DATA INTEGRATION SYSTEM FOR RDF DATA SOURCES

[1]YASSINE LAADIDI, [2]MOHAMMED BAHAJ

[1]PhD. Std., Department of Applied Mathematics and CS, Lab., LITEN, UH1, Morocco

[2]Prof. Faculty of Science and Technology, UH1, Settat, Morocco

E-mail: [1]yassine.laadidi@gmail.com , [2]mohamedbahaj@gmail.com

ABSTRACT

Decision-making systems aim to transform the data stream circulating through the organization to relevant information and knowledge published in the form of dashboards and reports. The Semantic Web (SW) is full of data sources serialized in various formats and extensions (e.g. RDF, OWL, XML, etc) and they are created from scratch or by the transformation of other existing sources (e.g. relational database), and so it became one of the most major data sources that can be used to fulfill the analysis' needs in a decision-making system. The OWL 2 ontology language as a W3C recommendation is built on the RDF data model and used to provide the means for defining and creating structured web ontologies. The purpose of this paper is to propose a new architectural system to perform a data integration process to populate an existing data warehouse using linked data (i.e. data from semantic web) as sources.

Keywords: *Data Integration, ETL, OWL, RDF, Semantic Web, Ontology, Decision-making System, Data Warehouse, Data Sources, Linked Data*

1. INTRODUCTION

A data warehouse (DW) is a subject-oriented, integrated, time-variant and a non-volatile collection of data in support of management's decision making process [1].

Integrated collection of data means that data collected from several sources (e.g. databases, applications, flat files, etc) must be integrated in order to homogenize and give them a unique sense.

Data integration is a set of processes that aim to combine data from disparate sources into meaningful and precious information which includes detection and resolution of schema and data conflicts. Due to multiplicity of data sources, many methods and systems have been developed to integrate data and obtaining a global view of business information across an enterprise. Extraction, Transformation and Loading (ETL) process is a set of tasks grouped into three main categories: an Extraction process in charge of collecting data from heterogeneous data sources, a Transformation process to make data adaptable to the organization of data in the warehouse and a Loading process based on a mapping schema used to load the data in the data warehouse.

One of the greatest advantages of ETL systems is the ability to perform complex data transformations, requiring calculations and aggregations.

In traditional decision-making systems most data sources are usually consistent of relational data bases, information systems or flat files known as structured data, lately semi-structured data sources provided from the Semantic Web (SW) have emerged and represent a new important source for potential relevant information.

The idea of the Semantic Web is to build a web with data that can be processed by machines, and thus, making data interpreted not only by people but also by the machine. To perform such a thing and help machines to recognize data and bring it together, internet users or information producers shall provide metadata (i.e. data that describe other data) which allows a data markup and building logical relationships between data.

The Resource Description Framework (RDF) is the official W3C recommendation for semantic web data models; it is used to decompose any knowledge into small pieces, called triples. A triple is the statement of a binary relation (S.R.S') where S is a subject, S' an object and R will be seen as the kind of relation that exists between subject and object called predicate or property.

It may happen that the same resource is both subject and object and we may find that the same subject is sharing diverse relations which give the appearance of a graph. In semantic web we refer to the things in the world as resources, a resource is identified by Uniform Resource Identifier (URI) that provides a global unique name and serves as means of accessing information describing the identified resource. If two agents on the Web want to refer to the same resource, recommended practice on the Web is for them to agree to a common URI for that resource [2].

OWL which stands for the web ontology language is the data modeling language recommended by W3C to produce RDF data and gives more facilities to define objects and their semantic relationships. OWL is used to describe ontologies in a richer form and made possible to specify cardinalities of object relations and data type properties and to use logical operators in definitions (e.g. use union of classes as a range of relation).

There are many serialization systems developed to write and encode ontologies in different format (e.g. RDF/XML, Turtle, N3, etc), however despite the difference in syntax between those formats, the main and basic building block characterize the RDF model still the same which is the triple pattern Subject-Predicate-Object.

With a large quantity of data, a RDF graph can be stored in a particular database optimized for RDF triples called triplestore and provides a mechanism for persistent storage, access of RDF graphs and query ability.

The query language SPARQL [3] is used to retrieve, manipulate or access ontology sources stored in resource description framework format and allows users to write queries against them without concern about the type of serialization.

