A2-D3

Requirements and specification of bioinformatics use cases

Project title:	Reasoning on the Web with Rules and Semantics
Project acronym:	REWERSE
Project number:	IST-2004-506779
Project instrument:	EU FP6 Network of Excellence (NoE)
Project thematic priority:	Priority 2: Information Society Technologies (IST)
Document type:	D (deliverable)
Nature of document:	R (report)
Dissemination level:	PU (public)
Document number:	IST506779/Dresden/A2-D3/D/PU/b1
Responsible editors:	Björn Olsson and Michael Schroeder
Reviewers:	Andreas Doms and Albert Burger
Contributing participants:	Bucharest, Dresden, Edinburgh, Jena, Linköping, Lisbon, Paris, Skövde
Contributing workpackages:	A2
Contractual date of deliverable:	31 August 2005
Actual submission date:	31 August 2005

Abstract
This deliverable specifies use cases based on bioinformatics research carried out by members of A2. The use cases involve the use of rules to reason over ontologies and pathways (Dresden, Edinburgh, Paris, Linköping) and rules to specify workflows to integrate bioinformatics data (Lisbon, Skövde, Jena, Bucharest). The use cases are designed as a reference point to foster the take up of A2 use cases by I-workpackages. Most notably, many of the use cases specify the need for querying and reactivity with languages like Xcerpt (I4), Erus (I5) and Prova (I5). The use cases range from basic research applications to fully deployed software with an international user base.

Keyword List
Use cases, bioinformatics, metabolic pathways, PubMed, GeneOntology, gene expression data, sequence alignment, PDB, SCOP, ontologies, web services, databases, rules, reasoning, XML

Project co-funded by the European Commission and the Swiss Federal Office for Education and Science within the Sixth Framework Programme.

Requirements and specification of bioinformatics use cases

Rolf Backofen[Jen], **Liviu Badea**[Buc], **Pedro Barahona**[Lis], **Mikael Berndtsson**[Sko], **Albert Burger**[Edi], **Gihan Dawelbait**[Dre], **Andreas Doms**[Dre], **Franois Fages**[Par], **Anca Hotaran**[Buc], **Vaida Jakonienė**[Lin], **Ludwig Krippahl**[Lis], **Patrick Lambrix**[Lin], **Kenneth McLeod**[Edi], **Werner Nutt**[Edi], **Bjorn Olsson**[Sko], **Michael Schroeder**[Dre], **Anna Schroiff**[Sko], **Loic Royer**[Dre], **Sylvain Soliman**[Par], **He Tan**[Lin], **Doina Tilivea**[Buc], **Sebastian Will**[Jen]

[Buc] National Institute for Research and Development in Informatics, Bucharest, Romania, [Dre] Technische Universität Dresden, Germany, [Edi] Heriot-Watt University/MRC Human Genetics Unit, Edinburgh, UK, [Jen] Friedrich-Schiller-Universität Jena, Germany, [Lin] Linköpings universitet, Sweden, [Lis] Universidade Nova de Lisboa, Portugal, [Par] INRIA Rocquencourt (Paris), France, [Sko] University of Skovde, Sweden

31 August 2005

Abstract

This deliverable specifies use cases based on bioinformatics research carried out by members of A2. The use cases involve the use of rules to reason over ontologies and pathways (Dresden, Edinburgh, Paris, Linköping) and rules to specify workflows to integrate bioinformatics data (Lisbon, Skövde, Jena, Bucharest). The use cases are designed as a reference point to foster the take up of A2 use cases by I-workpackages. Most notably, many of the use cases specify the need for querying and reactivity with languages like Xcerpt (I4), Erus (I5) and Prova (I5). The use cases range from basic research applications to fully deployed software with an international user base.

Keyword List

Use cases, bioinformatics, metabolic pathways, PubMed, GeneOntology, gene expression data, sequence alignment, PDB, SCOP, ontologies, web services, databases, rules, reasoning, XML

Contents

Contents

1 Introduction

The deliverable A2-D3 sets out to deliver the following contents:

> Requirements and specification of use cases
>
> Based on the previous deliverables detailed use cases will be specified. The use cases document typical flows of information and usage of tools and computational services, which could be automated and integrated in a bioinformatics Semantic Web. Strategies have to be developed of how these computational services can be integrated in a Semantic Web scenario. The use cases will pick specific computational services and show how they have to be extended. With the use cases defined, a concrete specification of a sample application will be laid down. Which specific reasoning and rule-based techniques are relevant for the above scenario? A report will be written detailing the specific nature of rules and constraints in a bioinformatics setting. The need for handling imprecise information and the role of ontologies will be discussed. Additionally, the need to integrate computational aspects with the reasoning and constraint satisfaction will be pursued. The resulting report prepares the ground to effectively communicate to other working groups the needs for rules in a bioinformatics Semantic Web context.

A2-D3 summarizes the use cases and tools developed by various A2 members, which have the potential to integrate REWERSE technologies. The described use cases cover a wide range of bioinformatics tasks that are representative examples of the kinds of problems addressed in real bioinformatics applications. These use cases have been selected also to be advanced and challenging, and illustrating the need for Semantic Web technology. The use cases should not be seen as separate stand-alone example applications, but rather as parts of a larger scenario, where a researcher applies a number of different bioinformatics tools for investigating different aspects of a particular biological system. A description of an overall sample application is provided at the end of this document.

Before we introduce the use cases in detail, let us summaries them according to their maturity, and the web and rule technologies, which are needed to realize the use cases. All of the proposed use cases are extending existing deployed or research prototypes and sketch opportunities for web/rule technologies. The scope of the systems ranges from basic research to fully deployed systems with many real users world-wide. Most of the proposed use cases can be directly linked to I4 and I5 technologies in that they require some form of querying and sometimes reactive behavior. Some of the use cases refer already directly to technologies investigated in workpackages I4 and I5, namely Prova (I5), Xcerpt (I4), and Erus (I5). Common to many use cases is the need to work with web services, ontologies, databases, and XML documents.

- Name: GoPubMed: Exploring PubMed with the GeneOntology

  GoPubMed is a novel search engine for the biomedical literature, which uses ontologies to classify search results.

  - Partner: Dresden

  - Status: Deployed (At peak times over 150 queries a day, in summer on average 50 hits a day)
  - Web technologies: XML, Xquery, Xpath, Cocoon
  - Rule technologies: Prova (I5), Xcerpt (I4), and Prolog to query GeneOntology.

- Name: Edinburgh Mouse Atlas

  The Edinburgh Mouse Atlas uses an anatomy ontology to explore a database of mouse tissue with gene expression data

  - Partner: Edinburgh
  - Status: Deployed
  - Web technologies: Java and queries to remote database
  - Rule technologies: Prolog and Prova (I5) to reason over anatomy

- Name: Chemera

  Chemera is a molecular modeling tool to explore protein structures. It integrates information from remote sources via web services.

  - Partner: Lisbon
  - Status: Partly deployed/partly applied research
  - Web technologies: An XML-based modeling language to integrate web services into the application
  - Rule technologies: Rules to integrate the services (I4,I5)

- Name: Biocham

  Biocham uses computational tree logic to specify and reason over metabolic pathways.

  - Partner: Paris
  - Status: Applied/Basic research
  - Web technologies: Ontologies
  - Rule technologies: Computational tree logic, inductive logic programming

- Name: Samba

  Samba is a tool to integrate ontologies.

  - Partner: Linköking
  - Status: Applied research
  - Web technologies: The ontologies are represented in DAML-OIL or OWL
  - Rule technologies: Description logic reasoning with Racer (I4)

- Name: Aligning sequences using prior knowledge

  Often researchers have specific background knowledge to be considered in sequence alignments, but classical tools cannot handle such constraints. The aim is to develop a tool to address this shortcoming.

  - Partner: Jena
  - Status: Basic research
  - Web technologies: -
  - Rule technologies: Constraints

- Name: Analyzing gene expression data

  To interpret gene expression data the biological literature is mined and networks are extracted.

  - Partner: Bucharest
  - Status: Basic research
  - Web technologies: Web services, ontologies, Xquery, XML
  - Rule technologies: F-logic, Prova, Xcerpt (I4,I5)

- Name: Consistent mirroring of bioinformatics resources with Erus

  Bioinformatics databases often change and there are dependencies. Exemplified by the protein databank PDB and the structural classification of proteins SCOP the use case illustrates how to maintain consistent mirrors of these databases.

