Ontology of sands: design and implementation of an OWL ontology for sand samples

José Ricardo Ferro[1]
[1]Instituto Superior Técnico, Universidade de Lisboa, Portugal

The web we know today is evolving. The semantic web and linked data principles aim to change the way we use computers on the web, by enabling them to not only display the data but also understand its meaning. By understanding the semantics of the data, computers can process it in much more meaningful ways and dynamically connect different data sources, creating a global web of data.

In the sedimentology domain, various databases exist, containing data resulting from the analysis of sediment samples, that can be used mainly for research purposes. However, these databases are not related in any way, restricting researchers to smaller datasets and losing any expressiveness that could come from correlating samples from different data sets.

In this work, we turn to the capabilities of the semantic web as a way of solving this problem. We describe the development of an ontology that intends to represent sand samples and their main characteristics. The purpose of this ontology is to provide a standard set of terms that enable applications in the area of sedimentology to expose their data in the semantic web, ultimately allowing them to relate data between each other seamlessly.

This work starts with a general overview of the current web service standards and semantic web technologies. This overview also covered the related work in the area of sedimentology, as well as other relevant projects on the semantic web. Next, the development of the ontology itself is described, detailing the central concepts and properties that compose the final vocabulary. Finally, the ontology is validated by relating data from different data sources using the terms described in it.

Index Terms—**Semantic Web, Sedimentology, Linked Data, Ontology.**

I. Introduction

NOWADAYS, with the evolution of media and communication technologies, the world wide web inevitably became one of the most important sources of data. However, information on the web is mostly targeted for humans. We can use popular search engines as a way of finding web pages that fit our research criteria and then take advantage of popular formats such as tabular data and written documents to retrieve the necessary information from the web. It is only after retrieving all these documents that we manually extract the useful information and combine everything to make valid conclusions.

If we take a closer look at the traditional research flow that was just described, we can see that apart from search engines, computers are used mostly for routine processing as well as parsing web pages in human-readable formats. Computers commonly discard the actual semantics of the data and leave it for humans to analyze while browsing the web. Finding a way for computers to understand the data instead of just displaying it would be a significant advantage for any researcher. It was in this vision that Tim Berners-Lee [1], [2] introduced the concept of the semantic web - as an extension of the current web - where data would be given well-defined meaning through clear and structured metadata. This would enable computers to not only display the information on the web but also process it in a meaningful way, helping humans to draw important conclusions in a fraction of the time.

One of the most promising things about the semantic web is it is potential to allow several databases to relate to each other on overlapping concepts. For example, if we have two databases that use the same identifier for the same concept, such as a location, it could be possible to relate these two and enable programs to combine information across both databases. This would allow the development of software that would be tailored to the user's needs. Following the previous example, if we have data about sediment samples on website A and extreme events on website B, if both databases use the same identifier for the same location, it would be possible to automatically correlate this data, seamlessly.

The primary goal of the semantic web is to open the extensive amount of already available data on the web to software processing, whilst enabling this data to be interlinked. However, there is still a long way to go in order to take advantage of the full potential of the semantic web. Most of the data on the web is stored in relational databases that are based on open standards - such as Structured Query Language (SQL) - that are not compatible with the semantic web. The structure of these databases is fixed, and the software that processes the data is usually designed and implemented for a single database schema. Additionally, while the meaning of the data is fairly easy to understand for a human, there is no explicit way for computers to understand it. This makes the integration of different data sources much more difficult. To change this, it is necessary for this data to be published in specific semantic formats. The *linked data* initiative [3], [2] suggests a set of best practices for publishing data on the semantic web, emphasizing the need for large-scale integration of data on the Web, thus creating a global network of data.

While the semantic web helps in combining information from different data sources, the amount of data available on the web today calls for a scalable solution. Understanding the

Corresponding author: José Ricardo Ferro
email: ricardo.carrapico.ferro@ist.utl.pt

specific terminology of a few data sources and integrating them is a manageable task even for a human. However, connect hundreds of data sources is not a task to be done manually. It becomes clear that it is essential to have a public vocabulary that can be used by all data sources to describe their own data, making sure that the same concept is described in precisely the same way in every one of them [4]. This allows programs to integrate all sources of data that implement the same vocabulary regardless of the internal structure of each one. These vocabularies are called ontologies and are very powerful as tools to help resolve heterogeneity problems among data sources [5]. In a more technical approach, an ontology is a collection of machine-readable terms that explicitly establishes concepts and their attributes, as well as relationships between them [6]. In theory, a single vocabulary can represent an entire knowledge domain which makes this technology very promising for integrating data about different fields and dynamically drawing complex conclusions.

In the sedimentology domain, the sample databases that exist are not connected, making it more difficult for researchers to correlate data among different data sources. The semantic web, together with the appropriate ontologies, could provide the necessary tools to change this situation.

In this work, we describe the development of an ontology within the sedimentology domain, more specifically sand samples and their analysis. The primary focus was to implement a way of retrieving data from different data sources, whether they are databases from colleges, institutions or private collections. With the semantic web and linked data concepts in mind, the ontology was developed in a way that the central concepts and properties could be compatible with other data sources. After developing the ontology, we also developed a simple online platform that could help us test the ontology. By preparing the web interface to be able to read any data source that implements this ontology, we allow multiple databases to integrate into the main website by merely exposing their data according to this ontology. As most databases today are relational ones, another problem occurred when publishing data as linked data. The software that processes relational data is designed for a particular database schema, making it incompatible with linked data protocols. With this in mind, and to further improve our work, we explain how a traditional relational database can be exposed as linked data using the already available software. The developed method was applied to a simple relational database, mapping the data according to the previously developed ontology, which made it possible for the website to display linked data stored in the database.

