The myGrid Ontology: Bioinformatics Service Discovery

K. Wolstencroft[1], P. Alper[1], D. Hull[1], C. Wroe[2], P.W. Lord[3], R.D. Stevens[1], C.A Goble[1]

[1]The School of Computer Science, University of Manchester, Manchester, M13 9PL, UK

[2]BT Global Services, London, UK

[3]The School of Computing Science,University of Newcastle upon Tyne, Newcastle upon Tyne, NE1 7RU, UK

E-mail: katherine.wolstencroft@manchester.ac.uk (corresponding author)

Abstract: In this paper we explore issues in the development of the myGrid ontology, which is an OWL ontology designed to support service discovery through service annotation. There are currently more than 3000 services offering programmatic access to bioinformatics resources. Composing these into workflows enables complex *in silico* experiments to be performed. These services, however, are highly distributed and heterogeneous, with inconsistent naming and descriptions, so service discovery and interpretation for the scientist is not only required, but also very difficult.
myGrid offers middleware to support *in silico* experiments in the Life Sciences, enabling the design and enactment of workflows as well as components to assist service discovery for workflow composition. The myGrid ontology is one component in a larger semantic discovery framework. We describe this framework and the issues that led to its development. Practical experiences have demonstrated that successfully exploiting the ontology is dependent not only on the coverage of the domain and the mode of constructing service descriptions, but also on the complexity of the discovery and annotation tools that accompany it. Here we describe the myGrid ontology and the way the exploitation of it has changed and diversified during the project. From an initial model of formal OWL-DL semantics throughout, we now adopt a spectrum of expressivity and reasoning for different tasks in service annotation and discovery. Here we discuss the implications of this and our experiences in semantic service discovery.
Keywords: Service discovery, annotation, OWL, Description Logic, RDFS, *in silico* Life Cycle

Bibliographical notes: Katy Wolstencroft is a Post-Doctoral Research Associate in the School of Computer Science at the University of Manchester. She received a BSc (Hons) in Biochemistry from the University of Leeds and an MSc and PhD in Bioinformatics from the University of Manchester. Her research interests include the application of ontologies to bioinformatics research and scientific workflows.

Pinar Alper works as a Research Associate in the School of Computer Science

at the University of Manchester. She received her BSc and MSc degrees in Computer Engineering from the Middle East Technical University, and her MPhil in Computer Science from the University of Manchester. She currently participates in the EU-IST project OntoGrid, and has participated in UK e-Science pilot project myGrid. Her research work focuses on the application of semantics in the areas of Web Services and the service oriented Grid.

Duncan Hull is a PhD student in Bioinformatics at the University of Manchester. He has a BSc in Plant Sciences and MSc in Computer Science also from the University of Manchester. His research is currently focusing on the use of reasoning to retrieve bioinformatics web services

Christopher Wroe currently works for British Telecom. He has an MD from the University of Cambridge and previously worked as a research fellow at the University of Manchester. His research interests include the use of ontologies to help represent life science resources

Phillip Lord is a Lecturer in computing science at the University of Newcastle upon Tyne. He has a BA (Hons) from Trinity College Cambridge and a PhD from the MRC Human Genetics Unit, Edinburgh. He previously worked as a Research Associate at the University of Manchester and his interests include knowledge representation, the semantic web and the semantic grid.

Robert Stevens is a senior lecturer in bioHealth Informatics at the University of Manchester. He has a biochemistry (hons) BSc (Bristol); biological Computation M.Sc. (York) and Computer Science D.Phil. York. His principle research interests are in the development and use of formal ontology to describe and analyse biological data.

Carole Goble is a Professor of Computer Science at the University of Manchester. Her research interests include the accessibility of information, particularly the use of terminological and ontological services for the representation and classification of metadata in a range of application domains. Recent work has been focused on two major areas: the Semantic Web and e-Science/Grids.

Introduction

In recent years, the bioinformatics community has embraced new techniques and technologies for analysing increasingly large, multi-dimensional data and for automating common and frequently repeated tasks. The use of Web Services and workflows are two examples of these technologies. The distributed nature of web services, however, produces the additional requirement for mechanisms for their discovery and interpretation, and to determine their invocation.

In the semantic web community, these issues are being addressed using ontologies. Ontology-aided web service discovery and composition are active areas of research. Initiatives such as OWL-S (Martin, Paolucci et al. 2004) and WSMO (Roman, Keller et al. 2005) have highlighted the importance of such tasks, and initiatives such as myGrid (Oinn, Greenwood et al. 2006) and BioMoby (Wilkinson 2004) have applied these techniques to the bioinformatics domain.

There are some 3,000 services (including web services) offering programmatic access to bioinformatics resources. The distribution and frequent lack of documentation, however, creates the requirement for easy service discovery. If services are available but unknown to the user, the advantages gained from using web service technology could be lost. This drives the need for the myGrid ontology of services (Wroe, Stevens et al. 2003). As the number of available services increases, this *semantic discovery* becomes more important.

In this paper, we describe the myGrid ontology which is designed to support web service discovery and composition in the bioinformatics domain. We describe the evolution of this ontology and the evolution of the ways we obtain service annotations for its exploitation. Our experiences show that ontologies are an essential component for service discovery and interpretation. Cost of obtaining service annotations to make use of the ontology is, however, high. The source and type of annotation vary greatly and these annotations exhibit a variety of requirements and capabilities. We have consequently developed a number of annotation strategies.

There are also various kinds of users of these annotations for discovery; from applications that support automated workflow composition to scientists with a range of sophistication and knowledge. Again, the annotations have to reflect this spectrum. Consequently, one size does not fit all. This impacts the nature, coverage and the expressivity of the ontology, and the technology used to express it.

1.1 Bioinformatics and myGrid

Bioinformatics is a relatively new discipline in biology and has emerged as a response to the exponential growth in the production of biological data. Much of this data can be stored electronically and is a rich source of research material for scientists. Bioinformatics encompasses computational and mathematical techniques for analysing, managing and storing this data and therefore performing *in silico* biological experiments.

The bioinformatics community is an open access one. Much of the data and many of the analysis tools are in the public domain, freely accessible over the internet. In the 2006 *Databases* special issue of *Nucleic Acids Research* there were over 850 different biological databases listed (Galperin 2006). Each one is a data resource available for bioinformatics, and many have associated analysis tools and algorithms, increasing the number of possible resources to several thousand. These resources have been developed over time by different institutions. Consequently, they are: distributed; highly heterogeneous; have few standards for data representation; and have few standards for data access. Therefore, despite the availability of resources, integration and interoperability present significant challenges to researchers (Davidson, Overton et al. 1995).

