Recycling Services and Workflows through Discovery and Reuse

Chris Wroe[1], Phillip Lord[1], Simon Miles[2],
Juri Papay[2], Luc Moreau[2], Carole Goble[1]
[1]Department of Computer Science
University of Manchester
Oxford Road
Manchester M13 9PL, UK
and
[2]School of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ, UK
[cwroe,plord,carole]@cs.man.ac.uk,
[sm,jp,l.moreau]@ecs.soton.ac.uk
http://www.mygrid.org.uk

Abstract

Workflows are a central component for representing e-Science procedures in *my*Grid. For *my*Grid to support their design, scientists must be able to discover appropriate services to orchestrate and also discover if colleagues have already designed something similar. *my*Grid integrates a number of software components to address these requirements. The *my*Grid registry stores service and workflow descriptions. PeDRo, a structured data entry tool, enables uses to annotate these descriptions. Taverna, the workflow workbench, closely integrates with the registry and PeDRo to ensure description and reuse of services and workflows is simple.

1 Introduction

*my*Grid supports the e-Scientist in managing and performing *in silico* experiments in biology. Web and Grid Services provide access to distributed resources whilst workflow techniques provide for the orchestration of these resources. Workflows enable the e-scientist to describe and enact their experimental procedures in a structured, repeatable and verifiable way [1]. However, a key challenge lies in supporting the rapid assembly of these workflows from disparate services, and their re-use in various scenarios. This challenge places additional requirements on *my*Grid infrastructure:

- Provide access to information on available services and associated workflows.
- Provide effective search of that information.
- Provide effective reuse of discovered services and workflows.

This paper will describe the workflow design life-cycle, the model we have developed for describing workflows and services, and then describe how specific *my*Grid components address these requirements through the various stages of the life-cycle.

2 Workflow design life-cycle

The *my*Grid project considers an experimental life-cycle that extends beyond its execution to include its design and publication for others to use. Before embarking on workflow design the author should consult a catalogue or *registry* of previously published workflows. Search facilities must exist to identify any existing workflows that perform a similar task and so can be used 'as is' or require slight modification. Once found it must be easy to transfer this workflow into a workbench for further editing and execution. If modifications require the use of additional or alternative services, the author must again be able to search for services that perform the required task. These too must be easy to integrate into the workflow design. Once the workflow has proved its worth it must be a simple task to publish so that others in the organisation can benefit. The author also has additional knowledge on the suitability of the original workflow for this task. It must also be possible for him to go back and annotate the original workflow with this experience.

3 The ^{my}Grid descriptive model for workflows and services

Reuse can only we achieved if there is a catalogue or registry of existing workflows and services. Each entry must be assigned some description to drive indexing and search. There are several options. Free text provides the most flexible mechanism for users to describe the nature of the service, but is opaque to both middleware and applications which cannot therefore provide support for reuse. Structured descriptions are therefore more desirable, but are more difficult to author by users, and can be frustrating if a service or workflow doesn't quiet fit the model. Existing standardisation efforts for service description include:

- Universal Description Discovery and Integration (http://www.uddi.org/) standard (UDDI)
- Ontology Web Language Services ontology (OWL-s) (http://www.daml.org/services)
- Web Services Definition Language (WSDL) (http://www.w3c.org/2002/ws/desc/).

However, within the e-Science context of ^{my}Grid we have found that to support reuse, structured descriptions must have the following properties, which are not necessarily addressed by these standards.

User centric These descriptions are to be browsed and searched by users and so must be in a form and use terminology understandable to users. WSDL documents are intended to provided a programmers level interface description for a web service. They are unintelligible for users and it is wholly inappropriate to present them with such a description. UDDI has a highly generic model of services designed to cope with a wide scope of services from the local florist to a genomic database. We have found it difficult to use such a generic model ”as is” for describing bioinformatics *in-silico* experimental resources in a manner that users can comprehend.

Operation focussed The primary aim of these descriptions is to find resources that can either be included as an operational step within a workflow, or are a workflow in their own right. UDDI's key entity is the service and makes no commitment to the description of operations provided by that service. In fact convention often delegates this task to an associated WSDL document.

