A Uniform Approach to Workflow and Data Integration

Lucas Zamboulis[1, 2], Nigel Martin[1], Alexandra Poulovassilis[1]

[1]School of Computer Science and Information Systems, Birkbeck, Univ. of London
[2]Department of Biochemistry and Molecular Biology, University College London

Abstract

Data integration in the life sciences requires resolution of 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. This paper presents an approach that addresses these problems in a uniform way. We present results from the application of our approach for the integration of data resources and for the reconciliation of services within the ISPIDER and the BioMap bioinformatics projects. The ISPIDER integration setting demonstrates an architecture in which the AutoMed heterogeneous data integration system interoperates with grid access and query processing tools for the virtual integration of a number of proteomics resources, while the BioMap integration setting demonstrates the materialised integration of structured and semi-structured functional genomics resources using XML as a unifying data model. The service reconciliation scenario discusses the interoperation of the AutoMed system with a scientific workflow tool. The work presented here is part of the ISPIDER project, which aims to develop a platform using grid and e-science technologies for supporting in silico analyses of proteomics data.

1. Introduction

An objective of the ISPIDER project (see http://www.ispider.manchester.ac.uk) is to develop middleware to enable distributed querying, workflows and other integrated data analysis tasks across a range of novel and existing proteome data resources. Integrating separate but closely related resources in this way will provide a number of benefits, such as more reliable analyses by virtue of access to more data and reducing the number of false negatives. Moreover, the integration of resources, as opposed to merely providing a common access point to them, relieves the biologist from having to have knowledge of each resource and reconcile their semantics and their technologies.

This objective requires solutions to the problems of heterogeneous data integration and reconciliation of services performing analyses over that data. Data services are created independently by many parties worldwide using different technologies, data types and representation formats; as a result, semantically compatible services often cannot directly interoperate within a workflow [18].

To address these problems, the ISPIDER project makes use of a number of different software tools: OGSA-DAI and OGSA-DQP (see http://www.ogsadai.org.uk) to provide common access and distributed query processing of data resources, AutoMed (see http://www.doc.ic.ac.uk/automed) to enable transformation and integration of heterogeneous data resources and Taverna (see http://taverna.sourceforge.net) to enable workflow creation supporting complex analyses over diverse resources.

This paper presents an approach that provides support for data integration and service reconciliation within workflows in a uniform way. We show the application of our approach for the integration of a number of data resources using AutoMed, OGSA-DAI and OGSA-DQP, and we also discuss how AutoMed can interoperate, either statically or dynamically, with a workflow tool such as Taverna for the reconciliation of data services. Although the application domain is bioinformatics, our approach is not limited to a specific domain. Section 2 provides an overview of the AutoMed system. Section 3 presents our approach and demonstrates its application to data integration and to service reconciliation; we also discuss aspects of our approach that make it appropriate for both data and workflow integration. Section 4 discusses related work. Section 5 gives our concluding remarks and plans for future work.

2. Overview of AutoMed

AutoMed is a heterogeneous data transformation/integration system that offers the capability to handle virtual, materialised and hybrid data integration across multiple data models. It supports a low-level **hypergraph-based data model (HDM)** and provides facilities for specifying higher-level modelling languages in terms of this HDM. An HDM schema consists of a set of nodes, edges and constraints, and each modelling construct of a higher-level modelling language is specified as some combination of HDM nodes, edges and constraints. For any modelling language M specified in this way, AutoMed 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 that have textual names, there is also a rename primitive transformation.

AutoMed schemas can be incrementally transformed by applying to them a sequence of primitive transformations, each adding, deleting or renaming just one schema construct. A sequence of primitive transformations from one schema S_1 to another 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 AutoMed's Schemas & Transformations Repository.

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 [11]. Also available are extend and contract primitive transformations which behave in the same way as add and delete except that they indicate that the extent of the new/removed construct cannot be precisely derived from the other constructs present in the schema. Each extend or 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/removed construct.

