An OWL Ontology for supporting Semantic Services in Big Data platforms

Domenico Redavid, Roberto Corizzo
CINI Big Data Laboratory
Consorzio Interuniversitario Nazionale per l'Informatica, CINI
Bari, Italy
Email: {domenico.redavid, roberto.corizzo}@consorzio-cini.it

Donato Malerba
Computer Science Department
University of Bari Aldo Moro
Bari, Italy
Email: donato.malerba@uniba.it

***Abstract*—In the last years, there was a growing interest in the use of Big Data models to support advanced data analysis functionalities. Many companies and organizations lack IT expertise and adequate budget to have benefits from them. In order to fill this gap, a model-based approach for *Big Data Analytics-as-a-service* (MBDAaaS) can be used. The proposed model, composed by *declarative*, *procedural* and *deployment* (sub)models, can be used to select a deployable set of services based on a set of user preferences shaping a Big Data Campaign (BDC). The deployment of a BDC requires that the selection of services has to be carried out on the basis of coherent and non conflictual user preferences. In this paper we propose an OWL ontology in order to solve this issue.**

***Keywords*-Big Data as a Service; Semantic Web Services; Big Data ontology;**

I. Introduction

The Model-based BDAaaS (MBDAaaS) proposed in the TOREADOR project[1], relies on the design of different types of models describing the entire BDA process and its artifacts [1] in order to support a non-expert customer/user during the deployment of a full pipeline that address their goals. For this purpose, the MBDAaaS relies on a *declarative model* that specifies the goals of a given analytic task in terms of pairs (*indicators*/*objectives*). However, the declarative model currently used in TOREADOR project lacks a formal semantics.

While the interest for BDAaaS increased, various technologies and approaches have been devised in the Semantic Web (SW) context in order to support automatic discovery, selection and composition of Web services. OWL for Services (OWL-S)[2] can exploit SW technologies and approaches in a straightforward manner because it is an OWL ontology[3] itself. However, OWL-S describe the services, e.g., the BDA services, through the use of other ontologies. Although a large number of platforms has been designed to provide BDA services, most of them exploit ontologies for the extraction of meaningful data from heterogeneous data and not for supporting / improving BDAaaS through OWL-S. From this aspect, the main contribution of this paper concerns the formalization of the TOREADOR declarative model as an OWL ontology.

[1]TrustwOrthy model-awaRE Analytics Data platform (TOREADOR) - www.toreador-project.eu

[2]OWL-S: Semantic Markup for Web Services W3C Member Submission 22 November 2004 - www.w3.org/Submission/OWL-S/

[3]OWL Working Group, OWL 2 Web Ontology Language: Document Overview. W3C Recommendation, 2009

II. Basics

In this section, we illustrate the basic notions concerning OWL inferences and OWL-S specs since they are exploited to enhance the MBDAaaS framework whose description is also given here.

A. OWL Inferences

Inference on the Semantic Web can be characterized by discovering new relationships. On the Semantic Web, data are modeled as a set of (named) relationships between resources. "Inference" means that automatic procedures can generate new relationships based on the data and based on some additional information in the form of a vocabulary, e.g., a set of rules. Some reasoning services that are used for making inference are: 1) Subsumption, which decides whether a concept is more general than the other one, 2) Satisfiability, which determines whether the input concept is not contradictory, and 3) Instance check, which decides whether a given assertion holds for all the interpretation of the knowledge base.

The OWL formal semantics of an ontology entails facts that are not literally present in the ontology by deriving its logical consequences which can be base on a single document or multiple distributed documents combined using defined OWL semantics. The inference, provided by OWL reasoners, can be schematize as follows:

It takes an Ontology and an Axiom as Input and returns as Output:

- Either:
  - True if the axiom holds for any interpretation
  - False if there is at least one interpretation falsifying the axiom
- Inconsistent ontology

Traditionally, the basic reasoning mechanism provided by Description Logic (DL) systems checked the subsumption of concepts. This, in fact, is sufficient to implement also the other inferences (see chapter 2 in [2]).

B. OWL for Services (OWL-S)

OWL-S is an ontology built upon the OWL language for describing Semantic Web Services. OWL-S provides a Semantic Web Services framework to formalize an abstract description of a service. In OWL-S, the *Service* class is used to describe the service and it is related with the following three other classes:

- *Service Profile.* It specifies the functionality of a service by means of several types of information: *Human-readable* information, *Functionalities*, *Service parameters*, *Service categories*.
- *Service Model.* It exposes to clients how to use the service by detailing the semantic content of requests, the conditions under which particular outcomes will occur, and, where necessary, the step-by-step processes leading to those outcomes. It defines the concept of *Process*, that describes the composition of one (*Atomic*) or more services (*Composite*) in terms of their constituent processes.
- *Service Grounding.* It describes, a communication protocol, a message format and other service-specific details, for instance those based on WSDL[4].

