Deliverable D4.7

Project Title:	Building data bridges between biological and medical infrastructures in Europe
Project Acronym:	BioMedBridges
Grant agreement no.:	284209
	Research Infrastructures, FP7 Capacities Specific Programme; [INFRA-2011-2.3.2.] “Implementation of common solutions for a cluster of ESFRI infrastructures in the field of "Life sciences"
Deliverable title:	Report on the scalability of semantic web integration in BioMedBridges
WP No.	4
Lead Beneficiary:	1: EMBL
WP Title	Technical Integration
Contractual delivery date:	31 December 2014
Actual delivery date:	19 December 2014
WP leader:	Ewan Birney
Contributing partner(s):	1: EMBL

Authors: Julie McMurry, Simon Jupp, James Malone, Tony Burdett, Andy Jenkinson, Helen Parkinson, Mark Davies, Marco Brandizi, Sarala Wimalaratne, Nicole Redaschi, Chris Morris, Martyn Winn

Contents

Figures

Tables

1 Executive Summary

The overwhelming majority of the world's data is stored in relational databases; however, direct integration of these databases is hampered by the heterogeneity of schemas, terminologies, identifiers, and the granularity of representation. European e-Infrastructure projects are increasingly turning to the "linked data" approach to address these challenges with semantic web technologies. The semantic web uses RDFS and OWL, schema languages that are designed from the ground up to connect disparate data sources. This approach is proving to be a solution to some of the emerging challenges in the life sciences. The BioMedBridges semantic web pilot spans several deliverables[1]; its goal is to test the suitability of a semantic web approach to the task of integrating research data and to report on our experience of running an RDF-based platform integrating multiple data resources.

In order to leverage experience where it exists and minimise the risks inherent in novel technology projects, a three-stage delivery was chosen for the pilot. As summarised below, these stages are reported on in separate deliverables.

[1]
4.4 Identification of feasible BioMedBridges pilots for semantic web integration
4.6 Pilot integration of Web Services based simple object queries
4.7 Report on the scaleability of semantic web integration in BioMedBridges
4.8 Report on Web Services based integration of BioMedBridges integration across all appropriate services

Table 1 Overview of 3-Phased Semantic Web Pilot

	Del. no.	Due	Focus	Project partners	Nature of the activities
Prep	4.4	2013	Planning	EMBL-EBI (ELIXIR), HMGU (Infrafrontier), TUM-MED (BBMRI), STFC (Instruct), UDUS (ECRIN)	Development of a roadmap for the semantic web pilot project overall
Phase 1	4.7	Dec 2014	SemWeb scale-ability	EMBL-EBI (ELIXIR)	ELIXIR establishes mature semantic web services, basic best practices, and technical guidelines. Benchmarks technology and assesses scaleability
Phase 2	4.6	June 2015	Data integration	ErasmusMC (Euro-Bioimaging), EMBL-EBI (ELIXIR), HMGU (Infrafrontier), TUM-MED (BBMRI), STFC (Instruct), UDUS (ECRIN), VUMC (EATRIS)	Infrastructures implement specific pilot projects in parallel, according to their individual roadmaps, using the technical guidelines and outcomes from phase I
Phase 3	4.8	Dec 2015	Data integration	ErasmusMC (Euro-Bioimaging), EMBL-EBI (ELIXIR), HMGU (Infrafrontier), TUM-MED (BBMRI), STFC (Instruct), UDUS (ECRIN), VUMC (EATRIS)	Infrastructures further coordinate efforts, align these solutions to other broader standards in the eScience, build systematic connections

ELIXIR’s role in the semantic web pilot includes forming a blueprint for other infrastructures to build upon (Phase 1). This first stage is intended to leverage the experience gained from recent projects at EMBL-EBI, SIB, STFC and OpenPhacts that have delivered semantic web components at various scales and in different institutional environments.

By following this schedule and aligning partner-specific roadmaps to the blueprint (Phase 1), the pilot projects will be developed synchronously and knowledge will be shared efficiently (Phases 2-3). This will enable more infrastructures to collaborate effectively and address common issues. To support this effort, a knowledge-exchange workshop ran on the 29-30 April 2014 at TMF, Berlin, Germany. The programme and course materials are available on the BioMedBridges website[2]. In December 2013 and December 2014 we ran tutorials at SWAT4LS[3] during which we demonstrated the queries and analyses the new RDF platform makes possible, and we provided a training course to Computer Scientists in Manchester on the practicalities of working with and running the RDF platform. All of the SemWeb pilot work is informed by the work of BioMedBridges WP3 concerning the choice and use of ontologies and use and re-use of identifiers.

2 Project objectives

With this deliverable, the project has reached or the deliverable has contributed to the following:

No.	Objective	Yes	No
1	Implement shared standards from work package 3 to allow for integration across the BioMedBridges project	x	
2	Expose the integration via use of REST based WebServices interfaces optimised for browsing information		x
3	Expose the integration via use of REST based WebServices interfaces optimised for programmatic access		x
4	Expose appropriate meta-data information via use of Semantic Web Technologies	x	

[2] http://www.biomedbridges.eu/trainings/knowledge-exchange-workshop-resource-description-framework-rdf
[3] http://www.swat4ls.org/workshops/berlin2014/

3 Detailed report on the deliverable

3.1 Background

This deliverable report will cover the activities of the first phase of the Semantic Web Pilot, in which ELIXIR has provided the following:

- Examples of technical architecture suitable for operating semantic web services. This illustrates the various software components and usage considerations
- Case studies of RDF transformation projects from the BioMedBridges project (Biosamples DB and Gene Expression Atlas) as well as other diverse resources such as UniProt, and ChEMBL. Several such services have already been made available in RDF form or are currently undergoing this activity. The lessons learned from these experiences will be presented in case studies
- Comparison of triplestore solutions from major vendors. This includes performance benchmarking, along with ancillary technical considerations such as data replication and data loading mechanisms
- Technical guidance documents for best practice in areas such as URI selection
- Guidance for integrating external data including strategies for the use of ontologies
- Development of a generic model for expressing dataset provenance

3.1.1 Overview of Semantic Web technologies

The basic structure of the Resource Description Framework (RDF) is simple: everything is represented as a 'triple' - two things related by some relationship (the relationship being the third thing).

Subject	**uniprot protein P15056**	**uniprot protein P15056**
Relationship (aka 'Predicate')	*is_product_of*	*is_reagent_in*
Object	gene X	reaction pathway Y

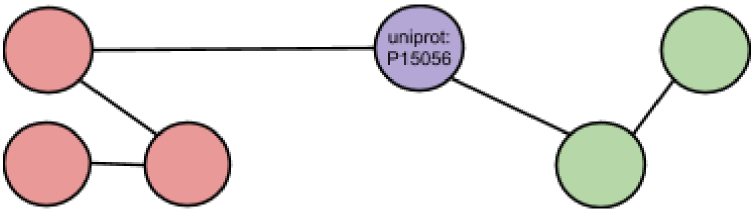

Figure 1 Two datasets both include triples referring to the UniProt protein "P15056" concept can be intrinsically integrated

RDF not only allows things to be connected, but enables the connections themselves to be named and described using ontology concepts. RDF is well suited to complex integration of data, in part because it can natively maintain a representation of ontology structures. These factors make RDF an especially suitable platform for biological discovery processes.

The concepts that are used in triples are expressed as Uniform Resource Identifiers (URIs). These are globally unique identifiers for any concept, be it a person, a place, a document, a protein or an abstract idea like "subcellular location". Because URIs are globally unique they can be shared between multiple datasets. Because RDF databases can load any triples, any two datasets mentioning the same URI are intrinsically integrated from the moment they are loaded. This allows data to be retrieved across dataset or domain boundaries with a single query and without requiring any changes to the database. For instance, UniProt protein P15056 can be represented as the product of a gene and also a component in a reaction pathway (Figure 1).

3.1.2 Approaches to RDF provision

Although RDF facilitates data interoperability and provides explicit semantics, most data is not in a native triplestore. This is because the technology is not as mature, and because there are few people skilled in it, making access to expertise and documentation a limiting factor. We have therefore contributed to training courses and materials, for which interest has been substantial. The following section will be incorporated into training materials currently in development. There are four basic approaches to providing RDF; examples of each approach are provided in

RDF-on-demand

RDF-on-demand means that an RDF representation of a concept is constructed “on the fly” in response to a specific request, e.g. a RESTful HTTP request. An application providing RDF in this way would typically access data about one data item (e.g. a protein) from a relational database and convert it to an RDF-compatible format.

