Using Semantic Web Technologies for Representing E-science Provenance

Jun Zhao[1], Chris Wroe[1], Carole Goble[1], Robert Stevens[1], Dennis Quan[2], and Mark Greenwood[1]

[1] Department of Computer Science
University of Manchester
Oxford Road Manchester
United Kingdom M13 9PL
{zhaoj,carole,robert.stevens,cwroe}@cs.man.ac.uk
[2] IBM T. J. Watson Research Center
1 Rogers Street Cambridge
MA 02142 USA
dennisq@us.ibm.com

Abstract. Life science researchers increasingly rely on the web as a primary source of data, forcing them to apply the same rigor to its use as to an experiment in the laboratory. The myGrid project is developing the use of *workflows* to explicitly capture web-based procedures, and *provenance* to describe how and why results were produced. Experience within myGrid has shown that this provenance metadata is formed from a complex web of heterogenous resources that impact on the production of a result. Therefore we have explored the use of Semantic Web technologies such as RDF, and ontologies to support its representation and used existing initiatives such as Jena and LSID, to generate and store such material. The effective presentation of complex RDF graphs is challenging. Haystack has been used to provide multiple views of provenance metadata that can be further annotated. This work therefore forms a case study showing how existing Semantic Web tools can effectively support the emerging requirements of life science research.

1 Introduction

Life science researchers have made early and heavy use of Web technologies to access large datasets and applications [1]. Initiatives such as the Human Genome Project [2] have meant that rather than sequencing human genes in the laboratory, it is possible to download sequences from the Web. As well as data, tools are also available on the Web, and data are moved between tools to perform analyses [1]. The greater reliance on these resources as primary sources of data means *ad hoc* web browsing is giving way to a more systematic approach embodied by the term e-Science within the UK research community [3]. By analogy to the laboratory, web-based procedures to analyze or integrate data are called *in silico* experiments.

S.A. McIlraith et al. (Eds.): ISWC 2004, LNCS 3298, pp. 92–106, 2004.

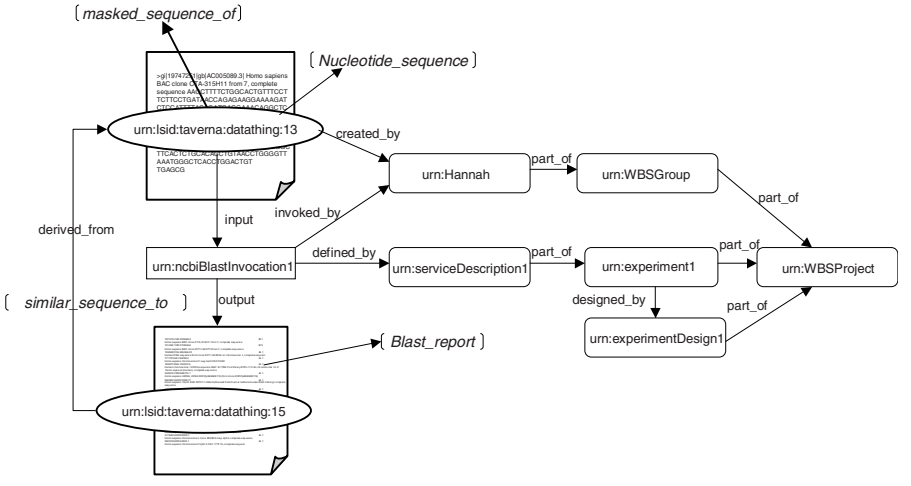

Fig. 1. An example e-Science provenance log.

The results of an *in silico* experiment are of reduced value if the materials, methods, goals, hypotheses and conclusions of an experiment are not recorded. Traditionally, in the lab, a scientist records such information in a lab-book and uses these records of where, how and why results were generated in the process of analysing, validating and publishing scientific findings. These records are the *provenance* of an experiment. Provenance data are also needed for e-Science *in silico* experiments, in order to help e-Scientists to verify results, draw conclusions and test hypotheses. In e-Science provenance logs, each piece of data used in an *in silico* experiment; the tools used to analyse those data; the results generated; and the associations and derivations of these data and tools need to be recorded. Part of such a provenance log for an e-Science experiment is shown in Fig 1. The recording, indexing, organisation and use of these provenance data are a key aspect of e-Science.

myGrid [1], as a pilot e-Science project in the U.K., aims to provide middleware services not only to automate the execution of *in silico* experiments as workflows in a Grid environment, but also to manage and use results from experiments [4]. Currently this project is being developed using several molecular biological scenarios. This paper uses the collaboration with biologists and bioinformaticians at Manchester University, researching the genetic basis of Williams-Beuren Syndrome [5]. Williams-Beuren Syndrome is a congenital disorder associated with the deletion of a small region of Human Chromosome 7. There is significant interest in characterising the genes in this region, in order to gain an insight into how an apparently small number of genes can lead to a complex set of cognitive and physical phenotypes.

