A Proposal for the Specification of Data Mining Services in Cloud Computing

Manuel Parra-Royon and Jose M. Benitez
Dept. of Computer Science and Artificial Intelligence (DECSAI), DaSCI, DiCITS, University of Granada, Spain

Keywords: Cloud Computing, Data Mining, Linked Data, Services Description.

Abstract: For more than a decade, languages such as WSDL, SA-WSDL, OWL-S and others have been proposed to tackle the problem of service description. These service description languages do not take into account key aspects of cloud computing. Inherent features such as interaction techniques between entities, service-level agreement or pricing are necessary when defining a cloud computing service. Regarding cloud data mining services, specific issues of experimentation and the execution process should be included, among others. Following the Linked Data proposal, it is possible to design a specification for the exchange of data mining services and achieve the highest level of interoperability. In this paper we propose a schema of definition of data mining service in cloud computing using Linked Data and validate its operation by defining a complete service. Our proposal is suitable for fully defining data mining services in a comprehensive approach, including all aspects associated with an on-demand cloud service.

1 INTRODUCTION

Data mining platforms such as Weka[1] or Knime[2] enable data analysis from a local environment. With the rise of cloud computing, some Internet service providers offer data mining services, called ML-as-a-service (Machine Learning as a Service). These vendors, such as *Amazon AWS*, *Microsoft Azure* or *Google Cloud Platform*, have the capability to run data mining algorithms as an on-demand service. One of the advantages of using these services in the cloud is the ability to support large datasets.

Unfortunately, each vendor offers its own catalogue of services with their particular specification details. There is no standard proposal among service providers for defining the kind of cloud services.

This means that each provider has its own definition of service that is not compatible with the others, therefore it hampers the migration from one supplier to another. A cloud data mining service should not only run experiments and data analysis, but also take into consideration critical aspects such as authentication, catalog, service-level agreement, pricing, interaction and service configuration. These aspects make it even more difficult to define a service and the relationship between cloud providers and cloud consumers from other providers.

[1]https://www.cs.waikato.ac.nz/ml/weka/
[2]https://www.knime.com/

The modeling of such cloud services cannot be fully addressed with SOA (Newcomer and Lomow, 2005) service definition languages such as WSDL (Christensen et al., 2001), Universal Description Discovery & Integration UDDI (Bellwood et al., 2002) or SoaML (Elvesæter et al., 2010) among others or the proposed OpenML (Vanschoren et al., 2014) that does not focus on cloud data mining. Semantic web services definition languages such as SA-WSDL (Kopeckỳ et al., 2007), OWL-S (Martin et al., 2004) or WSMO (Kopeckỳ et al., 2009) describe all relevant aspects of cloud services to automate discovery and composition.

Linked Data(Bizer, 2009) allows to leverage information from multiple data sources, linking to other domains and semantic terms that Semantic Webberners2001semantic offers. Links to other data enrich the definition schema being created. With Linked Data we can link with other vocabularies and definition schemas to use them in the semantic definition of the elements of our schema. A vocabulary is a collection of terms for a purpose (healthcare, business, libraries, etc.). A schema refers to a data model that represents the relationships between a set of concepts and terms. Some types of schemas are relational database schemas, taxonomies and ontologies. Linked Data

Parra-Royon, M. and Benitez, J.
A Proposal for the Specification of Data Mining Services in Cloud Computing.
DOI: 10.5220/0006776905410548
In *Proceedings of the 8th International Conference on Cloud Computing and Services Science (CLOSER 2018)*, pages 541-548
ISBN: 978-989-758-295-0

uses the RDF (Allemang and Hendler, 2011) standards for data interchange. It allows to define services using RDF/XML, RDFa (Adida et al., 2008), Terse RDF Triple Language (Turtle) (Beckett et al., 2011) and query with SPARQL (Prud et al., 2006).

In this context we have developed a proposal that strives to provide a framework for data mining cloud services specification and deployment. It is called *dmcc-schema* and follows the recommendations of Linked Data. Our proposal has been designed to reuse existing vocabularies and ontologies, being able to compose data mining services in the cloud. Vocabularies such as *GoodRelations* (Hepp, 2008), *Machine-Learning Schema* (*mls*) or Web API Authentication (waa) (Maleshkova et al., 2010) among others have been used to define cloud data mining services with *dmcc-schema*.

