



R2LD: Schema-based Graph Mapping of relational databases to Linked Open Data for multimedia resources data

Zhanfang Zhao¹ · SungKook Han² · JuRi Kim³ 

Received: 29 June 2017 / Revised: 1 November 2018 / Accepted: 27 January 2019 /
Published online: 7 February 2019
© The Author(s) 2019

Abstract

The Web of Data used to share and interchange the diverse data of heterogeneous types on the Web has been actively established. Ontology-based Linked Open Data (LOD) that allows computers to understand and process data semantics has emerged to extend the current Web of Documents. LOD is important for the modeling of multimedia resources since it provides an efficient way for unstructured data resources. This paper proposes a noble and practical schema-based mapping method to populate Linked Open Data sets from relational databases for multimedia resources. The proposed schema-based mapping R2LD realizes the seamless mapping for RDB-to-RDF by taking advantage of the compatible conceptual schema. The schema-based mapping can resolve the complicated mapping issues in RDB-to-RDF, such as primary and foreign key relationships. The mapping description is straightforward and flexible. It can define mapping information in the form of attribute-value pairs. Especially, the proposed mapping method is suitable for the unstructured multimedia resources. The schema-based mapping R2LD provides an efficient way to implement SPARQL endpoint into RDB and preserve the performance of SQL, which is vital to the dissemination of LOD.

Keywords RDF · Linked Open Data · direct mapping · SPARQL · SQL · Multimedia resources

1 Introduction

The research and development establishing the Web of Data has been accomplished by the opening and sharing of various heterogeneous data types such as the unstructured multimedia resources on the Web. The Linked Open Data (LOD) has emerged as a powerful enabler to

✉ JuRi Kim
cyanic@wku.ac.kr

¹ College of Information Engineering, Hebei GEO University, Hebei 050031, China

² College of Convergence & Liberal Arts, WonKwang University, City Iksan, JeonBuk 54538, Korea

³ Department of Computer Engineering, College of Liberal Arts, WonKwang University, City Iksan, JeonBuk 54538, Korea

extend the current Web of Documents to a Web of interlinked data and, ultimately, into the Semantic Web. Using domain ontologies, LOD can exploit machine-readable data from the diverse data resources in the Web by means of web-based standards for encoding datasets and linking them to other published datasets. In the last decade, numerous best practices of LOD, including DBpedia, YAGO and CAMO, have been published by an increasing number of researchers, governments, public organizations and data providers, creating a global data space that interlinks billions of assertions: the Web of Linked Open Data [4, 7, 9]. Therefore, LOD has evolved from a practical research idea into a very promising technology that can realize the Web as a platform for an intelligent information system with semantic search, query and reasoning capability. In order to accelerate LOD paradigm in the Web, the publication of machine-readable LOD sets should take precedence [4, 21].

Since the vast amount of useful data are still stored in relational databases (RDB), one of the most efficient ways to populate LOD sets is to map data in relational databases into RDF, which is the standard data model of LOD. Due to the importance of mapping RDB to RDF (RDB2RDF), the multifold mapping approaches have been proposed [11, 16, 18]. The most important step towards RDB2RDF is two standard recommendations by the W3C RDB2RDF Working Group: Direct Mapping and R2RML mapping language [1, 12, 14].

With these remarkable achievements for RDB2RDF, it has been expected that the seamless integration of RDB data with RDF datasets towards the Web of Data can be easily actualized. In practice, encouraged by the standard mapping language R2RML, many RDB2RDF systems adopting this common language have been developed in various areas [11]. However, in spite of earnest efforts to adopt this common mapping language, the publication of RDB data on the Web in machine-readable RDF format did not yield the significant results as expected. Although direct mapping and R2RML seem to be inevitable for RDB2RDF, this approach has some limitations [11, 13]. The complex mapping structures of R2RML written in Turtle hinders its practical applications. Due to its robust structures, we cannot find fully-fledged R2RML processors. R2RML also does not address some common questions that occurred in RDB2RDF, such as the implementation of translation process and the way to access the mapped RDF datasets with SPARQL [19]. The translation of SPARQL queries into equivalent SQL queries is one of the crux issues in RDB2RDF [5, 8].

This paper proposes a noble and practical method based on schema translation for RDB2RDF. Since the conceptual schema of RDB is similar to ontological modeling of a certain domain, the mapping of RDB schema into RDF Schema is more effective than the conventional instance-based approaches. We describe how to resolve the intrinsic differences between RDB and RDF in schema level. In addition, we also show the ways to map the operational differences, such as graph pattern matching in RDF and JOIN operations in RDB. To extend this approach, we present a new schema-level mapping method and an effective way to implement SPARQL endpoint in RDB.

The rest of the paper is organized as follows. Section 2 reviews related work on RDB-to-RDF mapping approaches, especially direct mapping method, and analyses the principles for the mapping approaches. Section 3 presents the detailed description of the proposed schema-based mapping: the underlying concepts, mapping method and mapping description. Section 4 deals with schema-based mapping system architecture and implementation of SPARQL endpoint in RDB with some typical mapping examples. Section 5 concludes the paper and discusses possible future work.

2 Related work

Since the vast amounts of information are stored in RDB, RDB2RDF for the publication of RDB data on the Web and the integration of data from different RDBs has been a crucial research topic for LOD applications. A myriad of approaches, techniques, and corresponding tools for RDB2RDF have been proposed over the last decade. With these research efforts, several studies have been conducted to compare approaches and techniques from diverse perspectives. The motivations, underlying principles, specifications, capabilities, and categorizations of RDB2RDF can be referred to these comprehensive surveys of proposed approaches [3, 8, 11, 16–18].

