Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 96 (2016) 446 – 455

Procedia Computer Science

20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016, 5-7 September 2016, York, United Kingdom

Big Data Integration: a MongoDB Database and Modular Ontologies based Approach

Hanen Abbes*, Faiez Gargouri

MIRACL Laboratory, Higher Institute of Computer Science and Multimedia, BP 1030, Sfax University, Sfax, Tunisia

Abstract

Big Data are collections of data sets so large and complex to process using classical database management tools. Their main characteristics are volume, variety and velocity. Big Data integration is a new research area that faces new challenges due to these characteristics. Ontologies represent knowledge as a formal description of a domain of interest. They are widely used in data integration. This paper illustrates an approach for ontology based Big Data integration taking into account their characteristics. Our approach is based on a NOSQL database namely MongoDB and modular ontologies. It follows three steps: wrapping data sources to MongoDB databases, generating local ontologies, composing the local ontologies to get a global one. A tool implementing the generation of the local ontologies is also detailed.

Peer-review under responsibility of KES International

*Keywords:*Big Data; NOSQL; MongoDB; Ontology learning; Modular ontologies;

1. Introduction

In the last decade, we face an unprecedented number of data sources and an amount of available data increasing continuously. Traditional data management systems are undoubtedly incapable to cope with these data since volumes reach the threshold of petabytes. This phenomenon is called Big Data.

* Corresponding author. Tel.: +216 23 649 645.
*E-mail address:*abbes.hanen@gmail.com

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2016.08.099

Big Data emphasize heterogeneity among systems which creates problems with interoperability and integration of different information systems due to the volume, variety, and velocity dimensions of Big Data.

Ontologies were extensively used in data integration systems because they provide an explicit and machine-understandable conceptualization of a domain. They are important particularly in data semantics. But with the advent of Big Data, their implementation faces new challenges due to the characteristics of volume, variety and velocity.

In this paper, we propose a data integration approach where the target schema is represented as an OWL ontology and the sources correspond to Big Data. The main difficulty in integrating several data sources in this context concerns the constraints posed by the volume, variety and velocity of these data. Our approach is based on NOSQL databases namely MongoDB and modular ontologies. We equally emphasize on the implementation of the main step of our approach which is ontology learning from MongoDB database.

This paper is outlined as follows. In the next section, we expose our research context, we present the motivating scenario and goals of this work and discuss related work. Our approach for Big Data integration and a tool implementing ontology learning from MongoDB are detailed in the third section. Finally, the fourth section draws conclusions and suggests further research.

2. Background

2.1. Research context

The term Big Data refers to a terminology widely used nowadays to designate very big amounts of data. Big Data may be defined as "the amounts of data just beyond technology's capability to store, manage and process efficiently"[1]. According to Gupta and colleagues[2], Big Data can be defined as data that exceed the processing capacity of conventional database systems. This implies that the data count is too large, and/or data values change too fast, and/or it does not follow the rules of conventional database management systems. Big Data can be characterized along three important dimensions, namely volume, variety and velocity[3, 4] known as 3Vs.

- Volume: means that the quantities of data are larger than those conventional relational database infrastructures can cope with. Data are spread in large volumes ranging from gigabytes to terabytes, petabytes, and even more,
- Variety: refers to the number of sources and types of both structured and unstructured data, data are rarely presented in a perfectly ordered form and are rarely ready for processing,
- Velocity: refers to the speed at which the data are generated. Data are always generated in an unprecedented speed and must be dealt with in a timely manner.

Big Data challenges cover not only the storage and management of a variety of data, but also the extraction of consistent knowledge from such data.

Despite the insurmountable growth of data in Big Data scenarios, users usually look for a unified view of the data available from heterogeneous data sources. Consequently, integration issues are attracting more and more attention.

Data integration[5] is concerned with unifying data that share some common semantics but originate from unrelated sources. It refers to combining data in a way that a uniform view is available to users. Working on data integration, heterogeneity must be dealt with. Heterogeneity creates interoperability problem when distributed systems need to cooperate. In order to solve this problem, both structural and semantic heterogeneity have to be dealt with[6]. The structural heterogeneity takes place when data and concepts are stored using different structural relationships, whereas the semantic heterogeneity concerns the contents and the foreseen meaning of an information item[7].

