From a Services-based eScience Infrastructure to a Semantic Web for the Life Sciences: The Sealife Project

Michael Schroeder[1⋆], Albert Burger[23], Patty Kostkova[4], Robert Stevens[5], Bianca Habermann[6], and Rose Dieng-Kuntz[7]

[1] TU Dresden, Germany
[2] Heriot-Watt University, Edinburgh, UK
[3] MRC Human Genetics Unit, Edinburgh, UK
[4] City University, London, UK
[5] University of Manchester, UK
[6] Scionics, Dresden, Germany
[7] INRIA, Sophia-Antipolis, France

1 Introduction

The objective of SeaLife is the conception and realisation of a semantic Web/Grid browser for the life sciences which will link the existing Web to the currently emerging eScience infrastructure. The SeaLife Browser will allow users to automatically link a host of Web servers and Web/Grid services to the Web content he/she is visiting. This will be accomplished using eScience's growing number of Web/Grid Services and its XML-based standards and ontologies. The browser will identify terms in the pages being browsed through the background knowledge held in ontologies. Through the use of Semantic Hyperlinks, which link identified ontology terms to servers and services, the SeaLife Browser will offer a new dimension of context-based information integration.

More specifically, if the user points the mouse at a Semantic Hyperlink the SeaLife Browser offers a definition of the encountered term, the application of services relevant to the term, and to add the term to a shopping cart. After browsing through various pages and adding various terms to the shopping cart, the user decides to check out. The SeaLife Browser presents the contents of the shopping cart including the list of items collected, the type of the identified terms, and the sources where they were collected by the user.

The SeaLife Browser offers to apply additional services considering combinations of terms. For example, if the user collected a set of proteins, then the browser will offer to apply a tool to compare the proteins' sequences against each other, to create a multiple sequence alignment, or to query the literature for co-occurances of the two proteins. The user can save the current state of the shopping cart and return at a later stage to continue the semantic exploration.

To summarise, the SeaLife Browser links the existing Web to the new eScience grid infrastructure paving the way for a future generation Web for the life sciences.

⋆ Correspondance to: Michael Schroeder, Biotec, TU Dresden, ms@biotec.tu-dresden.de, +49 351 46340060

2 Case studies

To illustrate the power of this vision, the SeaLife Browser will be developed in the context of use case applications from the domain of infectious diseases. These applications vertically integrate the molecular/cell, tissue/organ, and patient/population layers by covering high-level information stemming from the national library of infectious diseases to detailed studies of high-throughput screening data for endocytosis, the entry pathway into the cell.

The **evidence-based medicine** use case envisions a scenario in which a clinician consults the national electronic library of infections to get curated and trusted information on infections, and is then guided by the SeaLife Browser to further relevant information in other resources, such as Ensembl and the Protein Databank.

The second application, on **literature and patent mining**, is aiming at a user browsing a patent database who, once relevant patents have been found, is offered additional web pages and services providing further details on the patents of interest.

The **molecular biology** use case centres around the biological process of "endocytosis" and links a protein to its sequence, leading to further information being offered with respect to relevant multiple sequene alignments and gene expression data.

3 The Sealife Components and their Interplay

For the SeaLife Browser to work and provide genuine support for the above use cases, there is an obvious reliance on the computer having some notion of domain semantics– what is the relationship between symbols in the language of biomedicine? To achieve the above vision, the following semantic problems need to be solved:

- Ontologies: Design and integration of ontologies and associated infrastructure, which can serve as background knowledge for a Semantic Grid browser geared towards life science applications ranging from the molecular level to the person level.
- Text Mining and Concept Mapping: Bridging the gap between the free text on the current Web and the ontology-based mark-up for the Semantic Web and Grid by developing an automated mark-up modules for free text, which are based on text-mining and natural language processing technologies.
- Service Composition: Bridging the gap between the ontologies of the Semantic Web and the services of the Grid by linking suitable ontology mark-up to applicable services and by supporting the interactive creation of such mappings for complex services.

3.1 Ontologies

At heart, an ontology is a structured set of vocabulary terms and their definitions that captures a community's understanding of its domain. the idea is to create a shared understanding of the symbols (terms) used to communicate in that domain. Thus, the Gene Ontology creates an agreed set of vocabulary terms for describing the major attributes of gene products. However, it is not only a facilitator for human communication. By

capturing this knowledge in a knowledge representation language with strict semantics, it is possible to enable machines to manipulate these symbols through the semantics of the language.

The Web Ontology Language (OWL) is the WorldWideWebConsortium's recommendation for representing ontologies for the Semantic Web. OWL has a strict semantics and its description logic version (OWL-DL) can be used for reasoning over the ontology and its instances. Many bio-ontologies, however, are represented in a more simple language that describes a directed acyclic graph (DAG). This allows only minimal machine usage, but it is directly transformable to OWL. The large number of ontologies in this form (all those in the Open Biomedical Ontologies collelction) offer a potentially vast background knowledge for the SeaLife Browser. Medical ontologies are available in a variety of representations. Some are open and some of these can be mapped into OWL with ease. Others, such as the Medical Subject Headings (MeSH) are a simple thesaurus design for informaiton retrieval and are not really automatically transformaable to OWL. Nevertheless, there is a large amount of biomedical ontology already extant for SeaLife Browser.

Protégé is the most widely used ontology development environment. Its OWL plugin offers a GUI style interface for building and using OWL ontologies. protege's wide range of plugins make it a rich environment. SWOOP, however, offers a much lighter development environment, but has considerable debugging facilities. Outside the OWL world, DAGEdit and OBOEdit are the most widely used tools in bio-ontologies. the former produces the DAG format of the OBO collection. OBOedit, a later development than DAGEdit, offers a richer environment with more modelling constructs.

