Ontology Extraction Utilizing Augmented Direct Mapping for Movies Data Representation

Jiawei Wilson May[1], Su-Cheng Haw[1], Samini Subramaniam[1], Muhsin Hassan[2] and Fatimah Almah Saaid[2]

[1]Multimedia University, {wilsonmay.mjw@gmail.com, sucheng@mmu.edu.my, samini.subra@mmu.edu.my}
[2]Telekom Malaysia, {muhsin.hassan@gmail.com, almah@tm.com.my}

ABSTRACT

The volume of information is growing exponentially in today's world. Nevertheless, the huge data are stored in various type of formats at various sizes and various growing rate. In addition, these data are heterogeneous and not directly consumable by the analytic tools. These resulted as one of the main challenges for data integration. In order to facilitate data integration, many mapping tools have emerged. In this paper, some recent mapping tools will be investigated, and the rules specified to acquire the ontology from relational databases will be proposed and discussed. Throughout the paper, we utilize the Internet Protocol television (IPTV) domain as the example to demonstrate the mapping of individual concepts of Relational Database (RDB) into ontology representation.

Keywords: ontology extraction, data integration, ontology, relational database to ontology mapping, data mapping.

I INTRODUCTION

Towards the recent years, the data is growing faster in the exponential rate. According to IBM, by the year of 2020, every human will creates about 1.7 megabytes of new information in every second (IBM, 2017). It is produced in every seconds for every social media exchanged and any digital processes where by the data are usually being transmitted either by sensors, systems, or mobile devices. Nevertheless, vast data is still published in formats that are not able to be directly process by the existing analytic tool. In addition, the data is coming to us in structured, semi-structured and unstructured form; this making it difficult to be stored and to extract the insight knowledge. It will bring an advantage to the world-wide web user, if the semantic meaning of this huge data can be accurately represented into a conceptualization model. In addition, it can also be used in intelligence analytic to solve problem.

Data comes to people in different ways in either structured, semi-structured or unstructured form, this resulted the difficulty to extract the unified view of the information. To work on data integration, one of the most challenging parts is "heterogeneity" which involve various strategies for coping with the important and complex concept. In order to achieve semantic data integration, a mapping languages or defined rules that work on the mapping from heterogeneous data structures and machine readable ontological model is needed.

On the other hand, semantic data integration is the process of using a conceptual representation of the data and of their relationships to eliminate possible heterogeneities. The heterogeneity of data sources can be in syntax, schema, or semantics, thus, it become the challenges in semantic data integration. In order to resolve the semantic conflict, it involves the use of ontology. Ontology is a knowledge representation that can be shared establishing a shared vocabulary for different applications (Drumon & Girardi, 2008). Besides that, it can represent the database information structure using the Web Ontology Language (OWL) or Resource Description Framework (RDF) language which is useful in Semantic Web (Martinez et al., 2012). In addition, the primitives of ontology consist of the (1) a set of strings that describe lexical entries for concepts and relation; (2) a set of concepts; (3) a taxonomy of concepts of inheritance; (4) a set of non-taxonomic relation; (5) a hierarchy of relations; (6) relation of functional composite that relate concepts and relationship; (7) A set of axiom to describe additional constraints (Maedche & Staab, 2005).

II RELATED WORK

Ontology extraction is a "process of acquiring (constructing or integrating) an ontology (semi-) automatically" from various data sources (Petasis et al., 2011). Many researchers are using this word interchangeable with "ontology construction" or "ontology learning" (Sanchez & Moreno, 2008). Relational databases and XML databases are the most common data types that are widely studied in this area (Touma et al., 2015). In addition, there are other data source type such as Comma-separated values (CSV), spreadsheets, and JavaScript Object Notation (JSON) are being studied (Dimou et al., 2014).

The existing ontology learning tools is being grouped into Direct Mapping (DM), Augmented Direct Mapping, and Domain Semantics-Driven Mapping (Michel et al., 2014). This paper focuses on the Augmented DM group. Some of the existing approaches under this grouping are Relational.OWL (de Laborda & Conrad, 2005), RDBToOnto (Cerbah, 2008), DB2OWL (del Mar Roldan-Garcia & Aldana-Montes, 2008).