[1] `http://www.mygrid.org.uk`.

Despite the release of a draft of the Human Genome, there are gaps in the known DNA sequence for this region. Therefore, a regular task for the biologists is to search the genome databases for new submissions of DNA sequence which fill in these gaps. If a relevant sequence is found, it is subjected to a barrage of web-based analysis tools to discover possible genes encoded within the DNA, and the nature of any cellular proteins produced by those genes. The nature of the task means that at each stage the number of resources and data fragments under consideration is amplified. One search can lead to many candidate DNA sequences. These DNA sequences can contain many possible genes. Those genes may produce many different proteins. Therefore, when done by hand, the process can take days interacting with web page based analysis tools, and following hyperlinks in result pages to gather further information. This personal web that the scientist selectively navigates around is transitory. No complete record remains to explain the origin of the final results. Also, the mundane nature of the task often means not all possible avenues are explored, because the scientist either misses possible routes or discards apparently uninteresting results.

myGrid was used to automate this process, and so each web-based analysis tool and data resource were wrapped as a web service. Workflows were designed to automatically orchestrate the access to these resources and data flow between them. Figure 2 shows a graphical representation of one workflow from this project, which uses two main bioinformatics applications RepeatMasker and NCBI BLAST, in order to discover new DNA sequences in the gap region for this disease. The previously transient relationships between resources have been captured as a web of provenance information, part of which is shown in Fig 1. Often discarded intermediate results are assigned identifiers and published locally, so that they become first class resources of this personal web of science [6]. In order to build this personal web, Life Science Identifiers (LSIDs) [7] are used to provide location-independent access to these resource data and metadata. The Resource Description Framework (RDF) is used to represent the web of relationships between resources by associating semantic information with these resources and their relationships, creating a Semantic Web of Provenance data. Haystack, a Semantic Web browser [8], is used to deliver these provenance webs to the end user.

If provenance information is to be shared, we need to overcome their heterogeneiety and agree on a common understanding (or semantics) as to what the contents of each data item and service represents and the relationships we provide between resources. Semantic Web ontology languages such as DAML+OIL and its successor OWL have been used to capture this semantic information [9].

As with previous web technologies, there is much interest in the life science community in the potential uses of the Semantic Web. Key organisations such as Affymetrix are publishing data in RDF format [10] and mailing lists such as the WWW Semantic Web Life Sciences mailing list are discussing how best to embrace the new technology[2]. By using existing Semantic Web technology and software such as RDF repositories, ontologies, Haystack and Life Science

[2] Archives at `http://lists.w3.org/Archives/Public/public-semweb-lifesci/`

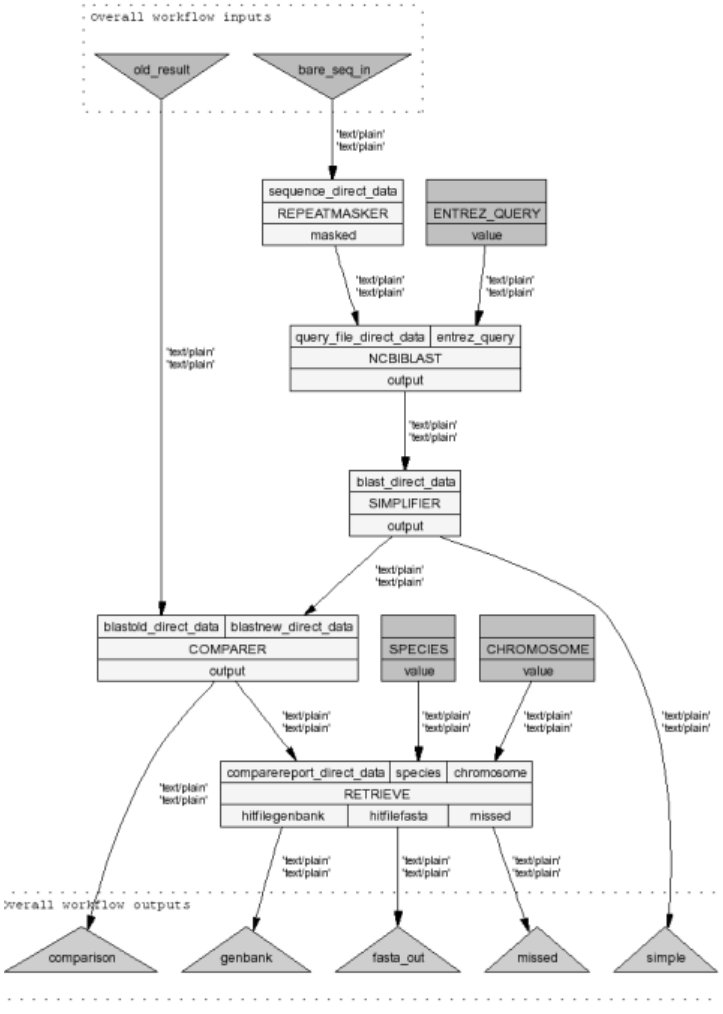

Fig. 2. An example workflow from the Williams-Beuren Syndrome scenario.