  - Partner: Skövde
  - Status: Basic research
  - Web technologies: HTML, XML, database queries
  - Rule technologies: Erus reaction rule engine (I5)

The sections below give further details on each use case and application. We conclude by providing an overall scenario that links many of these tools to put them into context.

2 GoPubMed (Dresden)

2.1 Description

The biomedical literature grows at a tremendous pace. PubMed, the main biomedical literature database references over 15,000,000 abstracts. Due to this size, simple web-style text search of the literature often does not yield the best results, and a lot of important information remains buried in the masses of text. GoPubMed aims to offer a web service which allows users to explore PubMed search results with Gene Ontology, a hierarchically structured vocabulary for molecular biology.

GoPubMed is an existing prototypical system. Current usage statistics show about 50 hits per day from institutions like Max Planck Institute of Molecular Physiology (269), the Medical College of Wisconsin (214), University of Michigan Medical Center (50) and Lever Brothers Ltd. (45).

The prototype uses web technologies like XQuery for indexing the Medline database of about 15,000,000 articles. The articles are indexed with a GO term extraction module and the results are cached in a relational database. Based on the web application framework Cocoon 2.1 an

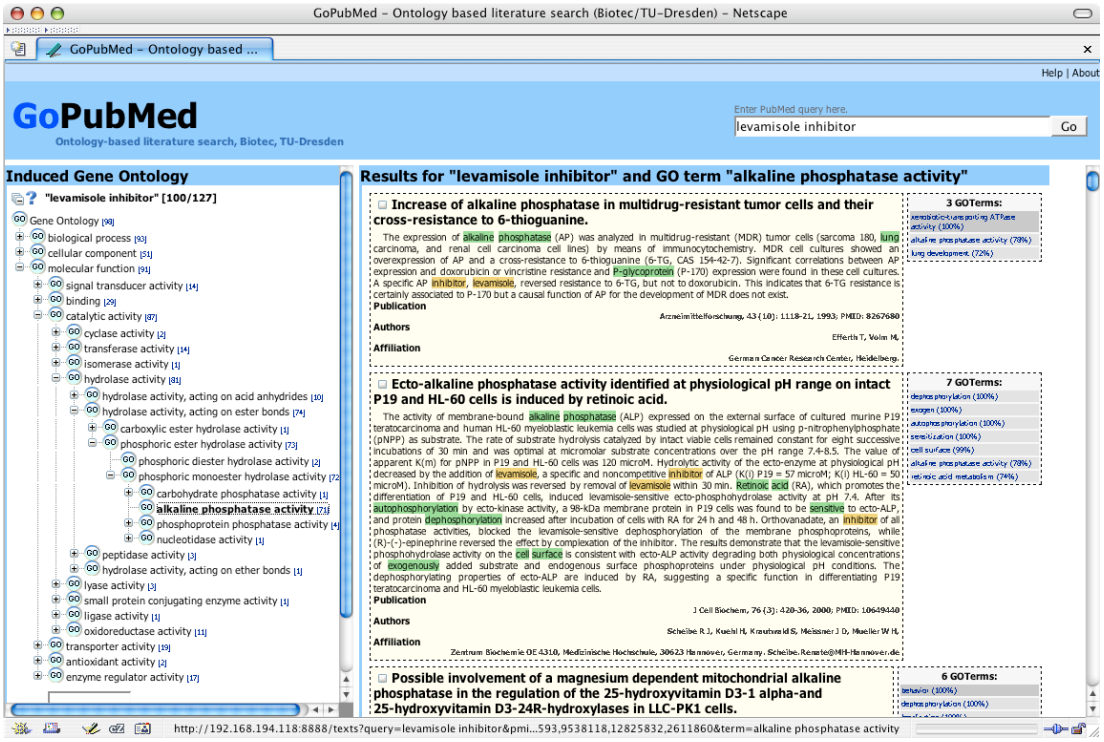

Figure 1: User interface of GoPubMed. The screen-shot of GoPubMed displays the results for the query "levamisole inhibitor" limited to 100 papers. On the left, part of the GeneOntology relevant to the query is shown and on the right the abstracts for a selected GO term. The search terms are highlighted in orange and the GO terms in green. Right of each abstract is a list with all the GO terms for that abstract ordered by an accuracy percentage. E.g. is the term *P-glycoprotein*, which is a synonym for the GO term *xenobiotec transporting ATPase*, is found with 100% accuracy, while *lung development* matches only with 72%, as only the word "lung" occurs in the abstract. Synonyms, such as the term *P-glycoprotein* above, are displayed in dark grey and the synonymous term is given in a tool-tip (please note, that Mozilla based browsers do currently not break lines in tool-tips). Moving the mouse over the term displays the definition of the term in a tool-tip. The ontology on the left shows the paths from the root of the ontology - *cellular component*, *biological process*, and *molecular function* - to the currently selected GO term. The number in brackets behind each GO term in the ontology is the number of papers the GO term or any of its children occur in. In the figure, the path from *molecular function* to *alkaline phosphatase* is shown and the number 71 behind the term *alkaline phosphatase* indicates that there are 71 papers mentioning alkaline phosphatase. Clicking on the term displays the relevant abstracts, which confirm that levamisole inhibits alkaline phosphatase. Overall, the number of papers containing a term and its children is a very good indicator to let users select the most frequent terms and thus best representatives. Instead of using the ontology to browse through abstracts, users can also display all the abstracts in the same order as in PubMed with the additional benefit of displaying the GO terms and search keywords. Users can also search within the ontology using the input field at the bottom of the ontology. GoPubMed searches are currently limited to 100 papers per query. Answering a query takes around 20 seconds.

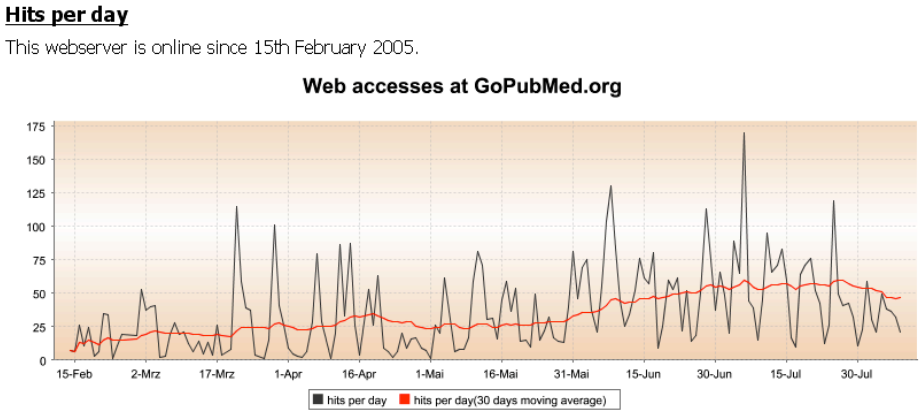

Figure 2: Hits per day for www.gopubmed.org

application pipeline was set up. The user issues queries to the GoPubMed web server. The application receives research articles in XML form from PubMed. In the processing pipeline this XML stream is annotated with terms from Gene Ontology, which is stored in an SQL database. The resulting annotated XML document is sent trough the processing pipeline of the Cocoon application to be rendered as an induced ontology tree. To calculate this tree the ontology is represented as a DAG and the graph must be traversed. Currently this is done with a proprietary reasoning component which is planned to be replaced by a more powerful reasoner. This reasoner must then be able to handle complex Boolean queries. The results are rendered to an XML document and later transformed (using XSLT transformations) into a highlighted HTML document.

2.2 Concrete scenario

A researcher wants to know which enzymes are inhibited by levamisole. A keyword search for levamisole inhibitor produces well over 100 hits in PubMed. To find out about specific functions, the researcher has to go through all these papers. He/she is interested in the relevant enzymatic functions. From the first titles it is immediately evident that levamisole inhibits alkaline phosphatase. A less well-known fact is however still buried in the abstracts. The abstract *The effect of levamisole on energy metabolism in Ehrlich ascites tumor cells in vitro* with PMID 2947578 is ranked very low (position 89 on 7/2/2005[1]) by PubMed. The abstract states that levamisole also inhibits phosphofructokinases. Most readers will miss this statement.