In the following section, we analyze the pros and cons of the current Web Service technologies. The Semantic Web and Linked Data concepts are also explained in a more detailed way. In Section 3, we explain the design and implementation of the developed ontology. Later, Section 4 describes how the ontology was evaluated and integrated with the website.

II. Web Services and Semantic Web

A. Web services

Sharing information between machines on the internet is not new. As the internet evolved and the demand for data increased, allowing machines to communicate and retrieve information from other machines became more and more important. For this purpose, programmers all over the internet implemented Web Services as a way of exposing data to external systems. According to the World Wide Web Consortium (W3C) [7], a web service is a software system designed to support interoperable machine-to-machine interaction over a network.

Web Services became a popular way of sharing data between systems, since they made it possible for different systems to integrate, regardless of the programming language or operating system of each one. To communicate, machines need to implement the same communication protocol. Two of the main technologies are Simple Object Access Protocol (SOAP) [8] and Representational State Transfer (REST). SOAP web services use eXtensible Markup Language (XML) [9] to structure the messages along a Web Service Definition Language (WSDL) [10] document that is used to describe the web service. All messages sent to a SOAP web service are verified to determine if the message's structure respects the one defined in the WSDL document. Unlike SOAP, REST is not a messaging protocol but an architectural style. REST defines a set of architectural principles that allow for the creation of web services that are lightweight, yet powerful and scalable. REST web services are stateless, which means that no sessions are kept server-side, improving overall performance. Additionally, a client must have the necessary information to delete or change the state of a resource by simply holding its representation. Principles like these make REST web servers more flexible and easy to integrate then SOAP, allowing data transfer between client and server without defining many standards.

The use of web services brings many advantages for systems on the internet. For instance, they allow for simplified and coherent communication between machines with possibly very different characteristics. However, most web services on the internet publish their data following strict, self-developed data structures that make it very difficult for machines to extract any meaning from the data automatically. Usually, when communicating with a web service, it is necessary to study the associated documentation to find out how the data is exposed and organized. This lack of flexibility makes it hard for systems that share information about the same topic to integrate with one another, often requiring custom implementation from both systems. These issues are the reason that motivated the development of the semantic web in the first place.

B. The Semantic Web and Linked Data

Although machines can display the data on the web, they cannot process its meaning. In an attempt to evolve the internet beyond this point, Tim Berners-Lee [1], [2] introduced the concept of The Semantic Web. The primary focus is to have the data on the web published in semantic formats, enabling software to process it. This is done by adding metadata to the webpages as a way of describing what the data is, thus making it machine-readable. This allows for links between data sources to be created in much more meaningful ways.

To make this possible, it is essential to also understand the concept of Linked Data, one of the core movements of the Semantic Web. Linked data is a set of design principles for sharing data on the web that were introduced by Tim Berners-Lee [3]. These set of principles were developed as a way of helping the large-scale integration of data sources on the web.

The Semantic Web is built on top of two core technologies that are already available on the current web. In the first place, XML is used to add structure to a resource. By creating meaningful tags and labels to annotate content within a webpage, we allow specialized software to process this data in complex ways. However, although XML allows users to add structure to their web pages and documents, it does not describe the data's meaning. This is added by the second technology in the core of the Semantic Web, the Resource Description Framework (RDF) [11].

The RDF data model provides an abstract way of encoding meaning in a way that allows machines to understand and process. This data model is based on the idea that resources can be seen as nodes of a directed graph and connections between resources translate to connections between nodes. This model eases data manipulation and processing for applications.

The basic element of RDF is a triple, which is composed of three parts: subject, predicate, and object. If we imagine the graph described above, each statement can be illustrated by two nodes (subject and object) connected by a directed arc (predicate), as seen in Figure1. Many times, it is also said that in a triple, the object is a property of the subject.

Fig. 1. Graph visualization of an RDF triple.

By decomposing the data about a resource into sets of simple sentences that follow this pattern, we make it possible for meaning to be encoded in RDF and consequently read by computers. Additionally, as the object of a triple can also be the subject of another triple, computers can create webs of information about related things.

For example, we can relate data about different sediment samples using the RDF data model, as seen in Figure 2

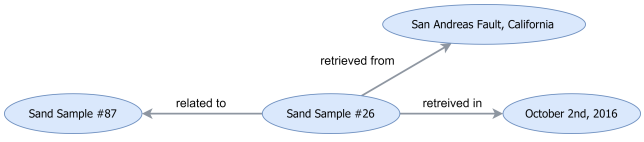

Fig. 2. Connection between resources using the RDF data model.

In the Semantic Web, each term that makes up a triple is identified by a Uniform Resource Identifier (URI). This ensures that each concept has a unique identifier and allows users to define new resources and predicates by simply creating the respective URI somewhere on the web. This concept is key to understanding the semantic web. Ideally, if we have data about the same resource in multiple data sources, all these data sources would identify this resource using the same URI, making it possible to create links between data sources in much more meaningful ways.

After defining the structure, meaning, and identity of the data, the next step is to understand how data from different databases can be linked using ontologies. The Semantic Web uses the Web Ontology Language (OWL) [12] to represent the structure of a knowledge domain. OWL is built on top of RDF Schema (RDFS) [13], which provides the necessary set of elements needed for the description of an ontology. An ontology defines classes, properties and relationships between them in a conceptual way.

It is essential to understand that in the definition of an ontology only the concepts are described, not the data itself. For instance, if we consider an ontology about geography, the class "city" should be included but terms like "London" and "Paris" should not, as they are instances of the "city" class.

