

The implications of Semantic Web technologies for support of the e-Science process

Jun Zhao, Phillip Lord, Pinar Alper, Chris Wroe, and Carole Goble
School of Computer Science
University of Manchester
Oxford Road
M13 9PL Manchester UK

zhaoj,plord,penpecip,cwroe,carole@cs.man.ac.uk,

Abstract

The vision of e-Science is progressively being realized as diverse computational resources are being used to collect, store, process and integrate scientific data and to execute the scientific method *in silico*. The *myGrid* project, which operates in the bioinformatics domain, has so far delivered the basic middleware for second generation *in silico* biology, which previously was undertaken via adhoc integration of web based resources. The Semantic Web (SW) vision on the other hand is a promise to enable increased mechanization by giving information a well-defined meaning. In *myGrid* we have applied certain SW technologies for two activities, namely resource discovery and provenance. Here we discuss our experiences in these two cases. We believe that our findings on utility of SW technology to bioinformatics can be generalized to the larger context Life Sciences and even e-Science.

1 Introduction

e-Science *in silico* experiments complement experiments performed in the laboratory by generating new information from available data and forming hypotheses for confirmation in the laboratory. The promise of e-Science—bringing together distributed, complex, heterogeneous resources—is an attractive one for bioinformatics. This field has traditionally stored data and provided tools from distributed centres around the world. A working bioinformatician hoping to perform *in silico* experiments will often require access to many different resources.

Workflows have been accepted as an efficient means of orchestrating complex tasks over a distributed set of resources by many scientific computing projects within the academic community. Different disciplines which apply *in silico* experiments to their research activities show diversified requirements. At

one extreme, particle physics experiments produce vast data sets and corresponding computational loads, with a corresponding requirement to deal with the machinery of classical high performance computing (HPC) and networking (HPN). The life sciences domain is characterised by massive variety in terms of data types and the resources to operate on them. Thus, *in silico* experiments for the biology plays a greater emphasis on composing a wide range of services and managing highly heterogeneous data resources provided by autonomous groups.

In this environment, the *myGrid* project¹ has developed middleware and applications which provide the necessary components for service based exposure of resources and tools (Soaplab²) and their orchestration into workflows (Taverna³ [15]). This enables biologists and bioinformaticians to develop and enact *in silico* experiments straightforwardly, where previously advanced programming skills would have been required. We have demonstrated the utility of this approach, with a number of case studies into Graves Disease [10] and Williams-Beuren syndrome [17].

Currently Taverna has integrated over 1,000 services into its environment, each of which falls into different service families. This eases the problem of resource harvesting and integration. However, composing complex workflows still remains as a challenge for biologists. Choosing from such a hefty number of heterogeneous services is problematic and rely on a user scanning a simple picking list. Name searches using text matching has limitations as many of the names are obscure and do little to convey the purpose of the service.

Provenance is the metadata about the origin of a data resource and how it was generated. When using workflows, we expect to automate not only the

¹<http://www.mygrid.org.uk>

²<http://industry.ebi.ac.uk/soaplab>

³<http://taverna.sf.net>

process of experiments but also the collection of experiment results and their provenance, as well as the provenance of other experiment resources (including the experiment design, the invoked services, etc). Collecting this rich source of knowledge capturing the history, explanation and justification for a wide variety of experiments requires a flexible and adaptable data model. Subsequently mining it and cross-linking it with other logs could lead to valuable insights but needs some very flexible integration and querying mechanisms.

Co-ordination of decentralised information is the aim of the Semantic Web (SW) [3]. The hope is that by giving information a well-defined meaning it should become accessible to both human and machine, encouraging the fruitful collaboration of both. Within *myGrid*, we have sought to exploit technologies provided by the SW community for supporting the tasks of service discovery and provenance knowledge management. We have made use of a large part of the available technology stack, from RDF to OWL-DL and reasoning technologies. We report here on our experiences of using these technologies and comment of the implications for the e-Science community as a whole.