Identifiers, this work acts as a case study describing what is already possible in supporting life scientists with a Semantic Web. In this paper Sect 2 describes scientists' requirements for provenance and the implications of the information recorded. Section 3 describes why Semantic Web technologies are well suited to supporting our requirements. Section 4 describes the implementation of provenance recording and visualization.

2 Provenance Requirements and Design

Provenance has a very broad scope within e-Science. At its simplest, it describes from where something has arisen. But many different parties may have a variety

of interests in which experiments have been performed and these interests will vary over time.

Scientists in the immediate term require low level information, exploring the provenance web in a bottom up manner, focusing on individual results for a number of purposes: (i) *Debugging.* Experiments may not produce the desired results. The scientist requires a log of events recording what services were accessed and with which data; (ii) *Validity checking.* If the scientist is presented with a novel result, he/she may wish to perform expensive laboratory-based experiments based on these results. Although sure that the workflow design is valid, she may still want to check how this data has been derived to ensure it is worthy of further investigation; (iii) *Updating.* If a service or dataset used in the production of a result has changed, the scientist will need to know what implications that change has on those results.

Supervisors and laboratory heads in contrast will take a top down approach to browsing this web, to summarise information about progress made in gathering information and to aggregate provenance webs from all their researchers.

Service providers are keen to know how well their services are performing and their patterns of use so that they can better tailor them to changing demand. They are less interested in specific results and should not have access to them but focus more on aggregated, process-centric information about service access.

Research groups from other organisations and regulatory authorities may have less trust in the validity of results. They therefore would like access to as complete a record of the *in silico* experiment as possible so that they can replicate it in their local environment and rigorously check the validity of its design and performance. A detailed recipe of how an experiment is executed should be supplied for reproducing an experiment.

These different requirements lead to a design of provenance pyramid embracing many different types of resources and relationships between them, as illustrated in Fig 3. Four types of provenance logs, including process, data, organisation and knowledge provenance, have been designed, providing four different views over this provenance web and supporting a personalised web of provenance data for these different user requirements.

The *process* provenance, which is similar to traditional event logs, recording the order of services called and data parameters used for these services, with all the time-stamps of service invocations and workflow executions. The *data* derivation provenance, builds a graph of data objects in a workflow run, including inputs to a workflow, data parameters for services, intermediate results from services and final results of a workflow. Data are linked to the services that used them, to other data they were generated from, to the source they came from with a version snapshot. *Contextual* information are provided for most if not all the provenance resources, to state the user of the data or service, the experiment design of this run, the project this experiment belongs to, the hypothesis this experiment or project is based on, etc. This contextual information not only provides additional materials for organising and exploring re-

sources, but also provides another means by which we can integrate provenance logs, based on a common user, a common hypothesis, or a common design. The *organisation* provenance records who, when and where these contextual information was created and how they evolved during experiments. The fourth type of provenance presents domain specific links between experiment resources, e.g. a BLAST service input which is a nucleotide sequence is a *similar sequence* to the output of this BLAST service. As shown in Fig 3, *associations* between experiment objects are built when attaching organisation provenance to data and building knowledge linking between experiment resources. Similarly, associations are made to record *derivations* of these objects through process and data provenance. Direct data linking is the focus of data provenance, which is partly inferred from process provenance. A personalised web of provenance logs is supported by their associated organisation and annotated knowledge level information.

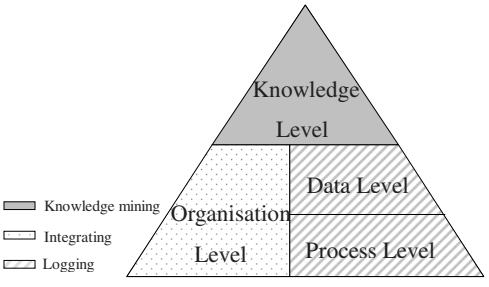

Fig. 3. Provenance design in myGrid.

