An infrastructure and approach for infering knowledge over Big Data in the Vehicle Insurance Industry

Aikaterini K. Kalou, and Dimitrios A. Koutsomitropoulos

High Performance Information Systems Laboratory (HPCLab),
Computer Engineering and Informatics Dpt., School of Engineering,
University of Patras, Building B, 26500 Patras-Rio, Greece

{kaloukat,kotsomit}@hpclab.ceid.upatras.gr

Abstract. In recent years, insurance organizations have turned their attention to tapping into massive amounts of data that are continuously produced accross their IT ecosystem. Even though the concept of Big Data provides the needed infrastructure for efficient data management, especially in terms of storage and processing, the aspects of Value and Variety still remain a topic for further investigation. To this end, we propose an infrastructure that can be deployed on top of the legacy databases of insurance companies. The ultimate aim of this attempt is to provide an efficient manner to access data on-the-fly and derive new value. In our work, we propose a Property and Casualty ontology and then exploit an OBDA system in order to leverage its power.

Keywords: Insurance sector, big data, ontology, linked data, OBDA.

1 Introduction

In recent years, an impressive growth of generated data has been considered across business and government organizations. New information is constantly added, and existing information is continuously changed or removed, in any format and coming from any type of data sources (internal and external). So, the manipulation of these massive amounts of data went beyond the power and the performance of conventional processes and tools. At the same time, the volume of data offers greater and broader opportunities for developing existing business areas or driving new ones by improving on insight, decision-making and detecting sources of profit [15,17].

Big Data [18], designated among the most common buzzwords dominating the IT world, can cover efficiently and effectively the aforementioned need and thus they are currently influencing most aspects of business units. In brief, the new technological achievement can be thoroughly described by the following characteristics: Variety (the number of types of data), Velocity (the speed of data generation and processing), Volume (the amount of data) [21] and Value (the knowledge). However, the lack of semantics seems to be restrictive towards the 4th V of Big Data. The aspect of the 1st V of Big Data also remains a topic for further discussion.

Insurance is a sector with traditionally high dependency in data availability as well

as knowledge-based decision making. Managing big volumes of insurance information not only has to face associated technical limitations, but also requires expressive knowledge analysis strategies to become valuable. Semantic technologies, including either the more lightweight (Linked Data) or the most expressive (ontology) version, can have considerable repercussions in addressing the Variety and the Value of Big Data for insurance. Regarding the Linked Data principles [11] the whole data management can be easily simplified. For example, data coming from unstructured databases and agents can be linked to data in traditional OLTP and OLAP stores without changing existing schemas.

On the other side, an ontology, by default, defines a common set of terms that are used to describe and represent the basic concepts in a domain and the relationships among them. With the power of reasoning, new facts, which are unexpressed explicitly in an ontology, can then be derived. Therefore, by exploiting the notion of ontology in the context of big data, new perspectives are added such as providing semantics to raw data, connecting data in various concept levels across domains and giving knowledge for further analysis, including more accurate risk identification and assessment. Customized insurance policies, fraud detection and marketing are only a few of the areas that can be identified as benefiting from applying semantic analytics methods over Big Data [9].

In this work, we make an attempt to take advantage of all the aforementioned benefits that Semantic Web technologies provide in terms of Big Data by implementing a real usage scenario in the Insurance sector. More precisely, based on the Property & Casualty Ontology [6] and a big dataset coming from an existing insurance company, we develop an infrastructure that could easily consume various data and infer knowledge in reasonable time and without sacrificing performance. The key component of the proposed infrastructure is an OBDA (Ontology Based Data Access [19,13]) system that ensures i) scalable big data infrastructure, ii) end-to-end processing of data and iii) managing the diverse roles of people in the data life cycle.

The rest of this paper is organized as follows. In Section 2, we start by discussing the usage of Ontologies in Insurance Business and providing some broad definitions and concepts about semantic technologies over Big Data and the approach of OBDA. Furthermore, in Section 3, we present in short the P&C ontology. Section 4 elaborates the architecture of our proposed system and the data manipulation strategy that we adopt. Next, Section 5 outlines several intelligent queries performed over real insurance data in order to illustrate the ability for knowledge inference. Finally, Section 6 summarizes our conclusions.