The W3C RDB2RDF Working Group has proposed two standards for RDB2RDF: Direct Mapping and R2RML (Relational Database to RDF Mapping Language) [1, 12, 14]. Direct Mapping is the recommended approach to directly translate RDB data and its schema to RDF representation. R2RML is a generic language for describing a set of customized mapping rules that transform RDB data into RDF datasets. The publications by the W3C have addressed a new, normalized discipline towards standardized RDB2RDF mapping and the development of compliant tools. After emerging R2RML, direct mapping becomes the dominant approach of RDB2RDF.

Broadly speaking, the direct mapping approach uses simple rules to convert RDB data into RDF format in a straightforward manner as follows [10, 11]:

- table-to-class: a table is translated into an ontological class identified by a URI.
- row-to-resource: each row or tuple of a table is translated into a resource that has triple structure in RDF model.
- column-to-predicate: each column of a table is translated into predicate in RDF triple, representing an ontological property.
- primary key-to-subject: the primary key used as an identifier of the table is translated into subject with URI in RDF triple.
- cell-to-literal value: each cell with a literal value is translated into object in RDF triple, representing a data property.
- cell-to-resource with URI: each cell with a foreign key constraint is translated into a resource with URI in RDF triple, representing an object property.

Figure 1 shows the typical example of direct mapping. Direct mapping is the efficient way to create RDF from relational database data, by virtue of being completely automated [2]. In reality, however, the conventional direct mapping approach based on the above rules has some practical limitations. Direct mapping only maps the instances in RDB to RDF, since it does not consider the conceptual schema of RDB. So, this approach is able to capture neither data semantics nor information preservation that is required to realize interoperable LOD. Since the separation between conceptual schema and instances is not considered, mapping language is too complicated to describe the required mapping effectively. One of the most difficult problems in direct mapping is the translation of SPARQL queries into equivalent SQL queries. To resolve these limitations, we have to extend the functionalities of the direct mapping concept or consider other mapping approaches such as schema-level direct mapping proposed in this paper.

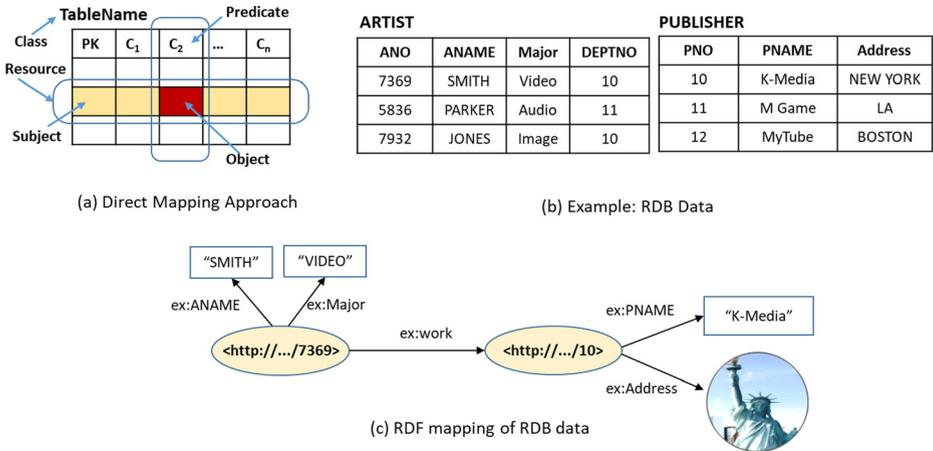


Fig. 1 Concept of direct mapping and RDB2RDF example

3 R2LD: Schema-based Graph Mapping of multimedia RDB to LOD

This section describes the concepts and the techniques of schema-based mapping of RDB2RDF. After reviewing the mapping requirements, we elaborated on the details of schema-based mapping, called R2LD, and explained how to describe mapping relations.

3.1 Requirements for RDB2RDF

The instances of LOD are based on RDF, the standard framework for expressing and interchanging information about resources on the Web. Resources can be anything, including documents, people, physical objects, multimedia and abstract concepts. In RDF, resources are represented in the form of subject–predicate–object expressions, known as triples. The subject denotes the resource, and the predicate denotes traits or aspects of the resource, and expresses a relationship between the subject and the object.

The triples in RDF to be the instances in LOD can be visualized as a connected directed graph. Graphs consists of nodes and arcs. The subjects and objects of the triples make up the nodes in the graph; the predicates form the arcs. Figure 2 shows the graph resulting from the sample triples [15].

While the instances of LOD are consisted of graph model of RDF, RDB data are contained in the structured tables. So the requirement analysis, considering the motivations and objectives of RDB2RDF, is necessary to develop an effective mapping method for two heterogeneous systems that have different constituent elements and query processing mechanisms. Some researches have analyzed formal requirements, such as information and semantics preservation [5, 6]. After analyzing the diverse use cases taking relational data and exposing it in patterns conforming to shared RDF schemata, W3C proposed 11 core requirements and 5 optional requirements. To sum up the core requirements of mapping approaches, the followings should be primary concerns for mapping method [6, 13, 20]:

- The local vocabularies used in RDB should be able to translate into common equivalent ontology vocabularies, since the main objective of RDB2RDF is to expose and publish

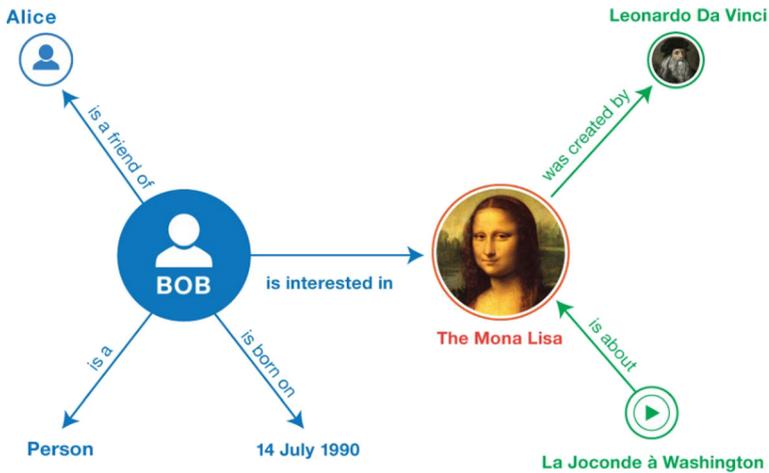


Fig. 2 Informal graph model of LOD

RDB data on the Web. This means the mapping method should provide the efficient ways to redefine column names of RDB tables into ontology vocabularies used in LOD according to their semantic attribute.

