Knowledge Spring Process
Towards Discovering and Reusing Knowledge within Linked Open Data Foundations

Roberto Espinosa[1], Larisa Garriga[1], Jose Jacobo Zubcoff[2] and Jose-Norberto Mazón[3]
[1]*WaKe Research, Universidad de Matanzas "Camilo Cienfuegos", Matanzas, Cuba*
[2]*WaKe Research, Dept. de Ciencias del Mar y Biología Aplicada, Universidad de Alicante, Alicante, Spain*
[3]*WaKe Research, Dept. Lenguajes y Sistemas Informáticos, Instituto Universitario de Investigación Informática, Universidad de Alicante, Alicante, Spain*

Keywords: Linked Open Data, Knowledge Discovery, Data Mining, Metamodeling, Model Driven Development.

Abstract: Data is everywhere, and non-expert users must be able to exploit it in order to extract knowledge, get insights and make well-informed decisions. The value of the discovered knowledge could be of greater value if it is available for later consumption and reusing. In this paper, we present the first version of the Knowledge Spring Process, an infrastructure that allows non-expert users to (i) apply user-friendly data mining techniques on open data sources, and (ii) share results as Linked Open Data (LOD). The main contribution of this paper is the concept of reusing the knowledge gained from data mining processes after being semantically annotated as LOD, then obtaining Linked Open Knowledge. Our Knowledge Spring Process is based on a model-driven viewpoint in order to easier deal with the wide diversity of open data formats.

1 INTRODUCTION

Nowadays, governments are worldwide generating open data for the sake of transparency. Besides, open data philosophy encourages the value of reusing data through participation and collaboration among citizens, public institutions and private organizations. The promise of open data is to improve citizens' life through (i) development of applications (Web, smartphones, etc.) that reuse and add value to existing open data, and (ii) data analysis to get new insights and acquire knowledge that support daily decision making process. However, new discovered knowledge from open data is not incorporated to open data sources again and it hampers its reuse. In order to overcome this situation, metadata on how the knowledge is discovered from data sources must be incorporated as new data sources. This metadata could be simple when a simple data analysis is performed, although some advanced analysis such as data mining requires specific techniques for dealing with metadata. Specifically, two problems need to be addressed when data mining is used: (i) regular citizens do not know how to extract insights from available open data (ii) once someone has done an analysis to discover some knowledge from data, it can not be reused and incorporated to the body of Linked Open Data (LOD) as new knowledge to be further reused. We addressed the first point in our previous work (Espinosa et al., 2013) and now, in this position paper, we start focusing on the second point by considering required metadata. Our hypothesis is that open data is increasingly available and reused by applying RDF schemas and getting LOD, so RDF can be also used to maximize reusing discovered knowledge from data mining results, since knowledge is included again in the LOD sets as a new resource.

We found many heterogeneous formats when examined open data sources in the Web, (CSV, JSON or RDF, etc.). Lately, RDF is proposed as the most adequate format for reusing data through LOD concepts. Therefore, when data is processed, metadata and results must be also converted into RDF to take part in the LOD sets, thus empowering their reuse. To do so, a model-driven development approach is proposed in this paper. This approach allows us to obtain homogeneous models from any kind of data source format. A knowledge discovery process is then applied and results are labeled with RDFs and incorporated as LOD. Due to the iterative nature of this process we call it the Knowledge Spring Process.

Next, we exemplify a possible scenario in which

Espinosa R., Garriga L., Jacobo Zubcoff J. and Mazon J..
Knowledge Spring Process - Towards Discovering and Reusing Knowledge within Linked Open Data Foundations.
DOI: 10.5220/0005111102910296
In *Proceedings of 3rd International Conference on Data Management Technologies and Applications* (DATA-2014), pages 291-296
ISBN: 978-989-758-035-2

our approach could be useful within the data journalism domain. A journalist requires applying mining techniques to know some patterns in the expenses done by candidates of several political parties in some election campaigns. Knowledge provided by this study is useful as it can be accessed and reused by other actors in combination with other existing open data, for example watchdog organizations.

