A Semantic Grid-based Data Access and Integration Service for Bioinformatics

Giovanni Aloisio, Massimo Cafaro, Italo Epicoco, Sandro Fiore, Maria Mirto
ISUFI/CACT, University of Lecce and NNL/INFM&CNR, Italy
{giovanni.aloisio, massimo.cafaro, italo.epicoco, sandro.fiore, maria.mirto}@unile.it

Abstract

Given the heterogeneous nature of biological data and their intensive use in many tools, in this paper we propose a semantic data access and integration (DAI) service, based on the Grid paradigm, for the bioinformatics domain. This service uses ontologies for correlating different data sets. The DAI proposed in this work is a fundamental component of the ProGenGrid system, a grid-enabled platform, which aims at the design and implementation of a virtual laboratory where e-scientists could simulate complex "in silico" experiments, composing some popular analysis and visualization tools (e.g. Blast and Rasmol) available as Web Services, into a workflow. The main goal of the DAI is to provide bioinformatics tools with advanced functionalities and data integration services for heterogeneous biological data banks, such as PDB and Swiss-Prot. A case study of our specialized data access service for locating similar protein sequences is presented.

Keywords: Bioinformatics, DAI, Ontologies, Web Services, Computational Grid, Grid Portal, Globus Toolkit.

1. Introduction

Complete genome sequences and protein-coding gene sets are becoming available for a growing number of organisms. While these are proving highly informative and invaluable for studying those and related organisms, at the same time they make it clear how far we still have to go before reaching an in-depth understanding of how a genome determines the lifestyle of an organism.

The increasing amount and complexity of biological data makes it increasingly difficult to access and analyse the data. These data, stored in different geographically spread repositories, are heterogeneous when we consider genomic, cellular, structure, phenotype and other types of biologically relevant information [1], and often describe the same objects utilizing different representations such as Swiss-Prot [2], where the protein is mapped just as amino acid sequence or Protein Data Bank (PDB) [3] that contains 3D structure.

The semantic relation among these data repositories is a key factor for integration in bioinformatics since it could allow a unique front end for accessing them, as required by many biological applications. Ontology could help here to localise the right type of concept to be searched for as opposed to identification of a mere label naming a search table. It includes definitions of basic concepts in the domain and relations among them, which should be interpretable both by machines and humans.

Moreover, biological repositories are often quite large and need to be updated for annotations or when we add new entries. To date, many tools exist for simulating complex "in silico" experiments, that is simulations carried out using biological data, as opposed to "in vitro" or "in vivo" ones that are conducted respectively outside or inside a living organism or cell. These tools need to access heterogeneous data banks, distributed on a wide area, and in particular need a supporting infrastructure for obtaining successfully a result [4]. Many of these tools are freely available on the Internet, and there is plenty of software such as EMBOSS [5] and SRS [6] for accessing different data banks.

SRS is the most widely used data integration system for biological, biochemical and biomedical databases. It enables users of all backgrounds to intuitively access data and permits internal data to be merged with data from the public domain. The most prominent public server at EBI (http://srs.ebi.ac.uk) currently holds more than 130 biological databases. A key problem with the current structure of SRS is that it is designed only for accessing local databases. This requires the SRS administrators to

- provide local copies of all the databases and
- keep these local copies continuously up to date.

This approach uses interconnected heterogeneous databases via web hypertext links at the level of individual data items. Data retrieval in such system takes place by using the results of one query to link and jump to a particular entry in the same or another data source. However most of the potential links among data in digital form are not readily available because the relevant data, when they exist, are in different databases. In addition, each database is typically based on different and incompatible database technologies and uses different languages and vocabularies to access data. These incompatibilities are especially significant when non-

0-7803-9074-1/05/$20.00 ©2005 IEEE

textual data, such as 3D images of protein structures, accessed by author-specified keywords, need to be linked with nucleotide sequences in other databases. Because each database is typically created as a standalone application to support one functionality, linking among databases is most often an afterthought. It is possible (using an integrated approach which considers the semantic meaning of data) to dynamically create links such as a search engine.

To date, a (de facto) specialized data access service for bioinformatics, able to provide access to data and distributed tools, does not exist (yet).

A data access service is involved in many biological experiments where Workflow techniques are needed to assist the scientists in the design, execution and monitoring of them. Workflow Management Systems (WFMSs) support the enactment of processes by coordinating the temporal and logical order of the elementary process activities and supplying the data, resources and application systems necessary for the execution [7].

The Grid [8] framework is an optimal candidate for executing bioinformatics workflows because it offers the computational power for high throughput applications and basic services such as efficient mechanisms for transferring huge amounts of data and exchanging them on secure channel.