- The mapped RDB resources should be identified by URIs of LOD. Dereferencing a HTTP URI about RDF data returns appropriate information. This is mandatory for the mapped RDB data to be LOD.
- The mapping method should provide not only translation of RDB table to RDF graph pattern, but also transformation of SPARQL queries to SQL queries. The mapping description plays the role of an arbitrator between RDB and RDF, unless the duplicated LOD sets of RDB data are generated.
- The mapping method should have competent facility to handle the primary and foreign key constraints and M:N relationships. Since these are the principal mechanism to join tables in RDB, these play an essential role in constructing RDF graph model.
- Some tables decomposed by normalization are used only to connect tables. The mapping method should provide a way to process these normalized tables since they do not represent the conceptual schema. In addition, even tables composed of conceptual attributes should be able to be decomposed on any occasion while they are mapped into RDF graph to maintain the consistent data model.

As most of the requirement analyses have mentioned, the above requirements are mandatory for RDB2RDF, however, it is easy to neglect their indispensable functionalities. Especially, many RDB2RDF mapping approaches have revealed the difficulties in handling the normalized tables, foreign key constraints and SQL query generation. This paper proposes a more efficient, noble approach to deal with these issues.

3.2 Schema-based Graph Mapping approach

The conceptual schema of relational database is usually represented with entity-relationship diagram (ERD), which is almost identical to RDF data model. Accordingly, it can be expected that the RDB2RDF mapping approach should be based on RDB schema of ERD. This

provides a consistent way of mapping and makes it possible to preserve information and conceptual structures of RDB in the process of the mapping.

An entity in ERD is a physical or logical object that can be uniquely identified in a domain, which is conceptual element corresponding to the class of RDF Schema. Since RDF data with triple structure, that become LOD instances, are generated by means of the classes and properties of RDF Schema, RDB schema defined by ERD should be the starting point of RDB2RDF. This schema-based mapping provides the notable feature of the separation of concerns between schema and instance. Focusing on conceptual schema, the mapping method can realize seamless translation of RDB data to LOD instances at schema level. The mapping description also becomes more succinct, since it need not specify the mapping definition for each instance as direct mapping does. Above all, schema-based mapping is expected to realize all the requirements for RDB2RDF.

3.2.1 Mapping RDB table schema to RDF graph

The schema-based mapping is applied to the primitive tables that contain conceptual attributes, not the relational tables that simply represent the association of tables. The primitive tables are usually defined in ERD as an entity, while the relational tables usually generated by means of the decomposition during the normalization are irrelevant to RDF data model. The mapping rules of schema-based mapping are straightforward as follows:

- database-to-namespace: the database name is mapped into the namespace of RDF. Since the database is in some senses kind of the domain, it defines the domain vocabularies to tables and columns.
- table-to-subject: the table name is mapped into the subject of RDF data model. While the subject of the triple in RDF model usually denotes the specified resources with URI, the subject by schema-based mapping plays a role of the organizer to compose the instances.
- column-to-predicate: the column as a property is mapped into the predicate of RDF data model.
- table.column-to-object: the cell value described in table.column is mapped into the object of RDF data model.
- row-to-instance: each row of the table corresponds to RDF triple.

In schema-based mapping, the primitive table schema is preserved and represented in RDF graph as shown in Fig. 3. The mapped RDF graph shows the conceptual schema inherent in the table, not RDF data graph of the instance triple. Note that the mapped object value is represented by the object value term, ‘table.column’. The instance data will be generated according to RDF graph when SPARQL query is executed against the RDB.

A table in itself as a resource is mapped into a subject with namespace and can be additionally associated with the common ontology vocabularies, such as `rdf:type` and `rdfs:subClassOf`. The table column C_i corresponding to the predicate of RDF data can be translated into the well-known vocabularies, such as DC, FOAF, CAMO, and vocabularies in schema.org. In this manner, the schema-based mapping can realize semantic interoperability of the mapped RDF data by using a simple, effective mapping description. The object value of the predicate is described in the value term, `TableName.Ci`. The value term notation makes it possible to translate SPARQL query into the equivalent SQL query.

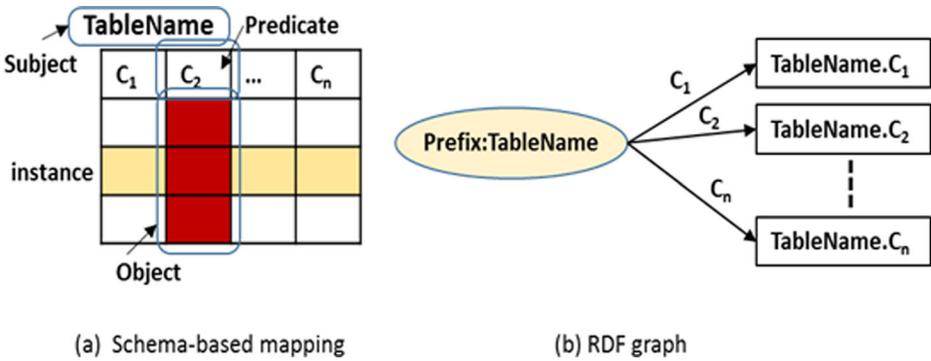


Fig. 3 Schema-based mapping Method

The subject of RDF triple usually denotes the specified resource. However, in schema-based mapping the subject acts as a virtual subject to compose the triple. In practical applications of LOD, since the instance can be identified by its properties, there is no need to use the superfluous identifier as the subject. Rather, the subject in schema-based mapping can represent the class type of the instance.

