Semion: a smart triplification tool

Andrea Giovanni Nuzzolese¹, Aldo Gangemi¹, Valentina Presutti¹, Paolo Ciancarini²

¹ Semantic Technology Lab, ISTC-CNR, Rome, Italy
² Dep. of Computer Science, Università di Bologna, Italy

Abstract  Data being published on the Web in order to bootstrap the Web of Data are typically triplified by using bulk recipes. Reengineering cannot be easily customized. Furthermore, their consumption or refactoring is often difficult due to mismatches between the semantics embedded in original structures, and the RDF or OWL semantics obtained through the recipes. Semion is a method and a tool for customizing and explicating the semantics of data reengineering and refactoring.

1 Introduction

Traditionally, data published on the Web has been made available as raw dumps in formats such as CSV [20] or XML [13], or marked up as HTML [12] tables, sacrificing much of its structure and semantics. In the conventional hypertext Web, that can be identified by the expression “Web of documents”, the nature of the relationship between two linked documents is implicit, as the data format (e.g. HTML) is not sufficiently expressive to enable individual entities described in a particular document to be connected by typed links to related entities. [4] However, in recent years, the Web has evolved from a global information space of linked documents to one where both documents and data are linked. Underpinning this evolution is a set of best practices for publishing and connecting structured data on the Web known as Linked Data. The original and ongoing aim of Linked Data is to bootstrap the Web of Data by identifying existing data sets that are available under open licenses, converting these to RDF according to the Linked Data principles, and publishing them on the Web. To this purpose, commonly accepted solutions for transforming non-RDF data sources into RDF are based on predefined domain semantics-driven approaches, that make implicit assumptions on the domain semantics of the non-RDF data source (e.g. a relational database is transformed mapping a table into an rdfs:Class, a table column into an rdf:Property and a table record into an instance of the specific RDF table class). These implicit assumptions do not allow:

- the customization of the transforming process (i.e. the user cannot define what and how has to be transformed)
- to adopt good design practices for the domain modelling (e.g. ontology design patterns)
- to use the expressibility of OWL [6] for describing the domain, in order to be better supported by inference engines.
Our solution is based on a method that makes no semantics assumption at the domain level and just transforms the data source into RDF triples driven by the OWL description of the data source structure, that can be defined and customized by the user. The lifted RDF triples can be modelled by aligning them any additional RDF or OWL ontology, which acts as either a meta-level “mediator” to the required semantics (e.g. SKOS or the OWL metamodel), or as a reference domain ontology (e.g. DOLCE or Gene Ontology). In particular, we exemplify the alignment of triplified data with the Linguistic Meta-Model (LMM) [10], an OWL-DL ontology that formalizes the distinctions of the semiotics. The most important feature of LMM is its ability to support the representation of different knowledge sources developed according to different underlying semiotic theories [10]. This is possible because most knowledge representation schemata, either formal or informal, can be put into the context of so-called semiotic triangle introduced by Peirce[10]. Furthermore the method allows the user to model the triples by using the ontology desing patterns. The contribution of this paper is a method, a tool that implements the method and an evaluation of the tool and, of course, of the quality of the method. According to the aims of the paper, in section 2 are listed and briefly explained the related works to Semion. In section 3 is described the method that is the core of Semion. Section 3 contains details regarding the tool that was designed and implemented, namely Semion. Section 5 contains the results of Semion evaluation made on the transformation of the WordNet database. Finally in section 6 there are the conclusion and the future work.

