Semi-Automatically Generated Hybrid Ontologies for Information Integration

Lisa Ehrlinger and Wolfram Wöß
Institute for Application Oriented Knowledge Processing
Johannes Kepler University Linz, Austria
{lisa.ehrlinger | wolfram.woess}@jku.at

ABSTRACT

Large and medium-sized enterprises and organizations are in many cases characterized by a heterogeneous and distributed information system infrastructure. For data processing activities as well as data analytics and mining, it is essential to establish a correct, complete and efficient consolidation of information. Information integration and aggregation are therefore fundamental steps in many analytical workflows. Furthermore, in order to evaluate and classify the result of an integrated data query and thus, the quality of resulting data analytics, it is previously necessary to determine the data quality of each processed data source.

This paper aims mainly at the first aspect of the mentioned twofold challenge. Both, data dictionary as well as information source content are analyzed to derive the conceptual schema, which is then provided as a machine-readable description of the information source semantics. Several descriptions of the semantics can be integrated to a global view by eliminating possible redundancies and by applying ontology similarity measures. Attributes for data quality metrics are included in the descriptions but not yet determined. The implementation of the presented approach is evaluated by extracting the semantics of a specific MySQL database, represented as RDF triples.

Keywords

Data Science Workflows, Data Analytics, Data Quality, Information Integration, Semantics.

1. INTRODUCTION

Fact-based strategic decisions of organizations and enterprises are frequently supported by analyzing and interpreting data that is stored in distributed and heterogeneous information sources. To ensure the quality of such decisions, which is directly depending from the quality of an integrated result set, the quality of each participating data source has to be determined previously. Consequently, comparability of heterogeneous data sources has to be enabled.

We introduce a twofold approach for improving information integration with benefits for data science workflows, data analytics or data quality assessment. Firstly, descriptions of the semantics of heterogeneous information systems are extracted from their metadata and information source ontologies are generated semi-automatically. This step is fundamental to establish comparability between the single information sources for further processing, as it produces a homogeneous view on heterogeneous data structures. Secondly, the resulting source ontologies are harmonized by applying ontology similarity measures and automatically integrated to a domain ontology for the entire integration infrastructure. This task enables comprehensive quality assessment across all information sources and provides additional information about existing redundancies.

The homogeneous view is achieved by generating a machine-readable ontology for each individual information source. This process is based on a common vocabulary in combination with a mapping of concepts from different data sources like relational databases, XML, spreadsheets or NoSQL databases, which are presented in Section 2. Furthermore, the generation of an initial ontology and the integration process of several other ontologies is described. For a machine-readable representation of the ontologies Resource Description Framework (RDF), RDF Schema and Web Ontology Language (OWL) are used.

An integration scenario is performed by a hybrid ontology Java implementation for semi-automatically generating ontologies from MySQL databases. For this proof-of-concept and the evaluation presented in Section 3, the focus is on relational databases, since they are still the most widely used data source. Section 4 covers the conclusion and open issues as well as further research activities in this field.

2. SEMI-AUTOMATICALLY GENERATED HYBRID ONTOLOGIES

This section introduces a unified vocabulary for the representation of different types of information sources and the mapping of their concepts to the vocabulary terms. Afterwards, it is explained how the resulting data source ontologies are stepwise integrated to a domain ontology for the integration infrastructure.

Table 1: Specification of information sources by *dsd*

Relational database	XML Schema	Spreadsheets	Cassandra	*dsd* Vocabulary
database	document/namespace	file	keyspace	`Datasource`
relation (table)	simple-, complexType	sheet	table/column family	`Concept`
attribute	attribute, element	column/row header	attribute/column	`Attribute`
relation (table)				`ReferenceAssociation`
relation (table)	extension, restriction			`InheritanceAssociation`
relation (table)	complexType		table/column family	`AggregationAssociation`
primary key	key			`PrimaryKey`
foreign key	keyref			`ForeignKey`
RDB specific data type	XSD data type	spreadsheet data type	CQL data type	XSD data type

2.1 Vocabulary for Data Source Descriptions

The presented approach uses the domain specific vocabulary *description of a data source (dsd)*[1] for the machine-readable representation, which is based on OWL, RDF and RDF Schema. Already defined properties are reused by integrating the World Wide Web Consortium (W3C) recommendations *dcterms*[2], *void*[3] and *foaf*[4].

