4th International SWAT4LS Workshop. Semantic web applications and tools for life sciences. -Workshop, Tutorials and Hackathon Report-

Natalia Díaz Rodríguez
Åbo Akademi University, Turku, Finland (ndiaz@abo.fi)

London, 6-9.12.2011.

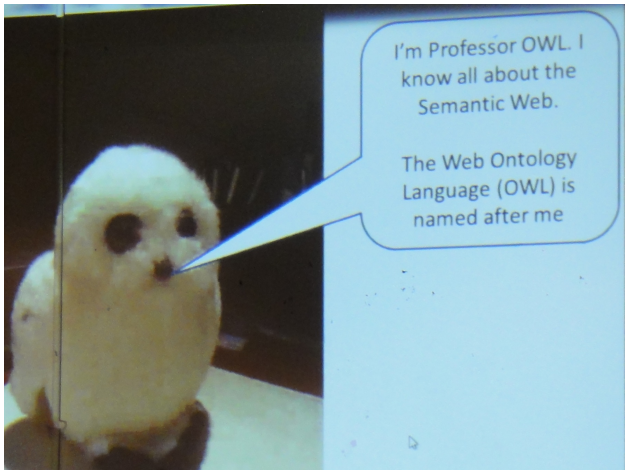

Figure 1: Hackathon lightning talk. The professor OWL.

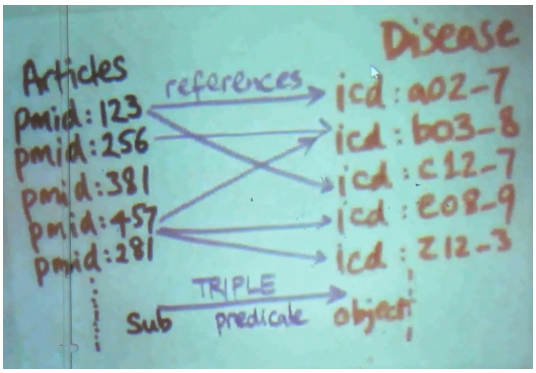

Figure 2: Introduction to RDF Triples (Subject, Predicate, Object) with an example: Articles-references-Disease.

1 SWAT4LS Hackathon (5-6.12.11)

(#devcsi #okfn) For one evening and one whole day, a hackathon was organized by DevCSI (Developer Community Supporting Innovation), OKF (Open Knowledge foundation) and JISC (Joint Information Systems Committee, which promoted also the List8D (open source reading list system)). Ideas regarding "Open Disease Research Reports" were proposed before the hackathon and then, after some lightning talks, people wrote in post-its some more ideas in which people could work on. Everyone expressed also, beforehand, what project or areas was interested in/working on at the moment.

The hackathon page and the working groups are shown below: `http://www.ukoln.ac.uk/events/devcsi/life-sciences-hackdays/index.html`, `http://wiki.okfn.org/Working_Groups/Science/swat4ls_hackathon`. Our group project on Disease Localiser can be found in the latter link and also as appendix at the end of this report.

This kind of hands-on events is really practical, useful and learning oriented. After ideas were exposed, in many cases, other expert colleagues provided further information on the idea, thus making groups to be formed naturally and by means of aggregating different disciplines knowledge. The organizer, Mahendra Mahey is happy to help organizing similar events. Previous hackathons showed, e.g. at Dev8D or Devxs, an AR.Drone using a Microsoft Kinect application (`www.vimeo.com/20521841`) or Kinect open source iphone app for home automation.

Some of the SWAT4LS Hackathon project groups were (among others shown in the hackathon wiki):

- Making an object model for Jena to substitute old JASTOR.
- Visualize drugs from DrugBank and their affected genes.

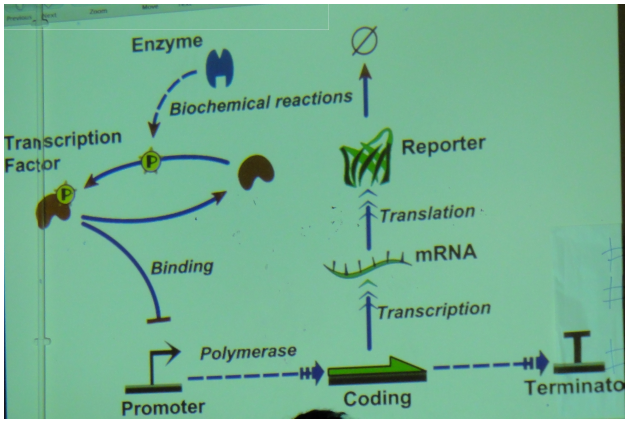

Figure 3: Hackathon example to model with semantic web technologies.

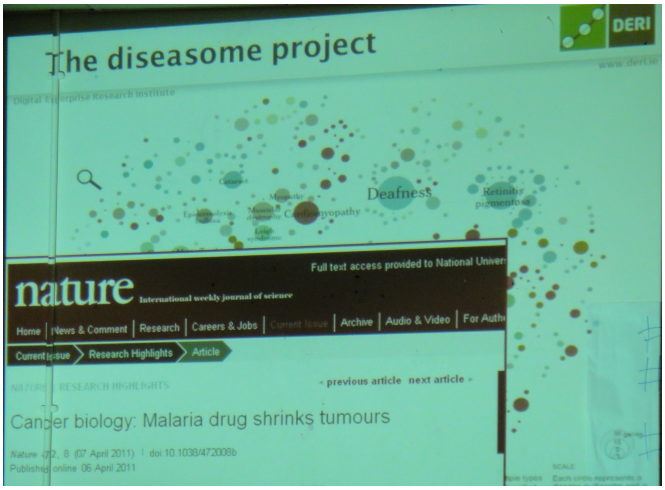

Figure 4: The Diseasome Project.

- Applications with Peregrine (text mining for biomedical texts for recognizing concepts).
- Open Research Report for diseases with `MIDI.org` format. Updating and uploading the xmd (XML Schema) for changes.
- Nanopublications. What kind of minimal information needs to be evaluated in order to reuse the information from articles which reference certain disease?

2 SWAT4LS Workshop Highlights (9.12.11)

1. SNOMED-CT (Systematized Nomenclature of Medicine-Clinical Terms) is a medical health care ontology to support the effective clinical recording of data with the aim of improving patient care. RIO framework identifies syntactic regularities through cluster analysis for grouping similar entities. RIO can be useful as an intermediate step of a more systematic quality assurance procedure for ontologies.
2. Bio2RDF project for solving data integration. Bioqueries (`http://bioqueries.uma.es`) is proposed as a wiki based portal to design, share and execute SPARQL queries. Usability test are available for user evaluation. Queries are searchable and have natural language descriptions. Users can edit others queries, which are tracked in a revision system (maintained by moderators). In the future, federated queries will be included in such way that users can query different biological endpoints simultaneously. The only difficulty for the biologists is not learning SPARQL, which in the

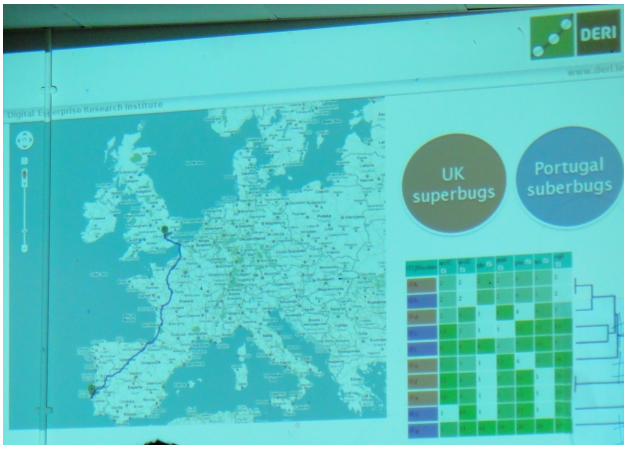

Figure 5: Use case at DERI: Superbugs tracking in UK and Portugal for fast decision making.