A typical bioinformatics experiment involves gathering data from multiple sources and performing a series of connected experiments on it. If the initial data set is small, this process can be managed by the scientist manually transferring data and results between resources. If the data set is large, however, as in the high-throughput experiments of microarray analyses or proteomics, this process becomes impractical and automated methods have to be employed.

A recent solution for automation is the use of Web Services (Curcin, Ghanem et al. 2005). Web Services enable programmatic access to remote databases and analysis tools, allowing the chaining together of distributed resources. The myGrid platform makes use of web services, and other services, and provides a suite of middleware components to support data-intensive *in silico* experiments in the Life Sciences. The flagship component is the Taverna workbench, which enables the design, enactment, and sharing

of workflows that link third-party bioinformatics web services (Oinn, Addis et al. 2004). To date, Taverna has been used in many areas of biological research, for example, in microarray analyses, proteomics, gene and protein characterisation pipelines, genotype/phenotype studies and many more (Li, Hayward et al. 2004; Stevens, Tipney et al. 2004). Figure 1 shows the Taverna workbench and a typical workflow experiment.

Taverna provides a solution to the problems of interoperation and automation, but the distribution and large number of web services creates the requirement for manageable service discovery.
When using services, scientists need to

- *Find Services* – what are they called, who hosts them and how do they access them;
- *Interpret Services* – what do the services do and what do the descriptions of services mean;
- *Know how to invoke them* – what data and initial parameters do they need to supply

Scientists generally know the methods or analyses they wish to use in an experiment, but they do not necessarily know what individual services are called, or where those services might be hosted. In order to address this problem, services need to be described with a common set of terms that describe their various attributes necessary for discovery. These will include, for instance, input, output and their biological task. These descriptions are delivered through annotations with a controlled vocabulary.
Annotations can be made with free text or a controlled vocabulary delivered either by a formal or informal ontology or classification scheme.

Free Text
In the Life Sciences, annotation by free-text has been standard practice. For example, the major primary sequence databases in bioinformatics, e.g. GenBank (Wheeler, Barrett et al. 2006) and UniProt (Wu, Apweiler et al. 2006) accept new DNA and protein sequences from the Life Science community tagged with free text describing the possible functions of the sequences. Unfortunately, this has lead to a large problem of inconsistency. The same sequence in different databases can be labelled with different functions, and with different names.

Controlled vocabularies
A controlled vocabulary is a commonly used form of ontology in the Life Sciences. It allows simple groupings of terms into a *taxonomy.* The Gene Ontology (GO) (Harris, Clark et al. 2004), for example, describes all gene products common across organisms. Annotating proteins from the sequence databases with Gene Ontology terms has gone some way towards alleviating inconsistencies in functional annotations, but expressing the relationships between terms in GO is still limited.

Formal Ontologies
Formalising a controlled vocabulary into a formal ontology enables relationships between terms to be described and exploited by humans and potentially also by machine reasoning. A formal ontology will allow the expression of taxonomic relationships together with properties of each term (or class). This means that terms can be defined computationally, rather than simply described as in the controlled vocabularies. A defined class has associated with it a collection of properties that are necessary and sufficient for membership of that class.

Languages for representing ontologies have been an active research area in computing for many years. The Semantic Web initiative, however, has stabilised the situation such that two languages have emerged as standards for encoding ontologies in a way that is machine processable on the Web.

The resource description framework (RDF) is a W3C recommendation for providing semantic content on the Web (Hayes, 2004). RDF is conceived as a set of triples in the form of subject, predicate or verb, object. The subject and object are resources, often Web pages, and the predicate links those resources. As well as Web pages, the resource can be literals such as words or phrases. In this way, RDF can be used to formulate vocabularies.

RDF Schema (RDFS) is a vocabulary that describes ontologies. It provides mechanisms for describing groups of related resources and the relationships between these resources. Relationships are constrained to be taxonomic relationships between classes (the isa relationship); typing resources to be a member of a class (the type relationship) and the domains and ranges of properties. Reasoning over an RDFS ontology is restricted to moving up and down the taxonomy tree to find more specific or more general examples of a class.

OWL-DL extends the expressivity of RDF (Smith et al, 2004). OWL-DL extends RDFS to allow for the expression of complex relationships between different RDFS classes and of more precise constraints on specific classes and properties, for example limiting the properties of classes with respect to number and type, the representation of one-to-one from many-to-one or one-to-many relationships and the ability to construct terms through the union and intersection of other classes.

At the core of OWL-DL lies a Description Logic (DL) that is a logical formalism amenable to automatic reasoning. This reasoning can be used to check the consistency of an ontology, to infer implied subsumption relationships – that is, "isa" relationships. This means that it is possible to infer that one an item with various properties is a member of a class. The latter mechanism can be used as a form of query, so that a description of a class can be reasoned over to find all the other descriptions that are subsumed by it. Consequently any resource, like a service, that is linked to that description is also found.

OWL uses the RDFS vocabulary as an export mechanism. RDF can be stored in specialist RDF stores and queried in a manner similar to a relational database.

In myGrid, semantic annotation is provided by using terms from the myGrid ontology (Wroe, Stevens et al. 2003), an OWL ontology. The coverage and expressivity of the ontology is paramount for the success of this technique. The cost and viability of obtaining the semantic annotations is the challenge. Services from the bioinformatics community often have little or no documentation and since they are not usually designed solely for use within Taverna, there is no incentive for service providers to supply annotations. The consequence of this is that each service has to be described, often from scratch, for use within the semantic discovery framework.

1.2 The Service Landscape

Before we can describe the myGrid ontology and the annotation strategies we employ in detail, we must first examine the types of service it will describe. To understand the appropriate and practical model for describing and discovering services, we need to understand the *in silico* scientific method undertaken by the Scientist when building and using their workflows and thus consuming services.

Industrial-Scale Service Providers
A large number of services are supplied by government-funded bodies, such as, the NCBI (United States) (Wheeler, Barrett et al. 2006), EMBL-EBI (Europe) (Cochrane, Aldebert et al. 2006) or DNA Databank of Japan, with responsibility to provide data to the global scientific community. These major providers generally have dedicated human resources for service development, API documentation, hosting and maintenance, but even between these large organisations, there is very little standardisation in the way web services are constructed and the way they are described.

Boutique Service Providers
The majority of service suppliers are "boutiques", individuals or small laboratories that have developed a tool, algorithm, database or workflow that they wish to share with others. The consequence of this is that many web services are independent of one another and are rarely designed to work together. Each might have different conventions for data input or outputs, which adds complexity to composing services into workflows.

