RDF Repository for Biological Experiments

Filipa Rodrigues Rebelo

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. José Luís Brinquete Borbinha

Examination Committee

Chairperson: Prof. Miguel Nuno Dias Alves Pupo Correia

Supervisor: Prof. José Luís Brinquete Borbinha

Member of the Committee: Prof. Ana Teresa Correia de Freitas

November 2014

Acknowledgments

I would like to thank to KDBIO Group for their time, patience and important input in explaining the main biological concepts, as well as, in helping me review the documents produced. I would also to thank to FCT (Fundação para a Ciência e a Tecnologia) since this work was supported by its national funds, under projects PEst-OE/EEI/LA0021/2013, TAGS PTDC/EIA-EIA/112283/2009, PTDC/AGR-GPL/109990/2009 and DataStorm EXCL/EEI-ESS/0257/2012, and by the project TIMBUS, co-funded by the European Commission under the 7th Framework Programme for research and technological development and demonstration activities (FP7/2007-2013) under grant agreement no. 269940.

At last, I would like to thank in a special way to Eng. João Edmundo for his infinite patience and support in helping me clear my thoughts, motivate me and also in helping me review the documents produced over and over again.

Abstract

Life Sciences researchers recognize that the reusability and sharing of data by interlinking all data about an entity of interest and to assemble it into a useful block of knowledge is important to give a complete view of biological activity. Simultaneously, the Web is evolving from a set of static individual HTML documents into a Semantic Web of interlinked data, which enables solutions like Linked Data to greatly contribute to its evolution. Therefore, this work applied Semantic Web principles to create a unified infrastructure to manage, link and promote the reusability of the data produced from biological experiments. This was based on the concept of a data repository that supports multiple ontologies to structure its data. To prove the repository concept, the IICT together with ITQB-UNL/IBET provided experimental data about *Coffea Arabica* plant to support the research to identify potential candidate biomarkers for resistance against *Hemileia Vastatrix* fungi (causal agent of coffee leaf rust). As test-case it was used a section of the data retrieved from coffee leaf rust assays which comprises the proteome modulation of coffee leaf apoplastic fluid, by greenhouse conditions, using 2D electrophoresis (2DE). Moreover, the ontology responsible to structure this data was the *Plant Experimental Assays Ontology* developed by KDBIO's Group. Technologically, the repository was developed using Jena framework to import and transform the data in RDF, interlinking it internally and with external sources.

Keywords: Ontology, Life Sciences, Data Repository, Biological Experiments, Linked Data

Resumo

Os investigadores da área das Ciências da Vida reconhecem que a partilha e reutilização da informação interligando todos os dados sobre uma entidade de interesse, juntando tudo num único bloco de conhecimento, é importante para dar uma visão completa da atividade biológica. Concomitantemente, a Web está a evoluir de um conjunto de documentos HTML estáticos para uma Web Semântica de dados interligados, permitindo que soluções como o Linked Data contribuiam fortemente para a sua evolução. Deste modo, este trabalho aplicou os princípios da Web Semântica para criar uma infraestrutura única para gerir, ligar e promover a reutilização dos dados produzidos pelas experiências biológicas. Baseou-se no conceito de um repositório de dados que suporta múltiplas ontologias para estruturar esses dados. Para provar o conceito deste repositório, o IICT em conjunto com o ITQB-UNL / IBET forneceu dados experimentais sobre a planta C*offea Arabica* para apoiar a investigação no sentido de identificar potenciais biomarcadores candidatos à resistência ao fungo *Hemileia Vastatrix* (agente causador da ferrugem do cafeeiro). Como caso de teste, foram utilizados uma parte dos dados obtidos dos ensaios à ferrugem na planta do café que compreende a modulação do proteoma do fluído apoplástico da folha do café, em condições de estufa, usando electroforese 2D (2DE). Adicionalmente, a ontologia responsável por estruturar esses dados foi a *Plant Experimental Assays Ontology* desenvolvida pelo grupo KDBIO. A nível tecnológico, o repositório foi desenvolvido usando a framework Jena para importar e transformar os dados em RDF, interligando-os internamente e com outros repositórios.

Palavras-Chave: Ontologia, Ciências da Vida, Repositório de Dados, Experiências Biológicas, Linked Data

Table of Contents

List of Figures

List of Tables

List of Acronyms

BGP	Basic Graph Pattern
CSV	Comma Separated Values
GUI	Graphical User Interface
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
LOD	Linked Open Data
N3	Notation3
OWL	Ontology Web Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SPARQL	SPARQL Protocol and RDF Query Language
TSV	Tab Separated Values
TURTLE	Terse RDF Triple Language
UI	User Interface
URI	Uniform Resource Identifier
XML	eXtensible Markup Language
W3C	World Wide Web Consortium

1. Introduction

We are surrounded by data every day. Increasingly, the access to data is easier giving us the means to make better decisions. It was the Internet explosion and its evolution the responsible for this growing, leading us to this fever of connection and share of data which are making us move from a Web based on documents to a Web of data that links arbitrary things – the Semantic Web [34].

The Semantic Web technologies have huge "potential to transform the Internet into a distributed reasoning machine that will not only execute extremely precise searches, but will also have the ability to analyze the data it finds to create new knowledge" [24]. Therefore, Linked data is the first step to achieve this vision. It uses the Resource Description Framework (RDF) and the Hypertext Transfer Protocol (HTTP) to publish structured data on the Internet and to connect it between different data sources effectively, allowing data in one source to be linked to data in another data source [11]. Furthermore, Linked Data can be used to share data openly (known as Linked Open Data or LOD) or privately using access control approaches. One example of LOD is data.europeana.eu[1] that is a current effort of making European metadata of cultural heritage objects available as LOD on the internet [28]. Once data is linked to shared ontologies (which provide a vocabulary to describe and structure the properties and relationships of objects) machines will derive new knowledge by reasoning about that content, rather than just understanding it, facilitating also the effort in publishing, consumption and integration of data, as well as, helping to promote reusability.

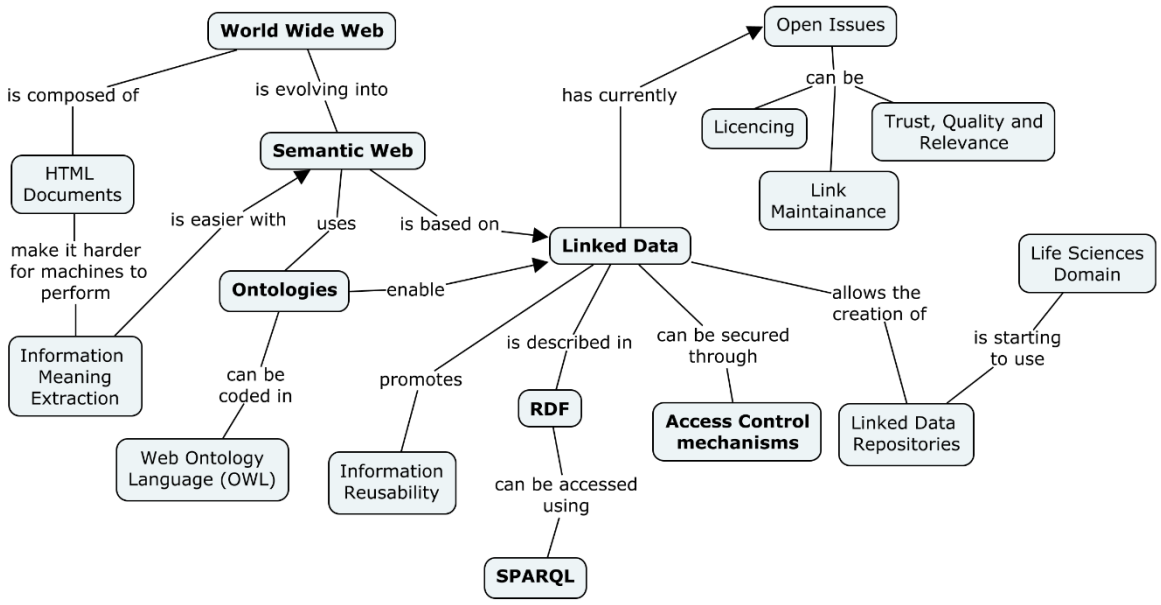


Figure 1. Concept map of the context of the problem

Although there are still some open research issues in Linked Data like link maintenance, licensing and the evaluation of the trustworthiness, quality and relevance of the data, is starting to be used in the Life Sciences domain to promote information reuse between the massive volumes of data available.

[1] http://pro.europeana.eu/linked-open-data accessed 27/12/2013

Linked Data repositories (also called triple stores) are one example of use for biological data with a high degree of linking.

In the **Figure 1** is possible to see an overview of the main concepts that will be addressed thoroughly in this work.

1.1. Motivation

The contemporary biological experimental studies produce a great wealth of heterogeneous and interdependent data which are influenced by the diversity of protocols, tools, data formats and context-specific parameters used at different steps and which makes the studies difficult to reproduce [43]. Moreover, there is currently no standard workflow established to support the management of all gathered data from the biological experiments.

1.2. Problem

Life Sciences researchers claim that it is difficult to assemble all relevant biological information about an entity into a useful block of knowledge. Although the data retrieved from the biological experiments is structured inside an Excel file, it doesn't allow a global view over all the data. Some annotations can be added by users to these files, but no structural changes are made. Additionally, the Excel files are stored into a server in a decentralized approach. Moreover, there can be redundancy amongst the data by repeating the information about entities present in different Excel files, and no reasoning or general analysis can be made in an integrated way since all the data is scattered.

1.3. Proposed Solution

New approaches must be considered to provide the means to efficiently gather biological experiments data into a unified infrastructure, maintaining its semantics and enabling the linkage with multiple sources, thus enriching the data. Because there is not an infrastructure to manage the data related with biological experiments, the goal of this work is to present a repository that can efficiently store and gather this information in a structured way. This can be achieved by interlinking data with RDF and using ontologies to promote the preservation of semantic relationships between the entities represented therein, making the interpretation of results and the integration of the data produced by different experiments easier, as well as, enabling a more in-depth analysis and reasoning over the data.

Hence, the main objectives of this dissertation are to:

- Create an infrastructure for extracting data from biological experiments and store it according to a given ontology and in a way that enables its linkage with external sources;
- Create a web Graphical User Interface (GUI) adequate to manage projects (a model that gathers all the information related to one experiment), ontologies and users;
- Define a uniform solution for importing data from biological experiments;

- Link, when possible, the data from the repository with external resources;

1.4. Main Contributions

The main contributions of this dissertation are:

- A web framework for collecting all data about biological experiments, in a structured and centralized way, capable of storing, importing and exporting data in a reusable format;
- Interlinking of the internal data within the repository and with external sources;

1.5. Document Structure

Following the Introduction, is the **Section 2** with the related work addressing the main concepts of Linked Data and the technology involved. An analysis of the emblematic applications of Linked Data is done, as well as, several articles about Linked Data in Life Sciences, closing this section with the open research issues. Next, the proposed repository solution is addressed in **Section 3** where the repository goals and its data structure and architecture are explained. In **Section 4,** the results are described and, in **Section 5,** a self-assessment of the work is done. Finally, this work finishes with **Section 6**, where the conclusion and future work are presented.

2. Related Work

This section starts to introduce the relevant state-of-the-art in Semantic Web domain and the main concepts related and relevant to the problem of this dissertation. It is also made a survey of emblematic applications of Linked Data as well as some examples of Linked Data in Life Sciences. Finally, it is made an examination of the open research issues of Linked Data.

2.1. Web Architecture and the Semantic Web

Nowadays, the Web's architecture is designed in a way that people are able to store and structure their own information such that it can be used by themselves and others, and referenced by everyone eliminating the need to keep and maintain local copies [7]. The classic Web is based on HTML which provides a standard for structuring documents, setting hyperlinks between Web documents - that are the basis for the navigating and crawling the Web - and allowing the integration of all Web documents into a single global information space [8]. The problem is that machines have difficult to extract any meaning from these documents themselves. However, the vision of the Web is reaching a new level. The share of information is not a problem anymore and now we want to do it in a way that adds value to society and protects individual privacy and preferences[2]. If we took all the HTML data in the world, and allowed its metadata to be treated and researched as if it were one database, the benefits of its automated research in comparison to today's tools and software would be tremendous[1]. This is what Semantic Web is all about. The Semantic Web is a Web of data, an extension of the principles of the Web from documents to data, requiring the need to create a common solution that allows data to be shared and reused across application, enterprise, and community boundaries, to be processed automatically by tools as well as manually, including revealing possible new relationships among pieces of data[3].

In short, the Semantic Web involves providing a language that expresses both data and rules for reasoning about the data and that allows rules from any existing knowledge-representation system to be exported onto the Web.

With the arrival of the Semantic Web, the current network of online resources is expanding from a set of static documents designed mainly for human consumption to a new Web of dynamic documents, services and devices, which software agents will be able to understand (**Figure 2**[4]) [46].

The number of Semantic Web applications is increasing every day[3]. For example, Life Sciences research demands the integration of diverse and heterogeneous data sets that originate from distinct communities of scientists in separate subfields. Scientists, researchers, and regulatory authorities in genomics, proteomics, clinical drug trials, and epidemiology all need a way to integrate these components [51].

[2] http://stm.sciencemag.org/content/4/165/165cm15.full accessed 05/11/2013
[3] http://www.w3.org/RDF/FAQ accessed 01/11/2013
[4] http://paanchiweb.blogspot.pt/2012/12/getting-essence-of-semantic-web.html accessed 25/11/2013

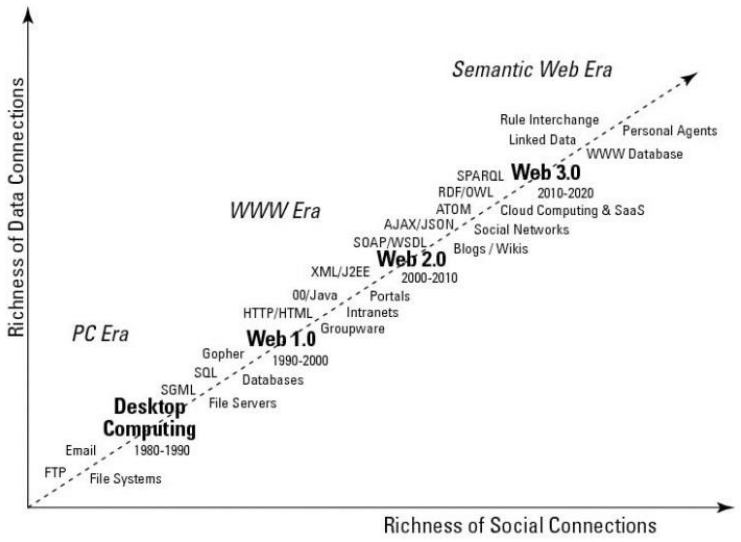


Figure 2. Four major waves of Web evolution

Web APIs are one of the solutions to deal with heterogeneous systems and the variety of information sources [2] [47]. Web APIs can have different techniques for identification and access, as well as, represent retrieved data in different formats. However, most of them don't assign globally unique identifiers to their data item, which makes it impossible to create hyperlinks between data items provided by different APIs. As a result, the Web becomes divided into different data silos which forces developers to choose a specific set of data sources for their applications, because "they can't implement applications against all the data available on the Web" [8].

Other way of dealing with heterogeneity arises through the adoption of ontologies. They can be described as a set of term definitions in the context of a specific domain. These definitions associate the names of entities (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names are meant to denote, and contain formal axioms that constrain the interpretation and well-formed use of these terms [26]. Several initiatives to develop ontologies are rising in areas such like biology, medicine, genomics and related fields as well as many other disciplines that are adopting what began in the life sciences. Then, these ontologies can become standard languages to define specific domains and can be easily deployed on the Web [51]. As a result, ontologies are becoming more and more popular providing "a shared and common understanding of some domain that can be communicated between people and application systems" [19].

Next, some technologies and solutions that contribute and follow the principles of the Semantic Web will be addressed.

2.1.1. Linked Data

Linked data is about using the Web to create typed links between data from different sources. It is based on a set of principles to publish structured data on the Web so that it can be interlinked and become more useful. These principles allow data to be published on the Web in order to be machine-readable, have the meaning explicitly defined, be linked to other sets of external data, and in turn with

the possibility of being linked from external data sets [10]. According to Tim Berners-Lee[5], a set of best practices have been set for the publication of data on the Web in a way that all published data becomes part of a single global data space. That set of rules - known as the "principles of Linked Data" - claim that every piece of data must has an associated URI (Uniform Resource Identifier) that, when looked up, should provide useful information, using the standards RDF and SPARQL (Simple Protocol and RDF Query Language). Farther**,** the data should include links to other URIs so that more things can be discovered either by people or by machines. While the current Web is based on HTML to describe untyped documents connected by hyperlinks, the Linked Data relies on documents containing data in RDF to make typed statements that connect arbitrary things in the world.

Summarizing, the principles of Linked Data provide the basic mechanism for publishing and connecting data using the infrastructure of the Web, taking advantage of its architecture and standards [10] and thus forming the Web of Data [18].

2.1.2. RDF

The RDF is a standard defined by the World Wide Web Consortium (W3C) for making statements to describe information resources. RDF statements, also known as RDF triples, are composed by a subject, predicate and an object. In **Figure 3**, it is possible to see an example of a simple RDF statement.

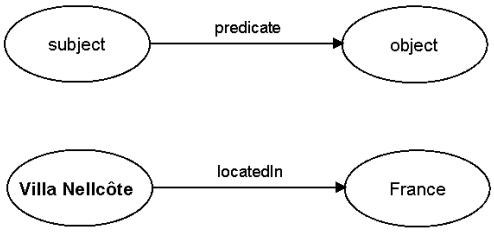


Figure 3. Example of the representation of a RDF statement

The collection of RDF statements describing resources is called the RDF graph**.** The collection of RDF graphs is called RDF dataset and is used to organize collections of RDF graphs. Furthermore, URIs are the basis mechanism to identify subjects, predicates and objects in RDF statements, due to its generic nature. The subject of a triple is the URI identifying the described resource; the object can either be a simple literal value or a URI of another resource that is somehow related to the subject; the predicate indicates what kind of relation exists between a subject and an object [34].

In order to represent RDF statements in a machine-processable way, RDF defines several formats:

- Extensible Markup Language (XML), referred to as RDF/XML;
- Terse RDF Triple Language (TURTLE);
- Notation 3 (N3);

[5] http://www.w3.org/DesignIssues/LinkedData.html accessed 13/10/2013

RDF was developed with the purpose of enabling applications to process web content in a standard and machine-readable way, simplifying the operation at Web scale. This technology is an essential foundation for the development of the Semantic Web [32].

2.1.3. Querying with SPARQL

As mentioned in **Section 2.1.1**, the language used to extract data from RDF graphs is SPARQL. It defines a standard query language and data access protocol for use with the RDF data model [48].

In order to exchange the results in machine-readable form, SPARQL supports four common exchange formats, namely the XML, the JavaScript Object Notation (JSON), Comma Separated Values (CSV), and Tab Separated Values (TSV)[6].

SPARQL queries are sent from a client to a service known as a SPARQL endpoint[6], using the HTTP protocol (**Figure 4**[7]).

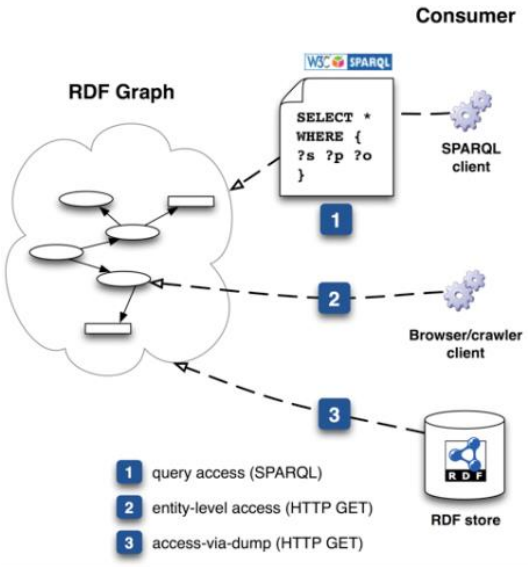


Figure 4. RDF access architecture

Usually, the interaction between the client and the endpoint is defined as machine-friendly interface that allows the user to enter the queries and to display the results in a meaningful way[8].

2.1.4. Ontologies with OWL

Ontologies were idealized to specify not only the definition of a controlled set of terms but also their relations with each other in a single domain context. An ontology defines the terms used to describe and represent an area of knowledge. Although XML Schemas[9] are sufficient for exchanging data between parties who have agreed to the definitions beforehand, their lack of semantics prevents machines from reliably performing this task with new XML vocabularies. Therefore, ontologies provide several

[6] http://www.w3.org/2009/sparql/wiki/Main_Page accessed 24/11/2013
[7] http://www.w3.org/TR/rdb2rdf-ucr/ accessed 25/11/2013
[8] http://semanticweb.org/wiki/SPARQL_endpoint accessed 24/11/2013
[9] http://www.w3.org/XML/Schema accessed 23/12/2013

advantages like the ability to share structured information between diverse users and software tools, to reuse the created language and make explicit domain assumptions [43]. Based on these ideas, the Ontology Web Language (OWL) arose to be used by applications that need to process the content of information instead of just presenting information to humans. This ontology language was developed by the W3C for the Semantic Web and facilitates greater machine interpretability of web content than that supported by XML, RDF, and RDF Schema (RDFS)[10] by providing additional vocabulary along with a formal semantics[11]. In fact, OWL is a vocabulary extension of RDF that enables the definition of domain ontologies and sharing of domain vocabularies. It is modeled through an object-oriented approach and the structure of a domain is described in terms of classes and properties [56].

Moreover, ontologies can be edited through tools like Web Protégé[12] (**Figure 5**) which is a web adaption of the popular open-source ontology editor software Protégé[13] that enables users to create Projects (collections of ontologies) and share them with their collaborators, adding them as viewers, commenters or editors. It also supports threaded discussions amongst users, change notifications and versioning control over the ontologies [36].

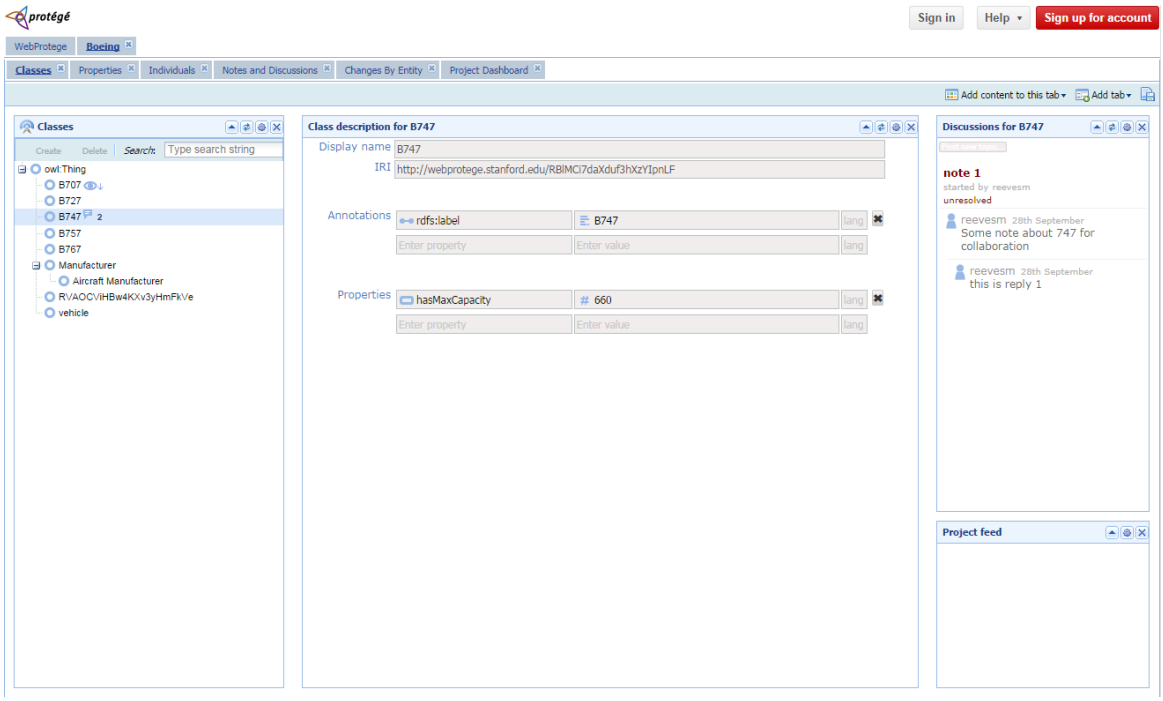


Figure 5. Web Protégé – Ontology about a Boeing aircraft

Some ontology tools can also perform automated reasoning using the ontologies, and therefore provide advanced services to intelligent applications such as conceptual/semantic search and retrieval, decision support, speech and natural language understanding and knowledge management.