Data centric The overwhelming majority of bioinformatics service operations used within ^{my}Grid go to form data pipeline workflows. Therefore a key distinguishing feature of an operation is the nature of the data flowing in and out. WSDL describes data from the bottom up often specifying data as programming types such as `String`. Users actually want to search top down, first on data's conceptual content such as `Protein Sequence`, and only then on any formatting or typing issues.

Technology independent Within ^{my}Grid different types of operation can be included as a step within a workflow, including another workflow, a web service operation as described by a WSDL document, a Soaplab service [5], a bioMoby service [6] (both using additional conventions for using WSDL), or a local fragment of Java code. Any description must therefore be able to abstract the key attributes shared by these resources.

For workflows we use the workflow language Scufl to describe the control and dataflow between its component operations[4]. It's primary aim is to provide a formal specification which can be run by a suitable workflow enactor such as FreeFluo (`http://freefluo.sourceforge.net`). As in the case of services, it is useful to have an additional high-level description which caters for user-centric search, and browsing.

In ^{my}Grid we have developed a user-centric model of services and workflows that focusses on their functionality in terms of operations and nature of data. This model can be used in parallel with UDDI, WSDL and Scufl as it provides additional annotation rather than overlapping information.

Figure 1 shows an overview of the model. The key entities are:

Abstract service This is the unit of *publication*. It's fields describe who published this service, what organisation they belong to; together with a free text description of the service. The service may often provide more than one operation. This is the case with many WSDL described web services. Therefore functionality is described using a separate entity the operation.

Operation This is the unit of functionality. To address user centric requirements the entity has four fields to describe high level attributes such as the overall task being performed (e.g., aligning); the method used to perform that task (e.g., an algorithm such as Watermann); the type of application used to provide the functionality (e.g., Basic Local Alignment Search Tool BLAST); and finally any static resource used to providing the functionality (e.g., a background database such as the Genome database Genbank).

Parameter A key distinguishing feature of many operations in ^{my}Grid workflows is the type of data

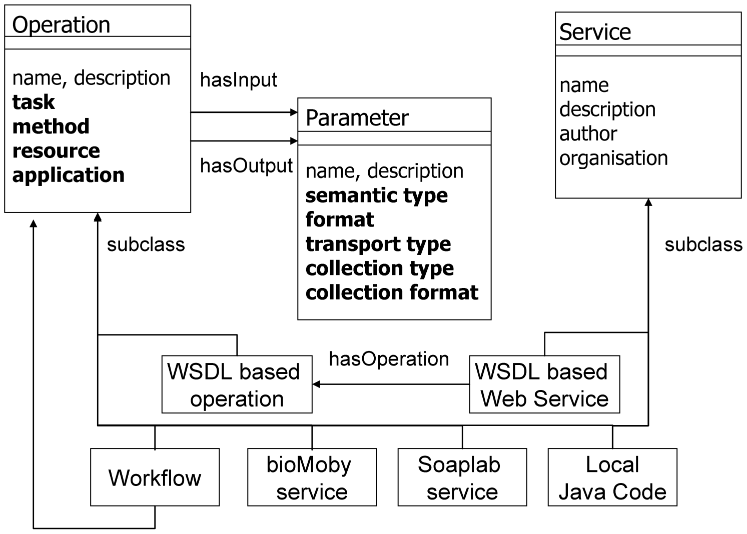


Figure 1: This shows the conceptual model of workflows and services within *my*Grid for the purpose of discovery. Fields whose values are filled with ontology concepts are shown in bold.

flowing in and out. In *my*Grid, we use the collective term *parameter* for these data types used or produced by an operation. Parameters can be described at several levels from a high level conceptual description such as ”protein sequence”, through formatting descriptions such as ”FASTA format” to low level types such as ”String” described in WDSL interface documents.

Figure 1 shows that currently, for the purposes of discovery, a workflow is modelled as an operation, with each of its individual steps seen as more atomic internal operations. Control and data flow is not represented as this would replicate information in the main Scufl workflow file. Although each major entity can be described in free text, the majority of fields are intended to be filled by terms provided by an ontology.

4 *my*Grid component overview

*my*Grid supports the design life-cycle by developing or integrating a number of middleware services and user components. Service Registries (built within *my*Grid) provide a searchable store of service and workflow descriptions. These searches are augmented by additional indexing and query services using ontologies and ontological reasoning to provide domain dependent knowledge. PeDRo, an ontology aware data entry tool built outside the *my*Grid project, provides users with the ability to add structured metadata to each registration. Plug-in components for the Taverna workflow workbench (built within *my*Grid) closely integrate workflow publication and discovery with workflow design and execution.

