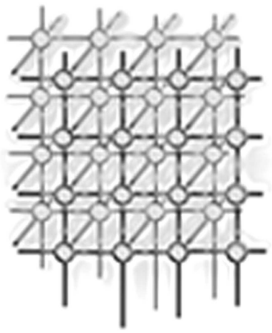
Recycling workflows and services through discovery and reuse

DOI:
10.1002/cpe.1050

Document Version
Accepted author manuscript

CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2000; **00**:1–7 *Prepared using* cpeauth.cls *[Version: 2002/09/19 v2.02]*



Recycling workflows and services through discovery and reuse

Chris Wroe[1], Carole Goble[1,*] Antoon Goderis[1], Phillip Lord[1], Simon Miles[2], Juri Papay[2], Pinar Alper[1], Luc Moreau[2]

[1] *School of Computer Science*
University of Manchester
Oxford Road
Manchester M13 9PL, UK
[2] *School of Electronics and Computer Science*
University of Southampton
Southampton SO17 1BJ, UK

SUMMARY

Scientific workflows are becoming a valuable tool for scientists to capture and automate e-Science procedures. Their success brings the opportunity to publish, share, reuse and repurpose this explicitly captured knowledge. Within the ^{my}Grid project, we have identified key resources that can be shared including complete workflows, fragments of workflows and constituent services. We have examined the alternative ways these can be described by their authors (and subsequent users), and developed a unified descriptive model to support their later discovery. By basing this model on existing standards, we have been able to extend existing Web Service and Semantic Web Service infrastructure whilst still supporting the specific needs of the e-Scientist. ^{my}Grid components enable a workflow life-cycle that extends beyond execution, to include discovery of previous relevant designs, reuse of those designs, and subsequent publication. Experience with example groups of scientists indicates that this cycle is valuable. The growing number of workflows and services mean more work is needed to support the user in effective ranking of search results, and to support the repurposing process. Copyright © John Wiley & Sons, Ltd.

KEY WORDS: Workflow, Reuse, Semantic Discovery

*Correspondence to: Carole Goble, School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK
†E-mail: carole@cs.man.ac.uk
Contract/grant sponsor: UK e-Science programme EPSRC; contract/grant number: GR/R67743

Received
Revised

1. Introduction

In e-Science, a key task is the creation and use of processes for experiment design, data analysis and knowledge discovery, that couple together a wide range of Web Service and Grid enabled resources. Workflow techniques are an important part of *in silico* experimentation, potentially allowing the e-Scientist to describe and enact their experimental processes in a structured, repeatable and verifiable way [2].

The ^{my}Grid project has developed a set of software components which support the e-Scientist in managing and performing biological *in silico* experiments. Web and Grid Services provide access to distributed resources whilst workflow techniques enable the orchestration of these resources. ^{my}Grid's Taverna and Freefluo workflow technology provides tools and infrastructure for the creation and management of such processes, as well as a methodology and mechanism for capturing, executing and monitoring process knowledge [2].

However, a key challenge lies in supporting the rapid assembly of these workflows from disparate services and their *reuse* in various scenarios. At the time of writing, ^{my}Grid allows access to a thousand services, and approaching a hundred workflows . These have been developed by users and service providers distributed throughout the global biology community and are accessible from within the ^{my}Grid Taverna workbench [12]. To avoid wasteful reinvention and to promote best practice by propagating knowledge verified by experience, these parts of these experiments need to be discovered, and in the case of workflows, potentially reworked [5]. Workflows created by one user might be used as is, or as a starting point by others.

Service reuse is a desirable goal of Service Oriented Architectures and Web Service middleware. Workflow reuse has received less attention, yet has the potential to: i) Reduce workflow authoring time by less re-inventing of the wheel; ii) Improve workflow quality through reuse of established and validated workflows rather than re-invention of new, and potentially error-prone, ones.

We have worked closely with two specific case studies to pilot workflow reuse:

1. Investigation of Williams-Beuren Syndrome [15]. Members of St Mary's Hospital Academic Unit of Medical Genetics, at the University of Manchester, have developed workflows i) to identify any newly deposited and relevant genome sequences in public sequence databases (later referred to as WBSwf1) ii) to characterise any genes in those new sequences using analysis tools iii) to gather related information from other databases iv) likewise to characterise proteins that will be produced from those genes.
2. Investigation of the genetic basis of Graves' Disease [8]. Members of Institute of Human Genetics at the University of Newcastle have developed a set of workflows to statistically analyse data showing the changed expression of genes in affected thyroid tissue, followed by characterisation of those genes.