The queries supplied with the primitive transformations can be used to translate queries or data along a transformation pathway – see [15][16] for details. The queries supplied with primitive transformations also provide the necessary information for these transformations to be automatically *reversible*, in that each add/extend transformation is reversed by a delete/contract transformation with the same arguments, while each rename is reversed by a rename with the two arguments swapped.

As discussed in [15], this means that AutoMed is a *both-as-view (BAV)* data integration system: the add and extend steps in a transformation pathway correspond to Global-As-View (GAV) rules as they incrementally define target schema constructs in terms of source schema constructs; while the delete and contract steps correspond to Local-As-View (LAV) rules since they define source schema constructs in terms of target schema constructs. An in-depth comparison of BAV with other data integration approaches can be found in [15]. [16][17] discuss the use of BAV in a peer-to-peer data integration setting. [12] discusses how Global-Local-As-View (GLAV) rules [8][13] can also be derived from BAV pathways. We note that AutoMed and BAV transform both schema and data together, and thus do not suffer from any data/schema divide.

We now briefly discuss the XMLDSS schema type for XML data sources used in AutoMed. The standard schema definition languages for XML are DTD and XML Schema. These provide grammars to which conforming documents adhere, and they do not explicitly summarise the tree structure of the data sources. In our schema transformation setting, schemas that do so are preferable as they facilitate schema traversal, structural comparison between a source and a target schema, and restructuring of the source schema. Moreover, such a schema type means that the queries supplied with AutoMed primitive transformations are essentially path queries, which are easily generated.

The AutoMed toolkit therefore supports a modelling language called *XML DataSource Schema (XMLDSS)*, which summarises the tree structure of XML documents, much like DataGuides [9]. XMLDSS schemas consist of four kinds of constructs: Element, Attribute, Text and NestList (see [20] for details of their specification in terms of the HDM). The last of these defines parent-child relationships either between two elements e_p and e_c or between an element e_p and the Text node. These are respectively identified by schemes of the form «i,e_p,e_c» and «i,e_p,Text», where i is the position of e_c or Text within the list of children of e_p in the XMLDSS schema.

In an XMLDSS schema there may be elements with the same name occurring at different positions in the tree. To avoid ambiguity, the identifier elementName$count is used for each element, where count is incremented every time the same elementName is encountered in a depth-first traversal of the schema.

3. A Uniform Approach to Workflow and Data Integration

Bioinformatics researchers frequently perform experiments and analyses over a multitude of local and remote datasets. These may be accessible directly, e.g. local relational

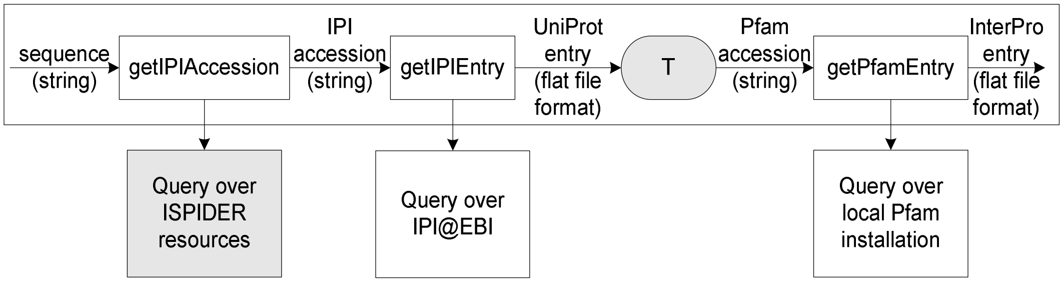

Figure 1: Sample Bioinformatics Workflow

databases, or indirectly, e.g. services producing flat-files or XML files. Scientific workflow tools such as Taverna provide facilities for creating highly complex workflows over such diverse resources.