These problems are accentuated with the advent of the Big Data phenomenon since we deal with data that exhibit the inherent characteristics of big volume, high generation velocity, and emerging with a variety of formats from various data sources.

Ontologies provide a solution to heterogeneity problems. They were widely used in data integration systems since they grant an explicit and machine-understandable conceptualization of a domain. They furnish a semantic model of the data sets under integration. According to Bontcheva and colleagues[8], an ontology is a specification of a shared conceptualization of a domain. Ontologies provide a common vocabulary on a specific domain and define terms meaning at various levels of formalization and the relationships between them[9]. The main purpose of ontologies is to capture knowledge about a specific domain and to provide common accepted representation for re-use and share.

Some of the existing ontologies have been generated manually. But, this process is very time consuming and prone to errors. It also raises problems in maintaining and updating ontologies. Thus, researchers are trying to propose methods to generate ontologies in more efficient ways. Ontology learning is then emerging as a field of ontology engineering to propose a solution to this issue. The main goal of ontology learning methods is to derive automatically an ontology from existing data[10]. It aims at facilitating the construction of ontologies to the ontology engineer[11]. Manual construction of ontologies in Big Data contexts is not easy at all since data are voluminous, of a wide variety, and of a wide velocity. So, it is important to set up ontology learning methods adapted to the specificities of Big Data.

In the next section sub-section, we concentrate on the basic motivations and goals of this work.

2.2. Motivating scenario and goals

We suppose that the managers of two big organizations, deploying Big Data architectures, want to merge their activities and tend towards operating centrally. Data integration and interoperability will be their main focuses since the two organizations may have different data management techniques before the merger and the data exchange involved is enormous. Data integration plays a key role in determining the efficiency of the resulting organization, either at the level of background systems integration or at processes integration, administrative tasks, and databases[5]. The complexity of data integration and interoperability concerns data storage and structure and the ways allowing the data to be integrated and operated as a single entity[12].

Data interoperability becomes a necessity for applications needing to contest in a Big Data environment. This challenge is emphasized by the heterogeneity of data they use. Interoperability in the context of Big Data enables to share information between individuals, providers and organizations so that systems and applications can exchange and use Big Data information without any special effort. This seems to be easy, but it becomes harder to properly implement since interoperability requires tightly controlled basics. Indeed, Big Data result in the proliferation of data from many sources. Collecting such data accentuates data growth, but connecting these data to access and handle is more difficult.

Ontologies were always proposed as a solution to the interoperability and data integration problems even before the advent of Big Data. According to Hashemi and Schneider[13], ontologies bring different contributions to Big Data. They help people to better understand the complexity of big engineered systems and enable integration among systems and data through semantic interoperability. Moreover, they improve models adaptability and reuse, reduce development and operational costs, enhance decision support systems, and help in knowledge management and discovery. Ontologies are important in Big Data scenarios since they provide cross-cutting meanings for terms. They make data (unstructured, semi-structured and structured) understandable to both humans and machines across sources. Thus, they provide a solution to semantic heterogeneity between data.

We aim at building ontology for Big Data integration while considering the characteristics of Big Data namely volume, variety and velocity.

2.3. Related work

We situate our research in the area of Big Data integration. We focus on ontology based NOSQL databases integration approaches and concentrate on recent solutions addressing ontologies for Big Data integration.

In[14], authors propose a data integration framework where the target schema is represented as an OWL ontology and the sources correspond to NOSQL databases namely MongoDB and Cassandra. The first step consists of generating a local schema for each integrated source using an inductive approach. It uses non-standard description logic reasoning services like Most Specific Concept (MSC)[15] and Least Concept Subsumer (LCS)[16] in order to generate a concept for a group of similar individuals and to define hierarchies for these concepts. The second contribution enables the specification of a global ontology based on the local ontologies generated for each data source. This global ontology results from the correspondences discovered between concept definitions present in each local ontology. For the source ontology generation, authors use containers, i.e. collections and column families in respectively document and column family databases to deduce a schema. The proposed approach considers that each container defines a DL concept and that each key label corresponds to a DL property that can either be a dataTypeProperty or an objectProperty and whose domain is the DL concept corresponding to its container. These concepts can be organized into a hierarchy of concepts using methods of Formal Concept Analysis (FCA)[17]. Finally an incremental schema generation approach is implemented. That is, each time a tuple is inserted or modified, the system checks if some labels are being introduced or deleted into the schema. This approach imposes that each update operation goes through this process. At the end of this step, a schema for each NOSQL source is created. To discover alignments between ontologies and to build the global ontology, authors follow several steps. The first step is to enrich the two ontologies to be aligned using the IDDL reasoner[18] to add subsumption relationships which are implicit in ontologies. The second step detects the simple correspondences using three classical alignment processes. The last step detects the complex correspondences.

