Proteome Data Integration: Characteristics and Challenges

K. Belhajjame[1], S.M. Embury[1], H. Fan[2], C. Goble[1], H. Hermjakob[4], S.J. Hubbard[1], D. Jones[3], P. Jones[4], N. Martin[2], S. Oliver[1], C. Orengo[3], N.W. Paton[1], A. Poulovassilis[2], J. Siepen[1], R.D. Stevens[1], C. Taylor[4], N. Vinod[4], L. Zamboulis[2], W. Zhu[4]

| [1] University of Manchester, Oxford Road, Manchester M13 9PL | [2] Birkbeck College, Malet Street, London WC1E 7HX | University College London, Gower Street, London WC1E 6BT | [4] EBI, Hinxton, Cambridge CB10 1SD |

Abstract

The aim of the ISPIDER project is to create a proteomics grid; that is, a technical platform that supports bioinformaticians in constructing, executing and evaluating *in silico* analyses of proteomics data. It will be constructed using a combination of generic e-science and Grid technologies, plus proteomics specific components and clients that embody knowledge of the proteomics domain and the available resources. In this paper, we describe some of our earlier results in prototyping specific examples of proteomics data integration, and draw from it lessons about the kinds of domain-specific components that will be required.

Introduction

Experimental proteomics is the study of the set of proteins produced by an organism, tissue or cell, with the aim of understanding the behaviour of these proteins under varying environments and conditions. As such, proteomics is an essential component of any comprehensive functional genomics study targeted at the elucidation of biological function. Its current popularity stems from the increased availability and affordability of reliable methods to study the proteome, such as 2D gel electrophoresis, multi-dimensional chromatography and mass spectrometry, as well as the ever growing numbers of tertiary structures and genome sequences emanating from structural genomics and sequencing projects respectively. Although other techniques in the functional genomics repertoire can be more accurately termed "genome-wide", proteomics remains a key area since proteins, the true gene products, rather than intermediaries such as mRNA transcripts, carry out the majority of biological "function" [8].

Experimental proteomics is a two stage process. In the first stage, the proteins in the sample are separated, generally by the technique of 2D gel electrophoresis or by liquid-phase chromatography. In the second stage, the separated sample is analysed, typically using a combination of mass spectrometry, to identify the masses of the protein fragments, and bioinformatics tools, to match the results with information about known proteins, in order to identify the protein(s) that are present within the sample.
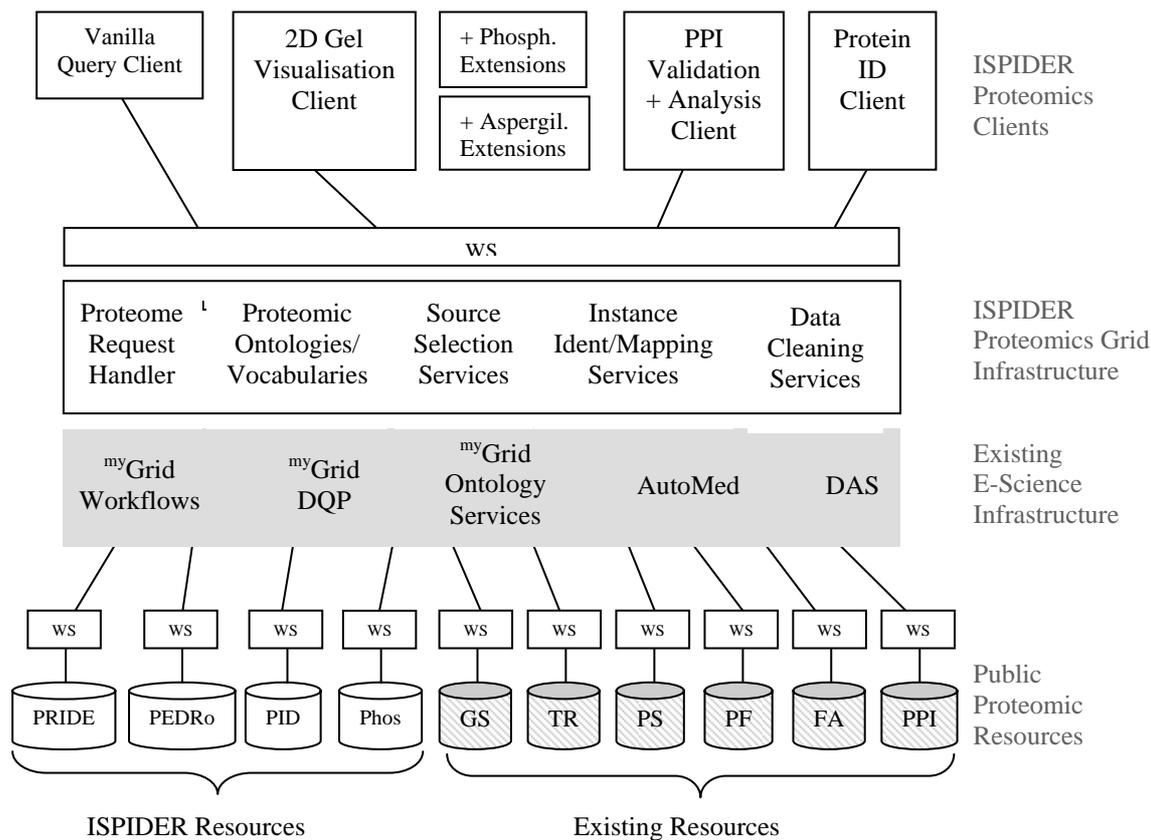
Current MS technology (such as MuDPIT – Multi-Dimensional Protein Identification Technology) allows hundreds of proteins to be identified in a single sample. These experimental results represent a rich and challenging data resource for functional genomics. As well as identification of protein expression patterns, it is also possible to use experimental proteomics to identify and validate protein-protein interactions and post-translational modification of proteins, as well as providing supporting evidence for predicted genes. In order to facilitate the sharing and use of this valuable proteomics data, a number of publicly accessible proteomics databases have been created, examples of which include PedroDB[1] and gpmDB[2]. The existence of such databases opens up many possibilities for new forms of bioinformatics analysis. However, most such analyses require the integration of experimental proteomics data with data and services offered by other resources – a non-trivial task that at present requires a significant amount of custom code to be written.