The SW introduces a new data model and powerful technologies for data integration. In our previous work [12], we presented an innovative method for designing a data warehouse based on an ontology schema; however, the data integration phase was not treated and existing approaches does not deal with complex transformation in the ETL process. The main goal of this paper is to complete [12] and presenting a new architecture system to perform a data integration process that populate an existing data warehouse using linked data (i.e. data from semantic web) as sources.

Our approach consist of gathering together all data from different sources into a single place which is the staging area materialized by the triplestore, and generating a global ontology schema (i.e. inspiring from GaV paradigm) describing the data stored and enables to SPARQL queries extracting useful data into temporary relational tables before finally performing all transformations and computations needed and loading data finally into DW (see figure 1).

Generally, the system will be based on his treatment on three main parties:

1) Extracting data from different sources into a data warehouse.
2) Performing transformations on RDF data and using a matching and merging process.
3) Extracting data using SPARQL queries and loading into temporary relational tables.
4) Performing complex transformations on data stored in temporary tables before loading finally into DW.

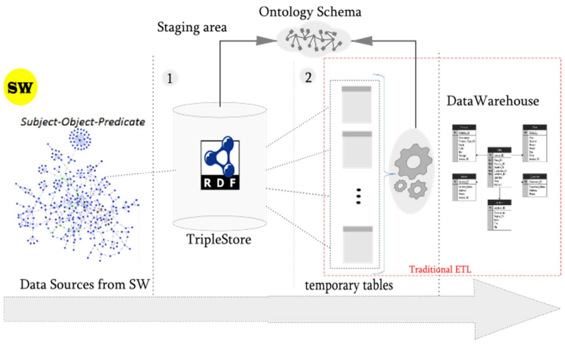


Figure 1 : Data Integration Process

This paper is organized as follows: in section 2, an overview is presented. We briefly review related works in section 3. We treat the ontology extraction and fusion process in section 4. In section 5, creation of relational database tables form the ontology and loading process is explained. Finally, in section 6 a conclusion is given.

2. OVERVIEW

The Resource Description framework (RDF) is a framework for representing information about “things” or resources in a graph form. RDF aims to facilitate the automatic processing of information from the web via software agents.

The basic structure of all RDF expressions is a collection of triples; each triple is composed from a subject, a predicate and an object. RDF schema (RDFS) allows creating a vocabulary using the RDF data model and describing relations between subjects and objects in RDF triples (i.e. nodes in RDF graph).

However, large limitations confine the capacity of expression of knowledge established by RDF schema. To overcome those limitations, the Web Ontology Language OWL [4] recommended by W3C is designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL ontology language is divided in three sub-languages: a) OWL Lite: the simplest OWL sub-language for users who need to express taxonomy and simple constraints, such as 0 and 1 cardinality. b) OWL DL: based on the Descriptive Logic (DL), allows a much greater expressiveness compared to OWL Lite. c) OWL Full: has no expressiveness constraints and does not guarantee any computational properties (e.g. Classes can be instances or properties at the same time). OWL DL and OWL Lite do not allow classes to be used as individuals unlike OWL Full. In this section we will focus in OWL DL language.

Description Logic (DL) represents a family of logic based knowledge representation formalism used to describe a domain in term of concepts, roles and individuals (see figure 2).

The knowledge representation system, based on DL, consists of two components: a Tbox and an Abox.The Tbox refer to the vocabulary of an application domain and consists of a set of axioms, for example:

$\{$

$\text{Doctor} \sqsubseteq \text{Person},$
$\text{HappyParent} \equiv \text{Person} \sqcap \forall\text{hasChild.}(\text{Doctor} \sqcup \exists\text{hasChild.}$

$\}$

The Abox contains assertions about individuals in terms of the Tbox (i.e. vocabulary), for example:

$\{$

$\text{Said:HappyParent},$
$\text{Said hasChild Khalil}$

$\}$

Based on DL, OWL DL language defines several constraints that disallow certain uses or combinations of constructs in order to retain decidability. A concept C from DL is referred to a class in OWL, a role r from DL is referred to a property in OWL and an individual i is represented in OWL as an instance of a class. For example:

