# A Completely Automatic Direct Mapping of Relational Databases to RDF and OWL

Juan F. Sequeda[1], Marcelo Arenas[2], and Daniel P. Miranker[1]

[1] Department of Computer Science, The University of Texas at Austin
[2] Department of Computer Science, Pontificia Universidad Católica de Chile

**Abstract.** Integrating relational databases with the Semantic Web can be accomplish by means of two primary approaches: automatic direct mapping or developers detailing application specific mappings. Both approaches are the subject of the W3C Relational Database to RDF (RDB2RDF) Working Group. Intuitively, a direct mapping is a default and automatic way to translate a relational database schema and its content to OWL and RDF. In this poster, we present a specification, expressed in Datalog, of a direct mapping inspired by the current Direct Mapping draft of the W3C RDB2RDF Working Group. We are currently studying four fundamental properties: monotonicity, information preservation, query preservation and semantics preservation. In particular, we observe that the combination of these properties needs to be addressed very carefully.

## 1 Introduction

We present a specification that directly maps a relational database to an RDF graph with OWL 2 DL vocabulary. Intuitively, a direct mapping is a default and automatic way to translate a relational database to RDF. One report suggests that Internet accessible databases contained up to 500 times more data compared to the static Web and roughly 70% of websites are backed by relational databases, making automatic translation of relational database to RDF central to the success of the Semantic Web [5].

Several approaches have been presented that directly map relational schemas to OWL and other ontology languages [9]. Currently, the W3C RDB2RDF Working Group is developing two standards to map relational databases to RDF: R2RML, a customized mapping language [3] and the Direct Mapping, an automatic mapping [2]. R2RML targets users who have expertise in databases and RDF. Hence, if a user with no knowledge of RDF would like to generate RDF from their database, he/she would encounter a steep learning curve. On the other hand, the Direct Mapping is simple because it is automatic and no input from a user is needed. An obvious criticism is that the Direct Mapping cannot generate customized mappings. However, a user can generate customized views of the data with SQL Views and then decide which tables and views to directly map.

Due to the need of simple and automatic methods to translate relational databases to RDF, we focus on studying the Direct Mapping. We study four properties that are fundamental to a direct mapping: monotonicity, information preservation, query preservation and semantics preservation. Our current results show that the combination of these properties is non-trivial. We refer the reader to [8] for a complete description of the Datalog rules and proofs.

## 2 Direct Mapping

The input of a direct mapping, $\mathcal{M}$, is a relational schema $\mathbf{R}$, a set $\Sigma$ of primary keys (PKs) and foreign keys (FKs) over $\mathbf{R}$ and an instance $I$ of $\mathbf{R}$. The output is an RDF graph with OWL 2 DL vocabulary. Assume that $\mathcal{G}$ is the set of all possible RDF graphs and $\mathcal{RC}$ is the set of all triples $(\mathbf{R}, \Sigma, I)$ such that $\mathbf{R}$ is a relational schema, $\Sigma$ is a set of PKs and FKs over $\mathbf{R}$ and $I$ is an instance of $\mathbf{R}$.

**Definition 1 (Direct mapping).** *A direct mapping $\mathcal{M}$ is a total function from $\mathcal{RC}$ to $\mathcal{G}$.*