Fig. 1 illustrates a simple bioinformatics workflow comprising three services: *getIPIAccession*, *getIPIEntry* and *getPfamEntry*. Each service obtains its data from a local or remote resource: *getIPIAccession* by executing a query over the virtual global schema of the ISPIDER integrated resource (see Section 3.1), *getIPIEntry* by executing a query at the EBI's IPI database (www.ebi.ac.uk/IPI) and *getPfamEntry* by executing a query over a local Pfam (www.sanger.ac.uk/Software/Pfam) installation.

This workflow presents two problems that commonly arise when creating bioinformatics workflows: (a) service *getIPIAccession* needs to access multiple known resources but the researcher is unfamiliar with the semantics of all of them, and (b) the output of *getIPIEntry* and the input of *getPfamEntry*, even though semantically compatible, cannot form a pipeline, as they have representation format, service technology and/or data type incompatibilities.

To address these problems, in the ISPIDER project we have used AutoMed to resolve the heterogeneity issues that arise, applying the BAV schema and data transformation and integration approach to address the following issues that are common to the problems of data integration and service reconciliation:

a) **Data model heterogeneity:** different resources may use different data models. It may also be the case that one or more resources do not have accompanying schemas (e.g. XML documents).
b) **Semantic heterogeneity:** schema differences caused by the use of different terminology, or describing the same information at different levels of granularity.
c) **Schematic heterogeneity:** schema differences caused by modelling the same information in different ways.
d) **Primitive data type heterogeneity:** differences caused by the use of different primitive data types for the same concept.

The workflow of Fig. 1 illustrates the need to support both data source integration, to enable *getPIAccession* to access ISPIDER resources as one integrated resource, as well as service reconciliation, to enable the output of *getIPIEntry* to form the input to *getPfamEntry*. Our approach deals with both issues in a single framework. Note that we do not discuss here the problem of service technology reconciliation, as this can be resolved using the Freefluo component of the Taverna workflow tool

Section 3.1 discusses the application of our approach for the virtual and materialised integration of bioinformatics resources. We first describe the architecture developed in the ISPIDER project to enable the integration of a number of heterogeneous proteomics resources in a grid environment and to subsequently query them in a distributed fashion. We then describe the materialised integration of structured and semi-structured resources within the BioMap project (www.biochem.ucl.ac.uk/bsm/biomap); this is to become one more resource accessible via the ISPIDER platform after its grid enablement. Section 3.2 then discusses the approach developed in ISPIDER to address the problem of bioinformatics service reconciliation, and Section 3.3 discusses the benefits of our uniform approach for data integration and service reconciliation.

3.1. Data Integration

We have applied our approach to the integration of four proteomics resources in the ISPIDER project and we have developed an architecture to support distributed querying of this resource by integrating the AutoMed system with the OGSA-DQP distributed query processor [1]. A more detailed description of this work is given in [21]. Figure 2 illustrates the integration setting and the architecture developed for the interoperation of AutoMed with OGSA-DAI and OGSA-DQP. Four different resources, gpmDB (http://gpmdb.thegpm.org), Pedro (http://pedrodb.man.ac.uk::8080/pedrodb), PRIDE (http://www.ebi.ac.uk/pride) and PepSeeker (http://www.ispider.manchester.ac.uk/pepseeker), were integrated under a virtual global schema developed by ISPIDER domain experts. This schema was based on the Pedro schema, since that has the widest proteomics coverage; this was then extended and fine-tuned as necessary.