2 ENABLING NON-EXPERT USERS TO USE DATA MINING

Nowadays, the increasingly use of ICT allows us to quickly access huge amount of information and support our daily decision making process. For example, someone using a Web search engine always expects the best answer in the shortest time in order to make some decisions on travelling, shopping or any other daily-life task. However, discovering advanced knowledge from available data (for example, patterns) requires that an expert take part in the process. Importantly, the process of knowledge discovery has historically linked up only to experts in data mining. Due to the fact that it is an intrinsically complex process (Nisbet et al., 2009; Vanschoren and Blockeel, 2009) and a great variety of techniques and algorithms of data mining exist. Unfortunately, sweeping changes are needed, in order to create mechanisms that allow non-expert users to consume the available information and discover useful knowledge. User-friendly data mining(Kriegel et al., 2007) is a step forward in this direction, since it fosters knowledge discovery without mastering concepts and data mining techniques. To realize user-friendly data mining, in our previous paper(Espinosa et al., 2013) a methodology that enables “masses” to apply data mining was presented.

Now, in this paper, we present a framework for supporting reusing discovered knowledge within an open data scenario. To this aim, several challenges are tackled (see Fig.1):

1. Overcoming non-expert difficulties when they try to use the diversity of existing open data formats.
2. Proposing a mechanism that allows non-expert users to identify their “knowledge discovery” requirements.
3. Facilitating knowledge discovery from Open Data sources by using data mining techniques without an expert.
4. Creating a mechanism that allows reusing the previously discovered knowledge by non-expert users.

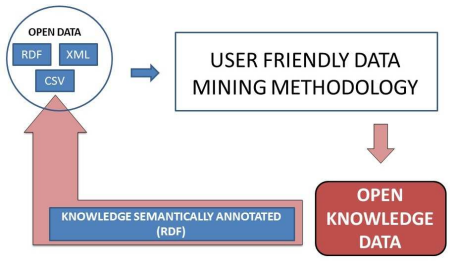

Figure 1: Our proposal for reusing knowledge.

Points 2 and 3 was considered in(Espinosa et al., 2013) and they are out of the scope of this paper; while it addresses points 1 and 4.

3 OPEN DATA FORMATS

Our Knowledge Spring Process begins when an user wishes to analyze some open data sources with the main objective of knowledge discovery (i.e., getting insights from data and making a right decision). Open data sources are available in many formats: RDF, JSON or CSV, etc. Each of those format structure information differently. Understanding each of them is quite difficult for non-expert users that try to acquire the right knowledge in order to make decisions. Fortunately, we also found common elements in the diversity of open data formats, in such a way that we can propose a homogeneous model to represent data independently of its format. To this aim, a model-driven development approach is proposed in order to obtain the extracted information from the open data files in a standard way. To this end, the Data Metamodel designed is shown in the Fig.2.

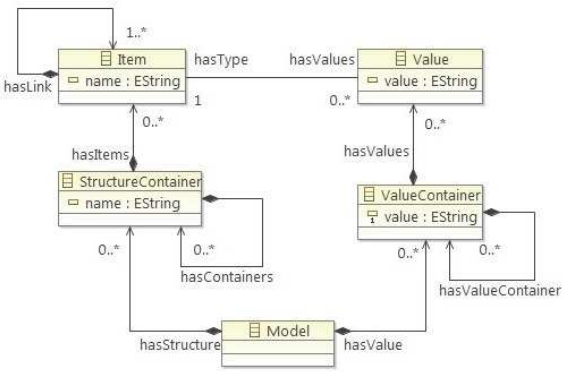

Figure 2: Metamodel for representing data sources.

3.1 Description of Data Metamodel

After examining the way in which the open data is structured, a metamodel is defined in order to repre-

sent data concepts and their relationships in a general manner. The goal is that this metamodel could be simple enough for defining any data representation by using models conforming this metamodel without losing schema semantics. Generally speaking, any data source can be always become into this kind of models. Next, we will explain in details the included concepts in our metamodel:

- Model: This is the main container class in the metamodel, it has associated `StructuralContainer` and `ValueContainer`. These elements could be accessed from the attributes `hasStructural` and `hasValue`, respectively.
- StructuralContainer: Store the types of data `hasItems`, that are elements of type `Items`. It can contain other `StructuralContainer` elements. The `name` attribute is the key of the container.
- ValueContainer: This class contains elements of type `Value`, through of the `hasValues` relation. They can contain other `ValueContainer` elements. The `value` attribute is the key of the container.
- Item: Store the types of data in the `name` attribute. An `Item` element can contain some values through the `hasValues` relation. An `Item` can have links to other elements of type `Item`, through the `hasLink` property.
- Value: contain the values in the `value` attribute. These values belong to a type `Item` and they can be accesed as from the property `hastype`.