The aim of the ISPIDER[3] project is to provide an information grid dedicated to the creation of bioinformatics analyses for proteomics. Building on state-of-the-art tech-

---

[1] http://pedrodb.man.ac.uk:8080/pedrodb
[2] http://gpmdb.thegpm.org
[3] http://www.ispider.man.ac.uk

KEY: WS = web service wrapper, GS = genome sequence, TR = transcriptomic data, PS = protein structure, PF = protein family, FA = functional annotation, PPI = protein-protein interaction data

Figure 1: ISPIDER Architecture

nology for e-science and data integration, our goal is to provide an environment for constructing and executing analyses over proteomics data, and a library of proteomics-aware components that can act as building blocks for such analyses. This paper describes the early results from the first stages of the project, in which we have been identifying a set of candidate use cases that the ISPIDER grid should support. These use cases are allowing us to determine the particular integration challenges found in the proteomics domain, as well as discovering how far existing data integration technologies can resolve them.

The Scope of the Integration Task

A range of existing data resources have links to the proteome, and are therefore potential candidates for inclusion within ISPIDER. At the heart of proteome data lies the sequenced genome, in which the protein sequences of the cell's functional entities are encoded. Comprehensive catalogues of protein sequences and associated data are available in public resources such as Uniprot[4] and TrEMBL[5]. More recently, the International Protein Index[6] has been created, which offers trackable protein identifiers and the ability to store details of proteins that have been subject to multiple splicing events. Integration of this data with experimental proteome data offers the ability to verify predicted protein sequences by comparison with the sequences that are actually observed in samples. It also facilitates identification of post-translational modifications (i.e. changes that occur to a protein after it has been created within an organism and which thus cause the protein's sequence to differ from that encoded by its gene), which are often impossible to spot by analysis of sequence data alone. Similar verification at the gene level is possible by comparison with gene sequences stored in resources such as GenBank[7].