RDF data is queried using a special language called Simple Protocol and RDF Query Language (SPARQL) prud2006sparql. In its essence, a SPARQL query is able to retrieve and manipulate data by matching a series of triple patterns against the database, retrieving the data that satisfies all constraints. These patterns consist of a set of triple expression and logical and/or operators.

C. Related work

In the Earth Sciences field, some vocabularies already exist that allow data to be connected on common concepts. Whether standalone or an extension of existing ones, different knowledge domains are described in ontologies that can be reused and linked together. By looking at these ontologies, it is intuitive to split them into two categories: general ontologies and domain ontologies. The first one includes concepts that are common to multiple knowledge domains. The second one is more focused on a single field, with more specific terms and properties.

The Semantic Web for Earth and Environmental Terminology (SWEET) [14], [15] is an investigation by the California Institute of Technology that aims to improve discovery and use of Earth science data through the semantic web. SWEET developed an ontology that has extensive coverage of the earth sciences. Several child ontologies were developed to describe the many domains of Earth Sciences, resulting in vocabularies describing oceans, land, and atmosphere, among others. General concepts like time, space and matter are also included. At the official website, SWEET included an interactive graph that enables users to browse through the terms that make up the ontologies. At this moment, these ontologies include general terms like *Sediment* and *Sand*, as well as types of minerals and sediments. However, the characteristics of these are not included.

Another general ontology is The North American Geologic-map Ontology [16], developed by the North American Geologic Map Data Model Steering Committee (NADM). This ontology's primary aim is to provide a data model for the description, classification, and interpretation of geological features. The model's main focus is the description of geoscience concepts and relationships that are somehow related to

geologic maps, including terms like *Mineral* and *GeologicAge*, *Particle Geometry* and *Composition*. These terms are all used to generally describe the data, without much specificity.

The different domain ontologies that exist describe more specific fields. There is one ontology aimed at rock classification [17], as well as an attempt to develop a domain ontology for Structural Geology [18]. Additionally, an ontology on Fractures was developed [19], with the later integration with the SWEET ontology in scope. Also in the Earth Sciences field, Tripathi [20] developed a modular Hydrogeology ontology by also extending the SWEET ontology, describing the different steps that were taken to accomplish a valid extension.

The Commission for the Management and Application of Geoscience Information [**?**] developed an ontology describing the geological age [21], including terms like *Eon* and *Epoch* as well as all their corresponding subdivisions. Additionally, a JSON formatted ontology on the same topic was developed [22], associated with a regional geologic age visualization of North America.

The British Geological Survey (BGS) developed a series of vocabularies to provide consistency across their classification systems. By creating these dictionaries, the BGS provides standardized terms to better control the description of scientific observations. Among the available ones, we can find vocabularies about rock classifications, sample colors, dating methods and mineral names. Additionally, in the sedimentology domain, vocabularies on grain shape, size, and sorting are also available.

D. GeoNames

GeoNames [23] is an open geographical database that aims to help in geo-referencing resources on the semantic web. This database contains linked data on millions of resources that are categorized into features like, countries, regions, cities, amenities and points of interest, among others. Each one of these resources is identified by a URI and is described using the RDF data model. The format of the data follows the *GeoNames* ontology [24].

Data from the *GeoNames* database can be accessed by downloading one of the daily-updated snapshots of the database or though a series of web services. Although there is no official SPARQL endpoint available, the fact that it is possible to download the whole set of data has encouraged several web services on the internet tho publish these snapshots as linked data using SPARQL endpoints.

III. Ontology for sedimentology data

A. Ontology design

In sedimentology, sediments are divided into different categories regarding the size of the particles that compose the sample. This project was aimed at sands [25] (sediments with particles between 0.0625mm and 2mm in diameter) and their main characteristics, such as geological age, color, texture, composition, and morphoscopy. We also took into account the main analysis methods used when studying and classifying sand samples.

The primary goal when developing an ontology was to provide a simple, yet meaningful way of describing a sand sample in a machine-readable format so that samples from different data sources could be shared. Ultimately, this ontology could provide a standard set of terms to be used when implementing applications related to the sedimentology domain, such as a global sample search engine that would include databases from institutions and private collections, among others.

1) Main classes

The ontology is divided into a series of classes and properties that reflect the characteristics of sands, as stated previously. As some concepts were common to multiple classes, an attempt was made to describe these concepts as generally as possible. This way, the corresponding classes could be reused and included in the description of other classes without the need to redefine concepts. In this ontology, all the classes are disjoint, as no individuals can be an instance of more than one of these classes.

The root of the ontology is the **Sample** class. Each sample is represented by an instance of this class. As we can see in Figure 3, the *Sample* class has some data properties that are are used to contextualize the sample within an application. These properties are *creator*, *title*, *description*, and *date*. In order to represent a sample in a way that other applications could process, these properties are related to their equivalent counterparts in the Dublin Core vocabulary using the *equivalentProperty* term of the OWL vocabulary.

To provide consistency in the data, the cardinality features of OWL were used to restrict the instances of the *Sample* class to have a maximum cardinality of 1 for *creator*, *title*, and *date*. Additionally, the *url* property can be used to indicate the location of a webpage with further information about the sample. The *Location* where the sample was retrieved is associated with this class, as well as a set of *Document* and *Photo* instances that may be relevant. The *environment* data property identifies the environment of the sample.

Finally, the *Sample* class is associated with other classes that describe the main characteristics of a sample. These classes are *GeologicalAge*, *Texture*, *Composition*, *Morphoscopy* and *Color*.