2 Background

2.1 Big Data Ontology Access

In recent years, quite a few data storage and processing technologies have emerged in terms of Big Data. Platforms like NoSQL, Yarn, Hadoop, MapReduce, HDFS are now some of the most familiar terms within this growing ecosystem [21]. On the semantic technologies' side, the "traditional" triplestores are continually evolving by following

the vision of exploring large data sets. Oracle Spatial and Graph with Oracle Database 12c , AllegroGraph, Stardog, OpenLink Virtuoso v6.1 are only some examples that expand their scalabilty and performance features to meet these premium requirements.

A lot of initiatives and synergies are on progress, having as main purpose the smooth integration of the widely-known Big Data technologies with whatever is coming from the area of semantic technologies. For example, AllegroGraph 6, has recently been released with a certification on Cloudera Enterprise[1], among the leader companies of Apache Hadoop-based software. Furthermore, AllegroGraph has implemented extensions allowing users to query MongoDB databases using SPARQL and to execute heterogeneous joins [10].

Besides all the above notable new features of triplestores, the issue of querying data from various legacy relational databases still stumbles on the required ETL (Extract – Transform - Load) processses [20]. In these cases, the data flows can be summarized in extracting the data from the database, transforming into triples and then storing in conventional RDF stores. The approach of OBDA systems seems to cope well with this scenario by leveraging the pure nature of ontology notion. With OBDA, an ontology is used so as to expose data in a conceptual manner by abstracting away from the schema-level details of the underlying data. The connection betweeen data and ontology is performed via mappings. The combination of ontology and mappings allows to automatically translate queries posed over the ontology into data-level queries that can be executed by the specific underlying database management system. Ontop[2], Mastro[3], Stardog[4] are among the most popular OBDA systems.

2.2 Modeling the Insurance Sector

The necessity for big data manipulation and analysis is only currently emerging in the insurance industry, possibly due to the tight margin profits and increasing competition [8]. A starting point for the semantic formalization of the business functions and associated processes of the Insurance sector can be offered by the Property and Casualty Data Model [5], developed by the Object Management Group [16]. Organizations that incorporate the aforementioned data model into their procedures can be substantially benefited. P&C data model can be applied in a wide variety of warehousing, business intelligence, and data migration projects. In addition, this model can be leveraged to improve the use of data and to support data governance within the enterprise.

The major components of the P&C data model are the entities, attributes and relationships. An *Entity* can stand for a person, organization, place, thing, or concept of interest to the enterprise by covering a very broad and varied set of instance data, or something very specific. This idiom is referred to as *levels of abstraction*.

A *Relationship* should be a verb or a verb phrase that is always established between two entities. A relationship is used in order to form a sentence between its two entities. Moreover, the type of relationship can be 'Identifying', 'Non-identifying' or 'Subtype'.

---

1 http://franz.com/about/press_room/ag-6.0_2-8-2016.lhtml

2 http://ontop.inf.unibz.it/

3 http://www.dis.uniroma1.it/~mastro/?q=node/2

4 http://stardog.com/

Attributes are usually defined within an entity and are considered as properties or descriptors for this entity. An attribute is meaningless by itself. Every attribute in the data model is connected to a domain that provides for consistent names, data types, lengths, value sets, and validity rules. The main elements in order to define an attribute for an entity are the Attribute Name, Entity Name and Data Type.

Figure 1 illustrates how several entities, such as Insurable Object, Vehicle, Claim and Claim Folder, can be specified in terms of attributes and how they can be linked to each other via relationships in the context of the P&C data model.