[10] RDF Schema is a vocabulary for describing properties and classes of RDF resources, with a semantics for generalization-hierarchies of such properties and classes.
[11] http://www.w3.org/TR/2004/REC-owl-features-20040210/ accessed 05/01/2014
[12] http://webprotege.stanford.edu/ accessed 23/12/2013
[13] http://protege.stanford.edu/ accessed 23/12/2013

2.1.5. Data Access Control

Data can be linked but not open, and can be open but not linked. For that reason it is important to point out the difference between Linked Data and Linked *Open* Data. Open data refers to data that is accessible to anyone, generally available on the Web and uses non-property formats. Thus, LOD can be defined as Linked Data released under an open license, which does not impede its reuse for free.

In **Figure 6**[14] it's possible to see the graph of Linked Open Data as of September 2011.

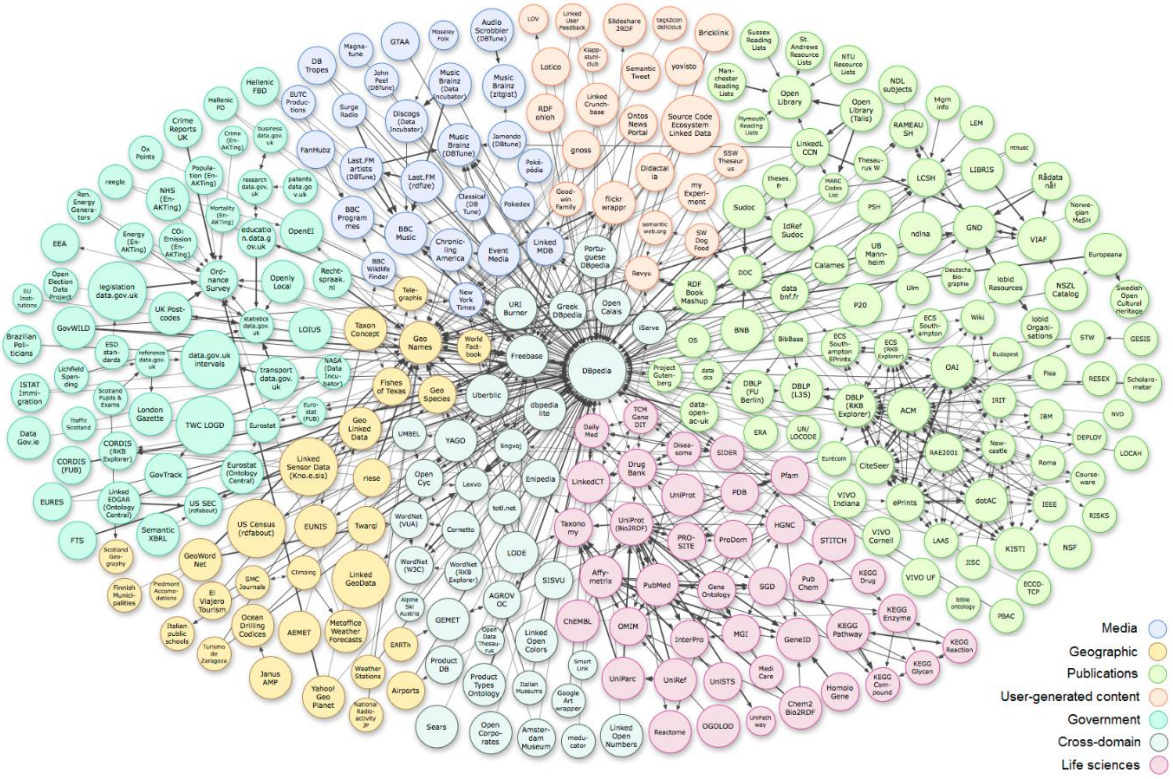


Figure 6. LOD cloud as of September 2011

More and more, datasets are being published in the Linked Data Cloud without the addition of any kind of metadata specifying the access control conditions under which the data is accessible [21], making the data publicly available. Nevertheless, we might want to have the control of who accesses our data. Due to this, panoply of solutions has been proposed to solve this problem wherein many of them rely in access control lists that define which users can access the data. This is the case of Web Access Control (WAC) [13] that is a vocabulary to describe access control privileges, enabling owners to create access control lists that specify access privileges to the users that can access the data. Nevertheless, this vocabulary is designed to specify access control to the full RDF document rather than specifying access control properties to specific data contained within the RDF document [35].

[14] http://lod-cloud.net/ accessed 26/11/2013

A Relation Based Access Control model (RELBAC) is proposed in [22], which provides a model of permissions based on description logics. The basic concepts of this model are subjects, objects, permissions and rules. It is based on hierarchies of permissions, where permissions are the relations between subjects and objects and rules express the kind of access rights that subjects have on objects.

Another solution was suggested in [21], were is presented a way of controlling access to RDF data with a high-level access control specification language that allows fine-grained specification of access control permissions at triple level and formally define its semantics. Here, the user must explicitly identify the accessibility of an item through the use of annotations.

In Fabian Abel et al [1] the policy permissions are injected in the query in order to ensure that the triples obtained are only the accessible ones. Given an RDF query, the framework partially evaluates all applicable policies and constraints the query according to the result of such evaluation. The modified query is then sent to the RDF store which executes it like a usual RDF query.

The presentation of a virtual model instead of a real one, generated by filtering the original model, is the idea of the framework proposed in [15]. The framework is composed for 4 parts: a query engine which can apply subset selection filters to a given model; a rule processor which decides whether a query filter is fired for a given action or not; a RDF schema which describes a basic vocabulary to store rules and query filters; and a access control processor, which starts the query engine and rule processor as needed and maintains some session data.

Two different approaches are defined in [20] to model Role Based Access Control (RBAC) using OWL. For each one, it is defined an ontology with the basic RBAC concepts. Since in RBAC permissions are associated with roles, and users are made members of appropriate roles [50], the complexity of the system revolves around how roles are represented and managed.

It is advocated the adoption of an access control policy models in [54] that follow two main design guidelines: context-awareness to control resource access on the basis of context visibility and to enable dynamic adaptation of policies depending on context changes, and semantic technologies for context/policy specification to allow high-level description and reasoning about context and policies. This access control model adopts a hybrid approach to policy definition based on Description Logic (DL) and Logic Programming (LP) rules.

2.2. Emblematic Applications of Linked Data

Based on the community-hosted collection of Linked Data applications[15], some examples explained by Michael Hausenblas [31], and the Linked Open Data cloud[16] (which shows all applications that share their Linked Data openly), a selection of Linked Data applications was put together. Almost all applications use, one way or another, the DBpedia [4].

[15] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData/Applications accessed 25/11/2013
[16] http://lod-cloud.net/ accessed 25/11/2013

These applications were grouped in four categories highlighting the main aspects, from a Linked Data usage point-of-view:

- **LOD Publishing**: applications that publish the data in the LOD (Linked Open Data) cloud;
- **Content reuse**: applications that mainly reuse content of datasets in the LOD cloud in order to save time and resources;
- **Semantic tagging**: applications that use HTTP URIs in the datasets for unambiguously talking about things;
- **Event data management systems**: applications that allow people to organize and query event-related data.

Although this categorization may be too uneven, it should help identify the various use cases one can be after using Linked Data.

2.2.1. LOD Publishing

TWC LOGD

International open government initiatives are releasing an increasing volume of raw government data directly to citizens via the Web [17]. For that reason, Li Ding et al [16] developed a solution to incrementally generate Linked Government Data (LGD) for the US government. Based on this solution Li Ding further cooperated with the Tetherless World Constellation (TWC), and created a Semantic Web-based application called TWC LOGD Portal [17] to support the deployment of Linked Open Government Data (LOGD)[17].

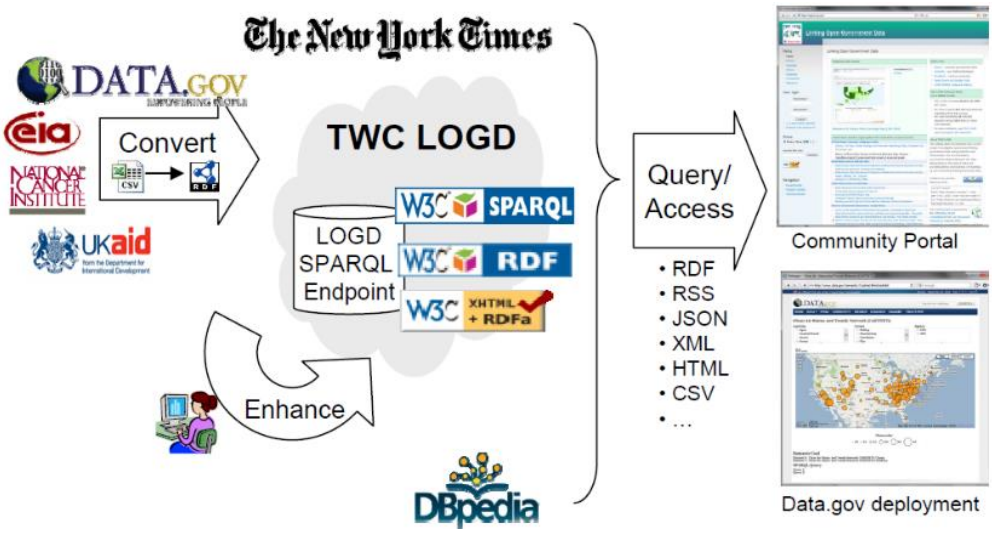


Figure 7. The High-level Workflow of the TWC LOGD Portal

[17] http://logd.tw.rpi.edu/ accessed 27/11/2013

The TWC LOGD Portal demonstrates a model infrastructure and several workflows for linked open government data deployment (**Figure 7**). The Portal has also served as an important training resource as these technologies have been adopted by Data.gov[18] - the US federal open government data site.

US Census Bureau

The U.S. Census data is provided by the Census Bureau[19] in a structured format (with an enormous amount of documentation) and yields on the order of 1 billion RDF triples. This data can be explored through the U.S. Census Bureau's interactive map application (**Figure 8**[20]) which is in fact a layer on top of Google Maps with interaction components.

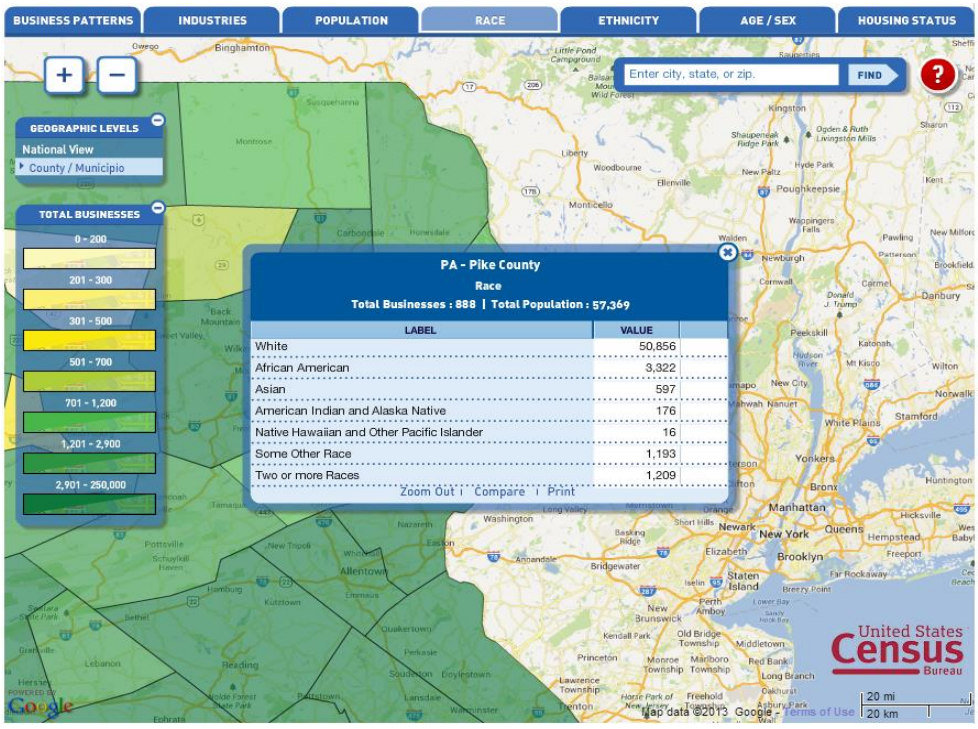


Figure 8. U.S. Census Bureau interactive map application

The data includes population statistics at various geographic levels, from the U.S. as a whole, down through states, counties, sub-counties (roughly cities and incorporated towns), ZIP Code Tabulation Areas (which approximate ZIP codes), and even deeper levels of granularity. The statistics themselves contain total population counts, counts by age, sex, and race, information on commuting time to work, mean income, latitude and longitude of the region, etc.

2.2.2. Content Reuse

BBC's Music and Programmes site

The British Broadcasting Corporation (BBC) uses Linked Data internally as a lightweight data integration technology. BBC manages numerous radio stations and television channels and traditionally,

[18] http://www.data.gov/ accessed 27/11/2013
[19] http://www.census.gov/ accessed 26/11/2013
[20] http://www.census.gov/cbdmap/ accessed 27/11/2013

they use separate content management systems. Therefore, BBC started to use Linked Data technologies together with DBpedia[21] and MusicBrainz[22] as controlled vocabularies to connect content about the same topic residing in different repositories and to augment content with additional data from the Linking Open Data cloud. Based on these connections, BBC Programmes and BBC Music build Linked Datasites for all of its music and programmes [40] in early 2009.

The BBC's Music site[23] was built around the Musicbrainz metadata and DBpedia identifiers. Music metadata such as related artists and latest tracks played on BBC are pulled from Musicbrainz, and for the links pointing to Wikipedia, the introductory text for each artist's biography is fetched from there via DBpedia interlinking.

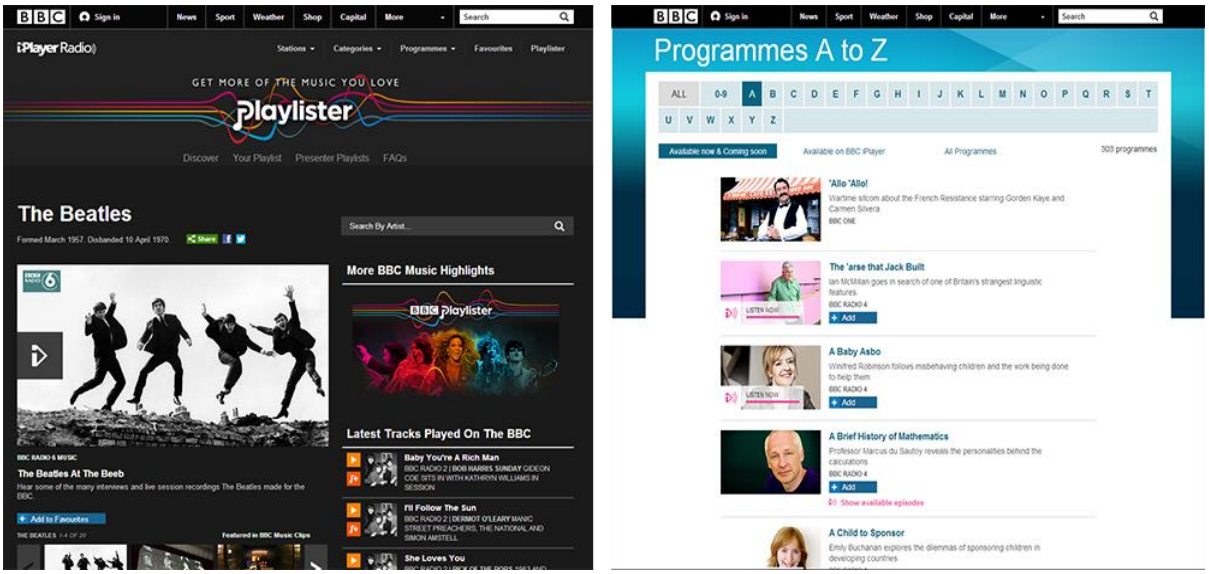


Figure 9. BBC Music Website showing an artist's information (Left). BBC Programmes from A to Z (Right)

The **Figure 9** shows on the left an example of "The Beatles" page showing their biography, BBC reviews and their latest tracks played on BBC. Also on the right there's an example of BBC Programmes ordered from A to Z.

UAd Analyser - A market researcher's to trace discussions

The Understanding Advertising (UAd) Analyser is a web application (implemented with the Google Web Toolkit[24]) for market researchers to trace discussions on the Web [52]. In addition to the interlinking of the discussions throughout various web-based discussion forums (via SIOC[25] and FOAF[26]), the UAd Analyser uses DBPedia categories along with the skos:narrower link property to pull in domain-specific information.

[21] http://dbpedia.org/About accessed 26/11/2013
[22] http://musicbrainz.org/ accessed 26/11/2013
[23] http://www.bbc.co.uk/music accessed 25/11/2013
[24] http://www.gwtproject.org/ accessed 25/11/2013
[25] http://sioc-project.org/ accessed 25/11/2013
[26] http://www.foaf-project.org/ accessed 25/11/2013

Currently, the UAd Analyser only works in the car's domain. The classification of cars (such as Mid-size cars, etc.) and concrete instances (e.g., a Ford Focus) comes from DBpedia thus minimizing the effort to model a certain domain and populating an ontology with instances.

LinkedGeoData

With the OpenStreetMap (OSM)[27] project, a rich source of spatial data became freely available. It is currently used primarily for rendering various map visualizations, but has the potential to evolve into a manifestation point for spatial web data integration.

The main goal of the LinkedGeoData (LGD) [28] project was to boost OSM's data into the Semantic Web infrastructure. This simplifies real-life information integration and aggregation tasks that require comprehensive background knowledge related to spatial features [53]. Such tasks might include, for example, to locally show the products available in bakery shop next door, to map distributed branches of a company, or to integrate information about historical sites along a bicycle track.

Most of the data is obtained by converting data from the popular OSM community project to RDF and deriving a lightweight ontology from it. Furthermore, LinkedGeoData performs interlinking with DBpedia, GeoNames[29], and other datasets, as well as the integration of icons and multilingual class labels from various sources. As a side effect, the LinkedGeoData project is striving for the establishment of an OWL vocabulary with the purpose of simplifying exchange and reuse of geographic data.

RKB Explorer

RKB Explorer [23] provides unified views of information (using graphical interfaces[30]) collected from a significant number of heterogeneous data sources. To resolve the problem that heterogeneous sources may publish different information about same set of entities it implements a set of consistent reference services, which are essentially knowledge bases of URI equivalence generated using heuristics. Also, its information infrastructure is mediated by ontologies and consists of many independent triple stores. In addition, it has a dataset with many tens of millions of triples, and is publicly available through both SPARQL endpoints and resolvable URIs. This solution is also used to explore the publications[31] made available by the Association for Computing Machinery (ACM[32]).

Sig.ma

The interactive information visualization application developed by Giovanni Tummarello et al [55] is named Sig.ma[33]. It is essentially a search engine that provides summary views of the entity the user selects from the results list, alongside additional structured data crawled from the Web and links to related entities. In addition, the search engine applies vocabulary mappings to integrate web data as

[27] http://www.openstreetmap.org accessed 28/11/2013
[28] http://linkedgeodata.org/ accessed 28/11/2013
[29] http://www.geonames.org/ accessed 28/11/2013
[30] http://www.rkbexplorer.com/ accessed 28/11/2013
[31] http://acm.rkbexplorer.com/ accessed 01/12/2013
[32] http://www.acm.org/ accessed 01/12/2013
[33] http://sig.ma/ accessed 28/11/2013

well as specific display templates to properly render data for human consumption. **Figure 10** shows the Sig.ma search engine displaying data about Steve Jobs that has been integrated from 20 data sources.

Another interesting aspect of the Sig.ma search engine is that it approaches the data quality challenges that arise in the open environment of the Web by enabling its users to choose the data sources from which the user's aggregated view is constructed. By removing low quality data from their individual views, Sig.ma users collectively create ratings for data sources on the Web as a whole.

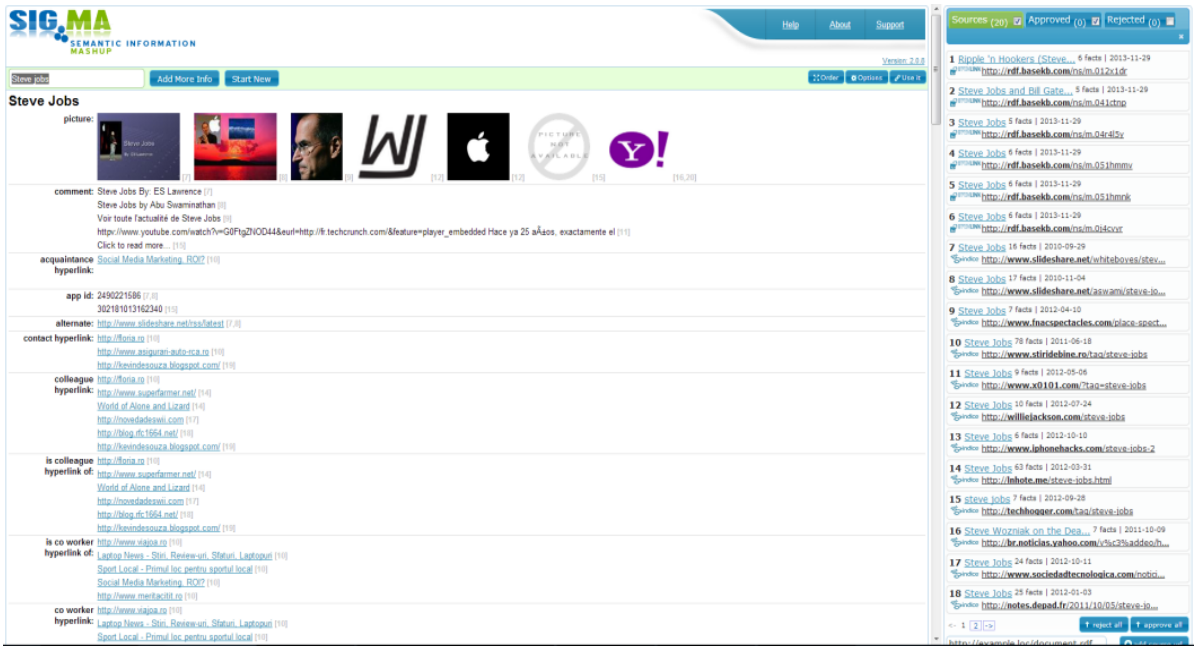


Figure 10. Sig.ma Linked Data search engine displaying data about Steve Jobs

DBpedia Mobile

In order to develop a mobile and geographically oriented application, Christian Becker et al developed the DBpedia mobile[34] [5]. It is basically a location-centric DBpedia client application for mobile devices which is based on the GPS signal of the mobile device. It is able to render a map showing the user's current location and all nearby points of interest retrieved from the DBpedia dataset. It can also use Revyu (application explained in the next section) to show the user detailed information about a point of interest. Besides accessing web data, DBpedia Mobile also enables users to publish their current location, pictures and reviews to the Web as Linked Data, so that they can be used by other applications. Instead of simply being tagged with geographical coordinates, published content is interlinked with a nearby DBpedia resource and thus contributes to the overall richness of the Web of Data.

[34] http://mes-semantics.com/DBpediaMobile/ accessed 28/11/2013

2.2.3. Semantic tagging

Faviki

As is explained in [31], Faviki[35] is a social bookmarking that allows tagging of web pages with “Semantic Tags” coming from DBpedia. The main purpose of the DBpedia URIs is, on the one hand, to provide unambiguous identifiers for concepts, and on the other enrich the tag's description (as shown on the right bottom side of **Figure 11**, a description of the tag “Semantic Web” can be found under the “tag info” panel).

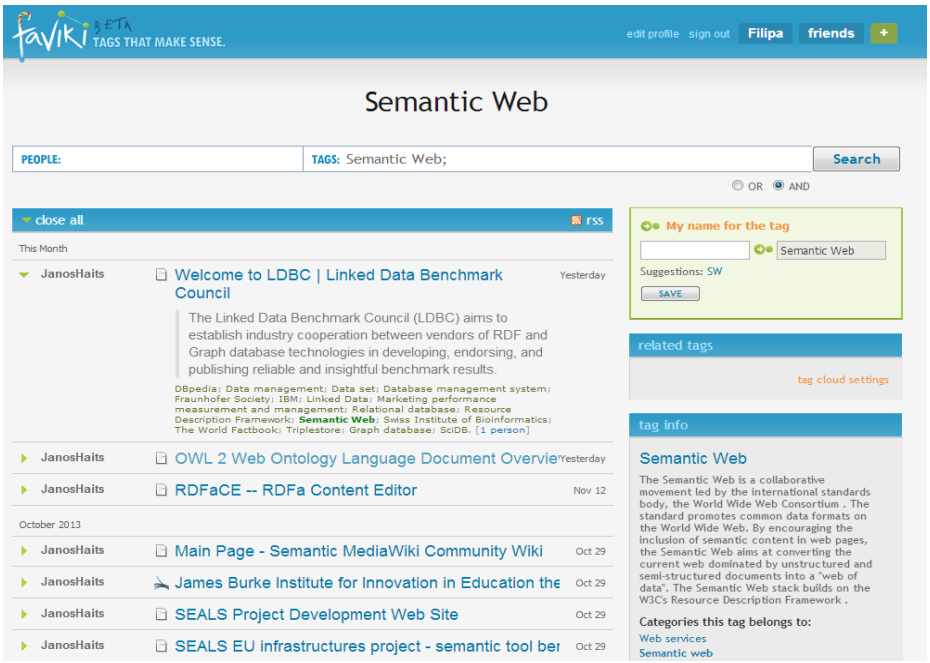


Figure 11. Faviki tagging system

Figure 11[36] shows a query for all bookmarks tagged with “Semantic Web”, represented through the DBpedia URI[37]. Consequently, this usage of DBpedia enables query disambiguation and supports integration tasks.