The schema-based mapping is more succinct and effective than direct mapping based on the instance. The schema-based mapping can resolve theoretical requirements of RDB2RDF as well as preserve the conceptual schema completely. The mapping approach need not be based on the instance modeling since the instance are stored in RDB and can be accessed anytime by SQL query.

3.2.2 Types of the predicate

In relational database, the primary key and the foreign key are used to establish the relationships between the tables. Sometimes these key attributes are independent of the conceptual schema, required only to connect tables. Besides, the relationships between tables are not explicitly specified, only represented in the conceptual schema. Since the relationships are denoted by the predicate in RDF data model, the implied relationships by the key attributes should be explicitly redefined. There are two types of the predicate by its function.

Attribute predicate This predicate represents the primitive concept and has the literal value described in the value term, $TableName.C_i$. In Fig. 4, for example, the column $ARTIST.Name$ in the table $ARTIST$ becomes the attribute predicate.

Link predicate The link predicate originates from the relationships by the primary key and the foreign key. There are two types of link predicates depending on the target table: internal link (i-link) predicate for the recursive relationship and external link (e-link) predicate for the different tables. The object value of the link predicate has the relation expressions that represent the link equation, $TableName.C_i = TableName.C_j$. In Fig. 4, for example, the column $ARTIST.CoWorker$ in the table $ARTIST$ becomes the i-link predicate and has the value, $ARTIST.CoWorker = ARTIST.ID$, as the object, while the column $ARTIST.DID$ becomes e-link predicate and has the value, $ARTIST.DID = PUBLISHER.ID$, as the object. The link predicate is represented with the ontology vocabulary in the RDF graph model.

ARTIST						PRODUCT			
ID	Name	DID	CoWorker	Major	Age	PID	Name	Type	Hosting
A03	john	102	E15	Video	30	P123	XGAME	Game	101
A05	tom	101	E15	Image	24	P463	TMOVIE	Movie	102
A08	judy	102	E05	Sound	37	P763	CM	Movie	103
A15	bill	104	E10	Photo	29	P845	QMUSIC	Sound	103

AP				PUBLISHER			CONTACT		
AID	PID	Role	Date	ID	Name	Location	CID	Phone	Email
A03	P123	director	2017	101	Net-X	London	A03	12345	john@
A03	P763	staff	2016	102	M2M	Boston	A05	34567	tom@
A05	P123	director	2017	103	Mable	LA	A08	34562	judy@
A15	P763	admin	2014	104	Koogle	Paris	A15	34568	bill@

Fig. 4 RDB Tables containing RDB2RDF requirements

The object value term or relation expression of the predicate can solve the difficult problems in RDB2RDF, such as M:N or foreign key relationships. The SQL query to access the instance data is compiled from SPARQL variables with the object value term or relation expression. Figure 4 is the typical example of RDB tables containing the unique features of RDB that should be considered in RDB2RDF. Although AP table is actually a normalized table, its own attributes, Role and Date, are added to show that schema-based mapping can handle the complexity of the relationships in RDB. The column CoWorker in ARTIST table is a recursive relation.

The schema-based mapping of these tables to RDF graph model is Fig. 5. The comprehensive view of the whole conceptual mapping at the schema level is clearly represented without introducing any complex structures shown in direct mapping. The link predicates with the relation expression (dotted line) resolve many complicated mapping issues efficiently, such as the foreign key relationships shown in the AP table. The common ontology vocabulary can be freely adopted as shown in foaf:name.

Note that a new relation between tables such as accomplish can be added without any difficulties during RDB2RDF mapping description. This can enable more complete conceptualization rather than simple mapping of RDB to RDF and generation of a new model appropriate for LOD.

3.3 Mapping description of R2LD

The mapping description of schema-based approach is straightforward, since all mapping information is revealed in the mapped RDF graph as Fig. 5. The mapping description is similar to the class and relation definition in ontology development since the mapped RDF graph is the conceptual schema of RDB. This paper presents a conceptual mapping description instead of the formal definition to clear understanding for schema-based mapping. The following Figure 6 is the part of the mapping description of Fig. 5.

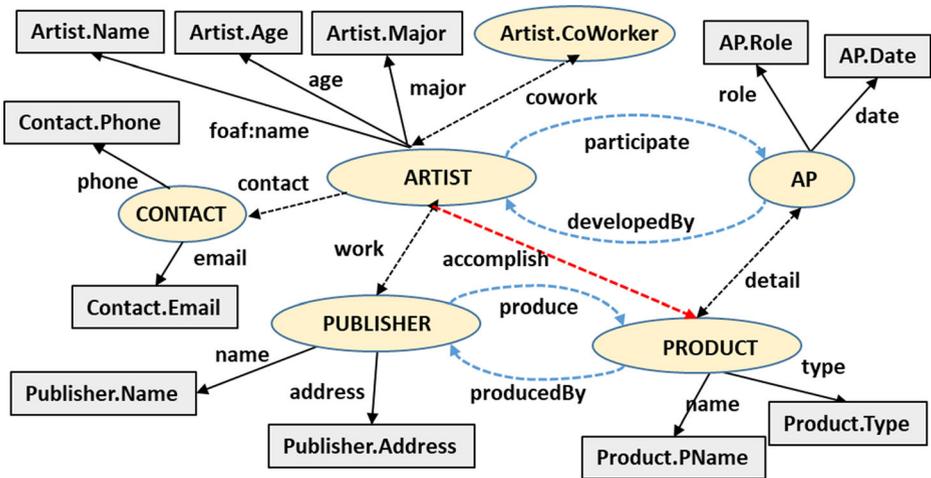


Fig. 5 Schema-based mapping of Fig. 4

The mapping of the attribute predicate shown in Fig. 6a is straightforward. Several RDB vocabularies such as ARTIST. Name, PUBLISHER.Name and PRODUCT.Name can be mapped into a same vocabulary. However, the correct mapping can be clarified by other predicates during the processing of RDF graph. In addition, the common ontologies such as FOAF and GeoNames can be easily imported without any restrictions. This makes it possible to redesign and generate more appropriate RDF graph model of RDB schema.