This work is organized as follows. Section 2 examines the state of the art of service definition languages at both syntactic and semantic levels and the most recent proposals with Linked Data. Section 3 defines the schema and scope of *dmcc-schema* as a tool for defining services. Our Linked Data schema proposal for service definition is compared with the options in section 4. Section 5 below provides an example of how a cloud data mining service is designed; finally section 6 the conclusions are addressed.

2 RELATED WORK

Within the scope of service definition languages, there is a wide range of options to choose. More specifically, in data mining service definition languages, each provider or publisher of Internet services has its own protocol, interface or model to enable the interchange of information and service demand. Nowadays, there is no standardisation in terms of service definition languages, nor for cloud data mining services. There are many different proposals on the definition of services.

The aim of the WSDL standard, is to describe technical elements of web services such as the interaction of interfaces and protocols. WSDL is a syntactic specification for service description. This specification is not sufficient, since the precision in exchanging information between different entities requires additional elements that allow understanding between programming entities. WSDL for a web service consists of an XML description of its interfaces, where the methods, parameters and responses are defined. UDDI enables the definition of services by supporting description and discovery, and the technical interface for accessing those services.

Not only it is necessary to have a syntactic specification at the technical and functional level of the service, but the tendency in cloud computing services is to complete the definition with a semantic representation of them.

The definition of services at a semantic level is fundamental for the improvement of search and discovery, composition and integration.

In order to make a cloud service consumable by programming entities, it must be discoverable and invocable as well as it can be composed, verified, and monitored.

Several proposals for semantic web-based services description languages have been developed for over a decade.

Initial semantic proposals such as WSMO, OWL-S, SA-WSDL are considered the service specification. Some variants for *Representational state transfer* (REST) services such as hREST (Maleshkova et al., 2009) and Web Service Modeling Ontology (MicroWSMO) (Kopeckỳ et al., 2009) have also been taken into account. They add a light semantic definition for REST services, providing support for service descriptions to render them processable and machine-readable. More extensive and general ones such as USDL (Kona et al., 2009) and Linked-USDL (Pedrinaci et al., 2014) have allowed cloud services to be built considering most of the overall complexity of cloud services.

Linked-USDL creates a vocabulary to capture and share descriptions of general cloud services in the cloud in an open, scalable and highly automated way using Linked Data. Linked-USDL is the most complete proposal for the definition of cloud services since it covers all the scenarios of entities involved in the consumption and usage of cloud services, such as billing, technical aspects, service-level agreement (SLA), etc. It lacks a specific module for the definition of data mining services, which should be implemented or linked from another vocabulary that includes it. The schema provided by Linked-USDL offers much more than needed and adds additional complexity to our goal of defining a more effective cloud data mining services taking into account our objectives: pricing, authentication, service-level agreement, catalogue and data mining aspects.

Another proposal is *Exposé* (Vanschoren and Soldatova, 2010), designed to describe machine learning experiments. It is built on top of *OntoDM* (Panov et al., 2008), and underlies *OpenML*, a collaboration and meta-learning platform for machine learning.

Vocabulary for dealing with Machine Learning algorithms is included in *MLSchema* (Esteves et al., 2016). This schema can be used to represent algorithms, tasks, implementation and executions, as well as input and output data. *MEX* vocabulary (Esteves et al., 2015) also addresses the problem of sharing specific information about processing Machine Learning techniques in a lightweight way. The above alternatives do not take into account aspects of a cloud Data Mining service.

The semantic proposal offered by Linked Data (Bizer, 2009), allows to link information and, in this case, vocabulary distributed and accessible on the Web from providers. Linked Data, enables the exchange and discovery of services by reusing vocabularies and schemas defined by other entities. These vocabularies can be operated by machines, where referenced information can come from different sources.

The proposals referenced to above are very generic to cover the full spectrum of cloud services or do not take into account specific aspects of this type of cloud data mining services. The idea with *dmcc-schema* is to safeguard the step between a generic cloud service and data mining services, proposing a schema and vocabulary by using Linked Data to unite both facets.