Revyu

Revyu [33] is a generic reviewing and rating site[38] that consumes Linked Data from the Web of Data to enhance the end-user's experience, exploiting the interlinking with DBpedia. Therefore, links are made at the RDF level to the corresponding item, ensuring that while human users see a richer view of the item through the mashing up of data from various sources, Linked Data-aware applications are provided with references to URIs from which related data may be retrieved. Similar principles are followed to link items such as books and pubs to corresponding entries in external data sets.

[35] http://www.faviki.com accessed 27/11/2013
[36] http://readwrite.com/2008/05/26/semantic_tagging_with_faviki accessed 27/11/2013
[37] http://dbpedia.org/page/Semantic_Web accessed 26/11/2013
[38] http://revyu.com/ accessed 26/11/2013

2.2.4. Summary

In the previous sections, some emblematic applications using Linked Data were mentioned. These applications are now compared in **Table 1**.

Name	Category	Domain	Size
TWC LOGD	Portal	Government	6.4×10^9
U.S. Census	Portal	Geographic	1×10^9
BBC Music/ Programmes	Portal	Media	10×10^6
UAd Analyser	Expert System	User-generated Content	15×10^3
LinkedGeoData	Recommender application	Geographic	3×10^9
RKB Explorer	Portal	Publications	60×10^6
Sig.ma	Special purpose application - Search	Media	200×10^3
DBpedia Mobile	Recommender application	Geographic	409×10^6
Faviki	Special purpose application - Tagging	User-generated Content	52×10^3
Revyu	Recommender application	User-generated Content	20×10^3

Table 1. Comparison table for the Linked Data applications discussed on the previous section

For the previous table, the comparison criteria were:

- **Category** - The categories' range resultant of Lidia Rovan's et al [52] research, which was an analysis of Semantic Web solutions resulting in a categorization of Semantic Web applications;
- **Domain** – The domain of the data set used by the application in the LOD cloud[39,40];
- **Size** - Size of the data sets used by the application (represented by the number of RDF triples)[41,42];

[39] http://lod-cloud.net/versions/2011-09-19/lod-cloud_colored.png accessed 12/12/2013
[40] http://lod-cloud.net/state/ accessed 12/12/2013
[41] http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets/ Statistics accessed 14/12/2013
[42] http://www.w3.org/wiki/DataSetRDFDumps accessed 14/12/2013

In **Table 1** it is possible to see that governments (in this case U.S. and U.K.) are increasingly sharing very large amounts of open information. They do so by exposing it through Linked Data and making it accessible through an infrastructure that provides secure, customizable, personalized, integrated access to dynamic content from a variety of sources, in a variety of source formats, wherever it is needed – also called a Semantic Web Portal. Similarly, BBC's Music and Programmes sitealso chose this approach to enhance its media data (Music and Programmes) and promote reusability and integration. On the other hand, RKB Explorer make use of several data sets on the LOD cloud (Like ACM's publications and Metoffice weather forecasts data) in order to dynamically show the user information from various sources in an integrated exploration interface.

In a different way, systems like UAd Analyser - A market researcher's to trace discussions, Faviki and Revyu focus more their attention on user-generated content, thus allowing: the analysis and decision making based on discussion forum's data; the enrichment of DBpedia Mobile's concept's metadata information through tagging; and the interlinking between Linked Data and external sites like IMDB based on user-given input.

Furthermore, applications like LinkedGeoData and DBpedia Mobile are more focused on geographic data and both are systems in which the user provides recommendations as inputs (in this case its location through GPS), which the system then aggregates and directs to appropriate recipients (recommending the user the interest points near him geographically).

Finally, Sig.ma is less focused on the social interaction but more on searching all the information available about a user-given input from different sources and show it to the user already filtered. It uses semantic technology to improve search results.

2.3. Life Sciences Ontologies and Data Repositories

In this section some examples of ontologies for plants are addressed, followed by several examples about data repositories for plants using relational databases, as well as, repositories based on Linked Data in the Life Sciences domain.

2.3.1. Ontologies for Plants

As ontologies are commonly used to structure the knowledge in Biology domain, some examples of plant ontologies will be addressed throughout this section.

Plant Ontology

The Plant Ontology is an example of an ontology which describes not only a plant's anatomy and morphology, but also its development stages. It has the goal to "establish a semantic framework for meaningful cross-species queries across gene expression and phenotype data sets from plant genomics and genetics experiments"[43].

[43] http://www.plantontology.org/ accessed 23/12/2013

Plant Trait Ontology

Plant Trait Ontology[44] is a controlled vocabulary to define each plant trait as a unique feature, characteristic, quality or phenotypic feature of a developing or mature plant, or a plant part. Examples are glutinous endosperm, disease resistance, plant height, photosensitivity, male sterility, etc.

Experimental Factor Ontology

The Experimental Factor Ontology (EFO) [41] models the experimental variables by providing information on gene expression patterns under different biological conditions. The ontology has been developed to increase the richness of the annotations that are currently made in the ArrayExpress repository, to promote consistent annotation, to facilitate automatic annotation and to integrate external data. The ontology describes cross-product classes from reference ontologies in area such as disease, cell line, cell type and anatomy (**Figure 12**).

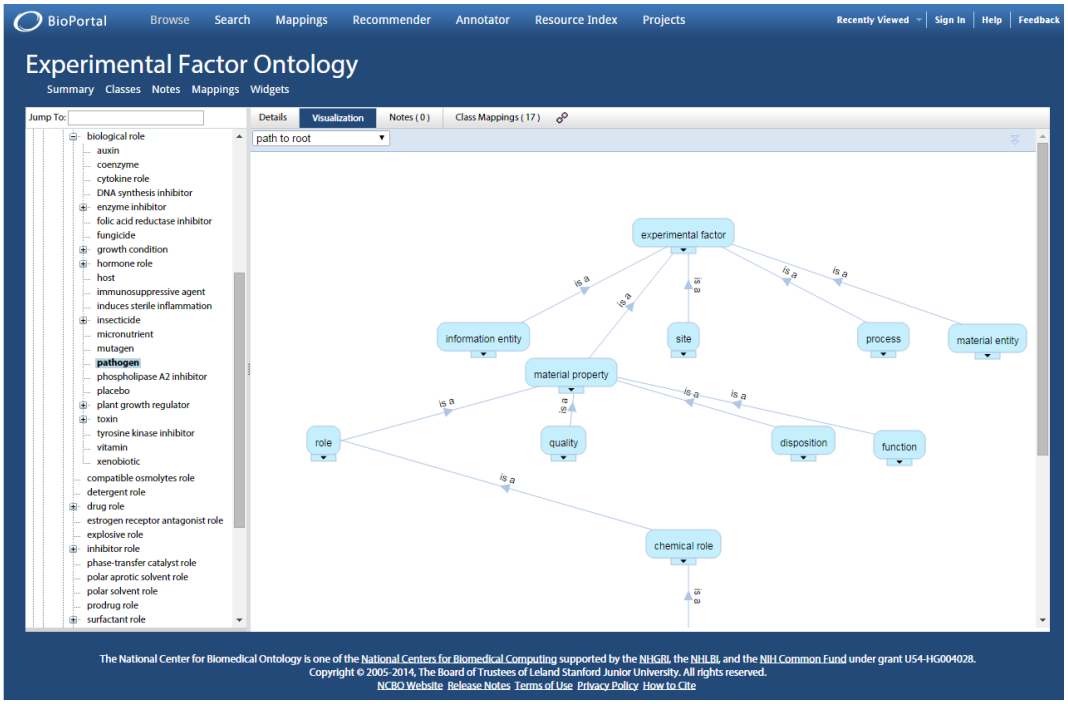


Figure 12. Experimental Factor Ontology visualization on the NCBI's BioPortal

BioAssay Ontology

The BioAssay Ontology (BAO)[45] describes chemical biology screening assays and their results including high-throughput screening (HTS[46]) data for the purpose of categorizing assays and data analysis. It has been designed to accommodate multiplexed assays and is an extensible and highly expressive description of biological assays making use of descriptive logic based features of the OWL lan-

[44] http://bioportal.bioontology.org/ontologies/PTO?p=summary accessed 23/12/2013
[45] http://bioassayontology.org/ accessed 23/12/2013
[46] http://www.scripps.edu/florida/technologies/hts/ accessed 23/12/2013

guage. Finally, all its main components include multiple levels of sub-categories and specification classes, which are linked via object property relationships forming an expressive knowledge-based representation.

Summary

Although exist these and other ontologies describing developmental and anatomical characteristics of plants, their foremost concern is the description of experimental design, hypothesis testing and the ultimate goal of the experiments [43].

2.3.2. Data Repositories for Plants

Repositories and databases have always been at the core of every storage information system's infrastructure as the means to organize a collection of data. This data is typically organized to model aspects of reality to support processes requiring information. The Biology area is not an exception and several repositories exist nowadays containing valuable information about multiple subjects. In this section some of that repositories will be addressed.

PlantFiles

A more general-public oriented repository, PlantFiles[47], is a community built solution for gathering information about plants. It contains detailed information and photos of over 207,700 different plants. Also, it allows the search of a plant by its common or botanical name or even by their characteristics (height, hardiness, etc.). Finally supports the browsing through hundreds of popular cultivars. Every user can propose new data which is then evaluated by more experienced gardener's and submitted on the repository if is valid.

WeedUS

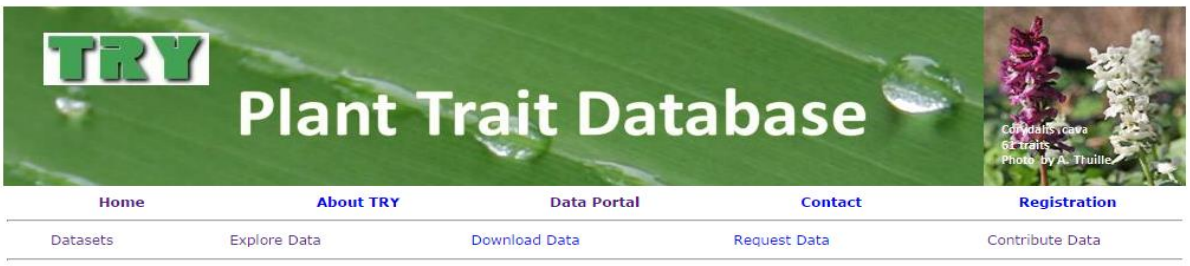
WeedUS[48] provides the most current and comprehensive compilation of plants that are invading natural areas in the United States affecting natural ecosystems. Data is gathered from several sources including publications, reports, surveys, and personal observations and is based on the observations and expert opinions of botanists, ecologists, invasive species specialists, and other professionals. Some applications of the repository are: to display state and regional level occurrence information for use in mapping occurrences of ecologically important invasive plant; ability to prevent and manage an invasive plant spread, therefore predicting a potential breakout.

TRY

The TRY repository [38] gathers plant trait data (the morphological, anatomical, physiological, biochemical and phenological characteristics of plants and their organs). This data represents raw materials that are used by many researchers from evolutionary biology, community and functional ecology to

[47] http://davesgarden.com/guides/pf/ accessed 23/12/2013
[48] http://www.invasive.org/weedus/distribution.html accessed 23/12/2013

biogeography. TRY gathers information from several databases worldwide and therefore creating a central repository where all information about plant traits is gathered (**Figure 13**).



Data Explorer

Bark thickness

TraitID: 24
Number of Observations: 183
Number of geo-referenced Observations: 183
Number of Species: 52

Download table Download

ObsdataID	ObservationID	AccSpeciesID	AccSpeciesName	Lon	Lat	Dataset	FirstName	LastName	Institution
5286959	925260	180	Acacia ataxacantha	32.0414722222222	28.0674722222222	South African Woody Plants Database (ZLTP)	William	Bond	University of Cape Town
5286895	925257	180	Acacia ataxacantha	32.0414722222222	28.0674722222222	South African Woody Plants Database (ZLTP)	William	Bond	University of Cape Town
5284944	925121	180	Acacia ataxacantha	32.0414722222222	28.0674722222222	South African Woody Plants Database (ZLTP)	William	Bond	University of Cape Town
5284926	925119	180	Acacia ataxacantha	32.0414722222222	28.0674722222222	South African Woody Plants Database (ZLTP)	William	Bond	University of Cape Town
5284908	925117	180	Acacia ataxacantha	32.0414722222222	28.0674722222222	South African Woody Plants Database (ZLTP)	William	Bond	University of Cape Town

Figure 13. Sample data in TRY for the Bark thickness plant trait

PLEXdb

The PLEXdb (Plant Expression Database) is a unified gene expression resource for plants and plant pathogens. As a repository, it allows "leveraging highly parallel expression data with seamless portals to related genetic, physical, and pathway dataworks" [14]. Also, it allows users to perform complex analyses quickly by providing methods to track how gene expression changes across many different experiments (**Figure 14**). Finally, it is complementary and synergistic to other expression data archives such as NCBI-GEO (Gene Expression Omnibus)[49] and ArrayExpress[50] which are public functional genomics data repositories and act as central data distribution hubs. All these repositories are compliant with MIAME (Minimum Information About a Microarray Experiment), which is a standard that provides a conceptual framework for core information to be captured from most microarray experiments.

[49] http://www.ncbi.nlm.nih.gov/geo/ accessed 23/12/2013
[50] http://www.ebi.ac.uk/arrayexpress/ accessed 23/12/2013

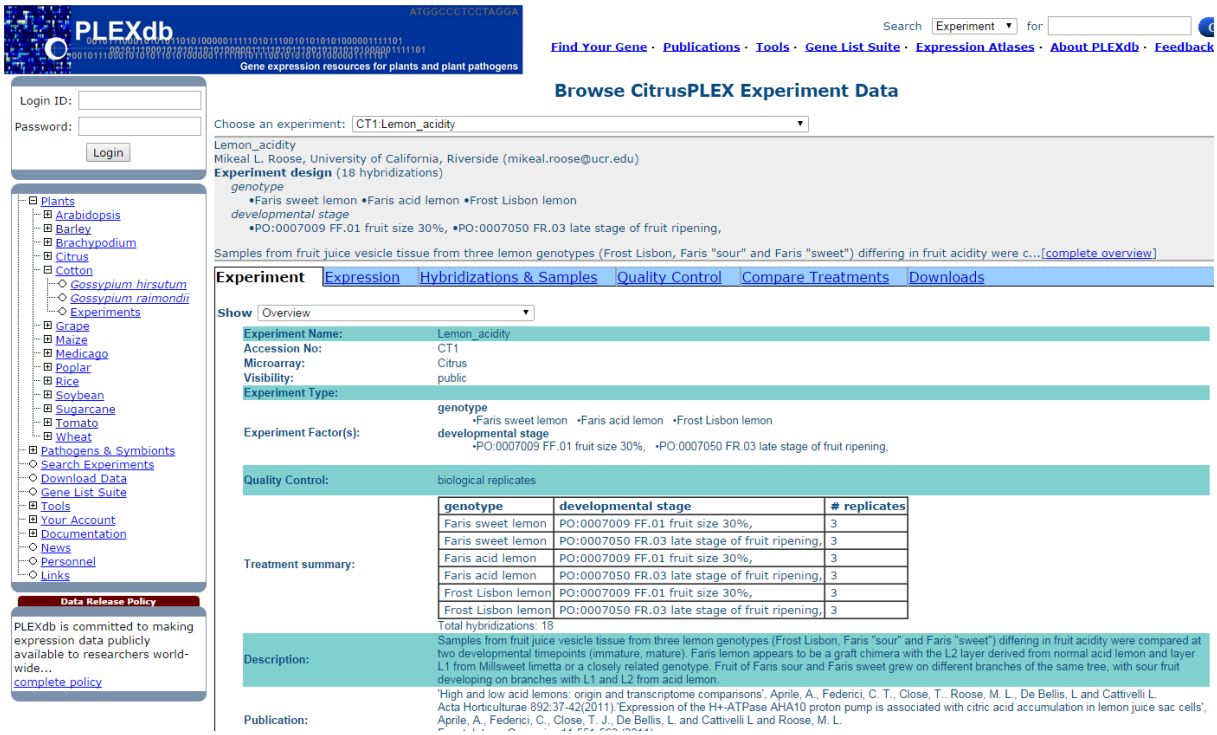


Figure 14. PLESXdb - Example of data of an experiment on lemon acidity

Summary

Although all these repositories structure all information about plants, they do so using relational databases which can sometimes duplicate information and are not designed to interlink resources. New approaches must be taken into account in order store the data in a way that is easy to be reused and linked to other resources either inside the same repository or external sources.

2.3.3. Linked Data Repositories in Life Sciences

"The growing abundance of data on the Web has intensified the need to develop new approaches to manage and integrate it" [49]. In order to overcome these challenges, more and more organizations are interested in data integration abilities that come from Semantic Web, such like include the aggregation of heterogeneous data using explicit semantics and the expression of rich and well-defined models for data aggregation and search. The reason is because ads to existing web standards and practices encouraging clearly specified names for things, classes, and relationships, organized and documented in ontologies, with data expressed using standardized well-specified knowledge representation languages [49].

Some mature examples of Linked Data repositories in the Life Sciences domain are shown in the next section.

BioLOD

BioLOD[51] (Broadly Integrated Ontological Linked Open Data) is a database that provides over 6,800 downloadable OWL/RDF graph files of mutually linked public biological data organized as a Semantic Web using standardized formats of the W3C LOD project. BioLOD mines numerous semantic links from original databases and re-classifies them into graph files based on ontology classifications. Relationships between the files are mutually and clearly referenced so it is easy to find other files associated by semantic links included in detailed data instances. BioLOD intensively surveyed both forward and reverse semantic-link relationships from 36 databases for humans and mice, 33 databases for plants and 16 databases related to proteins. BioLOD summarizes this information as archive files available for download in various useful formats. The BioLOD database uniquely provides Linked Open Data annotated contextually with biological vocabulary and supports visualization services to browse LOD data through SciNetS.org, repository services to deposit users' LOD through LinkData.org and SPARQL endpoint service for BioLOD data is through BioSPARQL.org [45].

Bio2RDF

Bio2RDF[52] is an open-source project that promotes a simple convention to integrate diverse biological data using Semantic Web technologies. It consists of scripts that automatically download and convert well known biological data sets into the RDF from their original formats, whether it be flat-files, tab delimited files, XML or SQL. Using SPARQL, Bio2RDF Linked Data can be uniformly explored and queried. Bio2RDF attempts to capture the intended meaning serialized by the original data providers in both content and structure. Each Bio2RDF dataset has a unique Linked Data vocabulary and topology and does not attempt to marshal the data into a common schema. It relies on a set of basic guidelines to produce syntactically interoperable Linked Data across all datasets. The infrastructure provides a federated network of SPARQL endpoints and provisions the community with an expandable global network of mirrors that host Bio2RDF datasets [6].

DrugBank

The DrugBank[53] linked repository [39] is a unique bioinformatics and cheminformatics resource that combines detailed drug (i.e. chemical, pharmacological and pharmaceutical) data with comprehensive drug target (i.e. sequence, structure, and pathway) information. It not only has a systematic collection of drug–protein interactions but also contains associations of proteins with consensus genetic annotations, such as UniProt[54]. The DrugBank database has been expanded by around 60% since its release to include further FDA-approved and experimental drugs, as well as data for almost 1,000 additional drug–target interactions. The database currently contains information on almost 6825 experimental, approved and withdrawn drugs, with up to 107 data fields for each drug that contain information including

[51] http://BioLOD.org/ accessed 23/12/2013
[52] http://bio2rdf.org accessed 23/12/2013
[53] http://www.drugbank.ca accessed 23/12/2013
[54] http://www.uniprot.org accessed 23/12/2013

current indications, documented drug–target interactions, target protein accession numbers and pharmacological actions.

Diseasome

Diseasome[55] [25] is a triple store that collects all known human disorder/disease gene relationships, which is presented to the user through an innovative graph-oriented explorer. It uses the Human Disease Network dataset and allows intuitive knowledge discovery by mapping its complexity. Currently, it publishes a network of 4,300 disorders and disease genes linked by known disorder-gene associations for exploring all known phenotype and disease gene associations, indicating the common genetic origin of many diseases. The list of disorders, disease genes, and associations between them was obtained from the Online Mendelian Inheritance in Man[56] (OMIM), a compilation of human disease genes and phenotypes.

LinkedCT

The Linked Clinical Trials[57] (LinkedCT) [30] project is an information repository for locating federally and privately supported clinical trials for a wide range of diseases and conditions. Consequently, it is a rough guide to the level of testing that various treatments have had by linking the drug and disease data sets mentioned in the previous sections to individual clinical interventions in LinkedCT, enabling a path between the drugs, affected genes, and trials relating to the drugs. For this to be possible the data exposed by LinkedCT is generated by not only transforming existing data sources of clinical trials into RDF, but also discovering links between the records in the trials data and several other data sources. These semantic links are discovered through approximate string matching and ontology-based semantic matching techniques.

LinkedCT shifts the responsibility of data integration to data providers by using a Linked Data approach. This is a much more efficient approach, as the data providers are the individuals who understand their data best. It also means that the integration only has to happen one time.

Sider

Sider[58] Side Effect Resource it's the only resource in machine-readable form (despite the importance of research on drugs and their effects) that extracts the information from public documents and package inserts and stores it, creating interlinked information on marketed medicines and their recorded adverse drug reactions (side effects) [12]. The available information include side effect frequency, drug and side effect classifications as well as links to further information, for example drug–target relations. Sider covers a total of 888 drugs and 1450 distinct side effects. It contains information on frequency in patients for one-third of the drug–side effect pairs.

[55] http://diseasome.eu accessed 23/12/2013
[56] http://www.omim.org/ accessed 23/12/2013
[57] http://linkedct.org/ accessed 23/12/2013
[58] http://sideeffects.embl.de/ accessed 23/12/2013

BioGateway

Biogateway is an integrated system offering a user interface the system can be explored/queried using SPARQL and a data backend which is composed by an RDF repository that holds the graphs corresponding to the integrated data.

The Biogateway combines information from various resources from the entire set of the OBO foundry candidate ontologies[59], the whole set of GOA files[60], UniProt, the NCBI taxonomy[61] as well as in-house ontologies. Also, it provides a single entry point for exploiting these ontologies and constitutes a step towards a Semantic Web integration for biological data[62]. It aims to support Systems Biology approaches by combining Semantic Web technologies which in turn enable data-driven research. The Semantic Web approach that has been taken enhances data exchange and integration by providing a standardized mechanism for interrogating such system [3].

Summary

All the previous solutions described to expose interlinked biological data are now summarized in the following **Table 2**: category of application, software used to publish the data, if it has an event[63] system, ability to do reasoning[64], native SPARQL endpoint and if it has a web interface.

	Category	Software	Events	Reasoning	Native SPARQL Endpoint	Web Driven
BioLOD	Biological Data	Proprietary	No	Yes	No	Yes
Bio2RDF	Life Sciences	OpenLink Virtuoso	Yes	Yes	Yes	No
DrugBank	Drug Effects	Proprietary	No	Yes	Yes	Yes
Diseasome	Human Diseases	Sesame	Yes	No	Yes	Yes
LinkedCT	Clinical Trials	Jena	Yes	Yes	No	Yes
Sider	Side Effects	Mulgara	No	Yes	Yes	No
BioGateway	Biological Data Integration	OpenLink Virtuoso	Yes	Yes	Yes	No

Table 2. Comparison of Linked Data Repositories

[59] http://www.obofoundry.org/ accessed 23/12/2013
[60] http://www.geneontology.org/GO.downloads.annotations.shtml accessed 23/12/2013
[61] http://www.ncbi.nlm.nih.gov/taxonomy accessed 23/12/2013
[62] http://www.semantic-systems-biology.org/biogateway accessed 23/12/2013
[63] Notifications given when changes occur
[64] Ability to infer logical consequences from a set of data

Although they have very distinct categories, all the reviewed approaches to build Linked Data repositories use open source triple store software (except for DrugBank and BioLOD which built their own):

Jena - Jena[65] is a java framework for building Semantic Web applications. It implements APIs for dealing with Semantic Web building blocks such as RDF and OWL.