We present a direct mapping, $\mathcal{DM}$, that integrates and extends the functionalities of the direct mappings proposed in [11, 2]. It consists of five parts: (1) predicates that encode the input relational schema to the $\mathcal{DM}$: $\textsc{Rel}(r)$ indicates that $r$ is a relation name in $\mathbf{R}$; $\textsc{Attr}(a, r)$ indicates that $a$ is an attribute in the relation $r$ in $\mathbf{R}$; $\mathrm{PK}_n(a_1, \ldots, a_n, r)$ indicates that $r[a_1, \ldots, a_n]$ is a primary key in $\Sigma$; finally $\mathrm{FK}_n(a_1, \ldots, a_n, r, b_1, \ldots, b_n, s)$ indicates that $r[a_1, \ldots, a_n]$ is a foreign key in $\Sigma$ and references $s[b_1, \ldots, b_n]$, (2) a predicate that stores the tuples of the relational instance: $\textsc{Value}(v, a, t, r)$ indicates that $v$ is the value of an attribute $a$ in a tuple with identifier $t$ in a relation $r$ (that belongs to $\mathbf{R}$) (3) predicates that are used to store an ontology: $\textsc{Class}(c)$ indicates that $c$ is a class; $\mathrm{OP}_n(p_1, \ldots, p_n, d, r)$ indicates that $p_1, \ldots, p_n$ $(n \geq 1)$ form an object property with domain $d$ and range $r$ and $\mathrm{DTP}(p, d)$ indicates that $p$ is a data type property with domain $d$. Additionally, we present Datalog rules that generate a putative ontology from the relational schema. These rules can be summarized as follows: A table is translated to an OWL Class unless the table represents a binary relationship, then it is translated to an OWL Object Property. Foreign Keys are translated to OWL Object Properties while attributes are translated to OWL Datatype Properties. (4) Datalog rules that generates the OWL 2 DL vocabulary from a relational schema which include rules to generate IRIs and express the ontology as RDF triples. (5) Datalog rules that generates RDF triples from a relational instance based on the putative ontology. We refer the reader to [8] for the full set of Datalog rules.

## 3 Fundamental Properties of a Direct Mapping

We introduce four fundamental properties of direct mappings, namely monotonicity, information preservation, query preservation and semantics preservation.

*Monotinicity.* Consider two database instances $I_1$ and $I_2$ such that $I_1$ is contained in $I_2$ (denoted by $I_1 \subseteq I_2$). A direct mapping $\mathcal{M}$ is considered monotone if for any such pair of instances, the result of mapping $I_2$ contains the result of mapping $I_1$. In other words, if we insert new data to the database, then the elements of the mapping that are already computed are unaltered. It is straightforward to see that $\mathcal{DM}$ is monotone, because all the negative atoms in the Datalog rules defining $\mathcal{DM}$ refer to the schema, the PKs and the FKs of the database, and these elements are kept fixed when checking monotonicity.

*Information preservation.* A direct mapping is information preserving if it does not lose any information about the relational instance being translated, that it, if there exists a way to recover the original database instance from the RDF graph resulting from the translation process. It is straightforward to see that $\mathcal{DM}$ is information preserving, because it involves providing an algorithm that can reconstruct the initial relational instance from the generated RDF graph.

*Query preservation.* A direct mapping is query preserving if every query over a relational database can be translated into an equivalent query over the RDF graph resulting from the mapping. That is, query preservation ensures that every relational query can be evaluated using the mapped RDF data. To prove that $\mathcal{DM}$ is query preserving, we build on the results of [1], where it is shown that non-recursive Datalog with safe negation (which is as expressive as relational algebra) has the same expressive power as SPARQL.

*Semantics preservation.* Intuitively, a direct mapping is semantics preserving if the satisfaction of a set of PKs and FKs by a relational database is encoded in the translation process. More precisely, given a relational schema $\mathbf{R}$, a set $\Sigma$ of PKs and FKs over $\mathbf{R}$ and an instance $I$ of $\mathbf{R}$, a semantics preserving mapping should generate from $I$ a consistent RDF graph if $I \models \Sigma$, and it should generate an inconsistent RDF graph otherwise. It is straightforward to find a counter-example, demonstrating that $\mathcal{DM}$ is not semantics preserving. Does this mean that our direct mapping is incorrect? What could we do to create a direct mapping that is semantics preserving? These are the questions that we are addressing in our ongoing work

## 4 Issues and Ongoing Work

We present simple extension of the direct mapping $\mathcal{DM}$ that make it semantics preserving. Additionally, we discuss other ongoing issues in our research.

*Semantics preserving direct mapping for PKs.* Consider a new direct mapping $\mathcal{DM}_{pk}$ that extends $\mathcal{DM}$ as follows. A Datalog rule is used to determine if the value of a primary key attribute is repeated. If such a violation is found, then an artificial triple is generated that would produce an inconsistency.