Even if the user would try to reduce the number of papers by filtering out the ones mentioning levamisole inhibitor (e.g. query PubMed for levamisole inhibitor NOT phosphatase), he or she would miss the less obvious hits like phosphofructokinase, if both terms occur in the same

[1]Please note, that all examples depend on PubMed's ranking of search results. Since the literature is growing, PubMed may return different articles for the same query at different time points. This means that GoPubMed may display different papers for the examples in this report. All queries in this paper were checked on 8 Feb 2005.

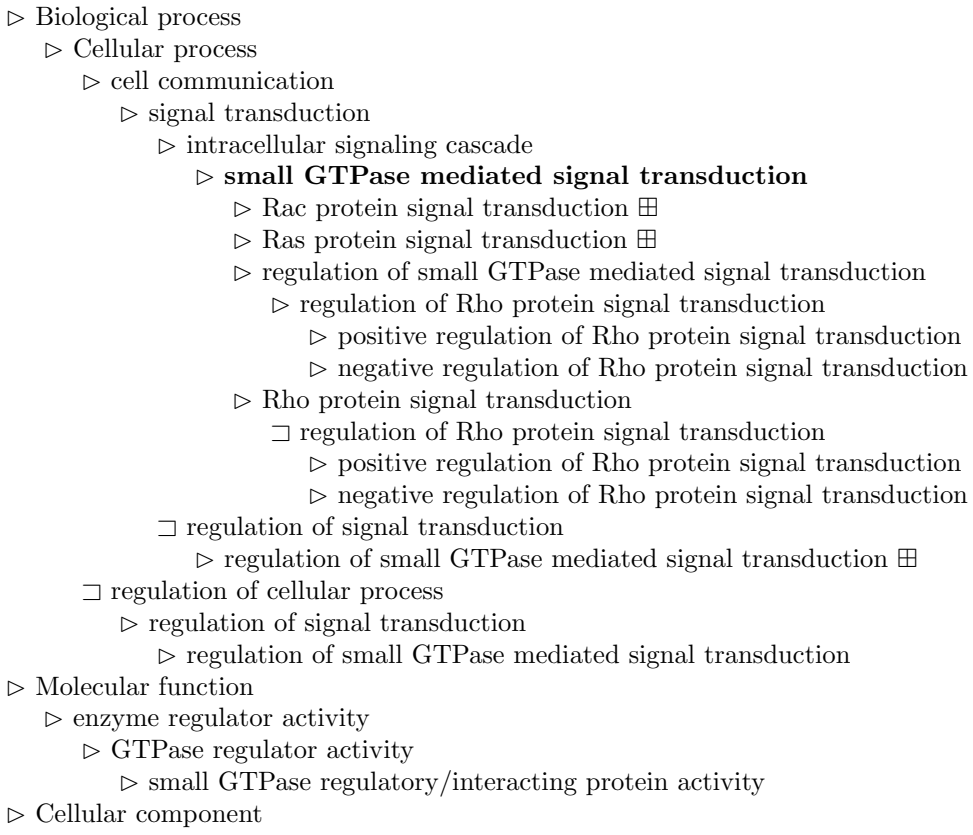

Figure 3: The GO sub-hierarchy containing terms related to small GTPases. ▷ symbolizes an *is_a* relation and ⊐ symbolizes a *part_of* relation. Nodes marked ⊞ hide more children related to "small GTPase mediated signal transduction". Note: this tree view is stripped down to the concepts of GO necessary to explain the example. The subtree related to *regulation of Rho protein signal transduction* is present twice because this GO term has multiple parents. The relations in GO are graphs.

abstract. Thus, even advanced PubMed queries with Boolean logic cannot always properly structure the search results.

Figure 3 shows a small fraction of GO. The available formats for GO are OBO XML, RDF, flat file and a relational database. Any query language used for GoPubMed in the later version must be able to reason over one of those formats - preferably over the standard format RDF.

For keeping the articles index up to date it would be useful to have some mechanism for automatic indexing of new available articles in the PubMed database. With the fully built index of all PubMed articles it is then possible to ask for all research publications on "'Small GTPases"' (and related concepts) but excluding abstracts related to a specific small GTPase like "'Rho"'.

For a query language to be used in the next version of GoPubMed it would also be useful to be able to ask for articles which mention concepts in GO not only in the same subtree but also siblings or cousins of the concept.

3 Edinburgh Mouse Atlas (Edinburgh)

3.1 Description

The Edinburgh Mouse Atlas (EMAP) and the Gene Expression (EMAGE) Database project [Brune et al., 1999, Davidson and Baldock, 2001, Davidson et al., 1997, Ringwald et al., 1994] has developed a digital atlas of mouse development which provides a bioinformatics framework to spatially reference biological data. The core databases contain 3D grey-level reconstructions of the mouse embryo at various stages of development, a systematic nomenclature of the embryo anatomy (the anatomy ontology), and defined 3D regions (domains) of the embryo models which map the anatomy ontology onto the spatial models. Through the 3D domains users can navigate from the spatial representation of the embryo to the ontology and vice versa. Data from an *in situ* gene expression database is spatially mapped onto the atlas allowing the users to query gene expression patterns using the 3D embryo model and/or the ontology as a reference.

As with all developments of ontologies, there is a trade-off between the effort that can be expended on its creation and the level of formalization and detail that can be achieved; the current version of the mouse anatomy ontology is relatively informal. With a view to integrate the Mouse Atlas system with other bioinformatics resources, and as part of the ongoing efforts to develop the ontology further, work has been carried out in formalizing aspects of the current representation of the anatomy, giving a more precise description of its semantics [Burger et al., 2004].

3.2 Concrete scenario

Having clarified some of these semantics, it is now possible to exploit further reasoning over gene expression data indexed by the Mouse Atlas anatomy ontology. Based on the meaning of the part-of relationships between tissues[2], e.g. the finger is part of the hand, which is part of the arm, a number of rules can be applied. In particular, the propagation of properties of tissues, such as gene expression, along the part-of hierarchy is of interest. Some of the rules that apply are:

- Given that gene X is expressed in tissue A, it is known that X is also expressed in all tissues B of which A is part of. For example, in general, a gene expressed in a finger is also expressed in the hand (and the arm).
- If one knows that a gene Y is not expressed in tissue A, i.e. nowhere in A, then all parts B of A will not have this gene expressed. For example, if a gene is not expressed in the arm, it cannot be expressed in the hand or finger.
- A gene X *may be* expressed in tissue B if X is known to be expressed in tissue A and B is part of A and there is no evidence that X is not expressed in B.

A selection of such rules has been implemented in Prolog. For this, content that is stored in the Mouse Atlas databases was exported into Prolog knowledge bases. While in principle this is a possible route to implement reasoning over Mouse Atlas content, note that the primary development environment for the Mouse Atlas is Java and data is stored in relational databases.

[2]The term "tissue" is used in a very general sense, i.e. referring to any anatomical structure, not just specific tissue types.