Sesame - Sesame[66] is an open source framework for storage, inference and querying of RDF data. Sesame matches the features of Jena with the availability of a connection API, inference support, availability of a web server and SPARQL endpoint. Like Jena, it provides support for multiple back ends like MySQL and PostgreSQL.

OpenLink Virtuoso - Virtuoso[67], is a native triple store available in both open source and commercial licenses. It provides command line loaders, a connection API, support for SPARQL and web server to perform SPARQL queries and uploading of data over HTTP. A number of evaluations have tested virtuoso and found it to be scalable to the region of 1B+ triples. In addition to this, a Virtuoso provides bridges for it to be used with Jena and Sesame.

Mulgara - Mulgara[68] is a native RDF triple store written in Java. It provides a Connection API that can be used to connect to the Mulgara store. Being a native triple store it has a 'load' script which can be used to load RDF data into the triple store. In addition to supporting SPARQL queries through the connection API, these can be performed through the TQL shell. The TQL shell[69] is a command line interface that allows queries on models present in the store.

All the presented solutions can natively respond to queries made with SPARQL 1.0 through their SPARQL endpoints (except Jena that uses Fuseki[70] for that purpose) but, in none of them, SPARQL queries can be filtered by access control at the statement level[71]. Also, Jena, Mulgara and Virtuoso are the only ones that can do reasoning through their built-in rule engines. Moreover, only Sesame, Jena and Virtuoso are able to provide notifications when something has changed.

Finally, though OpenLink Virtuoso has higher scalability and overall performance[72], it's not fully open source and not built for web development. Sesame doesn't have these setbacks and is widely focused on the Web, but lacks a reasoning engine. On the other hand, while Jena doesn't support a native SPARQL Endpoint it has architecture to deal with web development seamlessly backed up by a rule engine for reasoning.

[65] http://jena.apache.org/ accessed 30/12/2013
[66] http://notes.3kbo.com/sesame accessed 30/12/2013
[67] http://virtuoso.openlinksw.com/ accessed 30/12/2013
[68] http://www.mulgara.org/ accessed 30/12/2013
[69] http://code.mulgara.org/projects/mulgara/wiki/TQLUserGuide accessed 30/12/2013
[70] http://jena.apache.org/documentation/serving_data/ accessed 30/12/2013
[71] http://www.garshol.priv.no/blog/231.html accessed 30/12/2013
[72] http://www.biomedcentral.com/1471-2105/13/S1/S3 accessed 30/12/2013

2.4. Open Research Issues

By publishing and interlinking various data sources on the internet, the Linked Data community has created a clear starting point for the Web of Data and a stimulating workplace for Linked Data technologies to grow. However, to address the ultimate goal of being able to use the internet like a single global database, various remaining challenges must be overcome.

2.4.1. Link Maintenance

The content of Linked Data sources is constantly changing. Either the data about new entities is added, or outdated data is changed or removed. Nowadays, RDF links between data sources are updated only sporadically which leads to dead links pointing at URIs that are no longer maintained and to potential links not being established as new data is available. On the other hand, the architecture of the World Wide Web is, in principle, tolerant to dead links, but having too many of them leads to a large number of unnecessary HTTP requests by client applications [10]. Proposed approaches to this problem start with the recalculation of links at regular intervals using frameworks such as Silk [37] or LinQL [29], to data sources publishing update feeds, or even informing link sources about changes via subscription models to central registries such as Ping the Semantic Web[73] which keeps track of new or changed data items.

2.4.2. Licensing

Applications that consume data from the internet must be able to access the specifications of the terms under which data can be reused and republished. Therefore, the availability of appropriate frameworks for publishing such specifications is an essential requirement in encouraging data owners to participate in the Web of Data, and in providing assurances to data consumers that they are not infringing the rights of others by using data in a certain way [10]. With this in mind, initiatives such as the Creative Commons[74] have provided a framework for open licensing of creative works, reinforced by the notion of copyright. However, as discussed by Paul Miller et al [44], copyright law is not applicable to data, which from a legal perspective is also treated differently across jurisdictions. Consequently, frameworks such as the Open Data Commons Public Domain Dedication and License[75] should be adopted by the community to provide clarity in this area.

2.4.3. Privacy

The final goal of Linked Data is to be able to use the internet like a single global database [10]. The realization of this vision would provide benefits in many areas but will also aggravate dangers in others. One problematic area is the opportunities created to violate privacy that arise from integrating data from

[73] http://www.programmableweb.com/api/ping-the-semantic-web accessed 15/12/2013
[74] http://creativecommons.org/ accessed 15/12/2013
[75] http://opendatacommons.org/licenses/pddl/1.0/ accessed 15/12/2013

distinct sources. Protecting privacy in the Linked Data context is likely to require a combination of technical and legal means together with a higher awareness of the users about what data to provide in which context. Interesting research initiatives in this domain are Weitzner's work on the privacy paradox and information accountability [57].

2.4.4. User Interfaces and Interaction Paradigms

Possibly the key benefit of Linked Data from the user's point of view is the delivery of integrated access to data from multiple distributed and heterogeneous data sources. By definition, this may involve integration of data from sources not explicitly selected by users, as to do so would likely incur in an unacceptable cognitive overhead. Although the applications described in **Section 2.2** demonstrate promising tendencies in how applications are being developed to exploit Linked Data, several challenges remain in understanding appropriate user interaction paradigms for applications built on data assembled dynamically in this fashion. For example, while hypertext browsers provide mechanisms for navigation forwards and backwards in a document-centric information space, similar navigation controls in a Linked Data browser should enable the user to move forwards and backwards between entities, thereby changing the focal point of the application [10]. Linked Data browsers will also need to provide intuitive and effective mechanisms for adding and removing data sources from an integrated, entity-centric view. Sig.ma (explained in **Section 2.2.2**), gives an indication on how such functionality could be delivered. Nevertheless, other interface approaches should be considered when data sources are in the numbers of thousands or millions.

2.4.5. Trust, Quality and Relevance

An important concern for Linked Data applications is how to ensure the most relevant and/or appropriate data is presented to the user according to its. For example, in scenarios where data quality and trustworthiness are of vital importance, how can this be determined heuristically? A proposed approach for this problem was developed by Christian Bizer et al [9] and uses rating-based techniques to heuristically assess the relevance, quality and trustworthiness of data. Also, algorithms like PageRank[76] will likely be important in determining the popularity or significance of a particular data source, as a proxy for relevance or quality of the data. Still, such algorithms will need to be adapted to the linkage patterns that emerge on the Web of Data.

The problem of how to represent the provenance and trustworthiness of data drawn from many sources into an integrated view in an interface is a significant challenge (as explained more thoroughly in the previous section). Some approaches propose that browser interfaces should be enhanced with a quick way to support the user in assessing the reliability of information encountered on the internet. Whenever a user encounters a piece of information that they would like to verify, pressing, for example a button, would produce an explanation of the trustworthiness of the displayed information. This hasn't been done yet, however existing developments such as WIQA [9] and InferenceWeb [42] can contribute

[76] http://en.wikipedia.org/wiki/PageRank accessed 15/12/2013

to work in this area by providing explanations about information quality as well as inference processes that are used to derive query results.

3. Proposed Repository Solution

In this section is described the main goals and requirements of the proposed repository solution, followed by the repository data structure and architecture chosen, detailing each one of its components.

3.1. Repository Goals and Requirements

According to the previously described problem, and together with the KDBIO team, it was concluded that the infrastructure that will hold data results from biological experiments, must:

- Gather data from multiple sources into a single database, so a single query engine can be used to present data;
- Provide data integrity, removing the possibility for redundant information;
- Convert all imported data to an uniform format;
- Provide the means to allow reasoning and data analysis for internal enrichment by adding extra semantics;
- Expose the data through standard approaches so it can be accessed by external entities either human or machines;
- Provide a user interface to enable a user without deep ontology knowledge to manage the data;
- Log every transaction to enable data recovery and understand the reason behind any problems that might arise.

As requirements specify the properties a system needs to fulfill according to its objectives and scopes, they must result from the defined goals of the system and their analysis. Therefore, the outcome of this process was the definition of the following requirements:

[Req1]. Import a data set: It must be able to import a data set in Excel format.

[Req2]. Manage repository content: It must be possible to, through an intuitive interface, manage all the entities within the repository: ontologies, projects, excels and users.

[Req3]. The data as to follow a well-defined structure: All the data must be stored according to a well-defined structure responsible for the representation of the biological experiments data.

[Req4]. Transaction log: The system has to save all the operations done, including user that perform it and at what time.

[Req5]. Data exposure: All the data in the repository must be available to external services in a normalized way.

Hence, with goals and requirements set, it is now possible to define the basis for the repository infrastructure that will manage data results from biological experiments.

3.2. Repository Data Structure

This section addresses, in first place, the core data model, which is based in the concept of using ontologies to structure the repository data, and in second, the repository domain model, detailing its main concepts.

3.2.1. Ontologies as the core data model

As already said in **Section 1.2**, the contemporary biological experimental studies produce a great wealth of heterogeneous and interdependent data that are difficult to reproduce. Due to this, the KDBIO Group together with ITQB-UNL/IBET and ISEL teams developed an ontology with the purpose of preserve the semantic relationships between the entities represented in it. The ontology developed is composed by three distinct realms:

- **Biological** - referring to biological material or to manipulations;
- **Physical** - referring to non-living material;
- **Data** - referring to informational concepts and their manipulation.

Each one of these three distinct realms include experimental products, their relations and the protocols describing their manipulation [43].

Therefore, to gather biological experiments data maintaining its semantics the repository was developed to use ontologies as the core data model. This means that is only possible to insert data through an ontology.

3.2.2. Repository Domain Model

The domain model, representing the core entities responsible for the control and management of the repository, is described as a simple UML class diagram (**Figure 15**). The core concepts of the domain model are:

- **Project** - Aggregates all the information related to one experiment and it must have an associated ontology that serves as the data model for the information stored within. All the experimental data is persisted through Jena's TDB and the metadata through file system.
- **Ontology** – The information stored within a Project is stored according to an associated ontology. Thus, the Ontology is the data model of the information stored within a Project. It has any number of versions that correspond to the evolution of the ontology over the time.
- **Ontology Metadata** – Is the metadata of the uploaded ontology files (OWL or RDF/XML) containing information like the path for the file, uploaded time, who upload the file, if the version is currently in use or not, among others.

- **Ontology Class** – Encloses the information about the ontology classes existing in the ontology file, containing information about their parent-classes, sub-classes, datatype and object properties and restrictions.
- **Datatype Property** – Represents the information about the datatype properties that exists in the ontology file.
- **Object Property** – Represents the information about the object properties that exists in the ontology file.
- **Ontology Restriction** – Represents the information about the ontology restrictions that exists in the ontology file.
- **Individual** – Represents the instance of an Ontology Class containing all of its properties filled with a certain value.
- **Excel Metadata** – Is the metadata of the imported Excel files including information like the path for the original Excel file, the path for the correspondent Turtle file (which contains all the information about a Project just like is stored in the repository), when it was imported and by whom, among others. It also contains the type of script to be chosen during the importation process.

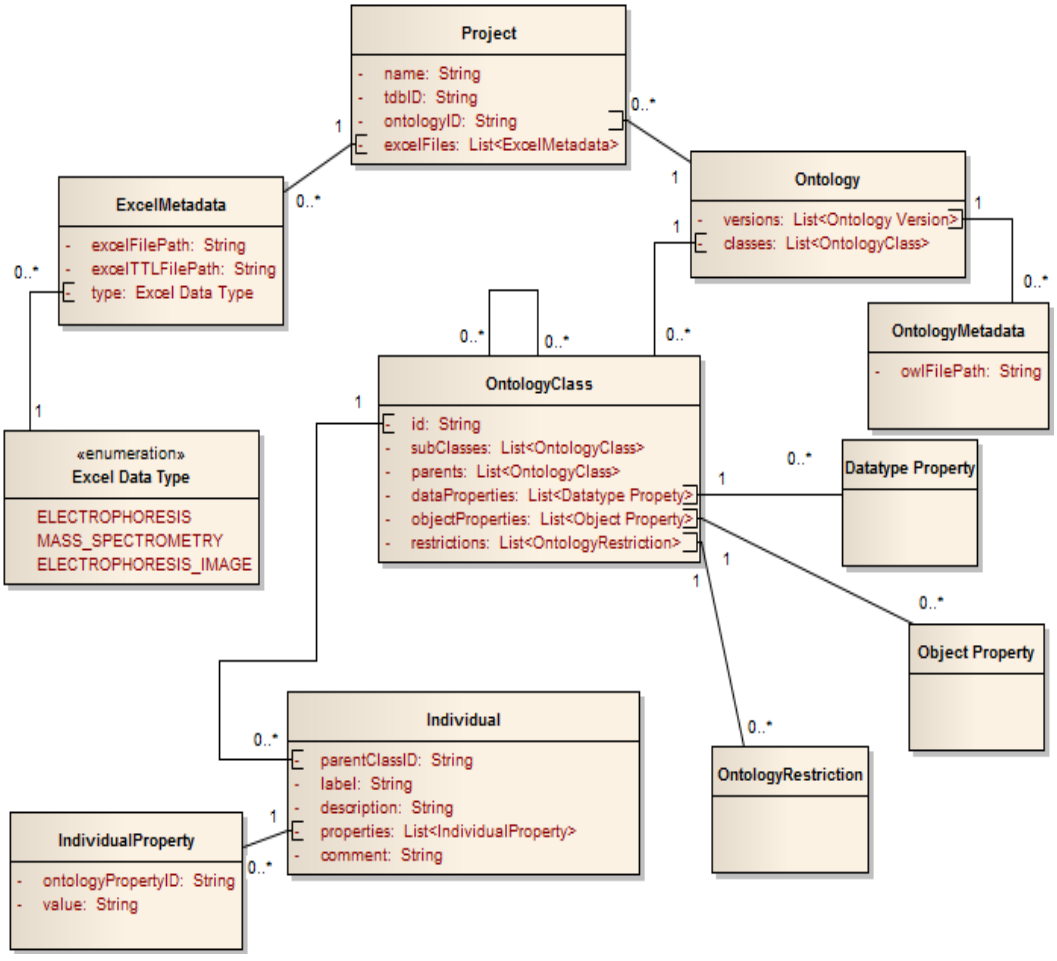


Figure 15. RDF Repository core domain model.

Based on this data model, the overall architecture of the system was developed and it is presented in next **Section 3.3**.

3.3. Repository Architecture

In order to solve the problem described in **Section 1.2**, it was developed an infrastructure to manage data from biological experiments using the frameworks Jena (to handle RDF data) and GWT (to the interface development). In **Figure 16** can be seen the architecture chosen for the RDF repository. On the left side of the diagram are present the Jena main components used in the implementation of this solution, showing how they interact with each other and which components of the developed architecture use them. On the right side, is present the architecture designed to store and gather biological experiments data. Additionally, the user can interact with all data through a user interface provided by the RDF repository.

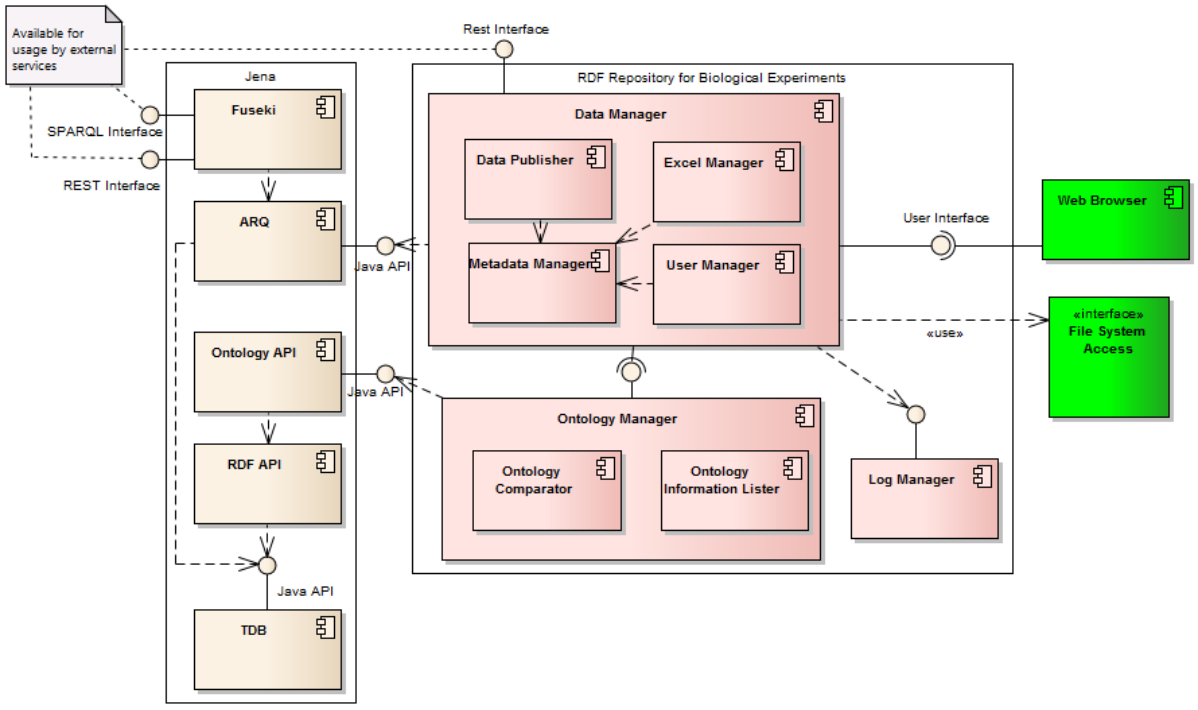


Figure 16. Architecture of the RDF Repository for Biological Experiments

Throughout **Section 3.3.1**, **Section 3.3.2** and **Section 3.3.3** all these components will be addressed in further detail.

3.3.1. Architecture

As shown in **Figure 16**, the architecture developed for the biological experiments repository is composed by the following components:

- **User Manager** – Responsible for the management of users and roles of the system;
- **Log Manager** – Responsible for the creation of a detailed daily log of every activity that occurs in the system, including the user that performed it and at what time (see **Appendix C**);

- **Data Manager** – Manages all creation/editing/deleting operations in the system and it is also responsible for the retrieval of the information from the repository.
    - **Metadata Manager** – Store the information about the entities required to organize the information system data in XML files. These entities are:
        - **Project** – A model that gathers all the information related to one experiment;
        - **Excel** – An Excel file containing biological experiments data and that will be imported into the repository;
        - **User** – A user of the system with a username, password and role;
        - **Ontology** – The metadata about the uploaded ontology file, like information regarding the location of the file, if it is an active version currently in use or not, when it was imported, among others.
    - **Excel Manager** – Manages the process of extract the data from an Excel document, preparing it to be ready for the publishing process by converting them to the standard format Turtle, accordingly with the ontology that is being used in the project where the import is taking place;
    - **Data Publisher** – Responsible for the publishing process that consists in taking the Turtle file prepared by the Excel Manager and run a validation process that asks the user to fill required datatype and object properties. All the changes and rectifications made by the user will be stored in the Turtle file. Once the validation is finished, the system allows the user to insert the imported data into the repository.
- **Ontology Manager** – Manages the ontologies and their versions. The ontologies can be imported in OWL or RDF/XML formats, but if the ontology imported is in OWL format this component has the ability to convert it to RDF/XML because Jena's API doesn't support OWL as an input format (the convert operation is done using the OWL API[77]). Once the OWL file is converted to a RDF/XML file, it can be stored in the file system and then loaded by Jena's **RDF API**. This component is composed by:
    - **Ontology Information Lister** – Each Project must has an associated ontology that will define how the data will be stored in the repository. Thus, this component is responsible for, to each Project, list all ontology classes and their correspondent datatypes and object properties, and generate the forms to create new instances;
    - **Ontology Comparator** – Manages the comparison between ontologies versions and shows the differences to the user;
- **User Interface** – All the data is accessible and can be managed by a web user interface that can be accessed through a web browser.

[77] http://owlapi.sourceforge.net/ accessed 07/10/2014

All these components together with the Jena components described on **Section 3.3.2** enable the system to meet all the requirements for the proposed solution.

3.3.2. Jena as a Semantic Web Framework

As described lightly in **Section 2.2.4**, Jena is a Java framework that enables the creation of Semantic Web applications through its main components:

- **RDF API** – An API that allows the creation or reading of a resource, which can be described in several formats like RDF/XML or Turtle, into a Java RDF Graph so it can be further manipulated.
- **Ontology API** – An API for ontology application development, independent of which ontology language it is being used. When working with an ontology in Jena, all the information is encoded in RDF triples stored in the RDF model. It also provides classes and methods that make it easier writing programs that manipulate the underlying RDF triples.
- **TDB** - A high performance RDF storage system that can be accessed and managed with the provided command line scripts or through the Jena API. When accessed using transactions, a TDB dataset is protected against corruption, unexpected process terminations and system crashes.
- **ARQ** – A query engine that supports the SPARQL 1.1 language[78], enabling not only the retrieval of data from a resource loaded from a file using the RDF API, but also from a specific model loaded on the TDB.
- **Fuseki** - A SPARQL server that provides REST-style SPARQL HTTP Update, SPARQL Query, and SPARQL Update using the SPARQL protocol over HTTP.

Therefore, Jena deals with RDF through its fundamental class, the *Model*, designed to have a rich API, with many methods intended to make it easier to write RDF-base programs and applications. A Model can be sourced with data from local files, databases, URLs or a combination of these, and also in triples serialized in formats like RDF/XML, Turtle, among others[79]. Additionally, to deal with ontologies Jena uses an extension of the *Model* class, the *OntModel*, which provides extra capabilities for handling ontologies and offers reasoning services. Finally, it has the ability to store the RDF data in TDB, Jena's native triple store that can be queried using SPARQL. All of these components where used widely in the proposed solution to achieve the best data quality, performance in data manipulation and querying.

3.3.3. Google Web Toolkit as Web Development Framework

Based on the previous experience with GWT[80], it was chosen as the main framework to the solution development. This prototype's architecture is composed by a server side responsible for retrieving the information from Jena and the TDB, which then sends it to the client side, where it is processed and showed in the user's browser. This type of architecture allows for a good performance, saving bandwidth

[78] http://www.w3.org/TR/sparql11-query/ accessed 01/10/2014
[79] http://en.wikipedia.org/wiki/RDF/XML accessed 09/05/2014
[80] http://www.gwtproject.org/ accessed 01/10/2014

for data exchange only, and being the UI fully loaded on the client side. Also it provides an easy deployment and cross-browser support. To allow a more attractive *look and feel*, it was also used an *open-source* GWT widget extension named GXT[81]. This enabled to create the web interface faster using GXT's widgets, which can be easily changed according to the user's needs.

[81] http://www.sencha.com/products/gxt/ accessed 01/10/2014

4. Results

To prove the repository concept, the Centro de Investigação das Ferrugens do Cafeeiro of Instituto de Investigação Científica Tropical (IICT[82]) together with Instituto de Tecnologia Química e Biológica/Instituto de Biologia Experimental Tecnológica (ITQB-UNL[83]/IBET[84]) provided experimental data about *Coffea Arabica* plants (developed in a greenhouse environment) that were infected with the fungus *Hemilea Vastatrix* (casual agent of coffee leaf rust) to identify potential candidate biomarkers[85] for resistance of coffee against coffee leaf rust.

During plant fungal infection, the plant triggers its response with impact at physiological, molecular and biochemical levels and, consequently, in the abundance or depletion of individual proteins. Thus, the identification of those proteins whose abundance varies across different conditions will enable the disclosures of the protein role in response to the infestation. Some of these proteins are known and biochemically characterized, but for the greater part, their identity is unknown and only partial information can be provided.

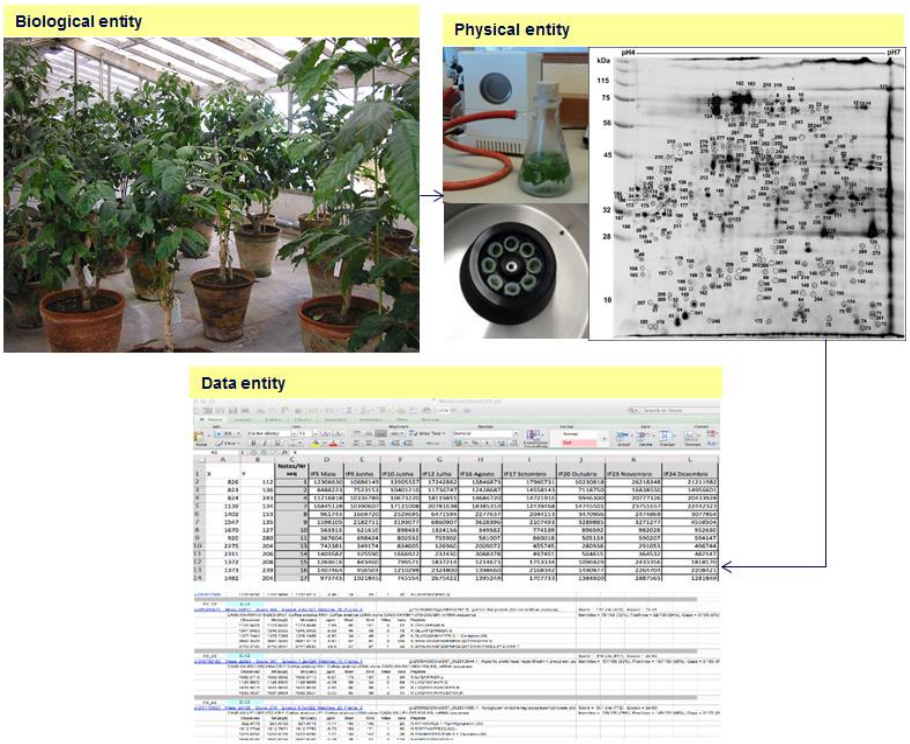


Figure 17. Coffee plant stress tests data gathering process. **Biological entity**: Growth conditions: Coffee plants growing in a green-house, IICT, Oeiras, PT. Biological samples, leaves, were collected at different times of the year; **Physical entity:** Extraction protocol (apoplast protein isololation from the collected leaves) and 2DE gel of the proteins from coffee leaf apoplastic fluid (numbers are the spotsID that were isolated from the gel); **Data entity:** For each spotID were associated the coordenates (x, y) in the gel and the volume and mass spectrometry of each spot allows the identification of the proteins.