Based on this core design of provenance, logs are automatically recorded during and after the workflow execution process in myGrid. Inter-related resources are identified and linked in a personal web, with an opportunity for users or third parties to annotate these logs either in free text or controlled vocabularies. Bioinformaticians, however, find it difficult to find information from a highly inter-linked provenance web. A higher level integration over this personal web is therefore required. By navigating and querying this web of provenance, scientists are able to verify results by tracing how they are produced, to test hypotheses and draw their conclusions supported by the historical record in provenance logs.

3 Semantic Web Technologies

In this section we present the Semantic Web technologies used in this project, which includes RDF, LSIDs and ontologies, in order to represent provenance logs and build a Semantic Web of these resources.

3.1 RDF

RDF was chosen as a representation model for provenance because: (i) it provides a more flexible graph based model with which to relate resources than for example the tree model of XML; (ii) it provides an explicit identification system (Universal Resource Identifiers (URIs)) for resources that allows metadata about a resource to be merged from several sources; (iii) and it provides a well-defined, but not overly constraining association with an ontology. This guides what can

be said about a resource, but does not limit description to this schema. Also from a practical standpoint several mature open source RDF repositories are available and straightforward to use.

3.2 LSIDs

Web standards, including RDF, make use of a family of identifiers collectively known as Universal Resource Identifiers (URIs). Members of this family include Unique Resource Name (URNs) and Universal Resource Locators (URLs). The latter is used for both specifying the identity and location of a resource. We are implementing a distributed system and so we must have a mechanism for retrieving both data and metadata for a distant resource from its identifier. URLs are therefore an obvious candidate. However, the myGrid project has chosen to use a type of URN called LSIDs.

LSIDs promise to uniquely and consistently identify data resources in life science, and provide a resolution protocol for the retrieval of that data and any associated metadata [7]. They have been initially developed under the umbrella of the I3C[3] and are now undergoing formal standardisation by OMG[4].

Every LSID is made up of five parts: the Network Identifier (NID); the root domain name system (DNS) name of the issuing authority; the namespace chosen by the issuing authority; the object id unique in that namespace; and finally an optional revision id for storing versioning information [7]. The structure of an LSID is shown as below:

```
urn:lsid:AuthorityID:NamespaceID:ObjectId[:RevisionID]
```

The resolution process of an LSID guarantees the persistence and uniqueness of the data object identified by the LSID, independent of the location of the data source. When resolving LSIDs, two software components are necessary, an LSID server and an LSID client. An LSID *server* is operated by the authority who publishes informatics data and assigns LSIDs to the data. An LSID client communicates to the server over the network by the LSID protocol to retrieve data or metadata identified by the LSID. Given an LSID, the *AuthorityID* is first resolved to identify the location of the LSID resolution service for that Authority by querying the Internet DNS. Once the Authority is identified, its IP address or hostname and TCP/IP port number are returned and used to construct an HTTP/SOAP endpoint URL of the following form:

```
http://hostname:80/authority
```

The LSID is then used as a parameter when querying the endpoint. A Web Service Description Language (WSDL) [11] description is returned, providing the means to retrieve the data or metadata object identified by the LSID. This WSDL document details all the functionalities available for that LSID supported

[3] Interoperable Informatics Infrastructure Consortium `http://www.i3c.org`
[4] Object Management Group `http://www.omg.org`

by the Authority. A wide variety of protocols can be indicated using this mechanism, including HTTP, FTP, file system and SOAP methods. Though the persistence of data identified by the LSID is guaranteed in the resolution process, the metadata associated with an LSID can change over time.

The perceived benefits of LSIDs to myGrid are:

1. A clean separation of data and metadata: there is no convention to define what the client receives when it resolves a URL. A document that is obtained can represent metadata, data or more likely a combination of both. The most accepted model on the web is to incorporate some RDF metadata inline into a document. However this is not applicable in myGrid:
   a) We are dealing with raw data rather than published documents. The workflow system needs to pass raw data between services without any metadata annotations ;
   b) Data and metadata can reside in different locations;
   c) Metadata may be attached to resources for which we have no control (3rd party metadata);
2. An explicit social commitment to maintaining immutable and permanent data. Provenance becomes worthless if we can allow the data resolved by a URL to change. However provenance becomes more valuable the more metadata is attached and updated.
3. A potential to apply compatible standards: we could have implemented a convention for how to retrieve metadata and data from a URL but that would have been myGrid-specific. LSIDs provided us with a ready built protocol that is undergone a standardisation procedure. By using this standard, we ensure that this RDF-based web of provenance can be merged with other Semantic Webs of information, such as protein structure metadata in the Protein Data Bank. Trial LSID authorities currently exist for the Protein Data Bank, GenBank (a genomic database) and PubMed (a life science literature database). Again, from a practical point of view, developers at IBM have supported the LSID standard by providing client and server implementations in Java, C++ and Perl, together with a demonstration user client called LSID Launchpad that forms a plug-in to Internet Explorer[5].