Direct mapping is an automatic mapping approach that automatically converts the relational data into a local ontology. It creates the RDF Schema (RDFS) or Web Ontology Language (OWL) vocabulary follow some simple rules defined by Tim Berners-Lee (Berners-Lee, 1998) to automatically create the Uniform Resource Identifiers (URI) based on the RDB schema and data. It is advantage of its simplicity to understand and the rapid creation of a direct representation of RDB schema with low semantic interoperability.

The Augmented DM automatically detect the design pattern of the database to express of the domain semantic. Some semi-automatic approach proposed an iterative process for end-user to validate or dismisses the proposed mappings.

Domain Semantics-Driven Mapping (DSDM) is develope to overcome the limitations of the insufficient semantic and description of a domain extracted from database. Mapping description languages is being used in the DSDM approach to bridge the gap of concept between the RDB and RDF. There are strategies to construct the mapping description languages.

On top of the existing tool implemented, Relational.OWL is designed based on the motivation of schema and data sharing between volatile distributed databases. OWL based ontology format is used as the representation technique. DM rules is used in this tool. Besides, it take the advantage of relational data to build the ontology. Relational.OWL is allowed for monitoring the relationship between the original database and the produced ontology.

W3C on September 27, 2012 released two official recommended standard DM (Das et al.,2012) and R2RML (RDB to RDF Mapping Language). DM provides an approach that transform the content of a RDB into RDF that represent the relational schema in the ontology.

The common issues of rule-based ontology extraction from input data source are (Astrova, 2009): (1) Loss of data: The original data should be described correctly in the result; (2) Loss of semantics: In some situation, the relational database cannot be mapped to certain ontology and the loss of information will be happened. Therefore, the quality of the transformation should be analyzed. (3) Focus on structures: Besides the mapping of database schema structures, the mapping of data should be in the mechanism. (4) Focus on data: Data should be mapped with the incorporation of data types. (5) Correctness: The ontology extraction should have certain of correctness.

The limitation of current ontology extraction approaches is they work only on a single data source and provide a different method for different data models. Several solutions exist to execute mappings from different file structures and serializations to RDF. For relational databases, different mapping languages beyond R2RML are defined (Hert et al., 2011) and several implementations are existed.

For RDB, schema information and instances value are the source for ontology extraction. It involves converting tables, columns and constraints into OWL representation as described in the next section.

III PROPOSED SOLUTION

This paper proposes an enhanced Augmented Direct Mapping by extending the mapping rules with knowledge derived from *relational database knowledge* and *domain data knowledge* to represent the data more accurately. The output representation of the ontology is represented in OWL language, which is widely used in ontology authoring with well defined semantics and high query answering performance. In the ontology extraction process, the conversion can be divided into three main parts.

The first part of the ontology process is the database component reading. It is to extract the database schema and relational metadata. In order to connect to the database, JDBC driver (java.sql.DatabaseMetaData) is act as a bridge to access the database component, including data tables, data columns, primary keys, foreign keys and metadata information.

Conversion between relational database and ontology is follow defined mapping rules or principles. Mapping rules are set to define the elements of the ontology that generated from each of the database component. The approach used in this study map a relational database to ontology by using the names of constructs of the relational database as the names of constructs of the ontology. The mapping rules and principle defined are cater of the *relational database knowledge* and *domain data knowledge* (the use of the data in the particular domain) which is usually lack of in the existing ontology learning tool (Mallede et al., 2013). Additionally, the subclass can be discovered in the database which the table contains tuples with repeating values in attributes.

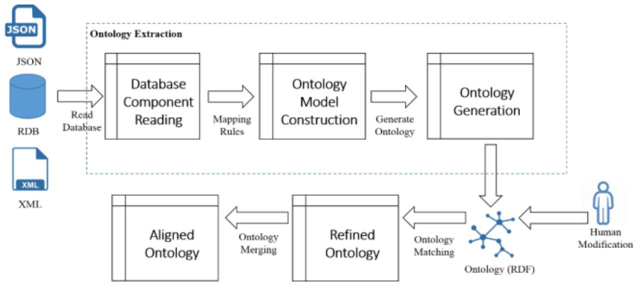


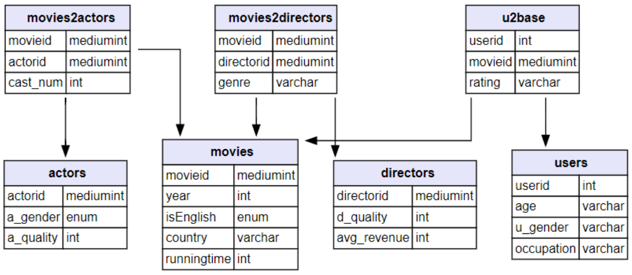
Figure 1. The Proposed Solution Architecture

After required database component is read, the ontology metadata model (GraphModel) is constructed based on the defined mapping rules. It consists of the (1) set of nodes: contain the concepts of property and name; (2) set of edges: contain the relationship/connection between the nodes.

