Semantic Support for model based Big Data Analytics-as-a-service (MBDAaaS)

Domenico Redavid, Donato Malerba, Beniamino Di Martino, Antonio Esposito, Claudio Agostino Ardagna, Valerio Bellandi, Paolo Ceravolo and Ernesto Damiani

Keywords: Big Data as a Service; Semantic Web Services; Big Data ontology; Parallel Patterns;

Abstract With the growing interest in Big Data technologies, companies and organizations are devoting much effort to designing *Big Data Analytics* (BDA) applications that may increase their competitiveness or foster innovation. However, BDA design requires expertise and economic resources that may not always available. To overcome this limit, the TOREADOR project has proposed a *model-based BDAaaS* (MBDAaaS) approach to guarantee the automation of BDA applications design, allowing users to focus on business cases without having to deal with technical aspects of data storage and management. Although many platforms providing BDA services are available, most of them exploit ontologies only for data representation and not for describing the BDA computation itself. This paper describes how the Semantic technologies meet MBDAaaS in the TOREADOR project.

Domenico Redavid
CINI Big Data Laboratory,Consorzio Interuniversitario Nazionale per l'Informatica, CINI, Bari, Italy, e-mail: domenico.redavid@consorzio-cini.it

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

Beniamino Di Martino, Antonio Esposito
Engineering Dept., Campania University "Luigi Vanvitelli" and CINI e-mail: beniamino.dimartino@unina.it, antonio.esposito@unicampania.it

Claudio Agostino Ardagna, Valerio Bellandi, Paolo Ceravolo, Ernesto Damiani
Computer Science Department, University of Milan e-mail: name.surname@unimi.it

1 Introduction

Big Data market is expected to substantially grow in the next years, and Big Data technologies are introducing a Copernican revolution in the areas of data storage, processing, and analytics. However, the impact and diffusion of Big Data technologies are lowered by the scarcity of professional profiles with the necessary background and competence to use them, especially in SMEs.

A recent trend has underlined the relevance of users' requirements and developed the idea that achieving the full potential of Big Data analytics needs to embrace a model-based approach [2]. Traditional data modeling, which focused on resolving the complexity of relationships among schemata [7], has been neglected as no longer applicable to Big Data scenarios. In TOREADOR [4], we take a different view: in addition to data representation, Big Data models should provide a shared specification of the process to manage data resources (including anonymization and privacy-preservation procedures) and of the computations to be done over them. These models also need to provide all the information to carry out Big Data analytics over commodity execution platforms. A practical goal for TOREADOR is to provide solutions where end users define their expectations on goals to be achieved with Big Data analytics, while smarter engines manage and compose solutions to deploy Big Data architectures and carry out the expected analytics.

2 TOREADOR Methodology

Model-Driven Big Data Analitycs-as-a-service (MBDAaaS) is responsible for all activities aimed to configure and execute the Big Data analytics involving the following roles: *i) Big Data customer* specifying the goals of its Big Data Campaign (BDC), *ii) Big Data consultant* helping the Big Data customer in specifying all customizations needed to execute her analytics, *iii) MBDAaaS platform* that is responsible for semi-automatically managing and executing a a BDC on a *Big Data platform*. Figure 1 shows the process of the proposed MBDAaaS that is composed