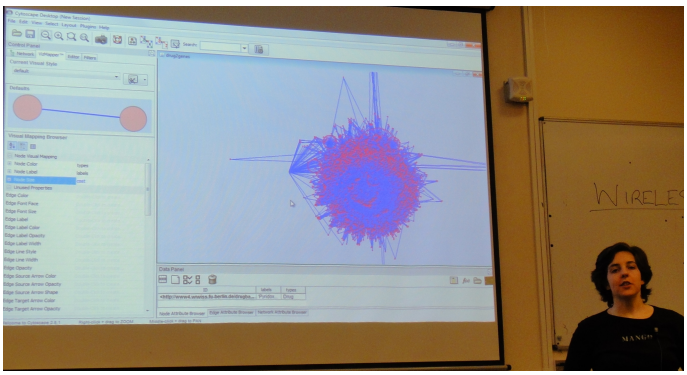

Figure 6: Hackathon group presenting (by Helena Deus) drug-genes interaction visualizations over DrugBank.

end is not that difficult, but rather knowing what is behind the endpoint, i.e., what schema or what kind of information can be obtained.

3. COEUS open source project: a framework for rapid application deployment of new applications in any research field, supported by a comprehensive ontology and RDF-based configuration files. `http://bioinformatics.ua.pt/coeus/`. Tools such as Sencha, BioPerl, Zepto.js, Symphony, Catalyst, Jena modules connectors and PhoneGap are examples of applications that can be programmed on top of COEUS. `http://bioinformatics.ua.pt/dc4`. Diseasecard is a tool for federation of knowledge regarding rare diseases. COEUS team in Portugal searches for collaborators to link their applications and make their datasets available.

4. Utopia documents `http://getutopia.com/documents/` is a tool that extracts graph information from academic papers to make them dynamically linked; the reader does not need to read a paper to obtain certain knowledge facts.

5. Dumontier Lab for Biomedical Knowledge Discovery in Canada aims at understanding how living systems respond to chemical agents by using computational semantic frameworks to make effective use of vast and diverse amounts of biomedical knowledge.

6. ICD-11 (International Classification of Diseases), launched by WHO, has an authoring RDF endpoint in which DBPedia textual definitions were used and MeSH id acted as an anchor to map codes in SNOMED-CT and ICE-10 through UMLS (Unified Medical Language System). This makes the causes of the diseases to be captured as well, and community can drive the granularity of the information. `http://apps.`

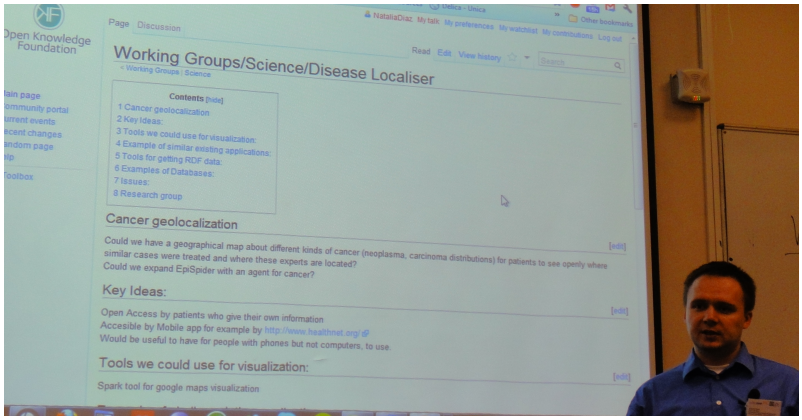

Figure 7: Our Hackathon group presenting (by Tomasz Kluza) the research done for a Disease Localiser.

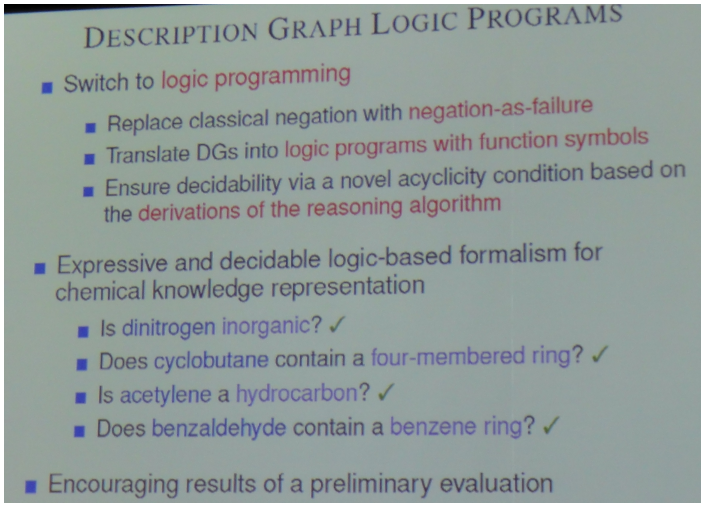

Figure 8: Description graph logic programs for chemical knowledge representation.

who.int/classifications/icd11/browse/f/en, https://sites.google.com/site/icd11revision/.

7. Galaxy is a web framework for combining tools and building workflows in bioinformatics. OPPL (Ontology Pre-processing Language) offers ontology trimming, structure pattern finding and automatic ontology manipulation. Mikel Egaña Aranguren presented OPPL-Galaxy (http://www.slideshare.net/MikelEganaAranguren/opplgalaxy-enhancing-ontology-exploitation-in-galaxy-with-oppl), a Galaxy wrapper for biologist for integration with other Galaxy capabilities. Refactoring is done with OBO2OWL pruning and one can get rid of structures not needed in the ontology. OPPL scripts can be inputted.

8. Spread sheets to OWL with Populous (a tool for populating OWL Ontologies from templates, http://www.e-lico.eu/populous). An application example is the development of a kidney and urinary knowledge base. www.slideshare.net/MikelEganaAranguren/populous-swat4ls-slidesslideshare.

9. Text mining projects in Manchester University have as goal, to link u-Compare with BioCatalogue. They show how UIMA tool allows (with CAS) communication between C and Java programs. U-Compare allows comparison and evaluation of workflows. Workflows are translated into web services that are stored into BioCatalogue register.

10. Selventa presented in the industry session a non-profit project funded by pharmaceutical partners such as Pfizer. BEL (Biological Expression Language) is a variation of

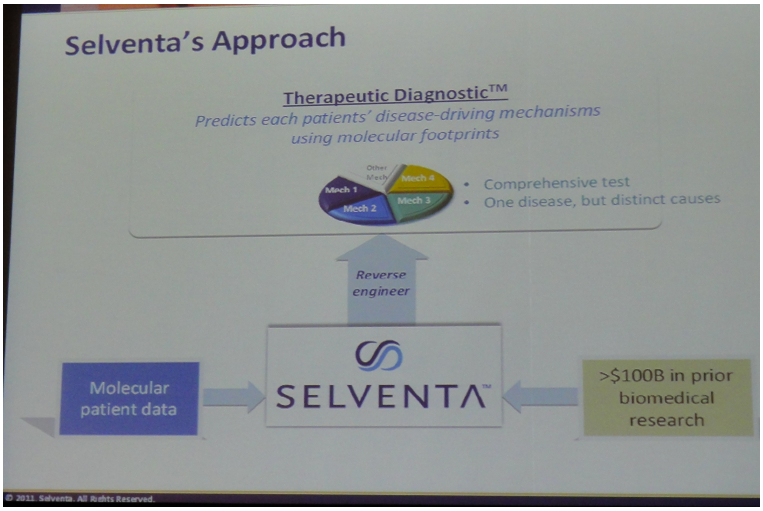

Figure 9: Industry track: Selventa's approach for predicting a disease-driven patient diagnostic.

