International Journal of Innovative Technologies
WWW.IJITECH.ORG
ISSN 2321-8665
Volume.08, Issue.02,
September, 2020,
Pages:106-110

Towards Artificial Intelligence Multiple Cloud Framework

R. SHILPA[1], P. R. RAJESH KUMAR[2]
[1]PG Scholar, Dept of CSE, Sri Krishnadevaraya University, Anantapuramu, AP, India.
[2]Lecturer, Dept of CSE, Sri Krishnadevaraya University, Anantapuramu, AP, India.

Abstract: Cloud users worldwide are looking at the next generation of artificial intelligence (AI) powered cloud management tools to automate cloud performance tuning and anomaly detection. To be effective across clouds, AI tools need a common representation of cloud services and support for machine learning optimization targeting multiple objectives. We put forward the notion that ontology-based models can support both.

Keywords: Cloud, Artificial Intelligence (AI), ERP, DBMS.

I. INTRODUCTION

Cloud computing service model, based on elastic on-demand allocation of virtual resources, turned out to be suitable for supporting data-intensive applications such as artificial intelligence (AI) pipelines, which exploit cloud scaling to perform large-volume data ingestion, preparation, model training, and inference. Today, many companies worldwide rely on the cloud for large AI workloads, making cloud management and control a key issue for AI pipelines. Experience has shown that different stages of an AI pipeline may have diverse nonfunctional requirements (e.g., different data confidentiality levels); for this reason, more and more organizations adopt edge-cloud or multi cloud deployment strategies, deploying each pipeline stage on a different public or private cloud. Multi cloud deployment promises to prevent provider lock-in, take advantage of dynamic resource pricing at run-time, and secure the content exchanged or stored on the multi cloud. Early approaches to multi cloud deployment were mostly programmatic: They consisted of multi cloud libraries, which allowed run-time mapping of computations to the resources of multiple cloud providers. However, programmatic control of multi cloud deployment hard codes deployment decisions in scripts, which may lead to lack of flexibility and, ultimately, to inefficiency and loss of control. Some recent models of multi cloud services, such as MANTUS,9 support decoupling of the architecture model and the cloud resources used.

This separation allows users to apply dynamic reconfiguration to tune the resources used on each cloud according to performance and cost targets. Still, reconfiguration procedures are mostly manually coded, and little support is available for automated management across clouds. Our own research efforts focused on cloud architecture representations that rely on a service ontology for defining their entities. 9 Using the Web Ontology Language for Services (OWL-S) standard to model cloud Resources, Services and Patterns5 provides a high-level framework that can be used not only for architecture description and maintenance, but also through automated reasoning can effectively support Multiple Cloud Service Discovery and Brokering, and architecture agnostic applications' development and deployment. The approach can be extremely useful when dealing with applications that can be deployed in a multi cloud environment. Let us consider a common business intelligence application in which an extraction, transformation and loading process retrieves data from a database (DBMS), a customer relationship management (CMR), and an enterprise resource planning (ERP) system, and preprocesses them for further analysis after their storage in a data warehouse system.

The CMR and ERP components utilize data coming from their own databases, while data recorded in the data warehouse are used by the OLAP system and by the data mining component to perform market analysis. Introducing such an architecture arises a series of concerns, first of all regarding interoperability, since the several application's components need to interact to achieve the business goal. If each of such components were to be hosted by different cloud providers, because of the specific functionalities they provide or for economic reasons, one could be concerned about the real capability of such components to communicate due to differences in the communication interfaces provided by the providers. A semantic representation, such as the one that will be presented in this paper, can ease the communication difficulties and enable interoperability. Such capabilities have been demonstrated through several industrial case studies within the FP7 MOSAIC project.10 Ontology-based models' feasibility for representing the computation of AI pipelines has been experimented in several industrial case studies within the H2020 TOREADOR project.

II. SEMANTIC REPRESENTATION OF CLOUD PATTERNS AND SERVICES

The semantic representation reported in Figure1 has been developed to ease the portability and interoperability issues that may arise when either trying to compose multiple cloud services, or migrating data and applications from a platform to another.8 The representation, of which we report an overview, is constituted by a multilayered stack of conceptual models, connected to one another in a graph like structure but that are still independent. The conceptual models are divided into the following patterns.