Both of these example applications have stemmed from a real and immediate biological need. For these applications, members of the ^{my}Grid project have worked closely with the end-user biologists; this was particularly important in the past when much of the ^{my}Grid technology was in early development stage. More recently, however, the majority of the user base are developing, deploying and using the technology with little or no contact with the ^{my}Grid developer community. This is evidenced by the fact that the ^{my}Grid Taverna workbench was

downloaded 1500 times in the first three months of 2005. In a recent *Science Magazine* issue [1], ^{my}Grid is referenced by three articles. The technology is now in use in various research projects and groups in the life sciences (including BioMoby[†], EGEE[‡], VL-E[§] and PathPort[¶]).

Supporting reuse places additional requirements on ^{my}Grid infrastructure, to:

- **identify** reusable services and workflows.
- support the **generation** of reusable services and workflows.
- **register** and **advertise** available services and workflows in a community accessible location.
- **annotate** these registrations.
- **search** over service and workflow information by consumers.
- effectively **reuse** discovered services and workflows.
- **track** a service or workflow's reuse history.

The registration and discovery of workflows may transcend any specific workflow environment. Within ^{my}Grid, workflows are constructed using the Taverna Workbench which generates workflow specifications in the Scufl workflow language [12]. These are executed using the workflow enactor FreeFluo [2]. Other workflow environments include Discovery Net [14], Kepler [3], Geodise [17], and Triana [18]. Each project varies in: the workflow languages they use; the kinds of domain and scientific process they represent; workflow deployment and execution environments; the tools and mechanisms for supporting workflow composition; the granularity of the services they orchestrate; and the way their workflows are used.

We suggest that by understanding the workflow design life-cycle, and by defining a model for describing services and workflows, we should be able to discover and reuse not only our own native workflows but also "foreign" workflows; incorporating them within our own workflows yet executing within their native environment. Initial experiments between Kepler and Taverna suggest this is feasible.

This paper describes the workflow design life-cycle, the model we have developed for describing their experimental parts, and how specific ^{my}Grid components address these requirements through the various stages of the life-cycle.

2. Workflow design life-cycle

Scientists use workflows as a means of encoding a scientific process so that they can be reused and exchanged as commodities in their own right. Workflow reuse is intrinsically linked to a desire that workflows be shared by the community as "best practice" scientific protocols that may be reused exactly as designed or varied through simple substitutions of data, parameter settings or equivalent services. ^{my}Grid is designed to enable the development

[†]Web site: www.biomoby.org
[‡]Web site: www.eu-egee.org
[§]Web site: www.vl-e.nl
[¶]Web site: pathport.vbi.vt.edu

of *ad hoc* experimental workflows. These workflows often evolve over many versions. Each version may be valuable, and can be reused by its author.

First we distinguish the parts of an experiment we consider to be reusable:

- a *service* is an atomically deployed and executable application published as a well defined interface definition using, for example, WSDL. For example the first Williams-Beuren Syndrome workflow, WBSwf1, uses a BLAST service (to search DNA sequence databases).
- a *workflow template* is an un-invocable, un-parameterised workflow whose services are unbound to a specific end point. For example, WBSwf1 can be configured to investigate other biological organisms and specific chromosomes, using different BLAST services or genome databases. However the generic knowledge, remains constant and is re-usable.
- a *workflow instance* is a partially or fully instantiated, parameterised workflow that can be enacted, such as WBSwf1 configured to specifically investigate Williams-Beuren syndrome. These are less likely to be published widely for reuse as they may carry Intellectual Property. We collectively refer to workflow templates and instances as *workflows*.
- a *workflow fragment* is a piece of an experimental description that is a coherent sub-workflow that makes sense to a domain specialist. For example, several operations in the WBSwf1 enable the identification of new sequences in a genome database similar to a query sequence. Others filter this result further and retrieve associated database records for these new sequences. Each fragment forms a useful resource in its own right and are identified at publication time.

Furthermore, we distinguish between reuse and repurposing:

- A user will *reuse* a service, workflow or workflow fragment that fits their purpose and could be customised with different parameter settings or data inputs to solve their particular scientific problem.
- A user will *repurpose* a workflow or workflow fragment by finding one that is close enough to be the basis of a new workflow modified for different purpose.

While reuse is well understood, repurposing is less so. It inherently requires metrics for measuring similarity, and repurposing actions to rework the workflow by adding, removing, or replacing steps, or by altering the control structure.

In Figure 1, we show an extended experimental life-cycle:

1. Before embarking on a new design authors should consult a catalogue or *registry* of previously published workflows. Search facilities identify any existing workflow that performs exactly what they want, which is parameterised and instantiated as such; exactly what they want if it were re-parameterised; or is similar to their needs with slight modification. Once found it must be easy to transfer this workflow into a workbench for further editing and execution.
2. Workflows or their fragments are potentially edited; services are parameterised or bound to end points but rarely altered. Other services, workflows or workflow fragments are sought, or new ones are created. These too must be easy to integrate into the workflow design, and assembled, instantiated and executed within the Taverna workbench.
3. We cycle through this process until the scientist is happy, and the workflow has proved its worth.