3 OUR PROPOSAL

Our *dmcc-schema* approach is designed to enable the complete definition of cloud data mining service. It supports the definition and execution of data mining algorithms as well as all the core elements of the web service that a service provider can offer, such as discovery, composition, security, authentication, billing, catalogue, and interoperability.

The schema provided by *dmcc-schema* addresses the following aspects:

- **Service Catalogue**. It provides a catalogue of algorithms and enables the discovery of data mining services. The catalogue allows algorithms to be classified according to the type and problem of data mining they can address, such as regression, classification, clustering, association rules and preprocessing among others.
- **Authentication**. Eases the management and access of users or agents. It contains the basic features and mechanisms to enable authentication capacities for the use of the cloud data mining service.
- **Costs and Prices**. Defines all the necessary entities to manage the costs associated with the use of the service. In diverse aspects such as used CPU time, storage, number of instances or calls to the service among others. The execution of data mining algorithms requires intensive computation and a large amount of infrastructure resources. Such resources must be monitored with regard to define the pricing of the service.
- **Business**. Aspects such as who is the provider or consumer of the service and all its related information are essential to identify the data of the entity that acts interacting with the service. This data includes information such as legal aspects, contact information, etc.
- **Service-level Agreement**. When such services are provided by providers where a minimum quality of service agreed between both parties, producer and consumer, must be established.
- **Interaction**. To use the cloud data mining service, you have to define the points of interaction with the service. These interaction points enable you to access and use the service. The definition of this interaction can be given as RESTful API, for example.
- **Algorithms, Experimentation and Results**. This is one of the main features of the schema, which contains the definition of all the elements concerned with the execution of an algorithm by the service. Such elements are the inputs and parameterization, experimentation, results and models, as well as the selection of the specific algorithm of the service catalogue. It also supports different implementations of the algorithms, so that you can have several variants of the same data mining algorithm.

The *dmcc-schema* is represented using the Linked Data principles. Linked Data is gaining popularity for knowledge modeling. In terms of integration, it allows data to be integrated as part of the web services. In addition, the use of RDFs (Allemang and Hendler, 2011) in conjunction with service identification through URIs provides a uniform interface for data access.

The fact of using Linked Data principles facilitates the incorporation of new vocabulary into the schema, so *dmcc-schema* has been designed for the complete definition of an on-demand cloud data mining service. Vocabularies such as GoodRelations, DublinCore Terms (DCterms) (Weibel et al., 1998), Machine Learning Schema (*mls*), *MEX-algo* (Esteves et al., 2015), Simple Knowledge Organization System (SKOS) (Isaac and Summers, 2009), and Linked-USDL have been used or taken into account to compose the comprehensive schema of *dmcc-schema*.

The integration of different vocabularies within the Linked Data proposal is very important for the definition of semantic concepts in order to define services. Thus, the vocabulary offers the semantic glue that allows mere data to become meaningful data.

In the proposal that offers *dmcc-schema*, attempts to compose a simpler and more direct schema than other more general proposals. Linked-USDL proposal, which considers almost the entirety of a cloud service, does not consider the data mining part explicitly, as it seeks to be an open proposal to accommodate much of the spectrum of services. *dmcc-schema* combines various schemas and vocabularies to shape a very compact service definition. Our proposal is more concise to address the problem of defining these cloud services as it is based on the study of how different Internet providers and data mining platforms define these services with their specifications.

The Figure 1 depicts the core of *dmcc-schema* that has been developed, where the boxes represent the classes and the arrows represents properties, subclass relations and part-of relations. The color of the boxes represent the source vocabulary used with Linked Data proposal, such that green color is for the definition of *dmcc-schema*, blue for *GoodRelations* classes, orange for *SKOS* and magenta for *mls*. Class SLA contains parts from Linked-USDL (usdl-sla)

The definition of a service is done using *Turtle*. The schema in this formats is available for use from our laboratory website[3].

The aim of the cloud data mining service is to run data mining algorithms. Any data mining or machine learning algorithm can be included in *dmcc-schema*. This is so since we have sought to make as broad a definition as possible of the concepts of input, experimentation and output algorithms. This allows a wide range of algorithms to be supported[4].