Throughout this research, we use movieLens 100k dataset obtained from GroupLens Research (MovieLens, 2016). It is a stable benchmark dataset which consist of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies, and the data contain the simple demographic information (age, gender, occupation, zip) of each users. The data sets has been collecting in for a periods of time and the relational data source format is available in CTU Prague Relational Learning Repository (PRLR) that can be accessed at https://relational.fit.cvut.cz/(Schulte & Khosravi, 2012)

The database includes demographic information about movie users and the ratings given to the movies (see Figure 2). The first column in each table is the primary key (PK) and columns with (FK) names are foreign key. In PRLR, it also provides the meta-data about the datasets.

These rules will be illustrated by a case study, which the data source used is provided by GroupLens Research. Relational data will be served as the example to illustrate the proposed ontology generation. The mapping rules that are used in this research (Telnora, 2010), (Zhang & Li 2011).



movies2actors	
movieid	mediumint
actorid	mediumint
cast_num	int

movies2directors	
movieid	mediumint
directorid	mediumint
genre	varchar

u2base	
userid	int
movieid	mediumint
rating	varchar

actors	
actorid	mediumint
a_gender	enum
a_quality	int

movies	
movieid	mediumint
year	int
isEnglish	enum
country	varchar
runningtime	int

directors	
directorid	mediumint
d_quality	int
avg_revenue	int

users	
userid	int
age	varchar
u_gender	varchar
occupation	varchar

Figure 2. MovieLens dataset in relational schema (Schulte & Khosravi, 2012).

MovieLens relational database schema is as shown in Table 1. The table is consist of the schema information including the relations, primary keys and foreign keys.

The mapping follow the defined rules by automatically create the vocabulary and the OWL representation from the knowledge of input source. Besides, complex mapping is generated by the interaction of users to generate SQL queries for mapping. The naming of classes and properties are manually named by the users.

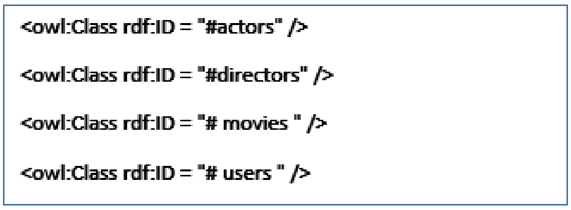
Table 1. MovieLens Relational Database Schema.

Relation	Primary Key	Foreign Key
actors(actorid, a_gender, a_quality)	actorid	
directors(directorid, d_quality, abg_revenue)	directorid	
movies(movieid, year, isEnglish, country, runningtime)	movieid	
users(userid, age, u_gender, occupation)	userid	
movies2actors(movieid, actorid, cast_num)	movieid	actorid, movieid Refers to relation actors, movies actorid, movieid
movies2directors(movieid, directorid, genre)	movieid	directorid, movieid Refers to relation directors, movies directorid, movieid
u2base(userid, movieid, rating)	userid	userid, movieid Refers to relation users, movies userid, movieid

Rule #1: Integrating Same Concepts Tables to one OWL Class
Two tables that contain of similar attributes/properties shall be integrated to form into one ontological class.

Rule #2: Mapping of Tables to OWL Classes

Ontology OWL class can be created from the relation (tables) from the relational database. The condition applies is when the table can be used to represent an independent identity. Based on rule#1, actors, directors, movies and users OWL class is created. It fulfiled the condition as the they got only one attributes as primary key.



```
<owl:Class rdf:ID = "#actors" />

<owl:Class rdf:ID = "#directors" />

<owl:Class rdf:ID = "# movies " />

<owl:Class rdf:ID = "# users " />
```

Rule #3: Handling of Bridge Tables

Bridge table are the table that the relationship as its foreign keys are from the tables participating in a many-to-many relationship, and one or more than one foreign keys as also the primary key. This tables cannot created as separate OWL class whereby will be created as object property (owl: Object property) as it represent the relations between two entity (ontology concepts). In addition, these association relations among relationship can be represented by one-to-one, one-to-many, and many-to-many relations. However, most of the time bridge table is hardly be detected (in this example, movies2actors, movies2directors, u2base).

