J. Biomedical Science and Engineering, 2008, 1,204-209

Using schema transforation pathways for biological data integration

Hao Fan & Fei Wang

International School of Software, Wuhan University, Hubei, P.R. China, 430072. Email:{hfan, feiwangg}@whu.edu.cn), Tel.: +86 27 6877-8605.
Received on September 10, 2008; revised and accepted on September 28, 2008

ABSTRACT

In web environments, proteomics data integrationin the life sciences needs to handle the problem of data conflicts arising from the heterogeneity of data resources and from incompatibilities between the inputs and outputs of services used in the analysis of the resources. The integration of complex, fast changing biological data repositories can be potentially supported by Grid computing to enable distributed data analysis. This paper presents an approach addressing the data conflict problems of proteomics data integration. We describe a proposed proteomics data integration architecture, in which a heterogeneous data integration system interoperates with Web Services and query processing tools for the virtual and materialised integration of a number of proteomics resources, either locally or remotely. Finally, we discuss how the architecture can be further used for supporting data maintenance and analysis activities.

Keywords: Evoked potentials (EPs), Alpha stable distribution, Blind source separation, Minimum dispersion (MD), Fractional lower order statistics (FLOS)

1. INTRODUCTION

In the life sciences, along with the deepening investigation into human genome, especially into their functionalities, study on proteomics becomes an important key issue of life sciences research. Since protein is the reflector of genome functionalities, research on proteomics is to investigating the structures and functions of protein, in order to interpret the variation mechanism of a life in physiologies or pathologies. Issues relating to protein innate existence forms and activity patterns, such as interpreting gene modifiers, protein interactions and configurations, require the solution of study on the protein complement of the genome, i.e. proteomics, which is also an essential component of any comprehensive functional genomics study targeted at the elucidation of biological functions.

Proteome databases and genome pools are generally used for proteomics research. However, global protein expression analysis refers to any experiment in which the expression of all genes is monitored simultaneously, which generate large amounts of data, but there is no universal system for the description of gene expression profiles. Global protein expression data are obtained predominantly as signal intensities on 2*D* protein gels.

A large amount of biological information is available over the Internet, but the data are widely distributed and it is therefore necessary to have efficient mechanisms for data integration and data retrieval. Grid computing technologies are becoming established which enable distributed computational and data resources to be accessed in a service based environment. These technologies offer the possibility of analysis of complex distributed post-genomic resources. To support transparent access, however, such heterogeneous resources need to be integrated rather than simply accessed in a distributed fashion.

This paper introduces a proteomics data integration architecture, using a heterogeneous data integration system interoperates with Web Services and query processing tools for the virtual and materialised integration of a number of local and remote proteomics resources. We also discuss how the architecture can be further used for supporting data maintenance and analysis activities, *i.e.* processing user queries, tracing data linages, and maintaining data incrementally.

Paper outline: Section II gives an overview of protein database resources available in the Internet. Section III presents a web-based heterogeneous data integration architecture, including the BAV data integration technologies and Web services, the transformation pathways for creating the global schema, and the key issues addressed in the architecture for applying to biological data integrations. Finally, Section IV gives our conclusions and directions of future work.

2. OVERVIEW OF PROTEOME DATABASE RESOURCES

Proteome databases receive more and more attentions in proteomics research, which strives to provide a high level of annotation, such as the description of the function of a protein, its domains structure, post-translational modifications variants, etc., a minimal level of redundancy and high level of integration with other databases [2].

2.1. Protein Sequence Databases

2.1.1. General sequence databases

EXProt (see http://www.cmbi.kun.nl/EXProt/) is aiming at including only proteins with an experimentally verified function, which is a non-redundant protein database containing a selection of entries from genome annotation projects and public databases. Its each entry has a unique ID number and contains information about the species, amino acid sequence, functional annotation.

UniProt (see http://www.uniprot.org) is formed by uniting activities of the Swiss-Prot1, TrEMBL2, and PIR3 protein databases, which provides a central resource on protein sequences and functional annotation with three database components, each addressing a key need in protein bioinformatics.

TCDB (see http://www.tcdb.org) is a curated relational database containing sequence, classification, structural, functional and evolutionary information about transport systems from a variety of living organisms. TCDB is a repository for information compiled from more than 10; 000 references, encompassing approximately 3; 000 representative transporters and putative transporters, classified into over 400 families.