2 Related work

Some existing tools make possible to expose and publish non-RDF legacy data sources into Linked Data. **D2R Server** [3] is a tool for publishing relational databases on the Semantic Web. It enables RDF and HTML browsers to navigate the content of the database, and allows applications to query the database using the SPARQL query language. D2R Server uses the D2RQ Mapping Language to map the content of a relational database to RDF. A D2RQ mapping specifies how resources are identified and which properties are used to describe the resources. The **Talis Platform** [19] provides Linked Data-compliant hosting for content and RDF data. Data held in the platform is organized into “stores” which can be individually secured if need be. Any kind of content can be added to a store along with arbitrary RDF metadata. The content and metadata becomes immediately accessible over the Web and discoverable using both SPARQL [11] and a free text search system with built in ranking of results according to relevance to the search terms. **Triplify** [2] is a small plug-in written in PHP for Web applications, which reveals the semantic structures encoded in relational databases by making database content available as RDF, JSON or Linked Data. **Virtuoso** [18] is a middleware and database engine hybrid that combines the functionality of a traditional RDBMS, ORDBMS, virtual database, RDF, XML, free-text, web application server and file server functionality in a single system.
Rather than have dedicated servers for each of the aforementioned functionality realms, Virtuoso is a “universal server”; it enables a single multithreaded server process that implements multiple protocols. QuOnto [1] is a JAVA ontology representation and reasoning tool implementing the DL-Lite family of ontology representation languages and uses relational DBMSs to store the extensional level of the ontologies and uses the Ontology Based Data Access (OBDA), that provides access to heterogeneous data sources through a mediating ontology. METAmorphoses [14] is a set of tools for flexible and easy-to-use generation of RDF metadata directly from a relational database. Metadata are generated according to the mapping from an existing database schema to a particular ontology. Krextor [7] is an extensible XSLT-based framework for extracting RDF from XML, supporting multiple input languages as well as multiple output RDF notations.

The motivations behind the design and the implementation of Semion have to be found in the transformation method that it implements. In fact, unlike existing tools, it allows to obtain high-quality dataset basing the transformation process first on the reengineering of the non-RDF data source to RDF with no domain semantics assumption and then on the refactoring of the latter with three alignments steps that want to add semantics to the reengineered RDF dataset in order to model it first in a semiotic-cognitive way according to the Linguistic Meta-Model (LMM) [10], then to a meta-model that allows to describe the dataset in terms of formal semantics and finally to a meta-model that allows to describe the dataset in terms of logics. With Semion it is possible to have high-quality datasets thanks to the alignment steps and to the fact that they ground the datasets themselves to a logics vocabularies (e.g. OWL) that can be inferred by reasoners.

A related technology is the Simple Knowledge Organization System (SKOS) [8]. SKOS is a common data model for knowledge organization systems such as thesauri, classification schemes, subject heading systems and taxonomies. SKOS aims to provide a bridge between these communities of practice within the library and the Semantic Web, by transferring existing models of knowledge organization to the Semantic Web technology context, and by providing a low-cost migration path for porting existing knowledge organization systems to RDF. The basic idea behind SKOS, that is in common with our method for reengineering non-RDF sources, is to represent the elements of the SKOS data model as classes and properties, and the “concepts” of a thesaurus or classification scheme as individuals in the SKOS data model. The informal descriptions and the links between those “concepts” as given by the thesaurus or classification scheme are modeled as facts about those individuals, never as class or property axioms [8].

3 Method

A very common approach for transforming non-RDF data sources into RDF is based on “bridges”. A bridge is an RDF description for mapping a non-RDF resource into an RDF one, e.g. a Class bridge maps into an rdfs:Class and a
Property bridge maps into an \textit{rdf:Property} the resources that the bridges describe. Tools based on this approach hide the choices thought by the designer for the transformation and generally follow approaches based on implicit semantic assumptions at the domain level. Contrariwise, our method is based on an approach that substantially divides the reengineering process from the modelling one, allowing an high quality degree of the obtained RDF data set. The reengineering process performs the lifting from the original data source to RDF making no semantic assumption at the domain level, but just extracting RDF triples driven by the OWL semantic description of the structure and the RDF description of the mapping between the data source object and resources from the structure description itself. Both descriptions have to be treated as inputs provided by the user. On the other hand, the modelling process allows to introduce semantics in the extracted data set according to good knowledge engineering practices derived by the use of the ontology design patterns. The general modeling aim is to represent data in a semiotic-cognitive way and grounds them first in a formal semantics and finally into logics. An ontology able to represent knowledge in a semiotic-cognitive way is the Linguistic-Meta Model (LMM) [10]. The most important feature of LMM is its ability to support the representation of different knowledge sources developed according to different underlying semiotic theories [10]. This is possible because most knowledge representation schemata, either formal or informal, can be put into the context of so-called semiotic triangle [9]. Figure 1 shows the basic key concepts that are behind our transforming method. The “Data source” double represents the input consisting of a non-RDF data source that is reengineered into an RDF data set according to its type, to its structure described by an OWL meta-model and to a defined mapping. The RDF dataset is then refactored (“Refactoring process” frame in figure 1) to the LMM framework according to specific customizable alignment rules. Once the RDF dataset is aligned to LMM it is possible to grounds it to a formal semantics and finally to express its logics.