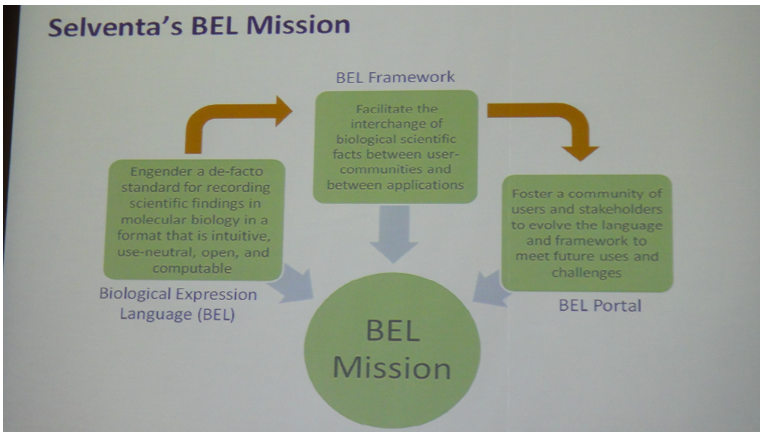

Figure 10: Selventa's BEL (Biological Expression Language)

RDF that uses consequence reasoning and takes provenance into account for showing how diverse information interacts. `www.belframework.com` is a machine and human readable framework to promote the collection, sharing, and interchange of structured knowledge within and among organizations. It uses a KAM store with a Java API and it supports SPARQL and SOAP Restful access.

11. Linked Life Data `http://linkedlifedata.com/` is presented (in the industry session by Ontotext) as a public and free ETL (Extract, Transform and Load) environment to warehouse RDF. The "semantic data integration platform for the biomedical domain" IDE is based on Talend (Open source integration company), with Java interfaces where needed. After a simple workflow is created visually, the code is generated and saved into OWLIM image.

12. Defects in ontologies (syntactic, semantic or modelling defects) are studied at the University of Linköping through the OAEI (Ontology Alignment Evaluation Initiative). Detection of missing is-a relations is done by using external knowledge through heuristics (finding explanations /justifications). When domain experts are not available in the testing period, Wordnet is used to check is-part-of relations in between classes.

13. The BioMoby system for interoperability between biological data hosts and analytical services allows faceted search from rich metadata or free text descriptions. Information extraction is achieved through semantic vectors containing extracted facets. `http://biomoby.open-bio.org/index.php/what-is-moby/`.

Figure 11: Selventa's patient data-driven analytics.

14. The IARC TP53 database, maintained by the International Agency on Cancer in Lyon, includes data on somatic mutations. First, mappings were done using D2RQ and LOD (Linked Open Data). A TDB (Jena component) triple store was made accessible as a SPARQL endpoint using Joseki and a Linked Data interface was created using Pubby (a tool to turn a SPARQL endpoint into a Linked Data server).

15. DOG4DAG, a plugin for Protégé and OBO-Edit (Götz Fabian, Dresden University). Semi automated ontology generation using OBO (The Open Biological and Biomedical Ontologies) and Protégé. The tool can generate term definitions from the web.

16. Frameworks within Tridium company (Jesse Van Dam, Netherlands) are the commercial Niagara framework (`http://www.niagaraax.com/cs/products/niagara_framework`) or the open source Sedona (`http://sedonadev.org/doc/index.html`). Obix initiatives in Fantom language (`http://fantom.org/doc/obix/index.html`).

17. `http://linkdata.jp/tutorial` (see for translation with Chrome) Converts data tables to RDF and publishes the work so that the world can appreciate a contribution. Federated SPARQL queries are automatically generated given a pathway query. Linkdata offers a solution for even more basic users than `Any23.org`, Triplify or D2R (converters to RDF triples). D2R Server is a tool for publishing relational databases on the Semantic Web by enabling RDF and HTML browsers to navigate the content of the database, and allowing applications to query the database it using SPARQL.

18. Dicoogle: Open Source information retrieval system for medical images using Indexing and P2P mechanisms and DICOM images (`http://www.dicoogle.com/`)

19. Jastor, an open source Java code generator that emits Java Beans from Web Ontologies (OWL) enabling convenient, type safe access and eventing of RDF stored in a Jena Semantic Web Framework model.

20. HTML Web forms don't separate the purpose from the presentation of a form. W3C XForms standard, in contrast, is comprised of separate sections that describe what the form does, and how the form looks. This allows for flexible presentation options, including classic XHTML forms, to be attached to an XML form definition.

21. BibSoup (`http://bibsoup.net/`), to find, manage and share bibliographies. BibServer (`http://bibserver.berkeley.edu/`), is a Python program which creates a network of displays of bibliographic data maintained in BibTeX by contributing authors and editors. Displays link whenever possible to full text in open archives such as arXiv and PubMed, in electronic journals, and on author's homepages. Links to

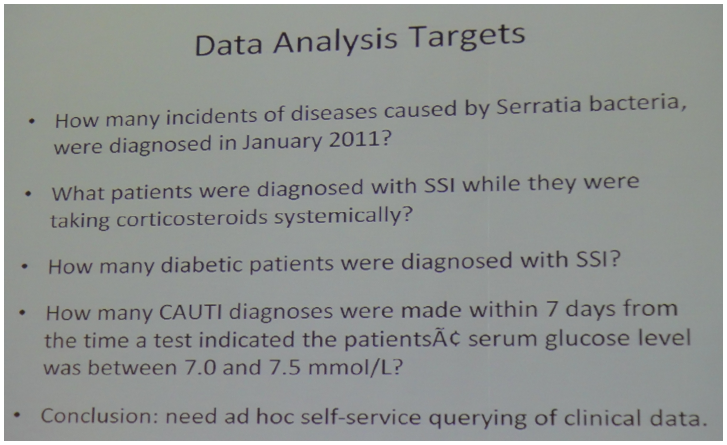

Figure 12: Examples of Data Analysis Targets

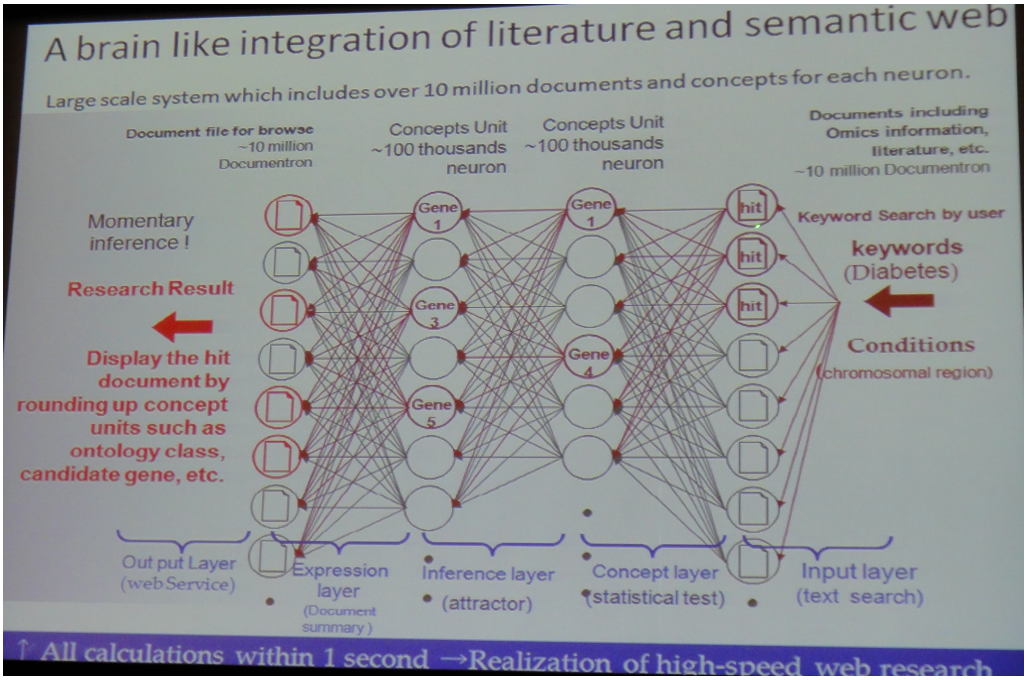

Figure 13: "A brain like integration of literature and semantic web".

related articles and citations are provided by Google Scholar. For articles in mathematics, links are provided if possible to abstracts and reviews in MathSciNet and Zentralblatt MATH.

22. ARC is a flexible RDF system for semantic web and PHP practitioners (for LAMP systems in general). It's free, open-source, easy to use, and runs in most web server environments. `https://github.com/semsol/arc2/wiki`.