1.3 Web Services and Other Services

There are over 3000 services available to myGrid. Of these, less than 5% would be considered plain web services. The currently supported types of services are:

- **'Plain' stateless Web services -** single web service operations that are described within WSDL (Web Service Description Language) documents. This class covers services from the major service providers such as NCBI; DDBJ and EMBL-EBI;
- **Stateful Soaplab services.** The Soaplab framework allows quick deployment of 'legacy command-line tools as web services (Senger et al, 2003);
- **BioMOBY Services**. The BioMOBY project (Wilkinson 2004) provides a registry and a messaging format for bioinformatics web services.
- **BioMart Services**. BioMart (Durinck et al, 2005) is an EBI data integration system that can be used by Taverna to interact with whole genome and other database resources.
- **Other Workflows.** Taverna workbench enables incorporation of previously designed workflows into others.
- **Local Java objects and scripts.** Taverna enables incorporation of local Java objects and beanshell scripts into the workflow design as operational steps.

From the perspective of the scientist, many of the differences between service types are hidden. Invocation details are buried in the Taverna workbench.

1.4 Service Types

Conceptually, there are two different types of web services used in myGrid; *domain services* and *shim services.*

Domain services perform a scientific function. These services are generally provided by third parties and cannot usually be altered or changed. For example, the GenBank, EMBL and BioMart databases and the NCBI_BLAST sequence alignment tool;

Shim services do not perform scientific functions. They are 'helper' services that are used to connect together domain services by, for example, changing the format of one output to something compatible with the input of the next (Hull, Zolin et al. 2006; Radetzki, Leser et al. 2006). The term 'shim' is derived from the engineering term shim, which is a thin, often tapered piece of material such as metal or wood used to fill in space between ill-fitting components.

For example, the shim service *'srs_links'* maps identifiers between databases, for example, a GenBank sequence ID to and EMBL sequence ID. Such a service might be required in a workflow linking an NCBI_BLAST service with a BioMart query of a genome database. NCBI BLAST returns sequence matches from the GenBank database, but the BioMart genome databases are based at the EBI and require EMBL sequence identifiers.

Unlike domain services, shims can often be created by the scientist designing a workflow. A rule of thumb for distinguishing a domain service from a shim service is that a workflow, when the shims services are invisible, is equivalent to the methods section of a scientific paper. If a service needs to be explicitly mentioned in the method, then it is not a shim. This distinction is an important one for describing services and affects the way scientists might discover them.

Building Workflows

Constructing a workflow in bioinformatics is analogous to designing a scientific experiment in a laboratory. Scientific method(s) chosen for laboratory experiments are either well established methods that have been subject to peer-review (i.e. the *best practice* for the task in hand), or they are ground-breaking new techniques which will be subject to peer-review upon the publication of results. The services in scientific workflows should also be considered in the same way. The completed workflow represents the scientific method in the experiment, so the choice of individual services to produce the workflow should consequently be in the control of the scientist. If there is more than one service that performs a particular analysis function, the scientists will need to investigate other factors, for example, the underlying datasets and their update strategies, to determine which service is the most suitable. Bioinformatics data changes daily, so any service that cannot accommodate this might not provide accurate results. This information might only be gained from individual scientist's experiences in the field. Consequently, automatically composing domain services into workflows in the bioinformatics domain is undesirable.

This is not the case for the entire workflow building process. Workflows are constructed in two stages.

1. **Assembly**: This first phase involves identifying and locating *domain* services which perform the scientific functions of the workflow;
2. **Gluing**: The second phase involves identifying how, and if, these services are interoperable. If consecutive services have incompatible outputs and inputs, *shim* services are required to join them together. If there is more than one shim service which performs the same function, choosing one over another cannot affect the

overall outcome of the experiment by definition. There is no scientific value of one shim over another, so automatic composition is desirable for this circumstance.

This two-stage workflow design process has a direct impact upon the methods employed to identify services and also on how we exploit the ontology. Automated composition requires the employment of the reasoning capabilities of OWL-DL, whereas providing a shortlist of suitable services does not. All that is required for this shortlist is querying the hierarchy of ontology terms. This can be achieved simply with RDFS and does not need anything more complex.

Consequently, for the deployment and exploitation of the myGrid ontology we make use of different representations and varying forms of reasoning:

- *For the initial service selection* phase an RDFS representation of the class hierarchy and simple RDFS reasoning suffices to describe and discover services.
- *For the detection of mismatches (incompatibilities)* between domain services and identifying suitable Shim services, we use the OWL based representation and OWL-DL reasoning.

By constructing the ontology in OWL and exporting to RDFS, we can support both of these activities.

2.1 The Semantic Annotation and Discovery Framework

Figure 2 describes different representations and their uses in the semantic annotation and discovery pipelines. There are four major components in the myGrid semantic framework:

1. The myGrid Service and Domain Ontology: describes the bioinformatics domain and the properties of services;
2. Annotation mechanisms and interfaces: allows services to be associated with ontology terms by a service curation team and by users and service providers;
3. Semantic discovery using the Feta semantic discovery tool which enables scientists to identify and interpret services. Here the annotations are consumed by a person – the scientist;
4. Mismatch detection mechanisms during workflow composition that automatically identify the requirement for shim services when incompatible domain services are connected in a workflow. Here the annotations are consumed by a piece of software machinery.

Points 3 and 4 effectively cause two parallel annotation pipelines, to reflect their two different requirements and different annotation consumers. The upper pipeline describes the process of annotation and discovery for scientists, and the lower one describes the annotation process for machine. Scientists use the descriptions to identify services and understand what they do. Machines use service discovery for *workflow match-making*, that is, identifying and substituting shim services between incompatible domain services. Richer descriptions are required for annotations by machines (described further in section 5), but in each case, the service annotation and ontology curation exercise is provided by expert curators.

In the remainder of this paper, we will concentrate on the ontology and the ways we can use it to annotate services. The other two components of the framework are consumers of these and are directly dependent on these strategies.

The myGrid Ontology

The purpose of describing services is to enable scientists to identify, interpret, and use them. The way that we describe the services is dependent upon how the descriptions will be gathered and how they will be used. Identifying a service is only the first part of the problem. The semantic descriptions of a service must allow a scientist to understand how they work, and also identify how to invoke them. To invoke a service the scientist must know the format of the input(s) it requires, and to combine services, they must also know the same for the outputs. The heterogeneity of the bioinformatics domain and a lack of standard formats mean that describing services with simple typing is impractical. Describing the syntactic interface does not provide enough information for the user to successfully invoke the service. In many cases, each input or output is just a string, and the question becomes, *what kind of string does this service require?* Therefore, the *semantic type* of each input and output must also be described

The myGrid ontology contains 710 classes and 52 properties. It describes the bioinformatics research domain and the dimensions with which a service can be characterised from the perspective of the scientist (Wroe, Stevens et al. 2003). Consequently the ontology is logically separated into two distinct components, the *service* ontology and the *domain* ontology. The domain ontology acts as an annotation vocabulary including descriptions of core bioinformatics data types and their relationships to one another, and the service ontology describes the physical and operational features of web services, such as, inputs and outputs.