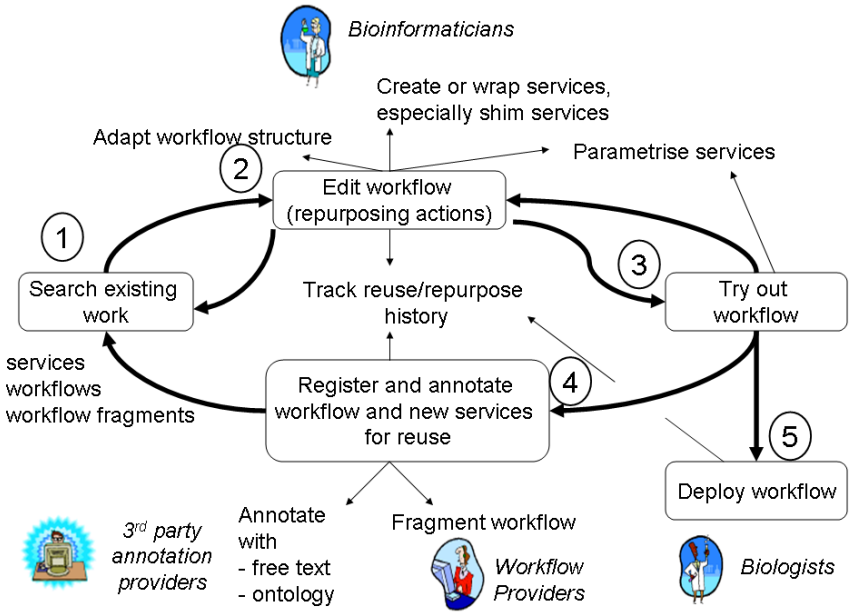


Figure 1. An e-Science experimental life-cycle which extends beyond design and execution of workflows, to encompass discovery of existing resources for inclusion, and publication of its design.

4. It must then be a simple task to publish the workflow template to a registry, annotate with a description and additional knowledge on the suitability of the original workflow for this task, so that others can benefit. Conscientious users might partition the workflow into coherent fragments and publish those; otherwise an automated process might attempt the same. It must also be possible to go back and annotate the original workflow with this experience.
5. The user also publishes the workflow to a portal so that it can be run by scientists with no workflow expertise.

This life-cycle is dependent on descriptions of reusable parts of a workflow and requires effort on the part of scientists in providing them. Given that this effort is significant, we must be sure these descriptions adequately support reuse and repurposing. The next section examines the requirements on these descriptions and the subsequent design of a descriptive model.

3. The myGrid descriptive model for workflows and services

Reuse can only be achieved efficiently if there is a catalogue or registry of existing workflows and services, with descriptions which drive indexing and searching.

3.1. Workflows mirror a scientist's view

Scientific workflows within ^{my}Grid are notable by their apparent simplicity (in terms of workflow constructs used). Scientists write them with little training in workflow technology, to orchestrate application level services such as genome sequence database access, genome sequence analysis tools, or simulation services. The workflow language Scufl [2] is designed so that each workflow step corresponds closely with what the scientist would regard as a single task, even though during execution this may embody more complex technical control structures such as iteration, or job submission. The focus of the workflow author is therefore on the scientific task performed by component operations, the workflow as a whole, and the experience others have gained in using them. At least in initial design, they are less concerned with technical information about how each operation is invoked.

A structured description of this scientific functionality provides more scope for assistance by the middleware during the workflow life-cycle. The more structure is included in the descriptions, the more complex the architecture required to exploit these, and the more effort is involved to author these descriptions. We have identified several stages or levels of description, illustrated in Figure 2.