4 PROOF OF CONCEPT

The approach we have developed is part of a platform for data mining services in the cloud that is being developed. This *dmcc* definition has been created with a cloud broker development in focus. The broker for cloud data mining requires a service definition that allows to overcome the different parameters, algorithms, input, experimentation, data, and models of all (or most) cloud services with these characteristics.

The reliability of *dmcc-schema* as a definition schema for the definition of data mining services in cloud computing is proven by using it within a proof-of-concept platform called *occml-platform*[5]. This platform works with all the details of a service definition built in *Turtle* or *JSON-LD* (Lanthaler and Gütl, 2012) and transforms all the definition information into a fully usable service. This means that it is possible to run the algorithms that have been included in the definition of the service from the cloud and use the service in its full scope.

Service definition is processed using *SPARQL* (Prud et al., 2006). This allows us to create the complete service and all the details of the algorithm catalogue it provides.

The catalogue published is directly consumable by agents and users who are demanding it. The platform, for the proof of concept, generates a REST API for interaction with the service and the selection of algorithms.

With *dmcc-schema*, the execution and results of catalogue algorithms can be fully defined. To address the differences that service providers have to classify and parameterize algorithms, the decision was made to follow the definition of the algorithms, according to the structure of programming environments such as R (Zhao, 2012), in terms of interfaces, datasets, parameter setting and data output, including models or datasets.

Part of the concepts and schemas from *MEXalgo* (Esteves et al., 2015) and *mls* have been integrated within the *dmcc-schema*, so they complement the definition of the schema for service definition. *MEXalgo* accurately defines a catalogue of Machine Learning and Artificial Intelligence algorithms. In *mls* a more comprehensive and general level is used for the definition of algorithms, allowing virtually any algorithm to be included and it is not as closely linked to the specification as *MEXalgo*.

Another advantage of using the proposed occml specification is the simplicity of creating the complete service. Creating a simple service that includes the K-Means (see Section 5) algorithm is done in a straightforward form.

5 A PRACTICAL EXAMPLE

For illustrative purposes we will describe in this section how some specific well-known techniques have been defined using *dmcc-schema*.

We will model a data mining cloud service with one algorithm as part of the services catalogue. Because of space considerations only part of the service will be defined.

[3]http://dicits.ugr.es/dmcc-schema/dmcc-schema.ttl

[4]The comprehensive list of algorithms can be found here: https://github.com/manuparra/ml_dm_taxonomy