[82] http://www2.iict.pt/ accessed 07/01/2014
[83] http://www.itqb.unl.pt/ accessed 07/01/2014
[84] http://www.ibet.pt/ accessed 07/01/2014
[85] In this context, biomarkers are the key protein which play an important role in the identification of plants that are at increased risk or resistant for the disease.

Therefore, as test-case it was used a section of coffee leaf rust assays which comprehends the proteome modulation of coffee leaf apoplastic fluid, using 2D electrophoresis (2DE). The data was provided through JPEG images (from the 2DE gels, **Appendix A**) the corresponding spreadsheets in Excel produced by gel analysis machines (**Appendix B**) and a spreadsheet with the protein identification. In **Figure 17** is possible to see an example of the data provided workflow according with the 3 realms of *Plant Experimental Assays Ontology* [27].

4.1. Ontology Management

This section describes how ontologies and their main features are managed, including the import of ontologies and their comparison.

4.1.1. Ontology Import

The system is not restricted to a single ontology and, on the other hand, an ontology can be composed by multiple ontologies which can also have multiple versions. Therefore, to import a new ontology into the system, it was created a concept where the user must create a set that will aggregate all the related ontologies and their versions. So, when the user clicks on the "Add Ontology" option what he's doing is to create a set by giving it a name, optionally add a description and finally, upload the ontology file. Then, if the ontology uploaded is composed by other ontologies, the system will ask for the upload of the missing dependent ontologies, and so on for all ontologies which are composed by others.

As referred in **Section 3.3.1**, if the file is in the OWL format then it is converted to RDF/XML using the OWL-API Java library. Finally, the file is saved on the file system and additional metadata related to the ontology version (like name, description, id of the ontology file, etc.) is stored in XML (see **Appendix D**).

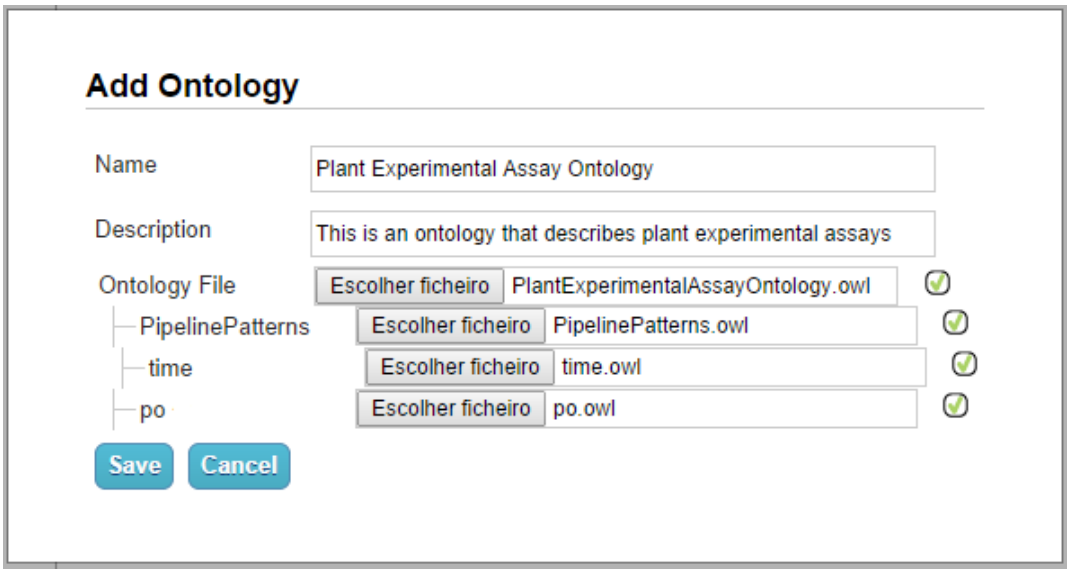


Figure 18. Ontology form to add a new ontology

In **Figure 18**, we can see an example of the form prompt to the user when adds an ontology. The name given to this ontology is *Plant Experimental Assay Ontology* and it has a description about it. We also see that the main ontology added, whose filename is *PlantExperimentalAssayOntology.owl*, is composed by two other ontologies, namely: *po* and *PipelinePatterns,* being the last one composed by the *Time* ontology.

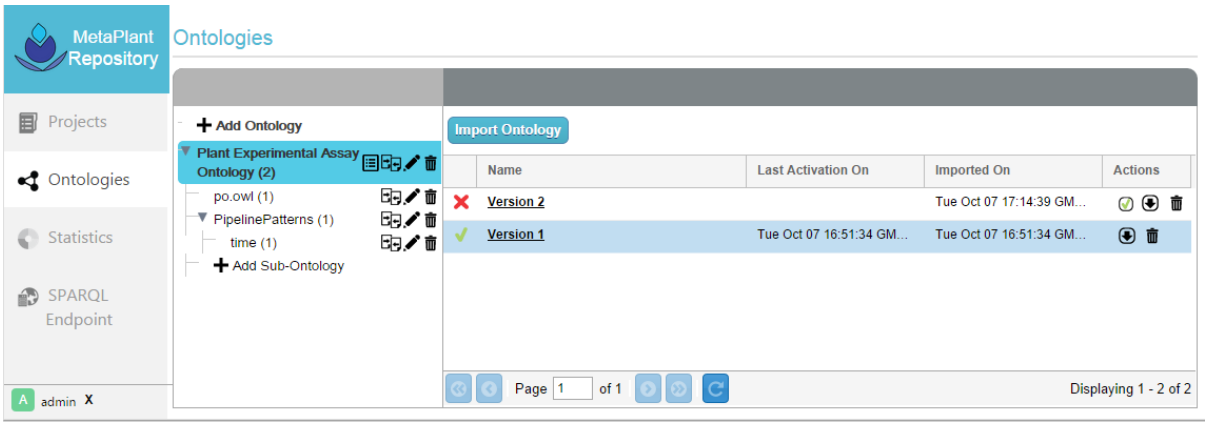


Figure 19. View of the repository ontologies and the existing versions of the *Plant Experimental Assay Ontology*

As previously referred in the beginning of this section, one of the reasons to create a set to aggregate related ontologies, was due to each ontology can has multiple versions. This way, the system keeps a list of all imported versions for each ontology and also provides information about when it was imported, if it is active, among others. An ontology active means that the data of all the projects in the repository using the ontology containing that version will be now managed using this active ontology's schema. Hence, only one version at time can be active.

In **Figure 19**, it can be seen in the left panel the *Plant Experimental Assay Ontology* set selected and, in the right pane, all the versions that were already imported into the system related to this specific ontology. In this panel we also perceive that the version of ontology currently in use is the *Version 1*, set active on 07 October, and *Version 2* is a new imported version, still inactive.

Finally, as an ontology can have one or more projects using it, the system maintains a list with the projects that are using it (**Figure 20**).

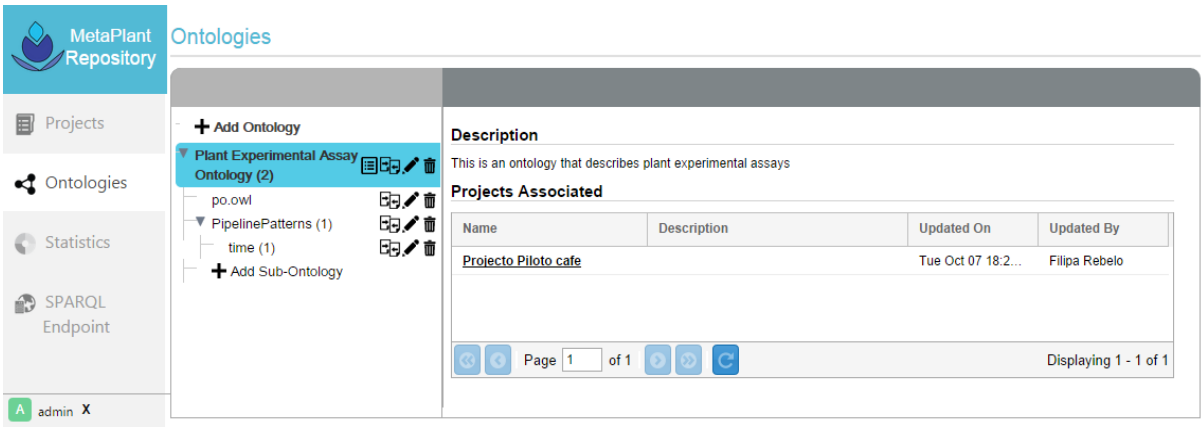


Figure 20. View of the projects associated to *Plant Experimental Assay Ontology*

4.1.2. Ontology Comparison

As described in previous **Section 4.1.1**, the system allows to import different ontologies versions. For that reason, it was realized that user needed a way to see the differences between two or more ontology versions to perceive how the changes can affect the data. Therefore, the system allows the user to compare consecutive ontologies versions through the creation of a timeline where it is possible to see the differences between the *version n* and the *version n+1*[86].

In **Figure 21** can be seen that in *Version 1* the class with the identifier *PEAO:000014* had the label *DataProcessing*, but now in *Version 2* that same label has changed to *DataProcessingV2*. This difference detection process is done by loading each ontology file onto Jena which does the comparison through its API. Finally, the results are shown in a before/after approach to see the changes.

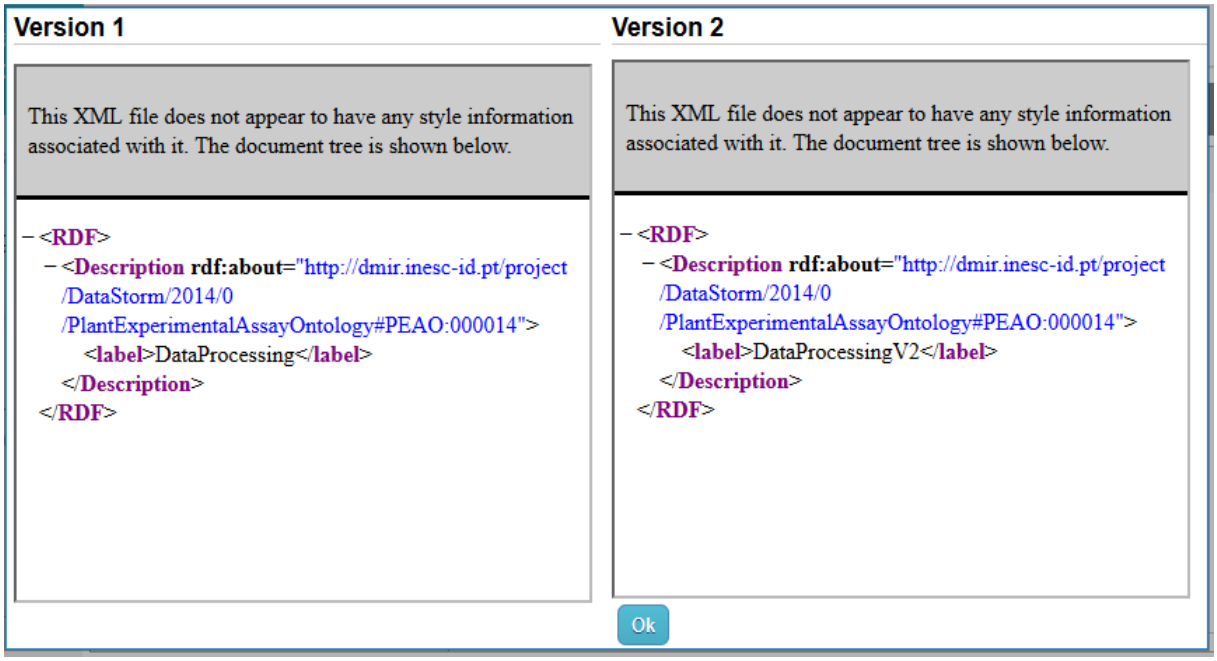


Figure 21. Example of the differences between two versions of the ontology *Plant Experimental Assay Ontology*

4.2. Project Management

With the purpose of enabling the results of each biological experiment to be stored separately in the repository, the concept of Project was considered. This allows the user to create different projects that are associated with different ontologies and can contain different data (**Figure 22**).

[86] This comparison can also be done through the REST Interface.

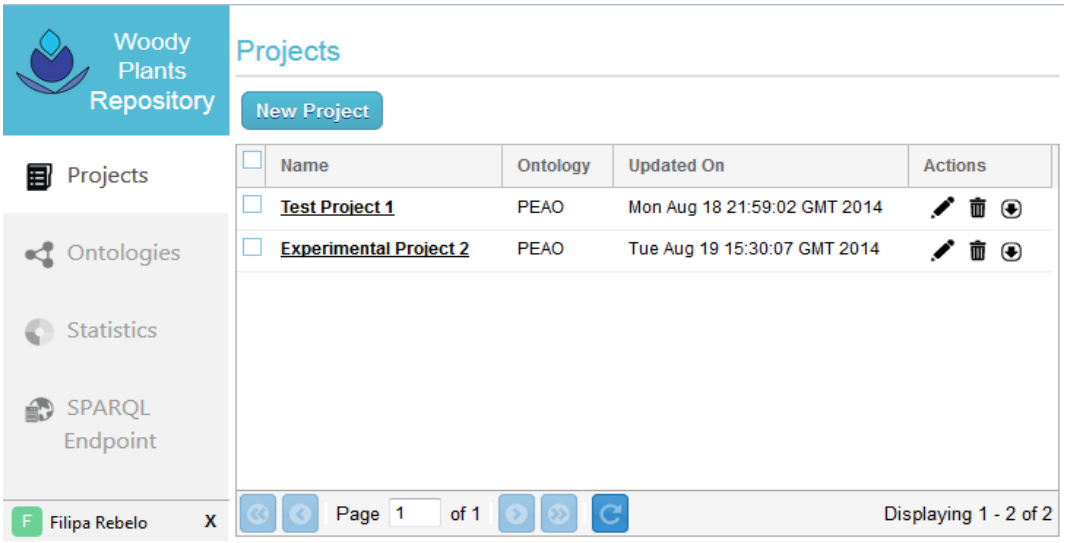


Figure 22. List of all projects in the repository.

When a project is created it must have an ontology associated and, if it doesn't, no data can be inserted in the repository because there is no schema in which the data can be based on. Furthermore, the system has the ability to allow the user to download all the data contained in a project into a Turtle file, which can be imported again at any time into other project (if it uses the same ontology).

4.2.1. Data Visualization

All the data related to a project can be manipulated and viewed through the provided web interface (**Figure 23**).

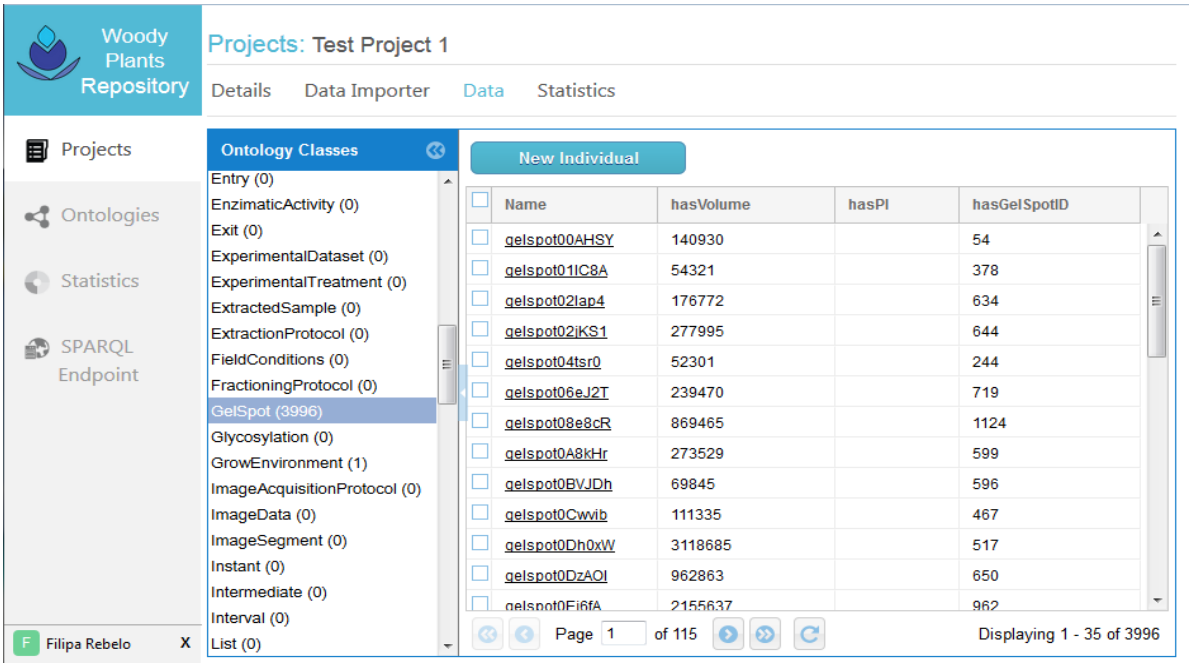


Figure 23. Repository data under the *Test Project 1* Project

In the previous **Figure 23**, it is possible to see in the left side of the panel a list of all the classes existing in the ontology that defines the data structure of the project (with the number of individuals in front of the class name). Once a class is selected, it appears on the right side of the panel a table showing all the information about the individuals of that class (datatype or object properties). Additionally, for a more natural navigation between individuals, the user access to all the details about an individual by clicking in its name. All this information workflow is retrieved dynamically, and therefore, it will work for any ontology added to the tool.

4.2.2. Data Management

The data of a project can be provided through Excel files or inserted manually. The following sections explain how to import data from Excel files and how to insert it manually.

Import Excel Data

To deal with the issue of importing the information gathered directly from the machines of biological experiments, it was introduced the feature “Data Importer”. Hence, when a new set of data is collected, it can be inserted as an Excel (**Figure 24**) and the publish process begins.

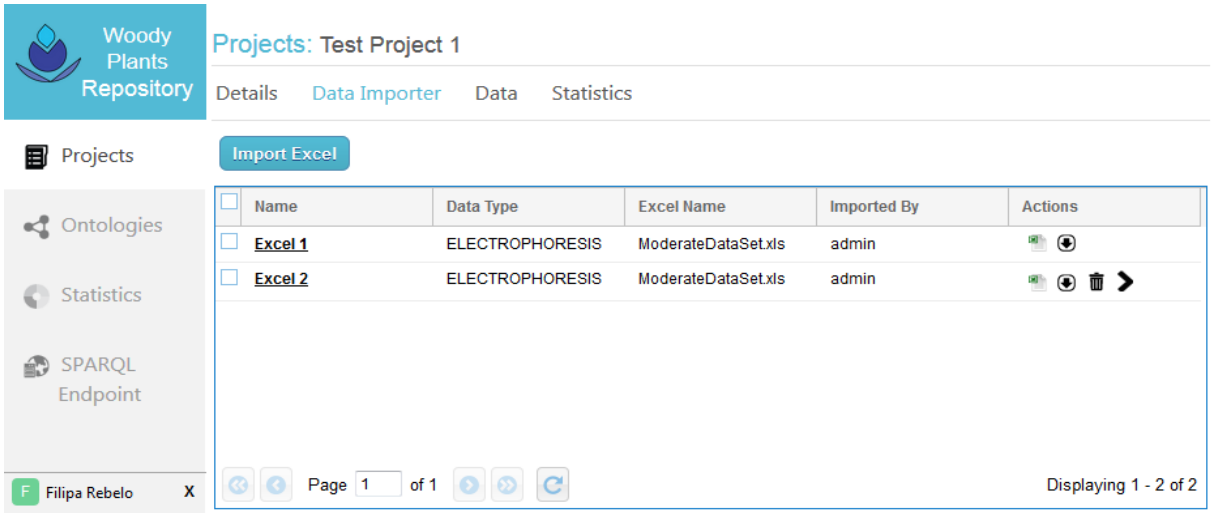


Figure 24. Data Importer view – list of all Excel files created

The publish process is composed by the following steps:

1. Creation of a Turtle file containing the information of the imported Excel file structured according to the active ontology version. This step is automatic and transparent to the user but allows for the manipulation of the data to be imported before being inserted into the repository;
2. The validation of all the required datatype properties in all imported individuals by user;
3. Creation of a Turtle file that represents a copy of the repository – similarly to Step 1, this is also a transparent step, performed because new individuals can be created in the repository to perform linkage, and this way it doesn’t interfere with the actual data in the repository.

4. The validation of all the required object properties and linking between imported individuals and the ones in the repository (including the creation of new individuals in the repository so new links can be created to them) by user;

5. Final publish of all the data into the repository, which consists in the union of the Turtle files of the imported Excel and the repository copy and the actual repository.

Although this process being composed by five steps, for the user is a process with only to steps which are discussed in more detail below.

Step 1 - Validation of the datatype properties

This is the first step of the validation process for the user, where is alerted to fill all the required data properties. **Figure 25** shows on the left side, all the imported classes that contain individuals. These classes are presented in a tree-like widget that, for each class node, displays one *VALID* and one *INVALID* child node. By selecting one of these nodes, the correspondent individuals appear on a table on the right side of the screen. It is possible to see in **Figure 25** that there are three invalid individuals under the *GelSpot* class, two of which are missing the *hasVolume* datatype property and one the *hasGelSpotID* property. The missing properties are shown in red in the table. Additionally, the user can edit the missing information by clicking on the individual's name.

Once the validation of the datatype properties is finished the user can go to the next step.

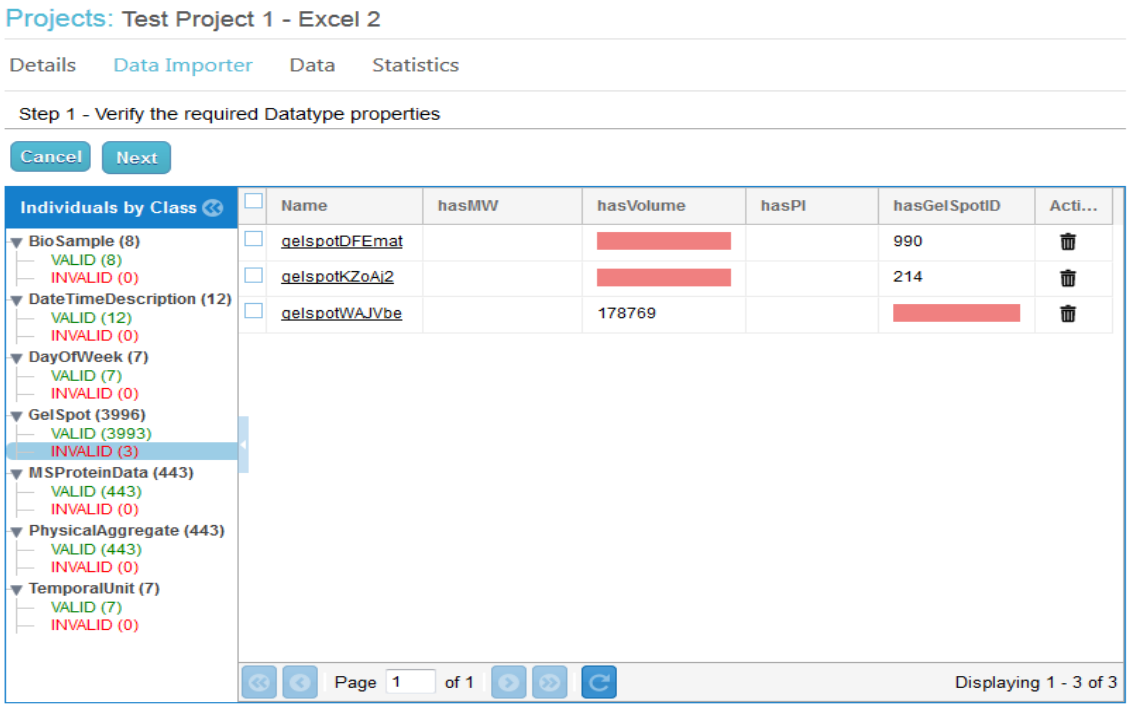


Figure 25. Step 1 of publishing Excel data into the repository – datatype properties validation

Step 2 - Validation of the object properties and individual interlinking

This is the second and last step of the validation process and, in the same way as the previous one, a *VALID/INVALID* tree of the imported data is presented on the left side. In addition, on the right there is a tree containing all the classes in the repository and all their individuals as child nodes (**Figure 26**).