Authors in[19] propose an approach for Big Data integration based on semantic heterogeneity reduction for Big Data in the domain of industrial automation. They deal with structural heterogeneity and semantic heterogeneity. To address the structural heterogeneity, they take into consideration different types of data sources such as text files, XML files and databases. To address the semantic heterogeneity, they create a shared ontology which ensures transformation of the data sources into the same “language”. The first step in this approach is the semi-manual creation of a shared ontology which ensures knowledge sharing. Authors first deal with structural heterogeneity. This problem falls into the preprocessing step. Data source processing strategies differ depending on its category. Next action is the construction of a shared ontology from the pre-processed data. A crucial step is to understand a given content and to identify the correspondent entities across all data sources. Some ontology matching systems are exploited for this task. Authors adopt a previously developed system MAPSOM[20] which focuses on user involvement in ontology matching. The other supporting tool for the shared ontology construction is Formal Concept Analysis (FCA) which is according to Obitko and colleagues[17] a theory of data analysis to support and simplify the ontology design process that identifies conceptual structures among data sets. After shared ontology construction, authors propose a transformation of data for a subsequent utilization. Two possible ways are described: data source transformation “on the fly” as well as the creation of a “snapshot” (a shared storage).

In[21], authors propose a semantic Extract-Transform-Load (ETL) framework for Big Data integration. The proposed framework generates a semantic model of the datasets under integration, and then generates semantic linked data in compliance with the data model. The use of semantic technologies is introduced in the Transform phase of an ETL process to create a semantic data model and generate semantic linked data (RDF triples) to be stored in a data mart or a data warehouse. The Transform phase still continue to perform other activities such as normalizing and cleansing of data. The Transform phase involves a manual process of analyzing the datasets, the schema and their purpose. Based on the findings, the schema has to be mapped to an existing domain specific ontology or an ontology will have to be created from scratch. If the data sources belong to disparate domains, multiple ontologies are required and alignment rules are specified for any common or related data fields. Extract and Load phases of the ETL process would remain the same. A case study was conducted to assess the effectiveness of the proposed semantic ETL framework using public datasets on educational data from Massive Open Online Courses (MOOC), National Household travel survey data and EPA’s Fuel Economy data.

The following table summarizes these works.

Table 1. Comparison between Big data integration approaches.

Approach	Input	Process	Output	Automation degree	Particularity
14	NOSQL databases namely MongoDB and Cassandra	-Local schema generation for each integrated database -Global ontology generation	OWL ontology	Automatic	-Use of containers to deduce a schema -Use of FCA to deduce concepts hierarchy
19	Big Data	-Data pre-processing -Shared ontology creation -Data transformation	-Shared ontology -shared	Semi manual	Heterogeneity reduction for batch and real time processing

			storage		
21	Big Data	-Semantic model generation of the data sets under integration -Semantic linked data generation in compliance with the data model	A data mart or a data warehouse that serves Big Data	Manual	Action on the Transform phase of the ETL process

3. Ontology building for Big Data integration

3.1. Our approach

Despite ontology learning is not a new research area, Big Data face it to new challenges due to their characteristics (velocity, variety, and volume). Thus, our approach to learn ontology from Big Data necessitates to take into account characteristics of Big Data.

Our departure corpus is formed by Big Data, whereas our target schema is OWL[†] (Ontology Web Language) ontology.

We consider that Big Data are huge amounts of data that are heterogeneous and swift and which are confronted to two kinds of evolution affecting data sources. Data sources in the context of Big Data are subject to an horizontal evolution that concerns the number of data sources which is always increasing exponentially, and a vertical evolution that concerns the quantity of data in each data source which is also rapidly growing. Data sources evolution in a Big Data context suppose that every source evolves independently from the other ones. This evolution concerns the data as well as the schema of the data source. These two kinds of data sources evolution are shown in the following figure (Fig. 1).