3 SWAT4LS Tutorials 8.12.11

3.1 Tutorial: SADI (Semantic Automated Discovery and Integration) Luke McCarthy (C-BRASS).

SADI (`http://sadiframework.org`) is going to be submitted to W3C. WSDL (Simple protocol for services) and SAWSDL (Semantic Annotation of WSDL services) and OWL-S are existing services protocols. OWL-S and UDDI trust in intelligent web agents able to do everything. However, we need something more, e.g., even if bioinformatics web services

simplify their services, they tend also to ignore standards. OWL-S requires too much from the user for the value it provides while UDDI is more used in biomedical services. In general, existing standard are too complicated, too much work for little gain. SADI is presented with the slogan "there is nothing to it", since is more like good practices, standards and it does not have specific namespaces or protocols. All exists already: SADI uses simple HTTP operations, consume and produce RDF and uses OWL to describe service interface. The OWL class is not as important as its properties, which are described enough by people who uses them. SADI aims at providing explicit relationships between input and output and transparent service operation semantics, i.e., machine readability and reusability by other services.

One can dynamically identify input instances and assemble them to the service being sent to the server, which expands the input dataset. A service is identified by an HTTP URL (used also to invoke the service). (It does not work in Chrome). An example of OWL input class specification is `www.owl.com#NamedIndividual` and an output OWL class is `www.owl.com#GreetedIndividual`.

Requirements for SADI Compliance: 1. Responses to HTTP POST by invoking service: Input is an RDF document and output as well (containing OWL classes instances for each input). 2. A reasoner can be in the server through JENA in JAVA. 3. To call a SADI service: Client sends POST request to service URL. Output serialization is received through an ACCEPT.

Reasoners can automatically find input data to services. Modelling a service establishes the type of input and output (in graph patterns). The hardest part for service composition is having the data model, but otherwise, in general, is simple compared with existing semantic services mechanisms.

There is a difference between input to the service (which is data) and parameters to the service (which are not data). The first make the output to be produced. The parameter class in SADI serves to the service parameter. The SADI service description class can be found on the web.

In the SADI registry anyone can register a service; services are indexed by the properties they attach. A SPARQL endpoint, Java and REST APIs contain the services registered to be queried.

If a service wraps a third party, then it is doesn't have authoritative status. It represents a warning that the data might be outdated.

Property restrictions describe data consumed by the service; all property restrictions must be satisfied. Instances should be dynamically identifiable. According to the OWA (Open World Assumption), one cannot assume something to be true if one does not know if it is true. Therefore, it is recommended to use necessary and sufficient conditions. Avoid universal and exact/maximum cardinality restrictions. Min and max cardinality will be dynamically checked, but not exact cardinalities (e.g. Card = Exact (1)). The latter would never be dynamically signed.

In output OWL classes, property restrictions describe data produced by the service and ranges must be specified for discrimination of services dynamically. Instances don't have to be dynamically identifiable (no harm if they are), not even for service compositions since this is based on assertions of I/O. An example was shown to create a BMI service producer out from a given height and weight. The first step was creating input and output OWL classes with Protégé (Create new ontology in Protégé: Ontology IRI: `http://sadiframework.org/ontologies/training.owl`). There are Java, Perl and Python libraries for generating a SADI service skeleton (after specifying in the SADI Protégé plugin). Service logic is added with maven Eclipse. The IRI must be resolved in the web to define where the OWL document is held for the SADI registry. A direct import from local file can be done (e.g.: `MGED.measurements.owl`).

Create InputClass in same open ontology. We make it equivalent to a intersection of several object restrictions: Height is an intersection of a Measurement: has_height some Measurement and has_mass some Measurement. This transforms to: has_height some(has_units some Unit and has_value some string)) and (has_mass some (has_unit some Unit and has_value some string)). This means that we don't care if we have to input measurements objects but we care about them having a value and a unit. A blank node has no identity outside your data code, so it cannot be sent to the server.

Now for the BMI output, we create a Data Property (Double). Label = body mass

index. Comment is set to how it is calculated (formula). We create an outputClass, with data restriction, some...etc.

Ontology training.owl is added to the sadiframework svn repository to make it public. (scp training.owl `http://sadiframework/ontologies/`).

Automatic discovery is possible by just defining the services and invoking them. Data type conversion can be done invisible for the service user as well. When initializing the SADI code generator in Protégé, the SADI Service Signature needs to be filled (service provider: `www.sadiframework.org`). This is recommendable for asynchronous services. An email address of the service author must be added to be contacted if your service is having problems. Once generated the code, in Eclipse Java, import project from the folder where the generated code is (training.owl folder). We add the code to calculate the BMI in the generated empty class called processInput(Resource input, Resource output):

```
1 Resource height = Input.getRequiredProperty(Vocab.has_height).
    getResource();
2 Double heightInMeters = Double.parseDouble(height.getRequiredProperty(
    Vocab.has_value).getDouble());
3 Resource weightInKg= Double.parseDouble(weight.getRequiredProperty(Vocab
    .has_mass).getResource();
4 Output.addLiteral(Vocab.MBI, [formula]);
```

Run SADI services in Jetty Maven plugin for building. Description can be seen in bmi-input.rdf (e.g. ENTITY bmi: sadiframework.com/training.owl). One can include the test service in the service definition in Eclipse code (so that this is included in the jar path):

```
1 @TestCases(
2 @TestCase(
3         Input = "/t/bmi-input.rdf",
4         Output ="/t/bmi-output.rdf",
5 ))
```

There is no specific code generator for Python but it works with Twisted library (`http://twistedmatrix.com/trac/`) quite simply. SADI-0.1. python egg is available though it doesn't include reasoning. Therefore, it's extremely simple to write a service. Fact, e.g., could be added as Python reasoner library. The advantages of SADI versus SOAP is the reasoning that Java integrates with JENA. If we would like to get people with BMI over certain level. this would not happen at the service level but in a workflow. This encodings can be done for example for risk factors in diseases, by making it explicit in OWL classes for clinical analysis. As said previously, the hard part is to have a correct data model. Some examples of SPARQL queries on `http://dev.biordf.net/cardioSHARE/queries.html` show how to encapsulate external agents with expert knowledge (see complex query n.12). SADI hopes to support rules in the properties in the future, not to have ugly SPARQL queries. Regarding workflows, Taverna is a framework for services workflows (non semantic based). However, in Taverna, the benefits of SADI go away, but it can allow integration with external services.

The SADI activity plugin for Taverna allows connecting spreadsheets data as input to services. Some alternatives are Scaffold and t2flow Taverna plugins. SAWSDL, a standard for annotating existing WSDL, is similar to RDFa, providing a matching between RDF and XML. Its lifting schema lifts non-semantic XML into semantic output. The standard imposes no requirements. W3C suggests XSLT for mapping XML to RDF. The lowering schema lowers semantic input into non-semantic XML. XSLT is insufficient. W3C suggests SPARQL + XSLT).

For using SAWSDL with SADI, RDF input is lowered to XML input of the WSDL service, which is lifted with a schema to an RDF output. Lifting schema: XSLT, XPath + RDFPath. Lowering schema: SPARQL + (XSLT or Velocity (Template language for which knowing XSLT is not needed)). SADI SAWSDL Generator creates SAWSDL from existing WSDL service automatically (it generates SADI interface, including I/O OWL classes and a SADI-compliant service endpoint). "Why would you want to make manual WSDL and annotate it with SAWSDL?"

There are different SADI clients: SADI client API, SHARE query client, SADI Taverna plugin.

SPARQLAssist can be downloaded separately for adding autocompletition when querying a SPARQL endpoint.

See example in google code: FindAndCallServices.java. You can have multiple registries of SPARQL end points, SADI registry or other private services. Jena finds services by certain properties. invokeService sends minimal information. discoverInputInstances discovers your instances.

The example GenerateClassFromIndividual shows how the intersection of classes will find instances with those restrictions (by Jena).