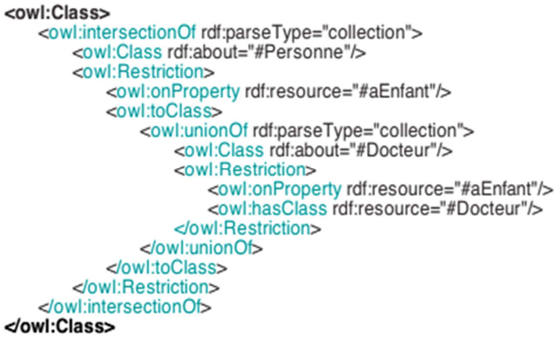
Let Personne and Docteur be concepts and aEnfant be a role:

$\{$

Personne
$\sqcap\ \forall\text{aEnfant.}(\text{Docteur}\exists\text{aEnfant.Docteur})$

$\}$

This sentence is represented in OWL DL as shown in figure 3 below:



```
<owl:Class>
    <owl:intersectionOf rdf:parseType="collection">
        <owl:Class rdf:about="#Personne"/>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#aEnfant"/>
            <owl:toClass>
                <owl:unionOf rdf:parseType="collection">
                    <owl:Class rdf:about="#Docteur"/>
                    <owl:Restriction>
                        <owl:onProperty rdf:resource="#aEnfant"/>
                        <owl:hasClass rdf:resource="#Docteur"/>
                    </owl:Restriction>
                </owl:unionOf>
            </owl:toClass>
        </owl:Restriction>
    </owl:intersectionOf>
</owl:Class>
```

Figure 2: Example Of OWL-DL Representation

A triplestore (i.e. RDF store) is typically accessed using a query language. Many query languages and RDF engines are available supported by some RDF-based products or open-projects.

One of most well-known RDF query language is called SPARQL.

Designed by W3C, the query language SPARQL is used to retrieve, manipulate and access RDF data sources stored in resource description framework format; it is implemented by all major RDF stores and shares many features with other query languages like SQL query language.

3. RELATED WORKS

In traditional decision-making system, the data treated during the ETL process is provided from relational databases or very structured spreadsheets. However, information is not always structured; the emergence of semi-structured data sources such as XML, RDF, OWL and other semi-structured format emphasizes the need to develop new approaches for treating semantic data (i.e. data from SW) especially for data integration processing.

In [5], propose a formal way for driving a conceptual ETL design, based on the well-established graph transformation theory, the idea is to perform lightweight transformations on ontology.

The difference is that in our approach, transformations are performed on the data stored in temporary relational tables not on the OWL ontology which allow us to make complex transformations.

Authors in [6] proposed a method for on-demand construction of OLAP cubes for ROLAP systems, the method focus on populating OLAP cube by extracting data from its RDF form by queries that are generated using the ontology of the OLAP schema.

In [7] proposed to integrate data based on global-as-view (GaV) approach by integrating all data sources into one global ontology schema and then generating the corresponding multidimensional model before finally populating it. However, those methods couldn't be used to integrate data into in extent instance of a data warehouse.

In this paper we present an approach to integrate semi-structured data provided from semantic web into an existing data warehouse by splitting the staging area into two parties and enabling more complex data transformations.

4. THE STAGING AREA

ETL process is used as a combination of processes and technologies for the extraction, preparation and loading of data into the DW, thus, it is necessary to make a conceptual ETL design to identify the sources and destinations of each entry. In practice and during a traditional ETL project, data is loaded firstly in a separate relational database called the staging area. The main goal of the staging area is to preparing the data and making all transformations and computations needed before loading data into the final target which is the DW.

Dealing with the issue of integrating data from SW sources, our approach consists of dividing the traditional staging area into two parts as shown in figure 1.

Unlike relational database management systems (RDBMS), which store data in relational tables and are queried using SQL, a triplestore is a database management system (DBMS) that store RDF triples and is queried using SPARQL. Some of the advantages of triplestores are facilitating the integration of multiple data sources particularly semi-structured data and allows the discovery of implicit knowledge through inference mechanisms. It's much recommended to use a native triplestore and not reusing the storage and retrieval functionalities of other database management systems, It is also preferable to have an RDF store and query engine that retain their performance even in the face of very large data sets[2].

Based on the principle feature of the RDF data model which is the triple <Subject, Predicate, Object>, RDF data will be extracted from all sources and gathered together in the triplestore using matching and merging techniques and dealing with concepts and properties similarities issues.