Most such mapping from RDB to RDF to date has been done using ETL (Extract Transform Load). It is now well established[4] that it can be very beneficial to do on-demand translation from relational data to an RDF representation. RDF offers a systematic ontology and a query language which can be used against the mapped data, without concern for the semantic heterogeneity inherent in independently arisen relational databases. Among other benefits, on-demand mapping removes the burden of maintaining yet another data warehouse. Datasets with complex schemas may be poorly-suited to SPARQL endpoints given the difficulty converting complex SPARQL queries into DRB query languages (e.g. SQL).[5]

[4] The mission of the W3C’s RDB2RDF Working Group, part of the Semantic Web Activity, is to standardize languages for mapping relational data and relational database schemas into RDF and OWL. The two languages are the Direct Mapping (DM) and the RDB2RDF Mapping Language (R2RML).

[5] http://sadiframework.org/content/links-and-docs/

Flatfile conversion

Many biological databases are comprised of flat text files submitted by individuals. Text files, whether TSV, CSV.

D2R/SPARQL to RDBMS query conversion

The D2RQ Platform[6] is a system for accessing relational databases as virtual, read-only RDF graphs. Once configured, SPARQL queries can be automatically converted to SQL queries. It therefore offers RDF-based access to the content of relational databases without having to replicate it into an RDF store. Using D2RQ you can:

- query a non-RDF database using SPARQL
- access the content of the database as Linked Data over the Web
- create custom dumps of the database in RDF formats for loading into an RDF store
- access information in a non-RDF database using the Apache Jena API

Native RDF

A native RDF database has the advantage that it can exploit the specific characteristics of the RDF data model, the RDF data, and the query language. RDF databases can be designed to take advantage of some RDF features such as unique resource identification by URI, the ability of a resource to be the subject of many statements due to the ability to store many properties.

Complete RDB-to-RDF Conversion

In a complete RDB-to-RDF conversion (aka ‘replication’), the RDB maintains the primary database format and RDF is periodically exported as a snapshot. This approach enables the novel and mature technologies to complement each other, while also minimising risk. This conversion is generally done from object models. Software libraries that may be useful in this process are listed in Appendix 2, point 5.

[6] http://d2rq.org/

3.2 Case studies

Introduction[7]

Facilitating application development. The ELIXIR RDF Platform[8] has been a coordinated effort, developed at the direct request of industry partners and thereby connecting BioMedBridges to industry requirements. More specifically, it is aimed at developers of biomedical applications. The hope is that the rich querying that can be performed will lead to useful applications which hide some of the complexities of SPARQL and instead allow a user to interact with the data in a more familiar mode. Examples for such a queries are: *"What are pathways that a gene of interest may be involved in? What diseases are connected to the proteins that this drug is targeting?"* or "*What types of genes are there in the Ensembl database and what external databases are they cross referenced to*"; a query for which a web application was recently developed[9] (results shown in Figure 2).

[7] This section excerpted and paraphrased from http://drjamesmalone.blogspot.co.uk/2013/10/announcing-EMBL-EBI-rdf-platform-but-what.html

[8] The EMBL-EBI RDF platform: linked open data for the life sciences. Jupp S, Malone J, Bolleman J, Brandizi M, Davies M, Garcia L, Gaulton A, Gehant S, Laibe C, Redaschi N, Wimalaratne SM, Martin M, Le Novère N, Parkinson H, Birney E, Jenkinson AM (2014) doi:10.1093/bioinformatics/btt765

[9] http://www.ebi.ac.uk/~jupp/d3sparql/d3sparql-sankey.html

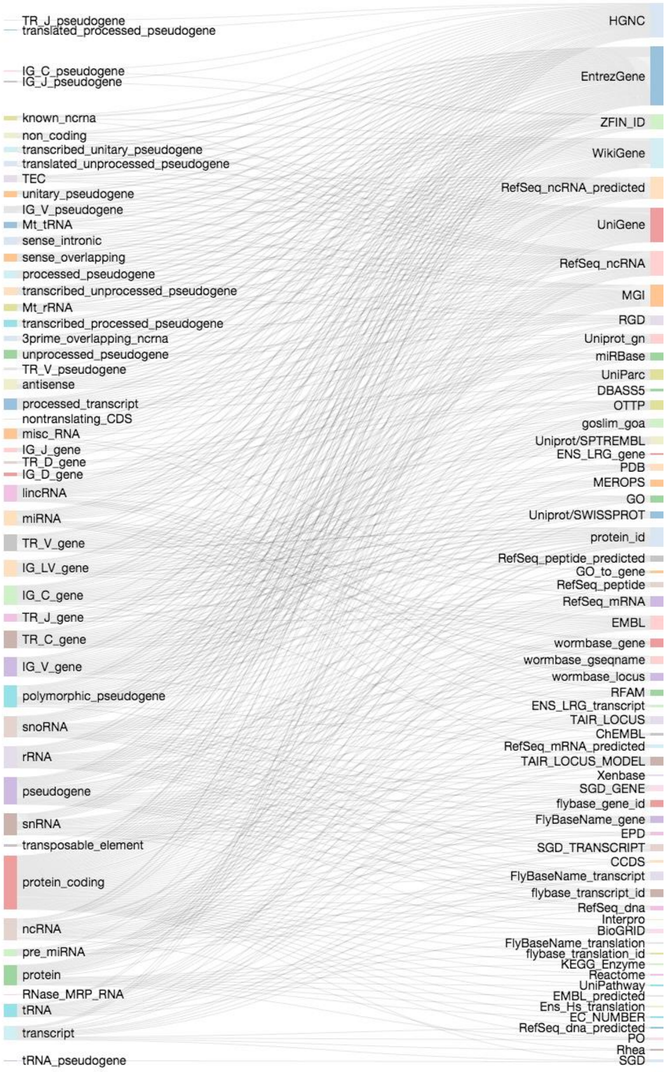

Figure 2 Visualisation of Ensembl genes and their cross-referenced databases

Because application development was the key thing we wanted to support from the outset, it was especially important to release the RDF platform with a commitment to quality and reliability as well as an abundance of documentation. We recently launched our first RDFApp - an R package for querying the Atlas RDF.[10] We will run an entry-level training course at in the spring of 2015. We will also hold an Appathon event in the near future.

[10] http://www.ebi.ac.uk/rdf/atlasrdf-r-package

3.2.1 Overview of case studies

Consistent with WP4 objectives 1 and 4, nine sets of data (Table 2) were converted to RDF, together comprising nearly one billion triples (Table 3). Five of the nine datasets are now production-grade and part of the ELIXIR RDF platform[11] while the remaining four are still in Beta. Although related, the datasets themselves were built for different purposes and are at different levels of maturity; architectural decisions were therefore made about the degree of unification (whether dataset centric or service centric). A hybrid model was selected. For the five of seven conversions done via replicating the RDB as RDF: we built five separate Virtuoso instances, each with its own SPARQL query endpoint. The remaining two datasets are converted on demand by way of query transformation. Although there was coordination on the level of common vocabularies, more emphasis on SIO specifically, would have been helpful. This report summarises our process and lessons learned.

For the five datasets in the ELIXIR RDF platform, we developed a number of technical components including a Drupal website[12], a SPARQL endpoint, and a Linked Data browser which provides a human-readable view of any resource[13]; each of these platform components is backed by a Virtuoso triplestore. Each component is operated within the EMBL-EBI's VMWare virtualised infrastructure as this approach is well supported within EMBL-EBI and more efficient and robust than the physical server alternative. Non-EMBL-EBI partners within the BioMedBridges consortium have made effective use of this virtualised infrastructure[14] as well (see D4.5). In addition to the virtualisation, the RDF platform components are load-balanced to guard against spikes in demand and Virtuoso failures. An overview of the RDF platform components is shown in Figure 3. Details of these components can be found in Appendix 1.