2.1.2 Protein properties

iProLINK (see http://pir.georgetown.edu/iprolink) facilitates text mining research in the area of literature-based database curation, named entity recognition, and protein ontology development. This collection of annotated data sources can be utilized by computational and biological researchers to explore **literature information on proteins and their features.**

PFD (see http://www.foldeomics.org/pfd/) has a database structure allows visualization of folding data in a useful and novel way, with aims of facilitating data mining and bioinformatics approaches, which is a searchable collection of all annotated structural, methodological, kinetic and thermodynamic data relating to experimental protein folding studies.

PINT (see http://www.bioinfodatabase.com/pint/) contains data of several thermodynamic parameters along with sequence and structural information, experimental conditions and literature information. Each entry contains numerical data for the free energy change, dissociation constant, association constant, enthalpy change, heat capacity change and so on of the interacting proteins upon binding, which are important for understanding the mechanism of protein-protein interactions.

2.1.3 Protein sequence motifs

PROSITE (see http://www.expasy.org/prosite) is a large collection of biologically meaningful signatures described as patterns or profiles. Each signature is linked to a documentation that gives useful biological information on the protein family, domain, or functional site identified by the signature.

Blocks (see http://blocks.fhcrc.org) are ungapped multiple alignments corresponding to the most conserved regions of proteins, which consists of blocks constructed from documented families of related proteins. A blocks multiple alignment consists of ungapped conserved regions separated by unaligned regions of variable size.

PRINTS (see http://www.bioinf.man.ac.uk/dbbrowser /) houses a collection of protein family fingerprints, which may be used to make familial and tentative functional assignments for uncharacterised sequences. It specializes in the provisional of hierarchical classifications of protein superfamilies, allowing fine-grained diagnoses, and provides the bulk of the hierarchical family annotation in InterPro. PRINTS also underpins the Blocks database from Seattle and eMOTIF resource from Stanford.

2.1.4. protein domains and classifications:

iProClass (see http://pir.georgetown.edu/iproclass/) provides an integrated view of protein information and serves as a bioinformatics framework for data integration and associative analysis of proteins. It presents value-added descriptions of all proteins in UniProtKB and contains comprehensive, upto- date protein information derived from over 90 biologicaldatabases. The source databases include those of protein sequence, family, function, pathway, protein-protein interaction, complex, post-translational modification, protein expression, structure and structural classification, gene and genome, gene expression, disease, ontology, literature, and taxonomy.

2.2. Protein Structure Databases

PDB (see http://www.rcsb.org/pdb/) is the single worldwide archive of structural data of biological macromolecules. A description of the architecture and functionality of the systems used to collect, archive, distribute, and query the data were described previously [3].

SCOP (see http://scop.mrc-lmb.cam.ac.uk/scop) provides a comprehensive and detailed description of the evolutionary and structural relationships of the proteins of known structure. It embodies an evolutionary classification produced by human experts. This allows users to use a theory of protein evolution that encompasses our knowledge of the great variety, and the full extent, of the different types of changes that take place during evolution.

CATH (see http://www.biochem.ucl.ac.uk/bsm/cath new) currently contains 34*;* 287 domain structures classified into 1*;* 383 superfamilies and 3*;* 285 sequence families. Each structural family is expanded with domain sequence relatives recruited from GenBank using a variety of efficient sequence search protocols and reliable thresholds. New sequence search protocols have been designed, based on these intermediate sequence libraries, to allow more regular updating of theclassification.

2.3. Proteomics Resources

PEDRo (see http://pedrodb.man.ac.uk:8080/pedrodb) provides access to a collection of descriptions of experimental data sets in proteomics [8]. It was one of the firstdatabases used for storing proteomics experimental data and has also been used as a format for exchanging

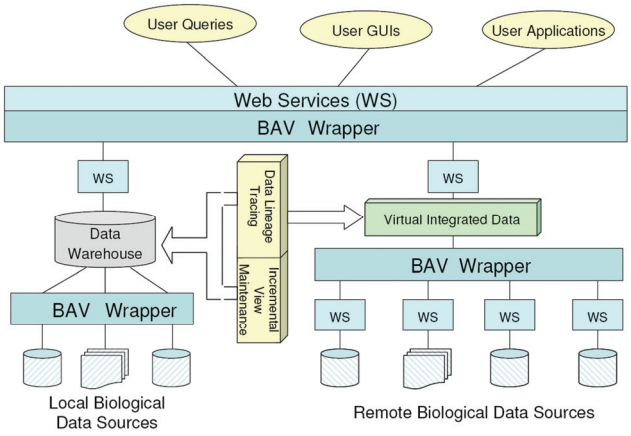

Figure1. A Proteomics Data Integration Architecture.