3.2 Obtaining the Data Model

In this section we point out how an input open data source is transformed towards the data model according to the metamodel previously presented. Taking into account the possible schemes of the files that appear in the open portals (RDF, JSON or CSV), we have implemented the correspondent grammars. Grammars were implemented using the XText framework[1] of the Eclipse platform[2]. Once the grammar was defined we can obtain the correspondent models. Finally, for the sake of simplicity, we only focus on RDF files examples. The RDF metamodel is obtained from the implemented grammar, and is shown in the Fig.3.

When a file is introduced in the system, it is validated by the corresponding grammar in order to ensure a correct structure. For each one of the possible input schemas files, ATL transformations (Atlas Transformation Language) (Jouault and Kurtev, 2005) were implemented to obtain the correspondent model in accordance with the Data Model metamodel. Through this process the uniformity of data in the mining process is guaranteed. The formal definition of a transformation is implemented through ATL rules. The segment of code shows the rule that transforms the element *rdfSequence* of the source model into the *Item* element in the target model.

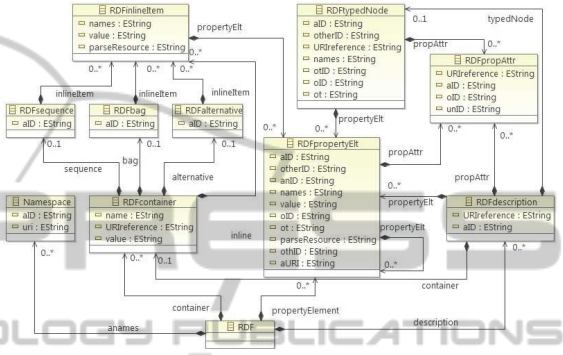

Figure 3: The RDF metamodel.

```
rule rdfseq2item{
 from rsq: RDF!RDFsequence(not(rsq.aID=OclUndefined))
 to di:DM!Item(name <- rsq.aID,
               hasValues <- rsq.GetSeqValue())
}
```

After obtaining the data model, the process continues applying the user-friendly data mining approach in order to discover knowledge (Espinosa et al., 2013). This approach is based on the construction of a knowledge base that contains information of previous data mining experiments in order to provide guidance to non-expert users to apply data mining techniques. To generate our knowledge base, a model-driven approach is defined, based on a Taverna workflow[3].

In the next section, we will discuss what treatment we give to the generated information after applying the aforementioned user-friendly data mining methodology.

4 OPEN KNOWLEDGE

The main advantage of our proposal consists of the potential for reusing the acquired knowledge in the data mining process. The contribution of our proposal is given by two fundamentals results to the user. The first one is the solution to the initial problem presented

[1] http://www.eclipse.org/Xtext/

[2] http://www.eclipse.org

[3] http://www.taverna.org.uk/

by the non-expert user, after being applied the corresponding model to input data according the suggestion of a recommender and their requirements. The other one is the generation of a RDF file semantically annotated with all the knowledge obtained in the data mining process. This output is very innovative because, in this way, useful information is shared and used by another user. See Fig. 4.

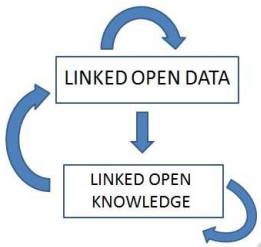

Figure 4: Generation of Linked Data Knowledge.

Users can reuse the LOD files published on the Web. While semantically annotated these may be associated with other LOD files. When these are analyzed by our proposal, they become Linked Open Knowledge files. The Linked Open Knowledge files can also be associated to other Linked Open Data files or Linked Open Knowledge files.

4.1 Description of the RDF Model

All the information that is generated in the applied data mining process (Espinosa et al., 2013) is stored in a model that conforms to the metamodel of Fig. 5. The question to solve is how a semantically annotated RDF file can be obtained that contains the information included in the returned model. To address this problem we present a metamodel according to RDF files, in order to apply ATL transformations between the DMKB metamodel and the proposed RDF metamodel.

