RICORDO

http://VPH-RICORDO.eu/

Researching Interoperability using Core Reference Datasets and Ontologies for the Virtual Physiological Human

Small or medium-scale focused research project (STREP)

Grant agreement number 248502 (Collaborative Project) within the European Commission FP7 Framework Programme

Deliverable D2.3: VPH Technical Infrastructure Requirements Report delivered to WP4

Task: Study and propose plan for dictionary space, annotation and technical infrastructure

Work Package: Establishing a resource interoperability plan for the VPH Toolkit Development

Due date of deliverable: 31-Mar-2010 **Actual submission date: 30-Nov-2010**
Start date of project: 1-Feb-2010 **Duration:** 24 months

Organisation name of lead contractor for this deliverable: EMBL/EBI

Author: Sarala Wimalaratne, Pierre Grenon, Robert Hoehndorf, George Gkoutos, Bernard de Bono

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

INTRODUCTION

The objective of this WP is to co-ordinate the joint development of methodologies to:

- Annotate Virtual Physiological Human data and model resources (VPHDMRRs) resources using terms in biological ontologies;
- Construct composites to represent complex biological concepts;
- Provide accessibility to such information for the VPH community through development of a software infrastructure.

VPH Technical Infrastructure Requirements deliverable task is discussed in this report. This focuses on requirements for:

- Software applications that are needed for users to annotate VPHDMRs, create composite ontology terms, and query across annotated VPHDMRs;
- Repository infrastructure that is needed to store VPHDMR annotations and ontological resources;
- Hardware infrastructure that is required to access these resources provided by the RICORDO project;
- External resources that will be needed for this project;
- Performance requirements for handling large number of VPH resources.

BACKGROUND

The Virtual Physiological Human (VPH) community deals with large collections of anatomical, physiological, and pathological data stored in computer readable format. These data can range from computational models to patient specific data in clinical databases. The representations of these VPH data and models (VPHDMR) are also heterogeneous.

Frequently the underlying biological concepts captured in VPHDMR which we refer to as meta-data are not explicitly captured. It is common practice to represent meta-data as a comment in text. Occasionally, the data is annotated with terms from biological ontologies. The lack of consistent annotation of the data makes it difficult to share these different types of data. Moreover, the biological concepts covered in the VPHDMR spans across multiple ontologies. Furthermore, there is no formal definition for integrating terms from multiple ontologies, making it difficult to correctly annotate VPHDMRs.

As part of the RICORDO project, a common standard is being introduced to support consistent structured annotation of VPHDMRs that can be processed by machines. This will promote sharing the body of knowledge contained in the different types of data and models relevant to the VPH community. A formal definition for integrating terms from multiple ontologies which facilitates the consistent communal annotation of VPHDMR is being developed. In this document, we provide an initial assessment of the infrastructure requirements to support these developments and support interoperability between VPHDMRs.

DESCRIPTION OF WORK

This deliverable presents requirements to support the development of the infrastructure in RICORDO's WP4. The present view emerges from the intention of developing an interoperability plan for the VPH. VPH resources must be annotated (WP3) on the basis of ontology terms coming from the CORDO dictionary, including composite ontology terms (WP5). These annotations need to be able to be queried so it allows users or agents to find relevant VPH resources on the basis of their queries. There are two main functionalities that need to be addressed, namely authoring and querying. There are a number of interconnections between the tasks involved in authoring annotations and querying them. In the following, we gradually present the infrastructural requirements that are needed to execute these tasks.

Main vision

The main vision comes from the multiplicity of VPH resources. For each resource, we want to be able to support its annotation. This leads to a number of requirements explained in the Authoring section. Once the annotations are created, they need to be stored and this leads to a number of requirements explained in the Storing section. Finally, the stored annotations need to be queried and this leads to a number of requirements explained in the Querying section.

A further element to the vision of RICORDO is to produce rich annotations of VPH resources, richer, in particular, than any single member of the CORDO dictionary set of IDs (discussed in RICORDO deliverable D2.1) would allow by itself. This is done through the production and maintenance of composite of ontology terms (RICORDO WP5). This layer adds a number of requirements which pertains to the creation of composites, their storing and the way they can be used in enriching the querying of annotations of VPH resources, specifically through reasoning. These will be addressed in the Composite section.