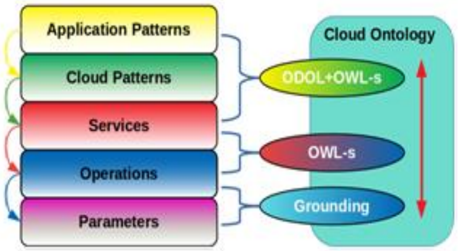

Fig1. Semantic representation of cloud patterns and services.

The Application Pattern layer describes general applications and their components, with no specific connection to an implementing technology or platform. The solutions represented by such patterns can be theoretically applied to different scenarios, independently from the platforms, programming languages, or technologies involved.

- The Cloud Pattern layer provides the description of cloud-based solutions to be applied when implementing applications. The semantic representation helps not only to efficiently categorize them, but also to identify eventual connections, similarities, and similar application scenarios. As many cloud patterns can be used together to form more complex one, the representation also supports pattern compositions. Cloud Patterns' components are represented by cloud services, which are described in the lower layer.
- The Cloud Service layer focuses on the description of cloud services and functionalities they expose. The objective of this layer is to provide a common semantics for the description of services exposed by cloud platforms so that comparisons can be done both at functional and operational levels.
- The Operation layer describes functionalities exposed by services and not referring to the cloud, including resources exposed through the web.
- The Ground layer represents the connection between the abstract description of input/output parameters and operations exposed by services, and their actual implementation. While the WSDL standard is the native format used for this purpose, any other standard way to represent the invocation interface of cloud or web services can be used.

Different technologies have been employed to implement the layers, but they are all based on the standard OWL language for ontology description. The application and cloud patterns layers have been implemented using a combination of ODOL,3 a semantic representation of design patterns which has been augmented and adapted to describe the static entities of patterns, and OWL-S, an ontology designed to describe web services which, in our case, has been exploited to define the dynamic behavior of patterns. OWL-S has also been used to describe the services and operations layers, for which it provides native support. Fig2 portrays the graphical representation of the semantic description of patterns and services that we have briefly introduced. In particular, Figure 2(a) shows the whole representation, with all the classes and connections existing among its components, while Figure 3.2(b) zooms in on the pattern class and its direct connections. All layers are supported by a background ontology, which provides a common set of terms to describe each component of the described patterns, services, and operations, and which will be further described in the following sections.

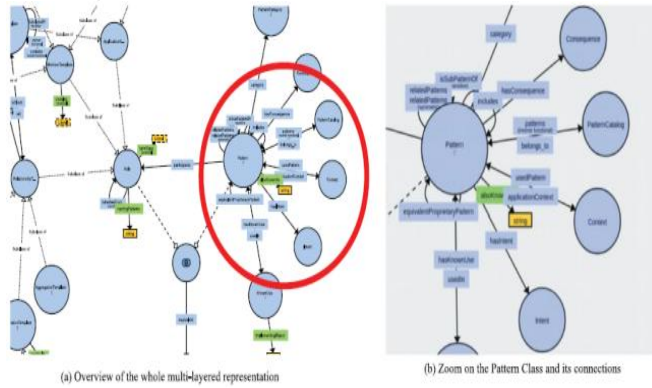

Fig2. Graphical representation of the semantic multilayered description of patterns and services.

A. Nonfunctional Properties Example for Legal Constraints

While functional composition of cloud services is by itself a complex, yet interesting, matter, there are nonfunctional aspects that should be taken in consideration when selecting a specific service to fulfill user needs. The semantic description presented in the previous section covers functional requirements of applications and cloud services, making it possible to select and compose them according to the required functionalities. However, it does not take in consideration nonfunctional requirements, and legal constraints regarding the use of data in a specific environment or country are a strong limitation when selecting a cloud service. The framework, 6 of which we report an overview here, provides a user-oriented tool, that is able to check the compliance of cloud services offered by different cloud platforms, in respect to the Italian regulations. A simplified graphical interface allows the user to interact with the framework, enabling her to specify the application's requirements. More specifically, users can do the following tasks.