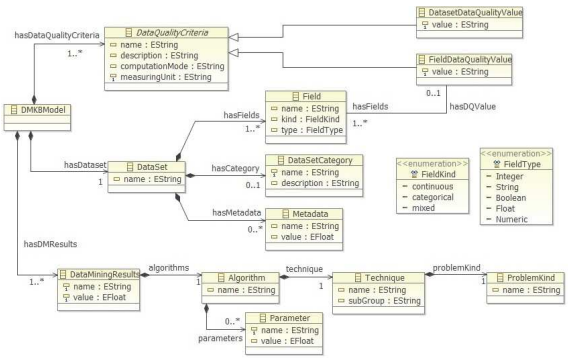

Figure 5: Our DMKB metamodel.

The RDF data model has been obtained from a corresponding grammar[4]. Also, it defines a simple model to describe relationships among resources in terms of designated properties and values. The basic RDF data model defined consists of three types of objects:

- Resources: All things described by RDF expressions are called resources. Resources are always denoted by URIs plus optional identifiers.
- Properties: A property is a specific aspect, characteristic, attribute, or relation used to describe a resource.
- Sentences: A specific resource together with a named property plus the value of that property for that resource is an RDF statement. These three individual parts of a sentence are known in the literature as a subject, predicate and object, respectively. The object of a sentence (ie, the value of the property) can be another resource or could be a literal; i.e., a resource (specified by a URI) or a simple string or other primitive data types defined by XML.

The main benefit of using a metamodel that represents RDF files is to generate the result that is in a DMKB model in a RDF file to included like linked open knowledge.

4.2 Mapping DMKB Model to RDF

We described the current prototype implementation which relies on model to model transformations. Transformations are the nucleus of the model-driven development. In this section, we describe the mapping between an input DMKB model to a RDF model. The transformation was specified by using the ATL plugin from the Eclipse platform[5]. We only present here some simple features of our transformations. In the Fig.6 an abstract of the informal definition was presented. Each *DMKBModel* model class is mapped to an *RDF* model class identified as being semantically convenient. In each class mapping, the properties of the DMKB class are mapped to equivalent properties in the RDF model. Transformation is automatically generated from a high-order transformation according to semantic links established in the mapping model (Hillairet et al., 2008a). The segment of code shows two examples of rules that allows to transform the *Algorithm* elements into their corresponding *RDFinlineItem* and, the *DataSet* element into the *RDFcontainer* in the destination model.

[4]Resource Description Framework (RDF) http://www.w3.org/TR/1999/REC-rdf-syntax-19990222

[5]http:/www.eclipse.org

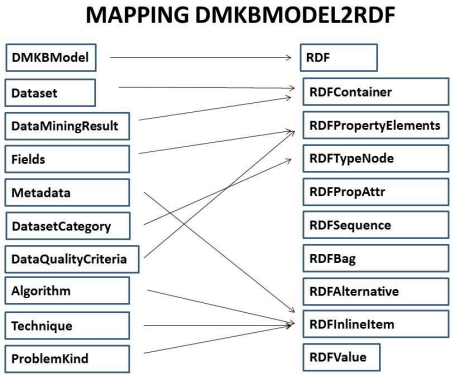

Figure 6: Mapping DMKBModel2RDF.

```
rule dataSet2container{
from dmkb:dmkb!DataSet
to   rdf:rdf!RDFcontainer( name <- 'DMKB:Dataset',
                           value <- dmkb.name,
              inline <- dmkb.GetInlineItemsMetadata())}

rule algorithm2rdfinlineitem{
from dmkb:dmkb!Algorithm
to rdf:rdf!RDFinlineItem(names<- 'DMKB:Algorithm_' +
    dmkb.GetParent(dmkb).Id.toString(),
    value <- dmkb.name)}
```