3.1 The reengineering process

Our reengineering is based on a triplification process that uses the semantic description of the data source itself as meta-model for data and makes no mapping semantic assumption at domain level. The meta-model has to be designed in order to represent the semantics of the data source as it is physically. For a relational database it should be provided a meta-model like that one in figure 2. In that meta-model are expressed the main concepts from the relational database domain (i.e. such as tabl, field, query, result set, etc...) as OWL classes, while the concrete schema and data of a specific database are expressed as individuals of this meta-model, e.g. a table “Customer” will be represented as an individual of the class “Table”. Relations between those tables as given by the database schema are modeled as facts about those individuals, never as class or property axioms. As a consequence, the semantics of the reengineered dataset will be that one provided by the meta-model and there is no implicit mapping to the domain semantics as it happens in the “bridge” approaches. What just said
about relational databases can be suitable for any structured data source. As a further example we may consider the meta-model in figure 3, that was designed
for representing PGN files used for storing chess games. The approach followed for reengineering a PGN data source to RDF triples is completely similar as before, what is changed is the meta-model that describes the source, so that, for example, the moves of a game stored in the PGN store will be represented as individuals of the class `ChessMove`.

![Figure 3. A meta-model for representing PGN files.](image)

### 3.2 The refactoring process

In order to add domain semantics, the dataset refactored. Our method provides an approach based on a semiotic-cognitive representation of the knowledge with LMM coupled any other vocabulary (e.g. SKOS, ontology design patterns, etc...). LMM, expands all social-cognitive aspects as they are defined in DOLCE [5] and in the Descriptions and Situations framework, in order to adapt them to a semiotic perspective. In addition, LMM is able to deal with multilinguality and to represent individuals and facts in an open domain perspective.

Figure 4 shows the core ontology of LMM that are directly bind to concepts of the semiotic triangle, namely:

- Reference is populated by any possible individual belonging to the universe of discourse, including e.g. physical objects, events, etc., and they have a explicit reference “in the world”.
- Meaning, that is populated by concepts that are related between each other both in hierarchies of subclasses by subsumption relations and by descriptions expressing the possibility for events to occur.
Expression, that is populated by social objects produced by agents in the context of communicative acts. They are natural language terms, symbols in formal languages, icons, and whatever can be used as a vehicle for communication.

Figure 4. The semiotic triangle in LMM [10].

Considering a relational database example, the refactoring to LMM can be performed making this kind of assertions:

Table(?x) → LMM_L1:Meaning(?x)
Record(?x) → LMM_L1:Reference(?x)

The previous assertions state that, starting from the database meta-model

– a table is a Meaning
– a record is a Reference

LMM follows an approach similar to SKOS modelling the data source into individuals its meta-model, but it is not a formal knowledge representation language, as the structure of the data source need to be transformed into a set of formal axioms and facts. For this reason, the LMM aligned datased need to be refactored to any ontology/vocabulary that is able to contain the distinctions of formal semantics and is useful to define the formal semantics of informal resources “when needed”, and independently from a specific logical notation or syntax. For instance, to this purpose it can be chosen the ontology FormalSemantics.owl.