So, bioinformatics platforms need to offer powerful and high level modelling techniques to ease the work of e-scientists, as for instance exploiting Computational Grids transparently and efficiently.

ProGenGrid (Proteomics and Genomics Grid) [9] is a software platform which integrates biological databases, analysis and visualization tools, available as Web Services, for supporting complex "in silico" experiments. The choice to couple Web Services [10] and Grid technologies produces components independent of programming language and platforms that exploit a grid infrastructure. ProGenGrid is based on the following key approaches: web/grid services, workflow, ontologies and data integration through the Grid.

In this paper we focus on the functions and architecture of a Data Access and Integration (DAI) service and its use inside the ProGenGrid platform. The use of the proposed DAI service in an experiment of searching similarity matching among proteins is presented. The outline of this paper is as follows: in Section 2, we describe the features of a bioinformatics DAI. In Section 3 we describe our DAI solution whilst in Section 4 we show the role of the DAI in the ProGenGrid system. We conclude the paper in Section 5.

2. Why Bioinformatics Grids and Web Services?

2.1. Bioinformatics Grids

The interconnection of computers using Grid middleware enables the user to utilize computing power and retrieve information from heterogeneous and distributed sources transparently and efficiently. A Computational Grid could be a solution to many bioinformatics issues because it allows the deployment, distribution and management of needed biological software components, the harmonized standard integration of various software layers and services, a powerful, flexible policy definition, and control and negotiation mechanisms for a collaborative grid environment. This could reveal useful information for understanding the complex interrelation between genetic information and hereditary diseases and hence can lead to important discoveries in life science.

Bioinformatics Grids are environments built for the specific domain of biology including hardware and software resources needed for solving issues related to biological experiments and simulations. Some examples of Bioinformatics Grids are Asia Pacific BioGRID [11] and myGrid [12]; the former integrates selected biomolecular applications with the Unicore infrastructure, the latter provides high-level grid services for bioinformatics applications for data and application integration. These projects are very useful for the scientific community because new techniques for solving various bioinformatics issues are designed and experimented.

2.2. Web Services

Web services describe an emerging XML-based distributed computing paradigm that differs from other approaches such as CORBA and Java RMI. The basic idea is to build a system out of existing Internet-based standards. Web services define the description of how to invoke service components, a protocol for conveying remote procedure calls (RPC, but also Document style Web services can be used), and the discovery mechanism for locating the service definition of relevant service providers. Web Services technology allows independence from platforms/programming languages and reusability of the code.

2.3. Integrating Grid and Web Services technologies to enable DAI service

Data access and integration service include key steps in the data life cycle process, such as data creation and

acquisition, use, modification, archiving and disposal. This process involves many data banks (data providers) and users/applications, which use the data. Coupling the Grid framework and Web Services makes it possible to build a bioinformatics DAI service satisfying the following features:

Accessibility: ease of use, support for multiple data models and database abstractions; using a Grid framework it is possible to access a large set of resources and data efficiently. Through easy to use user interfaces that hide the complexity of accessing the Grid (the so called Grid Portals), the user can access a variety of grid services.

Capacity and archiving support: local and remote data storage capacity, for the archival process, including space for expansion and annotation of the database; a Grid offers huge amount of data storage capacity and efficient mechanisms to move the data between grid nodes.

Intellectual property, privacy and security: the first regards ownership of sequence data, images, and other data stored in and communicated through the database, the second is the provision for preserving confidentiality of data and the last is the limit on user access. Each user is recognized in a grid infrastructure through proper credentials to access her own data or run applications on the grid. Through a single sign-on the user at first authenticates herself and then uses the resources for which she has permission rights (authorization process).

Interfaces: connectivity with other databases and applications; these represent the Web service interface to databases and application tools and are used either by the user or another service to send a query, to insert the parameters needed for the execution of a specific application and to obtain the results.

Portability on multiple platforms: using Web services technology it is possible to build platform independent components;

Performance: access time and data throughput; in particular using the GridFTP [13] protocol it is possible to transfer (through parallel streams) efficiently huge amounts of data;

However, there are other important issues of bioinformatics DAI that Grid and Web Service do not support such as:

Metadata Management: it includes the design, implementation, and maintenance of the metadata associated to different data sets whose semantic meaning is described through a data dictionary or ontology;

Multiple data formats: support for various data formats such as flat file, FastA and XML;

Data input support: hardware, software, and processes involved in feeding data into the database, from keyboard and voice recognition to direct instrument feed and the Internet;

Export/Import capabilities: provisions for importing and exporting data to and from different file formats;

Indexing: indexing methodology, including selection and use of the most appropriate controlled vocabulary;

Query Language: proprietary or standard query language for supporting complex query.