A data source consists of arbitrary many instances of the class `Concept` that represents real-world objects. Concepts can be related to each other by with instances of the class `Association`, which can **be** further be divided into three subclasses. A `ReferenceAssociation` describes a regular relationship, for example the employment of a person to a company. An inheritance relationship is modeled with `InheritanceAssociation` and an aggregation with `AggregationAssociation`. Properties of concepts and associations are described by the class `Attribute`, which is defined by a `xsd:Datatype`. In order to enable modeling of referential integrity in relational databases (RDB), the classes `PrimaryKey` and `ForeignKey` are implemented, which are assigned to `Concept` and are composed of several `Attributes`. The class `Stakeholder` inherits from `foaf:Agent` and is used to describe departments or persons and their access privileges to data sources or parts of them. Figure 1 depicts the taxonomy of *dsd*.

Relationships between *dsd* classes are described by a set of OWL object properties and data type properties with the prefix `dsd`. A `Datasource` may contain several concepts that are linked by the `hasComponent` property. Concepts consist of attributes, connected by the `hasAttribute` property, and include key attributes expressed by `hasPrimaryKey` and `hasForeignKey`, which in turn consist of attributes and refer to other keys.

Inheritance and aggregation associations are composed of parents and childs (`hasParent`, `hasChild`) or aggregations and their components (`hasAggregation`, `hasAggregationComponent`). They are modeled as classes instead of relations, to describe their completeness and disjointedness with the Boolean properties `isComplete` and `isDisjoint`. An association is disjoint, if all child or component concepts are disjoint. Completeness is given, if all individuals of the child/component concepts are also represented in the parent/aggregation concepts.

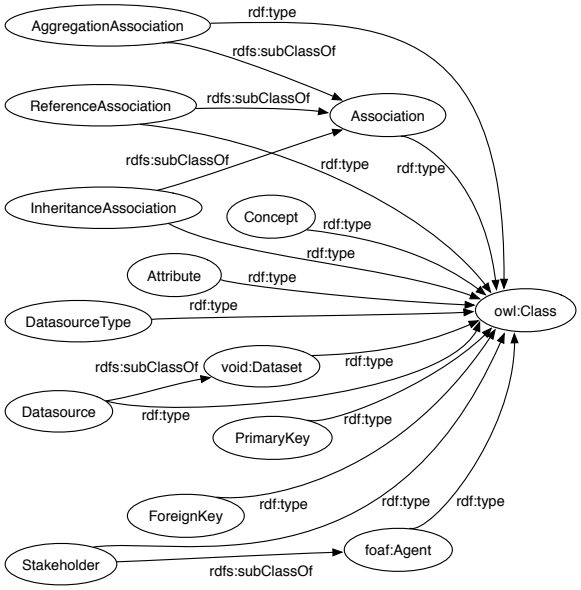

Figure 1: Classes of the *dsd* vocabulary

2.2 Description of an Information Source

For describing the semantics of a data source, classes of the *dsd* vocabulary are instantiated with concepts of the original data model. By using abstract and concrete properties, metadata is applied to those instances. Since the implementation focuses on relational databases, this type of mapping is described in detail. The other data sources depicted in Table 1 were selected according to their relevance during the conceptualization phase of this research project.

A relational database table can not be mapped uniquely to a class of *dsd* as a result of semantic loss. Consequently, it is not automatically decidable if a table in a RDB represents a real-world object (e.g. employee), a reference association (e.g. employee works in company), an inheritance association (e.g. manager inherits from employee modeled in two separate tables) or an aggregation (e.g. table containing company and specific departments). Although a lot of research has already been carried out to solve this issue, no satisfying solution has been proposed so far. Section 3.1 describes the implemented reverse engineering approach with a reliable

[1] http://ehrlinger.cc/voc/dsd [August 18, 2015]
[2] http://purl.org/dc/terms [August 18, 2015]
[3] http://rdfs.org/ns/void [August 18, 2015]
[4] http://xmlns.com/foaf/spec [August 18, 2015]

automatic detection of concepts and reference associations, whereas the cases `InheritanceAssociation` and `AggregationAssociation` require manual rework.

The eXtensible Markup Language (XML) is essential for data interchange purposes. Table 1 includes a mapping for XML Schema Definition (XSD). Documents that do no apply an explicit schema can be analyzed directly, although the quality of the representation might be low.