3.1 The myGrid Service Ontology

The myGrid service ontology describes the physical properties of web services, for example, where they are located, or how many inputs and outputs they have. The myGrid model differs from those presented by related initiatives, such as OWL-S and WSMO in that it only describes these physical properties from the perspective of the scientist. In myGrid, the invocation layer of the service does not need to be described for the user as services are invoked using the Taverna workbench. The processor abstraction in Taverna conceals the invocation details from the user. In OWL-S, for example, the *grounding* and *process* sections of their ontology expose this complexity to allow for fully automated discovery and composition performed by un-attended software agents. myGrid omitted such descriptions to reduce the complexity of using the ontology to provide service annotations in the interests of scalability.

The major classes and their relationships in the myGrid service ontology are given in Figure 3. The core entity in the model is the Operation, which represents a unit of functionality (i.e. the service) for the user. Operations can be grouped into units of publication represented by a Service. An Operation has one or more input and output parameters. In turn, each input and output parameter has a name, a description and belongs to a certain namespace denoting its semantic *domain* type and domain-specific format. This abstraction of a service as an operation, similar to WSMO's tasks (Roman, Keller et al. 2005), means that we again avoid descriptions at an implementation level and keep them at the level of the scientist.

3.2 The myGrid Domain Ontology

The domain ontology describes the bioinformatics research domain. It describes the types of algorithms and data resources used, and the types of data that may be derived from, or used by, these resources

The following concepts cover the scope of the myGrid ontology:

- **Informatics:** captures the key concepts of data, data structures, databases and metadata. The *data* and *metadata* hierarchies in the ontology contain this information
- **Bioinformatics:** This builds on informatics. As well as data and metadata, there are domain-specific data sources (e.g. the model organism sequencing databases), and domain-specific algorithms for searching and analyzing data (e.g. the sequence alignment algorithm, *clustalw*). The *algorithm* and *data_resource* hierarchies contain this information.
- **Molecular biology**: This includes the higher level concepts used to describe the bioinformatics data types used as inputs and outputs in services. These concepts include examples such as, *protein sequence*, and *nucleic acid sequence*.
- **Tasks:** A hierarchy describing the generic tasks a service operation can perform. Examples include *retrieving*, *displaying*, and *aligning*.
- **Formats:** A hierarchy describing bioinformatics file formats. For example, *fasta format* for sequence data, or *phylip format* for phylogenetic data.

Figure 3 describes the service ontology model and shows a section of the hierarchy from the *bioinformatics_data* sub-ontology. Table 1 describes the number of classes in each of the sub-ontologies and the depth of terms.

The scope of the ontology is focused on supporting scientist-centred service discovery and interpretation. Each hierarchy contains concepts to describe the bioinformatics domain at a high level of abstraction. By combining the terms from the ontology, descriptions of services are constructed. For example, an NCBI_BLAST service would be described in the following way:

Overall **task** being performed by the operation. (i.e. biological operation)			*Local sequence alignment*
Bioinformatics **algorithm** used (i.e. underlying scientific method)			*NCBI_BLAST Similarity searching*
Data **resource** it accesses			*Entrez-Gene and Entrez-protein from NCBI-Genbank* (Wheeler, Barrett et al. 2006)
Number of inputs	*1*	**Input 1**	*Inputs – a fasta sequence of DNA or Protein*
Number of outputs	*1*	**Output 1**	*BLAST report*

By describing the domain of interest in this way, users should be able to find appropriate services for their experiments from a high level view of the biological processes they wish to perform on their data.

3.3 Annotation Using the Ontology

The myGrid ontology is used to annotate both domain services and shim services. The annotations are, however, treated in different ways. The annotation of shim services often requires much more detailed semantic annotation. For example, SRS links (shown in figure 4) is a shim service that maps between biological identifiers e.g. GenBank to EMBL identifiers. A description of inputs and outputs however is not sufficient to determine this. A large service registry, populated with many services and workflows,

might contain many services described as has having input GenBankID and output EMBL-ID. Each of these services or workflows might well perform very different tasks. The relationship between input and output enables the correct shim to be identified.

To consider our shim and domain service examples from previous sections, the annotations for each are shown in table 2.

Annotation and Discovery using the Ontology

Most of the services available to myGrid are owned by third parties and provided with no description of their function. In order to gain advantages from semantics, we must first gather the descriptions and annotate the services.

Three approaches were considered for providing service annotations for myGrid; annotation by domain expert curators, annotation by the scientists that use them, or annotation by those who supply the services.

Service Providers Supplying Annotations In myGrid, most of the services are owned by third-parties. They are not provided specifically for the project and they are independent of one another. We cannot impose conditions on the service providers supplying annotations unless we are willing to exclude services without annotation from the Taverna workbench.

Users Supplying Annotation Users are familiar with the services they have already successfully invoked, so it would be prudent to try and capture their accumulated knowledge. This may, however, result in popular services being annotated at the expense of more unusual services. Unfortunately, users are more likely to need extra assistance with finding and invoking unusual services. There may also be problems associated with a varying skill-set amongst the users. If anyone can supply annotations, how would we cope with poorly annotated services and would this cause problems with discovery?

Expert Curators Supplying Annotation Domain experts familiar with both the bioinformatics domain and the ontology model should be ideal for the annotation task, but obtaining people willing to build and maintain the description framework may be a difficult task. The main advantages of adopting this approach would be the ability to perform "just-in-time" annotation. If the ontology model was insufficient to describe a new type of service, for example, the domain expert could extend the ontology and supply the new annotation. Restricting those able to alter the ontology ensures consistency in the model, but it could also result in a bottleneck. For this approach to be viable, it has to be demonstrated that the bottleneck created by expert curation is outweighed by the quality of the annotations produced.

4.1 Comparing User and Expert Annotation

An experiment was conducted to compare the quality of user annotation to annotation by the domain experts. The domain experts in the study were three bioinformaticians involved in the myGrid project and also involved in the development of the domain ontology. The users in the study were a group of MSc bioinformatics students undertaking projects using myGrid.

The study compared annotations from the two groups by counting the number of 'missing' terms, which are defined as sections of annotation that have not been completed, and the number of 'superficial' terms, which are defined as sections of annotation that have been completed using general terms from non-terminal (root) nodes in the ontology. Both cases could result in there being insufficient annotation for invoking services, so it would limit the use of the semantics. The following tables describe the differences between the two groups of annotators in each case. For each service description, the annotation for the biological task, algorithmic method, underlying data resource and input and output parameters were assessed.

The data in table 3 clearly demonstrates there is a difference between the two groups of annotators and that expert annotators out-perform the users.