---

[4] http://www.ebi.ac.uk/swissprot
[5] http://www.ebi.ac.uk/trembl
[6] http://www.ebi.ac.uk/IPI
[7] http://www.ncbi.nlm.nih.gov/Genbank

Tertiary structure is another important input to the process of function prediction for genes and proteins, since it is more highly conserved in proteins than the sequence itself. Various resources exist that describe the structural properties of proteins, determined by either experimental or computational means, in terms of standard hierarchical fold classification systems, such as those available in CATH[8] and SCOP[9]. Integration with such annotated sources would allow the results of proteomics experiments to be grouped according to their structural similarities, which would allow key relationships to become visible within the data. Similarly, we can envisage integrating results from protein identification experiments with functional annotations, thus allowing evidence for or against certain functional hypotheses to be collected.

Further integration with databases of protein-protein interactions, gene expression data and observed post-translational modifications is also envisaged within the ISPIDER environment. The overall aim is to provide a collection of integrated comparative functional genomics tools that can examine diverse proteome datasets in different states and across species boundaries, without requiring programming or configuration effort to be wasted on tasks, which are common to many proteomics analyses.

The proposed architecture of the ISPIDER platform, acting as an intermediary between the resources just described and the specialist clients that we will build within the project, is shown in Figure 1. Whilst we hope to make advances within the ISPIDER project at each of the four architectural levels shown here, in this paper, we focus on our efforts to understand the combination of existing e-science technologies that can be used to support ISPIDER, and on our early understanding of the proteomics-specific components that will need to be developed.

Example Use Cases

In order to elicit the specific integration needs of proteomics-based bioinformatics, we have begun to identify a number of use cases that show how existing proteomics data can be combined in order to answer new kinds of biological question. By prototyping these use cases with existing e-science technologies, we are able to distinguish aspects that require further technological support from those that are already well catered for.

In order to illustrate this process, we describe two contrasting use cases that we have prototyped, and the lessons that we have drawn from each.

1) Value-Added Proteome Datasets

Protein identification is not an end in itself. Instead, the information about what proteins are present within a particular sample is used as input into a further analysis process that attempts to gather evidence for or against a particular biological hypothesis. Or, it may be used to suggest a number of new hypotheses that will then be tested by different means. In both these cases, it would be useful to be able to augment the raw protein identification results with additional information about the proteins that might help the scientist to better characterise the situation under study.

For example, it could be useful to be able to extend a protein identification result with the functional annotations that are currently associated with each protein found to be present in the cell. This could allow the biologist to notice that, for example, a variety of immune response suppressor proteins have been up-regulated in one of the samples under study. Similarly, the identification result could be augmented with details of their protein family or fold classifications.

All these are examples of typical data integration problems, in which largely non-overlapping data sets must be integrated based on a common key (the only area of overlap). We need to be able to extract the identifiers of the proteins in the identification result, use them to extract the auxiliary information that is required (e.g. the GO terms associated with those proteins) using a further query, and finally to recombine the two pieces of information in a useful and convenient way.

The first task in prototyping this use case was to determine whether any existing web services could provide all or part of this functionality. Software for performing protein identification matches already exists within the ISPIDER portfolio, in the shape of the Pepmapper system[10]. This is a peptide mass fingerprinting tool that uses mass spectrometry data produced by the digestion of a protein to

---

PeptideMasses

Established using Pepmapper

0,N

Contains references to other databases entries including InterPro, Pfam and DIP

SwissProt_Entry

EntryName: String
PrimaryAccessionNumber: String
SecondaryAccessionNumbers: String[]
ProteinName: String
GeneName: String
Sequence: String