proteomics data. The integration of such a comprehensive data model, in addition to its data content, provides means for capturing a significant proportion of the proteomic information that is captured by other proteomics repositories.

gpmDB (see http://gpmdb.thegpm.org) is a publicly available database with over 2*;* 200*;* 000 proteins and almost 470*;* 000 unique peptide identifications [4]. Although the gpmDB is restricted to minimal information relating to the protein/peptide identification, it provides access to a wealth of interesting and useful peptide identifications from a range of different laboratories and instruments.

PepSeeker (see http://nwsr.smith.man.ac.uk/pepseeker) is a database targeted directly at the identification stage of the proteomics pipeline. It captures the identification allied to the peptide sequence data, coupled to the underlying ion series and as a result it is a comprehensive resource of peptide/proteinidentifications [11]. The repository currently holds over 50*;* 000 proteins and 50*;* 000 unique peptide identifications.

PRIDE (see http://www.ebi.ac.uk/pride/) is a database of protein and peptide identifications that have been described in the scientific literature. These identifications will typically be from specific species, tissues and sub-cellular locations, perhaps under specific disease conditions and may be annotated with supporting mass spectra. PRIDE can be searched by experiment accession number, protein accession number, literature reference and sample parameters including species, tissue, sub-cellular location and disease state. Data can be retrieved as machine-readable PRIDE or mzData XML, or as human-readable HTML.

3. DATA INTEGRATION ARCHITECTURE

Figure 1 illustrates a heterogeneous data integration architecture in Web environments. In this architecture, remote data sources use web service platforms producing data exchange and access processes with external users. The BAV data integration system interoperates with web services enabling query processing tools for the virtual and materialised integration of a number of distributed data resources.

Specially, in the scenario of integrating local databases, BAV wrappers apply to the local data sources directly, so that extracting data and data structure from data sources and producing global user queries. On the other hand, in the scenario of web data integrations, BAV wrappers exchange data with web services so that access the remote data sources. Both virtual and materialised integrated views can be created by the data integration system.

In this section, we discuss how this architecture can be proposed used for integrating proteomics data resources.

3.1. Data Integration Technologies

BAV Data Integration. Up to now, most data integration approaches have been either *global-as-view* (GAV) or *localas- view* (LAV). In GAV, the constructs of a global schema are described as views over local schemas1. In LAV, the constructs of a local schema are defined as views over a global schema. One disadvantage of GAV and LAV is that they do not readily support the evolution of both local and global schemas. In particular, GAV does not readily support the evolution of local schemas while LAV does not readily support the evolution of global schemas. Furthermore, both GAV and LAV assume one common data model for the data transformation and integration process, typically the relational data model.

Both-as-view (BAV) is a new data integration approach based on the use of reversible sequences of primitive schema transformations [10]. From these sequences, it is possible to derive a definition of a global schema as a view over the local schemas, and it is also possible to derive a definition of a local schema as a view over a global schema. BAV can therefore capture all the semantic information that is present in LAV and GAV derivation rules. A key advantageof BAV is that it readily supports the evolution of both localand global schemas, allowing transformation sequences and schemas to be incrementally modified as opposed to having to be regenerated.

Another advantage is that BAV can offers the capability to handle virtual, materialised and indeed hybrid data integration across multiple data models. This is because BAV supports a low-level *hypergraph-based data model (HDM)* and provides facilities for specifying higher-level modelling languages in terms of this HDM. For any modelling language *M* specified in this way, the approach provides a set of primitive schema transformations that can be applied to schema constructs expressed in *M*. In particular, for every construct of *M* there is an add and a delete primitive transformation which add to/delete from a schema an instance of that construct. For those constructs of *M* which have textual names, there is also a rename primitive transformation. BAV schemas can be incrementally transformed by applying to them a sequence of primitive transformations, each adding, deleting or renaming just one schema construct.

Each add and delete transformation is accompanied by a query specifying the extent of the added or deleted construct in terms of the rest of the constructs in the schema. This query is expressed in a functional query language, IQL, which is a comprehensions-based language and we refer the reader to the reference [9] for details of its syntax, semantics and implementation. Such languages subsume query languages such as SQL-92 and OQL in expressiveness [1].