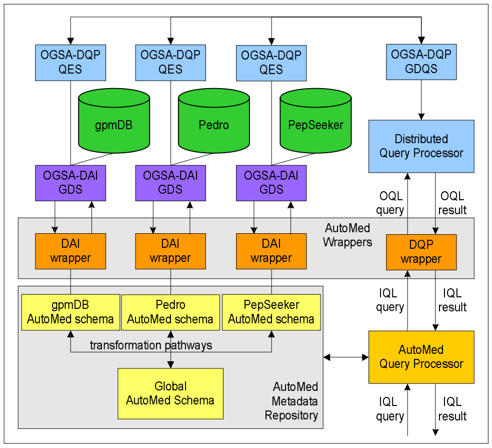

Figure 2: The AutoMed, OGSA-DAI and OGSA-DQP Architecture

In terms of the architecture, the data resources are exposed to the grid using OGSA-DAI grid data services. AutoMed-DAI wrappers interact with the OGSA-DAI services via XML documents to import the resources' schemas in the AutoMed Metadata Repository. AutoMed's schema and data transformation and integration functionality can then be used to create one or more virtual global schemas, together with the transformation pathways between the global schema(s) and the AutoMed representations of the resource schemas.

The integration semantics for this setting were derived by the ISPIDER domain experts and the data resource providers. We then produced the AutoMed transformation pathways, taking into consideration heterogeneity of types (2)-(4) discussed above (this setting does not present data model heterogeneity since all resources are relational).

Further virtual global schemas can be created as resources are developed holding information relevant to the initial virtual global schema. To enable querying across such schemas, a global 'super-schema' can then be created. The schema transformation pathways can also be used for incrementally materialising one or more of the integrated schemas [5] and for tracking data provenance [7]. These additional capabilities of the BAV approach and the AutoMed toolkit are also being pursued within the ISPIDER project.

Querying of a virtual global schema is supported by the interoperation of the AutoMed Query Processor (AQP – see Fig. 3) with OGSA-DQP. A user query, Q, expressed over the virtual global schema, is submitted to the AQP in the form of an IQL query. The *Query Reformulator* first uses the transformation pathways between the global schema and the data resource schemas to rewrite Q as a query Q_{ref} over the data resource schemas. Q_{ref} is then optimised using a number of different algebraic optimisations. The optimised query Q_{opt} is then processed by the *Query Annotator*, which detects and marks the largest subqueries q_i within Q_{opt} that can be translated to the subset of OQL that OGSA-DQP supports. These subqueries are then translated to OQL with the help of the AutoMed-DQP wrapper, and are sent for evaluation by the *Query Evaluator* to OGSA-DQP (see below). The *Query Evaluator* then processes the annotated query Q_{ann} using the results of the q_i and produces the final result.

OGSA-DQP uses a coordinating service (GDQS) to coordinate the query evaluation services (QES) that wrap the data resources (see Fig. 2). After evaluating a subquery q_i, OGSA-DQP returns its result to the AutoMed-DQP wrapper, for translation from OQL back to IQL.

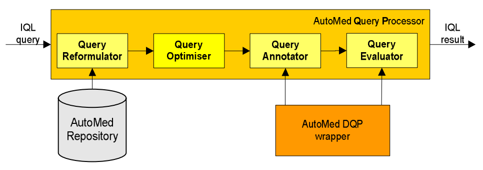

Figure 3: The AutoMed Query Processor

While the integration described above represents virtual data integration in a grid environment, grid resources may themselves be the result of materialized data integration, either for performance reasons or to enable value to be added to those resources through specialised local processing. The BioMap project represents an example of materialised integration to support fine-grained integration of specialised local resources with extensive distributed resources. To achieve this, a number of relational data resources were first integrated into a data warehouse using SQL queries. This integration process was labour intensive as all queries had to be manually designed. Reference [14] discusses how our schema transformation and integration approach was subsequently used within the BioMap integration. In contrast with the initial integration, [14] uses XMLDSS as the unifying data model in order to resolve data model heterogeneity. This decision was based on two factors: (a) to facilitate the future integration of XML and flat-file data resources into the BioMap warehouse and (b) to utilise our automatic XMLDSS schema and data restructuring algorithm (see below) in order to resolve schematic heterogeneity.