According to the FormalSemantics ontology it would be possible to assert that a Concept is a Class of FormalSemantics, namely:

LMM_L1:Concept(?x) → FormalSemantics:Class(?x)

Finally it is possible to grounds the dataset in formal logics by aligning it to any possible ontology/vocabulary that expresses logics, e.g. the OWL [6] vocabulary
itself, that formally expresses description logics, e.g. a rule to express a possible alignment from a Class of FormalSemantics to an OWL Class can be:

\[ \text{FormalSemantics:Class}(?x) \rightarrow \text{OWL:Class}(?x) \]

4 Tool

The method described in the previous section is implemented in Semion. Semion is available both as reusable components organized in two main Java libraries (i.e. it.cnr.istc.semion.reengineer and it.cnr.istc.semion.refactorer that are called SemionCore), that can be imported and used in other applications in order to perform the reengineering of non-RDF sources, and as a standalone graphical tool based on the Standard Widget Toolkit (SWT) \[16\] and JFace \[15\]. The graphical tool imports the two libraries adding a user interface that enables an easier use of their functionalities. The tool was designed according to the Model-View-Controller pattern, in which the view and the controller are part of the user interface, while the models are provided by the SemionCore libraries. Currently the tool is still a prototype and has been tested only for transforming relational databases, but it was designed to perform the transformation of any kind of data source. The figure 5 shows the reengineering interface of the Semion tool. It allows the user to define the meta-model of the database structure. Both because the manual definition of the this meta-model can be laborious and because the

![Figure 5. Semion tool: view of the reengineering interface.](image-url)
tool should be addressed also to non expert users in semantic technologies, it is provided an automatic acquisition of the meta-model by the tool. The interface also provides a window that allows to edit directly the source code of the meta-model in RDF/XML, RDF/XML-ABRREV or N3 and a window that provides a SPARQL [11] editor in order to query and retrieve data, expressed as individuals of specific declared classes in the database meta-model. Once the database is reengineered into an RDF dataset, it does not contain any domain semantics, but just that one deriving from the structure of the physical database itself, expressed by the meta-model previously defined. The refactoring interface allows the user to align the dataset to specific ontologies for adding semantics to data. The alignment ontologies can be chosen following the method the Semion implements i.e. first the dataset is aligned to the LMM framework, then to an ontology that contains the distinctions of the formal semantics and finally to an ontology that contains the logics. Semion performs ontology alignments through SPARQL CONSTRUCT, that are obtained from the rules written in human-readable syntax (see figure 6, that are based on the form:

\text{antecedent} \rightarrow \text{consequent}

Using this syntax, a rule asserting that being an instance of class \text{Table} in dataset meta-model implies to be a \text{Concept} of DOLCE [5] would be written:

\text{dbs:Table(?x)} \rightarrow \text{DUL:Concept(?x)}

This rule will be interpreted as the SPARQL query:
CONSTRUCT { ?x rdf:type DUL:Concept. }
WHERE { ?x rdf:type dbs:Table. }

With the Semion tool it was performed a transformation of the WordNet database from its MySQL version to OWL. The meta-model adopted as vocabulary for database was that one showed in figure 2 and the source was reengineered in RDF triples, such as the piece shown in figure 7. Form this dataset was derived an LMM aligned dataset according to some refactoring rules as shown in figure 6. In figure 8 is shown the Semion tool interface for refactoring during the alignment to the FormalSemantics vocabulary. It is possible to notice both the rules and a fragment of the obtained source code. Finally in figure 9 are shown the details of the Semion tool for the refactoring of the dataset to the OWL vocabulary. It is possible to use the tool also for transforming datasource different from relational database, such as PGN files, whose meta model was introduced in section 3. The refactoring of the dataset extracted from the PGN can be performed with the same approach as seen before.