Also available are extend and contract primitive transformations which behave in the same way as add and delete except that they state that the extent of the new/removed construct cannot be precisely derived from the other constructs present in the schema. More specifically, each extend and contract transformation takes a pair of queries that specify a lower and an upper bound on the extent of the construct. The lower bound may be Void and the upper bound may be Any, which respectively indicate no known information about the lower or upper bound of the extent of the new construct. The queries supplied with primitive transformations can be used to translate queries or data along a transformation pathway

A sequence of primitive transformations from one schema S_1 to another schema S_2 is termed a *pathway* from S_1 to S_2. All source, intermediate, and integrated schemas, and the pathways between them, are stored in a Schemas & Transformations Repository.

Web Services. Web services provide a standard means of interoperating between different software applications (see [4] http://www.w3.org/TR/ws-arch/), running on a variety of platforms and/or frameworks. A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically Web Services Description Language, WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

3.2. Data Integration Using Transformation Pathways

The aim of the proteomics data integration is to build technologies providing an environment for integrating proteomics data, constructing and executing analyses over such data, and a library of proteomics-aware components that can act as building blocks for such analyses. A web-based architecture is enabling existing proteomics data resources, creating new resources, producing middleware technologies for the integration of protein data resources — including tools for data integration, data analysis, producing user queries, visualisation applications and other types of client for biologist end users.

This section gives examples of using transformation pathways for creating a global schema. Supposing the table ProteinHit (all peptides matched, expect, score, threshold,protein, peptidehit), indicated as *hh*uproteinhit*ii*, in the global schema, which is composed of data originally from four different databases, PEDRo, gpmDB, PepSeeker and Pride. The following transformation pathways are used to create the *hh*uproteinhit*ii* schema, in which id2lsid is an IQL build-in function used to transform id features in source schemas into lsid feature in the global schema.

1. Creating ProteinHit from PEDRo database:

```
add  ⟨⟨uproteinhit⟩⟩
     [id2lsid k|k ← ⟨⟨proteinhit⟩⟩]
add  ⟨⟨uproteinhit, lsid⟩⟩
     [{k, k}|k ← ⟨⟨uproteinhit⟩⟩]
add  ⟨⟨uproteinhit, all_peptides_matched⟩⟩
     [{id2lsid k, x}|{k, x} ← ⟨⟨proteinhit, all_peptides_matched⟩⟩]
ext  ⟨⟨uproteinhit, expect⟩⟩ Void
add  ⟨⟨uproteinhit, score⟩⟩
     [{id2lsid k, x}|{k, x} ← ⟨⟨proteinhit, score⟩⟩]
ext  ⟨⟨uproteinhit, threshold⟩⟩ Void
add  ⟨⟨uproteinhit, protein⟩⟩
     [{id2lsid k, x}|{k, x} ← ⟨⟨proteinhit, protein⟩⟩]
add  ⟨⟨uproteinhit, peptidehit⟩⟩
     [{id2lsid k, x}|{k, x} ← ⟨⟨proteinhit, peptidehit⟩⟩]
```

2. Creating ProteinHit from gpmDB database:

```
add  ⟨⟨uproteinhit⟩⟩
     [id2lsid k|k ← ⟨⟨protein⟩⟩]
add  ⟨⟨uproteinhit, lsid⟩⟩
     [{k, k}|k ← ⟨⟨uproteinhit⟩⟩]
ext  ⟨⟨uproteinhit, all_peptides_matched⟩⟩ Void
add  ⟨⟨uproteinhit, expect⟩⟩
     [{id2lsid k, x}|{k, x} ← ⟨⟨protein, expect⟩⟩]
ext  ⟨⟨uproteinhit, score⟩⟩ Void
ext  ⟨⟨uproteinhit, threshold⟩⟩ Void
add  ⟨⟨uproteinhit, protein⟩⟩
     [{id2lsid k, x}|{k, x} ← ⟨⟨protein, proseqid⟩⟩]
ext  ⟨⟨uproteinhit, peptidehit⟩⟩ Void
```