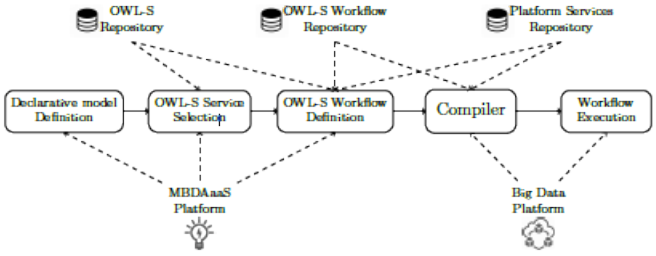

Fig. 1 MBDAaaS Methodology: Execution steps

of five main steps. In the first step (*Declarative Model Definition*), the Big Data customer produces a declarative model specifying the goals of a BDC. Within the TOREADOR framework the declarative model are grouped into five conceptual areas, namely Representation, Preparation, Analytics, Processing, and Display and Reporting. In the second step (*OWL-S Service selection*), the OWL-S services compatible with the declarative model specification are selected. We note that service selection is done on the basis of the annotations to the services in the OWL-S ontology, mapping them to indicators/objectives in the declarative model. In the third step (*OWL-S Workflow Definition*), the Big Data consultant uses the MBDAaaS platform to define the abstract workflow of the Big Data campaign. It is generated by composing the selected OWL-S services and represents the procedural model. In the fourth step (*MBDAaaS Compiler*), MBDAaaS platform transforms the OWL-S workflow in a platform-dependent workflow. The latter represents the deployment model, crucial to build a semi-automatic MBDAaaS and puts some strong constraints on the generality of the compiler, which needs to adapt to the selected target Big Data platform. Finally, in the fifth step (*Workflow Execution*), MBDAaaS platform executes the analytics on the target Big Data platform. In order to allow users to refine and tune the BDA applications obtained through the MBDAaaS approach, a Code-based approach has been developed. Such an approach, which is completely independent on yet integrated with the MBDAaaS, allows users to annotate their code with Parallelization Directives, based on Computational Patterns, which are interpreted by a Compiler to fill Skeletons and provide multi-platform deployments. Section 3.2 reports the semantic-based model behind the Directives' and Patterns' representation.

2.1 State of the Art

Several papers have tried to classify data analytics service. The work [13] proposes to exploit three types of service-generated Big Data to enhance the quality of a service-oriented system n order to provide the common functionality of Big Data management and analysis.

The authors of [5] highlight the need to develop appropriate and efficient analytical methods to leverage massive volumes of heterogeneous data in unstructured text, audio, and video formats as well as the need to devise new tools for predictive analytics for structured Big Data adopting statistical methods to devise inferences from sample data. Authors also remark that the heterogeneity, noise, and the massive size of structured Big Data calls for developing computationally efficient algorithms that may avoid Big Data pitfalls, such as spurious correlation.

The book [3] proposes general Big Data principles and indications on how to organize large volumes of complex data and how to achieve data permanence when the content of the data is constantly changing are given. General methods for data verification and validation, as specifically applied to Big Data resources as well as to find relationships among data objects held in disparate Big Data resources, when the data objects are endowed with semantic support.

A further Big Data analytics platform was proposed in [10], the aim is exploring and querying Big Data and developing algorithms through collaboration between data owners, scientists, and developers. In [12], a BDAaaS platform combining hierarchical and peer-to-peer data distribution techniques has been proposed to reduce the data loading time.

Recently, in [9] the authors propose to use semantic technology in assisting data analysts/data scientists when selecting the appropriate modeling techniques through the *Analytics Ontology* that supports inference mechanism for semi-automated model selection. Other works have been proposed for combining ontologies and Big Data. For example, [6] focuses on the combination of ontology-based approaches and Big Data as a solution for some problems related to extraction of meaningful information from various data sources.

As seen, an added value of the approach proposed in TOREADOR project with respect to those reported in this section is the systematization of the MBDAaaS into a model-driven specification process that covers the different areas defined by the TOREADOR Declarative Model [2, 1].

3 Semantic-based Representation Models

The steps of MBDAaaS presented in the section 2 have several points strictly related with the semantic of the information needed to accomplish the different targets. In particular, the semantic of this information can be useful to simplify some operations where the interaction with users with different skill level is required.

As we have seen, MBDAaaS is characterized by the use of OWL-S. OWL-S proposes a structure to represent the semantics of a service (atomic or compound) but the semantics of the information that the service uses are represented through existing OWL ontologies or defined ad-hoc. In details, OWL-S enables semantic descriptions of Web services using the *Service Model* ontology, which defines the OWL-S process model. Each process is based on the IOPR (Inputs, Outputs, Preconditions, and Results) model. *Inputs* represent the information required for the execution of the process. *Outputs* represent the information the process returns to the requester. *Preconditions* are conditions imposed on *Inputs* that have to hold in order to invoke the process in a correct manner. Since an OWL-S process may have several results with corresponding outputs, the *Results* provide a mean to specify this situation. Each result can be associated with a result condition, called *inCondition*, which specifies when that particular result can occur. When an *inCondition* is satisfied, there are properties associated with this event that specify the corresponding output and, possibly, the *Effects* produced by the execution of the process. The OWL-S conditions (*Preconditions*, *inConditions* and *Effects*) are represented as logical formulas. Since OWL-DL offers limited support to formulate constructs like property compositions without becoming undecidable, a more powerful language is required for the representation of OWL-S conditions. Furthermore, OWL-S Composite processes (decomposable into other Atomic or Composite processes) can be specified

by means of the following *control constructs*: **Sequence**, **Split**, **Split + Join**, **Any-Order**, **Choice**, **If-Then-Else**, **Iterate**, **Repeat-While and Repeat-Until**, and **As-Process**.