At the moment, there is no real alternative to SADI, but the closest is SSWAP (Simple Semantic Web Architecture and Protocol): `http://sswap.info/protocol.jsp` The SADI Wiki explains how to add more SADI services as external services for your registry. When this is done, existing services from these registries become immediately available. Sadi.client.properties is the file to modify to add third-party web semantic services to do the closer implementation to the Observer pattern with subscriptions. A crawler for Semantic Web services would be useful but SW doesn't work like that yet. JDBC pointer to it, they need to be a SAWSDL or semantic web service (RDF based).

3.2 Tutorial: NCBO Web Services and Development of Semantic Applications. Trish Whetzel (Stanford University). (@bioontology)

The tutorial can be found in: `www.bioontology.org/wiki/index.php/SWAT4LS_Tutorial`. The National Center for Biomedical Ontology (NCBO) is presented (`www.bioontology.org`). BioPortal manages Biomedical ontologies. In REST Web Services, accessed via HTTP, each URL represents an object. BioPortal is accessible through code, the BioPortal UI or through a browser (`http://rest.bioontology.org`).

Ontology services in `http://bioportal.bioontology.org` include Search, Traverse, Comment or Download. Mapping services map concepts from one ontology to another (including Create, Upload and Download options). There is not a biomedical format for mapping but one need to specify it explicitly. Widgets include tree-view, auto-complete and graph-view. Annotation of web services provides term recognition in free text, based on ontologies (i.e., data access services fetch data annotated with a given term). It is possible to add NCBO widgets to any site.

An example showed, once you have got your API key by joining the portal, how to request a list of all latest versions of available ontologies ("Signature: ./ontologies?apikey= YourAPIKey") (also in `http://bioportal.bioontology.org/ontologies`). Example clients are available in Java and Perl and next bioPortal releases will produce query results in JSON. Mapped concepts IDs are resolved to the newest version of the equivalent concept. Some mentioned examples of applications were `http://sysmo-db.org/rightfield` or an ECG Gadget (`http://wiki.cvrgrid.org/index.php/ECGGadget` that stores, visualizes, annotates, analyzes, and displays analysis results for ElectroCardiogram data in, among others, HL7 format. ISAcreator (`http://isatab.sourceforge.net/`) assists in reporting local management of experimental metadata from studies employing one or a combination of technologies, being targeted to experimentalists, curators and developers.

BioPortal Import Plugin serves for reusing ontology parts in Protégé. The community portal has also a Notes web service to add terms proposals and comments on ontology terms. The Mappings web service, available at `http://bioontology.org/wiki/index.php/BioPortal_Mappings_Service`, is applicable for any ontology. Examples of visualization widgets applications shown are Redfly or Knowledge Egg. Other application mentioned was an annotator workflow with UMLS (Unified Medical Language System) and BioPortal ontologies, an application for RELN Brain development or an Statistical Tracking of Ontological Phrases (STOP) with NCBO Annotator.

There is an interesting series of NCBO webinars at `http://www.bioontology.org/webinar-series`. "Making Sense of Unstructured Data in Medicine Using Ontologies "could be highlighted, for showing how, from electronic health records of adult patients from the STRIDE Clinical Data Warehouse, one can identify statistically significant patterns of drug use to conduct drug safety surveillance.

Other project within NCBO is ODiSSea (Ontology Driven Semantic Search), which enables users to efficiently locate biomedical data resources. Some other NCBO related links are:

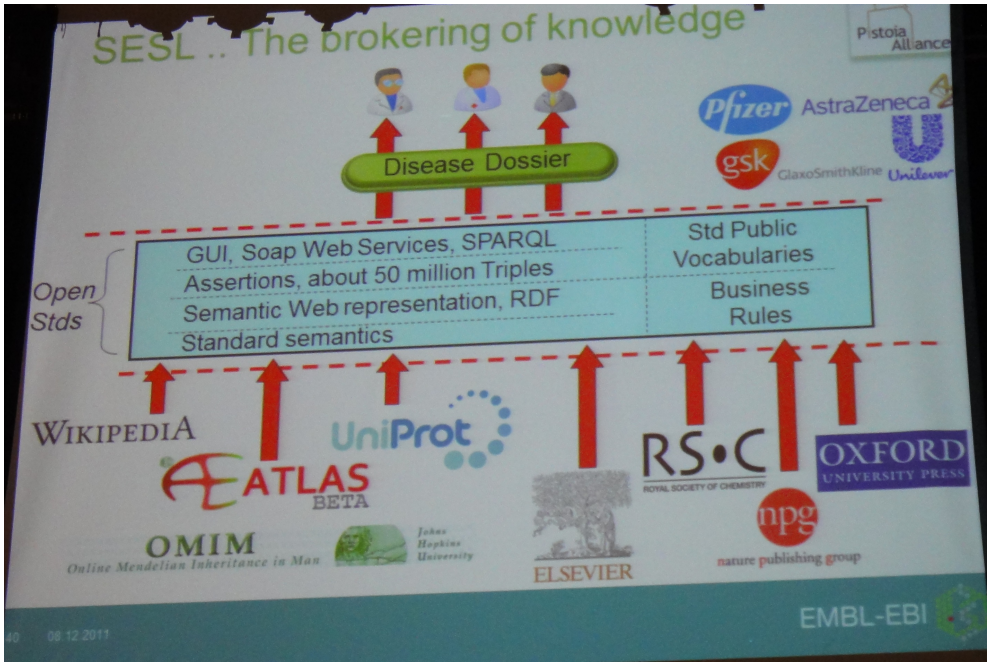

Figure 14: Overview of the SESL (Semantically Enriched Scientific Literature) Project at EBI (European Bioinformatics Institute).

- BioPortal SPARQL Endpoint with sample code: `https://github.com/ncbo/sparql-code-examples`.
- Documentation: `http://www.bioontology.org/wiki/index.php/SPARQL_BioPortal`.
- Web service documentation: `http://bioontology.org/wiki/index.php/NCBO_REST_services`.
- National Center for Integrative Biomedical Informatics (NCIBI): `http://portal.ncibi.org/gateway/`.

3.3 Tutorial: Using SPARQL with UniProt RDF. Jerven Bolleman (Swiss Bioinformatics Institute)

In this tutorial TopBraid (which has SPARQL support) was used. Reasoning is supported through OWL-RL and SPIN rules. The tutorial can be downloaded from `http://t.co/uSMbLGGd`.

3.4 Tutorial: Perennial identification, Nick Juty, EBI. MIRIAM registry and identifiers.org

"Identifiers.org is an annotation and cross-referencing framework which provides perennial, unambiguous and resolvable identifiers. The system is built on the information stored in the MIRIAM Registry, which catalogues a community developed shared list of dataset namespaces". To identify a collection of records, go to `http://identifiers.org/ec-code/`.

3.5 Tutorial: Integration of the scientific literature into the Semantic Web: facts from biomedical data resources, Dietrich Rebholz-Schuhmann (European Bioinformatics Institute (EBI))

This tutorial talked about biological context analysis. Within the SESL (Semantically Enriched Scientific Literature, `www.pistoia-sesl.org`) project, in which pharmaceu-