- **Level 1: Natural language description** Natural language is the most flexible representation, and most amenable for users to write. A simple example is the short description of example workflows given in at the end of Section 2. However the accuracy of search is reduced, because of variations in descriptive style and terminology. These descriptions are opaque to the ^{my}Grid components and so they cannot provide any further support for the task of repurposing beyond text search.
- **Level 2: Declarative description as an atomic service** By describing the overall inputs and outputs of a workflow, together with its overall capability, we can search for a workflow as though it is an atomic service–i.e. using no information about its internal structure. If this description is structured and uses a controlled vocabulary, accuracy of search will be increased over that of natural language. For example, WBSwf1 could be further described as accepting `sequence data`–a term from a controlled vocabulary– as input. However, the effort in producing these descriptions is higher. Components are needed to support authoring structured service descriptions, their registration, search, and the maintenance of the associated controlled vocabulary.
- **Level 3: Declarative description as a composite service** A composite service is one in which we have information about internal operations. We can treat a workflow as a composite service in which workflow operations are treated as a bag of internal service operations. By providing a structured description of these internal operations, we can support discovery of a workflow by searching for the components it contains. WBSwf1 contains a BLAST sequence similarity operation. Users wishing to use this operation could discover the workflow and use it as a best practice example. Again this places additional effort on the user in describing their workflows, but requires little added complexity in the architecture.
- **Level 4: Procedural description** Level 3 descriptions provide no information on how internal operations are ordered. This information is useful when searching for complete workflows based on their structure, and in particular workflow fragments. Although

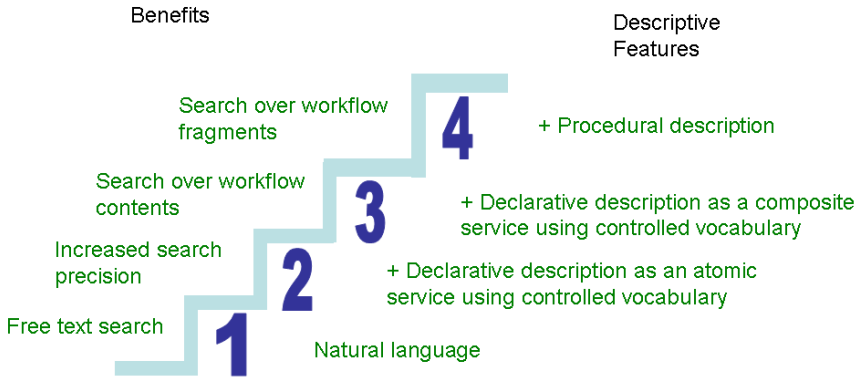


Figure 2. Incremental steps by which a workflow can be described. Each step incurs additional effort by the user, and additional complexity in software components

the ordering constraints are integral to the workflow specification, taking them into consideration during discovery and repurposing places additional requirements on the architecture. What constitutes a workflow fragment (encompassing a functional unit that is amenable to reuse) is not represented in the workflow specification. It requires additional user effort to identify useful fragments and describe them.

By recognising these different descriptive levels, we have been able to take an incremental approach to supporting discovery and repurposing. Currently, ^{my}Grid components are able to support level 1 to level 3 descriptions by leveraging off current service description infrastructure including:

- Universal Description Discovery and Integration (UDDI): the *defacto* standard for Web Service Registries (`http://www.uddi.org/`).
- Ontology Web Language Services ontology (OWL-S): a proposed standard for the semantic description of Web Services (`http://www.daml.org/services`).
- Web Services Definition Language (WSDL): the standard for the description of a Web Service interface (`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 the standards above.

Scientist centric These descriptions are to be written, and searched by scientists, and so must be in a form and use terminology understandable to them. WSDL documents are intended to provide a description of the programmatic interface of a Web Service. They are unintelligible for many users and it is not useful 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.

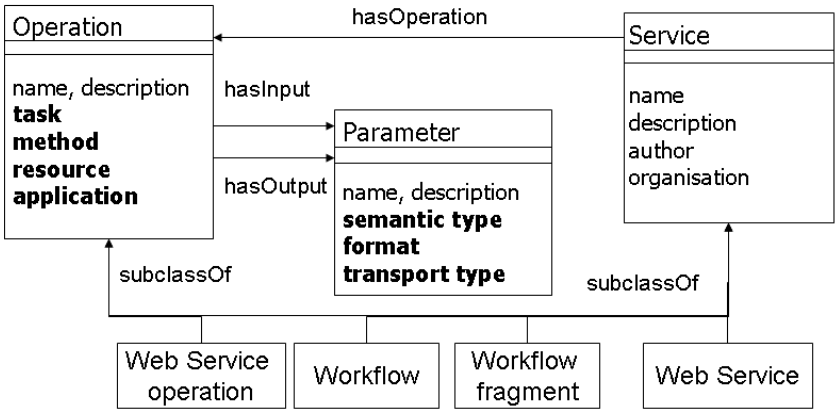


Figure 3. The abstract model of workflows and services within ^{my}Grid for the purpose of discovery. Annotation fields whose values are filled with ontology concepts are shown in bold.

Operation focused The primary aim of these descriptions is to find operations that can either be included in 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 constituent operations provided by that service. Best practice often delegates this task to an associated WSDL document.