In this paper, first we introduce the problems we face to and explain why existing tools and approaches are not appropriate for our problems in Section 2. Section 3 shows how *myGrid* tries to solve the service discovery problem when composing workflows and the provenance management problem when executing workflows using the Semantic Web technologies. Then we share our experience in Section 4 and conclude this paper in Section 5.

2 Heterogeneity in Bio e-Science

2.1 Heterogeneous Service Resources

Bioinformatics resources are unusually widely distributed, and highly heterogeneous. Moreover, bioinformatics lacks a formal or standard technology for structuring its data sets, often still relying on flat file technology. The practical result of this is that many services interfaces are poorly described by their WSDL, with highly, but informally, structured data being represented as `xml:string`.

The *myGrid* environment tries to simplify the task of accessing the heterogeneous biological data resources by integrating over a 1,000 publicly available services. Currently, a large number of these services come from external, autonomous service providers, with less than 5% being considered as *plain* web services. Other services consist of approximately:

25% Soaplab services Soaplab uses web services,

but exposes a stateful CORBA-like interface described later in this section.

30% BioMoby services The BioMoby project provides a registry and messaging format for bioinformatics services [19]. This is not described further, but again, imposes additional semantics over normal web service invocation.

30% Web based REST services The Seqhound [13] sequence retrieval system delivers its services through a Representational State Transfer (REST) style interface, where all the information that is required for the service invocation is encoded in a single HTTP GET or POST request.

10% workflows *myGrid* allows the incorporation of workflows into larger workflows.

This leaves the end user with a substantial problem in terms of selecting appropriate services for use, especially for the biologist who may not be highly skilled or knowledgeable about these services [18]. Even though a small number of stereotyped patterns can be identified from these different service families, it still is a challenging problem to build a unified *type* system inside *myGrid* determining the structuring of the data passed between services. As a result, our solution for integration causes new problems of interoperability.

Semantic Web Services augments standard Service Oriented Architecture with semantic descriptions of the services, in order to help agents (whether human or machine) interact with the service during its life cycle from service discovery, composition, execution to service monitoring. Considerable work in this community has resulted in technologies such as OWL-S [12] and, more recently, WSMO [1]. These technologies have focused largely on descriptions to enable automated composition of services. The requirement for full automation, and transparency of composition from the user perspective, has resulted in the application of extremely rich service descriptions.

While this automatic composition approach may be more appropriate in the B2C applications, it is less desirable within bioinformatics, where scientists are eager to participate in the process of services discovery and composition. Thus, *myGrid* narrows the semantic service descriptions for the sole purpose of discovery and takes the role of a human participant into account to be responsible for the final selections. In *myGrid* we have built a service discovery framework, called Feta [11], which supports a simplified, user-oriented data model for representing service descriptions. This abstract service description is supported by using an extensible workflow enactment engine, called Freefluo [14].

2.2 Heterogeneous Provenance Resources

Traditionally, in the wet lab environment, a scientist writes down large quantities of information in a lab-book recording how and why results were generated in an experiment process. This information serves as *provenance* of their experiments and results. By recording both raw and processed data, as well as how it was used and produced, provenance records become valuable resources to verify and reuse workflows and their results. By tracing the tools or services that a result was created by, and the version of the database that an experiment input was extracted from, provenance records can be used to cope with changes with databases or tools, to recover from experiment failures, and to explain the impacts of resource (databases, tools, etc) updates on the experiment results. Bioinformatics makes heavy use of inferential data. Knowledge gathered *in vivo* or *in vitro* about one protein, for example, is often transferred to other similar proteins. Combined with the informal data structuring as described earlier, this means that data circularity [9] and data staleness [20], are real problems without provenance.

Within bioinformatics much of this provenance data has been generated and stored by the expert curator, often as free text (such as the PubMed citations within UniProt) or loosely structured (such as the Evidence Codes within GO). This shows great limitations when scientists try to share their experiment results and inferred knowledge among their community in the e-Science environment. Because it offers standardized facilities for accessing the data, *myGrid* has enabled the automatic gathering of this provenance data in the form of *provenance logs*. Each piece of data used in an *in silico* experiment, the tools used to analyse this data, and the associations between other data needs can be recorded.