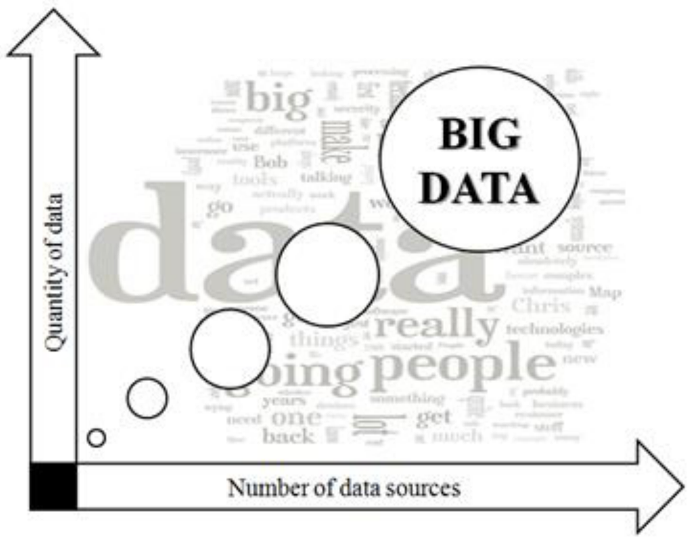

Fig. 1. Data sources evolution.

Thus, we face Big Data phenomenon when we deal with data from many data sources having different formats, each source contains a very big amount of data and grows and evolves independently from the others.

† http://www.w3.org/TR/owl-features/

Our choice is oriented to OWL as the ontology representation language since it is the standard recommended by the World Wide Web Consortium (W3C)[‡] to represent ontologies. We are interested particularly to OWL-DL since it supports the maximum expressiveness while retaining computational completeness and decidability.

Our ontology learning process is initiated from a corpus formed by autonomous and evolutionary data sources, in terms of data sources number and of data quantity in each source. To conduct the process, we need to reduce data speed (velocity), to homogenize data (variety) and to reduce data size (volume).

In order to address the velocity dimension of Big Data, we propose to deal with each data source apart from the other ones especially that data sources have different scheme and evolve independently to each other. This distributed processing will reduce the velocity of data.

As Big Data are very heterogeneous, we propose to copy all data sources to a common representation while preserving the independence of the initial sources. This conversion has for role to partially reduce data complexity and heterogeneity. It falls into the pre-processing step. This task is the work of wrappers. Data source processing strategies differ depending on its category. The main goal of this migration is to homogenize data and to address the variety dimension of Big Data.

Since NOSQL databases[§] are used as the backend stores of Big Data, we propose to use them as a common representation to store data. They will play the role of an intermediary representation between Big Data and the ontology in order to address the volume dimension of Big Data and to reduce the data heterogeneity.

To recapitulate, our approach to learn ontology from Big Data is based on wrapping each data source to a NOSQL database, which will be transformed later into ontology.

NOSQL databases group together heterogeneous data. They permit to store large volumes of structured, semi-structured, and unstructured data. Moreover, They provide high speed access to the stored data and are very flexible.

There are four types of NOSQL databases namely key/value stores, column family stores, document oriented databases and graph databases. We are convinced that document oriented databases are the best ones to initiate ontology learning process for many reasons. They are very flexible and may handle very large amounts of structured and unstructured data. Moreover, they are schema less which reduces the complexity. Our choice is oriented particularly to MongoDB[**] as a document oriented database to which we will wrap all data sources for many reasons. First, it is the fastest-growing new database in the world that provides a rich document oriented structure with dynamic queries. Second, it allows to compartmentalize data into collections in order to divide data logically. Thus, the speed of queries can increase dramatically by querying on a subset of the data instead of all it. Collections are analogous to tables in a relational database. Each collection contains documents that can be nested in complex hierarchies but still easy to query and index. A document is seen as a set of fields, each one being a key-value pair. A key is a string and the associated value may be a basic type, an embedded document, or an array of values. MongoDB can manage data of any structure without expensive data warehouse loads, no matter how often it changes. Thus, we can cheap new functionality without redesigning the database.