Data centric Most services used within ^{my}Grid for part of a data pipeline workflow. Therefore a key distinguishing feature of an operation is the nature of the data flowing in and out. The common practice in bioinformatics is to store, transfer and interact with data as flat files. Although this is slowly changing, most of these formats have no formal specification and do not use any standard data structuring technologies. WSDL describes data from the bottom up–specifying structure using XML Schema. Most service providers and consumers actually want the flat file representation because their legacy tools use them. As a result many of the bioinformatics services found make very thin use of XML Schema, essentially wrapping their complex representations in a single `xs:string` envelope; enforcing a standard XML representation is not an option. Therefore the data is very poorly described by WSDL. More over, users want to search "top down", based on the data's conceptual content, such as `Protein Sequence`, and only then on any formatting or typing issues.

Based on these requirements, in ^{my}Grid, we have developed a user-centric abstract model of services and workflows that focuses on their scientific functionality in terms of operations and nature of data. It can support description of workflows to level 3 as well as the range of services used within workflows. This model builds on existing elements of UDDI, WSDL together with additional scientist level annotation inspired by OWL-S. Figure 3 shows an overview of the model. The key entities are:

Service This is the unit of *publication*. It corresponds to the Business Service entity in UDDI and can describe a Web Service, a workflow or a workflow fragment. Its fields describe who published this service, what organisation they belong to, together with a free text description of

the service. Services can provide more than one operation. This is the case with workflows and their fragments, and many WSDL described Web Services. Therefore functionality is described using a separate entity *the operation*.

Operation This is the unit of *functionality*. This corresponds to the operation entity of WSDL, and describes the functionality of a workflow, a workflow fragment, or its component service operations. To address scientist centric requirements the entity has four additional annotation fields to describe high level attributes including 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`). To promote consistency of descriptions and so accuracy of search, these four fields are filled using concepts from an ontology.

Parameter We use the collective term *parameter* to describe the type of data used or produced by an operation. Parameters can be described at several levels from a high level semantic description such as `protein sequence` (provided by an ontology), through formatting descriptions such as `FASTA format` for protein sequences, to low level types such as `String` described in WDSL interface documents.

4. ^{my}Grid component overview

^{my}Grid supports the life-cycle and descriptive model described in previous sections with a collection of components as shown in Figure 4.

1. Search and Discovery: The **^{my}Grid Grimoires Registry** is an implementation of the business logic of UDDI. It provides a unified store of service and workflow descriptions (to level 3) by treating a workflow as a composite service (More Information and releases of Grimoires are available from `http://www.grimoires.org`) [11]. **Feta** is a additional indexing and query service, that does not store service descriptions, but provides the ability to search over the scientist centric descriptions in a subject specific way, using taxonomy information in the associated ontology [9]. Feta has a data model and associated ontology which is domain specific and describes services in the form that bioinformatics scientists think of their domain. For example, a scientist may describe a workflow as accepting an input labelled with the ontological concept of `sequence data`. In the ^{my}Grid bioinformatics ontology, this term is specialised by the term `protein sequence data`. Searches for the workflows accepting the latter should also return those accepting the former. This functionality is provided by the Jena toolkit. Feta is, in essence, implemented by a set of canned queries over an RDF data model. While this functionality could have been implemented with an RDBMS, Jena provides an convenient link to an ontology, allowing the specialisation and generalisation described above through use of its subsumption (RDF(S)) inference engine [13]. It is planned that in time the functionality of Feta will be implemented within Grimoires.

Access to the registry and Feta must be available to the user during workflow creation and reuse. The Taverna workflow workbench, therefore includes a Feta plug-in enabling search for services and workflows using a query builder. This search can be performed along a number