Authoring

In order to perform annotation of VPH resources, we have to access the resource that needs to be annotated. It is important to note that we will not manage the resources themselves, but be able to refer to the resources. In fact, due to the wide diversity and the large number of resources to be annotated it is evident that directly managing resources to be annotated will be impractical. This is because:

- Resources are stored in different types of repositories, they have separate access and the way to access these repositories as well as the way resources are stored (file format, *etc.*) may differ;
- Resources are not all equally available, some are proprietary or not public;
- Resource repositories can range from few hundred models to terabytes of data in clinical databases and, when taking into account the variety of resources, represent a large amount of data.

This indicates that the data for our annotation infrastructure must be different from the data sources. Our data will be annotation data, which we see as being created for the purpose of fulfilling the RICORDO objectives. Annotation data will be, in particular, created in accordance with standard guidelines. Since we cannot anticipate such guidelines at this stage in the project we can only make hypotheses on the basis of the foreseen independence of annotation data and source data.

The most generic and reusable scenario is one in which annotation data is created outside the specification of resources independently of source data. This also has the advantage of rendering the problem of dealing with a plurality of heterogeneous data which does not depend on the specification format used in the sources themselves. The annotations could be authored as a narrative of their own. Therefore the vision is one that corresponds to a self standing annotation authoring tool.

The basics of annotation, for this purpose, consist in linking a resource to an ontological term via an annotation relationship. There are three components to annotations and each imposes accessibility requirements for the authoring tools (see RICORDO deliverable D2.2 for a fuller discussion about this aspect). These are:

1. Access to identifiers for annotated resources;
2. Access to annotation relationships;
3. Access to ontological terms for annotation.

For the reasons expressed above, the aim of providing access to annotated terms is best understood as one in which an annotator (*i.e.* a user of the annotation authoring tool) declares an ID that stands for a term. This requires a representation of the resource, and possibly its elements. This is a key requirement for the annotation authoring tool that is discussed below.

Accessing IDs of ontological terms to annotate requires accessing the ontologies that contain them. The proposed ontological space which is referred to as CORDO dictionary currently includes FMA, GO, PATO, and Composites. Composites are being developed as part of WP5, which provides a methodology for representing the logical combination of ontology concepts. In particular from different ontologies to construct complex terms. These are the second type of data that the infrastructure will directly handle.

It is useful to enable finding IDs for complex ontological representations that will require querying the ontological space. This requires reasoning over the ontological space, *i.e*. preparing the ontological space

for running complex queries. This will require us to store the ontologies within the application server and use external services to work with them. This is further explained in the external resources section.

The simplest way of using IDs in the authoring tool is to allow direct input of the ID by the user into an apposite form. Provisions can be made for improving user experience in allowing search and auto completion. This involves relying on external elements as an option and will be dealt with in the section on external resources below.

It is also important that the authoring tool is readily accessible to the VPH community. Due to the large number of frequently updated resources, it is useful to always provide access to up to date resources. As RICORDO is a research project, there will be frequent updates to the application itself - a web application, therefore, would allow us to ensure that the VPH community has access to the latest application release, as well as to new resources.

Storing

Following on from the above discussion, it is clear that the infrastructure needs to make provisions for storing the result of annotation. This will involve storing relationships between resource IDs and ontology term IDs. Provision needs to be made for updating such storage.

In view of the nature of the resources and the prospective deployment of the annotation authoring tools in a web environment, it is envisioned that the storage of annotation also needs to be web-based. This also enforces update- and look-up-functionalities to be web based. This would impose hardware requirements on the prospective implementation of an infrastructure such as servers to deploy the applications and related data.

In relation to the elements of the annotation mechanisms above (resource term, relationship, ontology term), it is desirable to maintain the resource description necessary to the annotations together with the annotations themselves. Logically, there would be two stores: (i) one for the description of resources, forming a network of resource IDs, and (ii) one for their annotations, forming a knowledge base. While these two resources can be maintained separately, they will need to be handled in interoperable manner (in the same format). Thus we will treat them as one but an alternative would be to have two similar stores connected together.

To support annotation authoring, the steps taken to find IDs for annotation relationships and ontology IDs have to be planned. Even though extra resources are needed to maintain and store ontologies, this is inevitable due to reasoning requirements discussed in the earlier section. The ontologies that will be stored for our requirements will be a subset of the ontologies, thus this is not seen as duplicating resource availability.