In the next Section, we will discuss our solution for an efficient DAI.

3. The ProGenGrid Data Access and Integration (DAI) Service

Our DAI has been studied for supporting integration of biological data sources and high throughput applications such as Blast or Drug design applications. It is also responsible for mapping high level requests (user requests) to low level queries, specific for each data source. These ones are in general not structured. In the following part we describe in detail this service.

3.1. Data Integration

The main goal of data integration is to develop the technology to grant a user access to multiple information systems, to retrieve information and to perform computations transparently as if they were a single source. The first complexity in achieving this goal is that the information sources are often independent and autonomous, they have completely different scheme structures and use different data formats. To provide uniform access, an integration system must therefore face the problem of data heterogeneity at the system, syntax and structural level. Moreover there is a significant degree of semantic heterogeneity among different information sources. Unfortunately, the semantics of different data sources is hidden or unclear. The integration system [14] must provide a mechanism to bridge across this semantic difference. Current solutions involve a link-integrated database system and hence provide only partial, high-level integration with the growing number of rapidly expanding molecular biology databases. In Figure 1, we show an example of how Swiss-Prot and PDB are cross-referenced: Swiss-Prot identifies a protein with a proprietary identifier (P12544), but contains also the identifier used by PDB to identify the same protein (1HF1).

Another approach involves a data warehouse which combines data from a variety of databases in one physical location. It is very powerful for running queries against high volumes of data but it requires complex procedures for designing a global scheme and updating data.

The model that we propose is an extension of the middleware mediator approach [15], based on two-part

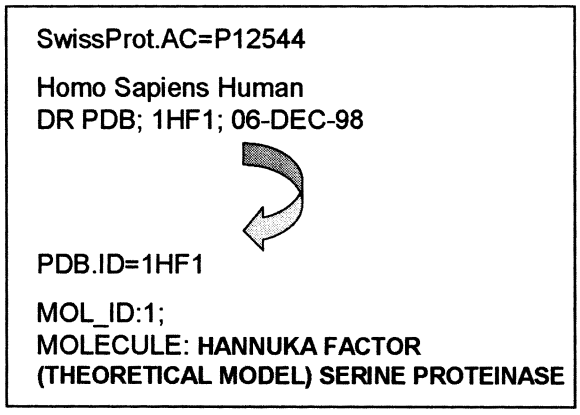

Fig. 1. Cross-referenced link between Swiss-Prot and PDB.

middleware and on clients which formulate queries. The first part (called wrapper) sits on top of each data source and often performs two different functions: i) it translates the data into a common data model and ii) it takes a query-fragment from the mediator and transforms it into an equivalent query in the query language of the sources. The second part (called mediator engine), built on top of all of the wrappers, first decomposes a query in a set of sub-queries for each wrapper, then takes the partial results from the wrappers and constructs the final result.

There are mediator systems that provide a semantic bridge across information sources in complex application domain such as biology such as TAMBIS [16] or BioDataServer [17], but these do not consider the integration of distributed data sources in a grid environment.

In this paper, we present an information integration system that follows the mediator architecture but extends it by incorporating domain specific bioinformatics knowledge in a grid environment.

As can be seen in Figure 2, our system is made of:

- Semantic Wrapper (SW), built on top of a data source, it includes
  i. Scheme, i.e. the (ER – Entity/Relation - or UML) data model of a source;
  ii. Ontology, that describes a specific data source;
  iii. Relations/associations, between the local ontology and the scheme;
  iv. APIs, for retrieving a specific attribute or field.
- Mapper, a catalog that gathers the schemes and their description coming from each SW; it is used to identify the data source of a query and to select the appropriate wrapper;
- Data Source Ontology (DSO): it virtualises data sources and maps the semantic links between them;
- Mediator which i) given a user query, searches semantic relations in the DSO and ii) consults the Mapper, reformulating the query, and splitting it into sub-queries, each one specific to a data source.

Regarding the Scheme (point i.), we have analysed the Swiss-Prot database (Figure 3 shows an entry) and we have built its E/R model. In particular some entities (Figure 4) involved in the scheme are:

- Entry: composed of *ID* (corresponds to ID – IDentification - tag of Swiss-Prot), *length* (sequence length which is the last field of ID tag, 262 in the example of Fig. 4), *seq* (SQ involves the sequence i.e. TTCCP ...), *Descr* (DE tag - description), *AC* (AC tag - accession number), *CodGen* (GN tag – codifying gene), *Keyw* (KW tag – keywords) fields;
- Taxonomy: involves *ID*, *Name* (OC tag - organism taxonomy), *Synonymous* (OX tag - taxonomy through cross reference) fields;
- Reference: comprises *ID*, *Title*, *Year*, *Volume* and *Journal* (RN,RP,RC,RX,RA,RT,RL tags contain the bibliographic reference) fields.