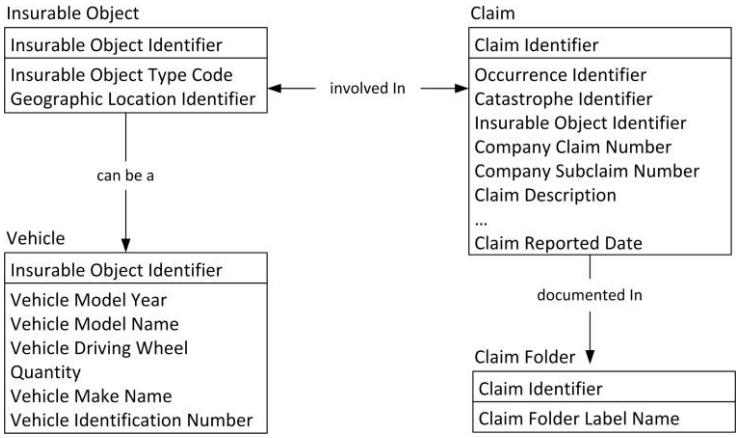

Fig. 1. A snippet of the P&C data model.

3 Insurance Ontology

In this section, we describe briefly how an ontology, fully expressed in the Web Ontology Language (OWL) [2], can be designed by considering the Property and Casualty model. We present how data entities representing most of P&C insurance business processes can be converted into ontology components. Taking into account the InfoSphere Data Architect guidelines [14], we can simply correlate logical data model's data types to OWL data types, logical data model's entities to ontology elements as well as logical data model's relationships to OWL object properties. In detail, Table 1 gathers all the restrictions that must be followed for an effective conversion of these relationships.

Table 1. Logical data model to OWL object mappings.

Logical Data model	**OWL ontology elements**
Entity	**owl:Class**
Entity - Name	rdf:about
Entity - Label	rdfs:label
Entity - Namespace	Namespace of owl:Class
Entity - Definition	rdfs:comment
Entity - Primary Key	owl:hasKey

Relationship	**owl:ObjectProperty**
Relationship – Name	rdf:about
Relationship – Label	rdfs:label
Relationship – Owner	rdfs:domain
Relationship – Namespace	Namespace of owl:ObjectProperty
Relationship – Child Entity	rdfs:domain
Relationship – Parent Entity	rdfs:range
Relationship – Annotation	rdfs:seeAlso, rdfs:DefinedBy, owl:FunctinalProperty,owl:InverseFunctionalProperty owl:AnnotationProperty
Attribute	**owl:DatatypeProperty**
Attribute – Name	rdf:about
Attribute – Label	rdfs:label
Attribute – Namespace	rdfs:domain
Attribute – Data Type	rdfs:range
Attribute – Length/Precision	xsd:maxLength, xsd:length, xsd:totalDigits
Attribute – Scale	xsd:fractionDigits
Attribute – Primary Key	owl:hasKey
Attribute – Documentation	rdfs:comment
Attribure – Annotation	rdfs:seeAlso, rdfs:isDefinedBy, owl:AnnotationProperty

Figure 2 depicts part of the neighborhood of the *InsurableObject, Claim, Agreement, Person* and other major entities of the model, presented as classes in the resulting ontology, based on the class hierarchy and property relations that may associate instances of one class with another.

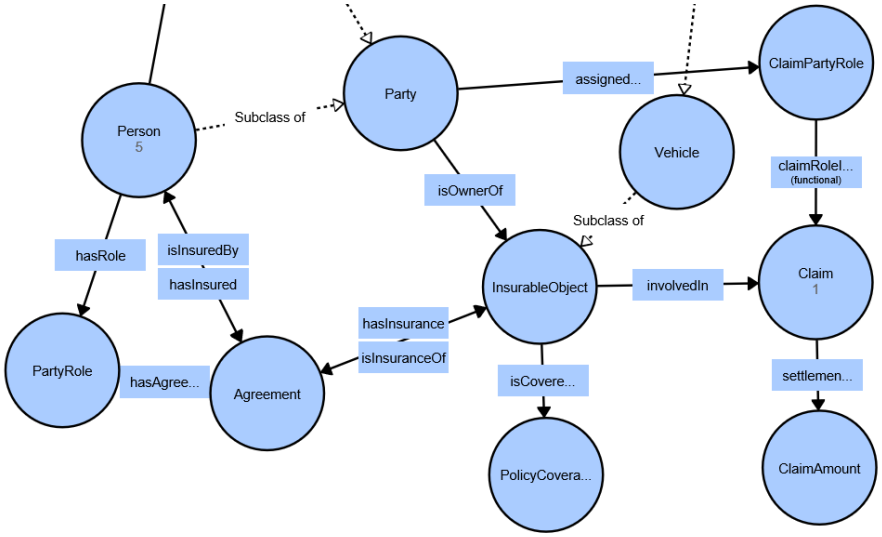

Fig. 2. The mapping of *InsurableObject* and other entities in the P&C ontology.