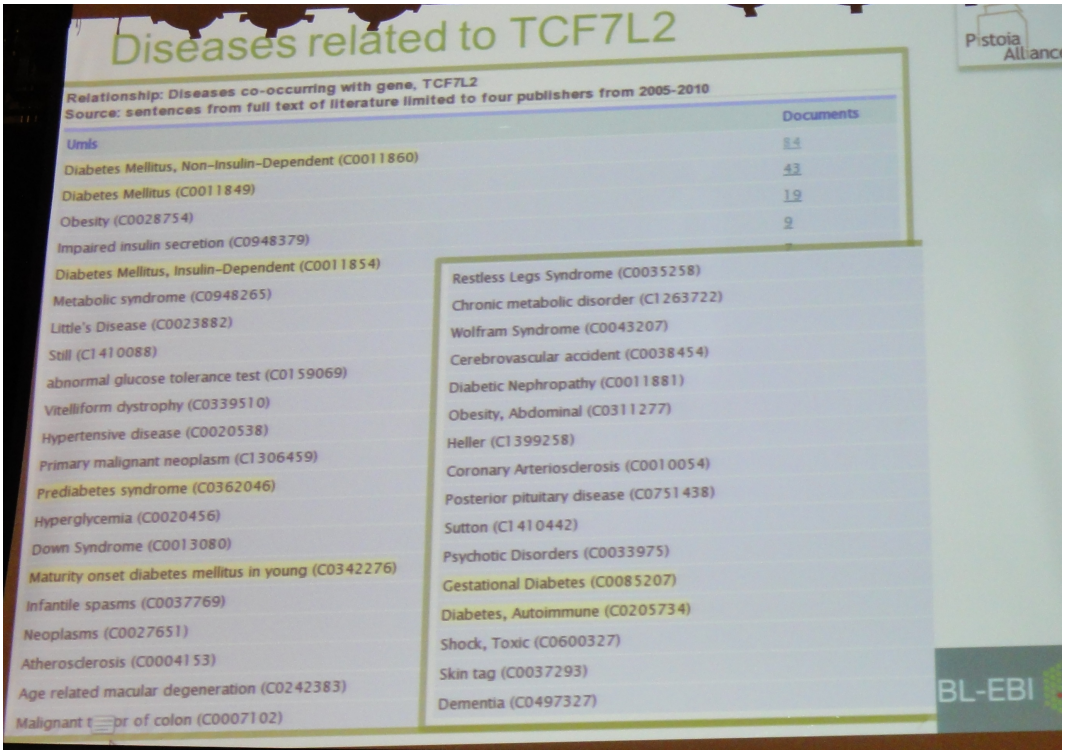

Figure 15: Example of relationship of diseases co-ocurring with genes obtained from full text literature.

tical companies are involved, the problem was: What evidence is available for gene-disease relations?

OMIM (Online Mendelian Inheritance in Man) is a database of human genes and genetic disorders. Open data (`OpenRDF.org`) with Sesame triple store was used in this application. 50 million Triples were made accessible via SOAP web services and SPARQL. Usual GUI was found to be a problem to integrate different resources due to long response times. Therefore, all data was integrated and it is now possible to query, e.g., which genes co-occur with which diseases is the SPARQL end point (and also UI).

Pistoia Alliance, Nature publishing group, Elsevier, ATLAS beta, IntAct, Disease Ontology, EBIMed and GO (Gene Ontology) Annotation are names of institutions and applications involved for a databases interaction through text mining.

Whatizit is the text mining tool used, built at EBI. The article "Text Processing through Web Services" (`http://bioinformatics.oxfordjournals.org/content/24/2/296.full.pdf`) can be seen for more information. So far, no reasoning behind Pellet or Jena is used; however, CB-Reasoner (`http://code.google.com/p/cb-reasoner/`) is suggested as a case applied to phenotypes, being 10 times faster than other general reasoners.

3.6 Tutorial in SWObjects

`https://sites.google.com/site/swobjectstutorial/files`.

This tutorial shows how to query MySQL through SPARQL using SWObjects. Often, databases such as CHR, Entrez Gene and Drugbank contain RDF data but however, they need to be queried separately. Diseasome project `http://diseasome.eu/` links genes to known diseases and disorders, indicating the common genetic origin of many diseases. The problem with drug discovery is proposed to be tackled through query federation. In this example, three SPARQL end points were opened, each on a different terminal window [-d for indicating dataset to use]. If, for instance, we would like to link a city with DBPedia URI without going to query DBPedia externally, the CONCAT SPARQL function can be used. SERVICE is a SPARQL 1.1 keyword that specifies the location of the endpoint. This is useful when, e.g., in genomics, datasets are so big that would be infeasible to download all

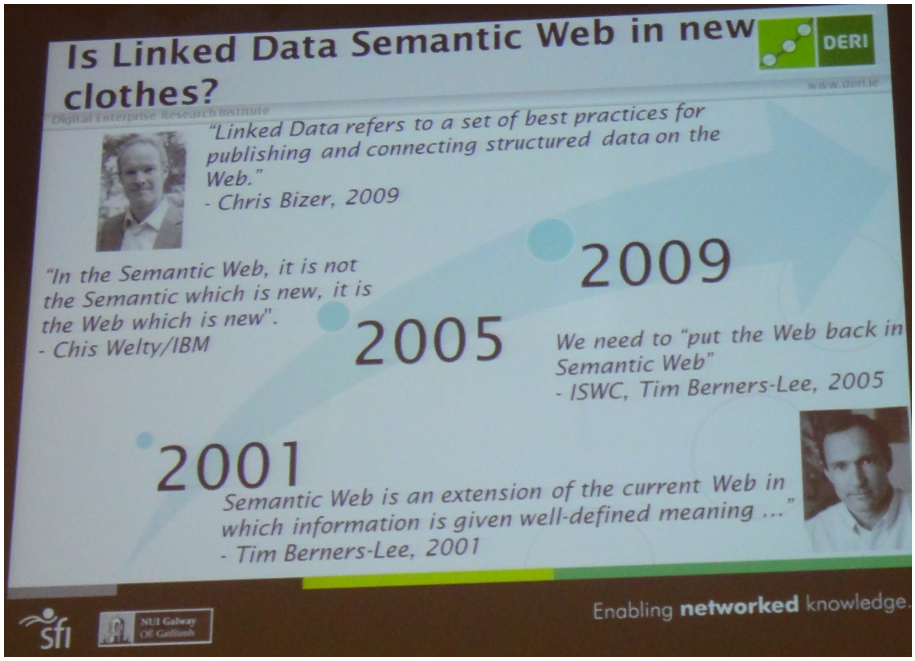

Figure 16: "Is Linked Data Semantic Web in new clothes?"

data. A file goProt.map needs to be created to specify the transformation of the SPARQL query into the SQL query automatically.

A list of the different SPARQL end points available at the moment, together with their status, can be seen in `http://labs.mondeca.com/sparqlEndpointsStatus`. Vocabularies and topics range from meteorological datasets to BBC program finders.

4 My contribution and comments

I presented the following poster in SWAT4LS:

Programming biomedical Smart Space applications with BioImageXD and PythonRules. Natalia Díaz Rodríguez, Pasi Kankaanpää, M Mohsin Saleemi, Johan Lilius, Ivan Porres. 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences (SWAT4LS 2011) , 7 - 9 Dec.

The poster can be seen in this address and in figure 17:
`http://f1000.com/posters/browse/summary/1089680`.

4.1 Poster Comments and Ideas

Could DrugBank Linked Data be used for adding a collaborative cancer image annotation task to BioImageXD?

Could we reuse any existing bioimaging ontology (from NCBO) to publish BioImageXD data and processes from the different scientific community in form of linked data? Juha Muilu from the Institute for Molecular Medicine Finland (FIMM) showed interest in an ontology which could represent BioImageXD capabilities. Their interests concern ontologies for annotating images (`http://www.fimm.fi/en/research/research_groups/lundin_group/`), data formats as a goal for transforming data into RDF using existing ontologies (`varioml.org`) and collaborations with Turku Biocenter and hospital regarding biobanking developments (`bbmri.fi` and `bbmri.eu`).

Diseasecard project (introduced by Pedro Lopes, together with COEUS) is open for hosting linked data from biomedical databases for allowing integration or federation with other sources. Could BioImageXD software provide experts' assessments in form of annotated analysis so that these are linked with a extended knowledge granularity via the biomedical open community?

A proyect related to PythonRules is the suggested IFTTT (If this, then that) `http://ifttt.com`), a tool for creating shareable and social applications recipes.