Spreadsheets as well as comma-separated values files are frequently used for basic import/export functions to exchange structured information between heterogeneous information systems or data sources. In the current development level spreadsheets are treated as a single relational table, which enables a unique mapping to the *dsd* vocabulary.

Finally, NoSQL databases are taken into account by exemplary using Cassandra, which is optimized for storing and processing large amounts of data with high performance requirements. Due to Cassandra's architecture, it can be assumed that reference or inheritance associations do not exist, although denormalized tables probably represent an `AggregationAssociation`.

Denormalized database tables concern not only NoSQL databases and spreadsheets, but also relational databases and entail the risk of redundancy, which is a matter of data source quality and currently main focus in a follow-up research project.

Each information source defines proprietary data types for attributes. In order to achieve comparability between instances of the class `Attribute`, it is necessary to initially map data source specific data types to common XSD data types. For example, Oracle `VARCHAR` is mapped to `xs:string`.

2.3 Generation of the Domain Ontology

In order to integrate several heterogeneous information sources to enable comprehensive data analytics, harmonization and consolidation of participating source ontologies is necessary. To optimize this process, a global reference ontology created by a domain expert would be desirable, but can not be assumed to exist in practice. This leads to the demand of an automatic integration.

The proposed process follows a *Global-as-view* (GaV) approach. According to Lenzerini's tutorial on data integration [4], the global schema is built up on views over the sources when modeling an integration system with GaV. This mapping perfectly supports querying, which is a basic requirement for analytics. The disadvantage of GaV is its inflexibility, because extensions might lead to complex refinements in the global schema. This drawback is acceptable since the automatic recreation of the global ontology is of little effort.

The integration starts with any of the data source ontologies that is initially compared with any other data source ontology, thus, building up the integrated ontology by considering similarities and differences. In the next iterations all remaining ontologies are each compared with the integrated domain ontology. For each resource r_{ds} of a sequentially added data source description, the similarity to each existing resource r_i in the domain ontology is calculated using a suitable similarity measure (described in detail in Section 3). Two resources are considered as equal, if the calculated value exceeds a pre-defined threshold. In this case a new relationship r_i `owl:sameAs` r_{ds} is added to the description of the integration system. If the similarity value of a concept is greater than the threshold and less than 1.0 (but not equal to 1.0), and thus, indicating minor differences, the `Concept` of the data source ontology is investigated concerning `Attributes` (again using similarity measures) that do not yet exist in the domain model. Those attributes are added to the domain ontology, whereas attributes already existing in the domain ontology, but not in data source ontology, remain in the domain ontology and are not removed. If no corresponding resource is identified for r_{ds}, the name and properties of this resource are copied to the description of the integration system and again a link to the original resource is added with `owl:sameAs`.

3. IMPLEMENTATION AND EVALUATION

This section presents the Java implementation of the hybrid ontology architecture. The evaluation of the proposed approach focuses on relational databases, since they are still the most widely used data source. The implementation firstly generates data source ontologies representing the semantics of MySQL databases and secondly, consolidates them to an integrated domain ontology. The implementation is evaluated by performing three test scenarios.

3.1 Ontology Generation

The characterization process of a single relational database is based on reverse engineering, in order to reconstruct semantic information that was lost through forward engineering. In the implementation this task is sub-divided into the following six steps:

1. Metadata about tables, attributes, primary- and foreign keys, which are stored in the data dictionary, are extracted and stored in Java classes that represent the corresponding concepts.

2. Tables are automatically classified according to the reverse engineering approach by Lubyte and Teassaries [5]. This approach identifies base relations (mapped to `Concept`) and relationship relations (mapped to `ReferenceAssociation`), but in some cases no classification can be found due to the constellation of foreign and primary keys.

3. A user is requested to approve the classified tables and assign tables that could not be classified automatically. This step should be performed by a person with knowledge about the local data source and the *dsd* vocabulary.

4. By using the Apache Jena Framework[5], an OWL ontology based on the tables and information obtained from reverse engineering is generated.

5. Finally, information about the completeness of associations can be determined by analyzing the tuples of the

[5]https://jena.apache.org [August 18, 2015]

corresponding association tables. If the aggregation or parent class stores a related tuple (resulting from a primary- foreign-key relationship) for every tuple stored in a component or child class, the property `isComplete` is set `true`, otherwise `false`.