[11] http://www.ebi.ac.uk/rdf/platform
[12] http://www.ebi.ac.uk/rdf/platform
[13] e.g. http://rdf.ebi.ac.uk/dataset/biomodels/28
[14] http://www.ebi.ac.uk/about/news/press-releases/Global-Alliance

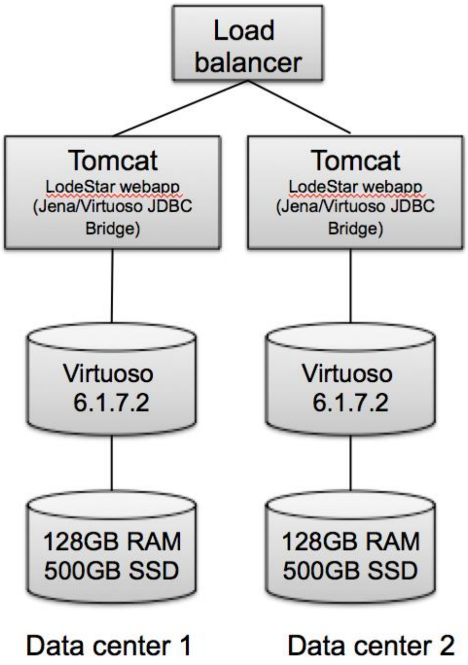

Figure 3 Diagram of RDF triplestore back-end

The range of datasets is continuing to expand, and at the same time external demand is increasing. Formal usage metrics are being collated for the next BioMedBridges periodic report. The services and their respective documentation are listed in Table 2 with an example query made possible by the RDF.

Table 2 Overview of RDF data integration

Datatype	Database	Example query	Data integration	Ontologies
Biological models	BioModels	All model elements with annotations to acetylcholine-gated channel complex (GO:0005892)	Taxonomy, EMBL-Bank, UniProt	DO, EFO, HPO, ICD9, ICD10, OMIM, GO, SBO, ChEMBL-EBI

Datatype	Database	Example query	Data integration	Ontologies
Biological samples	BioSamples	Samples treated with alcohol	Expression Atlas, Uniprot	EFO, ChEMBL-EBI, NCBI taxon, ICD10, OMIM
Drugs and other chemicals	ChEMBL	Find drug-like (but currently not approved) molecules which bind 7TM1 GPCRs with high affinity	UniProt, UniChem Sources (DrugBank, PDBe, IUPHAR, PubChem, Kegg Ligand, ChEMBL-EBI, NIH Clinical Collection, Zinc, eMolecules, IBM Patent Structure, Gene Expression Atlas, IBM Patent, FDA/USP SRS, SureChEMBL, PharmGKB, The Human Metabolome Database, Selleck Chemicals, Mcule) (BMB WP4), PubChem, Pubmed, Cellosaurus, canSAR, EC, Intact, Interpro, Pfam, PharmGKB, Timbal, Wikipedia	BAO, BIBO, ChEMBL-EBI, CHEMINF, CLO, UO, QUDT, SIO, NCBI Taxonomy, CITO, GO
Gene expression data	Expression Atlas	Under what experimental conditions is Ensembl gene ENSG00000129991 (TNNI3) expressed?	Biosamples DB (BMB WP4)	EFO

Datatype	Database	Example query	Data integration	Ontologies
Molecular interactions and pathways	Reactome	Pathways that references Insulin (P01308)	AraCyc, BioGPS, BioModels, Brenda, CAS, COMPOUND, COSMIC, CTD Gene, ChEMBL-EBI, DOCK Blaster, DOID, EC, EMBL, ENSEMBL, Bacterial Ensembl, Flybase, GO, GeneCards, GeneDB, HapMap, IntEnz, KEGG Gene, MOD, NCBI Gene, NCBI Nucleotide, NCBI_Protein, OMIM, ORCID, Orphanet, PRF, PlasmoDB, Protein Data Bank, PubChem Compound, PubChem Substance, RefSeq, Rhea, SGD, SO, TAIR, UCSC human, UniProt, Wormbase, dbSNP Gene, dictyBase, miRBase	DO, EFO, OMIM
Protein sequences	UniProt	What are the preferred gene name and disease annotations of all human UniProt entries that are known to be involved in a disease?	148 external databases including Reactome, Ensembl, Chembl; for details see http://www.uniprot.org/docs/dbxref	UniProt Core Vocabulary, BIBO, ECO, Dublin Core
Experimental data on protein production	PiMS	How many constructs have been designed for gene NOD2?	Planned integration with Uniprot	PROV-O, Dublin Core, PiMS, Uniprot

Datatype	Database	Example query	Data integration	Ontologies
Genomic data	e!	Get all mouse genes on chromosome 11 between location 101,100,523 and 101,190,725 forward strand	Reactome, HGNC, UCSC, the Database of Aberrant 3prime splice sites, DBASS3, UniProtKB, CCDS, EntrezGene, NCBI RefSeq	SIO, FALDO
Gene variation data	e!	Which disease pathways could potentially be implicated and targeted for a given patient's SNPs?	Reactome, HGNC, UCSC, the Database of Aberrant 3prime splice sites, DBASS3, UniProtKB, CCDS, EntrezGene, NCBI RefSeq	SIO, FALDO, OMIM

Table 3 Technical details of RDF implementation

Datatype	Database	RDF-ization approach	General tools and resources used for conversion	Number of resultant RDF triples[15]	Planned approach to outstanding challenges	RDF-ization supported in part by	Infra-structure
Biological models	BioModels	RDB conversion	SBML	11,385,558	N/A	BBSRC	ELIXIR
Biological samples	BioSamples	Flatfile conversion and RDB conversion	Java2RDF, ZOOMA	102,520,402	Parallelization and incremental updates rather than bulk	BMB WP 3.4 (ZOOMA), BMB WP4	ELIXIR
Drugs and other chemicals	ChEMBL	RDB conversion	Sesame OpenRDF, Groovy	374,762,364	Convert ontology annotations from text to abstract identifiers	Open PHACTS	ELIXIR
Gene	Expression Atlas	Flatfile conversion	OWL API,	447,149,547	N/A	BMB WP 3.4	ELIXIR

[15] December 2014

Datatype	Database	RDF-ization approach	General tools and resources used for conversion	Number of resultant RDF triples[15]	Planned approach to outstanding challenges	RDF-ization supported in part by	Infrastructure
expression data		and RDB conversion	ZOOMA			(ZOOMA)	
Molecular interactions and pathways	Reactome	RDB conversion	OWL API, BioPax	12,487,422	N/A	Open PHACTS	ELIXIR
Protein sequences	UniProt	RDB conversion	Sesame OpenRDF	9,024,662,088	Exploration of new hardware/software solutions	Swiss Federal Government	ELIXIR
Experimental data on protein production	PiMS	RDF-on-demand	Apache Jena API	Varies according to query	Rules requiring a higher level of inference are applied in a second step	BMB WP4	INSTRUCT
Genomic data	e!	RDB to RDF	Ensembl Perl API to N triples	316,381,125	Currently 21 species loaded. Scalability concerns if we load all Ensembl genomes	BMB WP4	ELIXIR
Gene variation data	e!	RDB to RDF	Ensembl Perl API to N triples	Human estimated to generate 80 million triples	Extracting data from Ensembl DB very slow, currently > 2 weeks to run script	BMB WP4	ELIXIR

3.2.2 Biological Model Data

The starting data model for the BioModels Database was the Systems Biology Markup Language (SBML) model specification[16]. A converter was written to convert SBML/XML models to SBML/RDF triples[17]. Although it was time-consuming to create an RDF schema to cover all SBML concepts, storing and querying the data was straightforward. Currently, scaling is not an issue because BioModels RDF only consists of curated, non-curated, and whole genome models (~3800). However, we are yet to generate RDF for metabolic

[16] http://sbml.org/

[17] https://github.com/sarala/ricordo-rdfconverter

and non metabolic models (~140,000) which may be more cumbersome. Scaleability issues are not anticipated in the near future.

3.2.3 Biological Sample Data

The BioSamples database exists to link disparately published datasets produced using the same biological sample. The BioSD linked data[18] are modelled after the structure of SampleTab[19]. As a result of collaboration with other members of the RDF platform team, the BioSD linked data are well integrated with other EMBL-EBI linked data sources, such as Gene Expression Atlas and ChEMBL. For instance, a search about gene expression in the Atlas can lead to experiments studying similar experimental factors with technologies other than microarrays, thanks to the links from the Atlas data to the ArrayExpress experiments (reported by the Atlas data set) and from these to other types of experiments and biomedical samples (reported by the BioSD data set). A portion of the published datamodel[20] is shown in Figure 4 using example data.