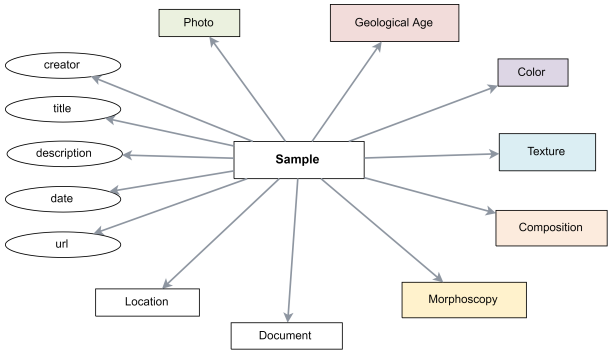

Fig. 3. *Sample* class.

The **Analysis Method** represents the different experiments and analysis that can be done to study a sample. The *name*, *description*, and *equipment* data properties give a general

context of the experiment while the *Document* and *Photo* classes help support the used protocol with useful attachments, as shown in Figure 4.

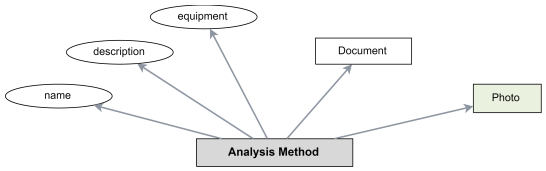

Fig. 4. Analysis Method class.

The **Document** class represents a document such as a paper, report, or protocol that may be useful to complement information on other resources. The *file_name*, *file_extension* and *url* provide any application with the technical information needed to display the document or a link to it's location. An additional *description* can be provided to add information on the document, such as a summary of the content. We can see the graph representation of this class in Figure 5.

As many other ways of identifying documents already exist, instead of an instance of this class it is also possible to describe documents using other means, like the International Standard Book Number (ISBN) and Digital Object Identifiers (DOI).

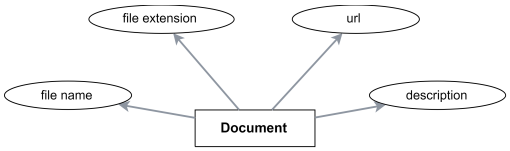

Fig. 5. Document class.

The **Photo** class describes an image that can be used to illustrate instances of the *Sample* or *AnalysisMethod* classes. Just as in the *Document* class, the technical parameters that allow an application to display the image are provided. Additionally, a short summary may be specified in the *description*. The *zoom* property allows the user to further specify the microscope settings used when the image of the sample was taken. When photographing samples, it is also important to specify the *scale* of the image. This is necessary as a reference to evaluate the size of the grains.

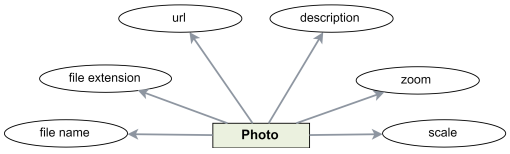

Fig. 6. Photo class.

The **Location** class indicates where a *Sample* was retrieved at. As we can see in Figure 7, the *longitude* and *latitude* properties provide the precise coordinates and the *geodetic_datum* property specifies the coordinate system used. In some cases a sample may be retrieved from underwater or elevations. The *height* and *depth* properties provide a way to specify additional information on these cases.

In order to be able to relate the location data of a sample with GeoNames resources, the *Location* was annotated as being equivalent to the *GeonamesFeature* class from the *GeoNames* ontology. This is the class that represents all the features provided by the GeoNames vocabulary. using the *equivalentClass* OWL term.

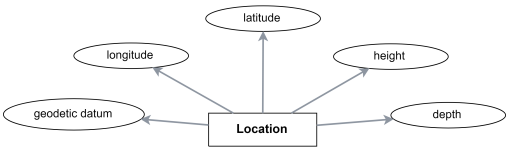

Fig. 7. Location class.

The **Classification** class is used to classify a certain characteristic of a sample. In Figure 8 we can see the structure of this class. The *classification_attribute* is used to specify the characteristic that the classification refers to, such as grain size distribution or grain sorting. Some characteristics are classified in different ways depending on the system that is used. With this in mind, the *classification_system* property was added to allow for the publisher to specify which system was used to classify a certain characteristic of a sample.

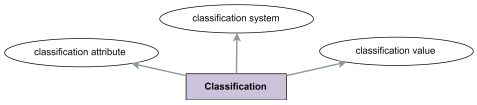

Fig. 8. Classification class.

The **Texture** class is used to describe the texture of a *Sample* which consists of an analysis of the particle size distribution of the sample, also known as a granulometry. We can see this class in Figure 9. Particle size ranges are divided in classes according to the Krumbein ϕ (Phi) scale [26]. Essentially, a granulometry consists of a series of index-value entries that state the particle distribution (value) within each size class (index). The *Granulometry* class allows the user to add a list of *Phi* measurements by specifying the *index* and *value* of each measurement. The *Granulometry* also allows the user to define the percentages of fine and coarse articles in a sand sample, using the *percentage coarse* and *percentage fine* properties. To provide additional information, an *Analysis Method* class can also be associated with the granulometry. Once the granulometry ϕ values are defined it is possible to calculate different statistical parameters that can be used to further analyze the texture of the sample. Finally, the *Texture* can be associated to *Classification* instances in order to provide a final classification of the sample regarding it's texture.

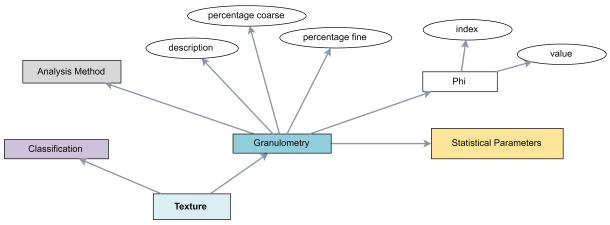

Fig. 9. Texture class.