3 *myGrid* Solutions

3.1 Service Discovery by Feta

Feta is based on a complex domain model, built using OWL-DL, with the support of reasoning services. However, we have found that for querying and service discovery, RDF(S) provides a rich enough expressivity.

3.1.1 Feta data model

The key differences between this model and that present within OWL-S are those of omission: we have nothing in this model equivalent to either the ground-ing or process models and only a subset of the service

profile. This results from the constraints that bioinformatics presents and the support that other parts of the *myGrid* architecture provide for the invocation of services. This model also provides a few additional features which models the ideas from users about services, but which do not map to the underlying middle-ware layer.

The majority of the domain information in this data model is captured in the *myGrid* service ontology [21], which includes descriptions of the core bioinformatics data types (e.g. `DNA_sequence`), a characterization of the tasks commonly performed (e.g. `Protein_Analysis`) and a description of the biological entities being investigated (e.g. `homologue`).

The main entities captured in the Feta model are `Service`, `Operation` and `Parameter`:

- The `Service` entity encapsulates information only relating to *publication*. Our analysis of the bioinformatics web services suggested that, in most cases, a `Service` presented a set of operations providing related but independent functionality. Thus, within this data model we distinguish between the core unit of functionality, i.e. the *operation*, and the unit of publication, i.e. the *service*. The `Service` entity is described using information about the provider organization name, the author of the service description, and a free text description of the functionality.
- In general, a service may provide one or more service operations. Conventional web services with no state are good examples of this. These operations do not map directly to operations at the WSDL layer. From the users perspective multiple WSDL operations might provide only a single unit of functionality. Soaplab services, therefore, are all modelled as a service with a single operation. For other service styles, such as *myGrid* workflows, or Seqhound services, there is no underlying WSDL representation to map to. Freefluo helps to remove the difficulty of linking between the abstracted service descriptions and these different invocation layers.
- The capabilities of `Operations`, within Feta, are characterized by the inputs, outputs and several domain specific attributes. The inputs and outputs of an operation are modelled through the `Parameter` entity, and described using attributes such as its *semanticType*, *format*, *collectionType* and *configurationParameter*. Those domain-specific attributes of `Operations` describe their *tasks*, their underlying *methods*, the

application to which the service belongs, and the resource that the service uses.

3.1.2 Architecture

In this section, we give an architectural overview of the Feta discovery system, as shown in Figure 1. The key characteristic of this architecture is its relative simplicity: the core components communicate through web services; service descriptions are developed using XML and by applying generic XML tooling; querying is performed with Jena [5] using only RDF(S) entailment rather than DL reasoning. Feta is meant as a light-weight semantic search engine rather than a full service registry. Registry functionality is deferred to the standard web services registry, namely UDDI [2]. As shown in Figure 1, when a user interacts with the Feta architecture:

1. The user first automatically generates the publication information about a service by mapping from low-level descriptions of services to the more abstract, user-oriented descriptions. The absence of formal structuring for most bioinformatics data types, mean that the information which can be obtained from the services themselves is limited to: i) The service name ii) The names and number of service operations. iii) The names and number of operation parameters. As these documents contain the basic structure for the semantic service descriptions, but little of the information required, we describe them as *skeletons*.
2. Following the generation of skeleton documents, manual annotation of these documents is required to provide full descriptions. This annotation process takes time. In the first instance most descriptions have been developed by expert bioinformaticians from within the *myGrid* project. In our experience, the key difficulty has been poor documentation of the services, requiring experimental invocation of the service with test data. More recent experience with service publishing frameworks such as Soaplab, provide documentation directly associated with services which eases this process considerably.

It is clear that tool support is required for this process to encourage either external service providers, or service consumers to generate their own semantic service descriptions. To this end, we use the Pedro application [8]. This provides a GUI based interface which allows users to generate XML instance documents conformant to a given XML schema. The tool is also ontology aware and can provide easy access to the vocabulary at the point of use. Annotation is limited

to named classes rather than fuller class expressions.