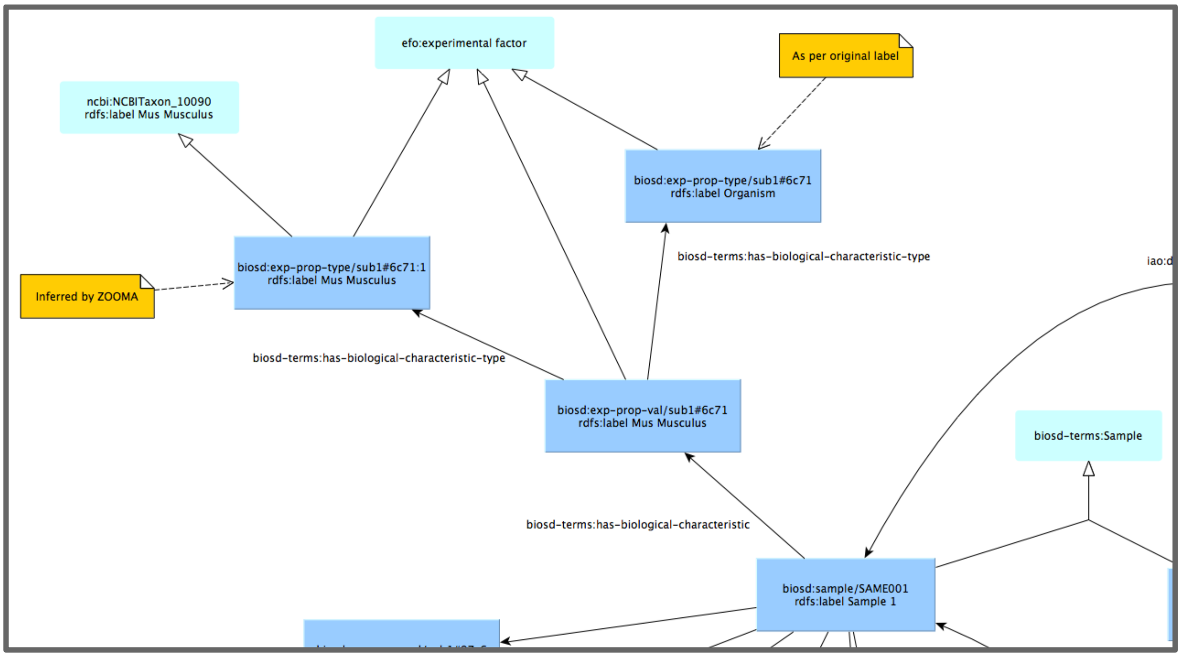

Figure 4 A portion of the BioSD data model using example data

[18] http://tinyurl.com/pns5dxs
[19] http://tinyurl.com/pbvyylx
[20] https://raw.githubusercontent.com/wiki/EBIBioSamples/biosd2rdf/biosd_samples_20140317.png

It also can be used with ontology-aware search engines, by referring ontologies such as the Experimental Factor Ontology (EFO), Biomedical Investigation Ontology (OBI), NCBI Taxonomy and the ontology for Chemical Entities of Biological Interest (ChEMBL-EBI). For instance, this makes it possible to search BioSamples DB for the term 'atrial fibrillation' with automatic query expansion to all synonyms and child terms[21].

We then developed a JavaBeans-based conversion tool to transform the original (BioSD OracleDB) data to RDF[22]. This BioSD-to-RDF converter is built on top of java2rdf[23], a generic Java library that we developed to declare and apply mappings between JavaBeans and OWL/RDF structures[24].

In addition to the basic principles of object-to-RDF mapping, our converter relies on ZOOMA[25], a tool which uses text-mining techniques and manual annotations from existing bio-repositories, to optimally match free-text annotations with existing ontologies a component comprising part of D3.4). ZOOMA also requires significant computational resources. The conversion process is parallelised using the EMBL-EBI cluster system (which currently have about 500 nodes and is based on the LSF software) and each node is further parallelised through multi-threading; even so, this process currently still takes about two days to complete. Because ZOOMA is a bottleneck in delivery of BioSamples RDF we are developing an 'incremental annotation' approach where new data can be identified and ZOOMA-annotated, rather than the whole database, and annotation results stored in Oracle, together with the primary source of data, rather than being fully generated at each update of linked data. This will speed-up the generation of RDF files, and will make available these ZOOMA annotations to all BioSD users, including the users of the BioSD web interface, and not just Linked Data users. BioSD data volumes are is growing; achieving this performance optimisation will therefore be crucial to our RDF scaleability. Indeed, based on the experience done in the

[21] http://www.ebi.ac.uk/biosamples/browse_samples.html?keywords=
[22] https://github.com/EBIBioSamples/biosd2rdf
[23] http://www.slideshare.net/mbrandizi/java2rdf
[24] https://github.com/EBIBioSamples/java2rdf
[25] http://www.ebi.ac.uk/fgpt/zooma/docs/

past months, we would advise this incremental annotation approach for anyone starting out. See the section on provenance for more details.

3.3.4 Chemical Entities and Drug Discovery Data

The generation of RDF for chemical entities was funded by the IMI OpenPHACTS project. The data modeling process was straightforward as the relational database schema clearly defined groups of tables, which could be used later to define the classes used in the RDF data model. Upon defining a basic ontology, known as the ChEMBL Core Ontology[26], we then 'populated' the RDF data model by generating triples using Groovy[27] scripts. The process of creating the ChEMBL RDF triples used SQL to select the RDF class-specific data from the database, which was then serialised to turtle format using the Java Sesame OpenRDF[28] libraries. This process is now part of the ChEMBL release cycle, which results in the ChEMBL RDF being made available to download from the ChEMBL FTP site as well as being loaded into the EMBL-EBI RDF Platform infrastructure.

New data relevant to drug discovery research is often added to the ChEMBL relational model, which we also aim to include in future ChEMBL RDF releases. The current RDF generation process allows us to easily introduce these data enhancements to the RDF data model. One future improvement we would like to make to the existing RDF model is to replace the current ChEMBL Core Ontology RDF class and predicate names e.g. `'Assay'`, with abstract identifiers, `e.g. CCO_000001`. This would allow us improve existing descriptions and definitions, without impacting on users' existing SPARQL queries.

3.3.5 Expression Data

There has been some prior work in representing gene expression data in RDF. Ontologies such as EFO[29], SIO[30] and OBI[31] already provide terms for

[26] http://www.ebi.ac.uk/rdf/documentation/chembl
[27] http://groovy.codehaus.org/
[28] http://rdf4j.org/
[29] http://www.ebi.ac.uk/efo/
[30] http://purl.bioontology.org/ontology/SIO
[31] http://obi-ontology.org/

describing experiments and experiment variables that are required to describe the context of a gene expression result. A paper by Deus et al. (2012) on “Translating standards into practice – One Semantic Web API for Gene Expression”[32] also provided some guidelines for this work. Although useful, these guidelines make some assumptions about what kind of data is available. The current Gene Expression Atlas (GXA)[33] data model[34] captures the core components of the atlas database, e.g.:

- Basic experimental meta data
- A description of the assays and samples
- A collection of derived differential expression values mapped to genes and proteins.

We developed a Java-based RDF converter that first collected data (from the Atlas REST API and CSV download) and then converted this merged data into an OWL model using the OWL API[35].

We spent a lot of time designing the model and engaged the external community to get some shared agreement. Although there was some interest and suggestions from external parties, no other groups were generating this kind of data in RDF. Our approach was to define a schema for the data in GXA using an OWL ontology[36]. To align this schema with reference ontologies that describe biological assays, we provide an additional ontology that describes equivalences across these ontologies[37]. For example, the GXA ontology has the concept of “Experiment” that is equivalent to the description of “Experiment” in the Semantic-Science Integration Ontology (SIO) and the Ontology of Biomedical Investigations (OBI).

3.3.6 Reaction Data

The BioPAX version of the Reactome datamodel was the starting point for RDF conversion. Although Reactome BioPAX is already in OWL/RDF it had to

[32] Deus et al. (2012), doi:10.1016/j.jbi.2012.03.002
[33] http://www.ebi.ac.uk/gxa/
[34] http://www.ebi.ac.uk/rdf/documentation/atlas
[35] http://owlapi.sourceforge.net

[36] available at http://rdf.ebi.ac.uk/terms/atlas
[37] http://rdf.ebi.ac.uk/terms/atlas-mapping

be improved to be linked-data friendly. The small size of the Reactome database made it easier to work with; however, querying is difficult due to the complexities of BioPAX: BioPAX-driven RDF produces inconsistent results depending on how the SPARQL query is structured. Consequently, for user-friendliness, it would have been better to generate RDF directly from the content in the Reactome database without relying on BioPAX. If users indicate that the SPARQL inconsistency is a barrier to their use of the Reactome RDF, the datamodel will need to be revisited. However, based on our experience, we would probably recommend that any new databases explore RDF production mechanisms that are not BioPax-driven. Figure 5 shows a portion of the datamodel for Reactome.