After wrapping each data source to a corresponding MongoDB database, we generate an ontology corresponding to each data source by means of transformation rules from MongoDB to the OWL language[22]. Then, we merge resulting ontologies into a global one.

We chose a modular conceptualization since the beginning of the ontology development cycle. Modularity is an important technology for collaborative knowledge development environments. It is central to reduce the complexity of designing and understanding ontologies, and to facilitate ontology verification, reasoning, maintenance and integration. Ontology modularization aims at presenting to users ontologies (ontology modules) with the knowledge they need. This reduces the scope as much as possible to what is strictly necessary. In particular, ontology modules ensure the following advantages. First, they facilitate knowledge reuse across various applications. Second, they are easier to build, maintain, and replace. Third, they enable distributed engineering of ontology modules over different locations and different areas of expertise. Finally, they enable effective management and browsing of modules[23, 24,

‡ https://www.w3.org/
§ http://nosql-database.org/
** http://www.mongodb.org/

[25]. According to d'Aquin and colleagues[18], an ontology module can be considered as a loosely coupled and self-contained component of an ontology maintaining relationships to other ontology modules. Thereby, ontology modules are themselves ontologies that cover a restricted point of view of the modeled domain. In the scope of this work, an ontology module represents a point of view covered by a data source containing data about the modeled domain.

The methodology adopted to construct the final ontology follows a composition approach. The different learned modules will be composed to constitute the global ontology. The composition approach is based on similarities discovery between concepts of the ontological modules to be merged.

Our approach to learn ontology from Big Data may be summarized by the following figure (Fig. 2).

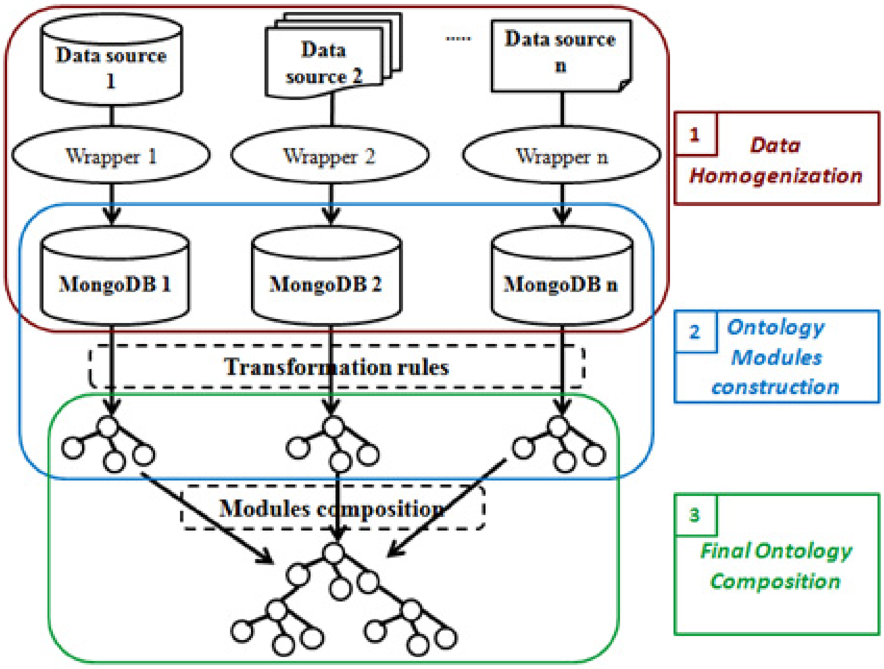

Fig. 2. Ontology based Big Data integration.

Our approach to learn ontology from Big Data facilitates global ontology updates and guarantees the autonomy and independence of data sources. We notice that Big Data are dynamic by nature due to the permanent arrival of new data, so they are subject to different updates. New sources of data may also appear. These updates must be sent up to the global ontology. Moreover, data about a domain may change in many different manners. On the one hand, new data may appear, and this leads to establish new concepts. These updates result on technological advances. On the other hand, some data may become obsolete, and so, some concepts must be removed from the global ontology. Besides, modeling a domain may necessitate concepts redefinition. New concepts that are more specific than the pre-existent ones can be defined if the domain needs to be more precise, or more abstract if we want to simplify the domain and facilitate its comprehension[27].