The **Statistical Parameters** class represents a statistical parameter resultant of a granulometry. The *Mean*, *Standard*

Deviation, *Skewness* and *Kurtosis* classes represent the different possible parameters and are sub-classes of the main *Statistical Parameters* class, as shown in Figure 10. This means that all four of these classes inherit the data properties from their parent class. In order to take into account that each statistical parameter can be calculated using different methods (with different accuracies), the *method* and *value* properties were added to the *Statistical Parameters* class.

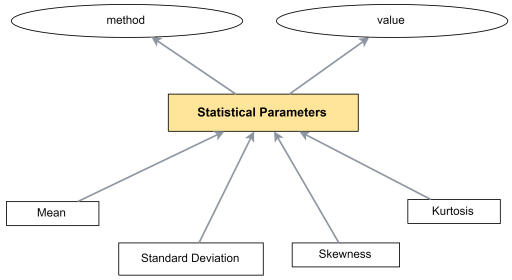

Fig. 10. Statistical Parameters class.

The **Geological Age** class describes the geological age of a sample [27]. The *eon*, *era*, *period*, *sub-period* and *age* properties describe the relative time period whereas the *approximation* provides the actual determined result. The analysis method used to determine the geological age of the sample can also be described using an instance of the *Analysis Method* class. In Figure 11 we can see the graph representation of the class.

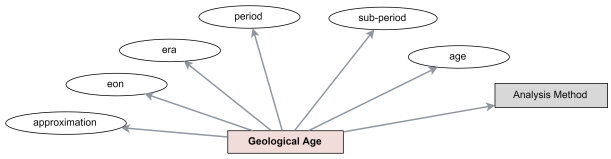

Fig. 11. Geological Age class.

The **Color** class describes the color of a sample. As we can see in Figure 12, the *name* and *description* properties are used to describe the color in a simple way. Additionally, it is possible to describe the color of the sample using the Munsell Color system [28], which is represented by the *MunsellColor* class. In order to illustrate the color, images can be associated using the *Photo* class.

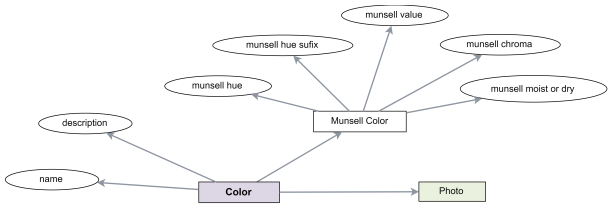

Fig. 12. Color class.

The **Morphoscopy** class represents a morphoscopy of a sample, which consists of a study of the shape of the particles that make up the sample. A description of the morphoscopy can be added using the *description* data property, while *Photo* instances can be used to illustrate it. In Figure 13 we can see a simple graph of this class. The **Composition** class is used to describe the different components that make up the sample. This class has different subclasses that describe each of the main components in its own way. The *Carbonates*, *OrganicMatter*, *PH*, *CoarseFraction* and *HeavyMinerals* classes are associated with the relevant fields that describe each of them. Each of these inherit the *AnalysisMethod* from the main *Composition* class.

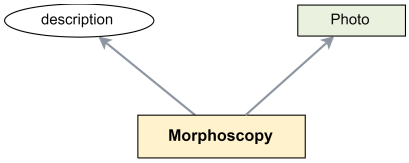

Fig. 13. Morphoscopy class.

As we can see in Figure 14, the *Photo* and *Classification* are used throughout each of the classes in order to fully describe the composition of the sample.

In the case of the *OrganicMatter* class, the percentage of organic matter can be determined with a series of measurements with different duration and temperature parameters. For this, it is possible to associate several instances of the *OrganicMatterMeasurement* class to a single *OrganicMatter* instance.

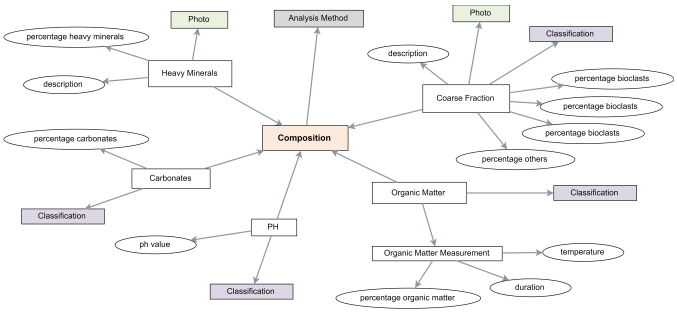

Fig. 14. Composition class.

2) Dublin Core

The Dublin Core Metadata Initiative (DCMI) [29] is an open organization that has developed a simple vocabulary to generally describe resources on the web, whether they are documents, web pages, images, or any other kind of resource.

The main goal of the Dublin Core vocabulary is to provide users with a way to add simple, yet meaningful metadata to their content in a way that is widely recognized on the web. By providing users with terms like *Creator*, *Date* and *Title* the DCMI creates a set of terms that serve as a baseline to generally contextualize any data on the web.

On the developed ontology, the properties *creator*, *title*, *description* and *date* of the *Sample* are equivalent to those of the Dublin Core vocabulary. This was done to represent a sample in a way that other applications could process, maximizing the possibilities of sharing resources among different systems.

B. Ontology implementation

The ontology was implemented using the open-source Protégé ontology editor [30]. The validity and consistency of

the ontology were assured using the HermiT reasoner [31], which is included as default in the editor.

