

StdTrip+K: Design Rationale in the RDB-to-RDF process

Rita Berardi¹, Karin Breitman¹, Marco A. Casanova¹,
Giseli Rabello Lopes¹, Adriana Pereira de Medeiros²

¹Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, RJ – Brazil CEP 22451-900
{rberardi, karin, casanova, grlopes}@inf.puc-rio.br

²Instituto de Ciência e Tecnologia
Universidade Federal Fluminense
Rio das Ostras, RJ – Brazil CEP 28890-000
adrianamedeiros@puro.uff.br

Abstract. The design rationale behind the triplification of a relational database is a valuable information source, especially for the process of interlinking published triplesets. Indeed, studies show that the arbitrary use of the *owl:sameAs* property, without carrying context information regarding the triplesets to be linked, has jeopardized the reuse of the triplesets. This article therefore proposes the StdTrip+K process that integrates a design rationale approach with a triplification strategy. The process supports the reuse of standard RDF vocabularies recommended by W3C for publishing datasets and automatically collects the entire rationale behind the ontology design, using a specific vocabulary called Kuaba+W.

Keywords: Triplification, mapping, matching, design rationale

1 Introduction

Linked Data refers to a set of best practices for publishing and connecting structured data on the Web [3]. One of the most popular strategies to publish structured data on the Web is to convert relational databases to the Linked Data format, in a process known as *RDB-to-RDF* or *triplification* [11], [13].

One of the major challenges of publishing Linked Data is to investigate the value of information based on the trustworthiness of its sources, the time of validity, the certainty, or the vagueness asserted to specified or derived facts [6]. This challenge is associated with the lack of analytical information about the published Linked Data, i.e. information that answers questions such as: (1) Did the original relational database suffer *changes* when published as Linked Data that could impact its quality?; (2)

Is the *translation* from the original relational database to Linked Data *correct*?; (3) Is the *chosen ontology* the most appropriate to represent the original relational database?; (4) Did the original relational database *lose* some relevant *information* when it was published as Linked Data? These details of the triplification process should answer the questions above mentioned which are reasoned in the decisions related to changes, correctness, choices and information losing during the triplification.

In general, the decisions taken during a design process, the accepted and rejected options, and the criteria used are called *design rationale* (DR) [8], or *triplification rationale* by analogy. Besides helping the reuse of datasets, the triplification rationale has a potential value for supporting design of new ontologies because all the experience acquired during a design can be transmitted and augmented by the reuse of previous DRs in new designs. Although there are several triplification engines, we are unaware of any previous work that applies DR in the Linked Data domain, i.e. that captures the triplification rationale. The details intrinsically involved in the mapping activity should reflect all aspects related to how the concepts of the underlying conceptual schema are mapped to the RDF terms. Furthermore, these detailed information can explicit some problems in the mapping process. For instance, if an *entity element* of an ER is mapped to a *property element* in RDF, the *attribute elements* of this entity may not be represented due to the lack of the domain representation, since the domain is represented as a property.

The matching step involves domain expert decisions regarding the construction of the vocabulary. The details inherent in the matching step should reflect aspects related to the choice of each term of the vocabulary that will be used to publish the database. The decisions of the designer involved in this activity have to consider the database domain and context. For instance, considering a domain of an university publication database where the entity “Authors” has the attribute “name”, the most adequate representation is *dc:creator* instead of *foaf:Person*, since *dc:creator* is more representative for the domain. Otherwise, if an entity “Students” has the same attribute “name”, *dc:creator* is not the best choice although both entities are in the same domain of “University”. The DR representation in the StdTrip+K process is executed through the Kuaba approach [12] that represents a more complete DR in respect to other DR approaches. So the major contribution of this paper is to address the incorporation of DR capture through the addition of Kuaba+W vocabulary in the StdTrip process [14], generating the StdTrip+K, that is, to the best of our knowledge, the first to address the capturing of the decisions behind the triplification task. The remainder of this article is organized as follows. Section 2 discusses related work. Section 3 details the StdTrip+K process along with a running example and describes the Kuaba+W vocabulary used to record the DR. Finally, Section 4 presents the conclusions and directions for future work.