The prominent feature of schema-based mapping is to use the join expression for link predicate as shown in Fig. 6b, c. Since the relations between resources are implicitly implemented with key types in RDB tables, it is reasonable to explicitly map the link predicate to the join expressions. The join expression can resolve the diverse difficult problems caused in RDB2RDF mapping. Any complex table relationships normalized or partitioned for database management can be accommodated with join expression in a simple and efficient manner.

The mapping description consists of simple one-to-one correspondence of the vocabularies between RDB and RDF. Many-to-one mapping shown in the attribute predicate can be also regarded as one-to-one since the ambiguity can be easily resolved in the process of RDF graph.

RDF.property	RDB.column	Datatype
name	Artist.name Publisher.name Product.name	literal
foaf:name	Artist.name	literal
age	Artist.age	int
role	AP.role	literal
type	Product.type	date
media	Product.type	date
address	Publisher.address	literal
email	Contact.email	literal
phone	Contact.phone	literal

(a) Column Mapping

RDF.relation	RDB.relation	RDB.table	Join.exp
collaborate	cowork	Artist	Artist.CoWorker + Artist.ID
work	work	Artist, Publisher	Artist.DID = Publisher.ID
accomplish	accomplish	Artist, AP	Artist.ID = AP.AID AP.AID = Product.PID
contact	contact	Artist	Artist.ID = Contact.CID

(b) Relation Mapping

Primary	Secondary	Join.exp
Artist	Contact	Artist.ID ^ Contact.CID

(c) Foreign Key Mapping

Fig. 6 Table representation of mapping description of Fig. 5 (part)

The mapping description is also flexible enough that some useful ontology vocabularies can be easily added to realize more competent model of LOD. The simplicity of mapping description also provides easy implementation and applications.

4 R2LD and SPARQL query processing of multimedia resources

To verify the effectiveness of schema-based mapping, this section describes an implementation of SPARQL endpoint into RDB. The schema-base mapping can duplicate RDB to LOD as direct mapping doses. Moreover, the translation of SPARQL to SQL is also straightforward. This section explains SPARQL-to-SQL translation under schema-based mapping.

4.1 Schema-based SPARQL endpoint architecture

The typical architecture of schema-based RDB2RDF system is shown in Fig. 7. The conceptual schema or table relationships of RDB are mapped in to RDF data graph that holds conceptual structures and table relationships. The mapping description is obtained from RDB schema or ERD as described in Section 3.

The SPARQL endpoint to publish RDB data into LOD is implemented with the simple interface to RDB. In the conventional mapping approach, such as direct mapping, robustness and performance have been serious problems. However, schema-based mapping realizes robustness by conceptual schema mapping and performance by SQL query over RDB data. Schema-based mapping can implement SPARQL endpoint with a simple add-on interface without duplication of RDB.

4.2 SPARQL query mapping

In schema-based mapping R2RS, the primary resources for writing SPARQL queries is the mapped RDF graph represented by the conceptual scheme of RDB. The mapped RDF graph promotes conceptual thinking so that this provides a more efficient way to write SPARQL queries. Since the nearly identical conceptual schema is used in both SPARQL and SQL, SPARQL graph patterns are efficiently translated into SQL queries. Through the typical use

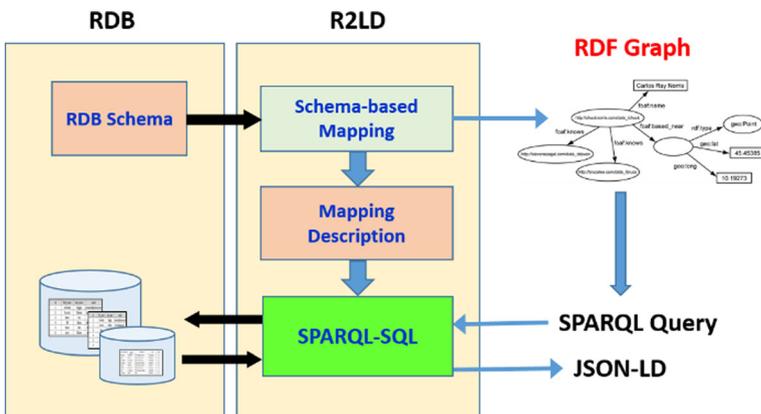


Fig. 7 System architecture with schema-based mapping

cases of SPARQL queries, the effectiveness of query processing in schema-based mapping is shown below.

4.2.1 Relationships between two tables with foreign keys

Although many suggestions have been proposed, the primary and foreign key issues still remain cumbersome obstacles in RDB2RDF mapping. Some proposed mapping description is also too complicated to apply for the real operational databases. However, in schema-based mapping R2LD, the foreign key relationships are explicitly represented in the mapped RDF graph, and the detailed mapping methods are specified as the join expressions in the mapping description. This provides an efficient way to solve the implicit foreign key relationships in RDB.