4.1 PeDRo: Ontology aware data entry

Providing rich metadata is often an altruistic activity and so it must therefore be as easy as possible to enter such metadata. The *my*Grid project uses PeDRo (`http://pedrodownload.man.ac.uk/`) to allow users to enter descriptions of services and workflows for publication into the registry. It allows users to enter structured data or metadata based on a predefined XML schema. It has intrinsic support for ontologies, which can be configured to provide the vocabulary for specific data fields. The focus is to make use of a controlled vocabulary straightforward. When used within *my*Grid it is configured with an XML Schema derived from the conceptual model described in section 3. The user can describe a workflow or service by simple form filling. Figure 3 shows a form for a bioinformatics workflow ready for user input. Many values are provided by concepts from the *my*Grid ontology. The ontology[1] is currently developed in the OWL language using ontology editors such as OilEd [2] and Protégé [3]. The expressivity of the OWL language (`http://www.w3c.org/2004/OWL/`) allows for the formal representation of rich relationships between concepts and subsequent description logic reasoning. A fully descrip-

[1] Available from `http://www.mygrid.org.uk` under the ontology service component page.

[2] `http://oiled.man.ac.uk`

[3] `http://protege.stanford.edu`

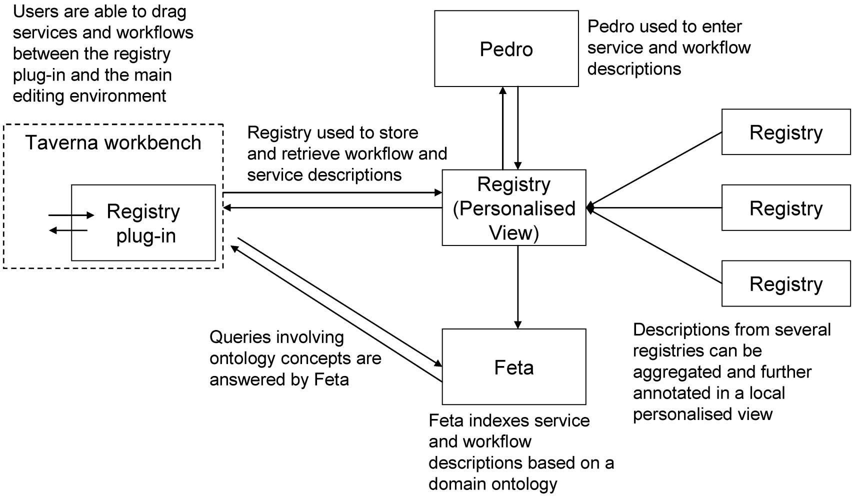

Users are able to drag services and workflows between the registry plug-in and the main editing environment
Taverna workbench
Registry plug-in
Registry used to store and retrieve workflow and service descriptions
Pedro
Pedro used to enter service and workflow descriptions
Registry (Personalised View)
Registry
Registry
Registry
Queries involving ontology concepts are answered by Feta
Feta
Feta indexes service and workflow descriptions based on a domain ontology
Descriptions from several registries can be aggregated and further annotated in a local personalised view

Figure 2: Architecture of workflow /service discovery components in ^{my}Grid

tion of the ontology design within ^{my}Grid can be found in Wroe et al [7]. Currently we make use of reasoning during construction and maintenance of the ontology, *not* during description of a workflow/ service or during query. Therefore the hierarchical structure of the OWL ontology is exported in the simpler RDFS (Resource Description Framework Schema) language and made available to PeDRo. PeDRo presents the user with an ontology browser from which the user can choose the appropriate concept. The structured description is then stored in the registry and available for query.