Superficial annotations are defined as services annotated with non-terminal nodes when more specific terms are more appropriate. For example, a service input that is described as consuming a biological sequence when it will only consume a protein sequence. A protein sequence is a type of biological sequence, so the annotation is not technically incorrect, but a DNA sequence is also a biological sequence. From this annotation, other users may wrongly assume that the service can consume either a protein sequence *or* a DNA sequence as well.

The results from the missing terms study are less clear-cut. For both groups of annotators, there were problems. Each service annotation is missing at least one type of annotation from each group. Table 4 shows these results in detail.

As the table shows, missing annotations occur in the expert and user annotated services. In the cases of input, output and method annotations, the expert annotators have proportionally more missing terms than the users. In the cases of Task and resource, there are very few from each group.

Closer inspection of the missing annotations against the ontology revealed that, in many cases, there were no suitable terms available in the ontology and therefore it did not have sufficient coverage to describe the bioinformatics domain in each category. Annotations involving the task and resource hierarchies seem to have greater coverage than the input, output and method hierarchies, which means that the basic biological functions and the databases that are commonly accessed were captured, but the specifics of particular data formats and analysis algorithms were not captured. As a result, the ontology's coverage had to be considerably extended. At the time of the study, the ontology contained 173 classes, now the ontology has more than doubled in size and contains 710 classes.

In cases where there were no suitable ontology terms for an annotation, the domain experts simply left the annotations empty, whereas the users tended to provide a general annotation from the non-terminal nodes of the ontology. In these cases, it could be argued that omitting annotations instead of supplying more general ones introduces less misleading information and also identifies problems with the coverage of the ontology. It could also be argued that having a curator for the service descriptions and ontology development would be the best way of ensuring consistency across service descriptions despite changes and advances in bioinformatics and the addition of new services. Having the same individual extend the ontology and supply the descriptions will mean a consistent application of terms, an in-depth knowledge of the ontology and services, and the ability to modify the ontology when required. It is this approach we have initially adopted in ^{my}Grid.

After a period of initial curator annotation, we anticipate that we will be able to open the annotation effort to service providers and suppliers. A large number of semantically

annotated services will demonstrate the added value of annotations to the community, so community engagement should increase. The initial testing of the ontology by the expert curator should reduce problems caused by omissions in ontology coverage, and the provision of annotation tooling to guide the annotation process, should dissuade *superficial* annotation. If this transformation can be made, the process becomes more scalable and sustainable.

4.2 Feta Semantic Discovery - Finding services

To assist the scientist in finding domain services, myGrid provides a lightweight semantic service selection system called Feta (Lord, Alper et al. 2005), which is a plug-in to the Taverna workbench. The objective of Feta is to provide a mechanism for searching over annotations and integrating the results into the workflow design environment. Feta has two components; a registry backend holding the annotations and a query user interface integrated into Taverna.

Feta allows RDFS based simple reasoning at the time of service discovery and operates over the user-oriented model of the myGrid ontology. Users construct service search requests through a simple user interface. Search criteria follow the domain ontology (i.e. bioinformatics task, data resource etc) with drop-down menus offering a choice of ontology terms to construct queries. The query interface of Feta is shown in figure 5.

The Evolution of the myGrid Ontology and Annotation

At present, the myGrid ontology contains 710 classes and 52 properties. The original myGrid ontology was written in DAML+OIL (a precursor to OWL, the Web Ontology Language) and was based on DAML-S (a precursor to OWL-S). As OWL was accepted by the W3C (World Wide Web Consortium) as a standard, the ontology was migrated from DAML-OIL to OWL. Both DAML+OIL and OWL were chosen for the myGrid ontology development because of the expressivity of the description logic language underpinning them and the ability to perform reasoning over the ontology (Stevens, Goble et al. 2002). This means that the ontology can be built in a logically consistent form in which the structure implied by the descriptions is complete. This form of the ontology can be exported in the RDFS format, which we have chosen for use in the Feta tool due to its lower overheads. The scope and coverage of the ontology has altered little since the design of the original prototype, but the use of the ontology in service descriptions and the use of the reasoning capability of OWL have evolved.

When the original design for the myGrid ontology was developed, the way in which we might exploit it to describe web services was an active topic of research. Initiatives such as OWL-S (formerly DAML-S) were just beginning and aiming to exploit the description logic reasoning of OWL (and DAML+OIL). The potential advantages from using these technologies were attractive, and it was a logical starting point for the myGrid ontology development. At the outset, the myGrid ontology was based on a subset of terms from the DAML-S ontology. The Taverna processor has an abstraction which hides service invocation details from the user, so this negated the need to describe the *service grounding* part of the DAML-S ontology, but the *service model* and *service profile* subsections formed the basis of the myGrid service ontology. These sections of DAML-S were also not adopted fully. The service profile contained many properties that involved

business processes (for example, cost) and geographical information (e.g. geographicRadius). Many of these properties are outside the scope of service discovery in bioinformatics because services are freely available and sometimes, there is only one service in the world which performs a particular biological task.

5.1 Reasoning over Annotations

We explored two ways to use the ontology to describe and identify domain services:

1. **The single description-whole service model**. The ontology is used to build a single complete description of a desired service. The service and domain ontologies are combined into one expression. This concept is used to *classify* the annotations and hence query the services. This approach requires complex OWL constructs to be formed and hence needs sophisticated interfaces to hide the complexity. On the plus side it enables automated service matching with complex OWL expressions. It is particularly useful for *decision making* when the emphasis is on precision, for example, a service must be decided upon by the workflow match-maker presented in section 2.1 and the bottom of figure 2.
2. **The service-domain controlled vocabulary model.** The service ontology acts as a service model and the domain ontology acts as a controlled vocabulary for the attributes of that model. Both are in RDF(S). Each attribute is reasoned over *independently* of the other. The terms are used to *query* the annotations. This has the advantage of simple querying, and only needs simple interfaces, like the Feta interface in Figure 5. Automated reasoning though is limited to each service property. This approach is useful for *decision support* when the emphasis is on recall, for example a shortlist of services can be inspected by a scientist, as in the top of figure 2.

At the outset of the project, method 1, the automated method was adopted throughout. Whilst technologically successful, this added complexity to both the interfaces and the expressions that needed to be constructed. Producing the OWL descriptions introduced a complexity that created a barrier to adoption for the user community. Current OWL-based knowledge acquisition tooling does not offer suitable interfaces which we can expose to end users. It was also unnecessary. In most cases it is sufficient to present the user with potentially suitable services and leave the final decision to their expert knowledge. The scientists are, after all, designing the workflows. Constructing a workflow is analogous to designing an experimental protocol, subject to peer-review and consequently in the control of the scientist. This requirement negates the need for complex but precise OWL-based reasoning. Thus decision support was all that was required for Feta-mediated discovery, presenting scientists with a shortlist of suitable services from which to select. Consequently, we moved to approach 2 of tagging services with terms from the ontology.