For example, in Fig. 5, the query “What publisher does John work in?” in RDF graph of Fig. 8a is written in SPARQL as Fig. 8b. Writing SPARQL query over the mapped RDF graph is straightforward, as seen in Fig. 8b. From the mapping description in Fig. 6, “John” of foaf:name has the object value ARTIST.Name, and the link predicate work has join expression ARTIST.DID = PUBLISHER.ID, respectively. With all this information, the predicate name between ?y and ?z can be deduced into PUBLISHER.Name. As a result, SPARQL query can be translated into SQL query of Fig. 8c.

Some tables in RDB such as CONTACT in Fig. 4 is only for practical purposes of database management, not related to conceptual schema. In this case, combining CONTACT with its master table ARTIST is natural. In schema-base mapping, any table combining necessary for more desirable RDF model can be easily accomplished with the join expression. For example, RDF query graph combining CONTACT with ARTIST shown in Fig. 9 is definitely possible.

For example, in Fig. 9, the subject ?x is related to the tables ARTIST and CONTACT by means of its predicates and the predicate name is resolved to ARTIST.Name. So from the mapping description in Fig. 6c, since the join expression to combine ARTIST and CONTACT is obtained, SPARQL query can be easily translated in to SQL query.

The join expression in schema-based mapping is an efficient way to solve the diverse problems caused in RDB2RDF. All complicated mapping problems can be accommodated into the mapping description. This grants more flexibility than rigid mapping for simple duplication of RDB.

4.2.2 Recursive relationships

The recursive relationships are easily implemented in RDB and RDF graph model. However, the mapping of recursive relationships is another matter in RDB2RDF. Although some mapping approaches suggest the plausible solutions, these methods accompany ineffective

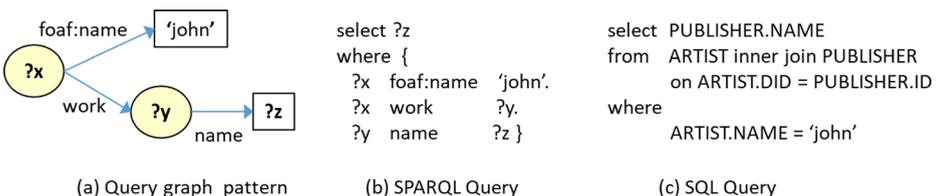


Fig. 8 Example: foreign key relationship

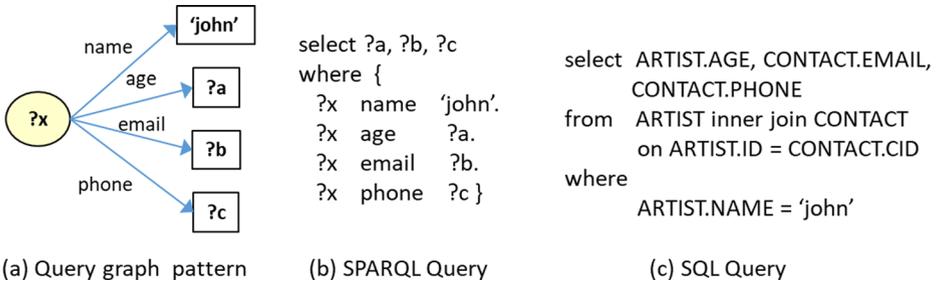


Fig. 9 Example: Combining tables with foreign key relation

and unpractical issues. Schema-based mapping can easily deal with this cumbersome problem by means of the join expression.

The query containing the recursive relationship contained in ARTIST table, such as “Who is the coworker of John?” can be efficiently modeled as shown in Fig. 10. Actually, this recursive relationship is similar to the simple foreign key relationship. From the mapping description of Fig. 6b, the join expression of collaborate can be obtained similar to foreign key relations in the previous examples. As a result, the corresponding SQL query can be easily obtained as in Fig. 10c.

4.2.3 Multi-tables relationships

SPARQL queries related with many tables is not the matter of the odd mapping in schema-based mapping. The join expression in the mapping description can resolve the complicate table management as usual. The tables related with SPARQL query in Fig. 11 are identified by means of link predicates, work and contact, and table join operations for SQL query is composed with their join expressions in the mapping description.

Even though SPARQL query use the same predicate such as name in ?x and ?y, this ambiguity can be resolved with the mapping information of other predicates. Schema-based mapping is flexible in the use of ontology vocabularies.

4.2.4 Multi-tables and M:N relationships

In schema-based mapping, any additional relationships can be inserted as a link predicate accomplish shown in Fig. 5. The additional relationships regardless of RDB schema can realize more reasonable RDF model.

The additional relationships contain many join operations and M:N relationships. In spite of the complexity of the additional relationships, these can be handled as usual as other link

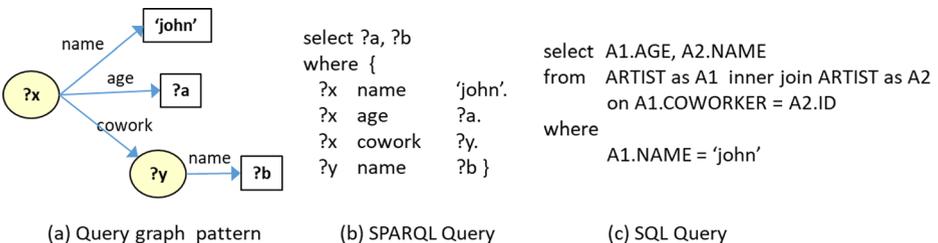
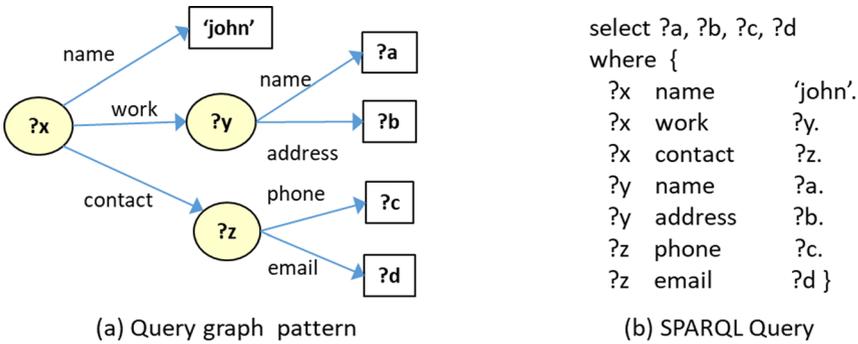