4.2 Registry

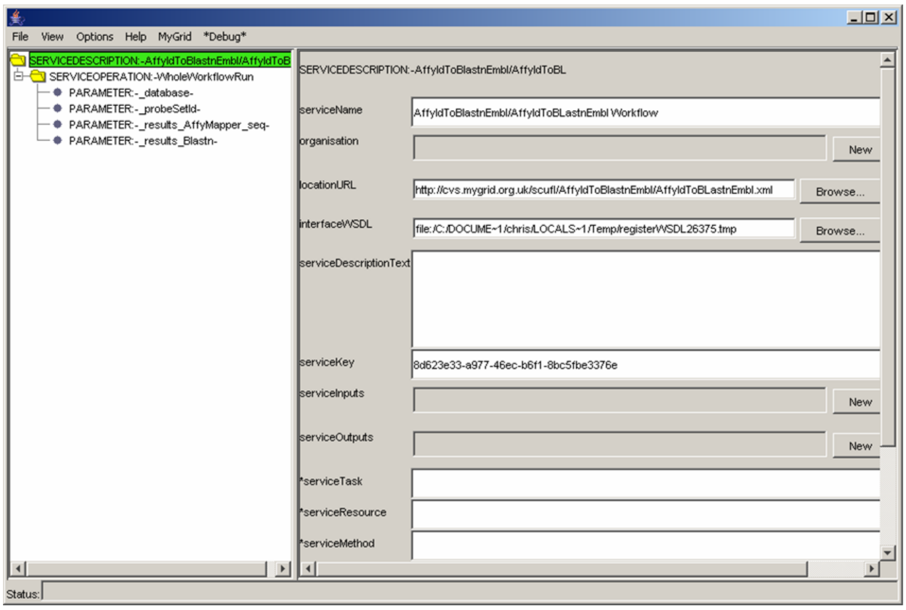
The ^{my}Grid registry built by Southampton University implements the Universal Description Discovery and Integration (`http://www.uddi.org/`) standard (UDDI). To address the specific requirements of e-Science, the registry supports further annotation of services and workflows with arbitrary structured metadata. Extensibility is achieved by using a Jena Resource Description Framework (RDF) repository (http://jena.sourceforge.net) for storage of descriptions, together with pluggable web service interfaces for registration and query by clients [2]. We have developed an RDF Schema from the conceptual model described in section 3 together with a mapping between the XML data produced by PeDRo, and the RDF used for storage and query.

The registry aims to store and manage descriptions of services covering a wide variety of domains and so does not commit to or indeed have knowledge of any specific ontology. The focus of the ^{my}Grid registry is on the management of descriptions as a whole including their federation and personalisation, together with structural queries that do not require domain dependent knowledge.

4.3 Personalisation for Re-use

The registry allows additional descriptive information to be appended to a service or workflow registration. For instance, whenever a workflow is used, the user may have feedback to provide such as the suitability of that workflow for their novel task. We provide an interface to allow such **third-party metadata** to be attached to already published workflows and services, and then to subsequently be used in discovery.

The second mechanism we provide allows users to filter the amount of information they search over in each act of discovery. The registry as a whole can be personalised, by deploying it as a **view** over other available registries. For example, if several public registries exist, containing a vast range of available services, then a bioinformatics community view would be a registry that held only those services which are likely to be useful to bioinformaticians. The bioinformatics view will be kept up to date through the registry's *notification* mechanism where it sends out notifications regarding new services that have been registered. If these services have been annotated with metadata marking them as useful to bioinformaticians then they will be included in the view. When a bioinformatician searches the community view, as opposed to the public registries, they are less likely to be presented with services that are irrelevant to their aims. Going further, an organisation can have its own view over the community view that only in-



SERVICEDESCRIPTION:-AffyIdToBlastnEmbl/AffyIdToB
SERVICEOPERATION:-WholeWorkflowRun
PARAMETER:-_database-
PARAMETER:-_probeSetId-
PARAMETER:-_results_AffyMapper_seq-
PARAMETER:-_results_Blastn-

SERVICEDESCRIPTION:-AffyIdToBlastnEmbl/AffyIdToBL

serviceName: AffyIdToBlastnEmbl/AffyIdToBLastnEmbl Workflow
organisation:
locationURL: http://cvs.mygrid.org.uk/scufl/AffyIdToBlastnEmbl/AffyIdToBLastnEmbl.xml
interfaceWSDL: file:/C:/DOCUME~1/chris/LOCALS~1/Temp/registerWSDL26375.tmp
serviceDescriptionText:
serviceKey: 8d623e33-a977-46ec-b6f1-8bc5fbe3376e
serviceInputs:
serviceOutputs:
*serviceTask:
*serviceResource:
*serviceMethod:

Figure 3: Screenshot of PeDRo showing the description of workflow input.

cludes services rated as high quality by that organisation, and a user within the organisation can have a personal view over the organisation view that includes only services they determine to be worth using.