Querying

The vision of the RICORDO infrastructure is to support the identification of relationships between VPH resources and to retrieve relevant resources upon request. A requirement for the infrastructure is therefore that it supports the querying of the annotation store. This includes typically a query engine and a query interface. The infrastructure will involve a number of online components. Updating the store is better thought of as a web-based task due to the complexity and the possible multiplicity of entry points from which to perform updates. It would also be useful to follow this procedure with a web-based query application. This has the advantage of fostering homogeneity in the infrastructure that includes applications for the creation format, storing and query of annotations.

Managing composite ontology terms

The requirements emerging from the role of composite terms in the overall vision of RICORDO involves the infrastructure in relation to:

- the authoring and querying of annotations with composites and
- the authoring, storing, and reasoning about composites themselves.

With respect to the use of composites in annotations, the strategy should ensure homogeneity between all sorts of annotations. Annotations using terms directly taken from the CORDO dictionary and annotations using composite terms need to be authored, stored and queried in a similar way. This means, in particular, that we expect composite terms to receive IDs, at creation time, which could then be used in annotations. As a result, we foresee, at this stage, no additional requirement imposed on the annotation management system when using composite terms.

With respect to authoring, storing and reasoning with composite terms, it is possible to draw a number of basic functional requirements for the supporting applications. In the first place, the authoring tools for composite terms need to support the assignment of IDs to these terms. Composite IDs, however, will not suffice since they will not carry the composite definitions in and of themselves. IDs are only pointers to definitions of composite terms. These definitions will need to be stored in a machine-interpretable form, in the form of an ontology. It is this store of composite definitions, the composite ontology as it were, that will lend support to reasoning with composite. Reasoning with composite will require a reasoning engine adapted to the storing formalism of composite definitions.

The only remaining requirement concerns the operational use of composite in relation to querying annotations. This is an integration requirement which needs to be fulfilled in order to allow enriching the annotation querying mechanism with intermediate reasoning on composite definition when adequate. In that connection, it is foreseeable that reasoning on ontologies, including composite definitions, will be deployed as a service in relation to annotation querying.

Motivation for Requirements

The following section presents the motivation for the explicit recommendations outlined in Table 1, and in the specific requirements sections that follow it.

In view of the above discussion, a web-based system is recommended. This will enable readily accessible framework to the VPH community with up to date annotated VPHDMRs, composites and applications. As discussed below, the aim is to provide web-friendly data formats for storing and representing annotations and composites.

With respect to authoring, querying, and storing annotations, the Resource Description Framework (RDF) is recommended for machine-interpretable metadata. It has a number of advantages, among which the fact that it is a W3C standard, that it is widely used and supported, that it comes with a query language (SPARQL) and that there is a number of tools and implementation of, for example, RDF stores readily available for prototyping.

The annotating step also requires defining the relationship between the elements and ontological resources. The Minimal Information Required In the Annotation of Models (MIRIAM) is a set of guidelines for annotation and curation processes of computational models to facilitate their exchange and reuse. MIRIAM supports a controlled annotation scheme for models based on Uniform Resource Identifiers (URI). The referenced information is described using a triplet {data type, identifier, qualifier}. Data type and identifier is combined in a single URI, such as: urn:miriam:dataType:identifier. Qualifiers refine the link between the model element and the piece of knowledge. MIRIAM specification suggests using RDF as a format for representing annotations. A number of VPH resources are already annotated using MIRIAM, such as SBML and CellML. Therefore, this presents a good starting point for our annotation standard. Accessing MIRIAM resources are discussed in the external resources below.

It is also important to consider the representation of composites. Existing knowledge bases such as GO, FMA *etc*. are mainly represented using OWL and OBO. OWL is more expressive and not all concepts represented in OWL can be transformed into OBO. Using OWL will also provide access to a variety of more or less powerful reasoners that can allow us to take advantage of composite definitions written in OWL.

Figure 1 below shows a high level diagram of the proposed architecture for the RICORDO infrastructure. The infrastructure is separated into three layers: web interfaces, core implementation, and data sources. Users will interact with the RICORDO framework via the web interfaces. These will allow users to annotate, create composites, and query annotated VPHDMRs. Core implementation handles user tasks executed by the web interface and interact with the RICORDO data sources. These include the metadata repository (which consists of the VPHDMR annotations) and ontology space which supports the RICORDO dictionary.