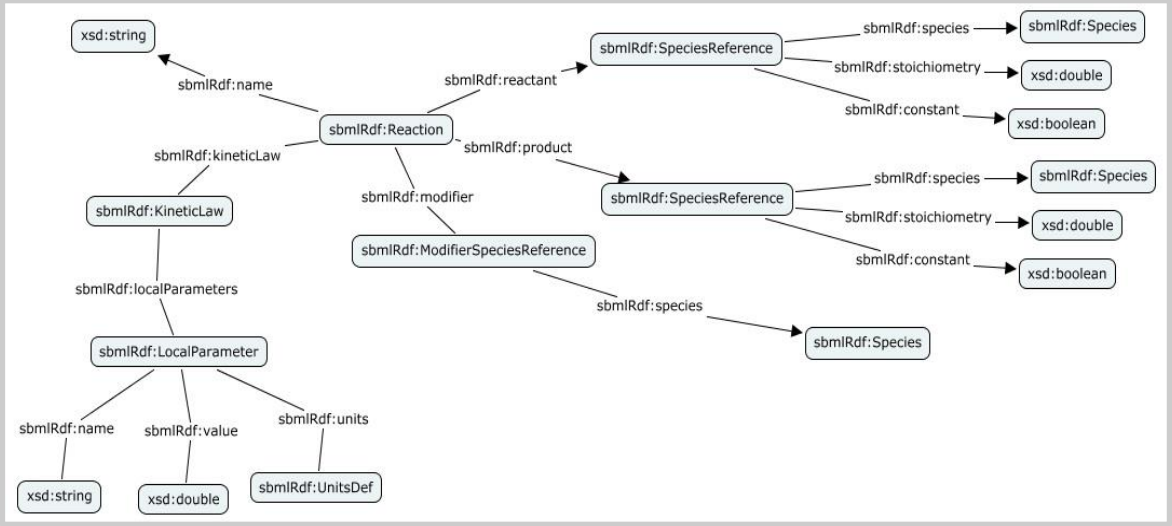

Figure 5 Data model for the 'reaction' portion of the Reactome database

3.3.7 Protein sequence data

The starting data model for UniProt was the internal SWISS-PROT data model (java classes). UniProt was an early adopter of RDF (conversion originally started in 2004), and some of the initial challenges encountered were due to the immaturity of SemWeb technologies and inexperience of both developers and users. One of the ongoing challenges is that UniProt represents over 30 years of protein data and the UniProt data model has evolved over time. Although the RDF technology is flexible enough to capture this data model heterogeneity, it can be cumbersome to generate and query the full complexity in RDF; therefore some compromises have been made for simplicity.

3.3.8 Genomic data

The Ensembl RDF is generated from the Ensembl core database. This dataset has been in high demand by many EMBL-EBI users. In particular projects like OpenPHACTS are interested in the Ensembl database cross references, and the Database Centre For Life Sciences in Japan are interested in genomic coordinates for genomic features. This dataset includes location information for all genes, transcripts and exons for a given genome. It also includes ortholog information for all genes. An overview of the current Ensembl core schema is shown in Figure 6. The Ensembl database cross references to over 100 external databases are available in RDF and use URIs from identifiers.org to identity entities. Identifiers.org URIs are being adopted by many other external RDF efforts in the life sciences, and this Ensembl cross reference data provides an important set of bridging links for integrating Bio RDF datasets.

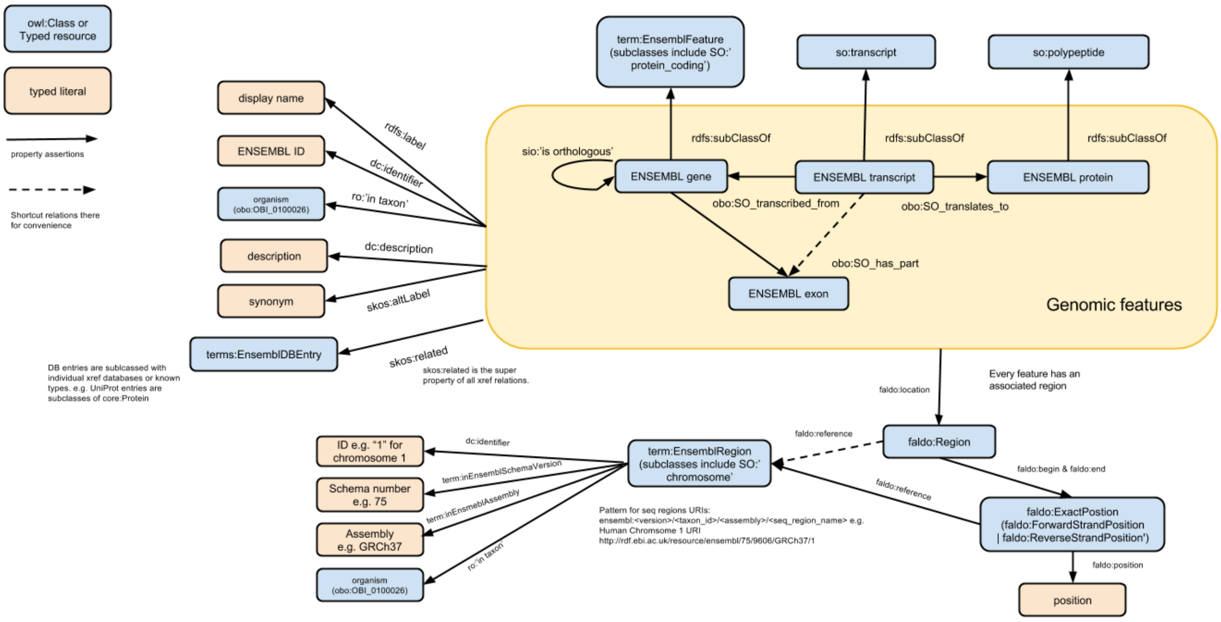
owl:Class or Typed resource
typed literal
property assertions
Shortcut relations there for convenience
display name
ENSEMBL ID
organism (obo:OBI_0100026)
description
synonym
terms:EnsemblDBEntry
DB entries are sublcassed with individual xref databases or known types. e.g. UniProt entries are subclasses of core:Protein
rdfs:label
dc:identifier
ro:'in taxon'
dc:description
skos:altLabel
skos:related
skos:related is the super property of all xref relations.
term:EnsemblFeature (subclasses include SO:' protein_coding')
so:transcript
so:polypeptide
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
sio:'is orthologous'
ENSEMBL gene
ENSEMBL transcript
ENSEMBL protein
obo:SO_transcribed_from
obo:SO_translates_to
obo:SO_has_part
ENSEMBL exon
Genomic features
faldo:location
Every feature has an associated region
faldo:reference
faldo:Region
faldo:begin & faldo:end
faldo:reference
faldo:ExactPostion (faldo:ForwardStrandPosition | faldo:ReverseStrandPosition')
faldo:position
position
term:EnsemblRegion (subclasses include SO:' chromosome'
dc:identifier
term:inEnsemblSchemaVersion
term:inEnsmeblAssembly
ro:'in taxon'
ID e.g. "1" for chromosome 1
Schema number e.g. 75
Assembly e.g. GRCh37
organism (obo:OBI_0100026)
Pattern for seq regions URIs:
ensembl:<version>/<taxon_id>/<assembly>/<seq_region_name> e.g.
Human Chromsome 1 URI
http://rdf.ebi.ac.uk/resource/ensembl/75/9606/GRCh37/1

Figure 6 Core Ensembl RDF schema

Ensembl core is currently generated using Perl scripts written against the Ensembl API[38]. These scripts can be used to extract RDF for any species in Ensembl or Ensembl Genomes. There are currently around 80 species in the primary Ensembl vertebrate database, and many more species are available

[38] available on GitHub here https://github.com/simonjupp/ensembl-rdf

in the Ensembl Genomes database for plants, metazoa and fungi. We are currently generating RDF for a subset of these species based on popular species and specific requests from users. As of the writing of this report, 22 species have been converted into RDF and the data is made available via a public SPARQL endpoint[39] for demonstration purposes. We have yet to decide whether we will pre-build RDF indexes for all species and make these available via SPARQL. This would require additional hardware on the existing RDF platform to deal with the magnitude of data that we predict would be in the 10s of billions of triples.