Combining the two pieces of functionality above provides further opportunity for personalisation. For example, the view mechanism allows opinions of other trusted individuals to be taken into account when performing discovery, as services can be filtered on the third-party metadata attached by those individuals. Further, if public registries do not allow third-party metadata to be attached, then views can be provided that copy the contents of the public registries and also allow such metadata to be attached.

4.4 Extensive Re-use

Another way of increasing re-use is to make discovery accessible by a wide range of users and applications. We aim to make discovery of services and workflows as accessible as possible.

Because the ^{my}Grid registry is itself a Web Service, it can be accessed remotely by a range of users and software tools on the Internet, increasing the extent of re-use. Software tools include those that process workflow descriptions to present the user with a choice based on their personal work context, as described in the rest of this paper, and those that perform discovery over the descriptions to replace services of one type with another in a workflow or use all services of a given type. The highly structured machine interpretable metadata stored by the registry as RDF allows these software tools to provide much more detailed support in choosing or substituting operations within a workflow.

The registry follows the Web Service de-facto standard for publishing and discovery, UDDI, so many users who have not previously used ^{my}Grid can easily move to using our registry. However, many users will start from using a different discovery technology to UDDI, such as bioMoby. We solve this by providing a pluggable interface to the registry, in which different APIs (provided either at the client side or the server side) can manipulate the same underlying data model. This allows services published using one technology to be discovered using another, again increasing the extent of re-use.

Although in this paper we have highlighted the need for user-centric descriptions, it is still essential to provide a formal interface description in order that client applications can discovery programmatic-level details and actually invoke service operations. The ^{my}Grid registry provides fine grained access to service and workflow interfaces as described using WSDL files. Work by IBM, e.g. [3], has shown that WSDL can be used to describe the interface of SOAP services, Java applications, workflows and other such components at a programmatic level. The registry parses these WSDL files and allows further metadata annotation of the programmatic entities described within them. In the case of WSDL based services in which each WSDL operation corresponds to a unit of functionality, it is possible to explicitly associate our high level description of operation described in section 3 with the corresponding entity within the

WSDL file. Unfortunately for cases such as Soaplab the mapping between functionality and WSDL operations is not straightforward and this feature cannot be used.

4.4.1 Domain-dependent indexing and query

The registry is designed to be domain independent. To keep this generality whilst allowing domain dependent indexing and query we have developed an architecture in which external indexing components with domain knowledge can act in cooperation with the registry. For example an author may describe a workflow that accepts as input ”sequence data” where ”sequence data” is a concept from a specific bioinformatics ontology which also states that ”sequence data” has a subtype ”protein sequence data”. A subsequent user querying for workflows that accept ”protein sequence data” using the same ontology would expect to find the aforementioned workflow. ^{my}Grid has developed such a component (called Feta) that makes use of domain dependent ontologies (in this case bioinformatics) and associated ontological reasoning. For uniformity, the Feta component also represents the descriptions in RDF and makes use of Jena’s reasoning capabilities to correctly answer queries based on ontological information, as described above.

4.5 User interaction through the Taverna workbench

Access to the registry and PeDRo must be available to the user during workflow creation and reuse. The Taverna workflow workbench (`http://taverna.sourceforge.net`) therefore includes a registry plug-in that allows the user to register, annotate and search for services and workflows. When the user begins designing a new workflow, they first launch the registry plug-in and use the query builder to search for existing services or workflows that are relevant to the task. This search can be performed along a number of axes including free text search of name and description, ontology based search over the semantic types of inputs, outputs, the kind of task performed, the kind of resource or application or algorithm used. Figure 4 shows such a query being created from within the Taverna workbench using terms from the bioinformatics ontology. The results of the search may lead to three distinct situations:

- The user has found a service or workflow that performs exactly what they require. They can drag this service into the workflow editor and run it with their data.
- The user has found a workflow, that provides similar functionality but requires modification. They drag this workflow into the editor and make those modifications before execution.
- The user has found a number of services, which provide fragments of functionality and must be orchestrated together. They drag these services into the editor and build a workflow by describing the necessary data and control flows between each service.