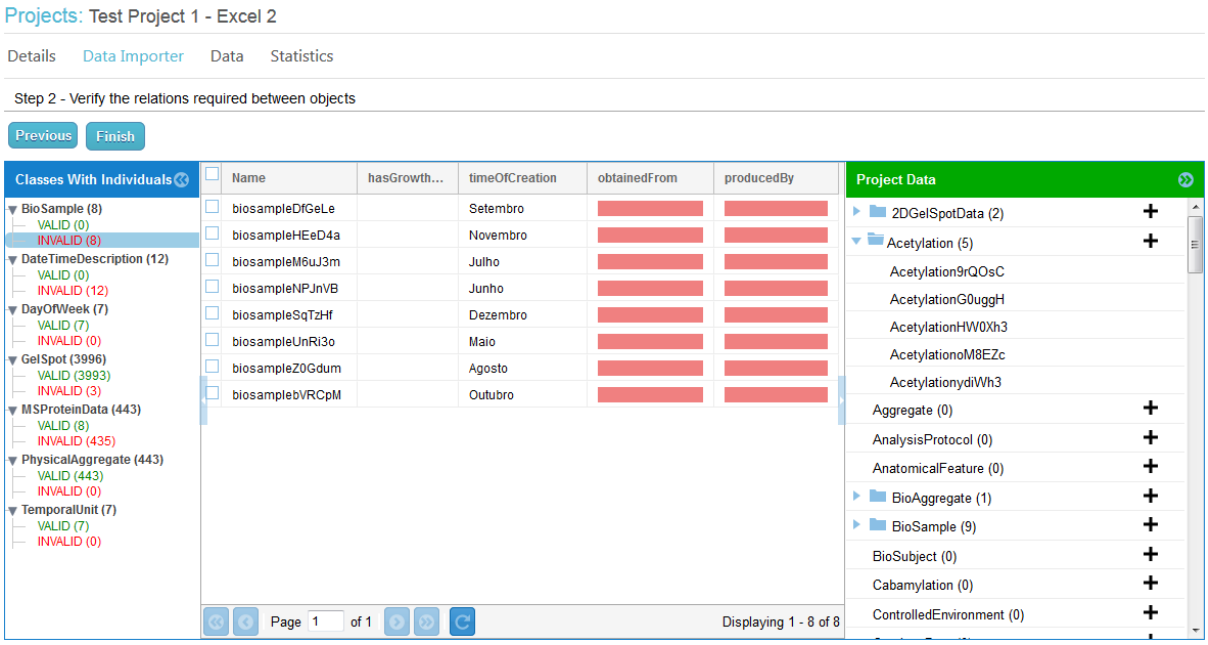


Figure 26. Step 2 of data publishing – validation of object properties and interlinking

As in the previous step, all the missing and required object properties are displayed in red. However, to correct these errors, the user can select a class on either the left or right trees, and their individuals will be displayed in the table in the center of the view. To create the interlinking between the imported individuals and the ones in the repository, a simple drag and drop is required.

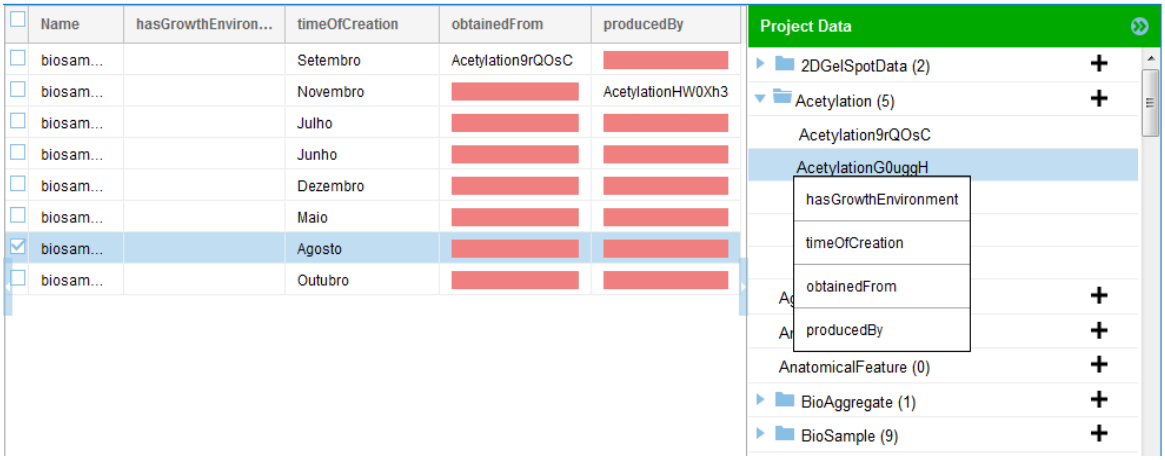


Figure 27. Interlink imported individuals with the ones in the repository by drag and drop

As shown in **Figure 27**, any number of individuals can be dragged from the table into another individual in the repository. Once the individual(s) are dropped a popup appears and enables the user

to choose which type of object property is going to be used. Finally, when the property is chosen the relation is created between the objects.

Available Excel file types

Currently and accordingly to the data provided to use as test-case, the system supports two types of data to be imported:

- **Electrophoresis** - A method for separation and analysis of macromolecules (DNA, RNA and proteins) and their fragments, based on their size and charge. It is used to separate a mixed population of DNA and RNA fragments by length, to estimate the size of DNA and RNA fragments (an example of the data can be seen on **Appendix A)**.
- **Mass Spectrometry** - An analytical chemistry technique that measures the mass-to-charge ratio and abundance of gas-phase ions.

Therefore, three separate scripts were developed to extract the information from the Excel files and structure it according to the ontology associated with the project where the import is taking place. Thus, when importing an Excel the user must choose one of the three unique formats so that the system knows what script is going to be used. The available formats are:

- **Electrophoresis** – An Excel containing data about Electrophoresis;
- **Electrophoresis With Images** – An Excel containing data about Electrophoresis and with the spots coordinates information. In this case, the user may upload later the corresponding image and see the spots identified on it. In **Figure 28** is an example where it is possible to see an individual belonging to *2DGelSpotData* class, containing the datatype property *hasImage* whose value is a small thumbnail of the gel image.

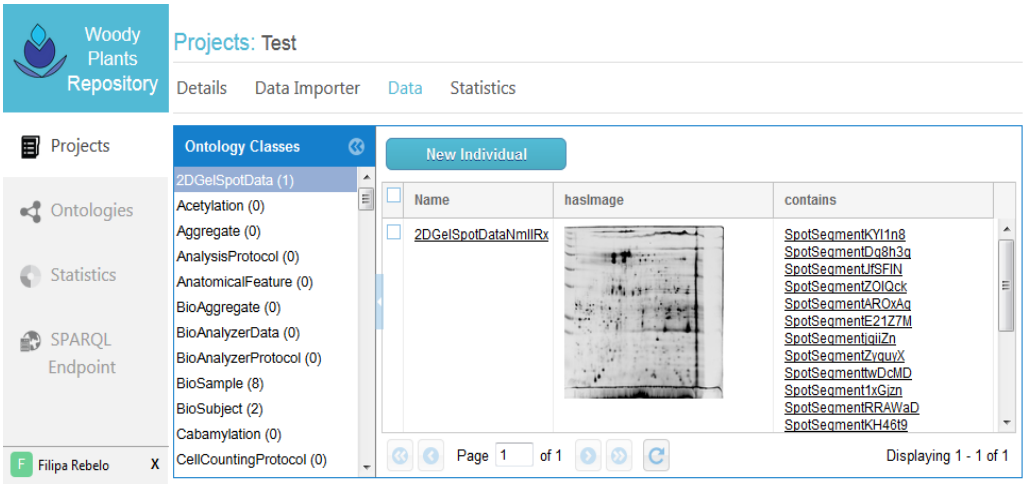


Figure 28. Individual with an image object property

To this specific case, and because all the contained *SpotSegment* individuals have X and Y coordinates which represent their location on the image, once the image is clicked a popup appears showing the image with the spots marked on it (**Figure 29**).

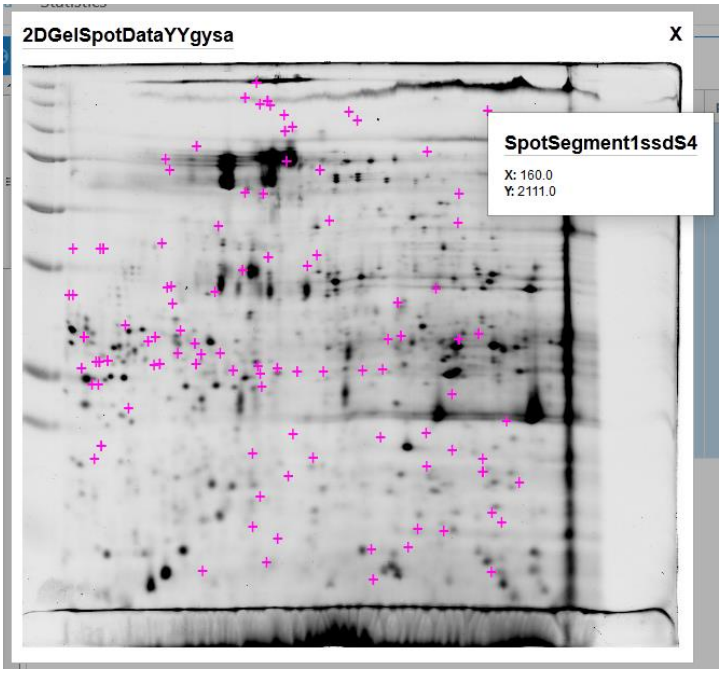


Figure 29. Gel image with all the spots coordinates shown in overlay

- **Mass Spectrometry** – An Excel containing data about Mass Spectrometry.

Finally, after a name assigned, the file type chosen and the file uploaded, the imported Excel is submitted to the publish process described in the previous sections.

Create/Edit Individuals

The user can create, edit or delete any individual. The **Figure 30** shows the creation/edit form for each individual. The form contains a small description of the class which is read from the ontology file itself, followed by the datatype properties represented through textboxes. Next, a table is used to define the object properties of the individual. All of these properties are verified by the class's restrictions, which allows to warning the user for required fields and other constraints.

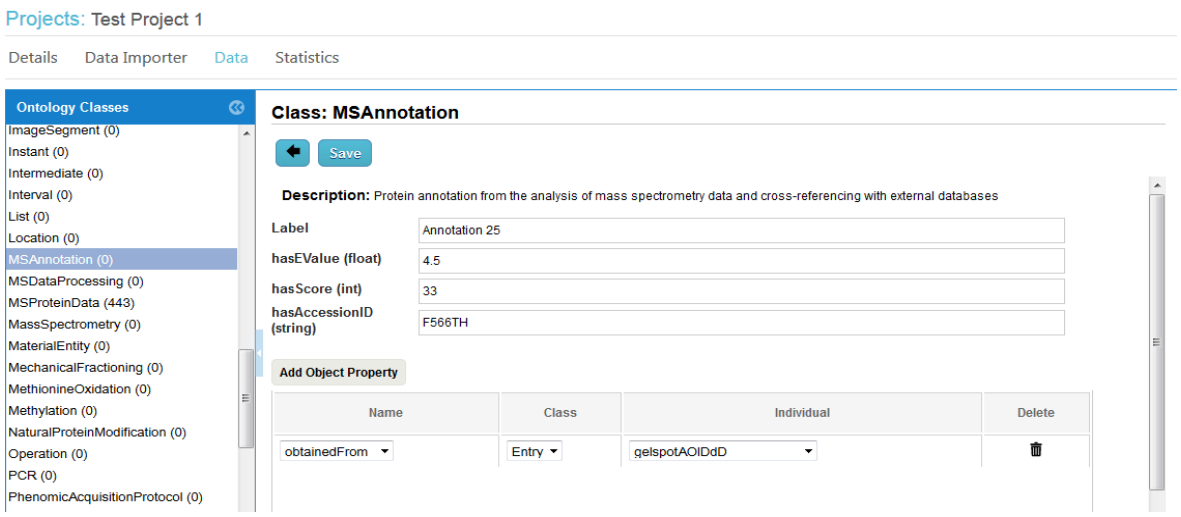


Figure 30. Form to add a new individual under the *MSAnnotation* class

SPARQL for TDB accesses

Although Jena provides a Java API that could be used to handle individuals operations, due to performance issues, it was used SPARQL to retrieve individuals and also to perform the insert and delete operations from/to Jena's TDB. Thus, all the create/edit operations are converted into SPARQL queries that are then sent to the TDB. Moreover, in SPARQL the update operation doesn't exist, so the standard solution is to delete the individual and add it again. To delete an individual, it is used SPARQL's DELETE operator in all the triples in the repository where the individual is present as subject, predicate or object. In **Figure 31** can be seen an example of the data inserted in the example of the **Figure 30,** in the form of SPARQL query.

```
prefix peao: <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#>
prefix ppo: <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PipelinePatterns#>
prefix obo: <http://purl.obolibrary.org/obo/>
prefix time: <http://www.w3.org/2006/time#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

INSERT DATA {
<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#MSAnnotationROGCCg> rdf:type
<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000036>; rdf:label "Annotation 25";
<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000091> "4.5";
<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000108> "33";
<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000086> "F566TH";
<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PipelinePatterns#P:00007>
<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#gelspotAOlDdD>; };
```

Figure 31. SPARQL query to insert an individual of the class *PEAO:000036*

4.3. Fuseki as a SPARQL Endpoint

Instead of creating one SPARQL Endpoint, and the fact that Jena provides its own (Fuseki), it was decided to use it as the main SPARQL Endpoint of the application. Fuseki is an HTTP interface to RDF data that supports SPARQL for querying and updating and runs as a stand-alone server using the Jetty web server[87]. It was then embedded in the application to create a more seamless interaction (**Figure 32** and **Figure 33**).

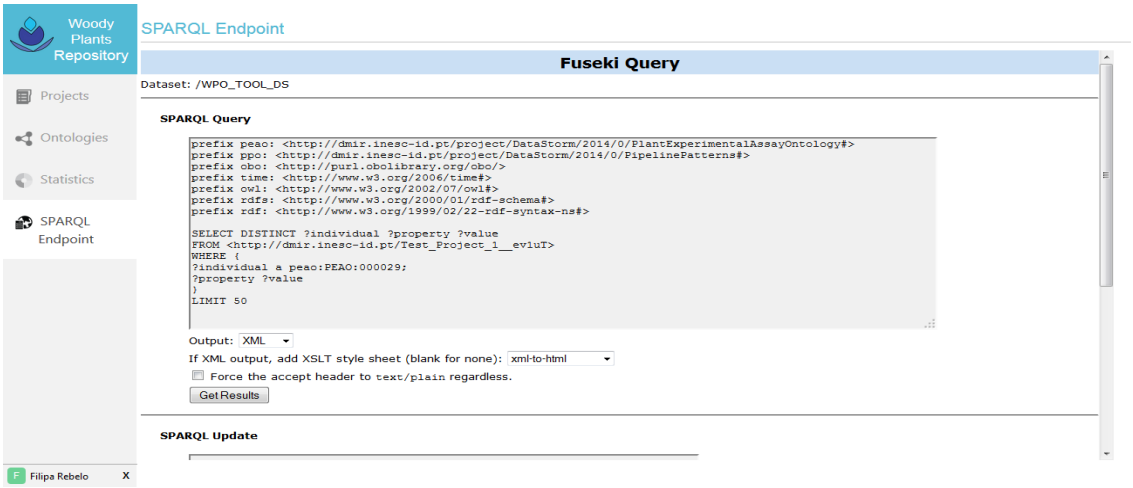


Figure 32. Embedded Fuseki server – The system's SPARQL Endpoint

[87] http://www.eclipse.org/jetty/ accessed 09/10/2014

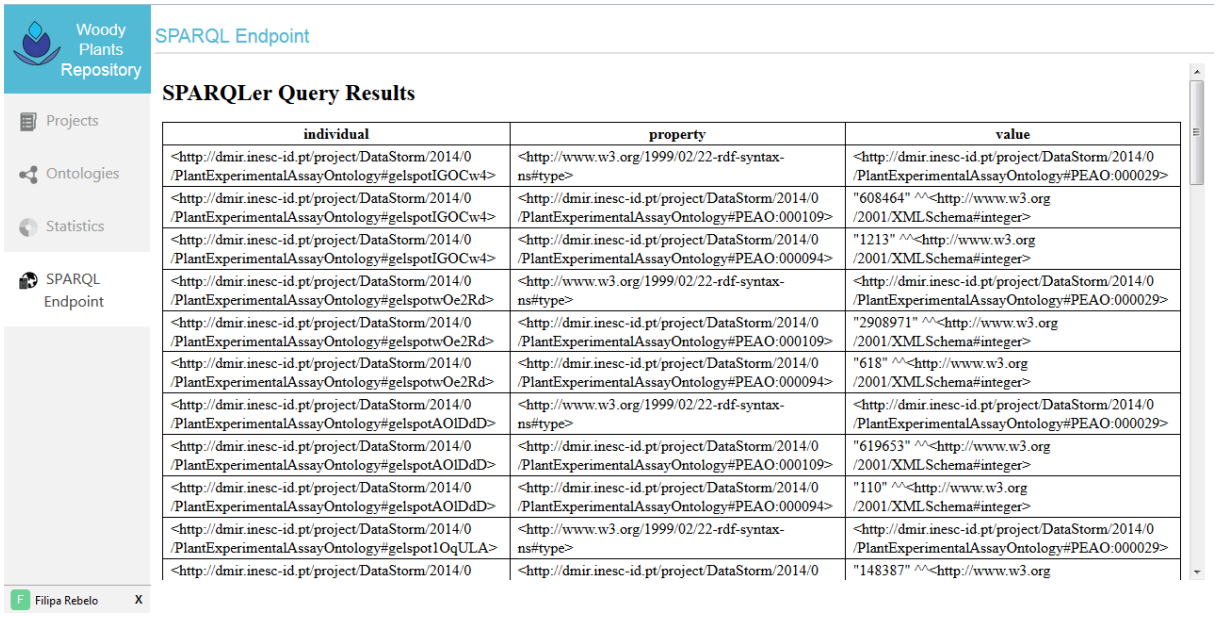


Figure 33. Result of a query to list individuals and their properties

In **Figure 32** and **Figure 33** it can be seen the same SPARQL Query explained in **Section 4.2.1** being introduced into Fuseki, and its result.

4.4. Statistics

To give an overview of the data present in the repository, user can access to some statistical measurements like number of classes, individuals, valid and invalid individuals, among others. These measures can be applied to a specific project or be an overall of all projects in the tool (**Figure 34**). Each of these measures is dependent on the currently active ontologies on the system.

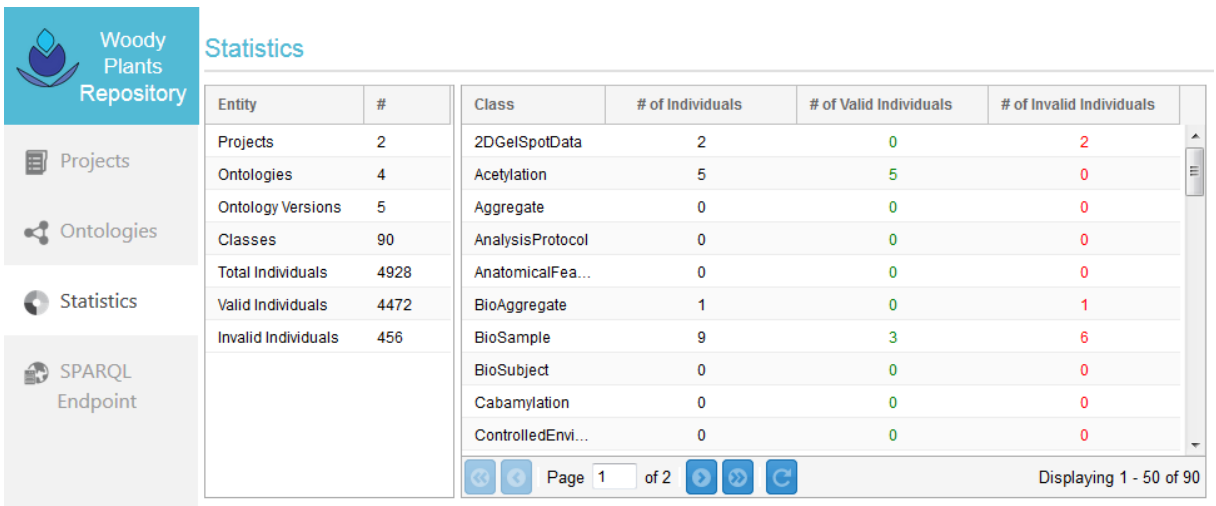


Figure 34. Statistics for all the projects in the system

5. Self-Assessment

Although the repository developed being only the basis for a richer repository, it should have been tested by the end-users to perceive interaction issues and explore other possibilities. However this wasn't possible due to availability issues which lead to a self-assessment of the work developed.

Therefore, the developed solution was tested with all the inputs provided and it was possible to collect valuable information about the implemented features and their limitations.

5.1. Ontology Management

The system as some limitations concerning ontologies format import. It only works with ontologies in OWL or RDF/XML format, but it should also accept other formats like Turtle and N3.

As described in **Section 4.1.2**, the repository enables the comparison of two ontology versions. However, when comparing two versions of the *PlantExperimentalAssayOntology* it was noticed that few changes can generate two very large and different XML, hard to read and perceive the differences (see **Figure 35**).

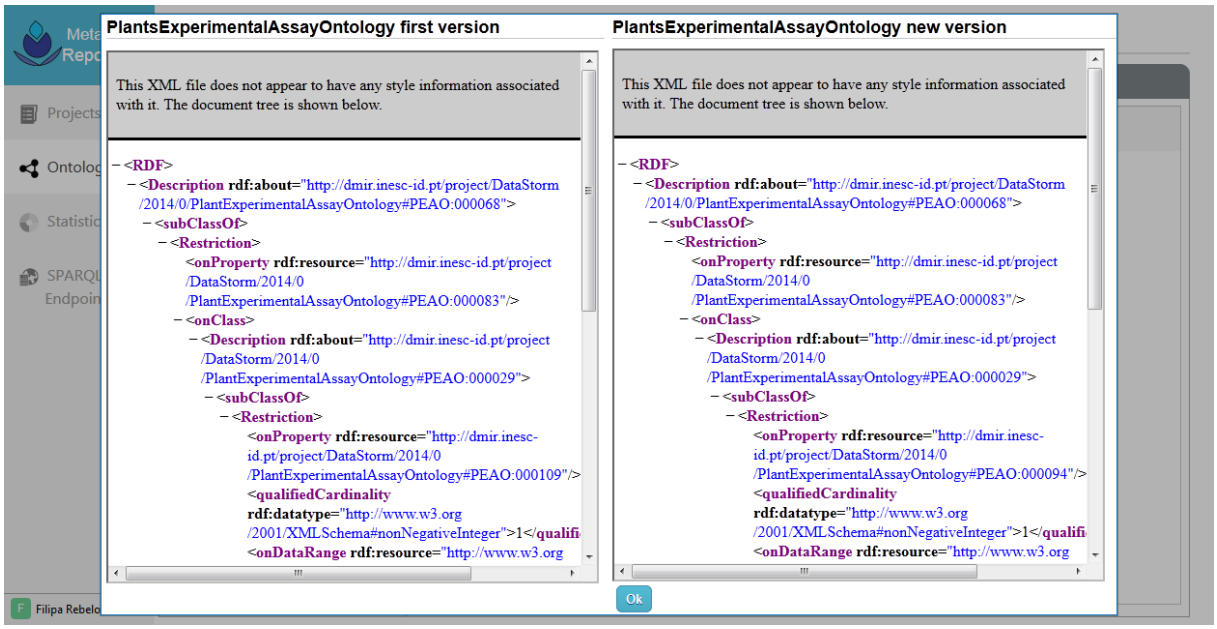


Figure 35. Comparison of two versions of the *PlantExperimentalAssayOntology*

An interpreter of these XML files should be implemented that analyzes, transforms and synthetizes them into comprehensible information showing the end user the differences in a much simpler way.

Another important issue that was not solved in the proposed solution relates to the problem when a new version of the ontology is imported to the system and set active, and there is already a Project in the repository with data using an older version. Currently, that data is visible if the classes of the older ontology version are consistent with the classes of the newer one, otherwise it will not appear to the

user. This is no solution and a process of migration of the data from the old ontology to the new must be considered.