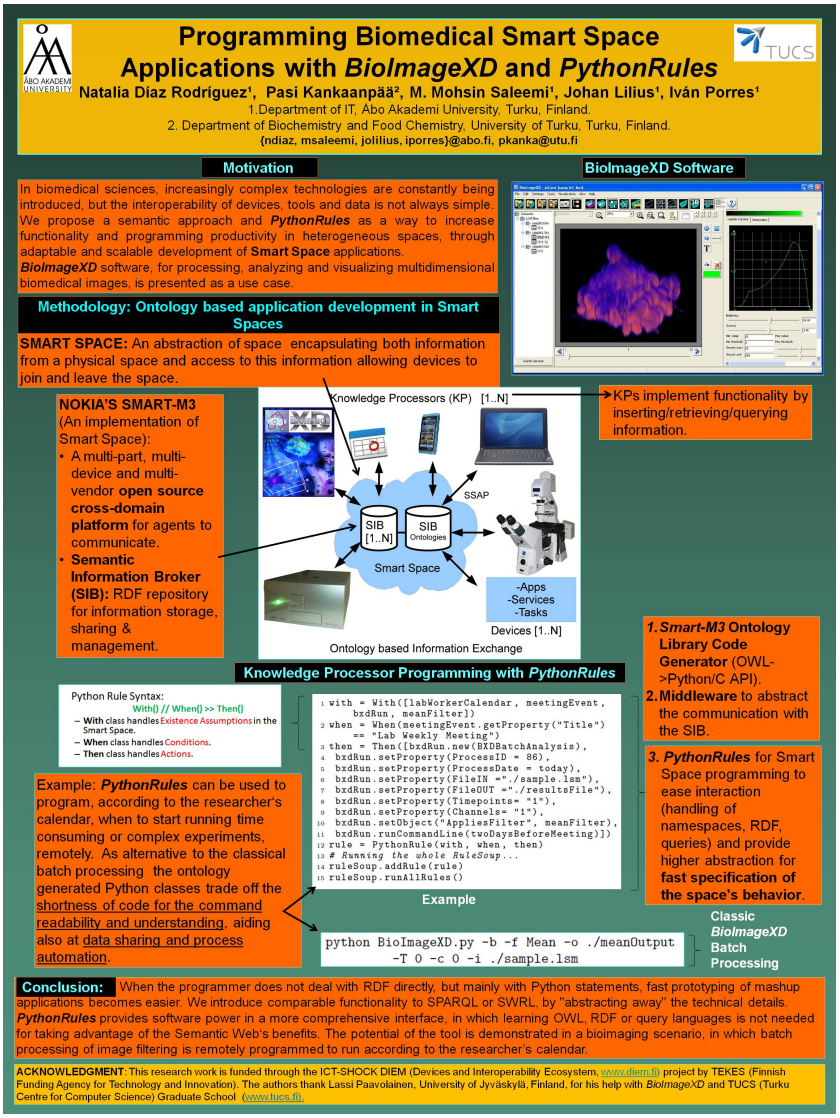

Figure 17: My poster on "Programming biomedical Smart Space Applications with BioImageXD and PythonRules".

5 Summary

Life science researchers need to deal with data in different formats to perform their daily research, but it is simply not possible for a single human mind to analyze all these data. The integration of data in the life sciences is a key component in the analysis of biological processes [1].

Diverse interests were observed in the SWAT4LS community, from bioinformatics and medical informatics to systems biology and knowledge discovery for drug design. Both academia and business show successful experiences with Semantic Web technologies. The Semantic Web has demonstrated to work effective and cleanly in applications around mutations, phenotypes, Gen2Phen (Genotype to Phenotype Databases), microarrays, GO (Gene Ontology) applications, lipids, pathway annotation [1], etc. However, this is just a small sample from the bioinformatics area efforts; there is still plenty to do on spreading the semantic web attitude and conveying its benefits to the rest of the science community. Data curating must be community driven, and open standards are crucial in these tasks.

6 "Take home messages"

"Get your data out on the semantic web, even if imperfect. 'Scruffy works'." - Wendy Hall.

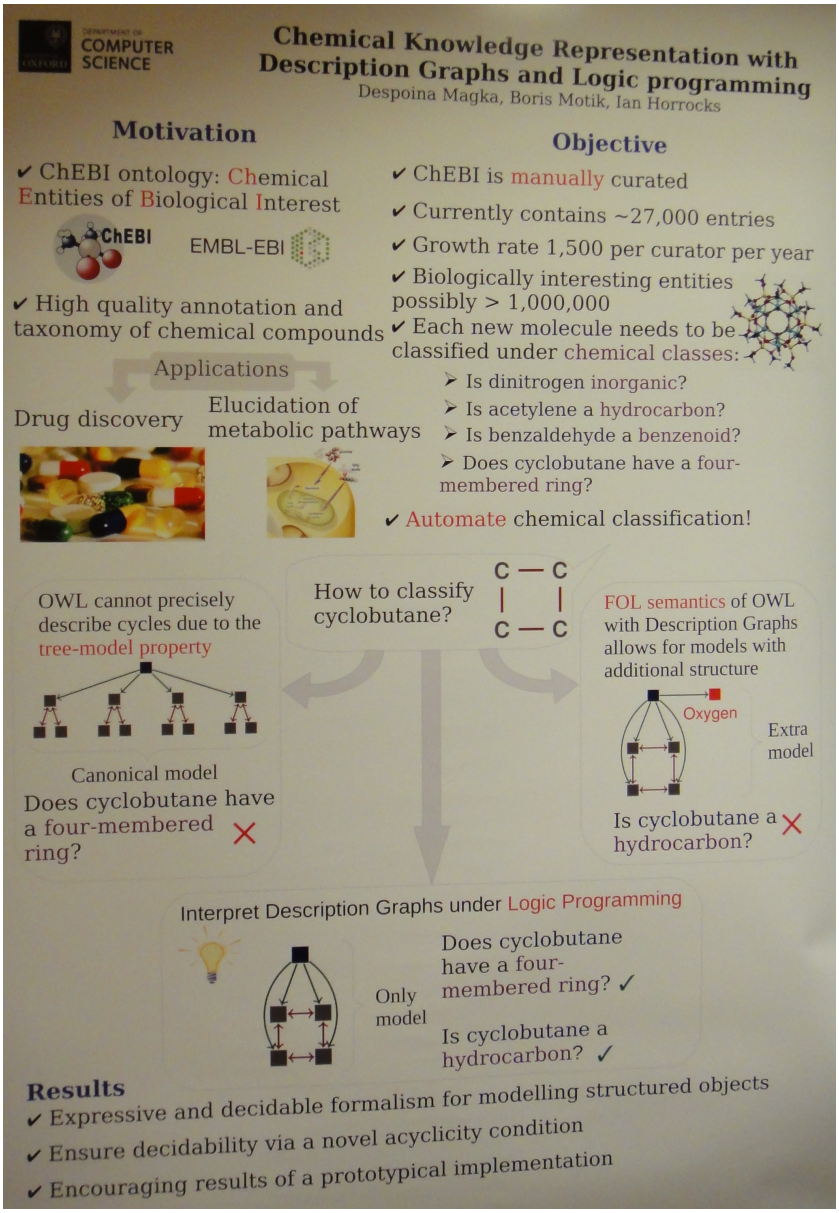

Figure 18: Poster on "Chemical Knowledge Representation with Description Graphs and Logic Programming".

"Biology and Computers are not so different". "Is Linked Data Semantic Web in new clothes?".

Data can be linked to help curing diseases. Superbugs can be tracked, visually through SPARQL queries, across databases, aiding a faster decision making process. If two persons have a same gene and an infection is detected in Portugal, the second patient in UK should be put in quarantine immediately. Some advantages of Semantic Technologies are, e.g., to be able to pick up a gene and obtain all information associated with it or being able to access this information as soon as it is available.

It is good to have metadata and provenance of all metadata; automate each step (transcription, translation, etc.).

"When studying software usability, i.e., what users want, and how they get to it through the UI, would not be much easier to achieve it by tracking speech interfaces than through complex menus?"