Each OWL-S process is based on an IOPR model. The *Inputs* represent the information that is required for the execution of the process. The *Outputs* represent the information that the process returns to the requester. *Preconditions* are conditions that are imposed over the *Inputs* of the process. Formally, *Input* and *Output* are subclasses of the class *Parameter* declared in its turn as a subclass of *Variable* in SWRL ontology[5]. Every parameter has a type (either an *OWL Class* or an *OWL Datatype*), specified using a URI. Such type is needed to refer it to an entity within the knowledge domain of the service.

For our aim, the OWL-S mechanism that gives semantic to the input and output parameters of a Web service can be summarized as follows:

- Each WSDL input parameter is associated with an input parameter name in OWL-S process model;
- Select an OWL ontology class as parameter type.

C. The MBDAaaS framework

The Declarative Model allows customers to formulate requests and defines a set of goals shaping a Big Data Campaign (BDC) by compiling a set of forms, while the Procedural Model allows to retrieve a set of services compatible with these goals. However there may be some incompatibilities among the various entities of the Declarative model that are involved in the Procedural model. These incompatibilities must to be handled for avoiding consistency problems in the request. For instance, suppose that the customer is interested to a particular type of analytics that requires the application of a supervised approach, but he has no availability of a labeled training set. In this case the MBDAaaS framework should be able to avoid the formulation of such a request. A possible solution can be the formalization of such incompatibilities of the declarative model through languages endowed with a formal semantics such as OWL. The ontology modeled via OWL can be also used to accomplish one of the main functions of the TOREADOR Procedural Model, that is, the OWL-S based retrieval service. In order to build useful OWL-S descriptions, it is necessary to formalize the declarative model describing the MBDAaaS knowledge domain with an OWL ontology.

III. The Ontology Model

In this section, we provide more details about the proposed Declarative model and then we report how the model can be formalized as an OWL ontology.

A. The Declarative Model

The proposed systematization [1] describes a Big Data process by different perspectives, often identifiable in the following five conceptual areas:

- *Data preparation*, all activities aimed to prepare data for analytics.
- *Data representation*, how data are represented and representation choices for each analysis process.
- *Data analytics*, the analytics to be computed.
- *Data processing*, how data are routed and parallelized.
- *Data visualization and reporting*, an abstract representation of how the results of analytics are organized for display and reporting.

A Declarative model is a set of entities used to specify a user goal (what the BDA should achieve) in terms of these conceptual areas. In details, they describe Functional Goals (FGs) expressing desired properties (from the user point of view) of a BDC. In other words, FGs express commitments on service properties made by ICT providers to their customers. A goal is measured by a Functional Indicator (FI), that is, a label expressing a way to assess the goal, and a Functional Objective (FO), that is, a threshold on a certain scale, either ordinal or metric, for the FI. Our general aim is to provide a method where Big Data FOs can be selected according to both internal and external constraints (e.g., incompatibilities among FIs and due to policies / regulations, resp.). Requirements have been defined in these five different conceptual areas using the Concept maps (CMAP) [6], a graphical tool for organizing and representing knowledge. They include concepts, usually enclosed in circles or boxes of some type, and relationships between concepts indicated by a connecting line linking two concepts. Labels above the lines, referred to as linking words or linking phrases, specify the relationship between the two concepts. Fig. 1 depicts

[4]Web Services Description Language (WSDL) - www.w3.org/TR/wsdl/

[5]Semantic Web Rule Language - www.w3.org/Submission/SWRL/

[6]Conceptual Maps, cmap.ihmc.us

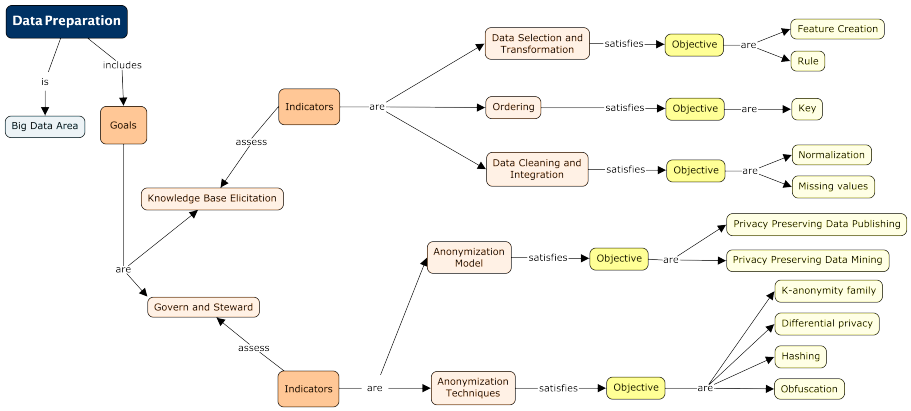


Figure 1: Part of the CMAP declarative model describing the Data Preparation area