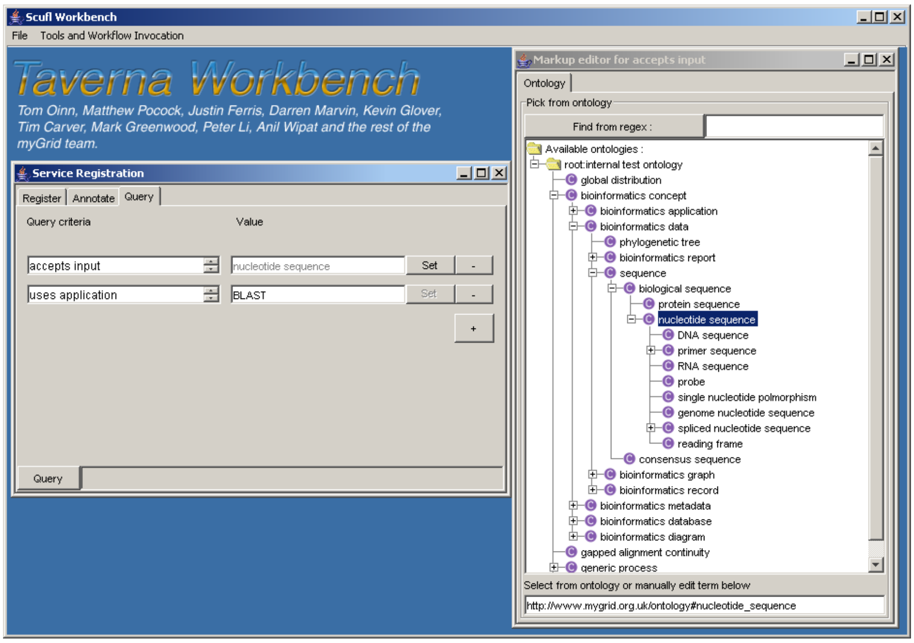
Once the workflow is proven, the user can add their experience to the registry using its third party metadata facility. If they are reusing workflows, or services, they can select those items in the registry using the Taverna registry plug-in, and then launch PeDRo to provide a structured description of their experience. If they have produced a new workflow, they can publish it with an associated description, again using the Taverna registry plug-in.

5 Deployment of discovery components

If the components are to be deployed in a bioinformatics setting it is assumed the current ^{my}Grid model of user-centric description is adequate and would not need amending. Also the concepts provided by the ^{my}Grid ontology will provide a starting point for forming descriptions. If the components are to be used for a different domain, it may be necessary to amend the model and it will certainly be necessary to build a new ontology for that domain. PeDRo’s dynamic generation of a user interface based on the XML Schema data model, means that any modifications will be instantly reflected in the user interface used to write descriptions. However, the use of RDF as the storage and query representation means that changes to the model require changes to the rules that map XML to RDF statements and also amendments to the pre-canned queries that are available to the user.

Once the model and initial ontology have been developed it is then possible to deploy the various components. The Registry is deployed as a Web Service in a suitable container such as Apache’s Jakarta Tomcat [4]. Taverna is a Java desktop application in which the registry plug-in can be included. Feta is currently a Java application but it is planned to turn this into a Web Service. All components can be obtained from `http://www.mygrid.org.uk` except Taverna which obtainable from `http://taverna.sourceforge.net`.

[4] Available from http://jakarta.apache.org


Scufl Workbench
File
Tools and Workflow Invocation
Taverna Workbench
Tom Oinn, Matthew Pocock, Justin Ferris, Darren Marvin, Kevin Glover, Tim Carver, Mark Greenwood, Peter Li, Anil Wipat and the rest of the myGrid team.
Service Registration
Register
Annotate
Query
Set
Set
-
-
+
Query
Markup editor for accepts input
Ontology
Pick from ontology
Find from regex :

Query criteria	Value
accepts input	nucleotide sequence
uses application	BLAST

```
Available ontologies :
  root:internal test ontology
    global distribution
    bioinformatics concept
      bioinformatics application
      bioinformatics data
        phylogenetic tree
        bioinformatics report
        sequence
          biological sequence
            protein sequence
            nucleotide sequence
              DNA sequence
              primer sequence
              RNA sequence
              probe
              single nucleotide polmorphism
              genome nucleotide sequence
              spliced nucleotide sequence
              reading frame
          consensus sequence
        bioinformatics graph
        bioinformatics record
      bioinformatics metadata
      bioinformatics database
      bioinformatics diagram
    gapped alignment continuity
    generic process
```

Select from ontology or manually edit term below

http://www.mygrid.org.uk/ontology#nucleotide_sequence

Figure 4: Using ontological knowledge to answer queries.