3.3.9 Gene variation data

We recently began to work on representing the Ensembl variation data in RDF. The approach is similar to the one undertaken for the core data where the Ensembl Perl API is used to generate RDF triples from the variation database. The schema outlined in Figure 7 shows specific sequence alteration, such as a SNP from the dbSNP database are connected through the allele variant to a given transcript[40]. The FALDO ontology is used to describe the region in the genome where the sequence variation occurs.

A significant challenge for the variation is dealing with the amount of data in the current database. The human variation dataset contains over 80 million variants alone. With each variant on average generating around 25 triples, we estimate that the complete dataset could contain up to 20 billion triples. While in principle this is well within the limit of Virtuoso, we would again need more hardware and infrastructure to be able to serve this data up via a public SPARQL endpoint. We are currently trying to generate the full human set, but have had some problems extracting that much data from the database. The current script takes over 2 weeks to run and often fails due to connection dropouts. Some work is needed to optimise the script to decrease the generation time. The SPARQL endpoint (BETA)[41] includes variation data for the BRCA2 gene for demonstration purposes.

[39] http://wwwdev.ebi.ac.uk/rdf/services/ensembl/sparql

[40] detailed version online at http://tinyurl.com/rdfdatamodelensemblvariation

[41] available at http://wwwdev.ebi.ac.uk/rdf/services/ensembl/sparql

The variation data provide us with real challenges for generating RDF for certain large resources. In doing this work we have begun to explore alternate approaches to dealing with large datasets such as variation. At a recent biohackathon event we developed an on-the-fly RDF generator for querying variation data directly from Variant Call Format (VCF) files. The on-the-fly approach allows us to keep the variation data in optimised indexes for VCF, such as Tabix. The application accepts SPARQL queries and, in real time, translates the SPARQL query into a query over the native index. Such approaches remove the need to materialise all of the RDF up front, and early results show that this may be a promising approach for this kind of data.

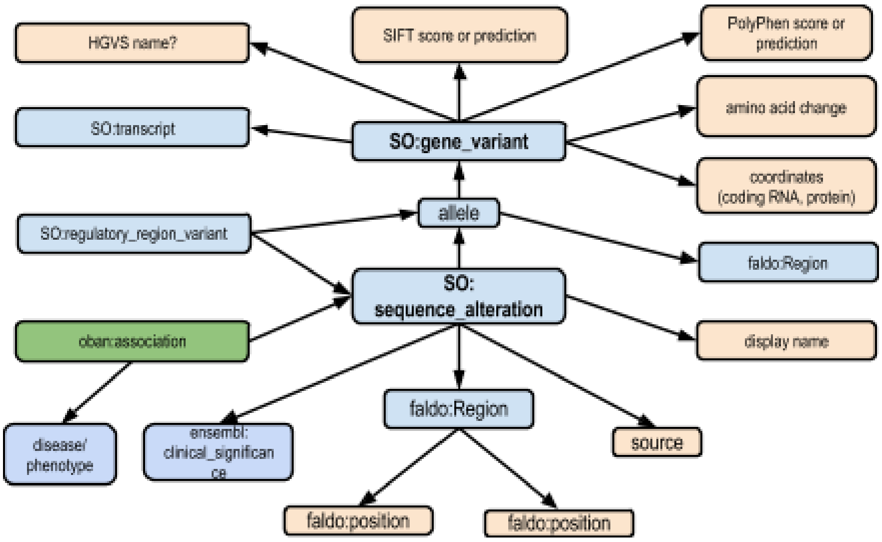

Figure 7 Ensembl variation datamodel (relationship names omitted for simplicity)

3.4 Best practices for representing data in RDF

In close coordination with BioMedBridges WP3 and in accordance with WP4 objective 1, we are developing guidance relevant to knowledge representation in RDF (Table 7).

Table 4 Summary of best practices for representing data in RDF

Del	Appendix	Title	Direct links
3.1	Appendix A	Online dictionary of molecular identifiers	http://tinyurl.com/identifiersdictionarysource (raw source) http://tinyurl.com/edamidentifiersbranch (destination ontology) www.identifiers.org (interface)
	Appendix B	Identifiers best practice	http://tinyurl.com/identifiersbestpractice
	Appendix D	Identifier Resolution and Conversion Tools	http://tinyurl.com/identifiertools (source) http://wwwdev.Ebi.ac.uk/fgpt/toolsui/ (interface)
4.7	Appendix 2	10 simple rules for linked data provision	https://github.com/dbcls/bh14/wiki/Ten-simple-rules-for-publishing-Linked-Data-for-the-Life-Sciences
		Provenance for linked datasets	http://tinyurl.com/linkeddataprovenance

3.4.1 Lessons learned with knowledge representation

There are still open challenges relevant to semantic web integration (Table 5); our guidance is specific to the biomedical domain, though the challenges themselves are often broader than that.

Table 5 Open challenges relevant to semantic web integration

Knowledge representation challenge	Guidance and lessons learned
SPARQL is difficult and requires specialist knowledge	Support more training in SemWeb technologies. Develop GUI interfaces that hide the complexity of SPARQL

Knowledge representation challenge	Guidance and lessons learned
	without sacrificing the power.
Current GUI interfaces for graph-based data are still immature.	Practice more user-centered design and perform more user-experience testing.
The flexibility of RDF is powerful, but can be easy to overdo; modelling is often approached with more granularity than users require, thereby causing significant and ongoing delays in the release of an RDF dataset.	Focus first on the most relevant parts of the data, driven by use cases, and release a simplified RDF dataset. The model can then be extended, if necessary, based on user feedback.
Even basic biological concepts like genes and gene variations can be modelled in so many different ways, that it can be time consuming to come to a decision.	Study existing biological datasets and where possible model basic concepts in a similar way, reusing existing ontologies, rather than devoting a lot of time exploring the universe of possibilities.
Some kinds of data are less well suited to representation in RDF (eg high-dimensional data) and things for which there is no semantic context (numbers).	Consider carefully whether your users really require high-dimensional data to be represented in RDF, and if so, how they would interact with it. Consider an RDF-on-demand approach.
It can be non-trivial to incorporate RDF into business process such that data updates do not break legacy sparql queries and corresponding downstream apps.	We currently store legacy RDF data in dumps that can be downloaded, however this approach can break sparql queries performed at the endpoint. In this case, users must load the RDF data locally and query from their own triplestore. In retrospect, the solution we would advise is to version sparql endpoints or to load data into versioned graphs within the sparql endpoint.

Knowledge representation challenge	Guidance and lessons learned
There are four general modelling patterns for provenance: — Singleton property-based (unique RDF predicate for every triple) — Graph-based (e.g. nanopublications) — OWL/axiom-based (e.g. SIO) — Reification[42] The challenge is that there is a dearth of documentation and consensus about which models to use.	Guidance is being developed regarding which of these models to use in which contexts.[43]

3.5 Technical challenges with storage and performance

3.5.1 Existing triplestore solutions

A triplestore is a purpose-built database for the storage and retrieval of triples (data entities composed of subject-predicate-object). Some triplestores are built from scratch to hold the triples natively, others are built on top of existing relational databases (e.g. SQL). While the latter reduces overall programming effort, the former typically has the advantage of better performance for a given hardware specification.

3.5.2 Benchmarking

We explored various triplestores for performance using the early versions of the RDF databases produced as described above[44] (Table 6).

[42] http://en.wikipedia.org/wiki/Reification_%28computer_science%29#RDF_and_OWL

[43] https://github.com/dbcls/bh14/wiki/Standardization-of-RDF-data-and-development-of-tools-ontologies

[44] Though now part of the RDF platform, BioSamples DB was not part of the benchmarking analysis as it was unavailable when benchmarking was carried out

Table 6 Triplestores examined during benchmarking

Entity	Dataset	Number of triples March 2014
Proteins	UniProt	6,419,569,246
Gene expression experiments	Atlas	339,991,787
Chemical compounds	ChEMBL	97,717,782
Reaction and pathways	Reactome	12,927,678
Biological models	BioModels	5,788,106

3.5.3 Technical challenges

Since the public release of EMBL-EBI's RDF Platform in October 2013, a number of technical issues with the software components have been identified.