4 Architecture

4.1 Data Manipulation Strategy

In our experiments, we considered large volumes of offline data originating from the data archives of a well-known vehicle insurance company. Data were actually SQL dumps into CSV files, each corresponding to a seperate relational table. Each of the 5 tables includes about 0.5M tuples containing between 16-56 columns, resulting in 31 columns on average. For a trivial mapping, a single column of each tuple can form a seperate RDF triple, therefore the resulting graph would contain about 77M triples.

To facilitate acess to these data and to enable intelligent queries on top of them, we use an OBDA system, namely Ontop [4]. Ontop was selected based on its ease of use, intuitive mapping support, and high performance capabilities. In addition, Ontop supports reasoning at the OWL 2 - QL level, which is a lightweight reasoning profile, but expressive enough to support inferences on very large volumes of instance data [7].

In systems such as Ontop, there are two main elements, an *ontology* which models the application domain by using a shared vocabulary, and the *mappings,* which relate the ontological terms with the schemata of the underlying data sources. Therefore, the ontology and mappings combined, expose a setting, so that end-users can formulate queries without deeper understanding of the data sources, the relation between them, or the encoding of the data.

The outcome of the mappings in conjuction with the defined ontology can be thought as an RDF view, which can be materialised or not. In the materialization case, the data are triplified and then are directly used within an RDF triplestore without requiring additional interaction with the initial data sources. In the second case, which is the one followed by our approach, the RDF view is called a *virtual graph* and can be queried at query execution time, therefore allowing for on-the-fly ontology based access to and inference on data. At the same time, this approach relieves from the burden of data replication.

Table 2. A partial set of mappings used for the semantic modeling of insurance data.

ID	Source (SQL Query)	Target (Triples Template)
1	SELECT plate, contract FROM InsuredItemVehicle	:{plate} a :Vehicle ; :hasInsurance :{contract} . :{contract} a :Policy .
2	SELECT customerCode as c, iv.plate FROM InsuredItemCustomer as ic, InsuredItemVehicle as iv WHERE ic.contract = iv.contract;	:{c} a :Person ; :isOwnerOf :{plate}.
3	SELECT policyNumer as pol, ClaimNumber as r, totalPayAmount as amnt, iv.contract, plate FROM Claims, InsuredItemVehicle as iv where pol = iv.contract ;	:Amount_{pol}_{r} :hasAmount {amnt}^^xsd:decimal . :Claim_{pol}_{r} a :Claim ; :settlementResultsIn :Amount_{pol}_{r} . :{plate} :involvedIn :Claim_{pol}_{r}

In our setting, the ontology is offered by our implementation of the P&C model into OWL, described in the previous section. To construct the mappings, we used the mapping tool integrated within the ontop Protégé plugin. Some of the mappings

designed to create the virtualized graph are shown in Table 2 in the form of *ID*, *Source* and *Target*.

4.2 Infrastructure

After the ontology definition and the design of mappings, the next step is to set up how raw data are consumed. Ontop can be configured to access DBMS data or other sources, through the Teiid data virtualization platform[5]. For our purposes, we interfaced the data dumps through a JDBC driver, by means of a H2/MySQL database. Given the appropriate mappings, it is also possible to include data flows from other sources as well, such as federated databases, insurance brokes, agents filling in claims, vehicle sensons and other possibly unstructured data, in a similar manner.