3.1 An OWL Ontology for TOREADOR concepts

In TOREADOR project we are developing an OWL ontology, named BDM Ontology[1] in order to model concepts coming from *Declarative*, *Procedural* and *Deployment* models [11]. At current stage, the *Declarative* Model ontology section is the most mature since its definition has started from the begin of the project (see Fig. 3,a)). The *Procedural* model section contains a set of concepts that are specific for the services currently defined, while *Deployment* section will be developed in accordance with the work being done on (*Workflow Execution*).

In these five different conceptual areas requirements have been defined using the Concept maps (CMAP) tool[2]. CMAP are graphical tools for organizing and representing knowledge by means of concepts and its relationships.

Firstly, we need to specify which are the Big Data Areas of the declarative model. 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.

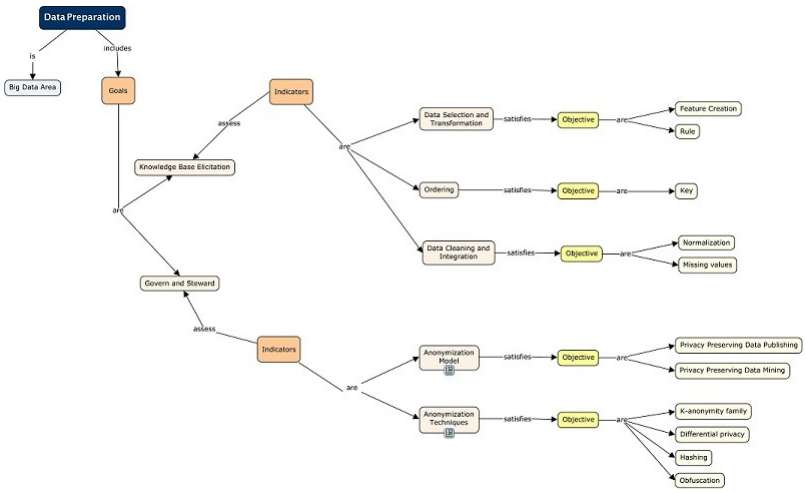

Fig. 2 Part of the CMAP declarative model describing the Data Preparation area

[1] Big Data Model Ontology `www.consorzio-cini.it/~lab-bigdata/BDMOntology.owl`

[2] Conceptual Maps, `cmap.ihmc.us`

bdmo:DataPreparation rdfs:subClassOf bdmo:BigDataArea. For sake of space, we will describe the details of the *Declarative* models focusing only on one of the areas reported above, i.e., Data Preparation, the other ones are been modeled in the same way.

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 the sub-hierarchy (see Fig.3, b)). In the case of the area describing Data Preparation, there is 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: an FI cannot assess both the Knowledge Base Elicitation and Govern and Steward goal. Thus it is reasonable to assume that such indicators aim 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*.

For avoiding cases where the same FG concerns two or more Big Data Area at the same time, we defined the concept representing all the functional indicators related a specific Big Data Area (see Fig. 3, c)). For the Data Preparation area, we introduced the ontology concept *bdmo:DataPreparationFunctionalIndicator* subsumed by *bdmo:FunctionalIndicator*. *bdmo:DataPreparationFunctionalIndicator* 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 personally identifiable information from data sets. This is made introducing the concepts *bdmo:AnonymizationTechniques* and *bdmo:AnonymizationModels*. Similarly to the FI, the FOs are grouped by the Big Data Area (see Fig. 3, d), 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

(a) Big Data Areas (b) Data Preparation Functional Goal (c) Data Preparation Functional Indicators (d) Data Preparation Functional Objective

Fig. 3 Fragments of the ontology derived from the declarative model

satisfied. Note that, as in the the FIs and the FOs have been declared as disjoint in order to prevent either cases of a FI assessing more FGs o FI satisfies more FOs.

In detail, the resulting ontology is composed of 636 axioms, 196 classes, 42 object properties and 11 data properties resulting in $\mathscr{ALCHF}(\mathscr{D})$ expressiveness. The ontology representation of 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 as sub-concepts (also for their properties) of the FGs / FIs / FOs related to that area. For instance, it is sufficient to add the new concept as 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 a 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. Furthermore, this ontology can be exploited to enhance the selection and composition operations on a given set of annotated web services in this domain.