The autonomy of data sources is ensured by migrating all data sources to the same formalism. This has also the advantage to factorize and simplify the ontology building process. Indeed, we aim at preserving the normal functioning of initial data sources and at making the resulting ontology light and not heavy with instances which are stored in the MongoDB databases. Moreover, instead of defining transformation rules from each data source to the ontology formalism representation, we wrap all data sources to the same intermediary representation so that the process of ontology modules learning is the same for all the data sources.

Our approach takes into account all the Big Data characteristics, not only variety, but also volume and velocity.

Compared with the approaches discussed in Table 1, our approach takes as input Big Data sets and produces as output an OWL ontology. The process described may be summarized into three main steps: data homogenization step, local ontologies generation and finally, global ontology composition. The process is fully automatic and has the particularity to involve an homogenization step to facilitate ontology generation.

3.2. Experiments

We focus on the task of ontology modules learning from MongoDB databases. To do this, we defined transformation rules to map MongoDB constructs into OWL ontology[22]. This work is based on five main steps. The first step is the creation of the ontology skeleton. It consists of defining ontology classes and detecting subsumption relationships between them. The second step is to learn concepts properties (objectProperties and dataTypeProperties). Individuals are identified in the third step. In the fourth step, we deduce class axioms (equivalence and disjoining), property axioms (inverseOf) and constraints (cardinality constraints, value constraints). Finally, we enrich the ontology with classes definition operators (union, intersection, complementarity).

Classes of the ontology are extracted from collections, subsumption relationships are extracted from the field "parent" in every document, dataTypeProperties are extracted from basic fields in the database documents, objectProperties are extracted either from embedded documents or from references with DBRef constructs in the database documents and individuals are extracted from the values of fields in the documents.

We implemented a tool (M2Onto: MongoDB to ontology) to perform these tasks. This work is directed by means of the JAVA programming language. To access MongoDB database, we used the mongo-java-driver-2.13.0 and to perform the ontology creation we used the OWL-API version 4.0.1. Figure 3 shows the main interface of our tool.

Fig.3. The M2Onto tool.

The M2Onto tool loads an existing database from the hard disk, executes transformation rules and generates the corresponding OWL ontology. It provides an OWL file as output as well as a graphical representation of the resulting ontology. The evaluation of the M2Onto tool was performed by checking the consistency of the generated ontology. To do this, we integrated the Pellet reasoner to the M2Onto tool. We tested our tool by means of the

”NorthWindMongo” database available from https ://github.com/raynaldmo/northwind-mongodb. We provide in the following figure (Fig. 4) the resulting ontology generated by our tool.

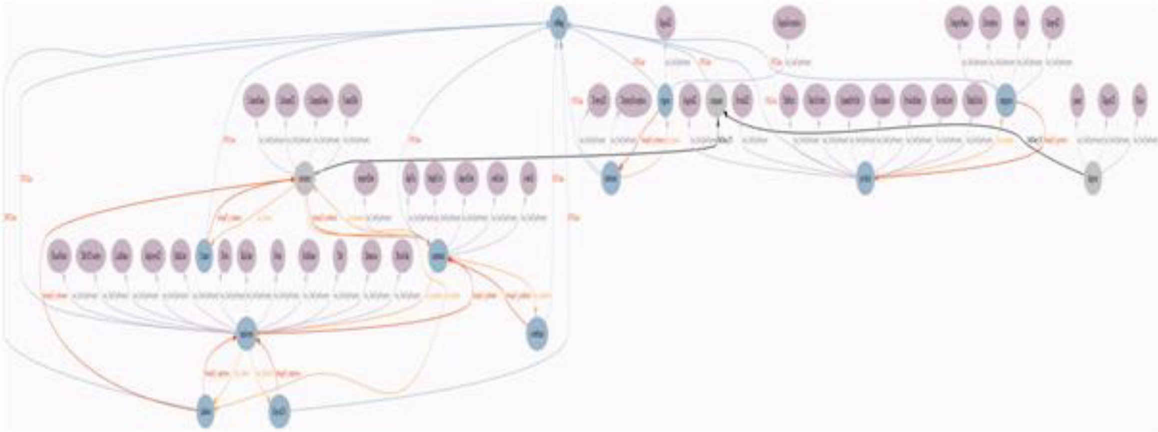

Fig. 4. Visualization of the “NorthWindMongo” ontology by means of the M2Onto tool.