Having established the connection to data, queries can then be performed using SPARQL. To this purpose, Ontop exposes a SPARQL interface via the Protégé plugin. For query evaluation, the system parses the query and communicates with its internal reasoner module, which has already loaded the ontology, in order to infer any implicit semantic implications. Next, the original query is internally transformed into one or more SQL queries addressed to the federation of data sources, by considering the mappings already defined and the outcomes of the reasoning proccess.

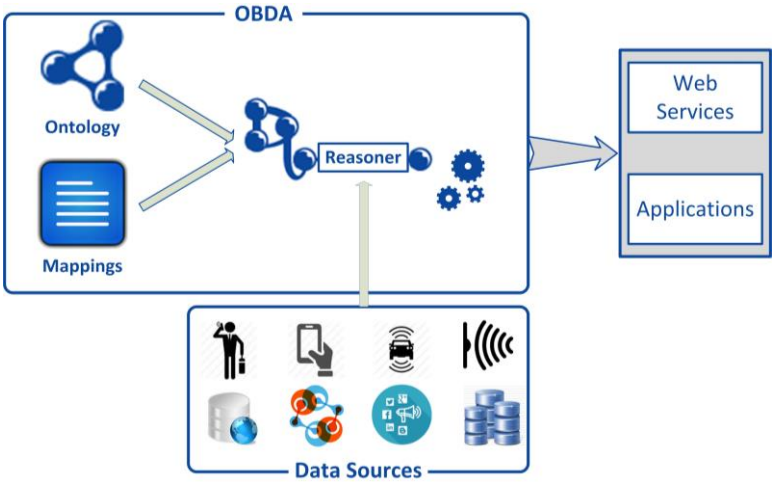

Fig. 3. Architecture of the experimental setup for data flow and query answering.

As a result, query answering and reasoning take place on-the-fly, without the need to replicate data into ontology instances first or materialize the graph resulting from the mappings. This also means that online updates to data are directly reflected into the answers received by SPARQL queries, as all evaluation occurs during query execution time. Ontop can also be used as a standard SPARQL HTTP endpoint by extending the Sesame Workbench, a web application for administering RDF repositories [3]. The data flow and the query evaluation process are depicted in Figure 3.

[5] http://teiid.jboss.org/

5 Intelligent Queries on Insurance Data

A set of example queries over the insurance dataset is presented in this section. All three queries involve some form of reasoning, so as to demonstrate the added-value ontology-based inferences can have on insurance data. They are also indicative of the expressivity of logical axioms allowed in OWL 2 - QL, like inheritance, hierarchy, inverse property and existential restriction.

In the first query, shown in Fig. 4, even though we have not designed a mapping rule defining instances of the *InsurableObject* classes, the instances of *Vehicle* are automatically classified as such, due to *Vehicle* being defined as subclass of *InsurableObject* in the ontology (see Fig. 2).

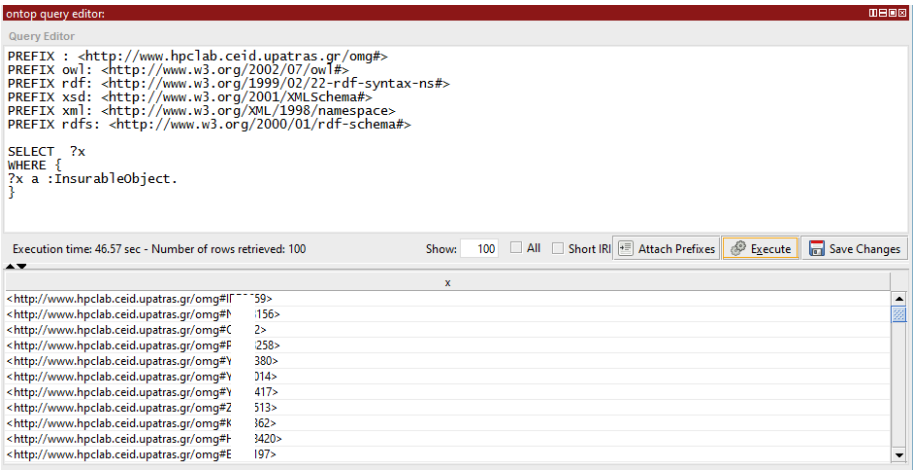

Fig. 4. SPARQL Query – Example 1.