To address data model heterogeneity, we first developed an algorithm for transforming relational schemas and data into XMLDSS schemas and XML data. Semantic and data type heterogeneity were handled by manually specifying the necessary AutoMed transformations. However, note that this process was significantly less laborious than defining the data warehouse using SQL queries, as we only needed to supply transformations for those cases where there was a semantic or data type conflict or mismatch between the data resource schemas and the global schema. The rest of the schematic differences, i.e. schematic heterogeneity, were handled by our XMLDSS schema restructuring algorithm [23][20]. This algorithm is able to automatically transform an XMLDSS schema X_1 and its underlying data to the structure of another XMLDSS schema X_2, assuming X_1 and X_2 do not present any semantic or data type heterogeneity. The use of this algorithm is a significant advantage over manual solutions, such as the manual creation of XSLT scripts, especially in settings that are dynamic or where resources have large schemas.

The use of XMLDSS as the unifying data model greatly facilitates the integration of the BioMap data resources. However, it also introduces a level of complexity, as the relational source data have to be transformed to XML data and then back to relational form if we are to materialise the relational global schema. An area of future work is the generation of SQL scripts from the current integration setting for materialising the data of relational data resources under the relational global schema. Another area of future work is the use of incremental view maintenance techniques over BAV pathways [5] for the maintenance of the data warehouse.

3.2. Service Reconciliation

We have applied our approach for the reconciliation of bioinformatics services that are semantically compatible, but cannot interoperate due to incompatibilities in terms of data types and/or representation format. We refer the reader to [22] for a more detailed description of this work. To resolve the data model, semantic, schematic and primitive data type heterogeneity issues discussed earlier, we propose a four-step approach, illustrated in :

Step 1: Representation format translation to and from XML. Differences in the representation format are handled by using XML as the common representation format. If the output /input of a service is not in XML, then a format converter is needed to convert to/from XML.

Step 2: Automatically generate XMLDSS schemas. We use the XMLDSS schema type for the XML documents input to and output by services. An XMLDSS schema can be automatically extracted from an XML document or automatically derived from an accompanying DTD/XML Schema, if one is available.

Step 3: Define correspondences to typed ontologies. We use one or more ontologies as a “semantic bridge” between services. Providers or users of services semantically annotate the inputs and outputs of services by defining 1—1 or 1—n GLAV correspondences between an XMLDSS schema and an ontology. Ontologies in our approach are typed, i.e. each concept is associated with a data type, and so defining correspondences resolves the semantic and primitive data type heterogeneity issues discussed above.

Step 4: Schema and data transformation. We use the AutoMed toolkit to automatically transform the XMLDSS schema of the output of service S_1 to the XMLDSS schema of the input of service S_2. This is achieved using the two automatic algorithms presented in [24][22], which utilise the correspondences defined in Step 3. The first of these two algorithms, the schema conformance algorithm (SCA), uses the manually defined correspondences between an XMLDSS schema X_1 and an ontology O, to automatically transform X_1 into a new XMLDSS schema that uses the terminology of O. The second algorithm, the schema restructuring algorithm (SRA), is an extension of the schema restructuring algorithm defined in [23][20], as it is able to use subtyping information derived from the ontology.

Note that our approach does not require the full set of correspondences to be defined: we allow the definition of only those correspondences between the XMLDSS schema and the ontology that are relevant to the problem at hand.

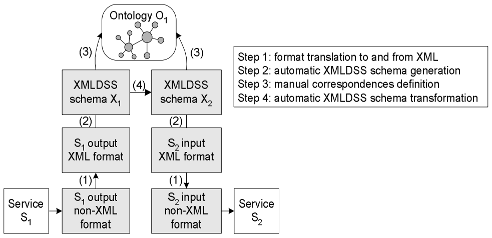
Ontology O_1
(3)
(3)
Step 1: format translation to and from XML
Step 2: automatic XMLDSS schema generation
Step 3: manual correspondences definition
Step 4: automatic XMLDSS schema transformation
XMLDSS schema X_1
(4)
XMLDSS schema X_2
(2)
(2)
S_1 output XML format
S_2 input XML format
(1)
(1)
Service S_1
S_1 output non-XML format
S_2 input non-XML format
Service S_2