The process begins by selecting RDF data sources needed and extracting corresponding graphs of triples, all triples are listed and put directly in the local triplestore located in the staging area. Since merged information from two graphs or more is as simple as forming the graph of all of the triples from each individual graph [2], the advantage of listing all triples is that allows implicit identification and matching of identical and similar (i.e. in a linguistic point of view) concepts during the rebuilding phase, for example: if the same entity is located in several ontologies (i.e. RDF graphs) with the same name (or URI), then by gathering together all those classes we will have as result a unique node.

The matching process can be performed by the user in a manual manner by selecting and specifying form all triples stored entities that are jugged similar and make all modifications needed and updating the local triplestore. In this case, the user of the system may have capability to add more modification on triples stored manually according to his business needs.

The second alternative is performing an automatic or semi-automatic method and comparison algorithms between existent entities saved and stored in the triplestore.

Many tools have been proposed to perform automatic or semi-automatic matching and merging methods on RDF triples. In [8], the system introduce a multi-matching technique as first process to find correspondence between entities by extracting matched concepts as first iteration, extracting matched properties of matched concepts as second iteration and extracting matched values of matched properties as last iteration, and a merging technique as a second process.

The matching process is based on two major techniques: a string method for searching and compare identical terms and another method for searching identical meaning filtered by user, the process begin by extracting matched concepts as first iteration, extracting matched properties of matched concepts as second iteration and extracting matched values of matched properties as last iteration.

In [9] an algorithm that provides a semi-automatic approach to ontology merging and alignment is presented, [9] algorithm aims to guide user in the creation of one merged ontology from two ontology sources by generating suggestions based on linguistic-similarities matches and indicate to the user conflicts and effects and possible solutions for those conflicts.

Other transformation may take place to simplify and optimize the triplestore by executing a simple algorithm on concepts and properties in separately way, first the algorithm treat concepts (i.e. classes in OWL language) by searching and replacing class names stored in the triplestore by their similar in a way to have a unique node describing a class not two.

The second part of the algorithm treats object and data type properties by eliminating redundancies.

The process is based on a selection of all object and data properties with their corresponding domains and ranges. If two or more properties have the same domain and the same range only one will be kept and the rest removed.

Class expressions and other modeling capabilities with properties as the inverse property, symmetric/asymmetric properties, transitivity, reflexivity, etc, are kept but not used or treated during this phase except for modified resources related to some class expression, the raison is to provide precise information about triples to support inference and reasoning on the data during the next phase of the process.

As a final operation, a global OWL ontology describing concepts and properties (i.e. only Tbox data) stored in the triplestore should be generated.

The global ontology contains a global data model plus construct mapping information[10], thus, it will be used to map sources (i.e. data from the triplestore) to the targets which are temporary tables (i.e. relational tables) used to store data as shown in figure 1.

5. CREATING AND POPULATING TEMPORARY RELATIONAL TABLES

Based on the global ontology generated which is describe classes, object properties and data type properties saved in the local triplestore, the next step of this approach consist of creating the corresponding relational database (RDB) tables as a second transfer zone. Those tables are temporary, and used to make all transformations needed before finally loading data into the DW. Some efforts that aim to create and populate relational database from OWL ontology documents have been already made and may be used to define RDB tables.

In [11], algorithms for transformation of ontology to relational database are proposed. The first algorithm begin by transforming OWL classes into relational database (RDB) tables, when OWL classes are mapped to tables, object properties are transformed into RDB relations, data-type properties are transformed into RDB data columns and ontology constraints are transformed into relational database metadata tables.

Authors in [12] present an automatic approach and a set of techniques to map OWL data to a relational schema, the OWL2DB algorithm proposed take as input an OWL document and create the corresponding relational database with preserving the constraint information while the mapping process. As a final result, corresponding table names, attributes and values of the instances are collected to populate the database.

Another alternative is to construct relational tables based on user specifications and populate them with data provided from the local triplestore by executing batch of SPARQL scripts (see figure 3).

In this case, the user of the system must take on consideration the schema of the global ontology (i.e. global Tbox) that describe the local triplestore and the schema of the data warehouse for identifying and modeling the temporary RDB tables.