3. Creating ProteinHit from PepSeeker database:

```
add  ⟨⟨uproteinhit⟩⟩
     [id2lsid k|k ← ⟨⟨proteinhit⟩⟩]
add  ⟨⟨uproteinhit, lsid⟩⟩
     [{k, k}|k ← ⟨⟨uproteinhit⟩⟩]
ext  ⟨⟨uproteinhit, all_peptides_matched⟩⟩ Void
ext  ⟨⟨uproteinhit, expect⟩⟩ Void
add  ⟨⟨uproteinhit, score⟩⟩
     map λ{k, x}.{id2lsid k, x} (distinct [{x1, x2}|
     {k1, x1} ← ⟨⟨proteinhit, proteinscore⟩⟩;
     {k2, x2} ← ⟨⟨proteinscore, Score⟩⟩; x1 = k2])
ext  ⟨⟨uproteinhit, threshold⟩⟩ Void
add  ⟨⟨uproteinhit, protein⟩⟩
     [{id2lsid k, x}|{k, x} ← ⟨⟨proteinhit, proteinID⟩⟩]
add  ⟨⟨uproteinhit, peptidehit⟩⟩
     [{id2lsid k, x}|{k, x} ← ⟨⟨proteinhit, fileparameters⟩⟩]
```

4. Creating ProteinHit from Pride database:

```
add  ⟨⟨uproteinhit⟩⟩
     [id2lsid k|k ← ⟨⟨pride_identification⟩⟩]
add  ⟨⟨uproteinhit, lsid⟩⟩
     [{k, k}|k ← ⟨⟨uproteinhit⟩⟩]
ext  ⟨⟨uproteinhit, all_peptides_matched⟩⟩ Void
ext  ⟨⟨uproteinhit, expect⟩⟩ Void
add  ⟨⟨uproteinhit, score⟩⟩
     [{id2lsid k, x}|{k, x} ← ⟨⟨pride_identification, Score⟩⟩]
add  ⟨⟨uproteinhit, threshold⟩⟩
     [{id2lsid k, x}|{k, x} ← ⟨⟨pride_identification, threshold⟩⟩]
ext  ⟨⟨uproteinhit, protein⟩⟩ Void
ext  ⟨⟨uproteinhit, peptidehit⟩⟩ Void
```

3.2. Using Schema Transformation Pathways

In the previous section we showed how schema transformation pathways can be used for expressing the process generating a global schema. In this section, we discuss how the transformation pathways can be used for the following further activities.

1) Schema Evolution: A recurring issue within the data integration architecture is that the source schemas will change as the owners of these autonomous data sources evolve them over time. Changes in global schemas may also be needed in order to support new requirements of the client components. An advantage of the BAV approach over GAV or LAV data integration is that it readily supports the evolution of both source and global schemas by allowing transformation pathways to be extended — this means that the entire integration process does not have to be repeated, and the schemas and pathways can instead be 'repaired'. This process can be handled largely automatically, except in cases where new information content is being added to schemas where domain or expert human knowledge is needed regarding the semantics of new schema constructs.

2) Transforming Data Schemas: In previous work [7], we show that transformation pathways can be used for expressing the process generating global schemas in biological data integration environments. Each transformation step is accompanied by an add/delete operation with a query specifying the extent of the added or deleted construct in terms of the restof the constructs in the original schema.

3) Processing User Queries: An WS wrapper imports schema information from any data source, via web services, into a metadata repository. Thereafter, schema transformation/ integration functionality can be used to create one or more virtual global schemas, together with the transformation pathways between these and the BAV representations of the data source schemas. Queries posed on a virtual global schema can be submitted to a query processor, and this interacts with 5 web services via WS wrappers to evaluate these queries. After the integration of the data sources, the user is able to submit to the query processor a query to be evaluated with respect toa virtual global schema.

4) Tracing Data Lineage: In the data integration architecture, proteomics data is integrated from distributed, autonomous and heterogeneous data sources, in order to enable analysis and mining of the integrated information. However, in addition to analyzing the data in the integrated Grid, we sometimes also need to investigate how certain integrated data was derived from the data sources, which is the problem of *data lineage tracing* (DLT).

In [5], we present a DLT approach which is to use the individual steps of these pathways to compute the lineage data of the tracing data by traversing the pathways in reverse order one step at a time. In particular, suppose a data source LD with schema LS is transformed into a global database GD with schema GS, and the transformation pathway LS → GS is $ts_1, ..., ts_n$. Given tracing data *td* belonging to the extent of some schema construct in GD, we firstly find the transformation step ts_i which creates that construct and obtain *td*s lineage, dl_i, from ts_i. We then continue by tracing the lineage of dl_i from the remaining transformation pathway $ts_1, ..., ts_{i-1}$. We continue in this fashion, until we obtain the final lineage data from the data source LD.