Figure 4: Reconciliation of Services S_1 and S_2

Also note that we do not assume the existence of a single ontology. As discussed in [24][22], it is possible for XMLDSS schema X_1 to have a set of correspondences C_1 to an ontology O_1, and for XMLDSS schema X_2 to have a set of correspondences C_2 to another ontology O_2. Provided there is an AutoMed transformation pathway between O_1 and O_2, either directly or through one or more intermediate ontologies, we can use C_1 and the transformation pathway between O_1 and O_2 to automatically produce a new set of correspondences C_{1new} between X_1 and O_2. As a result, this setting is now identical to a setting with a single ontology. There is a proviso here that the new set of correspondences C_{1new} must conform syntactically to the correspondences accepted as input by the schema conformance process. Determining necessary conditions for this to hold is an area of future work.

Our architecture for service reconciliation supports two different modes of interoperation of AutoMed with workflow tools:

Mediation service. With this approach, the workflow tool invokes service S_1, receives its output, and submits this output and a handle on service S_2 to a service provided by AutoMed. This uses our approach to transform the output of S_1 to a suitable input for consumption by S_2.

Shim generation. With this approach, the AutoMed system is used to generate shims, i.e. tools or services for the reconciliation of services, by generating transformation scripts which are then incorporated within the workflow tool.

With the second approach, AutoMed is not part of the run-time architecture, and so it is necessary to export AutoMed's mediation functionality. Format converters and the XMLDSS generation algorithms can be either incorporated within the workflow tool, or their functionality can be imported using services. On the other hand, the two XMLDSS schema transformation algorithms described in [24][22] are currently tightly coupled with the AutoMed system, since the algorithms use the Both-As-View data integration approach, which is currently supported only by AutoMed. In order to use our approach without dynamically integrating AutoMed with a workflow tool, we need to export the functionality of our schema transformation algorithms. To this effect, we have designed an XQuery query generation algorithm that derives a single XQuery query Q, able to materialise an XMLDSS schema X_2 using data from the data source of an XMLDSS schema X_1 (see [22]). The use of a query language such as XQuery instead of XSLT was deemed more appropriate in our setting, since our approach uses a query language, IQL, for describing the extents of constructs within transformations. However, deriving an XSLT script from an AutoMed XMLDSS transformation pathway is an area of future work.

3.3. Discussion

We now discuss aspects of our approach that make it appropriate for both data and workflow integration.

First, we note the ability of our approach to handle scenarios of data integration and service reconciliation in a uniform way, i.e. we use the same methodology to analyse the problems that arise in these scenarios and we use the same approach to address them.

Our approach is based on BAV data integration, which uses schema transformation pathways rather than view definitions, which are hard to maintain. An advantage of BAV is its ability to readily support the evolution of both data resource and global schemas (or source and target schemas in the service reconciliation setting) [6].

The use of HDM as a common low-level data model allows us to address the possible data model heterogeneity of services and resources in our problem settings. The HDM is able to express not only data resource schemas, but also ontologies, allowing us to utilise ontology semantics for data resources within a single framework.

The BAV approach allows the creation of arbitrary networks of schemas. It also allows for the virtual, materialised and indeed hybrid integration of data resources. It is possible to partially materialise a schema, and thus exploit the advantage of materialised integration (query processing performance) for some schema constructs and the advantage of virtual integration (avoiding the problems of materialised integration, such as data staleness) for other schema constructs.

As discussed in [5][7], AutoMed is able to support data warehousing functionality such as data provenance and incremental view maintenance, and this is an area of future work for ISPIDER and BioMap.

Finally, we note that the AutoMed toolkit is implemented in a modular fashion, and so offers the ability to interoperate easily with other independently developed software tools, such as OGSA-DAI/DQP and (in the future) Taverna.

4. Related Work

Diverse approaches to the issues of data integration and service reconciliation have been pursued in related work.