6. The final description of the data source is stored in Turtle[6] notation.

3.2 Integration Process

The automatic establishment of a domain model requires an arbitrary number of data source descriptions as input parameters, given by a human user. This list also determines the interpretation order of the single ontologies. The first data source ontology is therefore considered as basis and completely copied to the integration namespace. All other ontologies are step-wise integrated according to the process described in Section 2.3.

For each concept of a newly added ontology, the similarity to every existing concept in the domain ontology is calculated. At the most basic level, two attributes are compared by using the Jaccard coefficient $J(a,b) = \frac{|a \cap b|}{|a \cup b|}$ over the bit sets a and b. These sets consist of the attribute's properties {`dcterms:title`, `dsd:isNullable`, `dsd:isOfDataType`, `dsd:isAutoIncrement`, `dsd:isUnique`}, where for each comparison of a property a true/false assertion is made. String data types are checked on sub-string occurrences and other data types, like integer, Boolean or XML data types are verified according to their equality.

The similarity calculation between two concepts, associations, primary- or foreign keys is also performed by using the Jaccard coefficient, but those instances are represented by a set of their assigned attributes combined with properties like `dsd:isComplete` and `dsd:isDisjoint` for associations. A threshold has to be defined manually, in order to determine the minimum similarity value still identifying two concepts as equal. The threshold should be set to a value between 0.51 and 0.99, whereas data with lower quality requires a lower threshold.

3.3 Evaluation

The implementation is evaluated by extracting metadata information of a MySQL database (Sakila and Magento), transforming it into an ontology and calculating the similarity between resources of data source ontologies. The selected examples serve as basic proof-of-concept and do not claim to cover all possible occurring problems. In-depth evaluations including larger databases (like ERP, CIM and IRM systems) are planned.

Sakila. Sakila[7] is an official MySQL sample database for the administration of a film distributions. During the evaluation all tables were automatically assigned correctly to their corresponding classes in *dsd*, the most important ones are depicted in Table 2. This perfect result was achieved due to the simple database schema of Sakila.

[6]http://www.w3.org/TR/turtle [August 18, 2015]
[7]http://dev.mysql.com/doc/sakila/en/ [August 18, 2015]
[8]Abbreviation for `ReferenceAssociation`

Table 2: Automatic classification of Sakila

Table	Referenced Tables	*dsd* class
actor		`Concept`
address	→ [city]	`Concept`
category		`Concept`
city	→ [country]	`Concept`
country		`Concept`
film	→ [language, language]	`Concept`
film_actor	→ [actor, film]	`RefAssoc`[8]
film_category	→ [film, category]	`RefAssoc`[8]
language		`Concept`

Magento. Magento[9] is a popular open source software for e-commerce and uses MySQL for data storage. For the evaluation parts with higher complexity (compared to Sakila) of the Entity-Attribute-Value (EAV) system of the database schema version 1.7.0.2 were used in order to evaluate the description of inheritance relationships. Table 3 shows the automatic classification, clearly depicting the expected result, where two tables inheriting from `eav_attribute` could not be assigned to a class automatically. In this case user interaction is necessary in order to manually assign the tables to the corresponding classes.

Table 3: Automatic classification of Magento

Tables	*dsd* class
catalog_category_entity	`Concept`
catalog_category_product	`ReferenceAssociation`
ccatalog_eav_attribute	*not defined*
catalog_product_entity	`Concept`
customer_eav_attribute	*not defined*
eav_attribute	`Concept`
eav_entity_type	`Concept`

Integration. For the evaluation of the integration procedure, two views of the magento database with partially overlapping tables are extracted and represented as ontologies and finally integrated to a domain ontology. As expected, equal concepts are identified as equal, receiving a Jaccard coefficient value of 1.0. No mapping is detected for concepts that appear in only one of both data models, and therefore they are added to the integrated domain ontology. The table `eav_attribute` was slightly modified by removing an attribute in one of both views, and receives therefore a similarity value of 0.9444. Because the value is greater than the threshold set to 0.8, this table is correctly assigned to the original `eav_attribute` by `owl:sameAs`.

4. RELATED WORK

In the last years, several initiatives proposed ontologies as beneficial for information integration. Cruz and Xiao [1] propagate ontologies as well suited for data integration and mention hybrid ontologies as most appropriate approach for Local-as-View (LaV) integration systems. The LaV approach permits modifications in the sources without affecting the global ontology [1], which is an essential requirement of the proposed approach.