5.2. Project Management

All the data imported from the Excel files is structured according to the ontology and displayed to the user. In **Figure 36**, it is possible to see the project, called *KDBIO Use Case*, holding the imported Excel data with the protein annotations linking to NCBI. The imported data of *KDBIO Use Case* project have resulted in the creation of 381 individuals belonging to *MSAnnotation* class. As the number of individuals can greatly increase, they are listed with pagination allowing for better performance by retrieving them in chunks from the repository, but still creating an overview of the number of individuals each class contains.

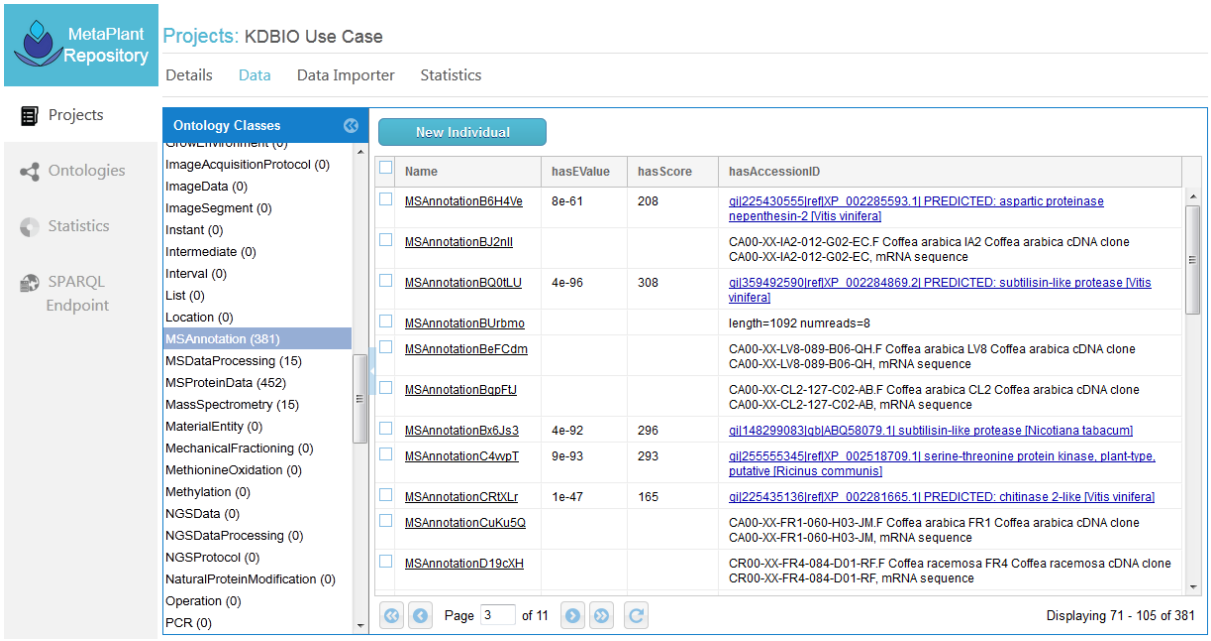


Figure 36. Stored information about coffea leaf rust interactions used in the *KDBIO Use Case*.

The *Plant Experimental Assay Ontology* is composed by two ontologies: PipelinePatterns and PO. During the tests, it was detected that in the cases that PO ontology was uploaded, when it was tried to associate the ontology containing also the PO ontology with a project, the system became very slow to the point it was impossible to interact with it due to the size of this ontology that contains thousands of classes. One possible reason is the usage of Jena's Java API to load the classes from an ontology. SPARQL should be used to overcome this issue and new tests should be done to make sure the system supports any ontology despite its size.

Additionally, the system should provide a way to deal with the images in order to, for example, when a gel image is uploaded, be possible (if needed) to correct the image axes to hit right with the coordinates information spots.

5.3. Repository Data Management

The Excel data import validation process was developed in a way that allows new data to be easily imported and linked with already existing entities in the repository. However, user is the main responsible for the data consistency and if a mistake is done, currently, it is hard to detect.

An example of the interface validating process can be seen in **Figure 37**. It contains a scenario reflecting a repository with already Electrophoresis data and that is currently running the validation process over the imported Mass Spectrometry data. Through a drag and drop process described in **Section 4.2.2**, four individuals were turned valid by associating them with individuals present in the repository (*BioSample7J4SVO* and *PhysicalAggregate0EkUTM* correspondingly). This approach, however, can have some problems when we want to associate large numbers of individuals (larger than the page's size) with the repository individuals. Also, the repository individuals shown for each class are limited to a fixed value. This should be reviewed and so that a more scalable solution can be found.

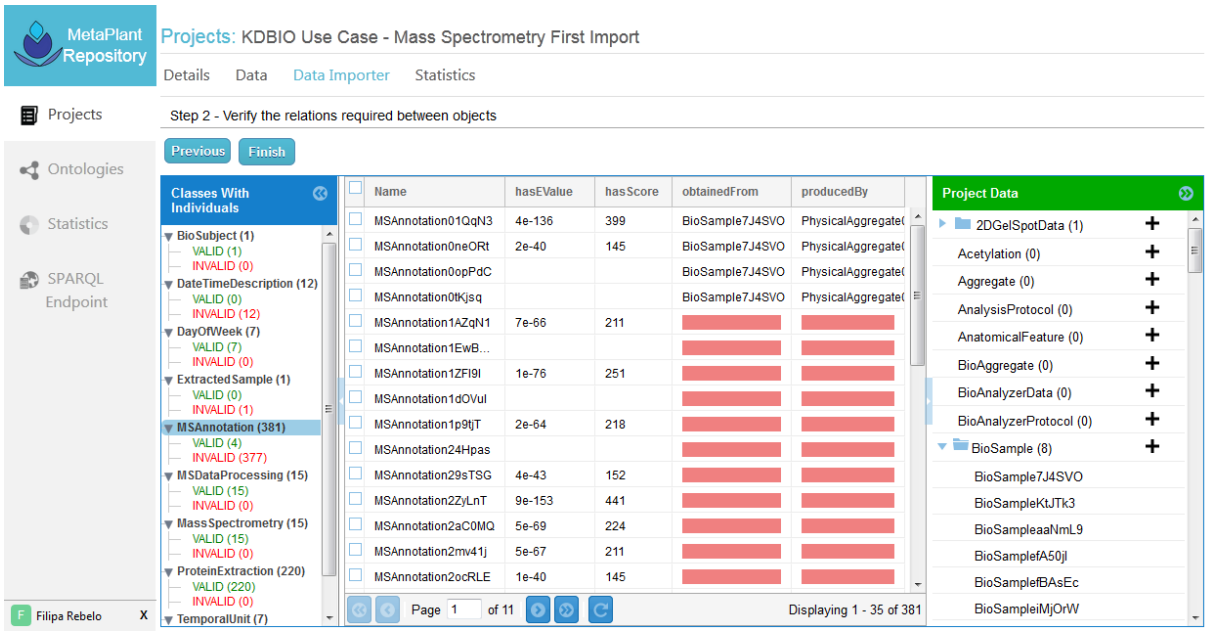


Figure 37. Data import process for Mass Spectrometry data

Additionally, it can be noticed in the same figure that 377 individuals of *MSAnnotation* are invalid because both the *obtainedFrom* and *producedBy* object properties are empty and, as displayed in red by the interface, they should be filled due to restrictions on the ontology so the individual can be valid.

5.3.1. Importing data through Excel

Three separate scripts were developed to extract the information from the Excel files and structure it according to the ontology chosen in the project where the import is taking place. Thus, when importing an Excel the user must choose the type of data contained in it (*Electrophoresis*, *Electrophoresis With Images* or *Mass Spectrometry*). This creates a limitation to the system in the sense that only information expressed in one of these three Excel formats can be imported into the system. Although users can

insert data directly into the repository through the user interface, when large amounts of data need to be inserted it can only be done through an Excel file formatted with one of these three options.

5.3.2. Create/Edit Individuals

To edit each individual the approach used includes input text fields for datatype properties and a table with combo boxes for object properties. Though it works for datatype properties, as there are generally few in this test-case, the solution chosen for object properties doesn't work so well.

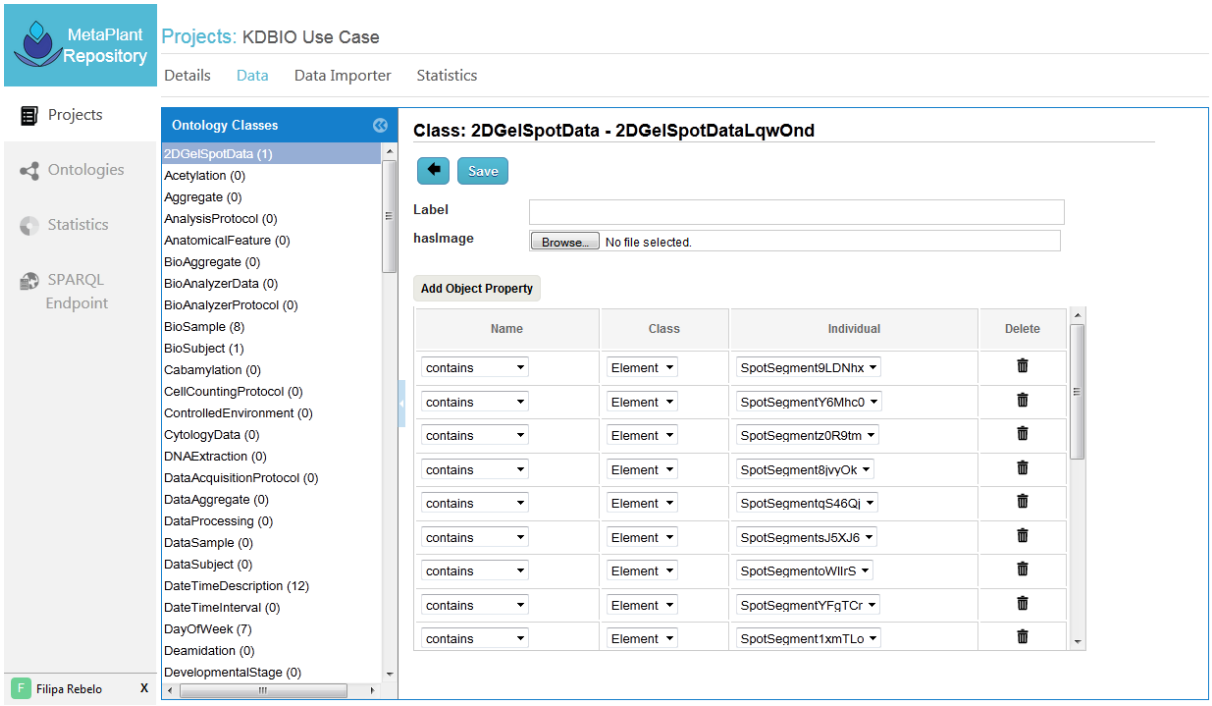


Figure 38. Edit individual *2DGelSpotDataLqwOnd* in the *KDBIO Use Case*.

In **Figure 38**, this individual of the class *2DGelSpotData* contains several hundred relations with other objects of the class *SpotSegment*. In this proposed approach each new relation is created by adding a new entry at the table, selecting the type of object property, the class of the target individual and finally its id. Although this approach can work when creating a small number of relations, when we want to relate one individual with dozens of others new visual solutions should be evaluated not only to create the relations but also to list them.

5.3.3. Repository data import and export

In order to allow users to modify their data in other tools, as well as, create backups, an option was created for exporting all the data contained in the repository. This feature exports all the data contained in a project to a Turtle file as well as it is saved in the repository (a small portion of the Turtle file of the test-case can be seen in **Appendix E**). This feature is complement by the import option that allows users to import a Turtle file into a project. However, the data related with the project like its name or description is not exported. In addition, the import feature is limited because if the ontology is not previously in the system and associated with the project the data won't be shown to the user. A possible

solution would be the extraction of the information detailing the ontology (which is available in the Turtle file) and automatically import the ontology into the system.

5.4. Jena's Fuseki as SPARQL Endpoint

To expose all the data available in the repository, Jena's Fuseki was chosen. It has a user interface and a HTTP and REST based interface to access the data both for human and machine users. Therefore it creates the means to make it available to other sources enabling its linkage to external resources. However, a technical limitation was found: when Fuseki is started (as a standalone application running on a Jetty server) it runs in the same virtual machine as Jena's TDB. Due to the TDBs accessing architecture to the filesystem. TDB locks its usage and whenever the data is updated, Fuseki loses the connection to the data and needs to be restarted.

5.5. Statistical Information

With the statistics information, it is possible to perform not only a quantitative assessment of the data but also a verification of which classes contain valid and invalid individuals. In **Figure 39**, we can see that many classes of the ontology are not being used yet.

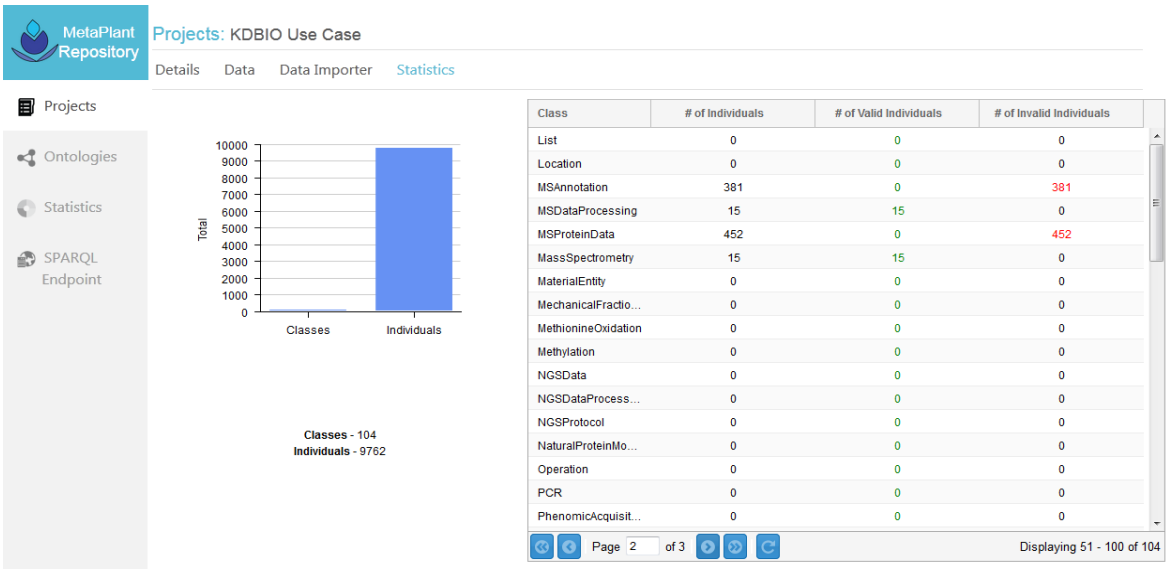


Figure 39. Statistical view for the *KDBIO Use Case*

Also, in this particular case, for a total of **104 ontology classes** only **13** are populated with a sum of **9762** individuals created from the Excel Files about Electrophoresis and Mass Spectrometry data. The distribution of these individuals by class can be seen in the next **Figure 40**.

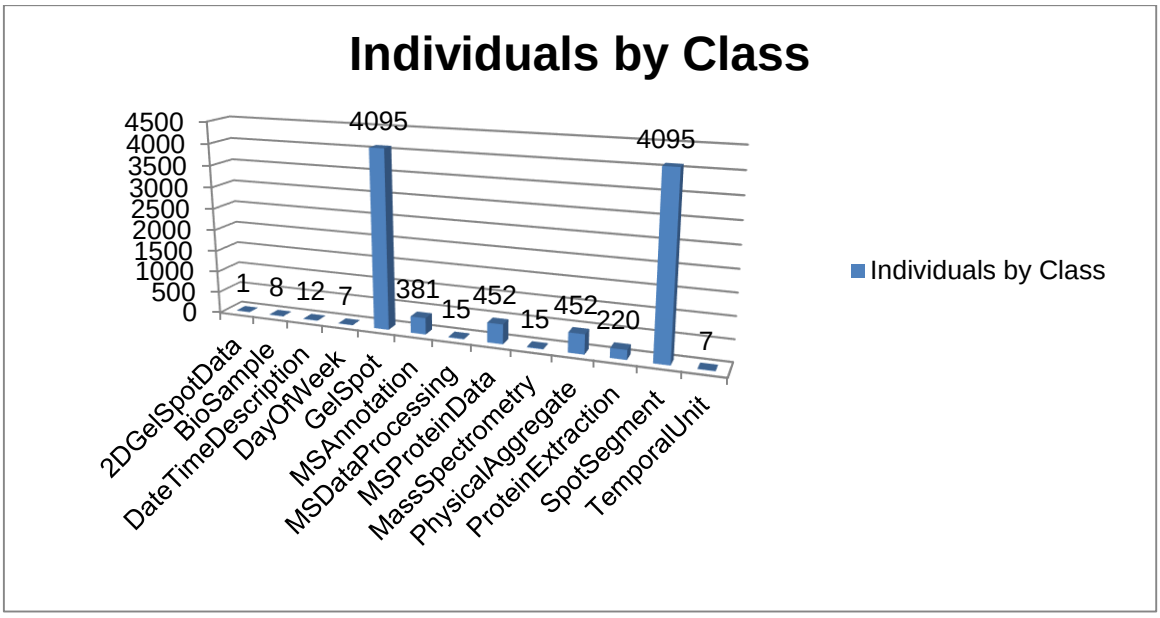


Figure 40. Number of individuals by class for this test-case

Finally, it can be seen the contrast from the total numbers of valid and invalid individuals present in the use case in **Figure 41**.

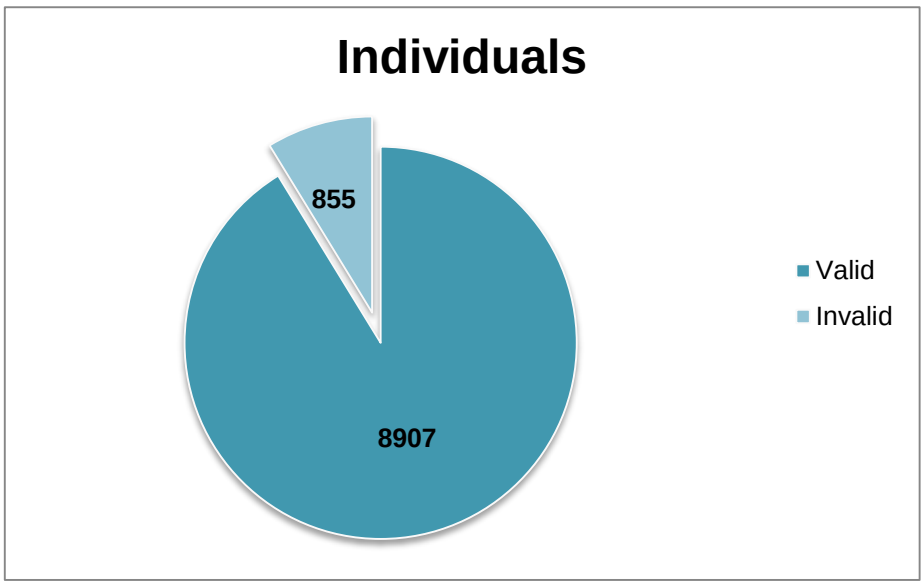


Figure 41. Valid and Invalid individuals for the KDBIO Use Case.

This means that the ontology used in this project contains some restrictions that are not being considered when the import of the data is being done. These restrictions are shown to the user and he may opt to obey them or not.

These are limited statistical indicators that need to be enhanced and new ones should be added to provide a more complete statistical analysis.

5.6. Consolidated Assessment

In summary, the data delivered by the IICT together with ITQB-UNL/IBET created the grounds to evaluate the repository. Next, a list of all goals is presented and, for each of the goals proposed for the envisioned solution, what approaches were used to reach them and what was not done:

Goal	Done	Not Done
Provide data integrity, removing the possibility for redundant information	Creation of a centralized RDF repository supported by ontologies.	Imported data should be matched with the repository's content to look for already existing data.
Convert all imported data to an uniform format	Conversion of the imported files to a turtle file based on the ontology responsible to structure the biological experiments data.	
Expose the data through standard approaches so it can be accessed by external services, either human or machines	The data is saved in RDF format and exposed through a SPARQL Endpoint (*Fuseki*) which allows to search for any existing content through SPARQL, HTTP and a REST-style interface.	Propose a solution for *Fuseki*'s technical issue which can make it to stop working every time the content is updated.
Gather data from multiple sources into a single database, so a single query engine can be used to present data	Data from different experiments can be represented in the system through the concept of a Project. The content of all projects can be accessed with SPARQL.	
Provide a user interface to enable a user without deep ontology knowledge to manage the data	A responsive and dynamic web interface was developed enabling the management of the repositories content.	Research alternative visual solutions to handle the linkage of large amounts of imported data with the one already existing in the repository.
Allow reasoning and data analysis for internal enrichment by adding extra semantics	Use of ontologies as the core schema to structure the biological experiments data, saved in RDF.	Configuration of reasoners to infer new knowledge and adding extra semantics to data.
Log every transaction to enable data recovery and understand the reason behind any problems that might arise	All operations performed by the user are persisted into a XML file.	Although every operation being recorded, new approaches should be explored to, if needed, rollback the action performed.

Table 3. Description for each goal of what was done and what is still missing in the proposed solution

Although many services were implemented and almost all the goals were fully achieved, it can be concluded that the proposed solution is only the foundation for a more complex and rich repository.

6. Conclusion

During this work it was learned that with the continued growth of published scientific data, its integration and computational service discovery became a challenge. This happens due to the unique data models used by several data repositories developed in relative isolation, that use different terminology and formats making it hard for researchers to find all data about an entity of interest, and to assemble it into a useful block of knowledge to give a complete view of biological activity. Even though many databases exist nowadays containing biological information like PlantFiles, WeedUS and PLEXdb (Plant Expression Database) [14], all of them store the data using relational databases which can sometimes duplicate information and are not natively designed to interlink resources. However, new approaches start to emerge. Some examples were addressed, from repositories using Linked Data like NCBI's Gene Expression Omnibus (GEO), the DrugBank repository, Diseasome among others, to the development of ontologies to structure Life Sciences knowledge like the Plant Trait Ontology, Experimental Factor Ontology and BioAssay Ontology (BAO) that were developed using tools like the BioPortal, Web Protégé and Protégé.

Several techniques and technologies were researched to resolve the necessity of managing all the data provided by biological experiments in unified way. This lead to the creation of a RDF repository that supports multiple ontologies to define its data structure, enabling the linkage inside the repository and with external resources. The solution developed promotes the preservation of the semantic relationships between the entities represented therein, making the interpretation of the results and the integration of data produced by different experiments easier.

Jena API was chosen to be the core technology to deal with RDF through its fundamental class, the *Model*, designed to have a rich API, with many methods intended to make it easier to write RDF-base programs and applications. A *Model* can be sourced with data from local files, databases, URLs or a combination of these, and also in triples serialized in formats like RDF/XML, Turtle, among others[88]. Additionally, to deal with ontologies Jena uses an extension of the *Model* class, the *OntModel*, which provides extra capabilities for handling ontologies and offers reasoning services. Finally, it has the ability to store the RDF data in TDB, Jena's native triple store that can be queried using SPARQL. Moreover, based on the previous experience with the GWT[89] framework for developing web interfaces and, in order to provide an easy deployment, cross-browser support and a more attractive *look and feel*, this tool was used.

Finally, the proof of concept was made using the data about the coffee leaf rust interactions experiment provided by the IICT together with ITQB-UNL/IBET and using the *Plant Experimental Assays Ontology* (which describes the pipeline of manipulations performed from specimens to data) developed by KDBIO's Group to structure the data.

[88] http://en.wikipedia.org/wiki/RDF/XML accessed 09/05/2014
[89] http://www.gwtproject.org/ accessed 01/10/2014

6.1. Results Achieved

In general, the solution developed allows the import, querying and manipulation of the data retrieved from biological experiments, in particular:

- Enables the import of data gathered from biological experiments and expressed into Excel files into the repository;
- Manages the import of ontologies and their versions;
- Can contain different ontologies associated with distinct projects allowing for totally different experiments to be carried out in the same system;
- Provides an intuitive interface to manipulate all the entities within the repository;
- Offers a statistical analysis of a project or a global view of all the projects in the system through a small set of indicators;
- Log all the operations done, including user that performs them and at what time as safety and error-recovery measures;
- Exposes the data through a standalone application named Fuseki that is embedded in the system. It offers a SPARQL Endpoint accessible over HTTP and REST-style interaction so it can be accessed by external entities either human or machines.

Through the evaluation with real data, it was concluded that almost all goals were fully completed but several issues were discovered that lead to the conclusion that the developed work is only the foundation to an improved repository where data can be richer and contain greater linking, which can be achieved, for example, with the implementation of reasoners.