The ontology was developed using the OWL Lite format, a lighter version of OWL that includes all the features that where used. This included property annotation and restrictions. OWL Lite also provides cardinality constraints of 0 or 1. Since this was also the maximum cardinality needed for the ontology, there was no need to evolve to more complete versions of OWL.

In the actual ontology, class and property names are according to the following format: class names are a composition of all the words together, with the first letter of each word capitalized (for example, the class *Analysis Method* is written as *AnalysisMethod*). Object properties follow the same logic as classes, except that the first word is lowercase (for example, the property *has granulometry* is written as *hasGranulometry*. Property names are a composition of all the words in lowercase and separated by the underscore sign (for example, the property *file extension* is written as *file_extension*).

The ontology, in OWL format, is available at:

http://web.ist.utl.pt/ist172870/thesis/sands.owl

IV. Evaluation

The main goal of the ontology was to provide a way to unify data about sands across the semantic web. Ultimately, different databases could use this ontology to expose their data in machine-readable formats. With this in mind, and to test the ontology, a client application was developed to serve as a centralized access point for linked data about sands. This application consisted of a simple website with a map displaying the samples in their respective locations. The goal of this application was to provide a simple web interface that could act as an information hub and display data about samples from multiple databases in a seamless way. To ensure a general implementation and facilitate the integration of different data sources, the application was developed to be able to read linked data from any endpoint that follows the developed ontology. This allows the application to integrate with different databases effortlessly as long as the data structure respects the ontology.

Since most data on the web is stored in relational databases, it is essential to fully understand how an application can consume linked data from such a database. For this purpose, a database with sand samples data was created and published as linked data to be integrated within the developed website. The idea was that samples could be added to the database through a web interface and then published as linked data, respecting the ontology. This data would then be read and displayed on the website. By doing this, we can study not only how the ontology allows an application to read data from any existing databases but also how institutions can have their data integrated with other systems.

In Figure 15 we can see the architecture of the project that was implemented.

As we can see, the main components of the system are a relational database, a D2R Server, and a client application. The database holds the sample data and can be queried using simple SQL queries. The role of the D2R Server is to transform the relational data into linked data that is read by the application using a SPARQL endpoint.

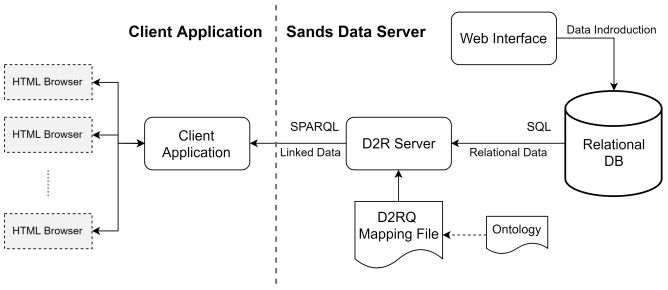

Fig. 15. Project architecture.

The website that served as a client application was developed using the *Python 2.7* programming language and the *Django Web Framework 1.11.6* as a web server. Additionally, the library used to enable the application to query the SPARQL endpoints was *SPARQLWrapper*. The database was implemented using *PostgreSQL*.

A. Publishing relational data as linked data

In order to implement the SPARQL endpoint, an additional effort was made to understand how a relational database can be exposed as linked data. For this, the method described by Bizer & Cyganiak [32] and Bizer & Seaborne [33] was applied to the second database using a D2R server and D2RQ mappings [34]. The D2R server enables browsers to navigate through non-RDF databases and allows applications to query a database using the SPARQL language. In order to achieve this, the D2R server uses the D2RQ mapping language to specify how a database schema maps to an RDF schema or OWL ontology. Essentially, this mapping explicitly states which tables and columns represent certain classes and properties of the ontology. The D2RQ mapping file can be generated automatically - with tools that are provided alongside the server - or written manually. In our website, this file was written manually in order to have more precise control over the mapping. Once the mapping file was written, we took advantage of the included D2R server web interface to make sure that the data was being mapped correctly. This interface also allowed us to test the SPARQL queries with the built-in query tester. In Figure 16 we can see the result of a query retrieving general information on the samples stored in the database. Once the data was published as linked data and using the *SPARQLWrapper* library on the client application, we were able to use the developed queries on the D2R server and finally display the samples on the website. The landing page of the website is a simple world map with the samples displayed using markers (Figure 17). By hovering above the markers, the correspondent sample name is displayed. The website can be accessed through the following URL:

http://146.193.41.162:8880/

SPARQL results:

sample	name	latitude	longitude
db:sample/2	"E-1"	38.7	-9.46667
db:sample/3	"E-2"	38.80206	-9.46305
db:sample/4	"E-3"	39.5373489	-9.1745249
db:sample/6	"F-2"	38.93333	-8.9
db:sample/7	"F-3"	39.41445	-7.62196
db:sample/9	"M-17"	37.13856	-8.53775
db:sample/8	"M-12"	38.44451	-9.10149
db:sample/5	"F-1"	38.58333	-7.83333
db:sample/10	"M-19"	37.00864	-8.94311

Fig. 16. Query results for sample contextualization.

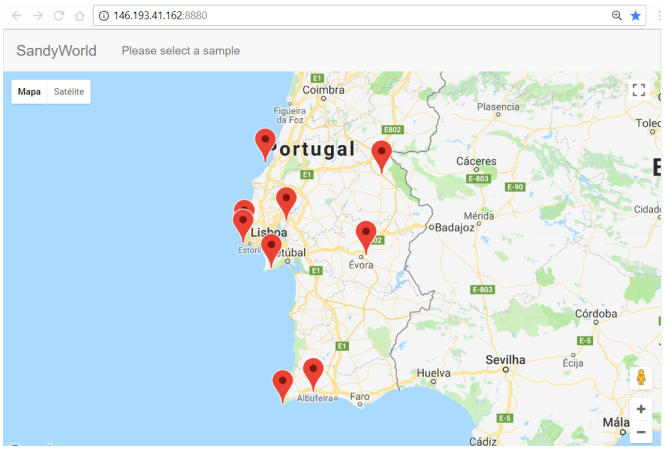

Fig. 17. Web site landing page.