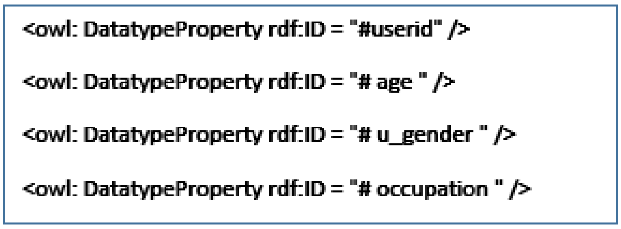
Rule #4: Mapping of Referential Integrity Relationships to Inheritance Hierarchy

Classes are arranged in hierarchy by identifying the relations between the tables. The hierarchy of classes will be mapped based on the foreign key of the table. Table with a foreign key of other table primary key will represented as the sub class of corresponding class. In this MovieLens scenario, there is no any sub-class can be mapped into a main class because there the foreign key are participating in a many-to-many relationship (bridge table).

Rule #5: Mapping of Non-Referential Integrity Columns into Datatype Properties

In OWL ontology, datatype propery (owl: DatatypeProperty) is created to represent the class instances. It links individuals to data values, and represent the class instance.

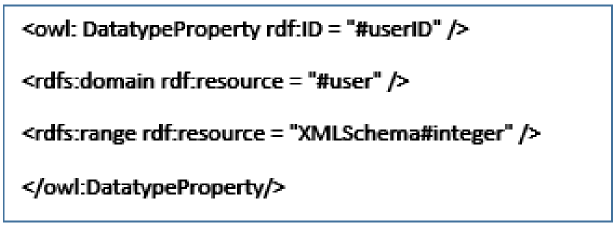
Attributes that in the target RDB, that which cannot be transferred to an object property (owl: Object property), can be transferred data type property (owl: DatatypeProperty). For instance, in the User class, the Datatype Properties are userid, age, u_gender, and occupation.



```
<owl: DatatypeProperty rdf:ID = "#userid" />

<owl: DatatypeProperty rdf:ID = "# age " />

<owl: DatatypeProperty rdf:ID = "# u_gender " />

<owl: DatatypeProperty rdf:ID = "# occupation " />
```

Rule #6: Representation of Datatype Property as Domain and Data Type as Range

Domain and range in the datatype property is used to represent the data type of the RDB attributes. Example as below, user as the domain and XMLSchema#integer as range.

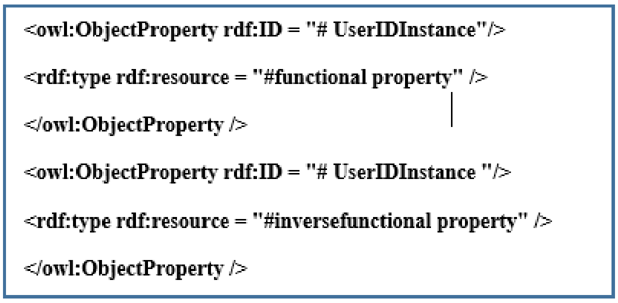


```
<owl: DatatypeProperty rdf:ID = "#userID" />

<rdfs:domain rdf:resource = "#user" />

<rdfs:range rdf:resource = "XMLSchema#integer" />

</owl:DatatypeProperty/>
```

Rule #7: Mapping of Relationships (Foreign key) into Object Properties

The attributes as the foreign key that form a relationship between table in target RDB will mapped into Object Properties. One object property for the relationship and another one for its inverse.