Fig. 10 Example: Recursive relationship



```
select PUBLISHER.NAME, PUBLISHER.ADDRESS, CONTACT.PHONE, CONTACT.EMAIL
from ARTIST
    inner join PUBLISHER on ARTIST.DID = PUBLISHER.ID
    inner join CONTACT ARTIST.ID = CONTACT.CID
Where
    ARTIST.NAME = 'john'
```

(c) SQL Query

Fig. 11 Example: Multi-tables and M:N relationship

predicates in schema-based mapping. Even though the query in Fig. 12 contains the additional link predicate **accomplish**, SQL Query can be generated with its join expression in the mapping description as usual as other SPARQL query translation.

4.2.5 Joined and merged-tables with M:N relationships

Any complicated SPARQL query can be easily translated into the corresponding SQL query in schema-based mapping. The following example shown in Fig. 12 is conceptually intuitive and suitable for LOD. However, this query contains very complex relationship problems among the tables of RDB.

The variable **?a** is related to the table CONTACT, however, the table is implicitly joined and merged with ARTIST. The variable **?b** is related with the Table AP that originally mediates the primary and foreign key relationships by the normalization, but has its own attributes. In addition, the link predicate **accomplish** related with the variable **?y** as seen in the previous example is the condensed relation that merges several relations in RDB.

With the join expressions in the mapping description, the identification of the related tables is systematically tractable. After resolving the variables and predicates, SQL query can be easily composed as in Fig. 13.

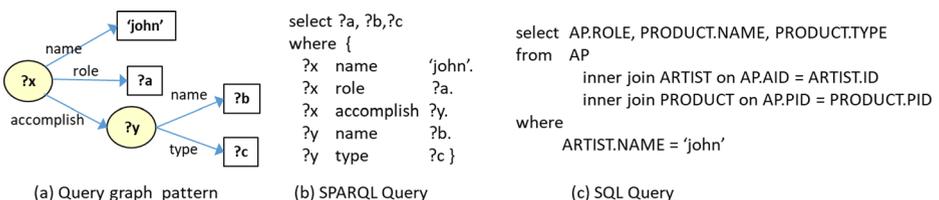


Fig. 12 Example: Multi-tables and M:N relationship

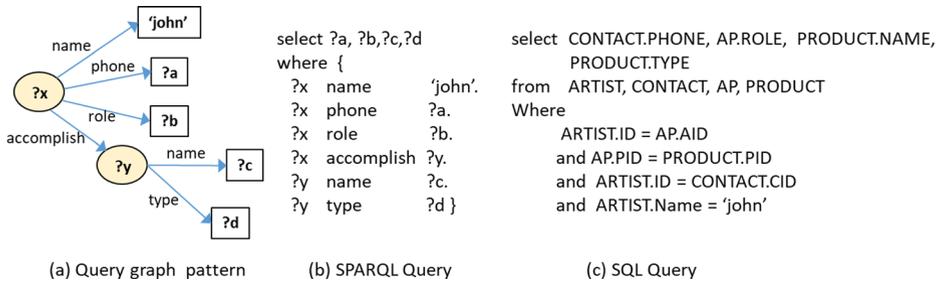


Fig. 13 Example: Joined and merged tables with M:N relationships

The schema-based mapping supports to build more natural SPARQL query adequate for LOD applications and provides the efficient way of the mapping to SQL query. This make it possible to implement SPARQL endpoint in RDB, which is vital to publish RDB to LOD sets.

5 Conclusion

The research and development involved in the realization of the Web of Data has been actively accomplished by opening and sharing various heterogeneous data types such as unstructured multimedia resources on the Web. Ontology-based LOD that allows computers to understand and process the data semantics is proposed as a standard data model. In this model, LOD becomes the enabler to realize the Web of Data by using the standard data model for both structured and unstructured information resources of the Web and a shared semantic representation.

The high qualitative data sets should be provided to realize the diverse intelligent services on the Web. The conventional approaches to develop LOD sets, such as ontology-based LOD generation and the translation of RDB to LOD, suffer from the complexity and difficulty of the approaches, the realization of domain peculiarities and practical adaptabilities. In addition, the development of information systems based on LOD has made very little progress due to the lack of appropriate, specialized methodologies and tools for the instances of LOD sets.

The most effective and practical way to populate the LOD sets is to publish the data stored in RDB on the Web as the standard form of RDF. Many studies about RDB-to-RDF mapping have been conducted to realize the Web of Data. Two standard drafts have been proposed as important achievements by the W3C: Direct Mapping and R2RML mapping language. However, the practical RDB-to-RDF mapping approach is still an open question.

This paper proposes a noble and practical RDB2RDF mapping method suitable for LOD at the conceptual schema level. Since the conceptual schema of RDB is similar to ontological domain modeling of RDF, the proposed schema-based mapping R2LD can achieve more coherent mapping than the conventional direct mapping approaches by dissolving the structural and operational differences, such as graph pattern matching and JOIN operations. The mapping description is straightforward on account of the compatible conceptual structures and can accommodate the complex relationships in an effective manner. In addition, the implementation of schema-based mapping is simple and intuitive as seen in several typical examples. So, the schema-based mapping R2LD provides an efficient way to implement SPARQL endpoint into RDB, which is vital to disseminate LOD.