Concerning data integration, Section 2 has given an overview of the BAV data integration approach and its implementation in the AutoMed system, while Section 3.3 has discussed the advantages of BAV over GAV, LAV and GLAV. More details are given in the papers referenced in these sections.

In the context of service composition, research has mainly focused on service technology reconciliation, service matchmaking and service routing, assuming that the outputs and inputs of services are a priori compatible. This assumption is restrictive, as it is often the case that two services are semantically compatible, but cannot interoperate due to incompatibilities in terms of data types and/or representation format.

In order to minimise this issue, ^{my}Grid (see http://www.mygrid.org.uk) has fostered the notion of *shims* [10], i.e. services acting as intermediaries between other services that need to interoperate. The problem with this manual approach is that it is not scalable due to the potentially large number of services available: ^{my}Grid currently gives access to more than 3,000 services.

[3] describes a scalable framework for reconciling services that produce and consume XML data. This framework makes use of one or more ontologies to address semantic heterogeneity and produces XQuery transformation programs to resolve schematic heterogeneity. In contrast, we also address the problems of data model and primitive data type heterogeneity. Moreover, we specify a methodology for reconciling services that correspond to more than one ontology, and we also allow the definition of more expressive correspondences than [3].

[19] also uses a mediator system, but for service composition. The focus there is either to provide a service over the global schema of the mediator whose data sources are services, or to generate a new service that serves as an interface over other services. In contrast, we use the AutoMed toolkit to reconcile a sequence of semantically compatible services that need to form a pipeline: there is no need for a single "global schema" or a single new service to be created.

Concerning the use of ontologies for data integration, a number of approaches have been proposed. For example, [2] uses an ontology as a virtual global schema for heterogeneous XML data sources using LAV mapping rules, while [4] undertakes data integration using mappings between XML data sources and ontologies, transforming the source data into a common RDF format. In contrast, we use XML as the common representation format and focus on the restructuring of source data into a target XML format, rather than on integration.

5. Conclusions and Future Work

This paper has demonstrated the application of a uniform approach to addressing the problems of heterogeneous data integration and service reconciliation. Using our approach, we are able to define workflows that utilise integrated resources and we are also able to semi-automatically reconcile workflow services.

Concerning data integration in a grid environment, we first described an architecture that combines the data integration capabilities of AutoMed and the grid enabling and grid distributed query processing capabilities of OGSA-DAI and OGSA-DQP for the virtual integration of proteomics resources. We then demonstrated the application of our approach for the materialised integration of structured and semi-structured data resources using XML as a unifying data model.

We are currently working on the integration of our service reconciliation approach using AutoMed with the Taverna workflow tool. We also plan to grid-enable the BioMap warehouse and make it available as an independent resource via the ISPIDER proteomics grid. Another strand of work within the ISPIDER project is the evaluation of the benefits of the BAV approach towards schema evolution, both in data integration settings and in bioinformatics service reconciliation. In the latter setting, we plan to investigate the evolution of service input/output schemas as well as the evolution of ontologies.

Finally, members of the AutoMed project have been working on P2P query processing in AutoMed [17], and we plan to investigate parallel and distributed query processing in grid and P2P settings using the AutoMed system.

Acknowledgements. We acknowledge the support of the BBSRC for the ISPIDER project and of the Wellcome Trust for the BioMap project.

References

[1] M. N. Alpdemir et al., *Service-based distributed querying on the Grid*, in Proc. Int. Conf. on Service Oriented Computing, pp 467-482, 2003.

[2] B. Amann, C. Beeri, I. Fundulaki, and M. Scholl, *Ontology-based integration of XML web resources*, in Proc. of Int. Semantic Web Conference, pages 117-131, 2002.

[3] S. Bowers, B. Ludäscher, *An Ontology-Driven Framework for Data Transformation in Scientific Workflows*, in Proc. of Data Integration in the Life Sciences (DILS'04), pp 1-16, 2004.