- Select the nature of the data that will be treated among a set of predefined categories (sensitive, health, judicial, or not subject to data protection).
- Decide the scope and aim of the data treatment, by selecting one of the available categories (scientific, statistical, historical, or generic).
- Specify the cloud service provider she wants to use for its resources, and eventually narrow down the selection of available services offered by that provider, according to the requirements of her application.
- Decide the location of the data center, choosing among the possible locations offered for the selected provider.

The framework was developed to run with the Italian legislation and in particular is based on the formalization of the Italian Legislative Decree 196/2003 and Italian Code for Digital Administration. It can be used in different ways. The basic usage allows the user to establish the compliance of a specific service, running in a certain location or exploiting a specific data-center, to the regulations. Of course, the kind of data used and the purpose of their processing should be known beforehand. Conversely, the user can discover the kinds of data processing that are allowed for a certain combination of cloud provider and data-center location. The knowledge base exploited by the framework contains two main information sets the semantic rules derived from the legislation the semantic description of the terms of service of the providers' cloud services (such as Amazon, Microsoft, IBM).

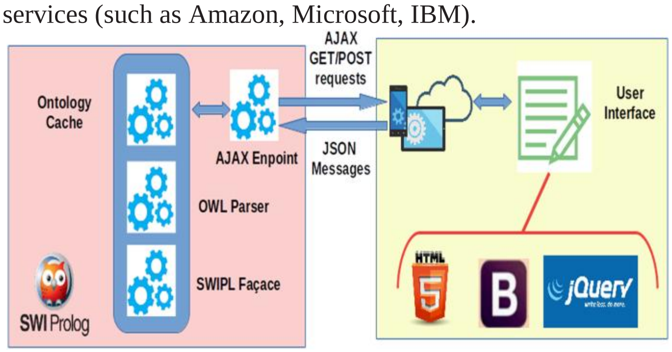

Fig3. Architecture of the legislation-aware framework.

The first set of information has been obtained by analyzing, via natural language processing (NLP) techniques, the reference laws on privacy. The NLP analysis translated prescriptive sentences into logical rules, while ontologies have been used to describe the terms of services exposed by the cloud providers. Figure 3.3 shows the main components of the framework and how they are related. There are two main components: the back end, composed of the Ontology Cache, the OWL Parser, and the SWIPL Facade, and the front end, represented by the user interface implemented in HTML. The OWL Parser extracts information from ontologies coded in OWL and converts them into Prolog facts that are then questioned using the rules by the SWIPL Facade component.

B. Cloud Ontology

An ontology is needed to provide a common background terminology for all the semantic layers of the cloud representation. The cloud ontology7 describes an approach, which aims at providing such an ontology, via a description of cloud services based on well-known semantic technologies, such as OWL and OWL-S. Here, we provide an overview of such an ontology, as it represents a set of semantic definitions which the ontology description provided in the previous sections requires to be effective. The whole ontology is constituted by a set of interrelated subontologies, connected through ad-hoc links and properties, which define a common representation base for existing cloud services, together with the operations they expose and the parameters they exchange. Three main layers compose the cloud ontology.

- The upper layer contains the Agnostic Service Description Ontology, which provides a common terminology to describe cloud services, resources, operations, and parameters. Through this ontology, it is possible to annotate cloud entities by using a general and shareable catalog of concepts, which enables their discovery and comparison.
- The central layer is represented by the Cloud Services Categorization Ontology, which offers a categorization of cloud services and virtual appliances. The several categories are based on the specific functionalities the different services and appliances offer, as declared by their respective vendors. By importing the upper ontology, references to platform specific services and resources, which are organized according to the proposed categorization, can be directly related to Agnostic descriptions to enable comparisons.
- The bottom layer is in turn composed of two different groups of ontologies, describing proprietary specific services, operations, and parameters. In particular, the Cloud Provider Ontology set defines proprietary concepts that describe a specific cloud provider's offer: a single and independent ontology exists for each cloud provider, which can be added or removed independently, in a modular fashion.

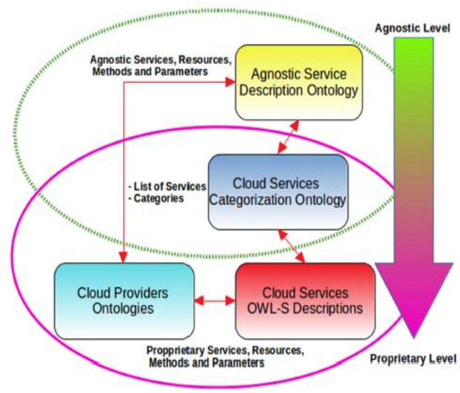

Fig4. Architecture of the cloud ontology.