0,N

CrossReference

DatabaseIdentifier: String
PrimaryIdentifier: String

0,1

ParentOf

Go_Term

AccessionNumber: String
Name: String
NameSpace: String

1

0,N

Indicates the function

0,N

Interpro_Entry

AccessionNumber: String
Name: String
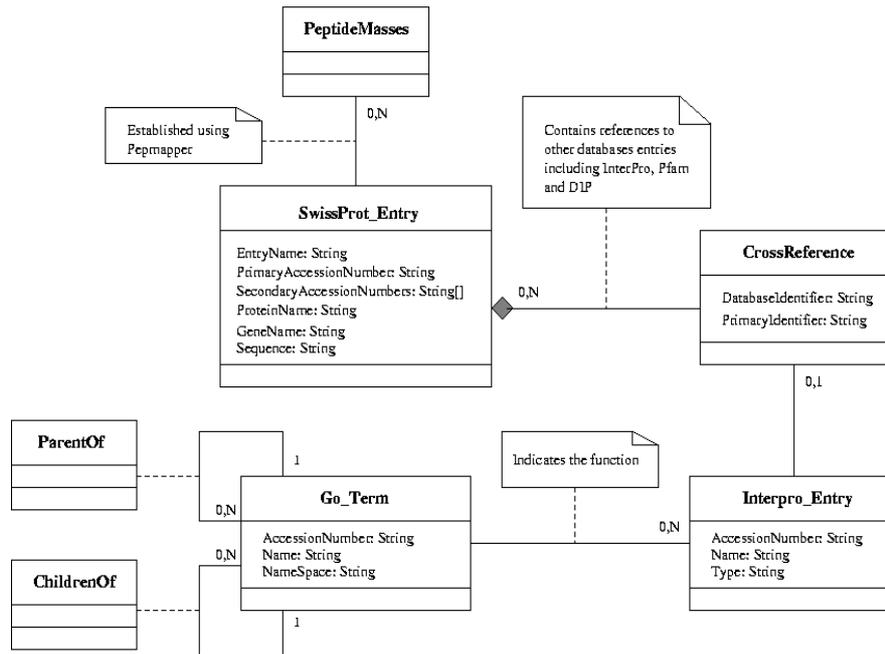Type: String

0,N

ChildrenOf

0,N

1

Figure 2: Data integration task expressed as a UML class

identify a match to a protein from a user-specified database [11]. The software was not initially wrapped as a web service, but it was a simple matter to do so for the purposes of ISPIDER. The resulting web service takes three input parameters: a list of the peptide masses to be identified; the name of the database which is to be searched for matches with the peptide masses; and the degree of error in the input masses that the user wishes to be taken into account when performing matches. The result of the web service is the Uniprot accession number of the protein that gives the best match with the given set of masses[11].

The second major requirement for this use case is the ability to retrieve the GO terms associated with a given project. The Gene Ontology (GO) project is a collaborative effort that addresses the need for consistent descriptions of gene products in different databases [9]. GO has developed three controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner. For instance, *cell growth and maintenance* is an example of a process-oriented GO term.

GO supplies a number of web services that provide operations useful in our context, namely:

- *GODBGetNameByID()*, which takes a Gene Ontology accession number as input and returns the associated (human readable) term name. For example, given the Gene Ontology accession number *GO:0005839*, this service would return the term *roteasome core complex (sensu Eukaryota).*

- *GODBGetClassification()*, which takes a Gene Ontology accession number and return the associated gene product role (e.g. *cellular_component*).

- *GODBGetParentOf()*, which navigates the GO hierarchy (or, strictly speaking, graph) to retrieve a list of the processes, functions or components situated directly above a given GO identifier.

In order to integrate the data produced by PepMapper and that provided by the GO web services, it was necessary to create a number of auxiliary web services that acted as the glue between these two systems. What is required is a navigation path from the Uniprot accession numbers provided by PepMapper and the GO accession terms required by GODB itself.

---

[11] This web service is available at http://rpc178.cs.man.ac.uk:8080/axis/services/pep_mapper?wsdl. In future, further versions may be provided, that return (for example) the full set of scored matches.
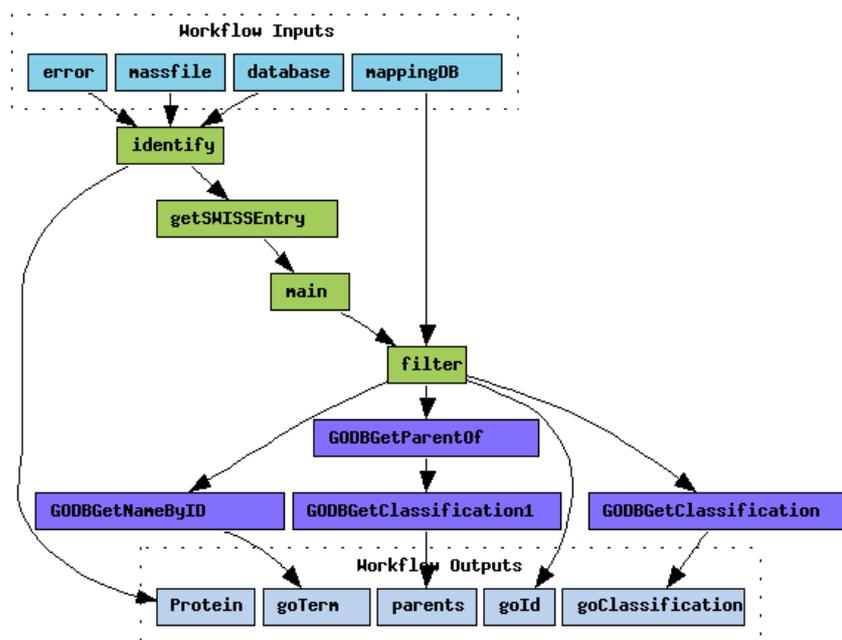
Figure 3: Protein identification and classification workflow