2 Related Work

There are several approaches RDB-to-RDF with different mechanisms to tackle this translation process. The more relevant approaches for the RDB-to-RDF process are

Triplify [1], D2RQ [2], Virtuoso RDF view [7] and RDBtoOnto [4]. Triplify motivates the need for a simple mapping solution using SQL (Structure Query Language) as a mapping language and transforms database query results into RDF triples and Linked Data. The mapping is done manually with no record of any rationale. D2RQ generates the mapping files automatically, using the table-to-class and column-to-predicate approach. It uses a declarative language, implemented as Jena graph, to define the mapping file, also with nothing about recording rationale. In the Virtuoso RDF view the mapping file, also called RDF view, is automatically generated with table-to-class approach. In this approach there is no reason to capture the rationale since it does not imply in options, arguments and decisions. RDBtoOnto brings a discussion on how to take advantage of database data in obtaining more accurate ontologies. This work also uses the table-to-class and column-to-predicate to create an initial ontology schema, which is then refined through identification of taxonomies guided by the tool. Although there is user interference, the decisions made are not recorded. There are other approaches like DB2OWL [5] and Ultrawrap [15], but still they do not cover the rationale issue. In the context of rationale models and tools, there are argumentation-based models such as IBIS [17], DRL [9], QOC [10] that allow the DR representation. However, they do not present a complete DR that includes accepted and rejected options and the reasons for that. Specifically in the Linked Data context, we have not found researches with this purpose. There are provenance models, like Open Provenance Model (OPM¹), that records the history of creating a dataset in general terms. Despite been very important and essential for Linked Data quality, it lacks in terms of decisions during the creation of a mapping file. We can conclude that the approach followed by most triplifying approaches has no concern with design rationale recording.

3 The StdTrip+K Process

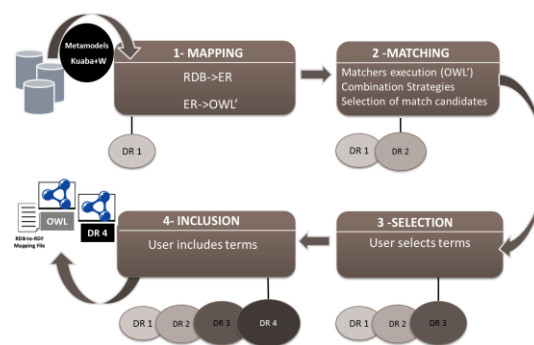


Fig. 1. StdTrip+K Process overview

¹ openprovenance.org/

The StdTrip+K process (Fig.1) is anchored in the principle of ensuring interoperability through the use of standards in schema design and through the DR recording. The process receives as input the RDB, the metamodels and the DR vocabulary Kuaba+W. At each stage, the respective DR is traced and recorded using Kuaba+W vocabulary that is incrementally recorded throughout the process execution. In the end, the process results in the RDB-to-RDF Mapping File, the OWL ontology and the final DR. The Kuaba+W vocabulary is described in Section 3.1 and the four steps (Mapping, Matching, Selection and Inclusion) of the StdTrip+K process are described in Section 3.2 using a motivation example.

3.1 Kuaba+W – A Design Rationale Vocabulary for RDB-to-RDF process

Kuaba+W extends the Kuaba approach [12] in the sense that it eliminates elements not necessary in the RDB-to-RDF domain. Moreover, the Kuaba+W extension is related to the addition of the *Description* element, which is related to a *Justification* and carries information regarding the reasons for the domain expert to accept or reject an idea. A description is also related to a *Metamodel*, also new in the extension, since there is more than one metamodel involved in the RDB-to-RDF. A metamodel registers which formal artifact was involved in each step of design process, for instance ER and RDF metamodels. Fig. 2 shows the main elements of the ontology, using a UML-like graphical notation to help visualization. A **Reasoning element** represents the design issue that the ontology designer should deal with (question, ideas and arguments).