3. Following the annotation phase, Feta descriptions are published, making use of a UDDI registry.
4. The Feta Engine engine then imports these descriptions, along with the RDF(S) version of the domain ontology, from where they can be queried. The decision to avoid the use of OWL and reasoning technologies at query time enables considerable architectural simplicity at this point. The Feta Engine is essentially a set of canned RDQL queries accessible via a web services interface. We currently use Jena [5] as our implementation backend as its query engine provides support for RDF(S) entailment. The canned queries that we currently support include:
 - An operation that accepts input of a given semantic type or something more general.
 - An operation that produces output of given semantic type or something more specific.
 - An operation that performs a given task, uses a method, uses a resource, or is part of an application, or something more specific.
 - An operation that is of type “WSDL based Web Service Operation”, “Soaplab Service”, “Scufl Workflow” etc.
 - An operation whose name/description contains a given phrase.
5. We provide a Feta plug-in for Taverna which is shown in Figure 2. The query interface enables the user to build a composite search query using the supported canned queries.

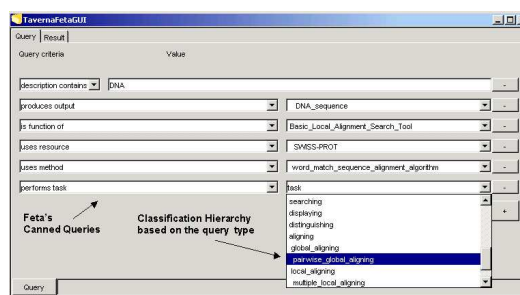


Figure 2: GUI Panel for Building Search Requests

Results of the search are then returned to the user in a results panel shown in Figure 3. Any additional information available about the service is also displayed enabling the user to make the final selection of the most appropriate service. These services can then be added to the workflow by

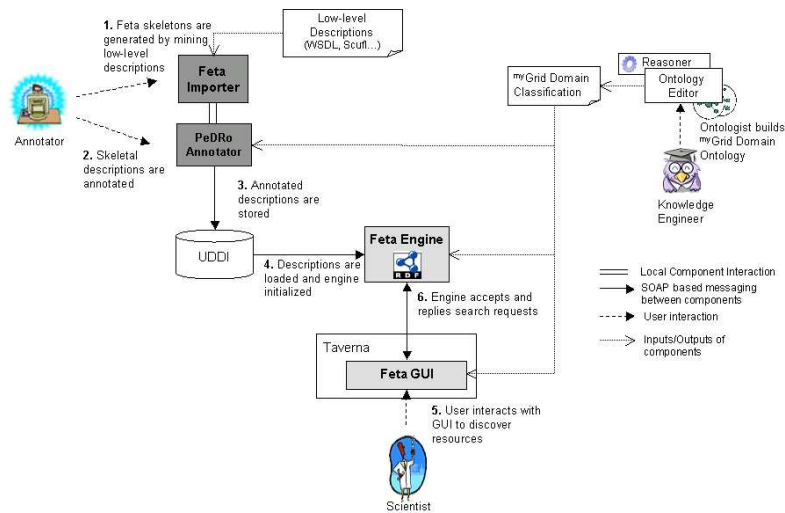


Figure 1: Architectural Overview of Feta

means of drag and drop. Currently, returned results are not ranked as most queries sufficiently narrow the total number of services from which the user can then select manually.

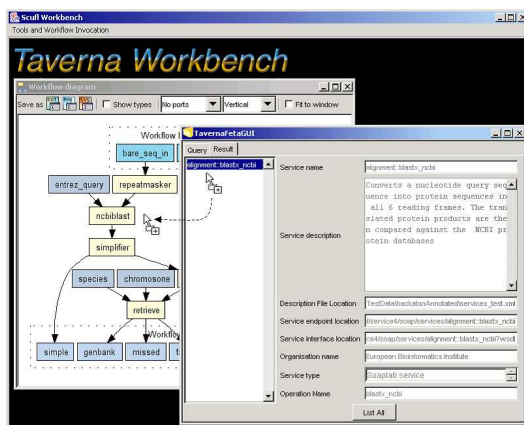


Figure 3: GUI Panel for Displaying Search Results

3.2 Provenance Management by KAVE

For provenance management, we have again largely used RDF based technologies [22]. *myGrid* offers standardized facilities for accessing the data to automate the gathering, preservation, archiving, viewing and using of provenance data, in the form of *provenance logs*.