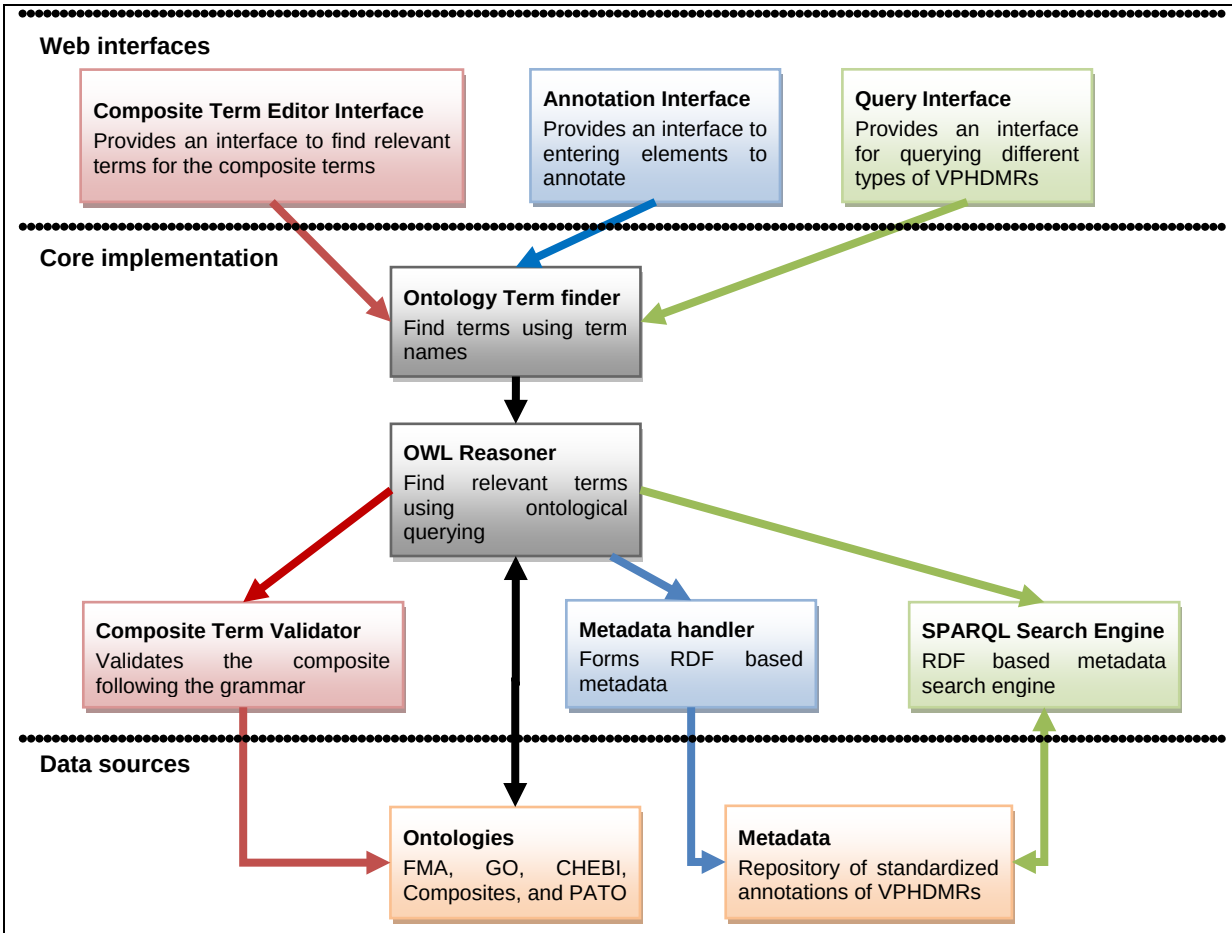

FIGURE 1: RICORDO INFRASTRUCTURE: Common functionalities used in RICORDO applications are shown using black boxes and arrows. The work flow of the composite editor application is illustrated by red boxes and arrows. Blue boxes and arrows follow the workflow of the annotation editor. Green boxes and arrows follow the workflow of the query application. The data stores accessed by the applications are shown using orange boxes.

Requirement Type	Requirement	Implementation strategy and accessibility
Hardware	32GB RAM 500GB HDD 4*CPU Redundant power supply Gigabit Ethernet/internet access	Hosted and maintained at EBI service infrastructure
Software	RDF triple store OWL reasoner Jena OWL-API GWT	Free software
Services	MIRIAM Ontology Lookup Service (OLS)	Services are available within the EBI.
External resources	Biomodels CellML OBO foundry	Collaboration with the community

TABLE 1: A CONCRETE LIST OF CONDENSED REQUIREMENTS.