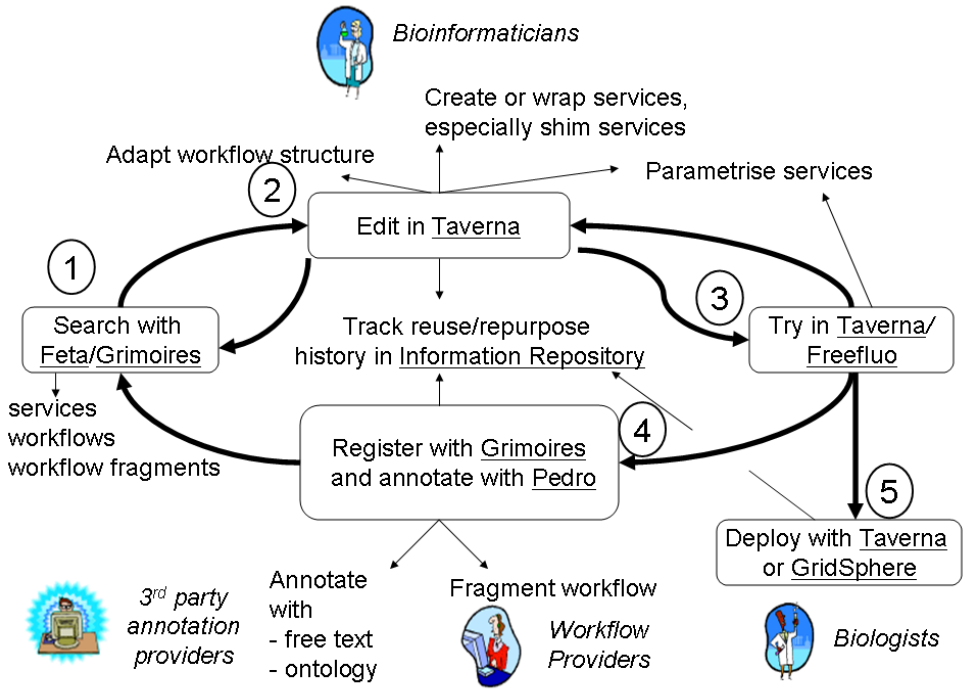


Figure 4. The workflow life-cycle is supported by a collection of ^{my}Grid components (underlined in this diagram).

of axes, including free text search of name and description, and 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 5 shows such a query being created from within the Taverna workbench using terms from the bioinformatics ontology. The ontology (available from `http://www.mygrid.org.uk`) is now developed in the OWL language using the ontology editor Protégé [‖]. It contains about 600 bioinformatics and molecular biology concepts.

2. & 3. Workflow editing and trial: The Taverna workflow workbench (available from `http://taverna.sf.net`) is used to edit workflows whilst FreeFluo enactor (available from `http://freefluo.sf.net`) executes them. Tracking how the workflow has changed is achieved using the ^{my}Grid Information Repository which stores and versions the actual Scufl specification file.

4. Registration and annotation for reuse: As already mentioned Grimoires supports the registration of workflows as well as services. In order to provide scientist level annotation,

[‖]`http://protege.stanford.edu`

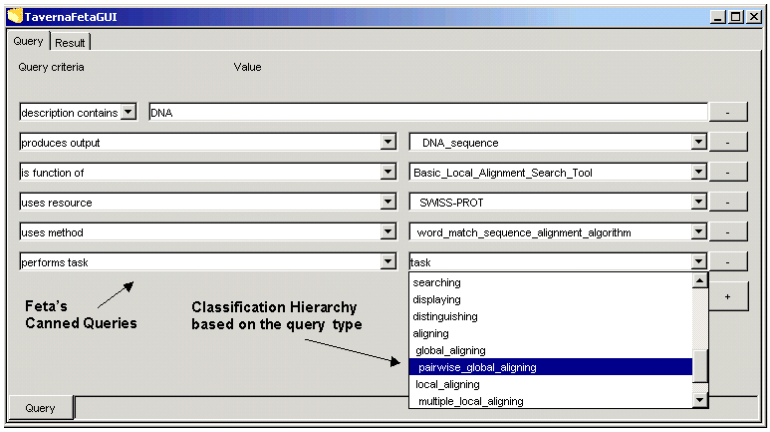


Figure 5. Using ontological knowledge to answer queries.

the workflow author uses Pedro [4], an ontology aware annotation tool (available from `http://pedrodownload.man.ac.uk/`). It allows users to enter structured data or metadata based on a predefined XML schema. Within ^{my}Grid, an XML Schema derived from the conceptual model described in Section 3 is used. It has integral support for ontologies, which provide vocabulary for specific fields. For these fields, Pedro presents the user with an ontology browser from which the user can choose appropriate concepts. The structured description is then stored in the registry and available for query. Grimoires supports further annotation by allowing arbitrary structured metadata from third parties. 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.

5. Portal publication: ^{my}Grid has developed a workflow portlet deployed in the GridSphere portal environment (`www.gridsphere.org/`). In general, workflow authoring is likely to be performed by one member of a biology lab. Many other members will wish to use the completed workflow however without the necessity of installing client side software. The portlet is therefore aimed at enabling users to enact specified workflows, to check on the status of running workflows and manage the end results. Initial prototypes are available from ^{my}Grid CVS (available from the website).

5. Experience: Supporting the scaling of reuse and repurposing

During the development of workflows enabling the investigation of Williams' syndrome, significant reuse of workflows previously developed for the Graves' disease investigation was possible; this resulted in a dramatic drop in the time required for workflow authoring. By writing workflow descriptions based on our model for the Graves' disease workflows and collecting feedback from scientists searching them, we have been able to assess the effectiveness of our discovery infrastructure. Initially, with only a small number of workflows available, it

was feasible to manually browse the results to find which were suitable for reuse. We now have close to a hundred available workflows using up to fifty services each; manual browsing is largely impractical. Currently, we are investigating how reuse and repurposing can be made more scalable, based on techniques for dynamically finding and ranking workflow fragments as well as a more interactive approach to convey results to users.