The Virtuoso processes regularly (e.g. twice weekly) crash without any forewarning or log events. Whilst the redundant architecture of the platform means the service remains operational, some disruption occurs during query execution when these events occur. Optimisation of the configuration of the databases has not been successful in solving the issue, although the implementation of additional monitoring has revealed that the cause of the crash is a rapid increase in the memory footprint of an instance resulting in the VM exceeding its memory allocation. It is unclear yet whether this can be entirely explained merely by growth in dataset sizes and growth in demand, or whether there is a defect in the Virtuoso software.

Defects have also been identified in Virtuoso's query execution engine, these can produce incorrect answers in response to certain SPARQL queries. Some of these have been fixed by the vendor after being reported, but others remain outstanding. We use the open source edition and therefore do not have a support contract. Whilst these defects can be mitigated to some extent by ourselves (e.g. we are able to disable some functionalities related to *reasoning*), they do affect some external users.

The Virtuoso infrastructure (and platform as a whole) are designed around datasets that are updated periodically and not continuously. This allows a relatively simple data migration and deployment mechanism to be used and does not require a live replication feature that is commonly reserved for commercially licensed products. The downside is that if resources such as Europe PubMedCentral[45] wished to use the platform, it might be necessary to further develop the infrastructure to support their more frequent update cycles.

The infrastructure has coped well with very large bursts in demand (just one dataset served ~46 million requests in one month on only two instances) and would appear to scale very efficiently. It is difficult to ensure the performance of one user's queries is not impacted by the queries of others. Any public web service faces similar issues, but the effect is magnified exponentially for SPARQL services because blocked queries can quickly build up if queries take several minutes to execute.

3.5.4 Technical solutions

Managing the potential for small numbers of users to impact performance is a challenge. EMBL-EBI operates a 10 minute timeout which mitigates the impact of poorly designed or expensive queries. Other limits (e.g. certain types of query) could be imposed to maintain performance. The downside of imposing limits is of course that some queries cannot be performed. Thus the endpoints are not suitable for downstream users to integrate into certain workflows, such as for example intensive data mining. Such users would need to download the RDF in bulk and operate their own triplestores. Managing expectations is key; however, future work might explore the possibilities of making "ready-built" triplestore databases or virtual machines available so that developers can easily run their own instance. Another potential solution would be to recognise challenging queries up front and put them in a separate queue; this would guarantee response times for simple queries.

[45] http://europepmc.org/

Given the current usage levels and planned release of new RDF datasets onto the platform[46], the platform requires an upgrade merely to sustain the current service. The quickest method is to increase the number of Virtuoso instances. This will either alleviate the downtime issue or it could reveal the existence of a software bug. Over time, the number and memory allocation of VMs will scale with the size of the data on the platform.

If the other query execution issues are to be resolved, upgrading the resources available to the platform is insufficient. One option could be to purchase the commercial edition of Virtuoso on condition that the outstanding issues are fixed. This would also grant us access to the data replication features of Virtuoso, which would allow us to potentially support datasets with incremental update architectures such as EuropePMC.

Alternatively, a different triplestore product may be considered. Since the initial technology evaluation was planned in 2012, the market has expanded and products have been improved. Triplestores differ in their management processes, systems requirements and performance criteria. Thus a switch is not a trivial exercise and the selection of a new product requires a new evaluation.

3.5.5 Summary of technical lessons learned

Table 7 Technical lessons learned

Technical challenge	Generalisable guidance
Triplestores have a higher failure rate than relational databases and determining cause of failure can be complex.	Build monitoring scripts and memory allocation surveillance. Use load balancing and redundancy.
Supporting backwards compatibility can be disk-intensive.	Make disk allowance for organic growth of the data as well as for backups.

[46] Ensembl and Literature text mining and updated baselines Gene Expression Atlas

Exposing a SPARQL endpoint to the public Internet is analogous to exposing an SQL database. In both cases it is difficult to maintain levels of service given the unknowable and diverse range of queries.	Impose limits and manage expectations; Explore integrating with cloud infrastructures to allow users to run their own instance.
Triplestore vendors typically have open-source and commercial versions. Functionality of the open-source versions is more limited, less robust, and not well supported relative to their paid license counterparts, some of which are cost-prohibitive in academic settings. Evaluating the offerings, licenses, and vendors has therefore been difficult, especially because of the fast pace at which new solutions are introduced to the market.	Re-evaluate technical decisions on a yearly basis (See also BioMedBridges WP11 Technology Watch).

3.6 Usage statistics

The RDF platform is now a dependency for fifteen ELIXIR applications and eleven external dependent groups, including several industry partners (see section 3.7 below). There are around 55 million hits to the site. The platform executes 99% of queries in less than one second. The platform has served over 115 million queries in 2014, though a precise number of unique queries is difficult to obtain since a very large fraction arise from federation of queries across multiple endpoints. Full usage data statistics are being compiled now for inclusion in the BioMedBridges periodic report.

3.7 User community and trends

Members of the EMBL-EBI Industry Programme (Eli Lilly, UCB and Syngenta) have recently committed to Linked Data strategies for their global integrated data operations. The scale of investments is large and may fuel a coalescence behind these technologies as suppliers seek to align to client requirements. Other large global corporations such as Novartis, Novo Nordisk, AstraZeneca and GlaxoSmithKline are also making use of the technology. New European project proposals such as Ruminomics and DIOGENES will require further development of resources in this area. Furthermore, RDF and Linked Data are focal points of the recommendations by The Data FAIRport[47] initiative for data interoperability and re-use.

3.8 Conclusion and future work

We have collaborated on the generation of seven RDF datasets, benchmarked three RDF triplestores. We have found the technology to be more immature relative to RDBs but much more capable for big data integration tasks. For instance, the RDF platform is a key dependency in the strategic plan for the Centre for Therapeutic Target Validation (CTTV).[48]

Development and adoption of semantic technologies is accelerating. The availability of loosely-coupled production-grade RDF data is expected to further reinforce this trend. Quality documentation and SemWeb training materials are increasingly easy to find, thereby lowering barriers to entry for programmers and database managers. Together, these factors support the further application of semantic web technologies to integrating research data.

Building on the lessons learned in Phase I, we will continue to address the issues we have encountered regarding scalability and performance. To this end, we will continue to monitor triplestore advances; some vendors such as Cray[49] offer hardware targeted at graph-like data structures similar to RDF. In Autumn 2014, Oracle reported that their proprietary hardware-software combination has been proven to handle one trillion triples[50].

We will also continue to monitor usage and will compile metrics to help inform the next wave of technical decisions and resources.

[47] http://www.datafairport.org/

[48] http://www.targetvalidation.org/

[49] http://www.cray.com/products/analytics/

[50] http://download.oracle.com/otndocs/tech/semantic_web/pdf/OracleSpatialGraph_RDFgraph_1_trillion_Benchmark.pdf

4 Delivery and schedule

The delivery is delayed: ☐ Yes ☑ No

5 Adjustments made

No adjustments were made to the deliverable.

6 Background information

This deliverable relates to WP 4; background information on this WP as originally indicated in the description of work (DoW) is included below.

WP 4 Title: Technical Integration
Lead: Ewan Birney (EMBL)
Participants: EMBL, STFC, UDUS, FVB, TUM-MED, ErasmusMC, HMGU, VU-VUMC

In work package 4 we will implement a federated access system to the diverse data sources in BioMedBridges. This will focus on providing access to data or metadata items which utilise the standards outlined in WP 3. Experience across the BioMedBridges partners is that executing a federated access system, in particular a federated query system, is complex for both technological and social reasons. Therefore we will be using an escalating alignment/engagement strategy where we focus on technically easier and semantically poorer integration at first and then progressively increase the sophistication of the services. In each iteration, we will be using biological use cases which are aligned to the capabilities of the proposed service, thus providing progressive sophistication to the suite of federated services.

Our first iteration involves using established REST based technology to provide userbrowsable visual integration of information. This will be useful for both summaries of data rich resources (such as Elixir) and summaries of ethically restricted datasets where only certain meta-data items are public (such as BBMRI, ECRIN and EATRIS). We will then progress towards lightweight distributed document and query lookups, where the access for ethically restricted data will incorporate the results of WP 5. Finally at the outset of the project we will explore exposure of in particular meta-data sets via RDF compatible technology, such as SPARQL, and the presence of the technology watch WP 11 will provide recommendations for other emerging technologies to use, aiming for the semantically richest integration.