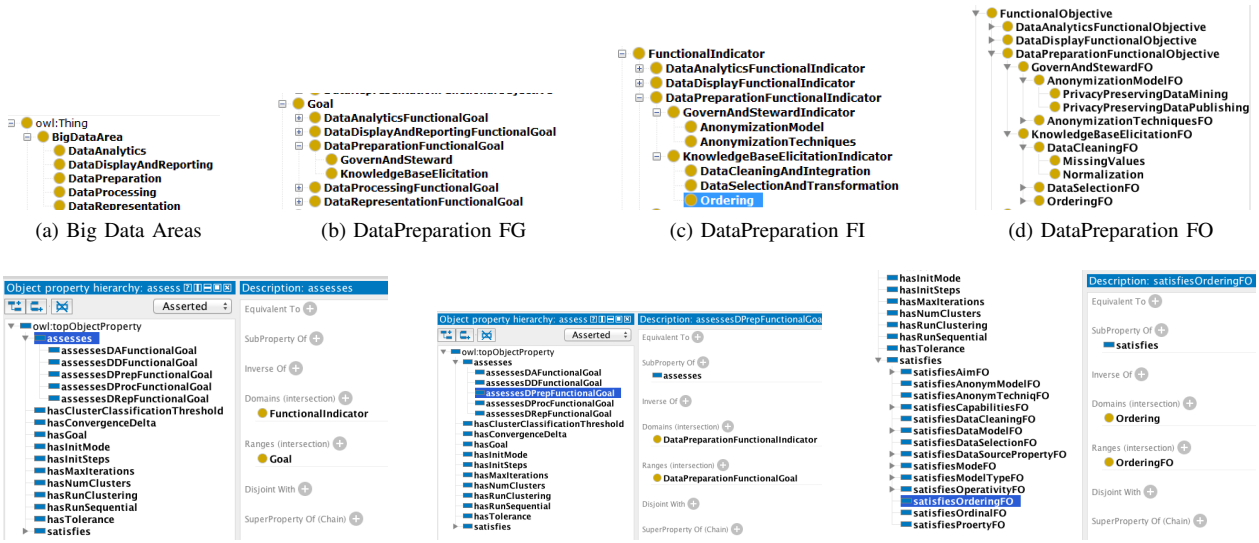
the CMAP declarative model describing the TOREADOR Data Preparation area. CMAP does not allow to serialize the modeled knowledge in a formal manner, i.e., understandable by machines, but can be considered a good starting point for its creation.

B. From the Declarative Model to the Ontology

In this section we illustrate how the declarative model has been formalized as an OWL ontology, named BDM Ontology[7]. For sake of space, we will describe the details of the models focusing only on one of the above areas, Data Preparation, the other ones can be modeled in the same way. Firstly, we need to specify which are the Big Data Areas of the declarative model. With the reference at Fig. 2a, this can be made by introducing one of the high-level concepts *bdmo:BigDataArea* and its children, whose name corresponds to the aforementioned areas. For doing this, some subsumption axioms (one per area) are added to the ontology, e.g. *bdmo:DataPreparation rdfs:subClassOf bdmo:BigDataArea*. The FGs can be modeled similarly to the way we modeled Big Data areas. Again, we need to introduce a high-level concept i.e. *bdmo:Goal* and its sub-hierarchy (see Fig.2b). In the case of the area describing Data Preparation, there are two kind of FGs: the Knowledge Base Elicitation and Govern And Steward FGs. Note that, according to the declarative model, the satisfaction of these goals is evaluated in terms of two disjoint sets of FIs: a FI cannot assess both the Knowledge Base Elicitation and Govern and Steward goal. Thus it is reasonable to assume that such indicators aims at evaluating two different goals which can be considered as disjoint. Therefore, in the ontology, we added a disjointness axiom between the concepts *bdmo:KnowledgeBaseElicitation* and *bdmo:GovernAndSteward*.