*On monotone semantics preserving direct mappings.* We prove that no monotone direct mapping $\mathcal{M}$ is semantics preserving. Hence, the desirable condition of being monotone is, unfortunately, an obstacle to obtain a semantics preserving direct mapping. The reason why we have not been able to create a semantics preserving direct mapping is because of two characteristics of OWL: (1) it adopts the Open World Assumption (OWA) while relational database adopts the Closed World Assumption (CWA) and (2) it does not adopt Unique Name Assumption (UNA). In other words, what causes an inconsistency in a relational database, can cause an inference of new knowledge in OWL.

*Non-monotone semantics preserving direct mappings for PKs and FKs.* Consider a new non-monotone direct mapping, $\mathcal{DM}_{pk+fk}$, which extends from $\mathcal{DM}_{pk}$, and checks if there is a violation of the FK integrity constraint beforehand. If such a FK violation exists, then it creates a artificial RDF triple which will generate an inconsistency with respect to OWL 2 DL semantics.

*A monotone semantics preserving direct mapping based on an epistemic operator.* If we want a semantics preserving monotone direct mapping, we would need to consider an alternative semantics of OWL for expressing integrity constraints. Because OWL is based on Description Logic, we would need a version of DL that supports integrity constraints, which is not a new idea. Integrity constraints are epistemic in nature and are about "what the knowledge base knows" [7]. Extending DL with the epistemic operator **K** has been studied [4]. Several researches have worked on different ways of having OWL handle integrity constraints [10, 6] Therefore, it is possible to extend $\mathcal{DM}_{pk}$ to create a monotone direct mapping that is semantics preserving, but it is based on a non-standard version of OWL including the epistemic operator **K**.

*Additional issues.* We consider only relational databases with set semantics. However, notice that in our setting each tuple has its own identifier. Thus, even if repeated tuples exist, each tuple will still have its unique identifier and, therefore, exactly the same rules can be used to map relational data under bag semantics. Besides, we focus on relational databases that do not contain null values as there is well-understood and standard semantics for relational algebra in this case. In order to consider null values, we must choose an alternative semantics for relational algebra or define a semantic of null values in SQL.

## References

1. R. Angles and C. Gutierrez. The expressive power of sparql. In *International Semantic Web Conference*, pages 114–129, 2008.
2. M. Arenas, E. Prud'hommeaux, and J. Sequeda. Direct mapping of relational data to RDF. W3C Working Draft 24 March 2011, `http://www.w3.org/TR/rdb-direct-mapping/`.
3. S. Das, S. Sundara, and R. Cyganiak. R2rml: Rdb to rdf mapping language. W3C Working Draft 24 March 2011, `http://www.w3.org/TR/r2rml/`.
4. F. Donini, M. Lenzerini, D. Nardi, W. Nutt, and A. Schaerf. An epistemic operator for description logics. *Artif. Intell.*, 100(1-2):225–274, 1998.
5. B. He, M. Patel, Z. Zhang, and K. C.-C. Chang. Accessing the deep web. *Commun. ACM*, 50:94–101, May 2007.
6. A. Mehdi, S. Rudolph, and S. Grimm. Epistemic querying of owl knowledge bases. In *ESWC (1)*, pages 397–409, 2011.
7. R. Reiter. On integrity constraints. In *TARK*, pages 97–111, 1988.
8. J. F. Sequeda, M. Arenas, and D. P. Miranker. On directly mapping relational databases to rdf and owl (extended version). Technical Report TR-11-28, University of Texas at Austin, Department of Computer Science, June 2011. `http://www.cs.utexas.edu/˜jsequeda/directmapping.html`.
9. J. F. Sequeda, S. H. Tirmizi, O. Corcho, and D. P. Miranker. Survey of directly mapping sql databases to the semantic web. *Knowledge Eng. Review*, To Appear 2012.
10. J. Tao, E. Sirin, J. Bao, and D. L. McGuinness. Integrity constraints in owl. In *AAAI*, 2010.
11. S. H. Tirmizi, J. Sequeda, and D. P. Miranker. Translating SQL Applications to the Semantic Web. In *DEXA*, pages 450–464, 2008.