3.2 Semantic Representation of Patterns and Directives

In the context of TOREADOR project, a mapping process useful to distribute the computation of a set of algorithms, described in terms of parallelization directives among computational nodes hosted by different platforms has been proposed. The parallelization paradigms that will be used to distribute the computation will be derived from the directives used to describe the algorithm to be parallelized and will be implemented by filling skeletons according to suitable patterns. In this approach, the analyzed Algorithm's representation is annotated with suitable parallelization directives, composed by semantically described micro-functions. Such directives determine a subset of possible Patterns, selectable from a shared knowledge base, which can be used to fill the Skeletons. Once one specific Pattern has been selected, either via the information deriving from the Declarative Model or directly by the user, an agnostic Skeleton (or set of Skeletons) is filled. Only after a specific target platform has been chosen, Vendor Specific Skeletons are produced together with deployment templates for that platform. The **Procedural Model** used to describe the Algorithms and their realization via Patterns and Skeletons, consists in a multi-layer representation, in which the top layer provides a high-level vision of the algorithm, which is enriched with new details as we descend towards the bottom layer.
The representation reported in figure 4, can be divided into two main sections:

- The **Semantic Description** of the Procedural Model in which, thanks to semantic-based technologies such as OWL and OWL-S, the Algorithm is internally represented in both its structural (needed computational nodes, data structures, and their relationships) and behavioral (workflow, decision points, control structures) aspects.

- The **Procedural Realization** which, instead, provides practical implementations of the modelled algorithm, through the refining of Skeletons and their mapping to Web and Cloud Services for their execution.

The overall semantic model, derived from existing representations of Application and Cloud Patterns as described in [8], is a graph-based representation, structured into five conceptual layers. The five conceptual levels are the following: – The *Parameters Level* represents the description of the data type exchanged among services as input and output of the operations. – The *Operations Level* represents the syntactic description of the operations and functionalities exposed by the cloud services; it provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns, expressed through WSDL, JSON strings or REST parameters, according to the supported technology. – The *Services Level* represents the semantic annotation of the provider dependent services (exposed through OWL-S) and the supporting ontologies needed to identify the provider supported operation, input, and output parameters. This level presents details of the provider platform architecture, the functionality exposed and the underlining details. – The *Computational Patterns Level* represents the semantic description of Technologically Independent and Technologically Dependent Patterns, realized through an OWL representation. – The *Algorithmic Semantic Description Level* represents models describing the algorithms to be ported and implemented. An *Algorithmic Semantic Description* is a composition of application components embodying application domain functionalities, services, and micro-functions

3.2.1 Directives' categorization via OWL

In order to make it possible for the TOREADOR user to discover the available Skeletons along the services offered by the TOREADOR platform, the parallel directives will be semantically described and then integrated within the OWL-S based definitions already produced for the description of services and micro-functions.
In particular, a simple categorization, based on the specific Parallel Computation

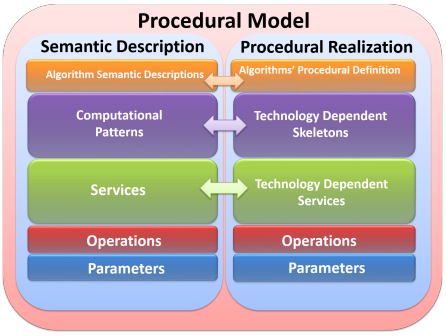

Fig. 4 The Procedural Model and its Components

Paradigm implemented by the directives, has been described. A representation of the classes composing such ontology-based categorization has been reported in figure 5. The ontology classifies the different directives which have been defined so far and allows for the further classification of the parameters and functions which represent their input. The three main classes are represented by:

- The **Parallel_Paradigm_Directive** defines all possible parallel directives. It contains two subclasses, namely the **Data_Parallelism_Directive** and the **Task_Parallelism_Directive**. For both classes, a generic and specific sub-class has been defined. The directives appear as instances of these classes.
- The **Skeleton** class representing the available skeletons, which can be produced by applying a set of rules.
- The **Directive_Input** class, acting as a placeholder for the three different input categories accepted by the directives: **Function**, **Data** and **Parameters**

The remainder of the sub-classes is not described here, as their names are self-explicative.