5) Maintaining Integrated Data Incrementally: The global schema might be materialised by creating an integrated database. A problem relating to materialised integrated data is *view maintenance*. The materialised integrated data need to be maintained either when the data of a data source changes, or if there is an evolution of a data source schema. If the source data is updated, the integrated data has to be refreshed also so as to keep it up-to-date.

If a materialised construct *c* is defined by an query *q* over other materialised constructs, [6] gives formulae for incrementally maintaining *c* if one its ancestor constructs c_a has new data inserted into it (an increment) or data deleted from it (a decrement). We actually do not use the whole view definition *q* generated for *c*, but instead track the changes from c_a through each step of the pathway. In particular, at each add or rename step we use the set of increments and decrements computed so far to compute the increment and decrement for the schema constructed being generated by this step of the pathway.

6) Extendable Exploitations: In peer-to-peer (P2P) systems, a number of autonomous servers, or *peers*, share their computing resources, data and services. The loose and dynamic association between peers has meant that, to date, P2P systems have been based on the exchange of files identified by a limited set of attributes. The lack of information about the data within these files makes it impossible to support general-purpose mechanisms by which peers can exchange and translate heterogeneous data. How the heterogenous data integration architecture can be used to integrate proteome data in different peers would be the extendable exploitation of our work.

4. CONCLUSION

In this paper, we give a brief overview of protein database resources available in the Internet, and present an architecture combining web services and a data integration software tools over the autonomous data resources which enables distributed query processing together with the resolution of semantic heterogeneity over autonomous data resources. We also discuss the key issues of applying the architecture to biological data integrations, namely schema evolution, transforming data schemas, processing user queries, tracing data lineage, maintaining integrated data incrementally, and extendable exploitations.

The final platform will provide researchers with more information than any of the resources alone, so allowing them to perform analyses that were previously prohibitively difficult or impossible. This integration process

both builds on and provides impetus to the development of data standards in the proteomics and related domains.

Future work incudes designing and implementing wrappers for extracting data and data structure over web services, implementing DLT and IVM algorithms over the architecture, and investigating into extending the system into P2P environments.

REFERENCES

[1] P. Buneman *et al*. (1994) Comprehension syntax. *SIGMOD Record*, 23(1):87–96.

[2] A. Bairoch and R. Apweiler. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res*, 28:45–48.

[3] H.M. Berman, J. Westbrook, and *et al*. (2000) The Protein Data Bank. *Nucleic Acids Res*, 28:235–242.

[4] R. Craig, J.P. Cortens, and R.C. Beavis. (2004) Open source system for analyzing, validating, and storing protein identification data. *Journal of Proteome Research*, 3(6).

[5] H. Fan and A. Poulovassilis. (2005) Using schema transformation pathways for data lineage tracing. In *Proc. BNCOD'05, LNCS 3567*, pages 133–144.

[6] H Fan. (2005) *Investigating a Heterogeneous Data Integration Approach for Data Warehousing*. PhD thesis, Birkbeck College, University of London.

[7] H. Fan and L. Li. (2007) Study on Metadata Applications for Proteomics Data Integration. In *Proc. ICBBE'07, IEEE*.

[8] K. Garwood et al. (2004) Pedro: A database for storing, searching and disseminating experimental proteomics data. *BMC Genomics*, 5.

[9] E. Jasper, A. (2003) Poulovassilis, and L. Zamboulis. Processing IQL queries and migrating data in the AutoMed toolkit. Technical Report 20, Automed Project.

[10] P. McBrien and A. Poulovassilis. (2003) Data integration by bi-directional schema transformation rules. In *Proc. ICDE'03*, pages 227–238.

[11] T. McLaughlin, J. A. Siepen, J. Selley, J. A. Lynch, K. W. Lau, H. Yin, S. J. Gaskell, and S. J. Hubbard. (2006) Pepseeker: a database of proteome peptide identifications for investigating fragmentation patterns. *Nucleic Acids Research*, 34.

[12] D.N. Perkins, D.J. Pappin, D.M. Creasy, and J.S. Cottrell. (1999) Probabilitybased protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18).

[13] L. Zamboulis, H. Fan *et al*, (2006) Data Access and Integration in the ISPIDER Proteomics Grid. In *proc. DILS*, pages 3–18.