Dynamically finding workflow fragments Workflow fragments are identified at publication time by the author or 3rd party annotators. It is hard to predict beforehand which particular piece of functionality will be in demand by potential users and thus should be encapsulated. Such manual fragmentation can be complemented by dynamically identifying fragments based on a user's current context. For example, when we progress to representation level 3, and ordering is taken into account, one can find out whether a fragment exists that generates one piece of data starting from another piece of data, or whether particular services have been connected in the past, possibly relying on so called "shim" services [6], to provide mediation. We have successfully performed initial experiments with a workflow ontology that supports queries like these based on OWL reasoning.

Ranking workflow fragments Given the possible number of search results, ranking them is of prime importance. We are investigating the use of rankings that are calculated based on similarity metrics for fragments of workflows, taking into account the user's context. The metrics we are currently implementing indicate the effort it would take a user to repurpose a piece of workflow in her current context, for example by calculating how many services need to be added, removed or replaced to match similar workflows. Similarity is being measured either using semantic similarity between the input and the inputs of two workflows or the structural similarity of Scufl workflow specifications.

6. Discussion

The rapidly increasing number of workflows and associated services requires a timely solution to support their dissemination across a wide community. ^{my}Grid has built key components to support this, which have been trialled successfully to describe, discover and reuse workflows between pilot groups of scientists. Despite this infrastructure, 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. Therefore, we must reduce the barriers for both discovery and publication [5]. The first barrier is that of convenience and is addressed by supporting the reuse life-cycle directly within the Taverna workbench. The current search mechanism is passive in that users must take the initiative and come up with the right queries themselves. We are investigating a more interactive presentation style, whereby the system actively takes into account the user's context. Such a system could remain active in the background, and provide suggestions as this context evolves over time.

Reuse 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 with a

level 1 natural language description, and provide more structured descriptions conforming to levels 2 to 4 as time goes on.

Another artificial barrier is that presented by different workflow systems. What if a relevant workflow for ^{my}Grid users has been written but in another workflow environment? To address this a collection of projects involved in developing scientific workflow environments including ^{my}Grid, BIRN (`http://nbirn.net/`), SEEK (`http://seek.ecoinformatics.org/`), GEON (`http://www.geongrid.org/`), GriPhyN and SCEC/IT (`http://epicenter.usc.edu/cmeportal/index.html`) have got together under the auspices of the EPSRC funded LinK-up project (`http://www.mygrid.org.uk/linkup/`). One aim of the project is to align the metadata description of workflows and associated services to enable effective sharing of workflow designs across more diverse projects. Another aim is to investigate how workflow interoperability can be improved, by identifying common patterns in workflow design, workflow enactment and results handling. Other e-Science projects such as DiscoveryNet and Geodise have explicit reuse infrastructure ([14], [17]) and interworking of these is also an important goal.

By unifying scientist-level description of workflows with that of services we are able to tap into the large amount of related work in service discovery. In terms of semantic web services standardisation efforts, the ^{my}Grid approach is inspired to some extent by OWL-S and related to WSMO initiatives.** A number of authors have experimented with service discovery based on more sophisticated OWL reasoning, using the OWL-S Profile (see for instance [16], [17]) as well as WSMO Capability descriptions [7]. Unlike the discovery done to date in OWL-S and WSMO, we also envisage discovery of workflow fragments, by including a simple notion of structure in the descriptions of levels 3 and 4 (workflow as a bag of services and simple orderings). In light of the OWL-S and WSMO work, we must guard against adoption of descriptions that require a prohibitive amount of effort and technical expertise to write. In contrast to these initiatives, ^{my}Grid focuses squarely on supporting humans (bioinformaticians), and not intelligent software robots that automatically combine services to reach a given goal. The more ambitious focus of these standards, greatly increases the complexity of descriptions and so the effort required to write them. In general, it does not seem likely scientists would be willing to relinquish control over the composition of *in silico* experiments, except when these are minor and "experimentally neutral" [10]. As a result, ^{my}Grid provides a relatively simple model used solely to support discovery.

We have still to answer several questions. How many users (in the wider community) 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? Are there additional ways that users want to search for services and workflows? As the user base for Taverna grows together with the collection of scientific workflows, we hope to revisit these questions.

