized data shared and reused, plays a key role in production practices. Through OPSDS, upper applications can directly access the underlying data resources and get the normalized data to be used for industrial production. The data of petroleum industrial applications show that our method can not only improve the production and recovery efficiency, but also save energy and reduce costs. Driven by application requirements, OPSDS connects production, study and research tightly, which is an effective way to promote scientific and technological progress and prove that science and technology is the first productive force.

In the future work, OPSDS will be used in more fields, and promoted to other oil areas like exploration and geology.

REFERENCES

[1] B. Ludäscher, K. Lin, S. Bowers et al. Managing Scientific Data: From Data Integration to Scientific Workflows[J]. GSA Today, Special Issue GSA Today, Special Issue.

[2] Carey M J, Onose N, Petropoulos M. Data services[J]. Communications of the ACM, 2012, 55(6): 86-97.

[3] Dong B, Byna S, Wu K. SDS: a framework for scientific data services[C]//Proceedings of the 8th Parallel Data Storage Workshop. ACM, 2013: 27-32.

[4] Halevy A, Rajaraman A, Ordille J. Data integration: the teenage years[C]//Proceedings of the 32nd international conference on Very large data bases. VLDB Endowment, 2006: 9-16.

[5] Smoot M E, Ono K, Ruscheinski J, et al. Cytoscape 2.8: new features for data integration and network visualization[J]. Bioinformatics, 2011, 27(3): 431-432.

[6] Bernstein P A, Haas L M. Information integration in the enterprise[J]. Communications of the ACM, 2008, 51(9): 72-79.

[7] Wang X V, Xu X W. DIMP: an interoperable solution for software integration and product data exchange[J]. Enterprise Information Systems, 2012, 6(3): 291-314.

[8] Bender A, Poschlad A, Bozic S, et al. A service-oriented framework for integration of domain-specific data models in scientific workflows[J]. Procedia Computer Science, 2013, 18: 1087-1096.

[9] Das Sarma A, Fang L, Gupta N, et al. Finding related tables[C]//Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. ACM, 2012: 817-828.

[10] Zdravković M, Panetto H, Trajanović M, et al. An approach for formalising the supply chain operations[J]. Enterprise Information Systems, 2011, 5(4): 401-421.