3.2.1 Provenance Ontology

We used a mixture of RDF and OWL ontologies to represent this provenance model and to assist the

provenance collection, which provide two levels of semantics:

- a high level **schema ontology**, describing the classes of resource that can be linked together such as *ScientificDatum* and *ServiceInvocation* and the relationship that can hold between them such as *output*. RDF was used to describe the relationships. More recent work uses an OWL ontology;
- a **domain ontology**, classifying types of resource such as data type (e.g. BLAST report), service type (e.g. sequence alignment service).

3.2.2 LSID

In this work, we have greatly benefited from the adoption of Life Sciences Identifier (LSID) technologies[6] enabling us to use a common identifier both for data generated by *myGrid* and external resources. While we have stored the provenance in RDF form, the data itself is stored in a conventional relational database.

While this split between data and metadata is both technically appealing and necessary, it requires that some common mechanism exists to related between the two kinds of data. For this, *myGrid* has used LSID's. We could have used URL's or applied other additional semantics to a URI, but LSID's provided us with a well-defined mechanism for resolving identifiers into data and metadata. The use of LSID's is attractive because of the efforts to standardise the specification through OMG⁴ which has resulted in both freely available infrastructure support and promising

⁴Object Management Group, <http://www.omg.org>

increasing uptake within the domain. Finally, LSID's provide an explicit social commitment to the maintenance of immutable and permanent data: an LSID should always resolve to the same physical bytes of data, which is clearly an explicit requirement for storing of provenance data.

LSIDs provide a convenient access mechanism to the provenance of an object. Using the LSID metadata protocol, an object can serve the RDF triples that present its origin, which is a useful mechanism when objects are shared between applications or exported.

3.2.3 Architecture

The basic architecture for gathering this provenance is shown in Figure 4.

The provenance metadata is stored using RDF in the *my*Grid provenance repository (using Jena). This technology was chosen to represent the model because: i) It provides a more flexible, graph based model, as opposed to an XML tree; ii) It provides an explicit identification system (URI's) for resources which allow metadata to be merged from several sources; iii) It provides an well-defined association with an ontology; iv) From a practical point of view, there are several mature, open-source, repositories are available for use.

As well as suitable technologies for storing this data, we also need to present it back to the user. To date we have used the Haystack [16] browser. As well as natively understanding the LSID protocol, it provides us with convenient facilities for filtering the RDF graph which is generated. This is essential as a complex, highly-connected, RDF graph quickly becomes impossible to display and interact with. However, the visual complexity is daunting, suggesting multiple view mechanisms over RDF to be a necessity.

The real benefit of using RDF should come when we integrate and aggregate across the provenance of different workflow runs, and across different experiments. We should also be able to assert new claims over data results by grounding these against the provenance statements of workflows as the provenance record of a workflow is the "proof" (cf the Semantic Web language layer model) of its outcome. Testing these hypotheses is the current focus of our work.

This architecture also supports *using* provenance to perform tasks such as understanding experiment results or experiment failures. Manual use of provenance is error-prone and not very efficient. We focus on providing a set of methods for *slicing* and *aggregating* provenance to support life science research activities. A *Provenance Query and Answer (ProQA)* framework is provided for querying and integrating

provenance repositories implemented using Jena and Sesame [4] RDF APIs.

A three-layer design is proposed for ProQA: i) the bottom layer is an *Atomic Provenance Component*, which retrieves the provenance about each atomic experiment resource class defined in the schema ontology. ii) the middle layer is an *Aggregation Component*, which integrates the provenance about multiple experiment resources. iii) the top layer is an *Analysis component*, which provides direct support for each of actions taken by the end users. This ProQA framework aims to query and analyze provenance for provenance *users*. If developers for provenance analysis need to extend methods for this framework, they could extend the *Analysis* component by accessing the underlying RDF access APIs and the Core API provided in our implementation.

4 Experiences

Our experiences with the use of SW technologies within *my*Grid lead us to a number of conclusions.