6 Discussion

Even at this early stage, there are over three hundred bioinformatics web services available to ^{my}Grid workflows and thirty bioinformatics workflows. We have found a great deal of commonality between workflows, and several common patterns are emerging. This reinforces the need for a registry and also raises the issue of how to represent and search for these common patterns. There is always a temptation to 'do it yourself' and not take the time to review what is already available in terms of workflows and services. It is therefore essential to bring workflow and service discovery into the workflow-editing environment making it as easy to reuse as to build from scratch. By providing a Taverna plug-in we hope to approach this goal. Reuse also depends on a rich mature registry full of previously published, well-described workflows. To support this we simplify registration by integrating a client into the Taverna workbench, and also reduce the amount of description required for initial registration. The author can therefore make the workflow available to others sooner rather than later, and provide a richer metadata description as time goes on. We have still to answer several questions. How many users will actually take the time to provide descriptions (however straightforward it is to do so)? If adoption is low, are their remaining usability barriers that can be addressed? How do we manage the maintenance of the ontology as users require more terms? How do users want to search for services and workflows? Do our current pre-canned queries reflect their requirements? As the user base for Taverna grows we hope to revisit these questions. Other projects both within the e-Science programme and internationally recognise the need for catalogues of workflows. For example, DiscoveryNet (`http://www.discovery-on-the.net`) is developing a workflow warehouse. We aim to align the metadata description of workflows written in ^{my}Grid with other projects to enable effective sharing of workflow designs across projects.

Acknowledgements

This work is supported by the UK e-Science programme EPSRC GR/R67743, & DARPA DAML subcontract PY-1149, Stanford University. The authors would like to acknowledge the ^{my}Grid team (past and present): Matthew Addis, Nedim Alpdemir, Pinar Alper, Rich Cawley, Neil Davis, David De Roure, Vijay Dialani, Alvaro Fernandes, Justin Ferris, Robert Gaizauskas, Kevin Glover, Chris Greenhalgh, Mark Greenwood, Yikun Guo, Ananth Krishna, Peter Li, Xiaojian Liu, Phil Lord, Darren Marvin, Karon Mee, Simon Miles, Luc Moreau, Arijit Mukherjee, Tom Oinn, Juri Papay, Savas Parastiditis, Norman Paton, Steve Pettifer, Milena Radenkovic, Peter Rice, Angus Roberts, Alan Robinson, Tom Rodden, Martin Senger, Nick Sharman, Robert Stevens, Victor Tan, Paul Watson, and Anil Wipat. We also thank our industrial parners: IBM, Sun Microsystems, GlaxoSmithKline, AstraZeneca, Merck KgaA, geneticXchange, Epistemics Ltd, and Network Inference.

References

[1] Matthew Addis, Justin Ferris., Mark Greenwood, Peter Li, Darren Marvin, Tom Oinn, and Anil Wipat. Experiences with eScience workflow specification and enactment in bioinformatics. In *Proc UK e-Science All Hands Meeting 2003*, pages 459–466, September 2003.

[2] Phillip Lord, Chris Wroe, Robert Stevens, Carole Goble, Sime Miles, Luc Moreau, Keith Decker, Terry Payne, and Juri Papay. Semantic and personalised service discovery. In *Proc UK e-Science programme All Hands Conference*, pages 787–794. EPSRC, 2003.

[3] Nirmal K. Mukhi. Web service invocation sans SOAP. *IBM developerWorks*, http://www-106.ibm.com/developerworks/webservices/library/ws-wsif.html, September 2001.

[4] 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. *accepted for Bioinformatics*, 2004.

[5] Martin Senger, Peter Rice, and Tom Oinn. SoapLab a unified Sesame door to analysis tools. In *Proc UK e-Science All Hands Meeting 2003*, September 2003.

[6] Mark Wilkinson and Matthew Links. BioMOBY: an open-source biological web services proposal. *Briefings In Bioinformatics*, 3(4):331–341, 2002.

[7] 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.