Purpose of this set of ontologies is to describe proprietary services, resources, and related operations with the providers' specific concepts, which can be categorized against the Cloud Services Categorization Ontology and further annotated through the Agnostic Service Description Ontology. On the other hand, the Cloud Services OWL-S Description set describes the several services offered by the different providers in terms of their internal work flow and grounding; in this way, the ontology provides the information needed to automatically instantiate such services. Figure 4 portrays the aforementioned ontology, highlighting the three different layers and their connections.

III. SIMULATION RESULTS

The prescriptive sentences has been translated in logical rules. An example of these rules is reported in figure 5. This rule states that if the data are statistical or historical or scientific, when the treatment ends the data must be destroyed. In this case the rule ensures to verify that the service implements the auto-scrub functionality. If not a warning, that indicates the actions to be undertaken manually in case you wish to use the service analyzed, will be notified to the user.

```
DEFINE service → featureDataAutoScrub

IF ( NOT scope →(historical OR statistic OR scientific ) AND NOT service → featureDataAutoScrub )

THEN WARNING [reference to the article] "It's necessary to delete data manually when the treatment ends."
```

Fig5.A logical rule derived from the law for data treatment.

The description of the terms of the services of the provider has been provided by using ontologies, in particular figure 6 reports the main classes and relationships used to describe the cloud services features.

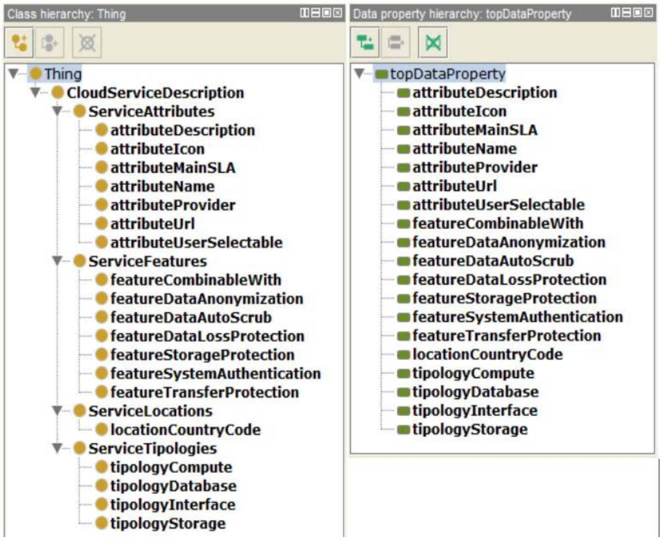

Fig6.Class and relationships of the Cloud Service Ontology.

By means of the input provided by the user that describes the application requirements, and the description of the cloud services functionalities, the rules that represent the law will be examined in order to verify the compliance of the services. A prototypical application has been implemented in order to test these rules. In particular figure 7 reports the architecture of the implemented prototype. There are two main components: the back end that is composed by the provider and the specific service, the location of the data center among the ones possible for the specific provider.

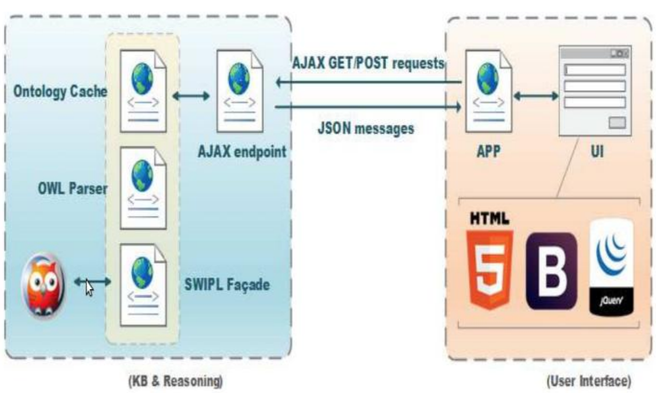

Fig7. Application architecture.

Fig8.Results of the query on the knowledge base.