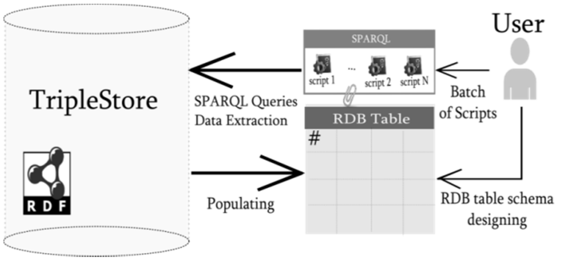


Figure 3: Process of loading data into RDB table using SPARQL queries.

All SPARQL scripts executed in this phase are aim to only to extract data from the local triplestore and to populate specific attributes located in a RDB table.

Once the extraction process is done and all data sources are stored in the RDB tables, the user may perform all traditional transformations and computations needed (e.g. filtering, sorting, aggregations, joining, etc.) by selecting the temporary RDB as a structured data source.

6. CONCLUSION

The Semantic Web (SW) is become one of the most major data sources that can be used to fulfill analysis needs in a decision-making system. To take advantage of this new wave, the domain of business intelligence must be enrich by new approaches and methods dealing with semi-structured especially RDF data.

The main goal of our work is to complete our prevous works and provide a new architecture system to perform data integration process. the objective of our approach consist of gathering together all data from different sources and formats (RDF/XML, RDF, OWL, etc) in the staging area materialized by the triplestore. In this point, matching and merging methods are performed to generate a unique global ontology schema.

Based on the global ontology schema (i.e. inspiring from GaV paradigm) that describe data stored in the RDF data store, series of SPARQL queries are executed to extract useful and relevant data into temporary relational tables before finally performing all transformations needed (sorting, filtering, aggregations, etc) and loading data finally into a DW.

REFRENCES:

[1] W.H. Inmon, Building the Data Warehouse, 3rd edition, John Wiley, New York, 2002.D.Allemang, J.Hendler, Semantic Web for the Working Ontologist, Effective Modeling in RDFS and OWL, second Edition, May 2011.

[2] World Wide Web Consortium, SPARQL 1.1 Query Language, Web source: http://www.w3.org/TR/sparql11-query/, 2013.

[3] World Wide Web Consortium, OWL 2 Web Ontology Language, Web source: http://www.w3.org/TR/owl-quick-reference/, 2012.

[4] D. Skoutas, A. Simitsis, and T. K. Sellis, “Ontology-Driven Conceptual Design of ETL Processes Using Graph Transformations,” J. Data Semantics, vol. 13, pp. 120–146, 2009

[5] M. Niinim¨ aki and T. Niemi, “An ETL process for OLAP using RDF/OWL ontologies,” J. on Data Semantics XIII, vol. 5530, pp.97–119, 2009.

[6] O. Romero, A. Simitsis, and A. Abell ´ o, “GEM: Requirementdriven generation of ETL and multidimensional conceptual designs,” in 13th Int. Conf. in Data Warehousing and Knowledge Discovery (DaWaK), ser. Lecture Notes in Computer Science, vol. 6862. Springer, 2011, pp. 80-95

[7] Susan F Ellakwa, El-sayed El-azhary and Passent El-kafrawy.,(Integrated Ontology for Agricultural Domain. International Journal of Computer Applications 54(2):46-53, September 2012.

[8] N. Noy and M. Musen. 2000. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment, Proc. of17th National Conference on Artificial Intelligence,(AAAI,), pp. 450–455, Austin, Texas.

[9] Priebe, T., Pernul, G.: Ontology-based Integration of OLAP and Information Retrieval. Proc. of the DEXA 2003 Workshop on Web Semantics (WebS 2003), Prague, Czech Republic, September 2003.

[10] Ernestas Vysniauskas, Lina Nemuraite.: Transforming Ontology Representation from Owl to Relational Database. Information Technology and Control, 2006, Vol.35, No.3a.

[11] Anuradha Gali, Cindy X. Chen, Kajal T. Claypool and Rosario Uceda-Sosa.: From Ontology to Relational Databases. Information Technology and Control, 2006, Vol.35, No.3a.

[12] Yassine LAADIDI, Mohamed BAHAJ Designing a Data Warehouse from OWL sources. International Journal of Soft Computing and Software Engineering, Vol.5, No.3, 2015.