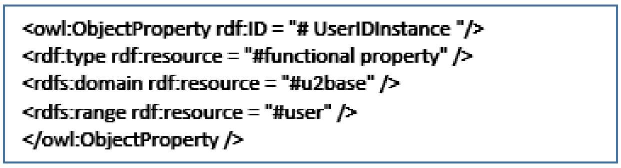
In this example, two Object Properties will be created between the classess between "user" and "u2base". It is represented by a by a UserID Functional Property within the "u2base" class and a UserID Inverse Functional Property within the "user" class.



```
<owl:ObjectProperty rdf:ID = "# UserIDInstance"/>

<rdf:type rdf:resource = "#functional property" />

</owl:ObjectProperty />

<owl:ObjectProperty rdf:ID = "# UserIDInstance "/>

<rdf:type rdf:resource = "#inversefunctional property" />

</owl:ObjectProperty />
```

Rule #8: Representation of Object Property

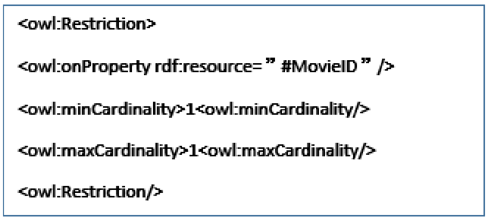
The domain and range in the Object Property is used to represent the relationship between 2 classes. As example below, the domain is u2base(with a functional property) ; a range is user (with an inverse functional property).



```
<owl:ObjectProperty rdf:ID = "# UserIDInstance "/>
<rdf:type rdf:resource = "#functional property" />
<rdfs:domain rdf:resource = "#u2base" />
<rdfs:range rdf:resource = "#user" />
</owl:ObjectProperty />
```

Rule #9: Mapping of Column Constraints into Property Cardinalities

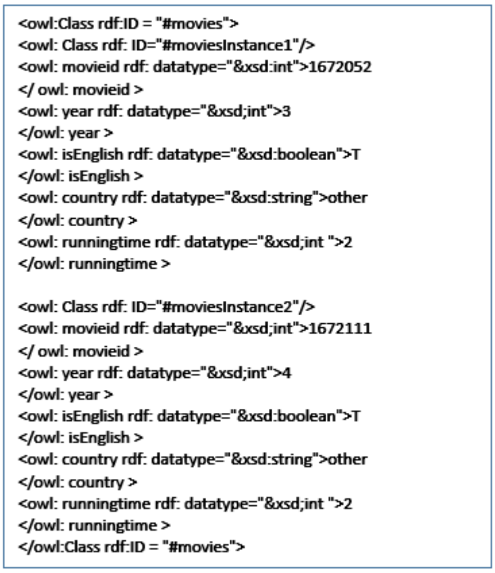
The constraints of database attributes are mapped into Ontology Property (OWL:OnProperty) Cardinalities. According to the rule, movieid as a primary key in movie table which declared as NOT NULL, minCardinality is 1 and maxCardinality is also 1.



```
<owl:Restriction>

<owl:onProperty rdf:resource=" #MovieID " />

<owl:minCardinality>1<owl:minCardinality/>

<owl:maxCardinality>1<owl:maxCardinality/>

<owl:Restriction/>
```

Rule #10: Mapping of Tuples to Individuals

All row of the relational data are mapped to individuals in OWL ontology. As example, movieid "1672052" and "1672111" instances are mapped to individuals.



```
<owl:Class rdf:ID = "#movies">
<owl: Class rdf: ID="#moviesInstance1"/>
<owl: movieid rdf: datatype="&xsd:int">1672052
</ owl: movieid >
<owl: year rdf: datatype="&xsd;int">3
</owl: year >
<owl: isEnglish rdf: datatype="&xsd:boolean">T
</owl: isEnglish >
<owl: country rdf: datatype="&xsd:string">other
</owl: country >
<owl: runningtime rdf: datatype="&xsd;int ">2
</owl: runningtime >

<owl: Class rdf: ID="#moviesInstance2"/>
<owl: movieid rdf: datatype="&xsd;int">1672111
</ owl: movieid >
<owl: year rdf: datatype="&xsd;int">4
</owl: year >
<owl: isEnglish rdf: datatype="&xsd:boolean">T
</owl: isEnglish >
<owl: country rdf: datatype="&xsd:string">other
</owl: country >
<owl: runningtime rdf: datatype="&xsd;int ">2
</owl: runningtime >
</owl:Class rdf:ID = "#movies">
```

Lastly, the transferring of relational data to ontology instance is according to the OWL ontology constructed. Next, the system will record the mapping result and generate the OWL document files. The files can encapsulates all mapping results into a standard input and evaluated through Protégé tool. Figure 3 shows the visualization of the ontology generated using Protégé tool.