Table 1 provides an overview of the technical requirements for RICORDO and our implementation strategy. The hardware requirements are derived from preliminary tests with deploying individual components of RICORDO, and the necessary hardware will be made available through the EBI, where an established service infrastructure for maintenance exists. The software on which RICORDO depends is available as free

software. The services are maintained by the EBI, and through collaboration with the maintainers, we ensure interoperability with these services. Furthermore, RICORDO depends on external resources that contain the annotated data such as models and reference ontologies. We will ensure access to these resources through active collaboration with the service providers, some of which are located at the EBI.

We draw upon the preceding section to provide specific requirements for the RICORDO infrastructure:

Requirements for browser based applications

RICORDO infrastructure is required to support three web applications which enable users to interact with VPHDMRs. These are:

- Annotation editor – to provide an interface to annotate VPHDMR resources with ontological concepts. The editor needs to support functionality to find relevant ontological terms which can be assigned to VPHDMR resources. The output of the tool needs to be an RDF statement which is stored in the meta-data repository.
- Composite editor – to allow users to integrate biological ontological terms to create complex ontological concepts. The editor needs to provide an interface to query for relevant ontological terms and combine multiple terms in conformance with the underlying composite grammar. The tool also needs to support the functionality to save composites into an OWL data store.
- Query service application - to allow users to search across annotated data repositories. This application needs to allow users to find ontological concepts that are in interest and search across the VPHDMR meta-data to find VPHDMRs that have mappings to the selected concepts.

Requirements for repository infrastructure

The infrastructure is required to support two repositories:

- Centralized RDF repository - to support storage of annotation information. The store needs to support basic information about the VPHDMRs and their elements, as well as their mappings to ontological concepts. Storing the data in RDF supports complex querying. SPARQL, a query language for RDF, is a good candidate for querying the RDF data store. Note that it is not our aim to store the model or data information.
- Store of ontological resources – RICORODO applications require reasoning over a number of large ontologies. It is a requirement to store the RICORDO ontologies in a server in order to be able to access the ontologies within the applications.

Requirements for hardware infrastructure

From the above analysis and the general requirements drawn, we propose prospective hardware requirements in support of the envisioned infrastructure.

Applications which are going to be deployed on the web will require server facilities. Similarly, data stores will need hosting. These are, more specifically, stores for:

- the RDF annotations;
- the composite definitions;
- the local storage of ontologies.

Based on the available ontological and data repositories and services, we envision that a minimum requirement would be 500 GB for server space.

In addition to server space for the above, the hardware infrastructure needs to make provision for supporting reasoning over ontologies. Such a task can be computationally expensive, and moreover, one has to take into account that the reasoning engine may function as a service, thus putting more stress on it.

We performed an experimental evaluation of reasoning over the CORDO ontologies. For this purpose, we used a development server with a dual CPU Intel Xeon 2.4GHz and performed two main tasks: classification of the FMA ontology, and queries over the FMA ontology. The FMA is the largest of the CORDO ontologies, and will therefore pose the greatest bottleneck. The first task (classification) is necessary only at a startup of the RICORDO service and took 38.2 seconds. Queries will have to be answered on a continuous bases, and take, on average, 0.4 seconds. Based on this experiment, we envision that 16 GB of memory will be required to achieve the goal of RICORDO. To guarantee that RICORDO can be extended with more ontologies, we require 32GB memory for RICORDO hardware.

Requirements for external resources

RICORDO requires access to large number of existing frameworks, services, and APIs:

- Java - as the implementation language to take advantage of other open source developments which are discussed in the following sections;
- Google Web Toolkit (GWT) - java version of GWT. It is a development toolkit for building and optimizing complex browser-based applications. It is an open source tool developed by Google to enable productive development of high-performance web applications;
- Jena API - Java based Jena API to handle RDF triples and SPARQL queries;
- MIRIAM Services - Mapped ontological terms are stored following the MIRIAM urn scheme. Therefore MIRIAM Web Services to resolve MIRIAM urns and MIRIAM resources;
- External repositories - We propose each external repository provider to upload their annotations to a FTP site with every release. We will access these FTP sites to update the central RDF store. This will allow us to synchronize the annotated data between their model files and our repository. It will also reduce the processing time when accessing large data sets.
- Ontology Lookup Service (OLS) - The applications that are developed in this work will require us to access a number of large ontological resources. Thus, OLS to query for ontological terms.
- Pellet reasoner - to reason across the OWL-EL version of the RICORDO ontologies to prepare the ontological space for running complex queries. Manchester query syntax to search the ontological space. The applications should allow users to construct Manchester queries using set of templates, so that the users are not required to have any understanding of the Manchester query syntax.