Evolution not Revolution: Our application of SW to provenance and service discovery will better support the way biology is actually performed, rather than radically alter it. Although currently our applications are at a prototype stage, they seem to support our specific applications reasonably well.

Decision Support not Decision Making: In the short term, biologists will wish to monitor the results of SW technologies closely, until they fully trust it. With limited exceptions, we need to aid the users decision process, not replace it. For example, the Feta component is used to assist scientists to discover services based on their semantic descriptions during their workflow composition process; the ProQA component is used to analyze automatically collected provenance to assist scientists to understand their experiment results.

Tool Use not Tool Generation: Getting knowledge from users and presenting it to them is hard. There is a severe lack of user facing tools at the moment. This includes tools for the developer, the bioinformatician, and the biologist. Within *my*Grid, we have generated, or customized many tools ourselves (for editing ontologies, for maintaining and versioning ontologies, for generating annotation, for viewing). If SW is to be used widely within bioinformatics this barrier to entry must be lowered.

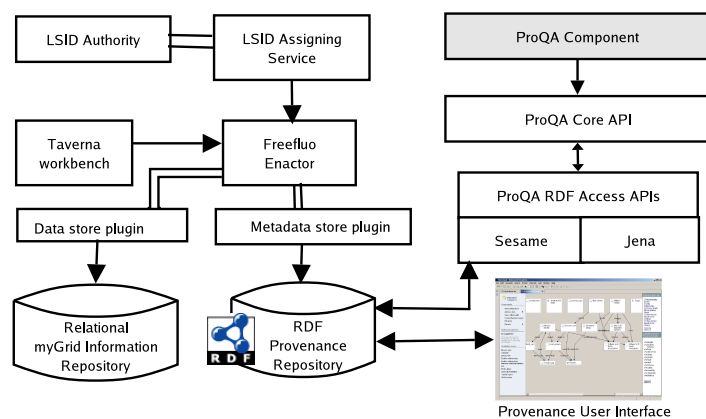


Figure 4: An architecture for provenance collection and visualization

Users vs Machines: Both service/workflow discovery and provenance management have highlighted the conflicting requirements of these two communities. On the one hand comprehensive models captured when publishing experimental components are desirable; on the other hand they are too complicated to be comprehensible to users. This suggests that view and filtering mechanisms over RDF graphs is crucial.

5 Conclusion

There are also a number of areas where we are less certain. **Scalability** has proven to be an issue for both of these applications, particularly with provenance data which will potentially be produced in huge quantities. Both of these applications use **mixed models**. Provenance data is stored partly in RDF, and partly in a RDBMS, while Feta makes use of both XML (for generating and storing service descriptions) and RDF (for querying). We have partitioned the data in a pragmatic rather than principled manner. Finally, whilst **aggregation** promises to enable common querying over data coming from a variety of different sources, especially for the provenance case, we have yet to convincingly demonstrate its utility with large scale “real world” examples, although we are encouraged by the experiences of Comb-e-Chem [7].

6 Acknowledgements

The *myGrid* project, grant number GR/R67743, is funded under the UK e-Science programme by the EPSRC. Discussions with other members of the *myGrid* team are gratefully acknowledged.

References

- [1] S. Arroyo, M. Stollberg, and Y. Ding. WSMO Primer. DERI Working Draft v01, 2004.
- [2] Tom Bellwood. UDDI Version 2.04 API Specification. UDDI Committee Specification, OASIS, July 2002.
- [3] Tim Berners-Lee, James Hendler, and Ora Lasila. The semantic web. *Scientific American*, May 2001.
- [4] Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema. In *Proceedings of The Semantic Web - ISWC 2002: First International Semantic Web Conference*, Sardinia, Italy, June 2002.
- [5] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: Implementing the Semantic Web Recommendations. Technical report, HP Labs, December 24th 2003.
- [6] T. Clark, S. Martin, and T Liefeld. Globally distributed object identification for biological knowledgebases. *Briefings in Bioinformatics*, 5(1):59–70, 2004.
- [7] J. Frey, G. Hughes, H. Mills, m. c. schraefel, G. Smith, and D. De Roure. Less is more: Lightweight ontologies and user interfaces for smart labs. In *The UK e-Science All Hands Meeting*, 2004.
- [8] Kevin L. Garwood, Chris F. Taylor, Kai J. Runte, Andy Brass, Stephen G. Oliver, and Norman W. Paton. Pedro: a configurable data entry tool for XML. *Bioinformatics*, page bth251, 2004.