The execution of this transformation returns a RDF model which can be serialized in a RDF document containing the description of the EMF objects. Next, we get the following RDF document excerpt:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">
<rdf:RDFcontainer name="DMKB:Dataset" value="audiology"/>
 rdf:about="http://audiology.data.gov.uk/data/audiology/2012">
 <rdf:RDFinlineItem names="DMKB:Metadata_Instances" value="226.0"/>
 <rdf:RDFinlineItem names="DMKB:Metadata_Attributes" value="70.0"/>
 <rdf:RDFinlineItem names="DMKB:Metadata_Percentage of Numeric
  Attributes" value="0.0"/>
 <rdf:RDFinlineItem names="DMKB:Metadata_Percentage of Nominal
  Attributes" value="100.0"/>
 <rdf:RDFinlineItem names="DMKB:Metadata_Number of classes"
  value="24.0"/>
 ...
</rdf:RDFcontainer>
<rdf:RDFcontainer name="DMKB:DataMiningResult_Correctos"
 value="192.0">
 <rdf:RDFinlineItem names="DMKB:Algorithm_1" value="AdaBoostM1"/>
 <rdf:RDFinlineItem names="DMKB:Technique_1" value="Classification"
  parseResource="Classification"/>
</rdf:RDFcontainer>
<rdf:RDFcontainer name="DMKB:DataMiningResult_Correctos" value="172.0">
 <rdf:RDFinlineItem names="DMKB:Algorithm_2" value="END"/>
 <rdf:RDFinlineItem names="DMKB:Technique_2" value="Classification"
  parseResource="Classification"/>
</rdf:RDFcontainer>
...
<rdf:RDFpropertyElement aID="DMKB:DataSetDataQualityCriteria"
 otherID="Null Values" anID="Percentage of null values" value="2.0"
 oID="DMKB:DataSetDataQualityCriteria" ot="Null Values"/>
<rdf:RDFpropertyElement aID="DMKB:DataSetDataQualityCriteria"
 otherID="UnbalanceColumns" anID="Percentage of unbalance columns"
 value="71.43" oID="DMKB:DataSetDataQualityCriteria"
 ot="UnbalanceColumns"/>
<rdf:RDFpropertyElement aID="DMKB:DataSetDataQualityCriteria"
 otherID="Average Entropy" anID="Average Entropy" value="0.1428"
 oID="DMKB:DataSetDataQualityCriteria" ot="Average Entropy"/>
...
<rdf:RDF/>
```

The obtained knowledge is part of open data philosophy, being published and ready to be reused. The obtained knowledge forms a new layer on the same data for future analyses. The cycle is closed since this information can be examined and enriched insofar as the data is analyzed by our proposal. Knowledge will be enriched every time our proposal is used. Since this is an iterative process, more knowledge is generated each time, acting like a "Spring", so our proposal is named the Knowledge Spring Process[6]

5 RELATED WORK

The search for mechanisms to implement the complex process of knowledge extraction in data sources automatically is an issue raised by several authors for several years. Due to the complexity of this process has been addressed from several angles. In (Getoor and Diehl, 2005) Link mining is presented, focused on finding patterns in data by exploiting and explicitly modeling the links among the data instances. The authors tackling the problem of mining richly structured heterogeneous datasets. On the other side there are proposals related to OLAP tools to enable better decision-making process, mainly for integrating heterogeneous data sources. This is the case of (Kämpgen and Harth, 2011) where authors provided a mapping from statistical Linked Data into the Multidimensional Model used in data warehouses. They presented a mapping between statistical Linked Data that conforms to the RDF Data Cube vocabulary[7] to a common Multidimensional Model. In (Niinimäki and Niemi, 2009) mapping between the source data and its OLAP form is done by converting the data first to RDF using ontology maps. Then the data are extracted from its RDF form by queries that are generated using the ontology of the OLAP schema. They combine these two powerful technologies, Semantic Web and OLAP, resulting in a method of creating a fully functional tool for analysis. Alternatives that support the analysis of open data have appeared to publicize the importance of its use in making common decisions. In (Bizer et al., 2009) authors presented the concept and technical principles of Linked Data, and situate these within the broader context of related technological developments. Emphasis is done in this paper in all of the edges related with Open Data's use like road for progression and the development of the branches linked to the analysis of data with an eye to obtain knowledge. Local and national governments are turning to open data through out initiatives to disclosure and using data. In (Hoffmann, 2012) some of the encouraging expe-