6.2. Future Work

The developed work is only the foundation to an enhanced repository where data can be richer and with greater linking. Linked Data should be contemplated to create additional linkage of data with external resources to enhance its reusability and interlinking. Although the repository uses RDF and URIs as identifiers, new techniques that are able to search and link the data available in the repository with other repositories therefore increasing linkage should be explored.

Reasoners can also be configured to run on the repository to detect modeling errors, which typically manifest themselves as unsatisfied concepts and unintended relationships. Also, they enable internal enrichment by adding extra semantics.

Finally, further testing with end-users and larger amounts of data should be performed to determine if the proposed solutions is indeed ready for real case scenarios with greater quantities of information.

References

[1] Fabian Abel, Juri Luca De Coi, Nicola Henze, Arne Wolf Koesling, Daniel Krause, and Daniel Olmedilla. Enabling advanced and context-dependent access control in rdf stores. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 1–14, Berlin, Heidelberg, 2007. Springer-Verlag.

[2] Karl Aberer, Philippe Cudré-Mauroux, Aris M. Ouksel, Tiziana Catarci, Mohand-Said Hacid, Arantza Illarramendi, Vipul Kashyap, Massimo Mecella, Eduardo Mena, Erich J. Neuhold, Olga De Troyer, Thomas Risse, Monica Scannapieco, Fèlix Saltor, Luca De Santis, Stefano Spaccapietra, Steffen Staab, and Rudi Studer. Emergent semantics principles and issues. In Yoon-Joon Lee, Jianzhong Li, Kyu-Young Whang, and Doheon Lee, editors, *Proceedings of the 9th International Conference on Database Systems for Advanced Applications (DASFAA'04)*, volume 2973 of *Lecture Notes in Computer Science*, pages 25–38. Springer, 2004.

[3] Erick Antezana, Ward Blondé, Mikel Egaña, Alistair Rutherford, Robert Stevens, Bernard De Baets, Vladimir Mironov, and Martin Kuiper. Biogateway: a semantic systems biology tool for the life sciences. *BMC Bioinformatics*, 10(S-10):11, 2009.

[4] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *In 6th Int'l Semantic Web Conference, Busan, Korea*, pages 11–15. Springer, 2007.

[5] Christian Becker and Christian Bizer. Dbpedia mobile: A location-enabled linked data browser. In Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee, editors, *LDOW*, volume 369 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[6] François Belleau, Marc-Alexandre Nolin, Nicole Tourigny, Philippe Rigault, and Jean Morissette. Bio2rdf: Towards a mashup to build bioinformatics knowledge systems. *J. of Biomedical Informatics*, 41(5):706–716, October 2008.

[7] Tim Berners-Lee. Www: Past, present, and future. *Computer*, 29(10):69–77, October 1996.

[8] Christian Bizer. The emerging web of linked data. *IEEE Intelligent Systems*, 24(5):87–92, September 2009.

[9] Christian Bizer and Richard Cyganiak. Quality-driven information filtering using the wiqa policy framework. *J. Web Sem.*, 7(1):1–10, 2009.

[10] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, Mar 2009.

[11] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked data on the web (ldow2008). In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 1265–1266, New York, NY, USA, 2008. ACM.

[12] Monica Campillos, Michael Kuhn, Anne-Claude Gavin, Lars Juhl Jensen, and Peer Bork. Drug target identification using side-effect similarity. *Science*, 321(5886):263–266, 2008.

[13] Luca Costabello, Serena Villata, Oscar Rodriguez Rocha, and Fabien Gandon. Access Control for HTTP Operations on Linked Data. In *ESWC - 10th Extended Semantic Web Conference - 2013*, Montpellier, France, May 2013.

[14] Sudhansu Dash, John Van Hemert, Lu Hong, Roger P. Wise, and Julie A. Dickerson. Plexdb: gene expression resources for plants and plant pathogens. *Nucleic Acids Research*, 40(Database-Issue):1194–1201, 2012.

[15] Sebastian Dietzold and Sören Auer. S.: Access control on rdf triple stores from a semantic wiki perspective. In *In: Scripting for the Semantic Web Workshop at 3rd European Semantic Web Conference (ESWC*, 2006.

[16] Li Ding, Dominic DiFranzo, Alvaro Graves, James Michaelis, Xian Li, Deborah L. McGuinness, and James A. Hendler. Twc data-gov corpus: incrementally generating linked government data from data.gov. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW*, pages 1383–1386. ACM, 2010.

[17] Li Ding, Timothy Lebo, John S. Erickson, Dominic DiFranzo, Gregory Todd Williams, Xian Li, James Michaelis, Alvaro Graves, Jinguang Zheng, Zhenning Shangguan, Johanna Flores, Deborah L. McGuinness, and James A. Hendler. Twc logd: A portal for linked open government data ecosystems. *J. Web Sem.*, 9(3):325–333, 2011.

[18] Kai Eckert. Provenance and annotations for linked data, 2013.

[19] Dieter Fensel, Ian Horrocks, Frank Van Harmelen, Deborah McGuinness, and Peter F. Patel-Schneider. Oil: Ontology infrastructure to enable the semantic web. *IEEE Intelligent Systems*, 16:200–1, 2001.

[20] Tim Finin, Anupam Joshi, Lalana Kagal, Jianwei Niu, Ravi Sandhu, William H Winsborough, and Bhavani Thuraisingham. ROWLBAC - Representing Role Based Access Control in OWL. In *Proceedings of the 13th Symposium on Access control Models and Technologies*, Estes Park, Colorado, USA, June 2008. ACM Press.

[21] Giorgos Flouris, Irini Fundulaki, Maria Michou, and Grigoris Antoniou. Controlling access to rdf graphs. In *Proceedings of the Third Future Internet Conference on Future Internet*, FIS'10, pages 107–117, Berlin, Heidelberg, 2010. Springer-Verlag.

[22] Fausto Giunchiglia, Rui Zhang, and Bruno Crispo. Ontology driven community access control. In *In SPOT2009 - Trust and Privacy on the Social and Semantic Web*.

[23] Hugh Glaser and Ian Millard. Rkb explorer: Application and infrastructure . In Jennifer Golbeck and Peter Mika, editors, *Semantic Web Challenge*, volume 295 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[24] Lisa Goddard and Gillian Byrne. Linked data tools: Semantic web for the masses. *First Monday*, November 2011. Available online at http://firstmonday.org/ojs/index.php/fm/article/view/3120/2633#p6, accessed 04/01/2014.

[25] K.I. Goh, M.E. Cusick, D. Valle, B. Childs, M. Vidal, and A.L. Barabási. Human diseasome: A complex network approach of human diseases. In Luciano Pietronero, Vittorio Loreto, and Stefano Zapperi, editors, *Abstract Book of the XXIII IUPAP International Conference on Statistical Physics*. Genova, Italy, 9-13 July 2007.

[26] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993.

[27] L. Guerra-Guimarães, A Vieira, I Chaves, V Queiroz, C Pinheiro, J Renaut, and C Ricardo. Effect of greenhouse conditions on the leaf apoplastic proteome of coffea arabica plants. 2014.

[28] Bernhard Haslhofer and Antoine Isaac. data.europeana.eu - the europeana linked open data pilot. In *DC-2011, The Hague*, August 2011.

[29] O. Hassanzadeh, L. Lim, A. Kementsietsidis, and M. Wang. A Declarative Framework for Semantic Link Discovery over Relational Data. In *Proceedings of the 18th International World Wide Web Conference (WWW2009)*, page 231, April 2009.

[30] Oktie Hassanzadeh, Anastasios Kementsietsidis, Lipyeow Lim, Renée J. Miller, and Min Wang. Linkedct: A linked data space for clinical trials. *CoRR*, abs/0908.0567, 2009.

[31] Michael Hausenblas. Exploiting linked data to build web applications. *IEEE Internet Computing*, 13(4):68–73, 2009.

[32] Jonathan Hayes. A graph model for rdf, 2004.

[33] T. Heath and E. Motta. Revyu: Linking reviews and ratings into the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):266–273, November 2008.

[34] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition, 2011.

[35] Presbrey-J. Berners-Lee T. Hollenbach, J. Using rdf metadata to enable access control on the social semantic web. In *Proceedings of the Workshop on Collaborative Construction, Management and Linking of Structured Knowledge (2009)*.

[36] Matthew Horridge, Jonathan Mortensen, Tania Tudorache, Jennifer Vendetti, Csongor Nyulas, Mark A. Musen, and Natalya Fridman Noy. Introducing webprotégé 2 as a collaborative platform for editing biomedical ontologies. In Michel Dumontier, Robert Hoehndorf, and Christopher J. O. Baker, editors, *ICBO*, volume 1060 of *CEUR Workshop Proceedings*, pages 138–139. CEUR-WS.org, 2013.

[37] Robert Isele, Anja Jentzsch, Chris Bizer, and Julius Volz. Silk - A Link Discovery Framework for the Web of Data, January 2011.

[38] J. KATTGE, S. DÍAZ, S. LAVOREL, I. C. PRENTICE, P. LEADLEY, G. BÖNISCH, E. GARNIER, and WESTOBY. Try – a global database of plant traits. *Global Change Biology*, 17(9):2905–2935, 2011.

[39] Craig Knox, Vivian Law, Timothy Jewison, Philip Liu, Son Ly, Alex Frolkis, Allison Pon, Kelly Banco, Christine Mak, Vanessa Neveu, Yannick Djoumbou, Roman Eisner, Anchi Guo, and David S. Wishart. Drugbank 3.0: a comprehensive resource for 'omics' research on drugs. *Nucleic Acids Research*, 39(Database-Issue):1035–1041, 2011.

[40] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee. Media meets semantic web - how the bbc uses dbpedia and linked data to make connections. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl, editors, *ESWC*, volume 5554 of *Lecture Notes in Computer Science*, pages 723–737. Springer, 2009.

[41] James Malone, Ele Holloway, Tomasz Adamusiak, Misha Kapushesky, Jie Zheng, Nikolay Kolesnikov, Anna Zhukova, Alvis Brazma, and Helen Parkinson. Modeling sample variables with an experimental factor ontology. *Bioinformatics*, 26(8):1112–1118, 2010.

[42] Deborah L. McGuinness and Paulo Pinheiro da Silva. Infrastructure for web explanations. In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, *The Semantic Web — ISWC 2003*, pages 113–129, 2003.

[43] Nuno D. Mendes, Pedro T. Monteiro, Cátia Vaz, and Inês Chaves. Towards a plant experimental assay ontology. In *10th International Conference on Data Integration in the Life Sciences*, 2014.

[44] Paul Miller, Rob Styles, and Tom Heath. Open data commons, a license for open data. April 2008. Copyright is held by the author/owner(s). LDOW2008, April 22, 2008, Beijing, China.

[45] Koro Nishikata and Tetsuro Toyoda. Biolod.org: Ontology-based integration of biological linked open data. In *Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences*, SWAT4LS '11, pages 92–93, New York, NY, USA, 2012. ACM.

[46] Natalya F. Noy, Michael Sintek, Stefan Decker, Monica Crubezy, Ray W. Fergerson, and Mark A. Musen. Creating semantic web contents with protege-2000. In *Protégé-2000. IEEE Intelligent Systems (2001*, pages 60–71, 2001.

[47] Aris M. Ouksel and Channah F. Naiman. Coordinating context building in heterogeneous information systems. *J. Intell. Inf. Syst.*, 3(2):151–183, 1994.

[48] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. *ACM Trans. Database Syst.*, 34(3):16:1–16:45, September 2009.

[49] Alan Ruttenberg, Jonathan Rees, Matthias Samwald, and M. Scott Marshall. Life sciences on the semantic web: the neurocommons and beyond. *Briefings in Bioinformatics*, 10(2):193–204, 2009.

[50] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *Computer*, 29(2):38–47, February 1996.

[51] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006.

[52] S. Softic and M. Hausenblas. Towards Opinion Mining Through Tracing Discussions on the Web. In *Social Data on the Web (SDoW 2008) Workshop at the 7th International Semantic Web Conference*, Karlsruhe, Germany, 2008.

[53] Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. Linkedgeodata: A core for a web of spatial open data. *Semantic Web*, 3(4):333–354, 2012.

[54] Alessandra Toninelli, Rebecca Montanari, Lalana Kagal, and Ora Lassila. A semantic context-aware access control framework for secure collaborations in pervasive computing environments. In *Proceedings of the 5th International Conference on The Semantic Web*, ISWC'06, pages 473–486, Berlin, Heidelberg, 2006. Springer-Verlag.

[55] Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker. Sig.ma: Live views on the web of data. *J. Web Sem.*, 8(4):355–364, 2010.

[56] X.H. Wang, D.Q. Zhang, T. Gu, and H.K. Pung. Ontology based context modeling and reasoning using owl. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 18– 22, March 2004.

[57] Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee, Joan Feigenbaum, James A. Hendler, and Gerald J. Sussman. Information accountability. *Commun. ACM*, 51(6):82–87, 2008.

Appendix

A. Example of a Gel image

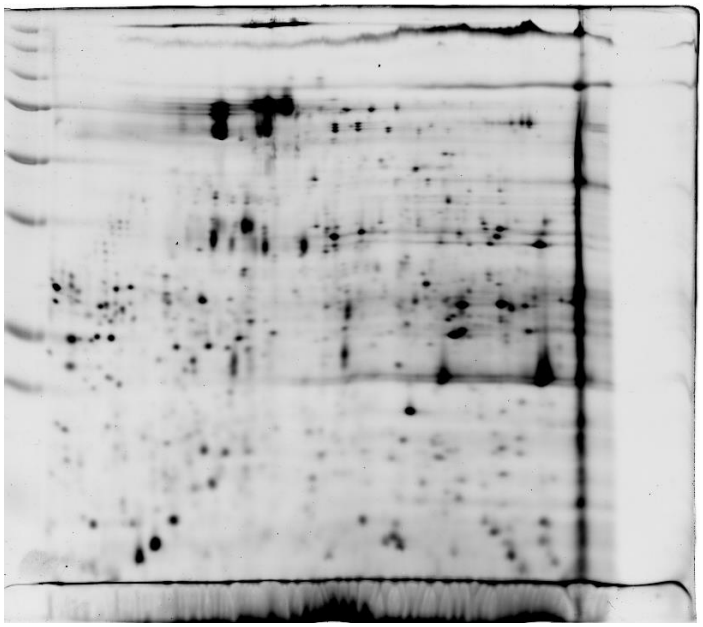


Figure 42. 2DE Gel image

B. Example of the gel analysis Excel

	A	B	C	D	E	F	G	H	I	J	K	L
1	X	Y	Notes/ Nr seq	IF5 Maio	IF9 Junho	IF10 Junho	IF12 Julho	IF16 Agosto	IF17 Setembro	IF20 Outubro	IF23 Novembro	IF24 Dezembro
2	826	112	1	1E+07	1E+07	1,4E+07	1,7E+07	15846873	17960731	10230818	26218348	21211982
3	823	136	2	8E+06	8E+06	1E+07	1,2E+07	12428687	14558143	7118750	16838550	14956601
4	824	243	4	1E+07	1E+07	1,1E+07	1,8E+07	14686720	14721916	9946300	20777326	20433928
5	1139	134	7	2E+07	1E+07	1,7E+07	2,1E+07	18385310	12439468	14745501	23751657	22442323
6	1402	153	8	961743	2E+06	2529695	6471594	2277637	2044153	3470966	2474869	3077864
7	1547	135	9	2E+06	2E+06	3193077	6860907	3628396	2107493	3289885	3271277	4558504
8	1670	127	10	563313	621610	898433	1324156	349582	774539	896592	982028	952630
9	920	280	11	367604	698424	802532	755902	581007	860018	505133	590207	594147
10	2275	204	13	742381	349174	834005	126960	2009072	455745	280938	291053	496744
11	2311	206	14	1E+06	925590	1666922	232430	3088378	497451	564615	364532	482147
12	1372	208	15	1E+06	843460	799571	1837214	1214673	1753334	1096929	2433356	1818170
13	1373	239	16	1E+06	956503	1210298	2324800	1398660	2168342	1490977	2264704	2208421
14	1482	204	17	973743	1E+06	745554	2675422	1395249	1707733	1384920	2887565	1281849
15	1482	234	18	919132	896317	955950	2962483	1495316	1965890	1692779	2611223	1994546
16	1629	227	20	3E+06	1E+06	1979237	3012320	2817466	1662118	1853233	3977032	3944204
17	1630	241	21	102661	186984	379061	640411	478171	115661	140930	677666	714572
18	1777	248	22	307100	242665	266169	1632255	308507	1317567	497368	965290	659718
19	1837	215	23	206480	138672	135139	167820	609421	386408	121028	259287	605530
20	1918	230	24	99162	92517	128938	376643	132096	241719	157134	180735	193299
21	1905	349	25	683257	638950	798339	736982	253108	1035737	455855	445652	433704
22	1937	351	26	293690	276370	403288	298412	104998	472302	209005	194054	128081
23	1274	474	27	1E+06	1E+06	1370766	1997588	1460555	1214080	703057	1369905	2155414
24	1763	421	28	719360	2E+06	1338751	617016	618085	929627	1632706	239218	221481
25	1920	578	30	2E+06	1E+06	1793321	1365669	709791	2585503	832595	699250	265891
26	2021	529	31	198824	183995	193452	128744	172607	275554	207785	95480	38816
27	2169	528	32	139389	193374	191568	47068	167319	257906	111144	109870	190121
28	26	997	33	3E+06	4E+06	3495190	4683413	4948984	3280531	2313530	6848146	9590329
29	22	1062	34	984059	1E+06	990610	932263	1337396	962863	430847	1222515	2209301
30	195	1251	36	1E+06	439861	182562	487780	182949	909791	834012	410508	1150375

Figure 43. Excel produced by Gel analysis machine

C. Log File Example

The Log Manager stores every activity that occurs in the system, from regular operations like creating an Excel to be imported, publishing the data on the TDB and removal of individuals, to the simple change of view in the interface and login on the system. All activities are dated and are associated with the user that performs them.

```
<entry userID="admin" date="18-08-2014 19-10-35" actionType="ENTER_VIEW" message="User entered the Excel list view."/>
<entry userID="admin" date="18-08-2014 19-10-37" actionType="ENTER_VIEW" message="User entered the Edit Excel view."/>
<entry userID="admin" date="18-08-2014 19-10-46" actionType="CREATE_EXCEL" message="User imported excel with name: Excel 1"/>
<entry userID="admin" date="18-08-2014 19-12-50" actionType="CREATE_EXCEL" message="User imported excel with name: Excel 1"/>
<entry userID="admin" date="18-08-2014 19-13-07" actionType="CREATE_EXCEL" message="User imported excel with name: Excel 1"/>
<entry userID="admin" date="18-08-2014 20-14-40" actionType="LOGIN_SAVED_SESSION" message="User entered the tool using saved login credentials."/>
<entry userID="admin" date="18-08-2014 19-14-40" actionType="ENTER_VIEW" message="User entered the projects list view."/>
<entry userID="admin" date="18-08-2014 19-14-52" actionType="ENTER_VIEW" message="User entered the statistics view."/>
<entry userID="admin" date="18-08-2014 19-14-56" actionType="ENTER_VIEW" message="User entered the Excel list view."/>
<entry userID="admin" date="18-08-2014 19-14-58" actionType="ENTER_VIEW" message="User entered the Edit Excel view."/>
<entry userID="admin" date="18-08-2014 19-15-04" actionType="CREATE_EXCEL" message="User imported excel with name: Excel 1"/>
<entry userID="admin" date="18-08-2014 20-15-37" actionType="LOGIN_SAVED_SESSION" message="User entered the tool using saved login credentials."/>
<entry userID="admin" date="18-08-2014 19-15-37" actionType="ENTER_VIEW" message="User entered the projects list view."/>
<entry userID="admin" date="18-08-2014 19-15-51" actionType="ENTER_VIEW" message="User entered the Excel list view."/>
<entry userID="admin" date="18-08-2014 19-15-52" actionType="ENTER_VIEW" message="User entered the Edit Excel view."/>
<entry userID="admin" date="18-08-2014 19-15-58" actionType="CREATE_EXCEL" message="User imported excel with name: Excel 1"/>
<entry userID="admin" date="18-08-2014 19-16-42" actionType="ENTER_VIEW" message="User entered the Step 1 publish view for excel Excel 1"/>
<entry userID="admin" date="18-08-2014 19-17-04" actionType="ENTER_VIEW" message="User entered the Step 3 publish view for excel Excel 1"/>
<entry userID="admin" date="18-08-2014 19-33-35" actionType="PUBLISH_DATA" message="User published data on TDB for excel: Excel 1"/>
<entry userID="admin" date="18-08-2014 19-42-50" actionType="REMOVE_INDIVIDUAL"
  message="User deleted individual with id: http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#Acetylation9rQOsC "/>
<entry userID="admin" date="18-08-2014 19-42-57" actionType="REMOVE_INDIVIDUAL"
  message="User deleted individual with id: http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#2DGelSpotData06dHvm "/>
<entry userID="admin" date="18-08-2014 19-43-06" actionType="ENTER_VIEW" message="User entered the statistics view."/>
<entry userID="admin" date="18-08-2014 19-52-03" actionType="LOGIN_SAVED_SESSION" message="User entered the tool using saved login credentials."/>
<entry userID="admin" date="18-08-2014 19-52-05" actionType="ENTER_VIEW" message="User entered the projects list view."/>
<entry userID="admin" date="18-08-2014 19-52-27" actionType="LOGIN_SAVED_SESSION" message="User entered the tool using saved login credentials."/>
<entry userID="admin" date="18-08-2014 19-52-27" actionType="ENTER_VIEW" message="User entered the projects list view."/>
<entry userID="admin" date="18-08-2014 19-52-30" actionType="ENTER_VIEW" message="User entered the statistics view."/>
<entry userID="admin" date="18-08-2014 19-52-31" actionType="ENTER_VIEW" message="User entered the statistics view."/>
<entry userID="admin" date="18-08-2014 19-53-08" actionType="ENTER_VIEW" message="User entered the statistics view."/>
<entry userID="admin" date="18-08-2014 21-42-09" actionType="LOGIN_SAVED_SESSION" message="User entered the tool using saved login credentials."/>
```

Figure 44. Example of a log file for 18/08/2014 with all the activities

D. Ontology metadata storage

All metadata about ontologies and their dependent ontologies and versions is stored on a XML file that contains specific data about who created the ontology, when it was imported and created, and also when the versions where activated and by whom, and if they are active now.

```
<ontologyList>
  <ontology id="jAUDV" name="Test Ontology" type="BASE_ONTOLOGY" description="Test Ontology for demo purposes"/>
  <ontology id="u4Y7R" name="PEAO" type="BASE_ONTOLOGY" description="Plants Experimental Assay Ontology">
    <versions>
      <version id="Y59ln" name="Version 2" importedDate="Tue Aug 19 15:30:32 UTC 2014"
        importedBy="admin" isActive="false" fileID="Version_2__Y59ln"/>
      <version id="pGPAg" name="Version 1" importedDate="Sun Aug 17 11:22:08 UTC 2014"
        importedBy="admin" isActive="true" fileID="v1__pGPAg"
        activationDate="Sun Aug 17 11:22:14 UTC 2014" activatedBy="admin"/>
    </versions>
    <ontology id="q1Jlk" name="PIPE" type="SUB_ONTOLOGY">
      <versions>
        <version id="kmKcW" name="v1" importedDate="Sun Aug 17 11:12:01 UTC 2014"
            importedBy="admin" isActive="true" fileID="v1__kmKcW"
            activationDate="Sun Aug 17 11:13:15 UTC 2014" activatedBy="admin"/>
      </versions>
    </ontology>
    <ontology id="m97vK" name="Time" type="SUB_ONTOLOGY">
      <versions>
        <version id="Au0xk" name="v1" importedDate="Sun Aug 17 11:12:16 UTC 2014"
            importedBy="admin" isActive="true" fileID="v1__Au0xk"
            activationDate="Sun Aug 17 11:13:29 UTC 2014" activatedBy="admin"/>
      </versions>
    </ontology>
  </ontology>
</ontologyList>
```

Figure 45. Ontology metadata XML file

E. Example of a project data exported to Turtle

```
<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#SpotSegment5LfvGF>
        a       <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000068> ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000083>
                <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#GelSpotrmzqi6> ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000088>
                1.0 ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000110>
                824 ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000111>
                243 .

<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#GelSpotrmzqi6>
        a       <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000029> ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000094>
                4 ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000109>
                10336789 .

<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#GelSpot6QQ0IS>
        a       <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000029> ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000094>
                57 ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000109>
                1226679 .

<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#GelSpotPiPSQw>
        a       <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000029> ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000094>
                153 ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000109>
                601987 .

<http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#SpotSegmentoeLBIB>
        a       <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000068> ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000083>
                <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#GelSpotYpwcJ1> ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000088>
                1.0 ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000110>
                1249 ;
        <http://dmir.inesc-id.pt/project/DataStorm/2014/0/PlantExperimentalAssayOntology#PEAO:000111>
                166 .
```

Figure 46. Project data exported into a Turtle file