In the preliminary stage, the coverage quality of the ontology created is being assessed by the reflection of the query result with manual verification. The ontology itself can be able to retrieve and manipulate in the ontological format by using the query language, SPARQL. These queries are illustrated as below:

Query 1: To find total number of movies which has been watched group by occupation.

Figure 4 depicts the returned result. From the result, it can be seen that MovieLen user who watch most movie is under the occupation categorized as '2'.

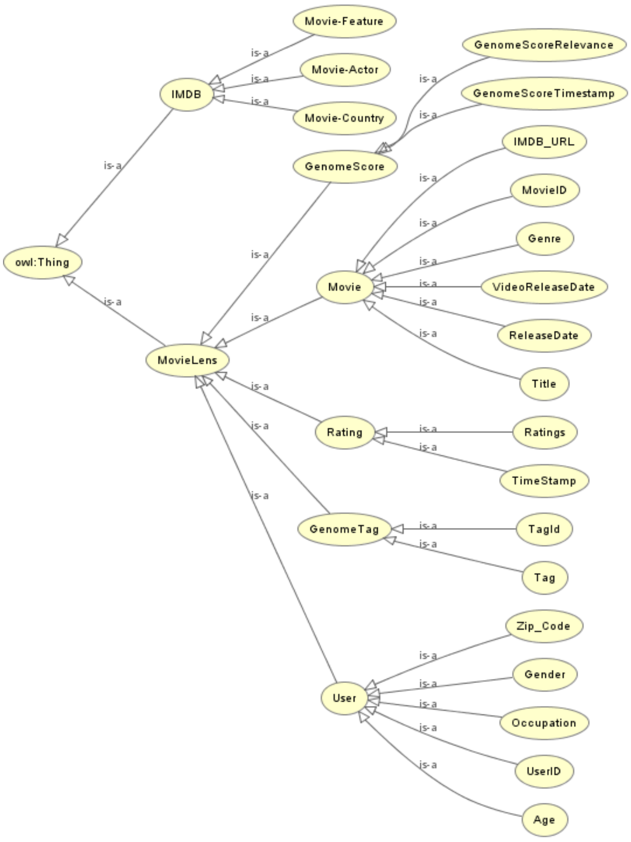
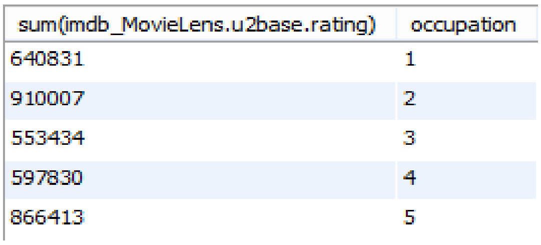


Figure 3. MovieLens Ontology (MovieLens, 2016).

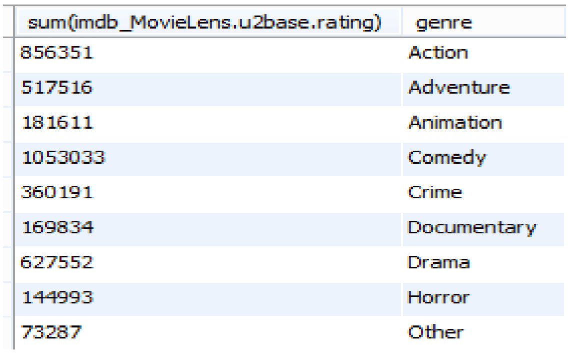


sum(imdb_MovieLens.u2base.rating)	occupation
640831	1
910007	2
553434	3
597830	4
866413	5

Figure 4. Returned Result of Query 1.

Query 2: To find all the genre of movies and their respective ratings rated by the users.

Figure 5 depicts the returned result. From the result, it can be seen that MovieLen user likes the genre 'Comedy' the best, and least prefer on 'Other'..



sum(imdb_MovieLens.u2base.rating)	genre
856351	Action
517516	Adventure
181611	Animation
1053033	Comedy
360191	Crime
169834	Documentary
627552	Drama
144993	Horror
73287	Other

Figure 5. Returned Result of Query 2.