Figure 7. Piece of the reengineered RDF data set from the WordNet database.
5 Evaluation

Semion aims at maximal flexibility either for a user who wants to customize reengineering and refactoring of ontologies to the tiny detail (e.g. for legacy databases, idiosyncratic data sources, metamodeling, etc.), or for a user who wants to reuse “good practices” such as recipes to convert typical kinds of databases,
thesauri, etc., and does not want to know anything about the reengineering and refactoring clockwork. The evaluation is based on comparison of Semion with other related tools according to a well defined set of criteria. The tools chosen for the evaluation are D2R [3], Triplify [2] and METAMorphoses [14]. D2R transforms relational databases in a different way respect to Semion, since it allows to access the source as it was an RDF graph translating, through a mapping file, SPARQL queries into SQL queries. The mapping file can be configured, but the transforming choices are made implicit in the “bridges” that it realizes. Triplify reveals the semantic structures encoded in relational databases, but the transformation and the domain semantics is not configurable as it is in Semion. METAMorphoses has some similarity to Semion, since it allows to configure the mapping from the data based on a metamodel, and to map the result to another ontology. However, it is not clear if the mapping can be made by means of any (even customized) metamodel for the source data and if the second mapping can be iterated to other ontologies. The criteria for the evaluation and how Semion and the other tools cover them are shown in table 1. According to the table and to the evaluation, all the examined tools are able (with different approaches) to reengineer non-RDF data sources to RDF, but only Semion is completely able to reengineer also the original schema of the data source, while the other tools uses a reference to the schema (like the mapping file in D2R) for accessing and extracting data, or uses a representation of whole structure of a database for mapping the resources in it to a third ontology that adds semantics (METAMorphoses). By full extendibility we mean the possibility for a user to configure, customize, or create a metamodel for the data source. The customization of the source reengineering is not supported by Triplify and partially supported by D2R and METAMorphoses, as the former allows customization only in the definition of the mapping file, while the latter allows customization only in the choice of the domain ontology to which data are mapped. By full extendibility to other source types we mean the possibility for a user to configure, customize, or create a metamodel for a specific data source (not necessary a relational database). This criterion is completely within the scope of Semion and it is only partially supported by Triplify, that supports the generation of provenance information for data in the config file. Triplification of the original data structure means that the source is reengineered making no semantic assumption at domain level, but keeping the semantics of the source that stores the data as meta-model of data themselves. This is a crucial point to distinguish Semion from the comparison tools, as they follow the “bridge” approach described in section 3. The criterion “open the workflow of alignments to arbitrary ontologies” is only supported by Semion, because the method that it implements is based on a workflow of alignments in order to model and to add domain semantics to data. Only Semion and METAMorphoses have the possibility to choose a translation recipe (“support for translation recipes” criterion), e.g. they can assume as target ontology for data SKOS or LMM, while D2R partially covers this criterion because it is possible to specify references to a target ontology in the mapping. Furthermore Semion is compliant with the good reengineering practices as the Ontology De-
sign Patterns (ODP) [17]. A full semantic web-based pipeline means that the tool uses in the reengineering pipeline only Semantic Web techniques, e.g. SPARQL CONSTRUCTs or SWRL rules.

Table 1. Functionalities implemented by Semion and comparation with other tools

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Semion</th>
<th>D2R</th>
<th>Triplify</th>
<th>METAMorphoses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reengineering RDF data from a non-RDF data source</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Reengineering non-RDF data source schemata to RDF</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Customization of source reengineering</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Full extendibility to other source types</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triplification of original data structure</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open workflow of alignments to arbitrary ontologies</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for translation recipes</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>OWL(2) and reasoning support</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full semantic web-based pipeline</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6 Conclusion

In this paper we have presented Semion, a smart triplification tool that is designed for transforming to RDF heterogeneous non-RDF sources. The method that is behind the Semion desing is based on the general idea that the reengineering process need to be customzed according both to the data source structure and to the domain. For this reason, Semion provides a pipeline that aims to keep divided the pure lifting of RDF triples from the refactoring of these triples. The lifting, as shown in the previous sections, accepts as meta-model for the data an RDF/OWL description for the structure of the source, while the refactorings accept any recipes that can help to add the correct semantics to data, e.g. LMM, FormalSemantics, SKOS, etc.

Semion, unlike other existing tools, drives the user in the pipeline and allows to extract high-quality datasets both for the design and for the semantics they express.

The evaluation, shown is section 5, is encouraging and we are designing and implementing new features in order to make Semion completely able to manage the tranformation of any kind of source and to interact with other tools or components, e.g. through semantic web services, OSGi bundles, etc.
References


Webography