Acknowledgements This paper was supported by Wonkwang university in 2017.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. A Direct Mapping of Relational Data to RDF. W3C Recommendation 27 September 2012. <https://www.w3.org/TR/rdb-direct-mapping>. Accessed 12 Oct 2018
2. A Direct mapping of Relational Data to RDF. W3C Working Draft 29 May 2012. <https://www.w3.org/TR/rdb-direct-mapping/>. Accessed 12 Oct 2018
3. A Survey of Current Approaches for Mapping of Relational Databases to RDF. W3C RDB2RDF Incubator Group report 1:113–130. https://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf. Accessed 12 October 2018
4. Bizer C (2009) The emerging web of linked data. *IEEE Intell Syst* 5:87–92. <https://doi.org/10.1109/MIS.2009.102>
5. Chebotko A, Lu S, Fotouhi F (2009) Semantics Preserving SPARQL-to-SQL Translation. *Data Knowl Eng* 68(10):973–1000
6. Erling O (2008) Requirements for Relational to RDF Mapping. <https://www.w3.org/wiki/Rdb2RdfXG/ReqForMappingByOErling>. Accessed 12 Oct 2018
7. Heath T, Bizer C (2011) Linked Data: Evolving the Web into a Global Data Space. *Synthesis Lectures on the Semantic Web: Theory and Technology* 1(1):1–136. <https://doi.org/10.2200/S00334ED1V01Y201102WBE001>
8. Hert M, Reif G, Gall HC (2011) A Comparison of RDB-to-RDF Mapping Languages. *Proceedings of the 7th International Conference on Semantic Systems, ACM, New York*, pp 25–32. <https://doi.org/10.1145/2063518.2063522>
9. Hu W, Jia C, Wan L et al (2014) CAMO: Integration of Linked Open Data for Multimedia Metadata Enrichment. *ISWC Lecture Notes in Computer Science*. https://doi.org/10.1007/978-3-319-11964-9_1
10. Lee TB (1998) Relational Databases on the Semantic Web. www.w3.org/DesignIssues/RDB-RDF.html. Accessed 12 Oct 2018
11. Michel F, Montagnat J, Zucker CF (2014) A Survey of RDB to RDF Translation Approaches and Tools. *Dissertation, I3S*. <hal-00903568v2>
12. R2RML and Direct Mapping Test Cases. W3C Working Group Note 14 August 2012. <https://www.w3.org/TR/rdb2rdf-test-cases>. Accessed 12 Oct 2018
13. R2RML and Direct Mapping Test Cases. W3C Editor's Draft 24 July 2012. <http://www.w3.org/2001/sw/rdb2rdf/test-cases>. Accessed 12 Oct 2018
14. R2RML: RDB to RDF Mapping Language. W3C Recommendation 27 September 2012. <https://www.w3.org/TR/r2rml>. Accessed 12 Oct 2018
15. RDF 1.1 Primer. W3C Working Group Note 25 February 2014. <https://www.w3.org/TR/rdf11-primer/>. Accessed 12 Oct 2018
16. Sahoo SS, Halb W, Hellmann S et al (2009) A Survey of Current Approaches for Mapping of Relational Databases to RDF. W3C RDB2RDF Incubator Group report 1:113–130. https://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf. Accessed 12 Oct 2018
17. Sequeda J, Priyatna F, Villazón-Terrazas B (2012) Relational Database to RDF Mapping Patterns. *Proceedings of the 3rd International Conference on Ontology Patterns (WOP'12)*. pp 97–108. <http://ceur-ws.org/Vol-929/paper9.pdf>. Accessed 02 Feb 2019
18. Spanos DE, Stavrou P, Mitrou N (2012) Bringing Relational Databases into the Semantic Web: A Survey. *Semantic Web* 3(2):169–209. <https://doi.org/10.3233/SW-2011-0055>

19. SPARQL 1.1 Query Language. W3C Recommendation 21 March 2013. <https://www.w3.org/TR/sparql11-query>. Accessed 12 Oct 2018
20. Use Cases and Requirements for Mapping Relational Databases to RDF. W3C Working Draft 8 June 2010. <https://www.w3.org/TR/rdb2rdf-ucr/>. Accessed 12 Oct 2018
21. Yoshimura KS (2016) Analysis of International Linked Data Survey for Implementers. D-Lib Magazine 22: 7–8. <https://doi.org/10.1045/july2016-smith-yoshimura>



Zhanfang Zhao is an assistant professor of Hebei GEO University, Hebei, China. She attended Hebei Institute of Technology in 2000 and University of Mining & Technology, Beijing, China in 2005. She is a senior researcher of Smart Technology Lab in Won Kwang University, Korea. Her main research areas are artificial intelligence, ontology, knowledge management and NoSQL databases.



Sung-Kook Han is a professor of Computer Engineering at Won Kwang University, Korea. He was a visiting of Semantic Technology Institute Innsbruck (STI2), Austria for years. He is a chair and advisory committee member of various academic societies such as Korea Information Society and The Korean Society for Cognitive Science. He is also a member of IEEE, ACM and AAAI. He was a committee chair of several international conferences. Prof. Han was awarded many prizes for his outstanding research contributions and services to research communities



Ju-Ri Kim is an assistant professor of Liberal Arts College of Won Kwang University. She received her PhD degree in Department of Computer Engineering, Won Kwang University in 2009. She is a steering committee member of many academic societies, to name a few, Korea Convergence Society and Korea Computer Education Society. She has participated in a reviewer of many national research projects. She is interested in 2D /3D image processing, semantic annotation of multimedia resources, Linked Data and machine learning.