[5]OCCML on DiCITS: dicits.ugr.es/occml/

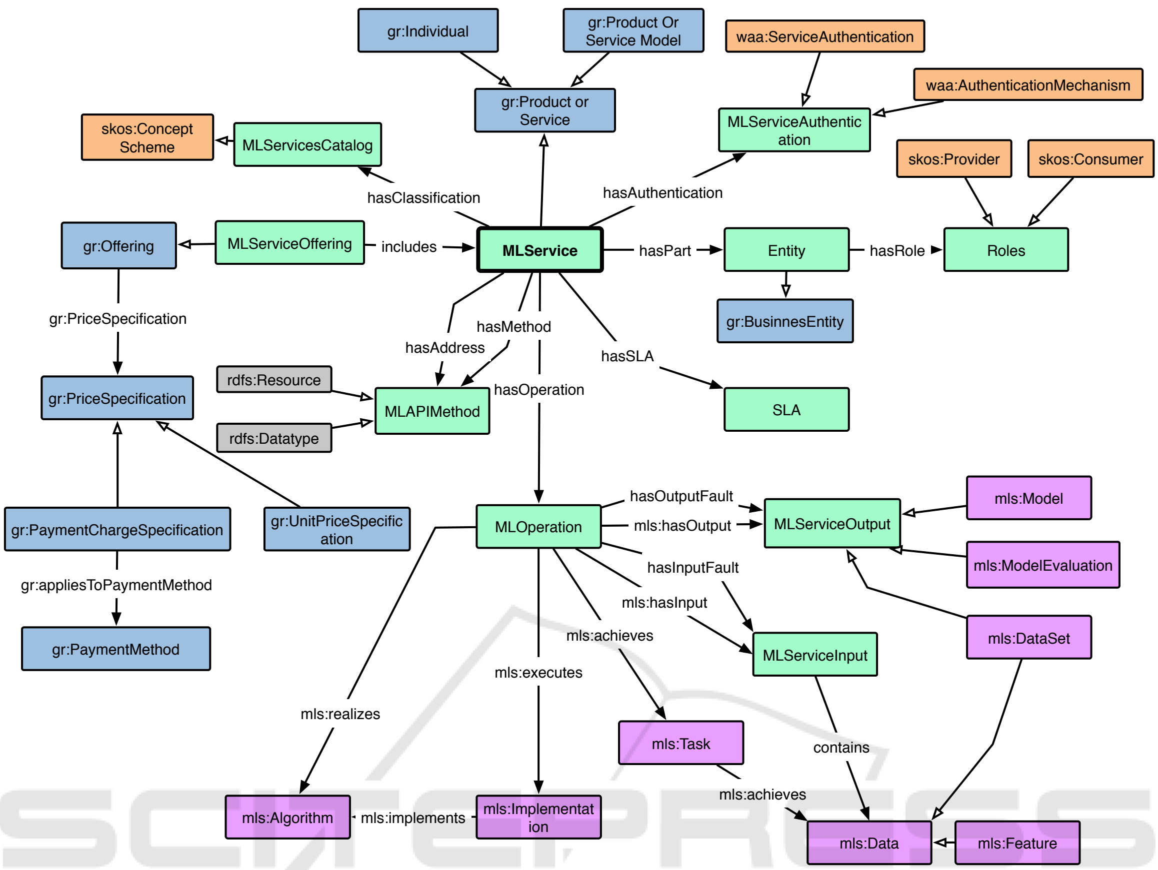

Figure 1: Classes, relations and vocabularies for *dmcc-schema*.

The service description has been done using the *Turtle* language. The first part to be defined corresponds to the class `dmcc:MLService` that provides the entry point for service description. A service is created within the catalog with the name `KMeans_Service`, thus associating a Data Mining algorithm with the service and specifying the service attributes, in addition to listing the interaction points provided by the service as seen in listing 1.

```
dmcc:KMeans_Service
 a dmcc:MLService ;
   dcterms:created "2017-04-20" ;
   dcterms:creator "Manuel Parra";
   dcterms:description
     "Perform k-means clustering" ;
   dcterms:modified "2017-05-04" ;
   dcterms:publisher "DICITS_ML" ;
   [...]
   .
```

Listing 1: New service definition.

In this case the new service requires authentication, for which it is necessary to include `dmcc:hasAuthentication` and `dmcc:KMeans_Auth` as indicated in the listing 2.

```
dmcc:KMeans_Service
  a dmcc:MLService ;
    dmcc:hasAuthentication
      dmcc:KMeans_Auth ;
    dmcc:hasOperation
      dmcc:KMeans_Operation ;
    [...]
    .
```

Listing 2: New service definition for KMeans.

In addition, as it indicated in the *dmcc* diagram 1, cloud provider and consumer require `dmcc: MLServiceOffering` ant it can be included to define prices and costs of the service, `dmcc: MLServicesCatalog` for the catalogue of services, `dmcc: MLApiMethod` for the points of interaction with the service and `dmcc:SLA` for detailing aspects of the licensing.

For each of the data mining services that are added, it is necessary to indicate the operation involved, so `dmcc:hasOperation` `dmcc:KMeans_Operation` is used as shown in listing 2. In the listing 3 the operation it performs includes information regarding the input parameters (lines 10-11) of the algorithm, data set (lines 3-4), output and algorithm to be executed (lines 12-13) the composition of the operation that requires an algorithm.

```
1  dmcc:KMeans_Operation
2    a dmcc:MLOperation ;
3     dmcc:hasInputParameters
4       dmcc:KMeans_InputParameters;
5     dcterms:description
6      "Operation performing
7         the service";
8     mls:executes
9      mls:KMeans_Implementation;
10    mls:hasInput
11     dmcc:KMeans_Input;
12    mls:hasOutput
13     dmcc:KMeans_Output .
```

Listing 3: Operations for the algorithm.