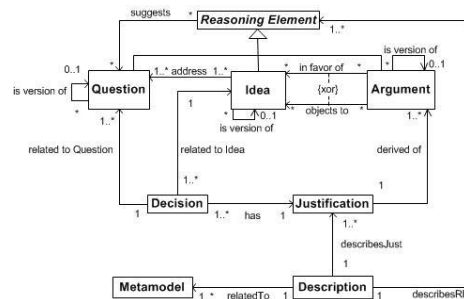


Fig. 2. The Kuaba+W ontology elements

An **Idea** represents a potential solution for the mapping or matching issue presented by the Reasoning Element Question. The **Argument** represents the criteria used to present an Idea for a Question. A **Decision** represents the acceptance or the rejection of an idea as a solution to a question. A **Justification** indicates the justification for each Decision that explains why an Idea was accepted or rejected as a solution for a particular Question. **Description** contains details about any Reasoning Element and justification, depending on the step of the process. **Metamodel** indicates which metamodel is accessed in the mapping process to automatically build the rationale RDF.

3.2 An example illustrating the execution of the StdTrip+K process

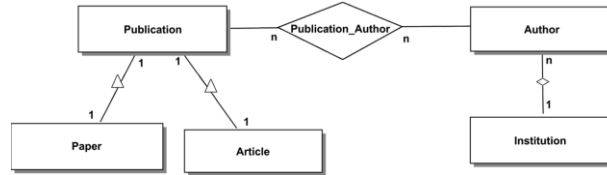


Fig. 3. Author-Publication ER diagram [14]

For the example we use the publication database depicted in Fig. 3. It is important to note that we implicitly assume that the input database is fully normalized, i.e., the input to the conversion stage must be in third normal form (3NF). Furthermore, we also assume that the user who follows this approach has some knowledge about the application domain of the databases. The result of the complete rationale captured can be seen in the illustration of the final stage (Fig. 4).

3.2.1 Stage 1 - Mapping

The general goal of this stage is to map the structure of the input relational database schema onto intermediate database ontology (we call OWL') and to trace the DR for the mapping (we call DR1). OWL' is not the final ontology because there is no execution of matching algorithms in this stage yet. To achieve the general goal, there are two sub stages: **(1.1) RDB-to-ER**, to transform the relational database schema into an Entity Relationship (ER) model and **(1.2) ER-to-OWL'**, to transform the ER model into an OWL ontology (OWL'). The rationale captured in this stage records the mapping rules used in the mapping since it is part of the domain expert decisions. The resulting (yet intermediate) OWL ontology is a model that simply mirrors the schema of the input relation database. To illustrate the rationale representation, we will consider only the part of the input database example regarding the mapping of the *Author* and *Institution* classes with their attributes and the relationship established between them, *ex:WorksFor*. We list the K-steps executed to capture the DR 1: **K1 - Identify reasoning elements from the ER model**. The reasoning elements *last_name*, *author*, *Author_Institution* and *Institution* were identified, because all of them are elements that will be mapped; **K2 - Identify the representation of the reasoning element in the ER model**. After having identified each reasoning element, the rationale representation records which element (Entity, Attribute, Relationship) it represents in the ER model, in order to keep the traceability of each element; **K3 - Record the corresponding mapping of the ER element onto the OWL element**. Having identified all the ER elements, the DR model records the correspondent OWL element mapped for each reasoning element; **K4 - Record the argument for the mapping**. For each reasoning element, the argument is the respective mapping rule used in the mapping. As the mapping rules are not rigid nor a consensus, this step records how each element was mapped as an argument form; **K5 - Record the corresponding OWL in-**

intermediate term. Finally, this step records the intermediate term mapped for each element.