[6]The "Spring" term has a double sense, because it is like the knowledge bloom.

[7]RDF Data Cube vocabulary http://www.w3.org/TR/vocab-data-cube/

riences are shown. We finally opted to use Model Driven Development in order to control the diversity of available data. Well defined languages can be designed by means of metamodeling (Bézivin, 2005), which provides the foundation for creating models in a meaningful, precise and consistent manner. Several solutions that use these technologies exist but no one focused on the analysis of heterogeneous data in order to apply mining techniques. In (Hillairet et al., 2008b) autors addressed the question of enabling the use of RDF resources as EMF objects, and presented a solution based on the EMF framework and the ATL model transformation language. This solution provides a prototype that offers a small Java library for the instantiation and serialization of EMF objects from, and to RDF resources.

As you can see there are some proposals to be focused in the data processing, in order to extracting knowledge. Most of them are partials solutions in order to resolve specifics problems, but we don't find a solution for non-expert user in order to discover and reuse knowledge within Linked Open Data foundations using data mining techniques.

6 CONCLUSIONS

Nowadays, it is essential that non-expert users can exploit the vast amount of information available in order to extract knowledge and make well-informed decisions. The value of the discovered knowledge could be of greater value if it is available for later consumption. In this paper, we present the first version of the Knowledge Spring Process, an infrastructure that allows non-expert users apply user-friendly data mining techniques in Open Data files. The main contribution of this paper is the concept of reusing the knowledge gained from data mining processes after been semantically annotated in the RDF file(Linked Open Knowledge). A model driven approach is used in order to maintain a standard structure having in account the diversity of the data formats. As future work, we plan to improve the process of obtaining Open Linked Knowledge.

ACKNOWLEDGEMENTS

This work is funded by IN.MIND project from University of Alicante and by the University Institute for Computing Research (IUII, http://www.iuii.ua.es/).

REFERENCES

Bézivin, J. (2005). On the unification power of models. *Software and System Modeling*, 4(2):171–188.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22.

Espinosa, R., García-Saiz, D., Zorrilla, M. E., Zubcoff, J. J., and Mazón, J.-N. (2013). Development of a knowledge base for enabling non-expert users to apply data mining algorithms. In *SIMPDA*, pages 46–61.

Getoor, L. and Diehl, C. P. (2005). Link mining: a survey. *SIGKDD Explorations*, 7(2):3–12.

Hillairet, G., Bertrand, F., and Lafaye, J. Y. (2008a). Bridging EMF applications and RDF data sources. In *4th International Workshop on Semantic Web Enabled Software Engineering*.

Hillairet, G., Bertrand, F., and Lafaye, J.-Y. (2008b). Mde for publishing data on the semantic web. In Parreiras, F. S., Pan, J. Z., Aßmann, U., and Henriksson, J., editors, *TWOMD*, volume 395 of *CEUR Workshop Proceedings*, pages 32–46. CEUR-WS.org.

Hoffmann, L. (2012). Data mining meets city hall. *Commun. ACM*, 55(6):19–21.

Jouault, F. and Kurtev, I. (2005). Transforming models with atl. In Bruel, J.-M., editor, *MoDELS Satellite Events*, volume 3844 of *Lecture Notes in Computer Science*, pages 128–138. Springer.

Kämpgen, B. and Harth, A. (2011). Transforming statistical linked data for use in olap systems. In Ghidini, C., Ngomo, A.-C. N., Lindstaedt, S. N., and Pellegrini, T., editors, *I-SEMANTICS*, ACM International Conference Proceeding Series, pages 33–40. ACM.

Kriegel, H., Borgwardt, K., Kroger, P., Pryakhin, A., Schubert, M., and Zimek, A. (2007). Future trends in data mining. In *Data Min. Knowl. Discov*.

Niinimäki, M. and Niemi, T. (2009). An etl process for olap using rdf/owl ontologies. *J. Data Semantics*, 13:97–119.

Nisbet, R., Elder, J., and Miner, G. (2009). *Handbook of Statistical Analysis and Data Mining Applications*. Academic Press.

Vanschoren, J. and Blockeel, H. (2009). Stand on the Shoulders of Giants: Towards a Portal for Collaborative Experimentation in Data Mining. *International Workshop on Third Generation Data Mining at ECML PKDD*, 1:88–89.