B. Integration with GeoNames

The main objective of linked data is that the connections between resources can be extended beyond a single graph, enabling computers to follow the links to other datasets that provide additional and useful information. To achieve this, ontologies provide a common set of terms that can be used to ease the process of linking data sets and leverage the features of the semantic web.

To test the ontology as a tool for integrating different applications, we showed how the published sample data can be retrieved by an external system using this ontology.

One of the main factors that are considered when analyzing a sand sample is the location at which it was retrieved. As explained in Section **??**, the *GeoNames* database can provide extensive location data, which could be linked to the data published by the D2R server.

Since the *GeoNames* database is not accessible through an official SPARQL endpoint, the third-party service *FactForge* [35] was used. This web service provides combined information from the *GeoNames* and *DBPedia* through an open SPARQL endpoint. Unfortunately, the used version of the D2R server does not support federated queries, so it was not possible to query the *FactForge* graph within the D2R server. However, the contrary was possible. By querying *FactForge* with a federated query that referenced the graph published by the sands data server, it was possible to integrate *GeoNames* resources with the sample data. Essentially, instead of querying the server directly, the client application queries the *FactForge* SPARQL endpoint using a federated query, as shown in Figure 18.

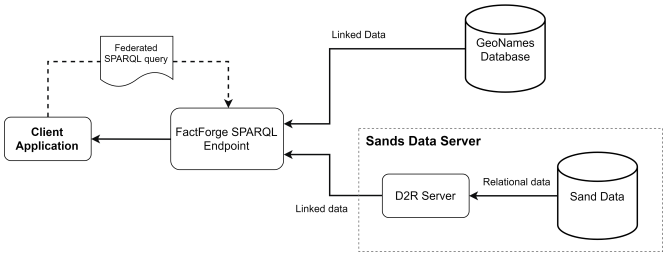

Fig. 18. Querying the *FactForge* SPARQL endpoint using federated queries.

The query that was used to test the integration between the sample data and the *GeoNames* resources was written to retrieve all samples within a certain distance from a specific location. For example, all samples within fifty kilometers of the center of Lisbon. The idea was that the *GeoNames* data would translate the location *Lisbon* to a set of coordinates that would then be used to create a bounding box of fifty kilometers around the location. Filtering only samples within that box, we are effectively relating data between the two applications and retrieving the request information. It is important to note that this integration is facilitated by the ontologies of both data sources. These ontologies provide a vocabulary that applications can use to structure and expose their data in known formats.

In Figure 19 we can see the samples being displayed through the *FactForge* SPARQL endpoint This example shows how

Filter query results — Showing results from 1 to 4 of 4. Query took 0.3s, minutes ago.

	sample	name	latitude	longitude
1	http://146.193.41.162/resource/sample/2	E-1	"38.7"^^xsd:double	"-9.46667"^^xsd:double
2	http://146.193.41.162/resource/sample/3	E-2	"38.80206"^^xsd:double	"-9.46305"^^xsd:double
3	http://146.193.41.162/resource/sample/6	F-2	"38.93333"^^xsd:double	"-8.9"^^xsd:double
4	http://146.193.41.162/resource/sample/8	M-12	"38.44451"^^xsd:double	"-9.10149"^^xsd:double

Fig. 19. Samples within fifty kilometers of Lisbon. Source: *factforge.net*.

different applications can relate their information with other systems using the semantic web. By studying the *GeoNames* ontology we were successfully able to process the sample data in much more meaningful ways. If other applications study and apply the developed ontology to their data, it would be possible to create a web of linked data on sand samples, which could benefit countless applications.

V. Conclusion

As the internet is getting more and more present in scientific research, it is vital that we take advantage of its full potential. Sharing data among different research projects within a knowledge domain or even between different ones can be a great advantage and help to draw meaningful conclusions.

The semantic web, together with the linked data principles, aims to change the way we use the internet, taking advantage of computers to not only display the data but also process

and connect it. To make the most out of the semantic web, ontologies are used as a way of creating a common set of terms that different data sources can map their data to. This allows us to define machine-readable vocabularies that enable machines to understand the semantics of data.

In this work, we developed an ontology that describes sand samples in an attempt to create a standard set of terms that would enable different data sources to integrate with each other over the semantic web. We took into account the main characteristics of sand samples, as well as the means by which the samples are analyzed. To motivate the future use of the developed ontology, different classes and properties were linked to already existing ontologies, like the *Dublin Core Metadata Initiative*, and *GeoNames*. The BGS grain sorting vocabulary was also used as a way of testing how normalized terms can help when publishing linked data.

As a way of evaluating the ontology, a relational database was created and published as linked data according to the main classes and properties of the developed ontology. Once the data was available through a SPARQL endpoint, the capability of the ontology to connect data sources was verified by successfully integrating the sample data with the *GeoNames* database. In addition, the final result of this integration was displayed using a simple web page along with a series of specialized queries. The development of this ontology allowed us to not only verify how an application can consume linked data from any data source, but also how to expose our data to other applications on the semantic web.

In conclusion, Linked data can be a significant advantage for the future of scientific research in every knowledge domain. The ability to relate data from different databases and drawing meaningful conclusions can be a very powerful tool. By developing and connecting ontologies, we can create a universal and machine-readable language that can enable computers to understand the meaning of the data and process it in much more elaborate ways.