For shim services, the challenge was to identify when two domain services were not compatible and to find a suitable connector. We do need high precision in the results, and benefit from the full expressivity of OWL. This required no interface to the scientist but did require the formation of more complex queries. Therefore, method 2 decision making was more appropriate.

The distinction between the two forms of annotation for the two kinds of services has led to the development of the two different modes of annotation and ontology exploitation we introduced in Figure 2.

- The ontology is developed and maintained by a specialist ontologist;
- Most domain services are tagged with terms from the ontology by scientists or the service providers themselves using specialist tools that reduce flexibility but disguise the complexity and hide the knowledge representation paradigms beneath. Simple RDFS querying determines matches. The scientist then chooses from a shortlist;
- Key domain services and shim services are annotated by a specialist curator using expert tools with terms from the ontology, but the relationships between those terms is queried using the OWL-DL reasoning so that suitable shims can be automatically inserted into a workflow.

Table 5 describes semantic discovery techniques for services and workflows

5.2 Content Evolution

The content and coverage of the myGrid ontology has altered little from the original pilot version (Wroe, Stevens et al. 2003). The largest addition is the sub-ontology describing bioinformatics file formats. The ability to describe not only the semantic type of a biological object, for example, a protein sequence, but the format of that object, for example, fasta format, is essential.

Other changes have been the result of the ongoing annotation effort. The expert curator approach to obtaining descriptions of services enables the scope and coverage to be continually assessed. Describing real bioinformatics services, produced and owned by third-parties, highlights areas of the domain that have not been sufficiently described in the ontology, and the coupling of the curation of the ontology and the service annotations enables *just in time* ontology development, i.e. extensions to the ontology are made as and when required during the annotation of a service. The advantage of this is the intricate knowledge the curator has of both the ontology and the service annotations.

To reduce the risk of missing resources and resource types in the ontology, we have been investigating the use of external collections upon which to base the ontology descriptions. One such source is the list of resources produced by the Nucleic Acids Research databases edition (Galperin 2006). This collection lists all *published* biological data resources. Whilst this does not address the separate issue of algorithms and analysis tools, it reduces the risk of being unable to recognise or describe a database identifier or data format.

Discussion

Ontology aided service discovery is a practical solution for the myGrid project and the ontology remains the largest currently available ontology of bioinformatics services. In fact, the myGrid service ontology is now also being used by BioMoby (Wilkinson 2004) after some feedback and contributions from BioMoby, another major service oriented project in bioinformatics. Service discovery is an important consideration in distributed systems and the need for architecture to support this is clear. Semantic Web technologies, such as OWL, OWL-S and RDFS lend themselves to these tasks, but the choice between

them is dependent upon whether the services will be discovered and composed by humans, or automatically by machines, and on the types of services to be discovered.

We have shown that, in practice, *automated discovery and composition requires* fine-grained descriptions, ontology reasoning and ranking of discovered matches, whereas *scientist mediated discovery requires* course-grained matches and short-list of services for the scientist to investigate further.

Intuitively, automated discovery and composition appear desirable, but in the context of *in silico* biological experiments, we have found this not to be the case. Scientists do not wish to relinquish control of the experimental design process, and automatically selecting services for a workflow would effectively cause this. When experimental (domain) services are not compatible, however, finding connecting shim services automatically *is* useful and desirable. For this reason, we have adopted different formalisms and different formats for the ontology and for querying it.

The increased expressivity required for shim annotations, to allow automated addition, adds cost to the annotation effort and expressly requires a domain expert curator, but the benefits to the users outweighs this.

Our experiences have shown that the myGrid ontology is the central and most important component of the service discovery framework, but it is not the most challenging to provide. Obtaining service annotations is a much greater difficulty. Even when using the ontology terms as a description skeleton, the amount of expert knowledge required to annotate services cannot be underestimated. Expert curation has been a vital part of the process. It is time-consuming and it produces a description bottle-neck, but the added value of both the annotations and the ability to improve and extend the ontology with experience compensates for this. For the future however, we must find a scalable alternative. When a critical-mass of services have been annotated, we postulate that the service provider and consumer community will engage. For this to be successful, the annotation interfaces offered to the community must be simple and must guide the user through the annotation process. Community annotators must not be exposed to any of the complexities of OWL, so development of simple interfaces that 'hide' the ontology are now under development.

In contrast, the myGrid ontology will remain a curated activity. Community authoring of OWL-DL ontologies can introduce inconsistencies, and may result, for example, in different scientists extending it in different places to describe the same things. The myGrid ontology is not large, but the complexity is such that extensions should only be made by bioinformaticians with ontology development experience.

In a distributed environment, even when using human-mediated discovery, services have to be located in a uniform fashion, and when attempting to use them together, the meanings of the annotations also have to be uniform. Also the scalability of providing annotation requires a restricted vocabulary. Each of these things can be achieved with a controlled vocabulary, but for any kind of automation, the requirement for expressivity increases. The advantage gained from automatically detecting workflow mismatches justifies the added cost of using OWL reasoning.

Our experiences have shown that using an ontology for service discovery is not a luxury, but a requirement.

Acknowledgments

We acknowledge the support of the EPSRC through the myGrid (GR/R67743/01, EP/C536444/1, EP/D044324/1, GR/T17457/01) e-Science projects. We

also thank Mark Wilkinson and the BioMoby team for their contributions to the ontology development effort.

References

1 Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P, editors, The Description Logic Handbook.. Cambridge University Press, 2003.

2 Cochrane, G., P. Aldebert, et al. (2006). EMBL Nucleotide Sequence Database: developments in 2005. Nucleic Acids Res **34**(Database issue): D10-5.

3 Curcin, V., M. Ghanem, et al. (2005). Web services in the life sciences. Drug Discov Today **10**(12): 865-71.

4 Davidson, S. B., C. Overton, et al. (1995). Challenges in integrating biological data sources. J Comput Biol **2**(4): 557-72.

5 Durinck S, Moreau Y, Kasprzyk A, Davis S, De Moor B, Brazma A, Huber W BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. Bioinformatics. 2005 Aug 15;21(16):3439-40.

6 Galperin, M. Y. (2006). The Molecular Biology Database Collection: 2006 update. Nucleic Acids Res **34**(Database issue): D3-5.

7 Harris, M. A., J. Clark, et al. (2004). The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res **32**(Database issue): D258-61.

8 Hayes P, Editor RDF Semantics, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

9 Hull, D., E. Zolin, et al. (2006). Deciding matching of stateless services. Twenty-First National Conference on Artificial Intelligence (AAAI'06), Boston, MA, USA.

10 Li, P., K. Hayward, et al. (2004). Association of variations on I kappa B-epsilon with Graves' disease using classical and myGrid methodologies. 3rd UK e-Science All Hands Meeting, Nottingham UK.