The input and output data of the algorithms must be included in the definition of the data mining operation to be performed. The input of data, which can be parameters `dmcc:KMeans_InputParameters` or datasets `dmcc:KMeans_Input`.

Input parameters of the algorithm `dmcc:MLServiceInputParameters` and the parameter list `dmcc:parameter_01, [...]` is shown in the listing 4.

```
dmcc:KMeans_InputParameters
  a dmcc:MLServiceInputParameters ;
    dmcc:Parameters
      dmcc:response_parameter_01,
      [...]
    dcterms:description
            "Input Parameters" ;
    dcterms:title "Input" .
```

Listing 4: Input parameters definition.

Definition of `dmcc:hasInputParameters dmcc:KMeans_InputParameters ;` allows you to specify the general input parameters of the algorithm. For example for K-Means `dcterms:title "centers"` (number of centers of the K-Means), as well as whether `dmcc:mandatory "false" ;` is mandatory and its default value, if it exists. The listing 5 shows the definition of one of the parameters `parameter_01` . The other algorithm parameters are defined in the same way.

```
dmcc:parameter_01
  a dmcc:MLServiceInputParameter ;
    dmcc:defaultvalue "3" ;
    dmcc:mandatory "true" ;
    dcterms:description
      "Either the number of clusters,
       or a set of initial cluster
       centres" ;
    dcterms:title "centers" .
```

Listing 5: Example of a parameter and features.

An `mls:Model` model and an evaluation of the `mls:ModelEvaluation` model have been considered for specifying the results of the K-Means service execution in `mls:hasOutput dmcc:KMeans_Output`. Model evaluation is the specific results if the algorithm returns a value or set of values. When the service algorithm is preprocessing the result is a dataset. For the model you have to define for example whether the results are PMML (Guazzelli et al., 2009) for example `dmcc:KMeans_Model a dmcc:PMML_Model ;` as shown in listing 6.

```
dmcc:KMeans_Model
  a dmcc:PMML_Model ;
    dmcc:storagebucket
      "dicits://models/" ;
    dcterms:description
      "PMML model" ;
    dcterms:title "PMML Model" .
```

Listing 6: PMML Model of the service output.

6 CONCLUSIONS

Due to the high and increasing demand and usage of data mining services, there is a clear need of suitable tools for data mining cloud services definition. In this work we have introduced *dmcc-schema*, a schema and vocabulary for defining data mining services in cloud computing using a Linked Data approach.

The description of data mining services is one of the most critical components since it is a vital element in communication and trading with different cloud service providers.

This enables services to be designed to execute data mining algorithms, considering the key aspects of a cloud service included in the schema. The definition implemented with *dmcc-schema*, offers the basic capabilities to define a cloud data mining service, taking into account authentication, pricing, licensing, catalog and interaction interfaces, among other aspects.

The main advantage of using *dmcc-schema* is that it greatly simplifies the design of a cloud service focused on data mining. This is because it unifies two environments: the cloud computing and services aspects, and the execution of data mining algorithms. Allowing key elements of this type of services such as the pricing of the execution, the storage costs of a dataset, or the authentication of the service.

In addition, our schema is being deployed on an open platform that will offer Big Data data mining services on demand, as validated in the proof of concept in the section 3.

As future work, we are working on expanding the definitions of the algorithms supported to include methodologies that allow us to include services such as Deep Learning techniques and time series processing.

ACKNOWLEDGEMENTS

Manuel Parra-Royon holds a "Excelencia" scholarship from the Regional Government of Andalusia (Spain). This work was supported by the Research Projects *P12-TIC-2958*, *TIN2013-47210-P* and *TIN2016-81113-R* (Ministry of Economy, Industry and Competitiveness - Government of Spain).

REFERENCES

Adida, B., Birbeck, M., McCarron, S., and Pemberton, S. (2008). Rdfa in xhtml: Syntax and processing. *Recommendation, W3C*, 7.

Allemang, D. and Hendler, J. (2011). *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier.

Beckett, D., Berners-Lee, T., Prudhommeaux, E., and Carothers, G. (2011). Turtle–terse rdf triple language. 2011. *URL http://www. w3. org/TeamSubmission/turtle*.