3.2.2 Stage 2 – Matching

The general goal of this stage is to find correspondences between the intermediate ontology obtained in the previous stage (Stage 1 - Mapping) and standard well-known RDF vocabularies. This stage comprises three sub stages: **(2.1) Matchers execution** – For each element in the intermediate ontology, there are partial candidates according to each matcher, with their respective similarity values. **(2.2) Combination strategies** – aggregation strategies are applied to define a unified similarity value for each pair of ontology terms. **(2.3) Selection of match candidates** – until here there is still more than one match for each term, so the final sub stage aims at applying a selection strategy to choose one final match candidate for each ontology term. The steps for the DR representation of Stage 2, DR 2 are: **K6 - Record the candidates for each intermediate term.** It records each candidate that is presented to the domain expert; **K7 - Identify and record the arguments (*in favor of* and *objects to*).** For each candidate, there is a final similarity value that represents the reason for this candidate to be part of the list presented to the domain expert. As the Kuaba+W DR model defines arguments as “*in favor of*” and “*objects to*”, they have to be identified and traced to keep all options the user currently have to make his or her decision. Due to space constraints, we illustrate only one case of different options with arguments *in favor of* and *objects to*, associated to *ex:last_name* example.

3.2.3 Selection Stage

The general goal of this stage is to select the terms resulting from the previous stages in order to build the final OWL ontology. In this stage, user interaction plays an essential role. Ideally, the user should know the application domain because he or she has to select the vocabulary elements that best represent each concept in the database. Similarly to the previous DR models, the DR of this stage (DR 3) is incrementally built from the preceding DR (DR 2) executing the following steps: **K8 – Record the user decision domain about each term.** The Kuaba+W model records all decisions involved in the acceptance (A) or rejection (R) of each term recommended by StdTrip+K. In the DR 3 model, these decisions are represented by the letters *A* and *R*, respectively; **K9 – Record the justification of the domain expert.** After each decision, the user expert justifies his or her choices. An example that represents the relevance of tracing the DR of this stage is related to the term *ex:last_name*, for which the expert domain decided to use the term with the lowest similarity value, and without the DR it would not be possible to know why.

3.2.4 Inclusion Stage

The general goal of this stage is to complete the final OWL ontology with terms that were not identified in the previous stages. This can happen when the Selection stage

does not yield any result or when none of the suggestions in the list is considered adequate by the user. The DR 4 is recorded through the following step: **K10 – Record the new term and the justification.** The expert domain justifies the inclusion of a description which explains why this is an appropriate term in the input database context.