The connection can be established by the following navigational steps:

- The existing *GetEntry* web service is invoked through the operation *GetSWISSEntry()*, in order to obtain a Swissprot entry in flat file format, given a Uniprot accession number[12].

- We have implemented a new web service, *GetInterpro* that implements the operation *main()*, which takes a Swissprot entry and extracts the set of Interpro accession numbers embedded within it[13]. Interpro is a database that provides information on sequence function and annotation [4]. It integrates information from a numbers of secondary protein databases on functional sites and domains, such as PROSITE, PRINTS, SMART, Pfam and ProDom.

- We have implemented an additional web service, *Interpro2GO* that implements the operation *filter()*, which takes an Interpro accession number and returns the associated Gene Ontology accession numbers[14].

---

[12] This service is available at
http://xml.nig.ac.jp/wsdl/GetEntry.wsdl
[13] This service is available at
http://rpc178.cs.man.ac.uk:8080/axis/services/GetInterpro?wsdl
[14] This service is available at
http://rpc178.cs.man.ac.uk:8080/axis/services/Interpro2GO?wsdl

The data set involved in the full integration task is illustrated by the UML class diagram shown in Figure 2.

In order to orchestrate the collection of web services that have been identified and created for this use case, we have used the ^myGrid workflow engine, Taverna [6]. This software allows rapid creation of bioinformatics analyses, specified as data-directed workflows over existing web services, and therefore provides the last component of our prototype. The final workflow created is shown in Figure 3.

Here, *identify()* is the name of the web service interface to PepMapper. Its output becomes part of the output of the complete workflow, but is also passed into the web service that retrieves the Swissprot entry for the given protein. The resulting flat file is parsed and the InterPro accession numbers identified and passed to the next web service. This uses a mapping file that associates InterPro entries to the corresponding Go identifiers. Such a file is available from the Gene Ontology web site[15]. It adopts the following format: "InterPro:Interpro accession number > GO:GO term name ; GO:id". For example the entry InterPro:IPR000037 is associated the GO identifier GO:0003723. Using such a file, the

---

[15] The file used for the mapping is available at
http://www.geneontology.org/external2go/interpro2go

web service allows navigation from the InterPro accession numbers to the relevant set of GO terms. More human friendly versions of these terms are then extracted and combined with the original protein identifiers to present the full integrated result to the user[16]. It is also possible to examine the protein thus obtained using visualisation tools such as Dasty [5].

While the *Interpro2GO()* web service provides us with a means of connecting protein accession numbers and GO terms, it is not an entirely satisfactory solution, since it is based on a static mapping that will result in a certain number of false positive matches. However, other alternative navigation routes are becoming available. For example, a better *Interpro2GO()* web service could be constructed using the mappings created by the GOA project[17]. This set of mappings is curated, and has had a number of incorrect mappings removed. It is also being maintained, and includes a number of additional high quality GO associations that have been created by the GOA team. We therefore plan to make use of this resource in the next version of the use case implementation.

2) Genome-Focused Protein Identification

Currently, protein identification searches are performed indiscriminately over large data sets, such as Uniprot or IPI[18]. By searching over a large number of proteins, the number of false negatives is reduced. However, since protein identification is not a precise operation, it is also the case that the larger the data set search over, the more likely it is that false positives will also be introduced into the results. This means that time must be spent sifting through the results obtained, in order to find those that are both good matches and of relevance to the topic under study.