- [9] Peter D. Karp, Suzanne Paley, and Jingchun Zhu. Database verification studies of SWISS-PROT and GenBank. *Bioinformatics*, 17(6):526–532, 2001.
- [10] Peter Li, Keith Hayward, Claire Jennings, Keith Owen, Tom Oinn, Robert Stevens, Simon Pearce, and Anil Wipat. Association of variations in nfkbie with graves’ disease using classical and mygrid methodologies. In *Proceedings of UK e-Science All Hands Meeting*, 2004.
- [11] Phillip Lord, Pinar Alper, Chris Wroe, and Carole Goble. Feta: A light-wieght architecture for user oriented semantic service discovery. In *European Semantic Web Conference*, 2005. ESWC 2005 :: 2nd European Semantic Web Conference 2005, Heraklion, Greece 29th May to 1st June 2005.
- [12] David Martin, Massimo Paolucci, Sheila McIlraith, Mark Burstein, Drew McDermott, Deborah McGuinness, Bijan Parsia, Terry Payne, Marta Sabou, Monika Solanki, Naveen Srinivasan, and Katia Sycara. Bringing Semantics to Web Services: The OWL-S Approach. In *First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, Lecture Notes in Computer Science. Springer, July 2004.
- [13] Katerina Michalickova, Gary D. Bader, Michel Dumontier, Hao Lien, Doron Betel Ruth Isserlin, and Christopher W.V. Hogue. SeqHound: biological sequence and structure database as a platform for bioinformatics research. *BMC Bioinformatics*, 3(32), 2002.
- [14] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R. Pocock, Anil Wipat, and Peter Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.
- [15] Tom Oinn, Mark Greenwood, Matthew Addis, Justin Ferris, Kevin Glover, Carole Goble, Duncan Hull, Darren Marvin, Peter Li, Phillip Lord, Matthew R Pocock, Martin Senger, Anil Wipat, and Chris Wroe. Taverna: Lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience*, 2005. Accepted for publication.
- [16] Dennis Quan and David R. Karger. How to make a semantic web browser. In *Proc. of the Thirteenth International World Wide Web Conference (WWW2004)*, pages 255–265. ACM, 2004.
- [17] R.D. Stevens, H.J. Tipney, C.J. Wroe, T.M. Oinn, M. Senger, P.W. Lord, C.A. Goble, A. Brass, and M. Tassabehji. Exploring Williams Beuren Syndrome Using *my*Grid. In *Bioinformatics*, volume 20, pages i303–310, 2004. Intelligent Systems for Molecular Biology (ISMB) 2004.
- [18] D. Tran, C. Dubay, P. Gorman, and W. Hersh. Applying task analysis to describe and facilitate bioinformatics tasks. *Medinfo*, 2004:818–22, 2004.
- [19] M. D. Wilkinson, D. Gessler, A. Farmer, and L. Stein. The BioMOBY Project Explores Open-Source, Simple, Extensible Protocols for Enabling Biological Database Interoperability. *Proc Virt Conf Genom and Bioinf*, 3:16–26, 2003.
- [20] Jonathan D. Wren. 404 not found: the stability and persistence of URLs published in MEDLINE. *Bioinformatics*, 20(5):668–672, 2004.
- [21] Chris Wroe, Robert Stevens, Carole Goble, Angus Roberts, and Mark Greenwood. A Suite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data. *The International Journal of Cooperative Information Systems*, 12(2):597–624, 2003.
- [22] Jun Zhao, Chris Wroe, Carole Goble, Robert Stevens, Dennis Quan, and Mark Greenwood. Using Semantic Web Technologies for Representing e-Science Provenance . In *3rd International Semantic Web Conference (ISWC2004)*, 2004.