Performance requirements

The RICORDO infrastructure is required to efficiently support:

- Querying of large number of VPHDMR metadata: Currently annotations are stored in each model making it difficult to carry out efficient searches across different types of model repositories. As described in the repository infrastructure section, a central repository of annotations would make the task of querying them much easier. This will allow speedy querying of the data. It may also be relevant to explore possibilities of distributed annotation stores which can be queried in parallel to achieve efficient data retrieval times;
- Reasoning over a number of large ontologies: Existing reasoning applications do not support efficient querying over large ontologies. Therefore, as part of this work we need to find methods to efficiently query ontological data. OWL-EL version of the ontologies can be reasoned over within acceptable time. OWL-EL is a subset of OWL which supports automated reasoning while sacrificing some of the OWL expressivity.
- Protocol for evaluation: As part of this work we will evaluate the performance of the overall infrastructure and its components. We envision that optimization will modular, thus we will be able to improve performance on individual components (ontological reasoning, RDF storage, and querying etc.) as well as the overall infrastructure.

Conclusion

In summary, RICORDO infrastructure will provide:

- Three browser based applications - annotation editor for annotating VPHDMRs with ontological concepts; composite editor for creating composites that conforms to the underlying grammar; query application for querying across VPHDMR metadata;
- Repository infrastructure - RDF repository for storing annotations of VPHDMRs which we refer to as metatada and a store of ontological resources containing subset of ontologies which will be specifically used to annotate VPHDMRs;
- Hardware infrastructure – access to EBI servers for deploying the applications and RDF store and space for storing RICORDO ontologies;
- External resources – access to OLS for looking up ontological terms, pallet reasoner for querying the ontological space, RDF API for running SPARQL queries, and MIRIAM services for accessing MIRIAM urns.

REFERENCES

SBML – http://sbml.org/Main_Page
CellML – http://www.cellml.org/
FieldML – http://www.physiome.org.nz/xml_languages/fieldml
RDF – http://www.w3.org/RDF/
OWL – http://www.w3.org/TR/owl-features/
GO – http://www.geneontology.org/
FMA – http://sig.biostr.washington.edu/projects/fm/AboutFM.html
PATO – http://obofoundry.org/wiki/index.php/PATO:Main_Page
Java – http://www.java.com/en/
GWT – http://code.google.com/webtoolkit/overview.html
Jena – http://jena.sourceforge.net/
MIRIAM – http://www.ebi.ac.uk/miriam/main/
Pellet – http://clarkparsia.com/pellet/
OLS – http://www.ebi.ac.uk/ontology-lookup/
Manchester query syntax – http://www.w3.org/TR/owl2-manchester-syntax/

ANNEX

RICORDO Status of Preliminary Plans

We have started implementing and setting up parts of the infrastructure following the above requirements:

- Browser based applications - current implementation supports the query service application. The application run on all widely used web browsers such as Firefox at http://bioonto.gen.cam.ac.uk:8080/ricordo. The source code is available from http://code.google.com/p/ricordo/.
- Repository infrastructure - current implementation uses metadata stored in RDF files in the local file system. The aim is to move to an RDF store for the next release. The current set of meta-data and ontologies used in the application can be accessed through the code repository.
- Hardware infrastructure - currently we are using University of Cambridge resources for hosting our applications and data. The aim is to move all applications and data into the EBI infrastructure where there are access to larger computation power and space.
- External resources – we are using all the external resources described in the previous section except external repositories. We are communicating with curators with in SBML, CellML, and FieldML to adopt our annotation scheme. This will provide us with a large number of annotated models. We are also educating clinical data and statistical model communities about biological annotations in order to access their data through metadata.
- Performance requirements – we have developed an algorithm to convert GO, FMA, PATO, Composite ontologies into OWL-EL which can be reasoned over within acceptable time.