11 Lord, P., P. Alper, et al. (2005). Feta: A light-weight architecture for user oriented semantic service discovery. 2nd European Semantic Web Conference, Heraklion, Greece, Springer-Verlag.

12 Martin, D., M. Paolucci, et al. (2004). Bringing Semantics to Web Services: The OWL-S Approach. First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, California, USA.

13 Oinn, T., M. Addis, et al. (2004). Taverna: A tool for the composition and enactment of bioinformatics workflows. Bioinformatics Journal **20**(17): 3045-3054.

14 Oinn, T., M. Greenwood, et al. (2006). Taverna: Lessons in creating a workflow environment for the life sciences. Concurrency and Computation: Practice and Experience(Special Issue on Scientific Workflows).

15 Radetzki, U., U. Leser, et al. (2006). Adapters, shims, and glue--service interoperability for in silico experiments. Bioinformatics **22**(9): 1137-43.

16 Roman, D., U. Keller, et al. (2005). Web Service Modeling Ontology. Applied Ontology **1**(1): 77-106.

17 Senger M, Rice P, Oinn T, Soaplab - a unified Sesame door to analysis tools pages 509-513, Proc UK e-Science All Hands Meeting 2003

18 Smith MK, Welty C, and McGuinness DL, Editors, OWL Web Ontology Language Guide, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owl-guide-20040210/

19 Stevens, R., C. Goble, et al. (2002). Building a bioinformatics ontology using OIL. IEEE Trans Inf Technol Biomed **6**(2): 135-41.

20 Stevens, R., H. J. Tipney, et al. (2004). Exploring Williams-Beuren Syndrome Using myGrid. 12th International Conference on Intelligent Systems in Molecular Biology, Glasgow, UK, Bioinformatics

21 Wheeler, D. L., T. Barrett, et al. (2006). Database resources of the National Center for Biotechnology Information. Nucleic Acids Res **34**(Database issue): D173-80.

22 Wilkinson, M. D. (2004). BioMOBY - the MOBY-S Platform for Interoperable Data Service Provision. Computational Genomics Theory and Application. R. P. Grant. Wymondham, U.K, Horizon Bioscience.

23 Wroe, C., R. Stevens, et al. (2003). A suite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data. International Journal of Cooperative Information Systems **2**(2): 197-224.

24 Wu, C. H., R. Apweiler, et al. (2006). The Universal Protein Resource (UniProt): an expanding universe of protein information. Nucleic Acids Res **34**(Database issue): D187-91.

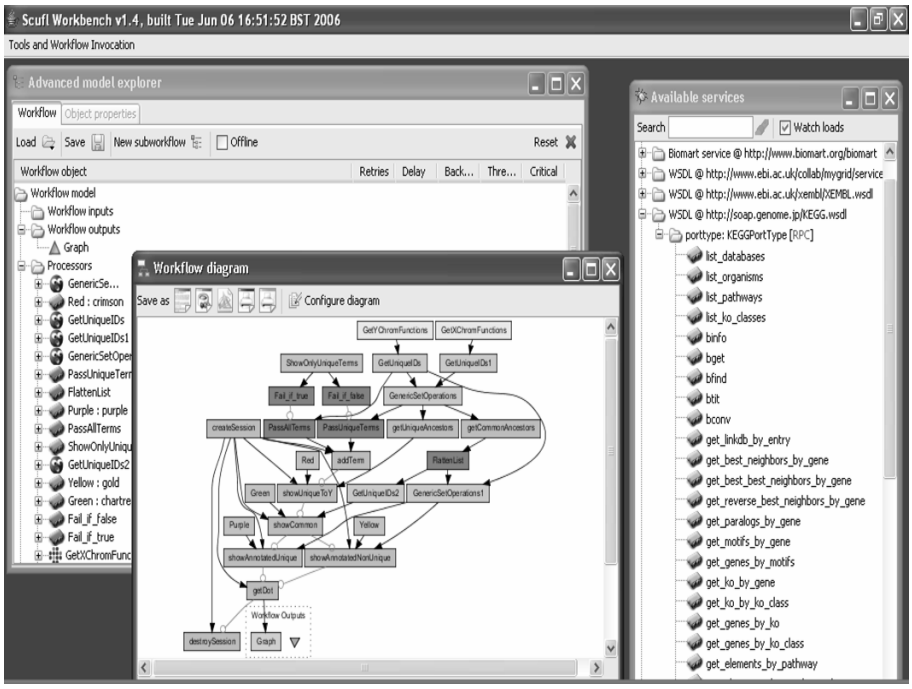

Figure 1 The Taverna workbench displaying three of its panels, the advanced model explorer, the workflow diagram and the services pallet. Users can construct workflows by dragging and dropping available services into the advanced model explorer and connecting inputs and outputs. The workflow diagram shows these connections and the flow of data between services

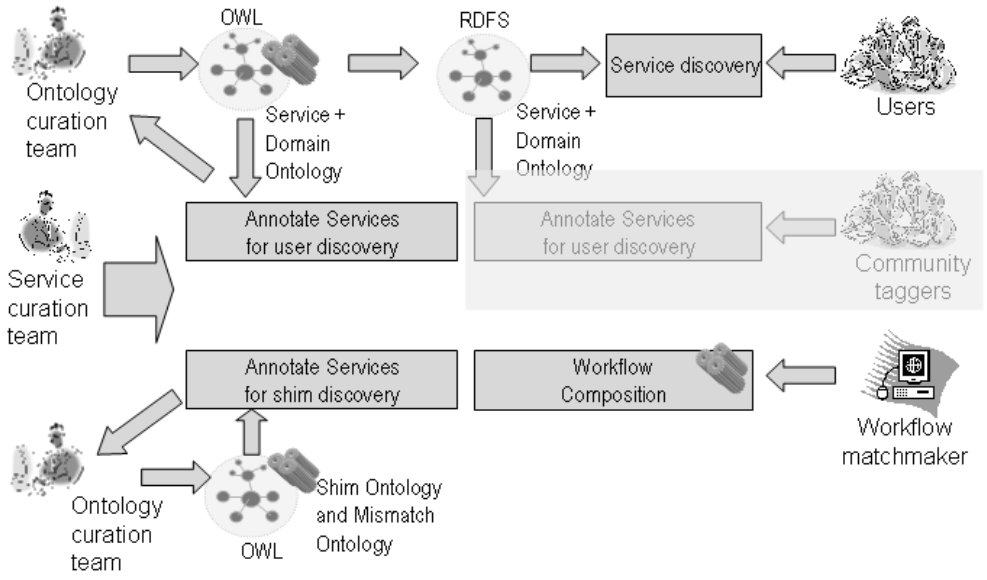

Figure 2 The Semantic Annotation Pipelines in myGrid. The Upper pipeline describes annotation for scientists. Expert curators construct the ontology in OWL and provide the annotations. The OWL ontology is exported to RDFS and simple interfaces enable RDFS-based searching by scientists. The lower pipeline describes annotation for machine-mediated workflow composition by the workflow matchmaker. All annotation is mediated by expert curation, but for sustainability and scalability, community annotation will be encouraged in the future.