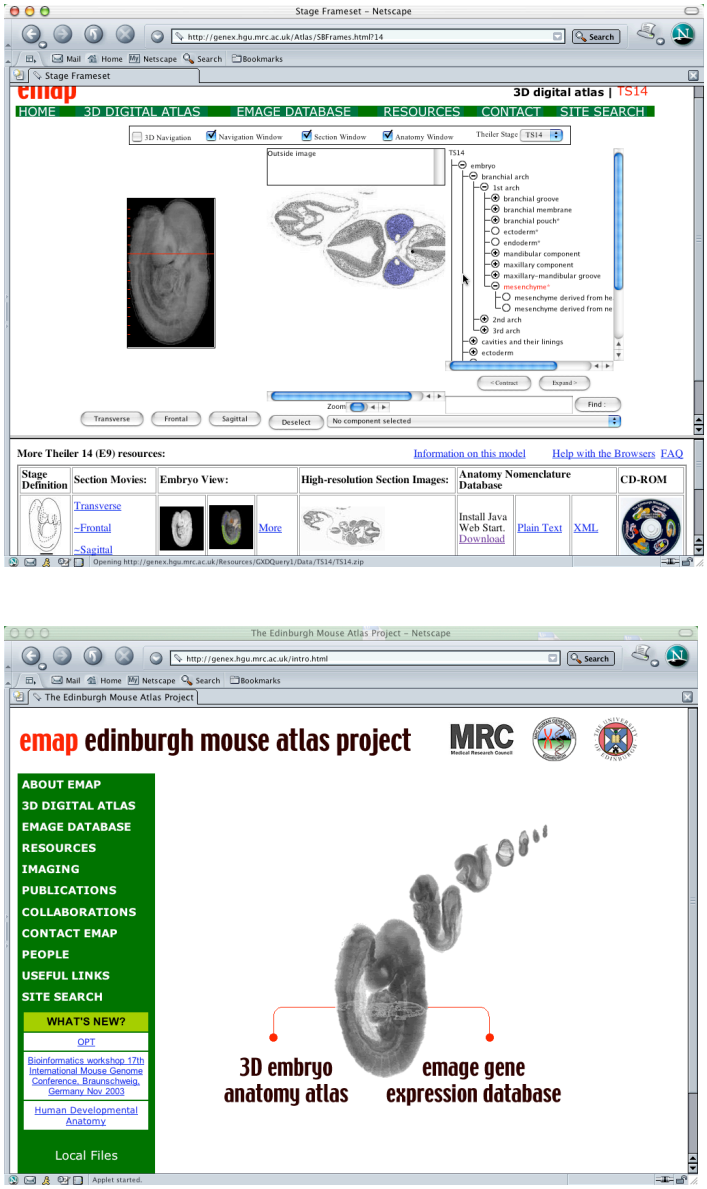

Figure 4: Screenshot of the Edinburgh Mouse Atlas

The purpose of this use case is to evaluate newer developments, specifically the Prova language, for the implementation of rules and reasoning over the Mouse Atlas resource. Prova has been selected because of its combined support for Java and Prolog-style reasoning, thus perhaps best matching the requirements for the Mouse Atlas.

The evaluation will consider functional as well as performance aspects, comparing a Prolog-based approach with a Prova-based approach in terms of effort that is required to implement rules such as the ones above and the time it takes to apply these rules over different gene-expression data profiles for the Mouse Atlas anatomy ontology.

4 Chemera (Lisbon)

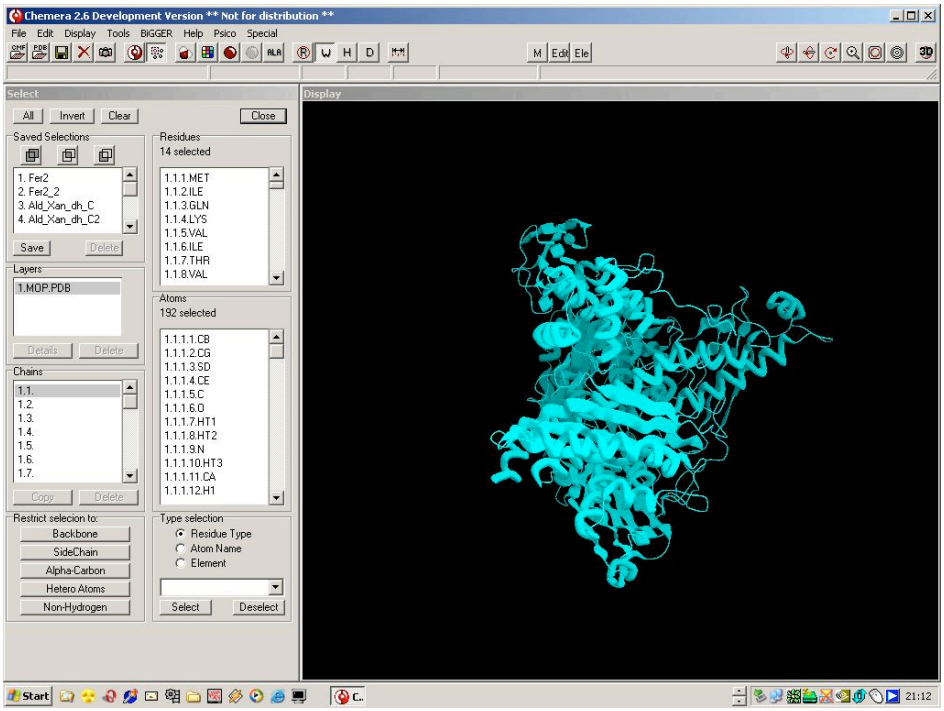

Figure 5: Screenshot of the Chemera molecular modeling application

4.1 Description

Chemera is a molecular modeling application that was developed by the Lisbon partners of the A2 group before the REWERSE project. Due to its graphical interface and capability to work with protein structures, Chemera provides a good foundation for incorporating technologies developed in the REWERSE project.

With these objectives, Chemera 3.0, released in June 2005, now implements the interface to two services available on the internet. These are the DSSPCont secondary structure assignment server at Columbia University and the Protein Domain Server at the Biomolecular Modeling Laboratory, Cancer Research UK. These pilot cases served to test the concept and assess the work necessary to integrate into a molecular modeling package the information available from internet services in general, and also increase the impact of the REWERSE project by providing a useful bioinformatics application of the research done in the project.

4.2 Concrete Scenario

The implementation of this test case suggested to us a more ambitious project, which Lisbon has begun to work on. In line with the goals of the REWERSE project, Lisbon aim to create a simple XML-based language to provide machine-readable instructions on how to use bioinformatics web services. This language will include service ontology, to provide human users with a structured description of what each service can do, and machine-readable parsing instructions and descriptions of the data and how to access it. This will potentially enable services to equip software capable of interpreting the language with the ability to interface with the service, and will also make it easier to keep up-to-date applications that use Internet services, like Chemera does, simply by updating the web pages providing the machine-readable service descriptions.

At this stage a first sketch of the interpreter implemented in Chemera is ready as part of a test case. Specifically, the markup specifications developed in WG I1 can provide a helpful framework for the syntax and definition of the ontology of the service descriptors; the composition techniques developed in WG I3 will help us identify and solve the problems of integrating different services; the work of WG I4 on reasoning-aware querying is useful for interfacing a large segment of the available bioinformatics services, which consist of databases; and at a later stage, the work of WG I5 on reactivity will provide the means for automated or semi-automated updating of the descriptor pages in response to changes in the services available on the Internet.

5 Biocham (Paris)

5.1 Description

The mass production of post-genomic data, such as ARN expression, protein production and protein-protein interaction, raises the need for a strong effort on the formal representation of biological systems. Knowledge on gene interactions and pathways is currently gathered in databases and ontologies, with a number of tools for making simulations based on these databases when numerical data are present [Backofen et al., 2004, Backofen et al., 2005].

Beyond numerical simulation, the possibility of performing symbolic computation on biomolecular interaction networks opens the way to the design of a new kind of automated reasoning tools for biologists/modellers.

Paris project with the Biochemical Abstract Machine [Fages et al., 2004] [3], started in 2002, is one attempt in this direction. BIOCHAM provides a precise semantics to qualitative biomolecular interaction maps as concurrent transition systems [Chabrier-Rivier et al., 2004]. Based on this formal semantics, BIOCHAM offers:

[3] `http://contraintes.inria.r/BIOCHAM/`

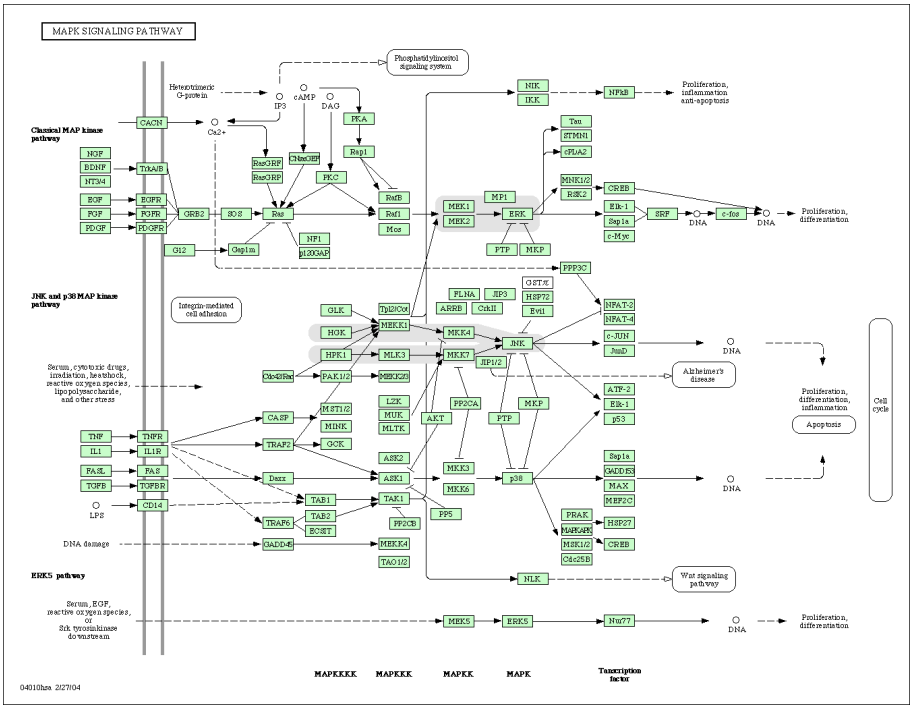

Figure 6: Screenshot of a pathway reasoned over with Biocham

- a compositional rule-based language for modeling biochemical systems, allowing patterns, and kinetic expressions when numerical data are available;
- numerical and Boolean simulators;
- an original query language based on temporal logic CTL [Clarke et al., 1999] for Boolean models and LTL with constraints for numerical models, used for expressing biological queries about reachability, checkpoints, oscillations or stability;
- a machine learning system to infer interaction rules and kinetic parameters from observed temporal properties.

The machine learning system in BIOCHAM allows to discover interaction rules from a partial model and *constraints* on the system behavior [Calzone et al., 2005]. These constraints are expressed using the temporal logic query language as a specification language. The learning process can be guided by the user by providing patterns for limiting the types of sought reactions, such as complexation, phosphorylation, etc. Similarly, the machine learning system also supports the learning of kinetic parameter values from a specification in temporal logic with constraints on numerical quantities.

5.2 Concrete scenario

As explained above, the machine learning system of BIOCHAM relies on the user for limiting the search for interaction rules, by providing patterns of rule types, like complexations or phosphorylations. However, the burden of defining these *constraints* on the search space is left to the user.

When one looks at the sum of knowledge present in the aforementioned ontologies, it seems nevertheless possible to automate most of the work of defining plausible reactions, by combining the interactive querying of web resources to complex reasoning in the corresponding ontologies.

Let us take the example of a modeler working on a model of the Mitogen Activated Protein Kinase Cascade, like the one of [Levchenko et al., 2000]. This model represents the transduction of a signal from outside the cell, into the nucleus. Suppose that the modeler is interested in adding interaction rules representing a second way to go from the receptor to the product of the cascade. Now, instead of writing patterns describing all the reasonable rules for each of the molecules already defined, the modeler would like to search the web about, for instance the MEK protein, and automatically get constraints on the type of reactions in which MEK can be involved.

Gene Ontology (GO) [Ashburner et al., 2000] - being at the same time reliable, thanks to its careful curation process, and comprehensive, both on the molecular function side and on the biological process side - will be taken as the main source of information for this use case.

A first query about the term MEK, in the molecular function ontology of GO, looking for exact terms or **synonyms**, will return the *MAP kinase kinase activity* term (GO:0004708). Using the defined relation **is a** (twice) one can observe that MEK's function is a *protein kinase activity* (GO:0004672). This function is often annotated with the term *protein amino acid phosphorylation* (GO:0006468) from the process ontology and with 95% significance. And that process **is a** special case of *phosphorylation* (GO:0016310), which is a general process that allows to derive a BIOCHAM pattern like: `$P =[MEK]=> $P~$Q`.

A map of the used hierarchy is provided by tools such as QuickGo[4].

[4] `http://www.ebi.ac.uk/ego/index.html`

This type of query could be repeated, to refine the pattern (a MAPKK only phosphorylates MAP kinases), to define other reactions involving MEK (MEK itself is phosphorylated by MAP kinase kinase kinases), and for the other molecules.

In this example, one sees the necessity of:

- a very efficient query mechanism, considering the size of the manipulated ontologies;
- term retrieval, based on the content of some fields;
- reasoning using the defined relationships of the ontologies;
- tackling the use of different hierarchies related by secondary fields and its corresponding significance measure.

Since the user can also define his patterns, the querying process must be fast, but also probably interactive, allowing to propose the output of the query to the user for refinement, validation or re-querying. This is especially true of *unsure* links like the above.

Finally, for the sake of simplicity, this example was restricted to only one information source, but the possibility to query several sources, especially for the definition of synonyms, is crucial to the usability of the system.

A prototype should thus, by querying web sources like GO and reasoning on them, output constraints on the plausible rules involving a given compound. Evolution of the ontology data does not seem to be very important in a first stage, thus querying languages like Xcerpt[5] should be enough for that task. However the definition of *rules* for treating the uncertainty of the co-annotation measure might be considered, in order to enhance the quality of the results. This prototype could itself become a web service, available not only to BIOCHAM's machine learning system, but to all the Systems Biology community.

6 Samba (Linköping)

6.1 Description

Many bio-ontologies have already been developed and many of these ontologies contain overlapping information. Often one would therefore want to be able to use multiple ontologies. For instance, companies may want to use community standard ontologies and use them together with company-specific ontologies. Applications may need to use ontologies from different areas or from different views on one area. Ontology builders may want to use already existing ontologies as the basis for the creation of new ontologies by extending the existing ontologies or by combining knowledge from different smaller ontologies. In each of these cases it is important to know the relationships between the terms in the different ontologies. Aligning two ontologies means to define the relations between terms in the different ontologies. Merging two ontologies based on the alignment relations between the ontologies creates a new ontology containing the knowledge included in the source ontologies. Currently, there exist a number of systems that support users in merging or aligning ontologies in the same domain. These systems use different techniques, but it is not clear how well these techniques perform for different types of ontologies. Few comparative evaluations on ontology merging and alignment have been performed [Lambrix and Edberg, 2003, Lambrix and Tan, 2005b, Fensel et al., 2002,

[5] `http://www.xcerpt.org/`

Fernández-López and Gómez-Pérez, 2002] and no tools for supporting these kinds of evaluations exist yet [García-Castro et al., 2004].

In this work Linköping develops a general framework for ontology alignment and comparative evaluations of ontology alignment strategies and their combinations. The group has identified different types of strategies and shows how these strategies can be integrated in one framework. Linköping developed a system according to this framework. Within this system there are already existing alignment algorithms as well as some new algorithms. The framework can also be used to experiment with combinations of strategies. Further, Linköping wants to experiment with different kinds of ontologies. For the tests several well-known bio-ontologies are used [Lambrix, 2004]. A first step has already been taken in [Lambrix and Tan, 2005a, Lambrix and Tan, 2005b] with experiments on some strategies and one type of ontology (taxonomies).

The work is basic research. A prototype for an alignment and merging system will be developed as a further development of the SAMBO system [Lambrix and Tan, 2005b, Lambrix and Tan, 2005a]. The current prototype accepts ontologies represented in DAML+OIL or OWL as input and generates DAML+OIL or OWL ontologies. The current prototype uses a description logic system (FaCT or RACER) to check whether the merged ontology is consistent, to find unsatisfiable concepts and cycles in the ontology and to remove redundancy. In future work, when experimenting with expressive ontologies, it is expected to use more complex reasoning technology to find alignment relationships.

6.2 Concrete scenario

The framework is tested using well-known bio-ontologies such as Gene Ontology (GO), Signal Ontology, Medical Subject Headings (MeSH) and the Anatomical Dictionary for the Adult Mouse as test ontologies, and WordNet and UMLS as auxiliary resources.

7 Aligning sequences using prior knowledge (Jena)

7.1 Description

Aligning DNA and protein sequences has become a standard method in molecular biology for performing similarity searches. Often, it is desirable to include partial, maybe imprecise, prior knowledge and conditions, which are formulated by rules and constraints, in tools for sequence alignment.

However in general, similarity searching tools and computational services on the web do not allow to take such prior knowledge into account automatically. The reason for this deficiency is of algorithmic nature. The most common and successful technique for efficient alignment algorithms is dynamic programming (DP). However, a weakness of DP is that one cannot include additional constraints without specifically tailoring new DP algorithms.