3.3 Ontologies

If information is to be integrated and shared across a large community, there must be an agreement on the domain semantics of that information. This is true, even for a single user. We must be able to specify semantics at two levels: first, we need a high level schema describing the classes of resource that can be linked together such as `data_item` and `service_invocation` and the relationships that can hold between them such as `output`; second, one needs a large domain vocabulary that further classifies types of resource such as data type (e.g. BLAST report), service type (e.g. sequence alignment service), and topic of interest (Williams-Beuren Syndrome). Such a shared understanding of the domain greatly facilitates querying by increasing recall and precision.

[5] `http://www-124.ibm.com/developerworks/oss/lsid/`

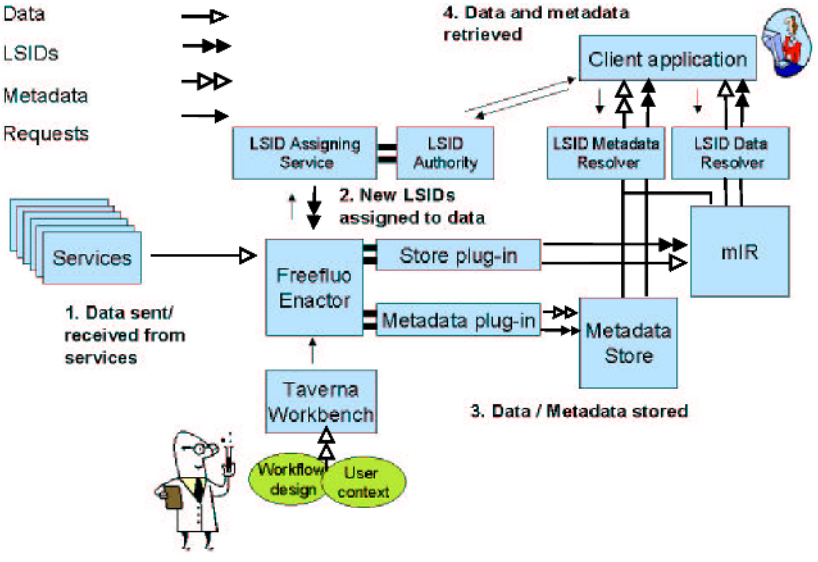

Fig. 4. Architecture for provenance generation and visualization in myGrid

4 Implementation

Having chosen RDF to describe our resources, LSIDs to identify them and ontologies to deliver a common semantic view of those data, provenance support has to be delivered within an environment for performing *in silico* experiments. Figure 4 shows the architecture of the provenance providing components in myGrid and how provenance data are drawn together. First the user starts a workflow using the Taverna workflow workbench[6]. The workbench holds organisation information such as user identity, which is passed to the workflow enactor together with input data and workflow specification. As the workflow is run, the enactor stores data (using the mySQL RDBMS), and metadata (in a Jena RDF repository[7]) corresponding to our four views of provenance. Each resource (including person, organisation, data and service) is assigned an LSID and made available via an LSID authority implemented using an open source framework[8]. Client applications can then visualize both metadata and data using the LSID protocol.

4.1 Supporting Provenance Recording with Ontologies

Section 3 described the benefits gained from committing to formally specified common semantics. Therefore, before generating any provenance, we must spec-

[6] Taverna and FreeFluo are both open source projects available from `http://taverna.sourceforge.net` and `http://freefluo.sourceforge.net`

[7] Jena is an open source Semantic Web framework for Java including an RDF repository `http://jena.sourceforge.net/`

[8] An LSID Java server stack available for download at: `http://www-124.ibm.com/developerworks/projects/lsid`

ify the schema by which it will be generated and the vocabulary of concepts used to classify, at a domain level, the resources involved. myGrid used the DAML+OIL ontology language because much of the work predated the later development and release of the OWL specification. We are currently migrating to OWL.

The schema ontology is relatively small in size (<30 classes, <20 properties) and specifies (at a domain independent level) each class of resource and the properties that can be held between them for the four views of provenance that we record.

1. To describe the *in silico* experiment execution process, we have modelled classes such as `service_invocation` describing the web service process, `invocation_event` describing specific sub-events of an often multifaceted process and `data_item` identifying the data flow in and out of the web service process. The property `created_by` relates data items to the service invocation that created them.
2. The data derivation view (at a domain independent level) is currently simple and uses a single property `derived_from` to link web service outputs with their associated inputs. Any more specific relationships are domain dependent and go to form the knowledge view of the provenance web.
3. Domain dependent relationships between data are specific to an individual service and may even be specific to the context of a particular workflow. For example, the output of a BLAST service is a `similar_sequence_to` the input. Therefore the author of the workflow is responsible for determining exactly which properties are used. He/she chooses appropriate properties that go to form a statement *template*.
4. The experimental organisation view relates the specific workflow run with the person, project and organisation for which it was run and a simple description of the scientific motivation.