Protégé, being a more robust and wide-ranging environment than the others, captures more of the principles for building ontologies. These can be split into two broad areas: First those that represent a software engineering approach and second, those that embody philosophical principles within ontology. The first are guidelines of requirements/scope; knowledge elicitation; design, conceptualisation; encoding; testing/evaluation; publication. these phases map onto a typical software engineering process and many tools and Protégé plugins exist for these stages. Philosophical aspects of ontology building represent the debate on what an ontology can and should represent; styles of building; writing definitions; etc.

One development principle not mentioned is that of re-using ontologies. As already mentioned, many ontologies exist in biomedicine. Once transformed to a common representation and thus a common language semantics, they must be either merged into one or mapped to one another. This is because ontologies can overlap etc. and these overlaps must be recognised and accomodated. A number of such integrations efforts exist within biomedical ontologies. One example is XSPAN.[8] This uses a cross-species ontology of anatomy from embryo stages to adult form. The terms from the various species have to be mapped and XSPAN have developed the COBrA tool to facilitate this mapping.

[8] `http://www.xspan.org`

3.2 Text-mining

The concepts of the ontologies have to be linked to text in web pages. This task is far from trivial as the concepts will occur in wide variations. The following problems need to be addressed:

- Information content of words: Consider the term *alkaline phosphatase activity* from the GeneOntology. A query on the literature database PubMed for *alkaline phosphatase* leads to more than two times more results than *alkaline phosphatase activity* and to more than ten times more results than *"alkaline phosphatase activity"*. This is particularly striking as the word *activity* is not very informative, as nearly one third of GeneOntology terms end in *activity*.
- Insertions and deletions of words: An ontology term may consist of several words, which are separated by inserted words in free text. For example, the text *...at a higher rate than freshly isolated monocytes upon activation...* should match the GeneOntoogy concept *monocyte activation* and the text *...large family of transcription factors that bind to ...* should match the term *transcription factor binding*.
- Stemming: Words such as *binding* and *binds* have to reduced to the stem *bind*.
- Sentence splitting: text-mining has to identify sentences as units. This is not trivial, as a dot separates two sentences, but it occurs also in abbreviations such as *ca.*, *etc.*, *C. elegans*.
- Special characters: Often ontology terms contain special characters such as slashes, commas, brackets, dashes, etc., which have to be treated appropriatedly. For example, the slash in the term *chromatin assembly/disassembly*, the slash acts as delimiter between two tokens, while in *Arp2/3 complex* the slash is no delimiter.
- Ambiguous concepts: Sometimes ontology concepts are not formulated unambiguously. for example, the term *small-molecule carrier or transporter* should have to match both *small-molecule carrier* and *small-molecule transporter*.

Sealife's text-mining module addresses these problems and thus maps concepts to text in the web pages.

3.3 Service Composition

Once terms have been identified in the SeaLife Browser, they are linked to other resources. A user can, for instance, put a sequence into their Sealife cart. This could be submitted to a service or series of services to perform an analysis. In many cases, more than one service will be used. The following issues will have to be addressed:

- Services will have to be discovered. Many thousands of services now exist. Currently, these are only described by their name and these are not necessarily informative. Efforts to semantically describe these services will reduce this barrier for both people and machines. What should be described? The following are some axes of description: input, output, task performed by the service, service name, algorithm used, etc.
- Once discovered, how are the services to be composed? Here the following issues are revealed:

- In many cases, bioinformatics services are implicitly typed. A service takes an input of string and gives an output of string. There is often much structure within one of these strings (for instance, a Uniprot record). Services are needed to locally impose some type on these strings in order to compose them.
- A minority of services have input and output in some structured XML document. Again, a variety of XML schema exist, so typing services are still needed. Nevertheless, the XML syntax of such input/output documents makes this process easier.
- A variety of typical type operations are needed in order to compose services: Access, coercion; etc.

An open system such as myGrid brings more of these problems than a closed system. In a closed system, it is easier to impose a type system, but it does place a barrier to third party services joining the system. SeaLife Browser will of necessity be open, so poorly typed services will be endemic. Composition of services will be part of the SeaLife Browser solution.

In all of the above areas, the work will build on existing protoype developments, such as the Gene Ontology Next Generation (GONG) project[9] , GoPubMed[10] and myGrid.

4 Conclusion

The SeaLife Browser will make eScience's web servers and services available to the bench scientists by using text-mining to identify ontology terms in free text and by linking the ontology terms to applicable services. The SeaLife Browser thus introduces the novel concept of semantic hyperlinks, which are generated on the fly and use the browser's background knowledge to dynamically link web pages to relevant services. The technical key challenges of the system are the design of ontologies, text-mining for concept mapping and service composition. For all three aspects, there are existing systems and results such as the ontology editor GONG, the ontology-based literature search engine GoPubMed, and the bioinformatics grid system myGrid. These will form the backdrop for the realisation of the SeaLife Browser, which will be applied to the study of infectious diseases ranging from the patient and clinician exemplified by the National electronic Library of Infectious diseases[11] to molecular biologists studying endocytosis.

Acknowledgements

A full-length version of this extended abstract is published by the HealthGrid 2006 conference, held in Valencia, Spain, 7-9 June 2006.

Funding by the EU projects Sealife (FP6-2006-IST-027269) and REWERSE (FP6-2006-IST- 506779) is kindly acknowledged.

[9] http://gong.man.ac.uk
[10] http://www.gopubmed.org
[11] http://www.neli.org.uk