Jena follows a declarative approach that is based on constraint satisfaction techniques and shows how it can be extended by formulating additional knowledge as rules and constraints. Jena takes special care to obtain the efficiency of DP for sequence alignment. This can be achieved by careful modeling and applying proper reasoning strategies, e.g. the constraint solving technique of cluster tree elimination. Jena's approach integrates data from different sources, namely sequence data with additional information on the molecules beyond pure sequence data, which may stem from annotations and ontologies.

7.2 Concrete scenario

Consider the case of a biologist, who knows that certain regions in her sequences share a common local motif. This knowledge could be (at least partly) automatically extracted from data sources in the web. Based on this knowledge, the sequences should be compared. Then, one needs to optimize similarity under the additional constraint that parts of such regions should be matched to each other. Another striking example is the enhancement of RNA or protein comparison by employing knowledge on the structure of the macromolecules, which again could stem from databases or (especially in the case of RNA) from computational structure prediction.

A tool based on these techniques could, as a web service, even automatically query and include information from available data sources. There, this approach provides the tools to integrate this data via extracted rules into the comparison of sequence information.

8 Analyzing gene expression data (Bucharest)

8.1 Description

This use case describes a scenario involving the use of biological literature databases for extracting the networks and pathways differentially activated in microarray data.

Microarrays have revolutionized biology by enabling the simultaneous measurement of expression levels (mRNA) of virtually all genes of an organism. In principle, this should allow deciphering the molecular-level mechanisms of complex diseases such as various types of cancer or type 2 diabetes by determining the genes that are differentially expressed between normal and disease samples. In practice, there are several tough problems with such a simplistic scenario.

1. Current microarray technology is extremely noisy and costly, which limits the number of samples available and thereby reduces the statistical relevance of the results of tests used for determining differential expression.

2. Even in a noise-free setting, there may be hundreds of differentially expressed genes and understanding the mechanism of action of a given disease requires determining the causal relationships between these genes.

3. Microarrays only measure the expression of genes at the mRNA level, while many signaling and metabolic pathways involve compounds that cannot be directly measured by this technology. (mRNAs are relatively easy to measure due to the digital nature of their code.)

The use case deals with a real-life lung cancer microarray dataset ([Bhattacharjee et al., 2001]), in which samples from different types of lung cancers (adenocarcinoma, small cell, squamous and carcinoids) are available.

Due to the small number of available samples, a strict cutoff in the tests of differential expression would entail a large number of false negatives (genes that are truly differentially expressed but fall below the threshold), although the number of false positives may be limited. Therefore, a lower cutoff must be used to be able to obtain more differentially expressed genes. This entails however a large number of false positives (genes that are not truly differentially expressed but happen to have by chance a signal-to-noise ratio above the threshold).

In the absence of any additional knowledge, microarrays are therefore of limited use, since the intrinsic noise in the measurements makes the results uncertain.

On the other hand, there is a huge body of experimental work that has been published in the relevant biological literature and which could be used to interpret the results of microarray

experiments. (Although many pathway databases currently exist, they are far from being up to date even combined w.r.t. the published literature.)

As web technologies the use case will employ XQuery for implementing the wrappers to the information sources and various Web services from NCBI (e-utilities) and ontologies (GO).

Reasoning technologies: Bucarest plans to use Bucarest's F-logic based Semantic Web system as well as Prova and Xcerpt for this application. There are certain important problems with these systems at this time: The Semantic Web system is currently under development (it will have the necessary facilities in its final version, but not at this time). Among other problems, Xcerpt does not currently allow variables in resources, requires well formed XML or HTML input (which is not the case with current data sources) and cannot call external programs (e.g. tidy) to convert these to a well-formed format. Currently Prova does not allow query planning or streaming (necessary in this application).

However, Bucarest intends to proceed with the implementation of the above scenario with one of the above-mentioned systems. The complexity of the application will undoubtedly reveal useful real-life constraints on the (semantic) web systems under development in the project.

8.2 Concrete scenario

Thus, given a set of genes (e.g. the genes that are differentially expressed between normal and squamous carcinoma samples), one needs to retrieve literature co-citations (e.g. phrases occurring in PubMed abstracts) mentioning (causal) links between genes from the given set.

Note that the (causal) links from the literature and the microarray data are complementary. As explained above, microarray data alone may be too noisy to allow the identification of networks and pathways, while the causal links from the literature have been determined in different contexts (different tissues and/or experimental conditions) and thus may not be indiscriminately combined in pathways without guidance from microarray data (which contains simultaneous measurements of virtually all genes and thus may be able to link the fragments from the literature into a coherent whole).

Note also that Bucarest is interested in literature citations that mention relationships between genes, rather than grouping genes based on their annotations (e.g. from the Gene Ontology), since the fact that two genes have the same annotation does not imply their involvement in the same pathway.

There are certain difficulties of a semantic nature related to the task of finding literature co-citations for a given set of genes: Genes may occur in the literature under various names (synonyms), for example the keratin 5 gene has the standard name KRT5 in the NCBI Genbank database, but occurs under various aliases in the literature, such as K5, CK5, EBS2, KRT5A, Cytokeratin 5, 58 kDa cytokeratin. The literature may be unspecific to a certain extent, e.g. mention just keratins (instead of referring to a precise keratin gene, such as KRT5). Certain gene names may be ambiguous, i.e. refer to several quite distinct genes (e.g. p150 may refer to either one of the following: SH2BP1, RAB3-GAP150, CHAF1A, PIK3R4, ABL1). Various shorthands used in the literature may be confused with gene symbols (for example the short-hand RA for the disease rheumatoid arthritis may be confused with the gene symbol RA). Thus the correct interpretation of a particular symbol depends on the context (which should be determined somehow without sophisticated Natural Language Processing tools). A certain paper may refer to a protein complex rather than explicitly to its components (e.g. the transcription factor AP-1, which is a JUN/FOS complex). Experimental papers frequently mention lists of genes, for instance specific markers which are tested in that paper, without there being a (causal) link

between these genes. On the other hand, one may be interested in genes that are somehow causally connected (e.g. genes that co-occur in the same sentence along with verbs such as regulates, activates, inhibits, interacts, etc.).

A workflow for this application starts by extracting sentences (with action verbs) from PubMed abstracts that mention at least a pair of genes from the given gene set (or their synonyms). The result of this initial phase is an initial network model, in which genes from the given gene set are connected if they co-occur in PubMed abstracts with action verbs. This model is then refined using a battery of refinement modules implemented using rules.

For example, (quasi-)cliques in the co-citation graph may be due to the corresponding genes being controlled by a common regulator or set of regulators (transcription factors). Thus, since transcription factor databases such as TRANSFAC are highly incomplete, one may search PubMed with gi promoter (where gi are the corresponding genes) and look for common transcription factors (genes with appropriate GO annotations).

For example, the genes VWF, RHOA and SPARCL1 are differentially expressed in squamous samples (w.r.t normal ones), but they are controlled by the common transcription factor ERG (which is differentially activated rather than controlled at a transcriptional level).

In Xcerpt syntax:

```
CONSTRUCT
binds_promoters[ var TF, all var Gene ]
FROM
and{ binds_promoter[ var TF, var Gene ], binds_n_promoters[ var TF, var N ] }
where { var N > 3 }
END

CONSTRUCT
binds_promoter[ var GeneName1, var GeneName ]
FROM
and {
in { resource {"diff_expr_genes.xml" },
diff_expr_genes {{ Gene_Symbol [var GeneName] }}
},
pubmed_query [  join(var GeneName,  promoter), var GeneName1 ],
go_query[ var GeneName1, molecular function, transcription regulator activity ]
}
END

CONSTRUCT
binds_n_promoters[ var TF, count (all var Gene) ]
FROM
binds_promoter[ var TF, var Gene ]
END
```

Even if there are no papers experimenting the binding of a transcription factor (TF) of interest to a given promoter, there may be papers dealing with a related TF (from the same family/class). Thus one may wish to query PubMed for citations containing gi promoter together with TFs with the same DNA binding domain as some other TF from the ones already found to be involved in the process under study. For example, the gene CITED2 has been cited to have an Ets-1 binding site in its promoter, as does CDH5. Since REG and Ets-1 are both from the same family of Ets transcription factors, one may wish to search for other genes with Ets binding sites in their promoters, e.g. TEK and ANGPT1 both have such sites.