Classes are represented by blue nodes. Sub-classes are represented by grey nodes. DataTypeProperties are represented by purple nodes. ObjectProperties are represented by red edges and their inverses by orange edges.

Due to visibility issues, we provide in the following figure an excerpt limited to only classes and objectProperties since the whole graph is very condensed.

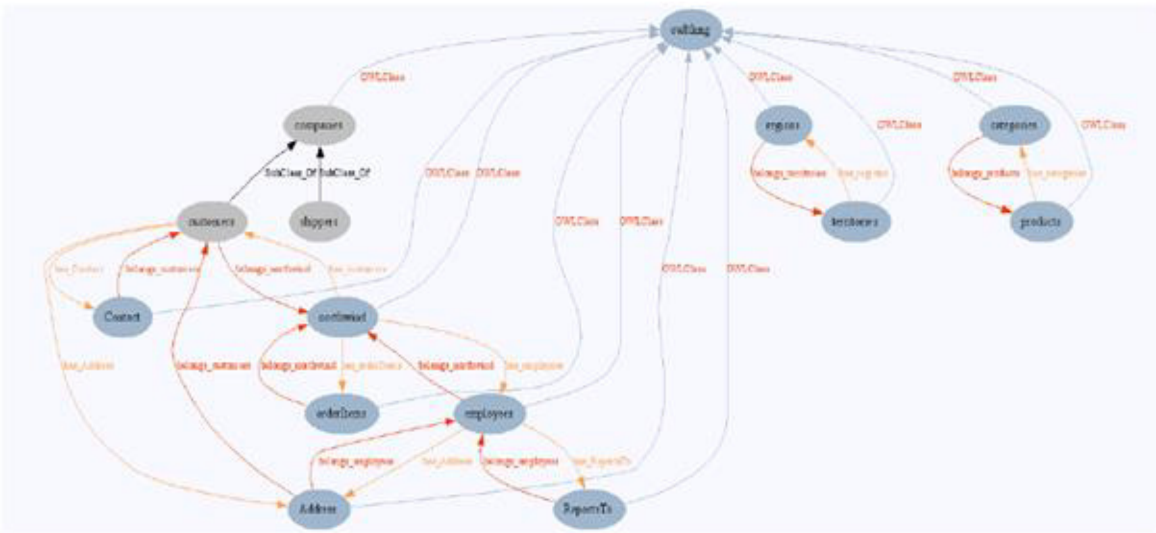

Fig. 5. Visualization of classes and objectProperties of the “NorthWindMongo” ontology by means of the M2Onto tool.

4. Conclusion and future works

Dealing with Big Data about a specific domain is an important issue worldwide. Finding solutions to ease Big Data integration is a difficult task. The basic issue in Big data integration is to automatically build the ontology model, and to bring out the hidden semantics which are not directly available from the data sources. The resulting ontology serves to represent knowledge integrated from Big Data sources and to provide a shared model for data sources.

This work concentrated on ontology building for Big Data integration. Our approach is based on NOSQL databases namely MongoDB and modular ontologies. We focused on the challenges that Big Data expose to data integration. We discussed state of art regarding Big Data integration and NOSQL databases integration. The intent is to develop a novel approach for data integration adapted to the characteristics of Big Data.

As future works, we envisage to formalize the step of ontology modules composition and to propose an approach to update the resulting ontology taking into account the data sources updates.