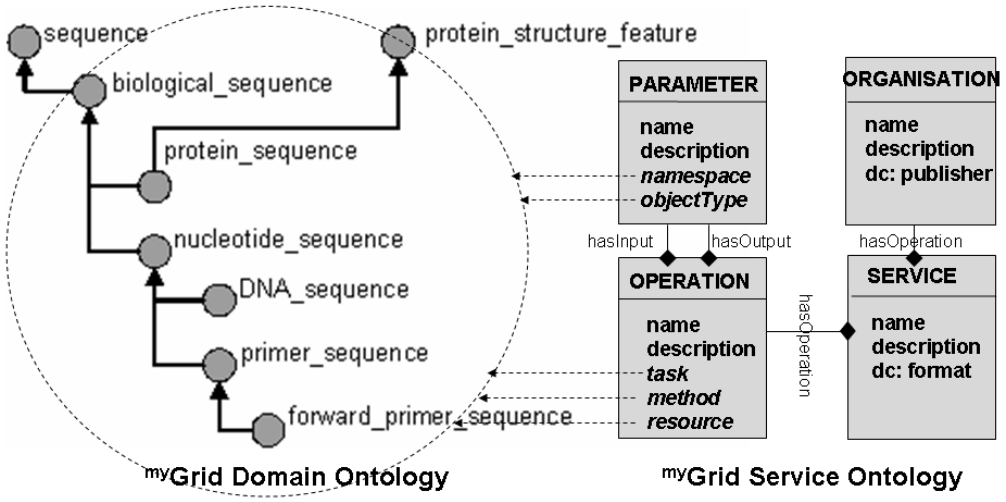

Figure 3 The [my]Grid service ontology model and an example of the relationships between data types in the domain ontology

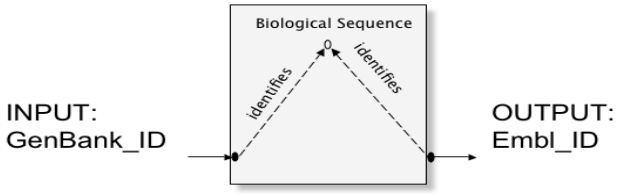

Figure 4 In shim service annotation, the relationships between input and output are important. In this case, the relationship between input and output is that the input GenBank identifier and output embl_identifier identify the same biological sequence. Other services with GenBank ID input and EMBL ID output, might have different functions, which means the resulting sequence might *not* be the same as the first.

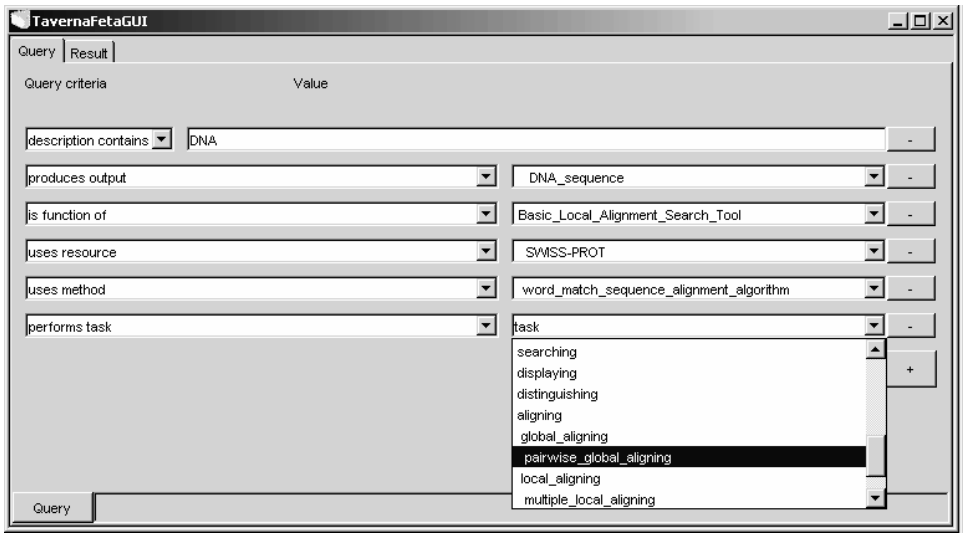

Figure 5 The Feta user interface enables the discovery of services by scientists. The left column is populated by concepts selected from the Service Ontology, consequently populated by concepts drawn from the Domain Ontology (on the right).

	Number Of Classes	Depth of Classes
Bioinformatics Algorithm	21	3
Bioinformatics Data	153	5
Bioinformatics Metadata	87	5
Bioinformatics Task	28	4
Bioinformatics Data Resource	319	4
Bioinformatics File Formats	21	4
Service	26	3

Table 1 The contents of myGrid ontology

	Task	Algorithm	Resource	Input(s)	Output(s)	Relationship
BLAST	Local Sequence Alignment	BLAST	NCBI-GENBANK	1.DNA Sequence 2.GenBank DNA database 3. BlastN	BLAST report	N/A
Blast Extractor	parsing	N/A	BLAST-Report	BLAST Report	(list of) Genbank IDs	Input sequence has similarity to output sequence
SRS Links	mapping	N/A	EMBL	GenBank ID	EMBL ID	Input and Output identify the same sequence
BioMart human genome (ensembl)	retrieving	N/A	Human ENSEMBL	EMBL ID	Affy Probeset IDs	N/A

Table 2 Annotations for domain and shim services. These annotations are taken from existing services, but are simplified for presentation here.

Number of Superficial Annotations	User Annotation	Expert Annotations
Input	13%	0%
Output	13%	0%
Method	14%	0%
Task	1%	0%
Resource	33%	0%

Table 3 Percentages of superficial annotations

Numbers of Missing Annotations	User Annotation	Expert Annotations
Input	28%	65%
Output	29%	69%
Method	84%	89%
Task	17%	20%
Resource	17%	20%

Table 4 Percentages of Missing Annotations

Discover	**Ontologies**	**Mechanism**	**Example query**
Domain service	Domain service	RDF(S)	Which services take in DNA sequence and produce a protein sequence?
Service mismatch	Domain service + mismatch	RDF(S)	Do RepeatMasker and NCBI BLASTN fit together?
Shim service	Domain service + shim	OWL	Which services convert given DNA sequences between different syntaxes e.g. FASTA to NCBI?

Table 5 Semantic discovery techniques for services and workflows