The larger domain specific ontologies[9] (yet still small in biomedical informatics terms) contain ∼600 classes and ∼50 properties. As many concepts have multiple parents (a feature hard to maintain by hand), each concept is assigned a single parent manually together with a formal concept definition in DAML+OIL. A description logic reasoner then uses the formal concept definitions to check consistency and infer additional subsumption relationships. A detailed description of this is beyond the scope of this paper. More information can be found in [12]. To be effective, the uses of the ontology must pervade the system:

1. The schema acts as a specification for the Taverna/FreeFluo workflow enactor of how to encode the provenance information. The enactor automatically generates RDF statements relating to information it has about what services were called, with what parameters, how long the service took to process the request, any sub events that occurred and the outcome in terms

[9] myGrid ontology available from `http://www.mygrid.org.uk/` in the ontology component section

of either result data or failure. All properties are sourced from the ontology, and all resources are typed with ontology concepts. LSIDs are generated corresponding to the workflow run, each service invocation, and sub-events. The relevant RDF statements are then published as the metadata of these resources. The enactor also relates this process-based view to a data derivation view. It directly links data items by first assigning each data item a unique LSID, and then generating RDF statements linking items together. As above, these statements form the metadata for the data items. The Taverna workflow workbench holds organisation information about the identity of the experiment, person, project,and organisation under which this workflow is run. Each is also identified by LSID, so RDF statements linking these to the specific workflow run can be published as additional metadata of these resources. Given the example workflow in Fig 2, users can semantically trace through its provenance logs: From where the *protein sequence* output of the workflow is *derived*, which service *created* this protein output, who is the *owner* of this protein sequence, etc. When integrating provenance logs from different runs, users are able to group all the logs that generate a *protein sequence* as outputs that were, for example, either *created* by a particular user or those that contain sequences that are *similar to* this protein output. Figure 1 shows the provenance graph generated by the workflow in Fig 2. The data graph is an inversion of the workflow graph, but further extended to co-ordinate with the process provenance.

2. The domain specific ontologies are used by the workflow enactor to classify at a domain level the data items produced or used by the workflow.
3. The domain specific ontologies provide properties that the scientist can use to relate inputs and outputs of a service to provide the knowledge view described above.
4. Both schema and domain specific ontologies are used by Haystack to aid visualization as described in the next section.

4.2 Visualizing RDF Based Provenance

Having generated these provenance data, we need to enable a scientist to perform the kind of provenance-based tasks described in Sect 2. Analysis and validation will often be based upon browsing. Yet this browsing needs to be done along many axes and from many viewpoints, according to the goal of the task. This means we need a generic, flexible basis for the scientist to interact with the provenance data.

By using the LSID protocol for retrieval of both data and metadata, we have been able to cleanly separate visualization components from those generating and storing provenance information, so much so that we have been able to reuse existing externally developed software with pre-existing LSID client functionality. The first such client we have used is the LSID Launchpad that renders RDF metadata associated with an LSID in the browser. Entering an LSID into Internet Explorer results in the LSID client finding the relevant LSID authority, and retrieving the associated metadata for that resource. As is common when

retrieving web documents, client applications can be registered to deal with particular MIME types so that when the data itself is retrieved it is displayed in the appropriate way.

Although a useful demonstration of LSID infrastructure, this style of RDF metadata visualization is intentionally naive. Users only see a restricted area of the graph one relationship wide. A few richer tools do exist to deliver Semantic Web technology to the user. One such Semantic Web browser, Haystack, from MIT, enables developers to provide customised views over RDF metadata [8]. To rapidly obtain a high level view of how resources interrelate, the user can view the provenance RDF as a labelled directed graph in Haystack, as with many other Semantic Web visualization tools.

Figure 5 shows an extract of such a graph, giving an example of how Haystack helps to visualize the *semantic* linking between these semantically enriched provenance resources for the experiment in Figure 2. In fact, we have found that the network of relationships is so dense that a complete rendering of the graph becomes difficult to interpret. We therefore filter out all but a small set of semantic relationships relevant to the user and task. In the case of Fig 5, only 13 RDF properties are being displayed: showing the relationship between data item (derived_from), the data flow between services (input, output), some organisation information of resources (e.g. created_by, design, group, project, etc), and some knowledge linking between data from different provenance logs (similar_sequence_to). By making use of Haystack's ability we can also separate these semantic links into four levels according to our provenance pyramid, which can be chosen from the view panel in Haystack, as shown in Fig 5.