[7]Big Data Model Ontology (bdmo is used as namespace in the rest of the paper), https://www.consorzio-cini.it/ lab-bigdata/BDMOntology.owl

For avoiding cases where the same FG concerns two or more Big Data Area at the same time, we have defined the concept representing all the functional indicators related a specific Big Data Area (this is illustrated in Fig. 2c). In the case of Data Preparation area, we introduced in the ontology the concept *bdmo:DataPreparationFunctionalIndicator* subsumed by the concept *bdmo:FunctionalIndicator*. This is specialized through two concepts describing the FIs assessing the goals Knowledge Base Elicitation and Govern and Steward, namely *bdmo:KnowledgeBaseElicitationIndicator* and *bdmo:GovernAndStewardIndicator*. On one hand, the *bdmo:KnowledgeBaseElicitationIndicator* groups those techniques that aims at manipulating a dataset to improve the quality of the data for subsequent analysis, such as *data cleaning*, *ordering* and *data selection techniques*. The resulting concepts are called *bdmo:DataCleaningAndIntegration*, *bdmo:Ordering* and *bdmo:DataSelectionAndTransformation*. On the other hand, the concept *bdmo:GovernAndStewardIndicator* describes aspects related to anonymization techniques for encrypting or removing personal identifiable information from data sets. This is made introducing the concepts *bdmo:AnonymizationTechniques* and *bdmo:AnonymizationModels*. Similarly to the FIs, the FOs are grouped by the Big Data Area (see Fig. 2d), through the concept *bdmo:DataPreparationFO* subsumed by *bdmo:FunctionalObjective*. The FO objectives are further grouped by the FGs and then w.r.t. the FIs that are satisfied. Note that, as in the case of the FGs, also the FIs and the FOs must be declared as disjoint in order to prevent either cases of a FI assessing more FGs or a FIs satisfies more FOs. After that the elements of declarative model have been expressed as concepts, it is important to represent relationships between such concepts. The relationship between a FI and FG is expressed by a very general object property named *bdmo:assesses*, whose OWL



(a) Big Data Areas (b) DataPreparation FG (c) DataPreparation FI (d) DataPreparation FO

(e) access objectProperty (f) DataPreparation access object prop. (g) DataPreparation satisfiesOrdering

Figure 2: Fragments of the ontology derived from the declarative model

domain is *bdmo:FunctionalIndicator* and OWL range is *bdmo:FunctionalGoal* (see Fig.2e). For this property, hierarchies of sub-object properties are defined (one for each Big Data Area). For instance in the case of the Data Preparation area, the property *bdmo:assesses* has a sub-property named *bdmo:assessesDPrepFunctionalGoal* whose domain is *bdmo:DataPreparationFunctionalIndicator* and range is *bdmo:DataPreparationFunctionalGoal* (see Fig.2f). Similarly, the ontology models the relationship between the FOs and FI through the object property *bdmo:satisfies*. Again, this is made by defining various sub-properties e.g. considering the FI *bdmo:Ordering* (see Fig.2c), this concept is the domain for another object property, i.e. *bdmo:satisfiesOrderingFO* ranging in *bdmo:OrderingFO* that is a sub-concept of *bdmo:FunctionalObjective* (Fig.2g). Giving formal semantics to the declarative model has several advantages. Firstly, for a given Big data Area the ontology can be easily extended with new FGs / FIs / FOs by adding new OWL concepts and propeties. For instance, it is sufficient to add a sub-concept of *bdmo:DataPreparationFunctionalGoal* to define a new goal for Data Preparation area. Moreover, the ontology can be used to handle incompatibilities between entities spread across different areas. Considering the large number of entities, the use of the ontology in combination with a reasoner has the clear advantage to dynamically detect and handle possible incompatibilities, avoiding the need of writing from scratch the code to handle each incompatibility at the application level. In perspective, this will also allow to detect and handle incongruences between concepts belonging to *declarative*, *procedural* and *deploy* models. Furthermore, this ontology can be exploited to enhance the selection and composition operations on a given set of annotated web services in this domain.

IV. Conclusions and Outlooks

The TOREADOR project proposes the MBDAaaS approach based on three (sub)models: *declarative*, *procedural* and *deployment*. In this paper, we propose the Big Data Model ontology in order to give a means to obtain a common conceptualization of the aforementioned models. In particular, the paper focuses on the conceptualization of the declarative model. The incompatibility management and the creation of OWL-S descriptions enabling different approaches for the selection task are only two of the possible advantages coming from the adoption of the proposed ontology. An important aspect to be considered is the possibility to extend in future work the ontology with new classes and properties coming from the conceptualization of procedural and deployment models.

Acknowledgment

This work was partially supported by the EU-funded project TOREADOR (ICT-16-2015, G.A. no. 688797).

References

[1] C. A. Ardagna, V. Bellandi, P. Ceravolo, E. Damiani, M. Bezzi, and C. Hebert, "A model-driven methodology for big data analytics-as-a-service," in *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017, pp. 105–112.

[2] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. Patel-Schneider, Eds., *The Description Logic Handbook: Theory, Implementation, and Applications*. New York, NY, USA: Cambridge University Press, 2003.