Work package number	WP4	**Start date or starting event:**	month 1
Work package title	Technical Integration		
Activity Type	RTD		

Participant number	1:EMBL	4:STFC	5:UDUS	6:FVB	7:TUM-MED	9:ErasmusMC	11:HMGU	13:VUMC
Person-months per participant	69	40	38	0	37	15	32	37

Objectives

1. Implement shared standards from WP 3 to allow for integration across the BioMedBridges project
2. Expose the integration via use of REST based WebServices interfaces optimised for browsing information
3. Expose the integration via use of REST based WebServices interfaces optimised for programmatic access
4. Expose appropriate meta-data information via use of Semantic Web Technologies
5. Pilot the use of semantic web technologies in high-data scale biological environments.

Description of work and role of participants

We will provide a layered, distributed integration of BioMedBridges data using latest technologies. A key aspect to this integration will be the internal use of standards, developed in WP 3 which will provide the points of integration between the different data sources. The use of common sample ontologies (WP 3) will provide integration between biological sample properties, such as cell types, tissues and disease status, in particular bridging the Euro-BioImaging, BBMRI, Elixir and Infrafrontier projects. The use of Phenotype based ontologies will provide individual and animal level characterisation which, when these can be associated with genetic variation, will provide common genotype to phenotypic links, and this will be used to bridge the ECRIN, EATRIS, INSTRUCT, BBMRI, Infrafrontier and Elixir Projects. The use of environmental sample descriptions and geolocation tags will bridge between EMBRC, ECRIN, ERINHA, EATRIS and Elixir. The use of chemical ontologies will help bridge between EU-OPENSCREEN, ECRIN, Euro-BioImaging, INSTRUCT and Elixir. By applying these standards in the member databases (themselves often internally federated) we will create a data landscape that theoretically can be traversed, data-mined and exploited. To expose this data landscape for easy use, we will deploy a variety of different distributed integration technologies; these technologies are organised in a hierarchy where the lowest levels are the semantically poorest, but easiest to implement, whereas the highest levels potentially expose all information in databases which are both permitted for integration (some are restricted for ethical reasons, see WP 5) and can be described using common standards. We will develop software with aspects appropriate for the distributed nature of this project taken from agile engineering practices, such as rapid iterations between use cases and partial implementation. In particular we will be using the enablement/alignment strategy (Krcmar H., Informationsmanagement, Springer) to ensure that the use cases that drive the project are aligned to feasible capabilities that can be delivered. The work package will be implemented in a collaborative manner across the BMSs, with frequent physical movement of individuals.

The proposed technologies are:

1. REST-based “vignette” integration, allowing presentation of information from specific databases in a human readable form. An example is shown in Figure 1. These resources allow other web sites to “embed” live data links with key information into other websites. This infrastructure would then be used to provide browsers that, on demand, bridge between the different BioMedBridges groups – for example, information which can be organised around a gene or a chemical compound would be presented across the BioMedBridges project.

2. Web service based “query” integration, where simple object queries across distributed information resources can be used to explore a set of linked objects using the dictionaries and ontologies present. Each request will return a structured XML document.

3. Scaleable semantic web based technology. We are confident that semantic based technology can work for the rich but low data volume meta data (eg, sample information) which we will expose using semantic web technologies such as RDF and SPARQL. However, it is unclear whether this scales to the very large number of data items or numerical terms in the BioMedBridges databases (such as SNP sets or numerical results from Clinical trials) We will pilot a number of semantic web based integration of datasets, using RDF based structuring of datasets In the latter phases of the project we will look to align these solutions to other broader standards in the eScience community, taking input from the Technology Watch (WP11) group; we hope in many cases our technology choice which has been already informed by alignment to future eScience technology (e.g. RDF/SPARQL) so this may only require appropriate registration/publication of our resources. Where unforeseen but useful technologies are developed we will build systematic connections from these BioMedBridges federation technologies to other federation technologies.

Appendix 1: Details of RDF platform components

The ELXIR RDF Platform (http://www.Ebi.ac.uk/rdf/platform) comprises four technical components:

- RDF Resolver - a simple Apache VM
- RDF Website - a Drupal site
- FTP site
- SPARQL endpoint/RDF browser webapps
- Triplestore databases

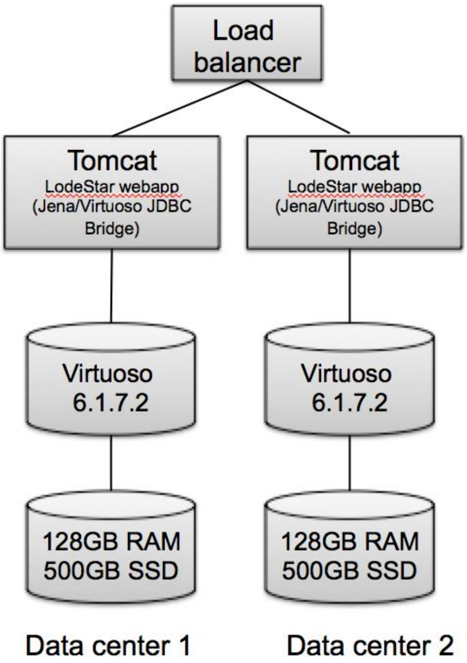

Figure 8 Diagram of RDF triplestore back-end

1 SPARQL Endpoint

The RDF Platform has five public SPARQL endpoints, each representing a different dataset from a different resource. The SPARQL endpoint is an interface supporting both interactive and programmatic access to the RDF data through SPARQL queries over HTTP connections. This complements the ability for users to download entire datasets by providing a way for them to experiment with the data or run small ad-hoc queries without needing their own SPARQL endpoint. It contains an interface for users to interactively explore the RDF data at a very basic level (literally a user-friendly formatting of the raw triples), a user interface for submitting SPARQL queries & visualising the results, and a SPARQL API (i.e. users can send SPARQL queries directly over HTTP). The endpoint is implemented as a Java web application called *LodeStar* that can operate on top of any RDF database (triplestore). The code is developed by SPOT, is open source, and aside from the EMBL-EBI RDF Platform has also been implemented by a small number of third parties. At EMBL-EBI, all the endpoints run inside a single Tomcat instance provided by Web Production, and is backed by our triplestore (see below).

2 Triplestore

A triplestore is a database for RDF, analogous to a relational database management system for tabular (relational) data. RDF data has a specific graph-like structure based on collections of *triples*, and thus databases designed to store RDF data are called *triplestores*. Most applications based on RDF assume that a triplestore will support the SPARQL query language. Thus a triplestore is a specialised kind of database - either a standalone product or a product with explicit support for these data types and functionalities. A triplestore might also be described as a subtype of *graph databases*.

The RDF Platform uses Virtuoso Open Source Edition as its triplestore software. This is run as an in-memory database on VMs provided by the Cloud & Virtualisation team (Systems Applications), and is managed by TSI. The Resource teams who own the data manage the data loading and deployment themselves through the use of automation scripts developed by TSI. Datasets are currently transitioning from version 6 to version 7 of Virtuoso, as the latter fixed some issues and also can operate as a *column store* that is more optimised for querying and is more memory efficient.

Each dataset on the platform has its own independent Virtuoso instance, accepting connections on different ports. This allows each Resource team to start and stop their instances, load data and propagate updates from development and staging VMs to the live VMs without affecting other datasets on the platform. Individually, the memory footprint of each instance varies significantly according to the size of the dataset (number of triples). However, all of the instances run on the same VM and thus consume a relatively large amount of memory.

Appendix 2: Resources

1. Best practices for representing data in RDF
    - Ten simple rules for linked data: https://github.com/dbcls/bh14/wiki/Ten-simple-rules-for-publishing-Linked-Data-for-the-Life-Sciences
    - Provenance for linked datasets: http://tinyurl.com/provenancelinkeddatasets
2. BioMedBridges RDF knowledge exchange workshop training materials: http://www.biomedbridges.eu/trainings/knowledge-exchange-workshop-resource-description-framework-rdf
3. SWAT4LS SemWeb training materials: http://goo.gl/xSkzaj
4. Triplestore benchmarking report: https://drive.g.oogle.com/file/d/0B7Tv2ysg_H9-NVJoaC1ibk9xN28/view?usp=sharing
5. Resources for RDB-to-RDF conversion from object models:
    - http://www.slideshare.net/mbrandizi/java2rdf
    - http://code.g.oogle.com/p/semanticwebpogos
    - http://code.g.oogle.com/p/jenabean
    - http://rdfbeans.sourceforge.net/
    - http://wiki.yoshtec.com/jaob