**Respective Web sites: `http://www.daml.org/services` and `http://www.wsmo.org`

Acknowledgements

This work is supported by the UK e-Science programme EPSRC GR/R67743. The authors would like to acknowledge the ^{my}Grid team. Hannah Tipney developed workflows for investigation of Williams-Beuren Syndrome and is supported by The Wellcome Foundation (G/R:1061183). We also thank our industrial partners: IBM, Sun Microsystems, GlaxoSmithKline, AstraZeneca, Merck KgaA, geneticXchange, Epistemics Ltd, and Network Inference.

REFERENCES

1. Science magazine: Special issue on distributed computing 308(5723), 6 May 2005. http://www.sciencemag.org/sciext/computers, site last accessed 14 December 2005.
2. 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.
3. I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludscher, and S. Mock. Kepler: An extensible system for design and execution of scientific workflows. In *16th Intl. Conference on Scientific and Statistical Database Management(SSDBM)*, pages 423–424, Santorini Island, Greece, June 2004.
4. Kevin Garwood, Phillip Lord, Helen Parkinson, Norman W. Paton, and Carole Goble. Pedro ontology services: A framework for rapid ontology markup. In *The Semantic Web: Research and Applications: Second European Semantic Web Conference ESWC*, Heraklion, Greece, May 2005.
5. Antoon Goderis, Ulrike Sattler, Phillip Lord, and Carole Goble. Seven bottlenecks to workflow reuse and repurposing. In *Fourth International Semantic Web Conference (ISWC 2005)*, volume 3792, pages 323–337, Galway, Ireland, 2005.
6. Duncan Hull, Robert Stevens, Phillip Lord, Chris Wroe, and Carole Goble. Treating shimantic web syndrome with ontologies, 2004. In First AKT workshop on Semantic Web Services (AKT-SWS04) KMi, The Open University, Milton Keynes, UK.
7. Uwe Keller, Ruben Lara, Axel Polleres, Ioan Toma, Michael Kifer, and Dieter Fensel. Wsmo web service discovery. WSML Working Draft WSML Deliverable D5.1 v0.1, University of Innsbruck, 12 November 2004.
8. Peter Li, Keith Hayward, Claire Jennings, Kate Owen, Tom Oinn, Robert Stevens, Simon Pearce, and Anil Wipat. Association of variations in i kappa b-epsilon with graves disease using classical and mygrid methodologies. September 2004.
9. Phillip Lord, Pinar Alper, Chris Wroe, and Carole Goble. Feta: A light-weight architecture for user oriented semantic service discovery. In *The Semantic Web: Research and Applications: Second European Semantic Web Conference ESWC*, Heraklion, Greece, May 2005.
10. Phillip Lord, Sean Bechhofer, Mark Wilkinson, Gary Schiltz, Damian Gessler, Carole Goble, Lincoln Stein, and Duncan Hull. Applying semantic web services to bioinformatics: Experiences gained, lessons learnt. In *Proceedings of 3rd International Semantic Web Conference (ISWC 2004) LNCS 3298*, pages 350–364, Hiroshima, Japan, November 2004.
11. Phillip Lord, Chris Wroe, Robert Stevens, Carole Goble, Simon 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.
12. Tom Oinn, Mark Greenwood, Matthew Addis, M. Nedim Alpdemir, Justin Ferris, Kevin Glover, Carole Goble, Antoon Goderis, Duncan Hull, Darren Marvin, Peter Li, Phillip Lord, Matthew Pocock, Martin Senger, Robert Stevens, Anil Wipat, and Chris Wroe. Taverna: Lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience*, Accepted for publication.
13. Dave Reynolds. Jena 2 inference support, November 2004.
14. S. Al Sairaf, F. S. Emmanouil, M. Ghanem, N. Giannadakis, Y. Guo, D. Kalaitzopolous, M. Osmond, A. Rowe, iJ. Syed, and P. Wendel. The design of discovery net: Towards open grid services for knowledge discovery. *International Journal of High Performance Computing Applications*, 2003.

15. 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.
16. Katia Sycara, Massimo Paolucci, Anupriya Ankolekar, and Naveen Srinivasan. Automated discovery, interaction and composition of semantic web services. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1):27–46, December 2003.
17. Feng Tao, Liming Chen, Nigel Shadbolt, Fenglian Xu, Simon J. Cox, Colin Puleston, and Carole A. Goble. Semantic web based content enrichment and knowledge reuse in e-science. In *CoopIS/DOA/ODBASE On The Move to Meaningful Internet Systems*, pages 654–669, Larnaca, Cyprus, 2004.
18. Ian Taylor, Matthew Shields, and Ian Wang. *Grid Resource Management*, chapter Resource Management of Triana P2P Services. Kluwer, June 2003.