IV CONCLUSION AND FUTURE WORK

The target of the proposed solution is aim to find a method for automatic generate ontology from the schema and instances needed in ontology construction to improve the efficiency of the construction. In the proposed method, the developed algorithm access the database metadata model using the relational database driver and map it into the OWL ontology format. The construction rules of ontology are based on the set of heuristics for accounting the relational database area knowledge (constraints, data types) and Domain data Knowledge (Transitive Chain of Relations, and Disjointness etc). An ontology automatic generation system based on relational database is designed and implemented according to the construction rules. However, the algorithm need to be extended to cater of other type of domain specific knowledge in order to perfect the automation generation of the ontology.

ACKNOWLEDGEMENT

This work is supported by the funding of TM R&D from the Telekom Malaysia, Malaysia.

REFERENCES

Astrova, I. (2009). Rules for mapping SQL relational databases to OWL ontologies. Metadata and Semantics, 415-424.

Berners-Lee, T. (1998). Relational databases and the semantic web (in design issues). World Wide Web Consortium.

Cerbah, F. (2008). Learning highly structured semantic repositories from relational databases. The semantic web: Research and applications, 777-781.

Das, S., Sundara, S., and Cyganiak, R. (2012). R2RML: RDB to RDF mapping language. [Online]. Available: http://www.w3.org/TR/r2rml/

de Laborda, C. P., and Conrad, S. (2005). Relational. OWL: a data and schema representation format based on OWL. Asia-Pacific Conference on Conceptual modelling, Volume 43, 89-96.

del Mar Roldan-Garcia, M., and Aldana-Montes, J. F. (2008). DBOWL: Towards a Scalable and Persistent OWL reasoner. IEEE International Conference on Internet and Web Applications and Services, 174-179.

Dimou, A., Vander Sande, M., Slepicka, J., Szekely, P., Mannens, E., Knoblock, C., and Van de Walle, R. (2014). Mapping hierarchical sources into RDF using the RML mapping language. IEEE International Conference on Semantic Computing, 151-158.

Drumond, L. and Girardi, R., 2008. A Survey of Ontology Learning Procedures. WONTO, 427, 1-13.

Hert, M., Reif, G., & Gall, H. C. (2011, September). A comparison of RDB-to-RDF mapping languages. International Conference on Semantic Systems, 25-32.

IBM. (2017). Retrieved October 05, 2017, from https://www.ibm.com/analytics/us/en/big-data/

IMDb. (n.d.). Retrieved December 06, 2017, from http://www.imdb.com/

Maedche, A., and Staab, S. (2005). Ontology learning for the semantic web. IEEE Intelligent Systems and Their Applications, 16(2).

Mallede, W. Y., Marir, F., Vassilev, V. T., and Jing, Y. (2013). Algorithms for mapping rdb schema to rdf for facilitating access to deep web. International Conference on Building and Exploring Web Based Environments, 32-41.

Martinez-Cruz, C., Blanco, I.J., and Vila, M.A. (2012). Ontologies versus relational databases: are they so different? A comparison. Artificial Intelligence Review, 1-20.

Michel, F., Montagnat, J. and Faron-Zucker, C. (2014). A survey of RDB to RDF translation approaches and tools (Doctoral dissertation, I3S).

MovieLens. (2016). Retrieved October 27, 2017, from https://grouplens.org/datasets/movielens/

Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A., and Zavitsanos, E. (2011). Ontology population and enrichment: State of the art. In Knowledge-driven multimedia information extraction and ontology evolution, 134-166.

Sánchez, D., and Moreno, A. (2008). Learning non-taxonomic relationships from web documents for domain ontology construction. Data & Knowledge Engineering, 64(3), 600-623.

Schulte, O., and Khosravi, H. (2012). Learning graphical models for relational data via lattice search. Machine Learning, 88 (3), 331–368.

Telnarova, Z. (2010). Relational database as a source of ontology creation. IEEE International Multiconference on Computer Science and Information Technology, 135-139.

Touma, R., Romero, O., and Jovanovic, P. (2015). Supporting data integration tasks with semi-automatic ontology construction. International Workshop on Data Warehousing and OLAP, pp. 89-98.

Zhang, L., and Li, J. (2011). Automatic generation of ontology based on database. Journal of Computational Information Systems, 7(4), 1148-1154.