The result of this specific request is illustrated in figure 8. Due to the specific kind of data and the particular kind of treatment the selected service need to be enriched with some feature that can be complemented by using one of the complementary services suggested. Furthermore the system lists the rules that are satisfied by the particular services and some warnings that represent tips to the user in order to advise him of some normative obligations, as instance to verify if the data owner have already signed a consent form.

IV. CONCLUSION

The adoption of cloud computing may bring along common and significant legal issues that can arise in contracts with vendors. Most of these issues are basically business ones, requiring business decisions, but most of them are related to legal aspects, in the sense that they may conflict with the legislation in force. Maintaining the levels of protection of data and privacy, confidentiality and security required by current legislation in cloud computing infrastructure is a new challenge, as is meeting the restrictions on cross-border data transfer (the problem of data location) and holding the ownership of data. Unfortunately, among the many initiatives that have been carried out to define standards for law representation formalism definition there aren't works that approach the topic from the point of view of automatize the checking of law compliance. In particular if we think at cloud computing issues and, as a significant example, at the specific scenario in which a public administration will store citizens data on the cloud, a very desired feature will be to verify the compliance of the vendors contracts with legislation of the country. This can be achieved by exploiting the existing standards already developed for law representation and annotation and by enriching them with semantic formalization of information related to the technical aspect of the normative disposition which can be useful to match the vendors contracts with the customer's requirements. In particular this can be implemented by representing the regulations that emerges from the legislation as semantic assertions and inferences rules, the characteristics of the cloud providers offers and the requirements of the customers as semantic assertions and by using a shared semantic vocabulary. In this paper we propose prototypical implementation of a semantic based application that can be used as a starting point to implement a framework that will support the automatic checking of cloud application compliance with legislation.

V. REFERENCES

[1] E. Damiani,C. Ardagna, P. Ceravolo, and N. Scarabottolo, “Towardmodel-based big data-as-a-service: The TOREADOR approach,” inProc. Eur. Conf. Adv. Databases Inf. Syst., 2017, pp. 3–9.

[2] B. Di Martino and A. Esposito, “A tool for mapping and editing of cloud patterns: The semantic cloud patterns editor,” Stud. Inform. Control, vol. 27, no. 1, pp. 117– 126, 2018.

[3] J. Dietrich and C. Elgar, “An ontology based representation of software design patterns,” in Proc. Des. Pattern Formalization Techn., Jan. 2007, pp. 258–279.

[4] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, Cloud Computing Patterns—Fundamentals to Design, Build, and Manage Cloud Applications. New York, NY, USA: Springer, 2014.

[5] L. van derMaaten and G. Hinton, “Visualizing data using t-SNE,” J.Mach. Learn. Res., vol. 9, pp. 2579–2605, 2008.

[6] B. Di Martino, G. Cretella, and A. Esposito, “Towards a legislation-aware cloud computing framework,” Procedia Comput. Sci., vol. 68, pp. 127–135, 2015.

[7] B. Di Martino, G. Cretella, A. Esposito, and G. Carta, “An OWL ontology to support cloud portability and interoperability,” Int. J. Web Grid Services, Indersci., vol. 11, no. 3, pp. 303–326, 2015.

[8] B. Di Martino, A. Esposito, and G. Cretella, “Semantic representation of cloud patterns and services with automated reasoning to support cloud application portability,” IEEE Trans. Cloud Comput., vol. 5, no. 4, pp. 765–779, Oct./Dec. 2017.

[9] A. Palesandro, M. Lacoste, N. Bennani, C. Ghedira-Guegan, andD. Bourge, “MANTUS: Putting aspects to work for flexiblemulti-cloud deployment,” in Proc. IEEE10th Int. Conf. Cloud Comput., 2017, pp. 656–663.

[10] D. Petcu et al., “Experiences in building a mosaic of clouds,” J. Cloud Comput., vol. 2, no. 1, 2013, pp. 2–12.

[11] A. Bar-Hillel, “Learning distance functions using equivalence relations,” in Proc. 20th Int. Conf. Mach. Learn., 2003, pp. 11–18.

[12] N. Guarino and C. Welty, “Identity, unity, and individuality: Towards a formal toolkit for ontological analysis,” in Proc. 14th Eur. Conf. Artif. Intell., 2000, pp. 219–223.