Note that the literature contains information that can be used either as: causal explanations for finding differentially expressed genes (if g1 is cited to activate g2 transcriptionally, then this causally explains why both are found to be differentially expressed), or functional associations that may justify why the genes are coordinately controlled (if g1 and g2 interact somehow, e.g. belong to the same protein complex, or are involved in the same signaling pathway, then this represents a reason for them being transcribed in a correlated manner, although this does not reveal the causal details of how such a transcriptional regulation is actually performed).

Even if certain mechanistic details of some pathways are currently known, many other details are still obscure, so it is typically very useful to be able to query whether some of the differentially expressed genes are involved or controlled by known pathways. For instance, the TGF-beta pathway is important in the case of squamous lung cancer.

Other queries may concentrate on determining potential chromosomal translocations involved, so one may be interested in differentially expressed genes with similar chromosomal location.

Implementing the above scenario requires tools that go beyond currently implemented ones. Still, certain aspects can be realized with current technology. Starting with a set of differentially expressed genes (taken from diff_expr_genes.xml), the following Xcerpt rule returns lists of co-cited genes together with the corresponding PubMed IDs (assuming that the in construct allows for variables):

```
CONSTRUCT
gene_pubmed_gene [gene [var GeneNameLL], pubmed [all var PMID], var GeneNameLL1]
FROM
and {
in { resource {"diff_expr_genes.xml" },
diff_expr_genes {{ Gene_Symbol [var GeneName] }}
},
in { resource {
join("http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=gene&te
rm=",
var GeneName, "[GN]+'Homo sapiens'[Organism]" }, eSearchResult {{ desc Id
[var GID] }},
},
in { resource {
join("http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=gene&id=
",
var GID, "&mode=xml") },
Entrezgene-Set {{ Entrezgene {{ desc Gene-ref_locus [var GeneNameLL] }} }}
},
in { resource {
join("http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=gene&
db=pubmed&id=",
var GID) },
eLinkResult {{ desc Link {{ Id [var PMID] }} }}
},
in { resource {
join("http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=pubme
d&db=gene&id=",
var PMID) },
eLinkResult {{ desc Link {{ Id [var GID1] }} }}
},
in { resource {
join("http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=gene&id=
",
var GID1, "&mode=xml") },
```

```
Entrezgene-Set {{ Entrezgene {{ desc Gene-ref_locus [var GeneNameLL1] }}
}}
}}
END
```

9 Consistent mirroring of bioinformatics resources with Erus (Skövde)

9.1 Description

EruS is based on *Use Case 6.4.2: Mirroring, Actuality, and Consistency of data in SCOP and PDB*, a bioinformatics use case previously reported in the I5- D2 deliverable. It considers the bioinformatics databases PDB (Protein Data Bank) and SCOP (Structural Classification of Proteins) and includes the use of a rule system as basis for the implementation. Thus, EruS links the work of the REWERSE working groups A2 and I5 by implementing consistent integration of bioinformatics data by means of rules.

PDB contains data about protein structures. SCOP consists of a hierarchy of protein structures, with data almost completely derived from that in PDB. SCOP provides parseable files for download which are used locally for large scale analysis as opposed to using the original remote version of SCOP (or one of its mirrors). The current version of SCOP, 1.69, released in July 2005, is based on 25973 PDB entries and one literature reference. These are all proteins of known structure as of 1 October 2004 [Murzin et al., 2005]. The previous update was released in February 2005 because the complex update procedure, which is mostly performed manually, takes time so that new releases are usually published biannually. PDB in contrast is updated weekly, including the addition of new datasets and new protein structures [Berman et al., 2000].

The described situation comprises limitations and problems concerning the use and maintenance of the data sources. First of all, the manual update handling in SCOP is not only inconvenient and time consuming, it also makes research error- prone since the *actuality* of the data used can not be guaranteed. Another problem that needs to be tackled are *inconsistencies* arising from the different update rates in PDB and SCOP. They lead to inconsistencies in SCOP as e.g. references to PDB may have been updated or deleted since the last SCOP update. In case of a deletion of a PDB entry, the SCOP entry is based on a non- existent PDB entry, in case of an update the derivation from the PDB entry might no longer be correct. The fact that even data in the remote, original version of SCOP can not be guaranteed to be up-to-date with current research already integrated in PDB induces flaws and slows down progress in research.

The following specific use cases can occur in the described environment:

Deletion of a PDB entry: In case of a deletion in PDB, the corresponding SCOP entries include information about a PDB entry that no longer exists. This inconsistency must be handled by updating the SCOP entry and adding the information about the now obsolete PDB entry and its substitute.

Update of a PDB entry: If new information about a protein structure is available or an error in a PDB entry is discovered, this entry will be updated in PDB. The handling of this update in SCOP is not trivial, as the change to the PDB entry may effect the classification of the structure domain in SCOP or, on the other hand, may not be relevant for SCOP at all. The classification of domains is mostly done manually, therefore it is not possible to change it as part of the automatic handling of the PDB update event. An

alternative is to edit the SCOP entry and add the information about the modified PDB entry to indicate the potential change of its classification in the next SCOP release, if the update is relevant for SCOP (e.g. modification of PDB entry's coordinates).

New entry in PDB: A more advanced use case would be not only to handle inconsistencies, i.e. updates or deletes of existing data in PDB, but also to assure the actuality of SCOP data, i.e. to handle inserts in PDB immediately to keep PDB and SCOP synchronized. This includes the classification of new structures. As mentioned above, the classification is not an automated procedure. Therefore, the handling of new entries is not tackled at this stage of the project.

Thus, it can be summarized that the use cases regarded in EruS so far are the deletions and updates of PDB entries.

The limitations and problems described above can be tackled by the automation of the synchronization process between the data sources. As the systems in use can not be changed, an additional system is needed, which can act as a mediator for the interaction of the existing systems.

The task to be performed in *Use Case 6.4.2* of I5-D2 is the handling of events (e.g. updates, deletes) in PDB by performing an appropriate action (e.g. update) in a local copy of SCOP. This behavior can be established by a rule system containing event-condition-action rules. The use of a rule system can achieve a more efficient interaction between the databases by replacing manual handling with an automatic mechanism.

The rule system filters the update events for the relevant modification types and executes an action by updating the SCOP parseable files. At the current stage, information about relevant updates and the deletes including the replacing entry is added to a comment file. An additional script is provided to look up the comments for a specific SCOP node which can be used in case the comment file is not handled by the particular the application used with the SCOP parseable files.

The rule system used as basis for the EruS prototype is ruleCore^{TM}. It provides a GUI to define ECA rules in an XML-like format called rCML, a rule engine to process incoming events, evaluate rules and execute actions and also monitoring and debugging devices [MS Analog Software kb, 2003].

9.2 Concrete Scenario

A researcher is examining a protein and uses the SCOP parseable files in an application for large scale analysis of protein structures. He is looking for evolutionary related proteins and gets a set of protein domains as results for further examination. The problem he faces is that he might be dealing with out- of-date information since the current SCOP release only includes PDB data up to October 2004. To find out if there have been any relevant changes to the protein structures he has to look up the PDB entry for each protein domain in the result set and check if there were any changes to the PDB entry. If so, he has to find out whether those changes have any impact on the protein domain he obtained from SCOP and its classification.

To simplify this procedure, the researcher uses EruS to obtain an updated version of the SCOP data. Now the only requirement is to check the comment of the resulting SCOP domain using the provided script. Only relevant updates of PDB entries are listed here, i.e., changes to the structures and deletes. This way he can check the results of his analysis and thus avoid building assumptions and arguments on wrong data. E.g., if the PDB entry of one of the found

SCOP protein domains has been deleted, the researcher will be informed about the PDB entry replacing the now obsolete entry.

The concrete scenario described above comprises the current stage of the project. In a future scenario, the EruS can be extended with composite rules and the changes can be applied to all parseable files, not only the comments file, always regarding the SCOP hierarchy. In more advanced stages, semantics should be added to the EruS rule base in order to include the handling of new PDB structures to provide a completely updated SCOP copy which is consistent and complete with regard to the latest PDB update.