Bellwood, T., Clément, L., Ehnebuske, D., Hately, A., Hondo, M., Husband, Y. L., Januszewski, K., Lee, S., McKee, B., Munter, J., et al. (2002). Uddi version 3.0. *Published specification, Oasis*, 5:16–18.

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, 284(5):28–37.

Bizer, C. (2009). The emerging web of linked data. *IEEE intelligent systems*, 24(5).

Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). Web Service Definition Language (WSDL). Technical report, World Wide Web Consortium.

Elvesæter, B., Panfilenko, D., Jacobi, S., and Hahn, C. (2010). Aligning business and it models in service-oriented architectures using bpmn and soaml. In *Proceedings of the First International Workshop on Model-Driven Interoperability*, pages 61–68. ACM.

Esteves, D., Lawrynowicz, A., Panov, P., Soldatova, L., Soru, T., and Vanschoren, J. (2016). ML Schema Core Specification. Technical report, World Wide Web Consortium.

Esteves, D., Moussallem, D., Neto, C. B., Soru, T., Usbeck, R., Ackermann, M., and Lehmann, J. (2015). Mex vocabulary: a lightweight interchange format for machine learning experiments. In *Proceedings of the 11th International Conference on Semantic Systems*, pages 169–176. ACM.

Guazzelli, A., Zeller, M., Lin, W.-C., Williams, G., et al. (2009). Pmml: An open standard for sharing models. *The R Journal*, 1(1):60–65.

Hepp, M. (2008). Goodrelations: An ontology for describing products and services offers on the web. *Knowledge Engineering: Practice and Patterns*, pages 329–346.

Isaac, A. and Summers, E. (2009). Skos simple knowledge organization system. *Primer, World Wide Web Consortium (W3C)*.

Kona, S., Bansal, A., Simon, L., Mallya, A., Gupta, G., and Hite, T. D. (2009). Usdl: A service-semantics description language for automatic service discovery and composition1. *International Journal of Web Services Research*, 6(1):20.

Kopeckỳ, J., Vitvar, T., Bournez, C., and Farrell, J. (2007). Sawsdl: Semantic annotations for wsdl and xml schema. *IEEE Internet Computing*, (6).

Kopeckỳ, J., Vitvar, T., Fensel, D., and Gomadam, K. (2009). hrests & microwsmo. *STI International, Tech. Rep*.

Lanthaler, M. and Gütl, C. (2012). On using json-ld to create evolvable restful services. In *Proceedings of the Third International Workshop on RESTful Design*, pages 25–32. ACM.

Maleshkova, M., Pedrinaci, C., and Domingue, J. (2009). Supporting the creation of semantic restful service descriptions. *8th International Semantic Web Conference (ISWC 2009)*.

Maleshkova, M., Pedrinaci, C., Domingue, J., Alvaro, G., and Martinez, I. (2010). Using semantics for automating the authentication of web apis. *The Semantic Web–ISWC 2010*, pages 534–549.

Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., et al. (2004). Owl-s: Semantic markup for web services. *W3C member submission*, 22:2007–04.

Newcomer, E. and Lomow, G. (2005). *Understanding SOA with Web services*. Addison-Wesley.

Panov, P., Džeroski, S., and Soldatova, L. (2008). Ontodm: An ontology of data mining. In *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*, pages 752–760. IEEE.

Pedrinaci, C., Cardoso, J., and Leidig, T. (2014). Linked usdl: a vocabulary for web-scale service trading. In *European Semantic Web Conference*, pages 68–82. Springer.

Prud, E., Seaborne, A., et al. (2006). Sparql query language for rdf.

Vanschoren, J. and Soldatova, L. (2010). Exposé: An ontology for data mining experiments. In *International workshop on third generation data mining: Towards service-oriented knowledge discovery (SoKD-2010)*, pages 31–46.

Vanschoren, J., Van Rijn, J. N., Bischl, B., and Torgo, L. (2014). Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60.

Weibel, S., Kunze, J., Lagoze, C., and Wolf, M. (1998). Dublin core metadata for resource discovery. Technical report, Network Working Group.

Zhao, Y. (2012). *R and Data Mining: Examples and Case Studies*. Elsevier Science.