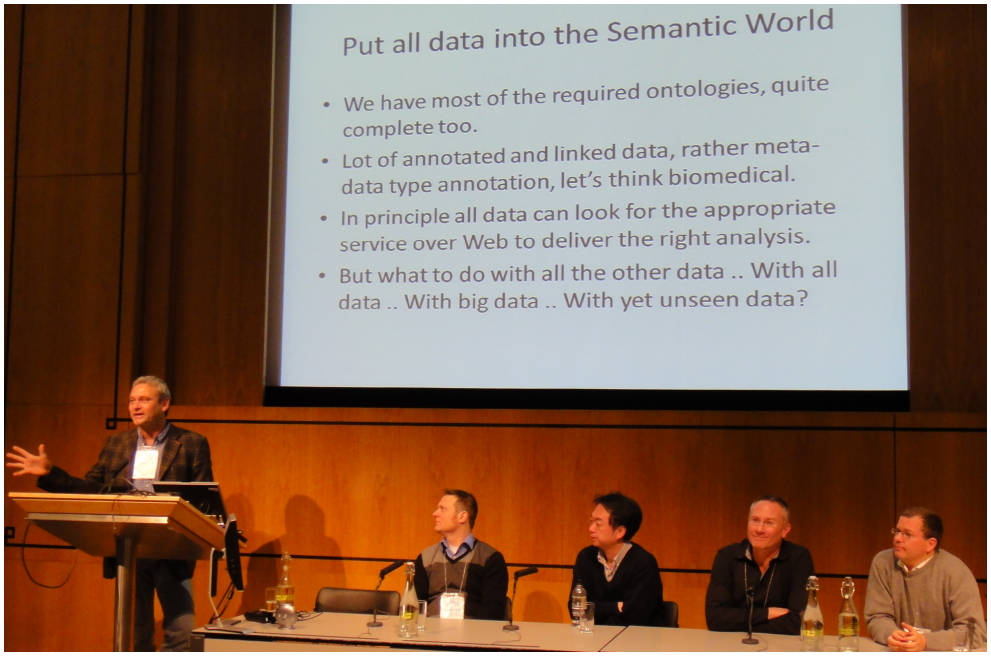

Figure 19: Final Panel Session Questions (1)

References

[1] I. N. Delgado, A. Real-Chicharro, M. A. Medina, F. Sanchez-Jimenez, J. F. Aldana-Montes, A. Mechouche, X. Morandi, C. Golbreich, and B. Gibaud. Social pathway annotation: extensions of the systems biology metabolic modelling assistant. 2010.

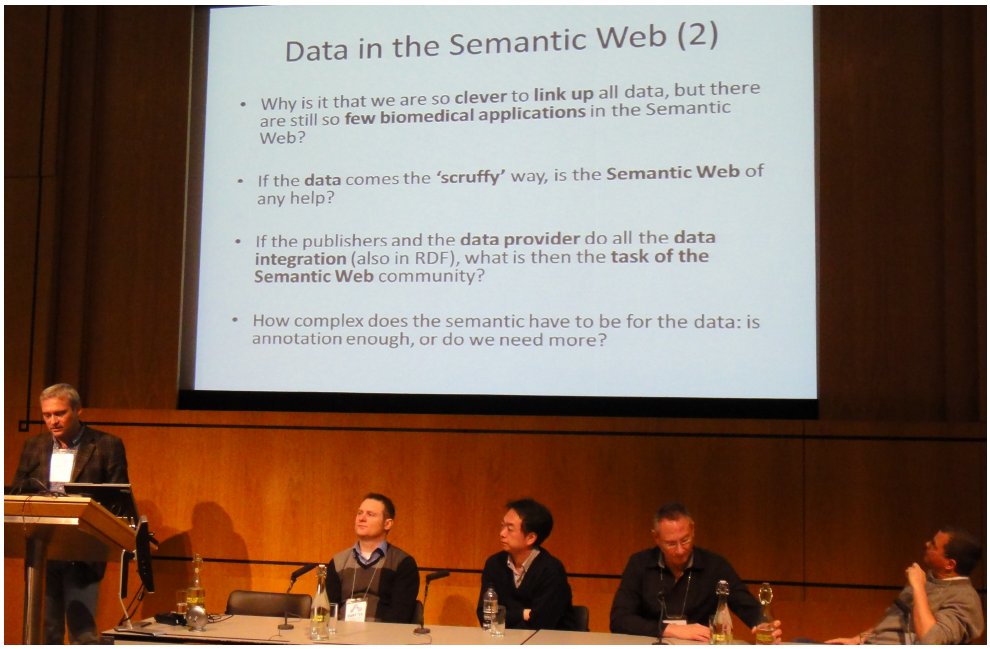

Figure 20: Final Panel Session Questions (1)

Figure 21: Workshop Dinner with British Christmas hats at Covent Garden.

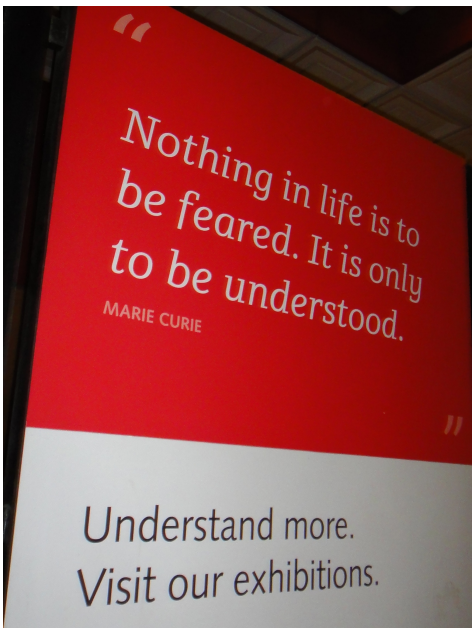

Figure 22: Take home message from library exhibition in St. Pancras.

Working Groups/Science/Disease Localiser

< Working Groups | Science>

http://wiki.okfn.org/Working_Groups/Science/swat4ls_hackathon
http://wiki.okfn.org/Working_Groups/Science/Disease_Localiser

Contents

Cancer geolocalization

Could we have a geographical map about different kinds of cancer (neoplasma, carcinoma distributions) for patients to see openly where similar cases were treated and where these experts are located?
Could we expand EpiSpider with an agent for cancer?

Key Ideas

Open Access by patients who give their own information
Accesible by Mobile app for example by http://www.healthnet.org/
Would be useful to have for people with phones but not computers, to use.

Tools we could use for visualization

Spark tool for google maps visualization.

To visualise JSON data in CytoScape (http://www.cytoscape.org/) via the web plugin like the Gene-Drug Visualization group did.

The literature suggests GIS (Geographic information systems) and other spatial analytic methods to provide solutions in cancer control.

Example of similar existing applications

http://healthnet.org/ as a way for gathering accurate data.
http://healthmap.org from promedmail.org is a narrow example, non semantic.
EpiSpider is a semantic application based on geographical infectious diseases. It does not include cancer.

http://www.epispider.org/index.php

Collaborative science framework Harvard Medical School: http://sciencecollaboration.org/

Daylife API for mashups: http://developer.daylife.com/
Promed: http://www.promedmail.org/ It does not include academic articles, only news. Only infectious diseases.

Tools for getting RDF data

D2R server. Relational DB to RDF. Also any23.org or Triplify.

http://linkdata.jp/ To transform plain data (tables or excel) to RDF. Used for homogeneizing different scientific DBs.

Tool for creating Linked Data from a SPARQL End point: http://www4.wiwiss.fu-berlin.de/pubby/

Linked data databases for diseases: Diseasome, LinkedCT Clinical trials.

Examples of Databases

Example of DB for Cancer Somatic mutations: TP53/IARC LOGVD SPARQL end point. NCIT namespace (National Cancer Institute). http://bioinformatics.istge.it/logvdsparql/ 24772 individuals available with their nationality, age and smoking habits.

Catalogue Of Somatic Mutations In Cancer, COSMIC: http://www.sanger.ac.uk/genetics/CGP/cosmic/

Issues

Privacy of people versus the interest to socialize and find common results.
Existing solution tool: http://www.patientslikeme.com/

Research group

Natalia Díaz Rodríguez Department of Information Technologies. Åbo Akademi University

Tomasz Kluza Poznan University of Life Sciences Poland

Przemyslaw Jan Nowak Poznan University of Life Sciences Poland