It is often the case that the biologists knows in advance what kinds of proteins will be relevant to his or her work. It would be more efficient in these cases to undertake a more focussed search, attempting matches only with

proteins from a particular species, for example, or found within a particular tissue type. By searching over smaller, more relevant data sets, the protein identification process will be made more efficient, and the amount of effort required to interpret the results (i.e. in filtering out irrelevant hits) will be reduced.

Ideally, we would like to implement this functionality by composing existing services, rather than having to extend an existing protein identification web service. In fact, the PepMapper web service is already flexible enough to support this, as the caller can specify the database to be used as a parameter to the service; it is not hard-coded into the service. Because of this, we can implement this use case simply by providing a pre-processing step that creates a temporary "database" containing the required proteins, and then directs PepMapper to match against it.

We have used the DQP system [1] to implement this pre-processing step. DQP is a query processor for the Grid that allows data from multiple distributed sources to be retrieved, filtered and combined just as if it were stored in a single local database. By combining it with PepMapper, we have been able to create within Taverna a new web service (illustrated in Figure 4) which takes as input a set of peptide masses, an error threshold and an OQL query describing the specific protein set that the masses are to be matched against.
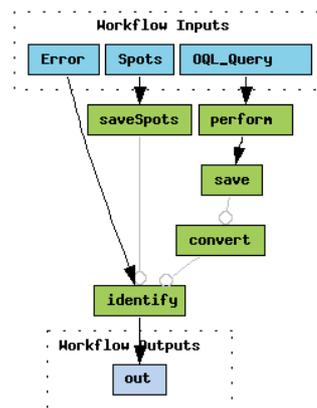

Figure 4: Focused protein identification

For example, the following OQL query describes the set of proteins of human origin:

```
select p.Name, p.Seq
from p in db_proteinSequences
where p.OS='HomoSapiens';
```

---

[16] Note that the workflow shown in Figure 3 contains two activities that correspond to invocation of the *GODBGetClassification()* service, but with distinct parameters. A number is appended to the name of the second call to allow it to be distinguished from the first.

[17] http://www.ebi.ac.uk/GOA/
[18] http://www.ebi.ac.uk/IPI/

Here, the protein database used is the IPI (i.e. the International Protein Index). We chose it, as it integrates sequence data from many resources including SwissProt, RefSeq and TAIR. `db_proteinSequences` is mapped to the database (or databases) of proteins to be queried, the `OS` attribute is the organism species, and the two attributes returned as the result of the query are the name and sequence of the selected proteins.

The OQL query is then evaluated using DQP, which fetches information about the resources involved in the query, produces an optimised query evaluation plan and then orchestrates the parallel distributed evaluation of each query sub-plan.

The results of query evaluation are returned from DQP as an XML document, containing (in this case) the name and sequences of the proteins to be used for protein identification. This document is cached locally (using the operation called *save()*) and is then converted into a file of the format required by the PepMapper web service (i.e. Fasta format). We have had to construct a web service to perform this transformation ourselves (*convert()* in Figure 4). The transformed file is then used as the input to the protein identification process, as if it were a normal protein database.

Conclusions and Future Directions

Our experiences in implementing the use cases described in this paper have demonstrated that the generic e-science technologies currently available go a long way towards providing a rapid prototyping environment for bio-informatics analyses. In particular, the combination of the workflow orchestration services provided by Taverna and the distributed query processing facilities provided by DQP is very powerful. Web service orchestration alone forces the user to write custom code for many data manipulation tasks that could be provided much more cheaply and efficiently by standard query processing facilities. On the other hand, a query processor alone forces the user to write custom code for the non-declarative elements of the analyses. Given the diversity of the biological resources on offer today, these non-declarative elements make up a significant element of the analyses, since we must cope with differences in file types and other more serious kinds of semantic mismatch.

Even given the combination of web service orchestration and query processing, we still had to write a certain amount of custom code for ourselves. Some of this was simple housekeeping, relating to the processing and transforming of the outputs of web services so that they are in the appropriate form to be used as inputs to other services. For example, in use case 1, it was necessary to process Swissprot files in order to extract just one piece of information, the set of InterPro accession numbers. And in use case 2, custom code was required to transform the XML documents produced by DQP into the Fasta files required by PepMapper. The need for this kind of custom code arises due to the lack of standards and conventions in the creation of web services. It is generally simple to write, but tedious. Hull et al. have proposed a facility for discovery of such services, to allow sharing and reuse [3]. The development of standard data formats and the use of generic e-science technologies such as DQP will also help to alleviate the need for this kind of custom code.