The next example, shown in Fig. 5, retrieves the insurance policies for all vehicles that are owned by a specific *Party*. Note that we are able to use the *isInsuranceOf* property in the query, instead of *hasInsurance* as specified by mapping #1 in Table 2, because they are defined as inverses in the ontology (see Fig. 2).

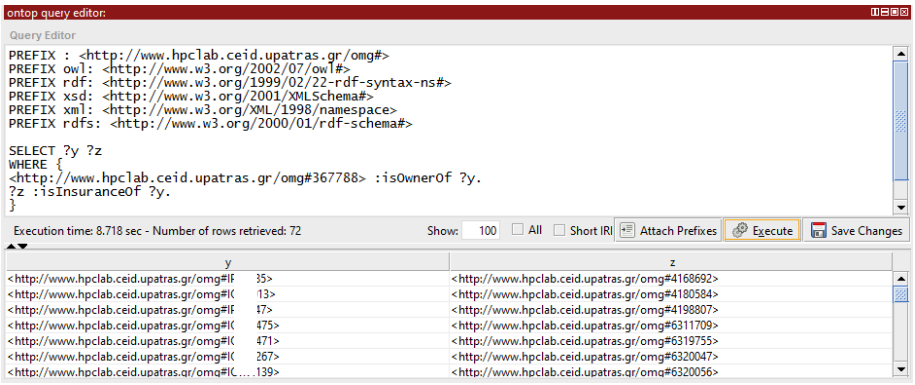

Fig. 5. SPARQL Query – Example 2.

With mapping #3, we relate a *Vehicle* to a *Claim* and the *Claim* to its resulting settlement *ClaimAmount*. The following query (Fig. 6) discovers vehicles *involvedIn Claims* that have already been settled with a *ClaimAmount* through *settlementResultsIn*. This is possible using the auxiliary class *AlreadySettled*, specified as an existential restriction on the *settlementResultsIn* property.

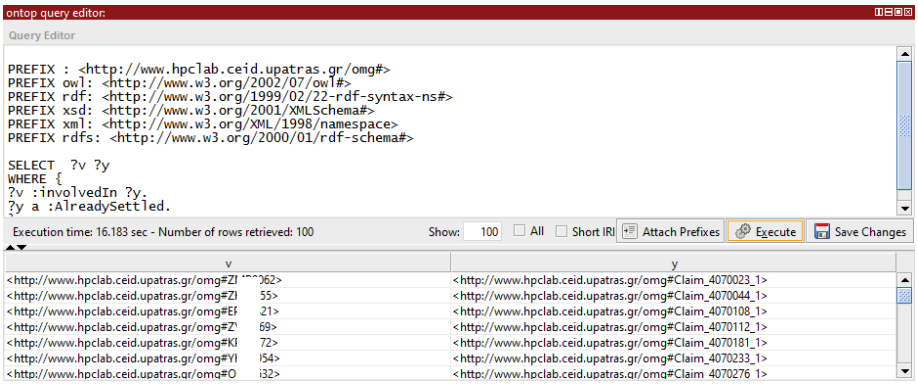

Fig. 6. SPARQL Query – Example 3.

6 Conclusions and Future Work

Nowadays, the data ecosystem of a typical insurance enterprise is significantly expanded by including not only internal databases but also third-party data sources such as social media, smartphones, sensors and other consumer and industrial devices. To this data heap, the public-sector data, recently available in the form of "Open Data" [12], can be appended. Even though such proliferation and variety of data can considerably profit the insurance sector by providing additional inputs to alleviate fraud and risk management, the obstacle of data gathering from different access points in different formats, as well as its semantic analytics, still remains.

In this work, we have exploited the capabilities of OBDA systems so as to highlight the contribution of semantic technologies in the industry of vehicle insurance. Having at our disposal a voluminous dataset from an existing insurance company, we set an infrastructure to inference knowledge in an efficient manner. The key component is our proposed P&C ontology and a set of logical axioms and mappings that correlate the data with the ontology. We have shown that the ontology can be utilized as an intermediate level for accessing directly legacy, existing databases and perform reasoning-enabled queries on them.