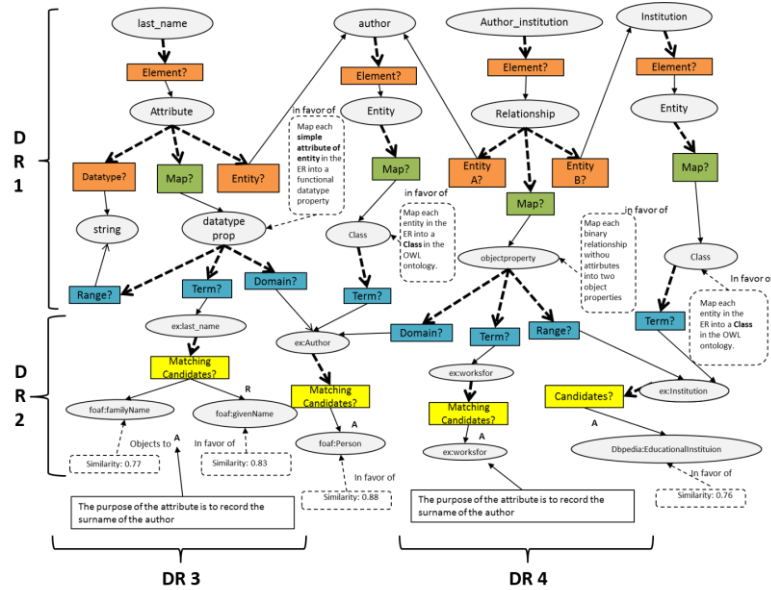


Fig. 4. Resulting design rationale captured for the example

4 Conclusions and future works

In this article, we introduced the StdTrip+K process. It allows the translation of a relational database to RDF triples reusing standard vocabularies and recording the DR from the translation. The StdTrip+K provides objective information about the RDB-to-RDF process and it is possible to answer the questions that still arise when using triple sets in the Linked Data cloud. (1) *Has the database suffered changes when published as RDF triples that could impact in its quality? May the original relational database have lost some relevant information when it is published as RDF triples?* As the DR shows the original form of the dataset (as ER model), it is possible to compare the database initial form and the mappings, and, consequently, evaluate the differences impact, if any. (2) *Is the chosen ontology the most appropriate to represent the database? Is the translation correct from the original relational database to RDF?* Once DR shows the options abandoned; accepted and the reasons for that, it is possible to evaluate the choices done. Also, the DR allows having access of one-to-one and one-to-many mappings despite not having been addressed in the running example of

this article. We believe our work can be further improved as follows: Implementing the reuse of DR in the mapping process, adding recommendation functionality in the StdTrip+K making use of previous decisions regarding abandoned options in similar domains; Providing a more compact visualization of the captured DR allowing a detailed visualization just when required by the triple set consumer; and Incorporating the rationale model to other RDB-to-RDF strategies that presents different characteristic from StdTrip. The last further work emphasizes that the rationale model may be adapted to capture the triplification rationale in other RDB-to-RDF processes and it is not a specific solution for StdTrip approach.

5 References

1. Auer S, Dietzold S, Lehmann J, Hellmann S and Aumueller D (2009) Triplify: light-weight linked data publication from relational databases. In: WWW '09, pp. 621–630. ACM, New York, NY, USA.
2. Bizer C and Seaborne A (2004) D2RQ-treating non-RDF databases as virtual RDF graphs. In Proceedings of the 3rd International Semantic Web Conference (ISWC2004).
3. Bizer C, Heath T, Berners-Lee T (2009) Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems (IJSWIS), Vol. 5, No. 3., pp. 1-22.
4. Cerbah F (2008) Learning highly structured semantic repositories from relational databases. The Semantic Web: Research and Applications, pp. 777–781.
5. Cullot N, Ghawi R and Yétongnon K (2007) DB2OWL: A Tool for Automatic Database-to-Ontology Mapping, SEBD, pp. 491–494.
6. Dividino R, Schenk S, Sizov and Staab S (2009) Provenance, Trust, Explanations – and all that other Meta Knowledge. Künstliche Intelligenz. KI 23 (2):24-30.
7. Erling O and Mikhailov I (2009) RDF support in the virtuoso DBMS. Networked Knowledge-Networked Media, pp. 7–24.
8. Lee J (1997) Design Rationale Systems: Understanding the Issues. IEEE Expert Volume 12, No. 13, pp 78-85.
9. Lee J, Lai K (1991) What's in Design Rationale. Human-Comput. Interaction, No. 6 (3-4), pp 251-280.
10. Maclean A, Young R, Bellotti V, Moran T (1991) Questions, Options, and Criteria: Elements of Design Space Analysis. Human-Comput. Interaction, No. 6 (3-4), pp 201-250.
11. McGuinness D and Harmelen F (2004) OWL web ontology language – W3C Recommendation. Retrieved Feb 2013. <http://www.w3.org/TR/owl-features/>.
12. Medeiros AP, Schwabe D (2008) Kuaba approach: Integrating formal semantics and design rationale representation to support design reuse. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, v. 22, p. 399-419.
13. Prud'hommeaux E, Hausenblas, M (2010) Use cases and requirements for mapping relational databases to rdf. Retrieved November, 27, 2012 from www.w3.org/TR/rdb2rdf-ucr/.
14. Salas P, Viterbo J, Breitman K, Casanova MA (2011) StdTrip: Promoting the Reuse of Standard Vocabularies in Open Government Data, In: D. Wood (ed.) Linking Government Data, Springer Verlag , pp. 113–134.
15. Sequeda J, Depena R, Miranker (2009) Ultrawrap: Using SQL views for RDB2RDF. In Proceedings of International Semantic Web Conference. ISWC 2009.