4 Conclusions

In this paper an MBDAaaS approach to support the design of Big Data Analytics applications has been proposed, to address the limitations encountered by small companies which lack expertise and economic resources when it comes to Big Data. In particular, the MBDAaaS has at its core OWL-S descriptions of services which can be used to build-up BDA applications. However, while OWL-S can represent the services, there is the need to also describe the applications' characteristics, via a semantic-based representation which can guide the composition of applications. To serve this purpose semantic models, based on OWL ontologies, have been provided

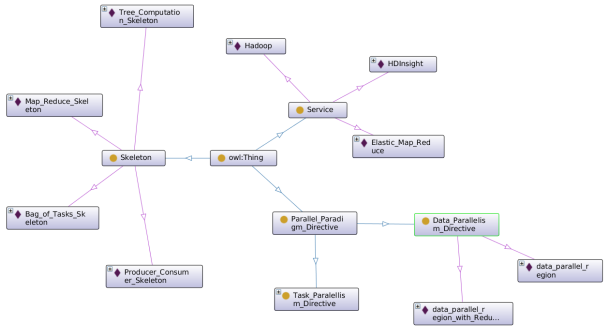

Fig. 5 The ontology used to categorize the parallel directives and their inputs

for both the Declarative and Procedural models describing an application. Also, directives to annotate source code and automatize its parallelization according to a set of Parallel Patterns have been provided and semantically categorized, to support users in developing their own BDA solutions from scratch. Future works will focus on a further integration between the semantic models behind the Declarative and Procedural models, and the Deployment of applications on target platforms.

Acknowledgements This work was partially supported by the EU-funded project TOREADOR (ICT-16-2015, contract no. H2020-688797).

References

1. Claudio A. Ardagna, Valerio Bellandi, Michele Bezzi, Paolo Ceravolo, Ernesto Damiani, and Cedric Hebert. Model-based Big Data Analytics-as-a-Service: Take Big Data to the Next Level. *IEEE Transactions on Services Computing*, 2018.
2. Claudio A. Ardagna, Paolo Ceravolo, and Ernesto Damiani. Big data analytics as-a-service: Issues and challenges. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 3638–3644. IEEE, 2016.
3. Jules J. Berman. *Principles of Big Data: Preparing, Sharing, and Analyzing Complex Information*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2013.
4. Ernesto Damiani, Claudio Ardagna, Paolo Ceravolo, and Nello Scarabottolo. Toward Model-Based Big Data-as-a-Service: The TOREADOR Approach. In *Advances in Databases and Information Systems*, pages 3–9. Springer, 2017.
5. Amir Gandomi and Murtaza Haider. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2):137 – 144, 2015.
6. Agnieszka Konys. *Ontology-Based Approaches to Big Data Analytics*, pages 355–365. Springer International Publishing, Cham, 2017.
7. Sam Madden. From Databases to Big Data. *IEEE Internet Computing*, 16(3):4–6, 2012.
8. Beniamino Di Martino, Antonio Esposito, and Giuseppina Cretella. Semantic Representation of Cloud Patterns and Services with Automated Reasoning to Support Cloud Application Portability. *IEEE Transactions on Cloud Computing*, 5(4):765–779, Oct 2017.
9. Mustafa V. Nural, Michael E. Cotterell, and John A. Miller. Using Semantics in Predictive Big Data Analytics. In *2015 IEEE International Congress on Big Data*, pages 254–261, June 2015.
10. Kyounghyun Park, Minh C. Nguyen, and Heesun Won. Web-based collaborative big data analytics on big data as a service platform. In *2015 17th International Conference on Advanced Communication Technology (ICACT)*, pages 564–567, July 2015.
11. Domenico Redavid, Roberto Corizzo, and Donato Malerba. An OWL Ontology for supporting Semantic Services in Big Data platforms. In *2018 IEEE International Congress on Big Data, BigData Congress 2018, San Francisco, CA, USA, July 2-7, 2018. (Accepted)*, 2018.
12. Luis M. Vaquero, Antonio Celorio, Félix Cuadrado, and Rubén Cuevas. Deploying Large-Scale Datasets on-Demand in the Cloud: Treats and Tricks on Data Distribution. *IEEE Trans. Cloud Computing*, 3(2):132–144, 2015.
13. Zibin Zheng, Jieming Zhu, and Michael R. Lyu. Service-generated Big Data and Big Data-as-a-Service: An Overview. In *2nd IEEE International Congress on Big Data*, pages 403–410, 2013.