However, we expect this graph based view of the whole provenance log will not provide a complete solution for the end user. The traditional web is itself a graph of documents related by hyperlinks, yet it is rarely rendered as such. Therefore, much more of our attention has been focused on providing a web browser metaphor for users to browse the provenance web. Haystack provides an application framework that allows the developer to render complex RDF subgraphs into user interface screens that behave like the much more familiar frames of web pages. Each class of resource can be associated with one or more of what Haystack terms "views". This is a specification, written in a novel high level programming language, Adenine, of how the metadata associated with that resource should be interpreted in order to construct a web page like interface for that resource. For more information see Quan *et al.* [13] and [8].

Our experience in using Haystack is promising. It has provided a rapid prototyping environment in which we can explore with our users the most effective way of interacting with RDF-based provenance information. By providing an application framework in which we can build up provenance views, it has enabled us to bypass the considerable effort needed to code core application and user interface components. When showing RDF-encoded provenance in Haystack to our collaborating bioinformaticians, they are satisfied that semantic relationships are made explicit when browsing these RDF logs in Haystack. By navigating and clicking, they can follow the hyperlinks between resources, directed by underlying

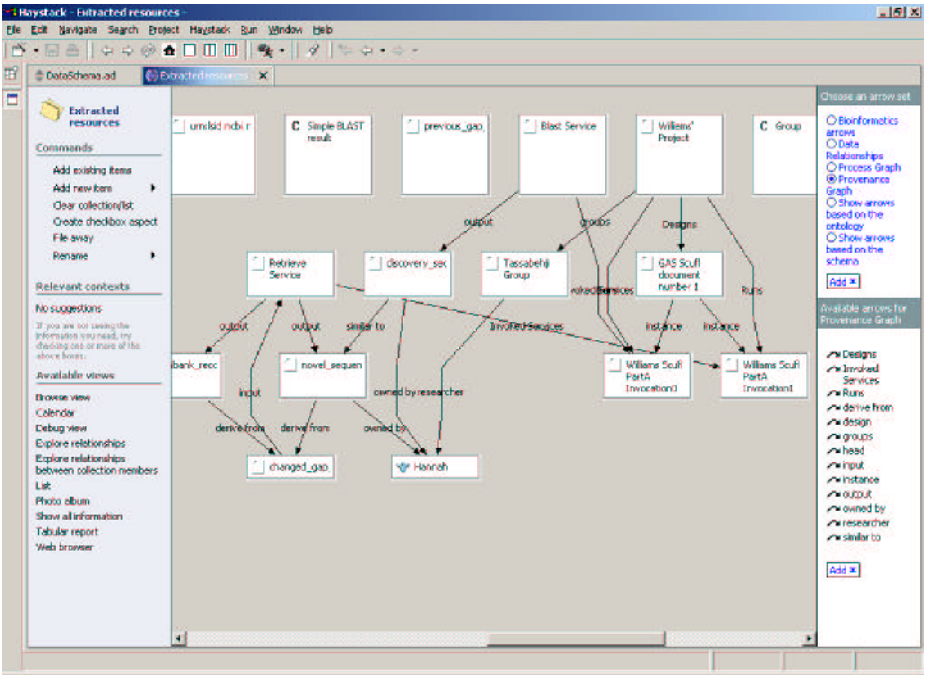

Fig. 5. Haystack screenshot of visualizing example provenance log.

semantic relationships. Also, when facing large amounts of data from provenance logs, users can choose different sets of predicates, in order to conceptually group provenance data in multiple ways according to the task at hand. For example, as introduced in Sect 2, when scientists at the stage of "debugging" experiment results, they could be interested in viewing provenance logs from a *process* aspect, to trace how experiments are executed. During the stage of "validity checking", a *data* view can be more supportive for scientists to track the data derivation path in an experiment.

Haystack's reliance on ontologies to drive the presentation of metadata fits well with the core role of ontologies in our provenance. Ontology classes associated with resources during provenance generation in the Taverna application have a direct impact on how those resources are displayed to the user in Haystack.

Haystacks LSID client functionality has also been of benefit. The user can enter an LSID of a provenance resource in Haystack and the metadata will be retrieved from the myGrid LSID metadata resolver and displayed appropriately.

However, within myGrid we have come across several issues with the Haystack and LSID approach. Haystack is a large desktop application requiring significant resources. This contrasts with bioinformaticians preference for light-weight web-based applications. Also using the architecture shown in Fig 4 we must rely on retrieving provenance metadata via the LSID protocol. As the protocol only allows for retrieval one resource at a time, to provide a high level expansive

view covering many resources we must enter into a piecemeal repetitive retrieval process and slowly build up the complete picture in the local Haystack RDF repository.