10 Conclusion

All of the above use cases use web/rule technologies and some show direct links to technologies developed in I4 and I5. Besides the realization of the above use cases with REWERSE's rule and web technologies, the use cases can be put together to specify a broader scenario of a Semantic Web for the life sciences. Consider the following scenario.

A researcher is investigating the genetic background of a particular type of cancer. He/she may first use Pubmed to find candidate genes/proteins, using a keyword search. This will return a set of candidate genes which have been mentioned in the literature as associated with the type of cancer of interest. Using the contribution from the Dresden node (GoPubmed), the researcher can then refine the candidate set, by exploring the initial search results with Gene Ontology.

The researcher also has access to gene expression data for this cancer type, showing the expression of genes in tumor samples and in samples taken from healthy subjects. Using this expression data, the researcher extracts a set of differentially expressed genes and then applies the method developed by the Bucharest node to retrieve literature co-citations mentioning causal links between genes from this set. As a result from this analysis, the user obtains a set of genes which can be added to (enriches) the initial candidate set that was obtained by the GoPubmed keyword search.

The researcher might align some of the sequences from the enriched candidate set and observe the presence of a common motif. He/she now wants to search in public sequence databases for homologues to the initial set of genes. Unlike when using standard homolog search techniques, such as BLAST, the researcher wants to take into account the previous knowledge that any homolog matches should also include the identified motif, as is possible by using the method developed by the Jena node.

As result from previous analysis steps, the researcher now has a set of genes that are candidates for being involved in the pathways regulating the growth of the cancer tumor. The researcher may then use the method developed by the Paris node (Biocham) to build a model of this system, by defining the interaction rules from current knowledge obtained from the literature. If temporal gene expression data is available, the researcher may also use machine learning to infer interaction rules. Having build this model, the researcher can formulate different queries to better understand the dynamics and behavior of the system.

The researcher now wants to "zoom in" on parts of the system to gain an understanding of the molecular-level details of particularly interesting parts, for example the binding and/or complex formation of specific enzymes or the binding of key transcription factors to the target genes that they regulate. He/she therefore applies the system developed by the Lisbon node (Chemera) to build a molecular-level model of the relevant parts of the system. This allows the

testing of hypotheses regarding the molecular interactions of key components of the system.

It is hoped that parts of such a far-reaching scenario can be realized in a REWERSE Bioinformatics Semantic Web.

References

[Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29.

[Backofen et al., 2005] Backofen, R., Badea, L., Barahona, P., Burger, A., Dawelbait, G., Doms, A., Fages, F., Hotaran, A., Jakoniene, V., Krippahl, L., Lambrix, P., McLeod, K., Möller, S., Nutt, W., Olsson, B., Schroeder, M., Soliman, S., Tan, H., Tilivea, D., and Will, S. (2005). Usage of bioinformatics tools and identification of information sources. deliverable A2-D2, Lisbon.

[Backofen et al., 2004] Backofen, R., Badea, M., Burger, A., Fages, F., Lambrix, P., Nutt, W., Schroeder, M., Soliman, S., and Will, S. (2004). State-of-the-art in bioinformatics. Deliverable A2-D1, Dresden.

[Berman et al., 2000] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Research*, 28:235–242.

[Bhattacharjee et al., 2001] Bhattacharjee, A., Richards, W., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E., Lander, E., Wong, W., Johnson, B., Golub, T., Sugarbaker, D., and Meyerson, M. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences of the USA*, 98(24):13790–13795.

[Brune et al., 1999] Brune, R., Bard, J., Dubreuil, C., Guest, E., Hill, W., Kaufman, M., Stark, M., Davidson, D., and Baldock, R. (1999). A Three-Dimensional Model of the Mouse at Embryonic Day 9. *Developmental Biology*, 216(2):457–468.

[Burger et al., 2004] Burger, A., Davidson, D., and Baldock, R. (2004). Formalization of mouse embryo anatomy. *Bioinformatics*, 20:259–267.

[Calzone et al., 2005] Calzone, L., Chabrier-Rivier, N., Fages, F., Gentils, L., and Soliman, S. (2005). Machine learning bio-molecular interactions from temporal logic properties. In Plotkin, G., editor, *CMSB'05: Proceedings of the third Workshop on Computational Methods in Systems Biology*.

[Chabrier-Rivier et al., 2004] Chabrier-Rivier, N., Fages, F., and Soliman, S. (2004). The biochemical abstract machine BIOCHAM. In Danos, V. and Schächter, V., editors, *CMSB'04: Proceedings of the second Workshop on Computational Methods in Systems Biology*, volume 3082 of *Lecture Notes in BioInformatics*, pages 172–191. Springer-Verlag.

[Clarke et al., 1999] Clarke, E. M., Grumberg, O., and Peled, D. A. (1999). *Model Checking*. MIT Press.

[Davidson and Baldock, 2001] Davidson, D. and Baldock, R. (2001). Bioinformatics Beyond Sequence: Mapping Gene Function in the Embryo. *Nature Reviews Genetics*, 2:409–418.

[Davidson et al., 1997] Davidson, D., Bard, J., Brune, R., Burger, A., Dubreuil, C., Hill, W., Kaufman, M., Quinn, J., Stark, M., and Baldock, R. (1997). The mouse atlas and graphical gene-expression database. *Seminars in Cell and Developmental Biology*, 8(5):509–517.

[Fages et al., 2004] Fages, F., Soliman, S., and Chabrier-Rivier, N. (2004). Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *Journal of Biological Physics and Chemistry*, 4(2):64–73.

[Fensel et al., 2002] Fensel, D., Corcho, O., Fernández-López, M., Gómez-Pérez, A., Angele, J., Sure, Y., Stutt, A., and Christophides, V. (2002). Deliverable 1.3: A survey on ontology tools. Technical report, OntoWeb Consortium.

[Fernández-López and Gómez-Pérez, 2002] Fernández-López, M. and Gómez-Pérez, A. (2002). Deliverable 1.4: A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies. Technical report, OntoWeb Consortium.

[García-Castro et al., 2004] García-Castro, R., Maynard, D., Wache, H., Foxvog, D., and González-Cabero, R. (2004). Deliverable 2.1.4: Specification of a methodology, general criteria, and benchmark suites for benchmarking ontology tools. Technical report, KnowledgeWeb Consortium.

[Lambrix, 2004] Lambrix, P. (2004). Ontologies in bioinformatics and systems biology. In Dubitzky, W. and Azuaje, F., editors, *Artificial Intelligence Methods and Tools for Systems Biology*, chapter 8, pages 129–146. Springer-Verlag.

[Lambrix and Edberg, 2003] Lambrix, P. and Edberg, A. (2003). Evaluation of ontology merging tools in bioinformatics. *Proceedings of the Pacific Symposium on Biocomputing*, 8:589–600.

[Lambrix and Tan, 2005a] Lambrix, P. and Tan, H. (2005a). A framework for aligning ontologies. In *Third Workshop on Principles and Practice of Semantic Web Reasoning*.

[Lambrix and Tan, 2005b] Lambrix, P. and Tan, H. (2005b). Merging DAML+OIL ontologies. In Barzdins and Caplinskas, editors, *Databases and Information Systems - Selected Papers from the Sixth International Baltic Conference on Databases and Information Systems*, pages 249–258. IOS Press.

[Levchenko et al., 2000] Levchenko, A., Bruck, J., and Sternberg, P. W. (2000). Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties. *PNAS*, 97(11):5818–5823.

[MS Analog Software kb, 2003] MS Analog Software kb (2003). *ruleCore 1.0 Users Guide v. 0.1.3*. www.rulecore.com.

[Murzin et al., 2005] Murzin, A. G., Chandonia, J.-M., Andreeva, A., Howorth, D., Conte, L. L., Ailey, B. G., Brenner, S. E., Hubbard, T. J. P., and Chothia., C. (2005). Scop classification statistics. Accessed 30/07/2005.

[Ringwald et al., 1994] Ringwald, M., Baldock, R., Bard, J., Kaufman, M., Eppig, J., Richardson, J., Nadeau, J., and Davidson, D. (1994). A database for mouse development. *Science*, 265:2033–2034.