References

1. Kaisler S, Armour F, Espinosa JA, Money W. Big data: issues and challenges moving forward. 6th Hawaii International Conference on System Sciences (HICSS) 2013. IEEE; p. 995–1004.
2. Gupta R, Gupta H, Mohania M. Cloud Computing and Big Data Analytics: What Is New from Databases Perspective?. In: Srinivasa S, Bhatnagar V, editors. Big Data Analytics, Volume 7678 Lecture Notes in Computer Science. New York: Springer-Verlag; 2012. p. 42-61.
3. Zikopoulos P, Eaton C. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw–Hill/Osborne Media; 2011.
4. Boden C, Karnstedt M, Fernandez M, Markl V. Large-scale Social-media Analytics on Stratosphere. In: Proceedings of the 22nd International Conference on World Wide Web Companion; 2013. p. 257-260.
5. Lenzerini M. Data Integration: A Theoretical Perspective. In: Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. Madison, Wisconsin, USA; 2002. p. 233-246.
6. Malucelli A, Oliveira E. Ontology-Services to Facilitate Agents' Interoperability. In: Lee J, Barley M, editors. Proceedings of the Sixth Pacific Rim International Workshop on Multi-Agents (PRIMA). Springer-Verlag, LNAI 2891, Korea; 2003. p 170-181.
7. Wache H, Vögele T, Visser U, Stuckenschmidt H, Schuster G, Neumann H, Hübner S. Ontology-Based Integration of Information-A Survey of Existing Approaches. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), Workshop: Ontologies and Information Sharing, Seattle, USA, 2001.
8. Gruber TR. A translation approach to portable ontology specifications. Knowledge Acquisition. 5(2); 1993. p. 199-220.
9. Benjamin V, Perez AG. Knowledge system technology: ontology and problem solving methods. 1999.
10. Bontcheva K, Sabou M. Learning Ontologies from Software Artifacts: Exploring and Combining Multiple Sources. In Proceedings of the 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE); 2006.
11. Cimiano P, Mädche A, Staab S, Völker J. Ontology Learning. Handbook on Ontologies; 2009. p. 245-267.
12. Kadadi A, Agrawal R, Nyamful C, Atiq R. Challenges of data integration and interoperability in big data. IEEE International Conference on Big Data; 2014. p. 38-40.
13. Hashemi IA, Schneider T. Ontology Summit 2012 Communique - v1.01 Ontology for Big Systems; 2012.
14. Curé O, Lamolle M, Le Duc C. Ontology Based Data Integration Over Document and Column Family Oriented NOSQL. In The Computing Research Repository (Corr); 2013.
15. Baader F, Calvanese D, McGuiness DL, Nardi D, Patel-Schneider P. The Description Logic Handbook: Theory, Implementation, Applications", Cambridge University Press, Cambridge, UK; 2003.
16. Baader F, Sertkaya B, Turhan AY. Computing the least common subsumer w.r.t. a background terminology. In Journal of Applied Logic; 2004. p. 400-412, Springer.
17. Obitko M, Snasel V, Smid J. Ontology design with formal concept analysis. CLA. vol. 110; 2004.
18. Zimmermann A, Le Duc C. Reasoning with a network of aligned ontologies. In Proceedings of the 2nd International Conference on Web Reasoning and RuleSystems (ICWRRS); 2008. P. 43-57.
19. Jirkovský V, Obitko M. Semantic Heterogeneity Reduction for Big Data in Industrial Automation. In: Information Technologies - Applications and Theory ITAT; 2014.
20. Jirkovský V, Ichise R. Mapsom: User involvement in ontology matching. In Proceedings of the 3rd JIST Conference, Springer; 2013
21. Bansal SK, Kagemann S. Integrating Big Data: A Semantic Extract-Transform-Load Framework. Computer, vol.48, no. 3, 2015. pp. 42-50.
22. Abbes H, Boukettaya S, Gargouri F : Learning Ontology from Big Data through MongoDB Database. In: the 12th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA); 2015.
23. Jarrar M. Towards Methodological Principles For Ontology Engineering. Phd Thesis, Vrije Universiteit Brussel Faculty of science ; 2005.
24. Pathak J, Johnson TM, Chute CG. Survey of modular ontology techniques and their applications in the biomedical domain. In: Integrated computer-aided engineering, 16(3) ; 2009. p 225–242.
25. Djedidi R. Approche d'évolution d'ontologie guidée par des patrons de gestion de changement. Human-Computer Interaction. PhD thesis. Université Paris Sud - Paris XI ; 2009.
26. d'Aquin M, Haase P, Rudolph S, Euzenat J, Zimmermann A, Dzbor M, Iglesias M, Jacques Y, Caracciolo C, Aranda CB, Gomez JM. D1.1.3: Neon formalisms for modularization: Syntax, semantics, algebra. Deliverable 1.1.3 of the NeOn Integrated Project; 2008.
27. Abbes H, Gargouri F. Towards ontology building and updating from Big Data. In: Advances on Decisional Systems Conference (ASD); 2014. p 61-66.