A second kind of custom code required for our use cases is representative of a more serious problem in data integration: that of mismatches between keys and identifiers. For example, in the first use case, it was necessary for us to find a way of joining two data sets with incompatible keys. This is a classic problem in data integration, and usually requires details knowledge of the domain and resources in question to resolve. In our case, we were able to find an intermediate identifier (InterPro) that allowed us to perform the navigation steps required. This suggests that a service which collects expert knowledge about relationships between the keys used in different systems, and that can advise on ways of navigating between systems, might be a useful facility. Our future work in ISPIDER will allow us to explore mechanisms like this in a domain-specific context.

The next steps for ISPIDER will involve prototyping of further use cases and building specialist clients for certain key applications. In particular, we are currently examining a very different kind of data integration than that demonstrated by the two web services described in this paper. Whereas these involved the integration of largely non-overlapping data sets (i.e. where the overlap occurs only in terms of the keys used to join the data sets), it is often also necessary to integrate databases that contain the same kind of information, but structured in very different ways. For example, several databases have been created by independent parties to store the results of

protein identification experiments. It would be useful if these could be queried together, so that the results of experiments by different scientists concerning the same tissue type could be compared, or so that all the experiments that have identified a particular protein could be examined by the biologist.

The schemas used by each of these systems are very different (e.g. compare the schema of Pedro [10] with that of gpmDB [2]). The distributed query processing powers of DQP can provide location transparency for the integration of such databases, but it is not equipped to deal with the degree of structural heterogeneity found within these systems. We are therefore working to combine DQP with a schema transformation system, AutoMed [7], which will be able to resolve the structural heterogeneities, and provide a common query interface to a range of proteomics databases. This will remove the need for custom components that translate between different file and database formats.

As a result of our efforts, we hope to provide a range of useful web services for the proteomics community, as well as learning lessons for the e-science community about the usefulness of the generic technologies currently on offer, and the kinds of domain-specific components that need to be created before their power can be harnessed in specific application areas.

References

[1] MN Alpdemir *et al., OGSA-DQP: a Grid Service for Distributed Querying on the Grid*, in Proc. of EDBT04, pp. 858-861, 2004.

[2] R Craig, JP Cortens, and R Beavis, *Open source system for analysing, validating, and storing protein identification data*, Journal of Proteome research, 3:1234-1242, 2004.

[3] D. Hull, R. Stevens, P. Lord, and C. Goble, *Integrating bioinformatics resources using shims*. In the 12th International Conference on Intelligent Systems for Molecular Biology, Glasgow, UK, 2004.

[4] Interpro Consortium, *Interpro – an integrated documentation resource for protein families, domains and functional sites*, Journal of Bioinformatics, 16(12), 2000.

[5] P. Jones *et al, Dasty and Uniprot DAS: a perfect pair for protein feature visualization*, Journal of Bioinformatics, Application notes, 21(14), 2005.

[6] P Lord *et al., Applying Semantic Web Services to Bioinformatics: Experiences Gained, Lessons Learnt*, in Proc. of ISWC04, pp. 350-364, Springer-Verlag LNCS, 2004.

[7] PJ McBrien & A Poulovassilis, *Schema Evolution in Heterogeneous Database Architectures, a Schema Transformation Approach*, in Proc. of CAiSE02, Springer-Verlag LNCS, pp. 484-499, 2002.

[8] SG Oliver, *Guilt by Association goes Global*, Nature 403:601-603, 2000.

[9] Open Biomedical Ontologies, *Gene Ontology*, 2004. Accessible at: http://www.geneontology.org

[10] CF Taylor *et al*, *A systematic approach to modelling, capturing, and disseminating proteomics experimental data*, Nature, 21:247-254, 2003.

[11] University of Manchester, *PepMapper – Protein search tool, 2004*. Accessible at: http://wolf.bms.umist.ac.uk/mapper