While graph materialization appears to perform well with Ontop, there is some delay when importing raw data into the relational schema. This may prove a bottleneck further on, so it might be worth investigating further concurrent data flow approaches for RDF virtualization and SPARQL querying, like C-SPARQL [1].

References

1. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E, Grossniklaus, M.: Querying RDF streams with C-SPARQL. SIGMOD Rec. 39(1), 20-26, (2010)
2. Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., Mc Guinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference, W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-ref-20040210/
3. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A generic architecture for storing and querying RDF and RDF schema. In: Proc. of the 1st Int. Semantic Web Conf. (ISWC). Lecture Notes in Computer Science, Vol. 2342, pp. 54–68. Springer (2002)
4. Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M., Xiao, G.: Ontop: Answering sparql queries over relational databases. Semantic Web (in press)
5. Jenkins, W., Molnar, R., Wallman, B., Ford, T.: Property and Casualty Data Model Specification (2011)
6. Kalou, A.K, Koutsomitropoulos, D.A.: Linking Data in the Insurance Sector: A case study. In: Proc. of the 10th Int. Conference on Artificial Intelligence Applications and Innovations (AIAI 2014), Workshop on New Methods and Tools for Big Data (MT4BD 2014), pp. 320-319. Springer, (2014)
7. Lanti, D., Rezk, M., Xiao, G., Calvanese, D.: The NPD Benchmark: Reality Check for OBDA Systems. In: Proc. of the 18th International Conference on Extending Database Technology (EDBT), pp. 617-628 (2015)
8. Llull, E.: Big data analysis to transform insurance industry. Tecnical article, Financial Times (2016)
9. Marr, B.: How Big Data is changing insurance forever. Technical artcile, Forbes (2015)
10. Michel, F., Faron-Zucker, C., Montagnat, J.: A Mapping-based Method to Query MongoDB Documents with SPARQL. In: Proc. of the 27th International Conference on Database and Expert Systems Applications (DEXA 2016), Sep 2016, Porto, Portugal. (2016)
11. Mitchell, I., Wilson, M.: Linked Data: Connecting and exploiting big data. White paper. Fujitsu UK, (2012)
12. World Bank Group. Transport and ICT: Open Data for Sustainable Development. Technical report (2015)
13. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. Journal on Data Semantics, pp.133–173, (2008)
14. Soares, S.: IBM InfoSphere: A Platform for Big Data Governance and Process Data Governance. MC Press Online, LLC, February (2013)
15. Sodenkamp, M., Kozlovskiy, I., Staake, T.: Gaining IS Business Value through Big Data Analytics: A Case Study of the Energy Sector. In: Proc. of the Thirty Sixth International Conference on Information Systems (ICIS), Fort Worth, USA, pp. 13-16, (2015)
16. The Object Management Group (OMG). MDA Guide Version 1.0.1, (2003)
17. Tsai, C.W., Lai, C.F., Chao, H.C., Vasilakos, A.C.: Big data analytics: a survey. Journal of Big Data, Vol. 2, No.21, pp.1-32, (2015)
18. Ylijoki, O., Porras, J.: Perspectives to Definition of Big Data: A Mapping Study and Discussion. Journal of Innovation Management, Vol.4, No 1, pp. 69-91, (2016)
19. Lenzerini, M.: Ontology-based data management. In: Proc. of CIKM 2011, pp. 5–6, (2011)
20. Rodriguez-Muro, M., Calvanese, D.: Quest, an owl 2 ql reasoner for ontology-based data access. In: Proc. of the 9th Int. Workshop on OWL: Experiences and Directions (OWLED 2012), Vol. 849 of CEUR Electronic Workshop Proceedings (2012)
21. Laney, D.: 3D data management: Controlling data volume, velocity and variety. META Group Research Note 6, 70 (2001)
22. Press, G.: Top 10 hot big data technologies. Technical article. Forbes (2016)