[4] I. F. Cruz, H. Xiao, and F. Hsu, *An ontology-based framework for XML semantic integration*, in Proc. of IDEAS'04, pages 217-226, 2004.

[5] H. Fan, *Using Schema Transformation Pathways for Incremental View Maintenance*, in Proc. of DaWaK'05, pp 126-135, 2005.

[6] H. Fan, A. Poulovassilis, *Schema Evolution in Data Warehousing Environments - a schema transformation based approach*, in Proceedings of ER'04, pp 639-653, 2004.

[7] H. Fan, A. Poulovassilis, *Using Schema Transformation Pathways for Data Lineage Tracing*, in Proc. of BNCOD'05, LNCS 3567, pp. 133-144, 2005.

[8] M. Friedman, A. Levy, and T. Millstein, *Navigational plans for data integration*, in National Conference on Artificial Intelligence, pp 67-73, 1999.

[9] R. Goldman and J. Widom, *DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases*, in Proc. VLDB'97, pp 436-445, 1997.

[10] D. Hull et al., *Treating shimantic web syndrome with ontologies*, in Proc. of Advanced Knowledge Technologies workshop on Semantic Web Services, 2004.

[11] E. Jasper, A. Poulovassilis, L. Zamboulis, *Processing IQL queries and migrating data in the AutoMed toolkit*, AutoMed Technical Report 20, July 2003.

[12] E. Jasper, N. Tong, P.J. McBrien, A. Poulovassilis, *View generation and optimisation in the AutoMed data integration framework*, in Proc. of 6th Baltic Conference on Databases and Information Systems, 2004.

[13] J. Madhavan and A.Y. Halevy, *Composing mappings among data sources*, in Proc. of VLDB'03, pp 572--583, 2003.

[14] M. Maibaum, L. Zamboulis, G. Rimon, N. Martin, A. Poulovassilis, *Cluster based integration of heterogeneous biological databases using the AutoMed toolkit*, in Proc. Data Integration in the Life Sciences, pp 191-207, July 2005.

[15] P.J. McBrien, A. Poulovassilis, *Data integration by bi-directional schema transformation rules*, in Proc. ICDE'03, pp 227-238, 2003.

[16] P.J. McBrien, A. Poulovassilis, *Defining peer-to-peer data integration using Both As View rules*, in Proc. of DBISP2P Workshop (at VLDB'03), 2003.

[17] P.J. McBrien, A. Poulovassilis, *P2P query reformulation over Both-as-View data transformation rules*, in Proc. of DBISP2P Workshop (at VLDB'06), pp TBC, 2006.

[18] L. Stein, *Creating a bioinformatics nation*, Nature, 417:119-120, May 2002.

[19] S. Thakkar, J.L. Ambite, and C. A. Knoblock, *Composing, optimizing, and executing plans for bioinformatics web services*, VLDB Journal, 14(3):330--353, 2005.

[20] L. Zamboulis, *XML data integration by graph restructuring*, in Proc. British National Conference on Databases (BNCOD'04), pp 57—71, 2004.

[21] L. Zamboulis, H. Fan, K. Belhajjame, J. Siepen, A. Jones, N. Martin, A. Poulovassilis, S. Hubbard, S.M. Embury, N.W. Paton, *Data access and integration in the ISPIDER proteomics Grid*, in Proc. Data Integration in the Life Sciences, pp 3-18, July 2006.

[22] L. Zamboulis, N. Martin, A. Poulovassilis *Bioinformatics service reconciliation by heterogeneous schema transformation*, to appear in Proc. Data Integration in the Life Sciences, pp 89-104, 2007.

[23] L. Zamboulis, A. Poulovassilis, *Using AutoMed for XML Data Transformation and Integration*, in Proc. DIWeb Workshop (at CAiSE'04), pp 58-69, June, 2004.

[24] L. Zamboulis, A. Poulovassilis, *Information sharing for the Semantic Web - a schema transformation approach*, in Proc. of DISWeb Workshop (at CaiSE'06), pp 275-289, June 2006.