[9]http://magento.com [August 18, 2015]

SemWIQ [3] is a generic architecture for integrating different types of data sources based on a mediator-wrapper system in combination with a static mapping influenced by the content level. In contrast, our approach aims at an automatic generation of ontologies focusing the concept level and metadata extraction for increasing expressiveness of the represented model.

Particular attention is paid to relational databases and their reverse engineering, because they are still the most widely used systems for storing data. Although many attempts for reverse engineering a RDB have been proposed so far, a satisfying automatic solution for reverse engineering is not available. Spanos, Stavrou and Mitrou [7] give an overview of techniques for semantically describing RDBs. Common methods like Relational.OWL[10], R2RML[11] or the basic mapping "table-to-class, column-to-predicate" to RDF graphs from Tim Berners-Lee are not sufficient for the proposed approach, because they represent a simple 1:1 mapping without gaining additional knowledge about the data. For enhancing the expressiveness of the model without loss of quality, the semi-automatic reverse engineering process introduced by Euzenat et al. [2] is applied.

Several tools for calculating the similarity of ontologies are available, for example the Java library OntoSim[12] or the Java-based GoodOD Similarity Evaluator[13]. Both tools focus on the comparison of ontology structure and classes, whereas our approach compares individuals. Due to the common *dsd* vocabulary, classes and their properties are represented equally for each ontology and facilitate the integration process with additional domain knowledge. Based on the evaluation of similarity measures in [2] and the availability of detailed domain knowledge about the *dsd* vocabulary, similarity calculation is performed using the Jaccard coefficient, similar to the work of Muthaiyah and Kerschberg [6].

5. CONCLUSIONS

The presented approach describes a semi-automatic semantic description of an information integration infrastructure, considering its heterogeneous and distributed information sources. The final result is provided in machine-readable form, ideally suited for subsequent information processing and data analytics.

In practice, many data sources come with a simple schema without aggregations or inheritance hierarchies. Therefore, a description can be generated automatically in most cases, because the classes `Concept` and `ReferenceAssociation` are classified reliably as proved by the evaluation use cases. The domain ontology, although data quality dependent, definitely provides a structured view with high semantic expressiveness on the entire integration infrastructure. This information is a very important prerequisite for subsequent data processing and data analytics, e.g., redundancy detection, as well as data quality assessment. In order to cope with similarities in structure, we will establish a repository for structure descriptions to define similarities as equivalencies, analogous to synonyms and homonyms defined in thesauri and vocabularies.

There is still potential for further improvements, especially in terms of reverse engineering of relational databases, e.g., considering replacement of substring similarity by a thesaurus and performing in-depth research of structural graph similarity. Data quality assessment was not part of this work, but is currently in the focus of two follow-up research projects, one dealing with schema quality and the other with data quality on tuple level.

[10]http://www.dbs.cs.uni-duesseldorf.de/RDF/relational.owl [August 18, 2015]
[11]http://www.w3.org/TR/r2rml [August 18, 2015]
[12]http://ontosim.gforge.inria.fr [August 18, 2015]
[13]https://github.com/goodod/evaluator [August 18, 2015]

6. REFERENCES

[1] I. F. Cruz and H. Xiao. The Role of Ontologies in Data Integration. *Journal of Engineering Intelligent Systems*, 13:245–252, 2005.

[2] J. Euzenat, C. Allocca, J. David, M. D'Aquin, C. Le Duc, and O. Sváb-Zamazal. Ontology distances for contextualisation. Contrat, INRIA, Aug. 2009. https://hal.inria.fr/hal-00793450 [August 2015].

[3] A. Langegger. *A Flexible Architecture for Virtual Information Integration based on Semantic Web Concepts*. PhD thesis, Johannes Kepler University Linz, 2010.

[4] M. Lenzerini. Data Integration: A Theoretical Perspective. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, pages 233–246, New York, NY, USA, 2002. ACM.

[5] L. Lubyte and S. Tessaris. Extracting ontologies from relational databases. Technical report, In Proc. of the 20th Int. Workshop on Description Logics (DL'07), 2007.

[6] S. Muthaiyah and L. Kerschberg. A Hybrid Ontology Mediation Approach for the Semantic Web. *International Journal of E-Business Research (IJEBR)*, 4(4):79–91, 2008.

[7] D.-E. Spanos, P. Stavrou, and N. Mitrou. Bringing Relational Databases into the Semantic Web: A Survey. *Semantic Web*, 3(2):169–209, Apr. 2012.