ACKNOWLEDGMENT

The author would like to thank Prof. João Nuno de Oliveira e Silva for the overall guidance and technical support, and Prof. Anabela Gonçalves Cruces for the input on the sedimentology domain.

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001.

[2] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," *International journal on semantic web and information systems*, vol. 5, no. 3, pp. 1–22, 2009.

[3] T. Berners-Lee, "Design issues: Linked data (2006)," *URL http://www. w3. org/DesignIssues/LinkedData. html*, 2011.

[4] M. A. Musen, "Dimensions of knowledge sharing and reuse," *Computers and biomedical research*, vol. 25, no. 5, pp. 435–467, 1992.

[5] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?" *International journal of human-computer studies*, vol. 43, no. 5-6, pp. 907–928, 1995.

[6] N. F. Noy, D. L. McGuinness *et al.*, "Ontology development 101: A guide to creating your first ontology," 2001.

[7] H. Haas and A. Brown, "Web services glossary," *W3C Working Group Note (11 February 2004)*, vol. 9, pp. 784–786, 2004.

[8] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, and D. Winer, "Simple object access protocol (soap) 1.1," 2000.

[9] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible markup language (xml)." *World Wide Web Journal*, vol. 2, no. 4, pp. 27–66, 1997.

[10] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana *et al.*, "Web services description language (wsdl) 1.1," 2001.

[11] G. Klyne and J. J. Carroll, "Resource description framework (rdf): Concepts and abstract syntax," 2006.

[12] S. Bechhofer, "Owl: Web ontology language," in *Encyclopedia of database systems*. Springer, 2009, pp. 2008–2009.

[13] D. Brickley, R. V. Guha, and B. McBride, "Rdf schema 1.1," *W3C recommendation*, vol. 25, pp. 2004–2014, 2014.

[14] SWEET. Semantic web for earth and environmental terminology. Accessed May 2018. [Online]. Available: https://sweet.jpl.nasa.gov/

[15] R. G. Raskin and M. J. Pan, "Knowledge representation in the semantic web for earth and environmental terminology (sweet)," *Computers & geosciences*, vol. 31, no. 9, pp. 1119–1125, 2005.

[16] M. D. Team, "Nadm conceptual model 1.0-a conceptual model for geologic map information," *US Geological Survey Open-File Report*, vol. 1334, 2004.

[17] L. Struik, M. Quat, P. Davenport, A. Okulitch *et al.*, "A preliminary scheme for multihierarchical rock classification for use with thematic computer-based query systems," 2002.

[18] H. A. Babaie, J. S. Oldow, A. Babaei, H. G. A. Lallemant, A. J. Watkinson, and A. Sinha, "Designing a modular architecture for the structural geology ontology," *SPECIAL PAPERS-GEOLOGICAL SOCIETY OF AMERICA*, vol. 397, p. 269, 2006.

[19] J. Zhong, A. Aydina, and D. L. McGuinness, "Ontology of fractures," *Journal of Structural Geology*, vol. 31, no. 3, pp. 251–259, 2009.

[20] A. Tripathi and H. A. Babaie, "Developing a modular hydrogeology ontology by extending the sweet upper-level ontologies," *Computers & Geosciences*, vol. 34, no. 9, pp. 1022–1033, 2008.

[21] Comission for the Management and Application of Geoscience Information. Geologic timescale model. Accessed May 2018. [Online]. Available: http://resource.geosciml.org/ontology/timescale/gts

[22] C. Wang, X. Ma, and J. Chen, "Ontology-driven data integration and visualization for exploring regional geologic time and paleontological information," *Computers & Geosciences*, vol. 115, pp. 12–19, 2018.

[23] M. Wick and B. Vatant, "The geonames geographical database," *Available from World Wide Web: http://geonames. org*, 2012.

[24] B. Vatant and M. Wick. (2012) Geonames ontology. Accessed May 2018. [Online]. Available: http://www.geonames.org/ontology/

[25] G. M. Friedman, J. E. Sanders *et al.*, *Principles of sedimentology*. Wiley, 1978.

[26] W. C. Krumbein, F. J. Pettijohn *et al.*, "Manual of sedimentary petrography," 1938.

[27] F. M. Gradstein, J. G. Ogg, M. Schmitz, and G. Ogg, *The geologic time scale 2012*. elsevier, 2012.

[28] A. H. Munsell, D. Nickerson *et al.*, *Munsell color system*. Published by the American institute of physics for the Optical society of America, 1940.

[29] S. Weibel, J. Kunze, C. Lagoze, and M. Wolf, "Dublin core metadata for resource discovery," Tech. Rep., 1998.

[30] S. University. Protg, open-source ontology editor. Accessed May 2018. [Online]. Available: https://protege.stanford.edu/

[31] R. Shearer, B. Motik, and I. Horrocks, "Hermit: A highly-efficient owl reasoner." in *OWLED*, vol. 432, 2008, p. 91.

[32] C. Bizer and R. Cyganiak, "D2r server-publishing relational databases on the semantic web," in *Poster at the 5th international semantic web conference*, vol. 175, 2006.

[33] C. Bizer and A. Seaborne, "D2rq-treating non-rdf databases as virtual rdf graphs," in *Proceedings of the 3rd international semantic web conference (ISWC2004)*, vol. 2004. Proceedings of ISWC2004, 2004.

[34] C. Bizer and R. Cyganiak. The d2rq plattform. Accessed May 2018. [Online]. Available: http://d2rq.org/

[35] Factforge sparql endpoint. Accessed May 2018. [Online]. Available: http://factforge.net/sparql