5 Discussion

The capture of *in silico* experiments as workflows and their outputs form a natural web of data. These can be described and identified by RDF and LSIDs respectively. In the myGrid project, these provenance data are recorded along four views: process, data derivation, organisation and knowledge. The knowledge or semantic component is needed in order to offer a common view over the heterogeneous semantic types within the provenance data and allows the provenance-based tasks to be accomplished.

Our implementation automatically traces the process and data derivation provenance and attaches the necessary organisational provenance to an experiment's data. The Taverna workflow environment offers the facility to attach knowledge level provenance through its template mechanism, supported by the myGrid ontology of bioinformatics services. This machinery also automatically provides RDF descriptions linked to the data through LSIDs. These provenance data can then be browsed by a scientist through the Haystack tool.

This success has, however, revealed several issues. As suggested in Sect 4.2, there is an issue of how these provenance data are presented to users — is a graph or a familiar web form appropriate? There is also a wider issue of usability, from the micro-scale of button and link labels, keystrokes and speed, to a macro-scale of effectiveness, efficiency and satisfaction in its support for provenance-based tasks in *in silico* experiments. Our current implementation and experience has offered no clues about the scalability of our handling and presentation of provenance data. As many users run many experiments, vast quantities of highly linked and annotated data will be generated. There is an issue about whether both the presentation and computational handling of all these data are scalable.

Given our addition of a semantic description of these data, we fully expect to be able to add *machine processing* of these data to the current human emphasis of our work. By fully implementing and managing an ontology for provenance data, such semantically supported processing over these data could be enabled by reasoning over the semantics associated with data resources. This would allow semantic querying of these logs, provide trust information about data based on logs, etc.

In our description of the use of provenance, we have presented a scenario where all users can browse all provenance data and even make annotations on other people's data. Obviously there are profound *security and authorisation* issues in such a scenario. Biologists are, as are other people, very security conscious about new data and associated discoveries etc. Many levels of authorisation need to be incorporated into the RDF paradigm in order to support access to subgraphs within an experiment repository.

These points do not contradict, but build upon the success of our demonstration of a Semantic Web of provenance for *in silico* experiments. The provision of such support for provenance is vital for the performance of *in silico* experiment and offers a rich test-bed for the concepts and technology of the Semantic Web.

Acknowledgements. The myGrid project, grant number GR/R67743, is funded under the UK e-Science programme by the EPSRC. The authors would like to acknowledge the other members of the myGrid team for their contributions.

References

1. Karp, P.: A Strategy for Database Interoperation. Journal of Computational Biology **2** (1995) 573–586
2. Program, H.G.: Genomics and its impact on science and society: A 2003 primer. Technical report, U.S. Department of Energy (2003)
3. Fox, G., Walker, D.: e-Science Gap Analysis. Technical report, Indiana University and Cardiff University, UK e-Science Center (2003)
4. Stevens, R., Robinson, A., Goble, C.: myGrid: Personalised Bioinformatics on the Information Grid. In: International Conference on Intelligent Systems in Molecular Biology. (2003)
5. Preus, M.: The Williams Syndrome: Objective Definition and Diagnosis. Clinical Genetics **25** (1984) 422–428
6. Hendler, J.: Communication: Enhanced Science and the Semantic Web. Science **299** (2003) 520–521
7. Martin, S., Senger, M., Niemi, M.: Life Sciences Identifiers second revised submission. Technical report, I3C (2003)
8. Quan, D., Karger, D.R.: How to make a semantic web browser. In: Proc. of the Thirteenth International World Wide Web Conference (WWW2004), ACM (2004) 255–265
9. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The making of a web ontology language. Journal of Web Semantics **1** (2003) 7–26
10. Cline, M., Shigeta, R., Wheeler, R., Siani-Rose, M., Kulp, D., Loraine, A.: The effects of alternative splicing on transmembrane proteins in the mouse genome. In: Pacific Symposium on Biocomputing. Volume 9. (2004) 17–28
11. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web Services Description Language (WSDL) 1.1 (2001) W3C Note.
12. Wroe, C., Stevens, R., Goble, C., Roberts, A., Greenwood, M.: A Suite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data. the International Journal of Cooperative Information Systems **12** (2003) 597–624
13. Quan, D., Huynh, D., Karger, D.R.: Haystack: A platform for authoring end user semantic web applications. In Fensel, D., Sycara, K., Mylopoulos, J., eds.: Proc. of the 2003 International Semantic Web Conference (ISWC 2003). Number 2870 in Lecture Notes in Computer Science, Springer (2003) 738–753