With regard to the ontology related to each data source (point ii.), it contains semantic relations between concepts described in the data source. In particular Figure 5 shows a fragment of the ontology for Swiss-Prot, where some features for each protein (e.g. taxonomy, function etc.) are mapped. It is worth noting here that in this database some information are correlated, so using E/R scheme and the ontology it is possible to try all of the relations among data.

A possible relation among data obtained by scheme and ontology ties together entry and taxonomy with associated IDentry and IDTaxonomy (point iii.). So IDTaxonomy corresponds to the organism terms in the ontology.

We would like to integrate the following databases:

- Structure: PDB and CATH [18];
- Sequence: Swiss-Prot;
- Function: ENZYME databases [19].

To build the SW component, we need to model each data source using a ER model and an ontology. In particular, we plan to use Gene Ontology [20] for collecting the needed ontologies for modelling the data of interest. The APIs indicated in point iv. (see Semantic Wrapper description) are simple functions that allow binding and unbinding to/from the physical database, to search a given attribute or move between entries of the database. Moreover these are needed for populating the relational scheme automatically.

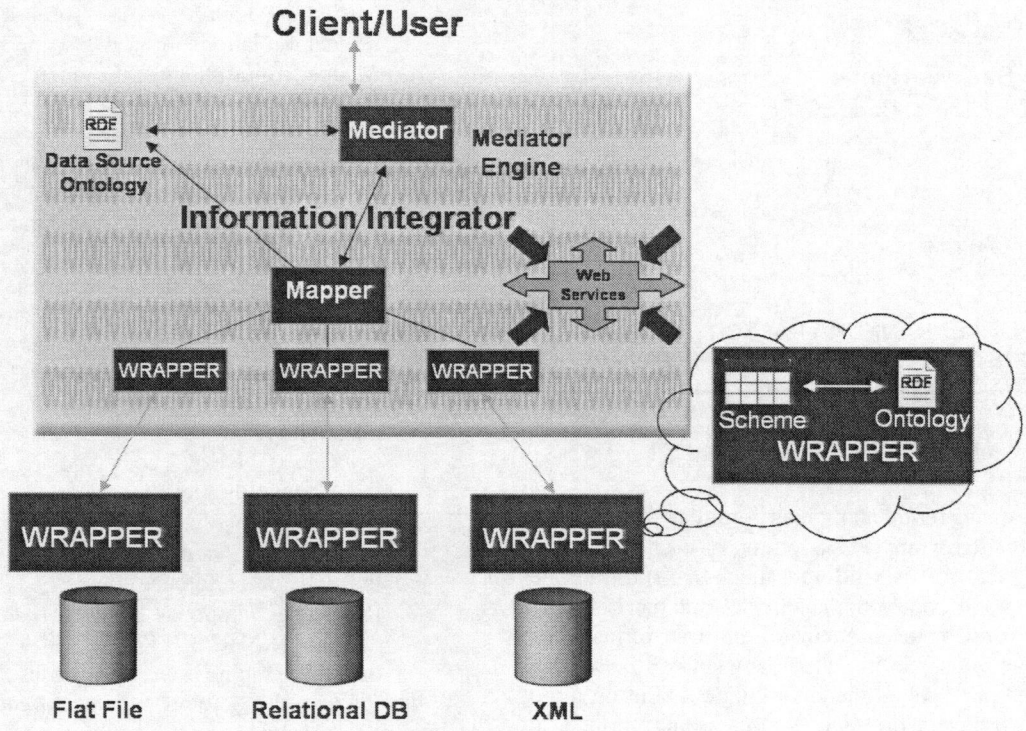

Fig. 2. ProGenGrid DAI Architecture.

Indeed, for each analysed wrapper we have implemented in C language some functions that translate the data source into an XML scheme and carry out the ingestion of the entire database in our relational data model. These features have been provided jointly with the GRelC library [21].

ID GRAA_HUMAN STANDARD; PRT; 262 AA.
AC P12544;
DT 01-OCT-1989 (Rel. 12, Created)
DT 01-OCT-1989 (Rel. 12, Last sequence update)
DT 01-OCT-2004 (Rel. 45, Last annotation update)
DE Granzyme A precursor (EC 3.4.21.78)
GN Name=GZMA; Synonyms=CTLA3, HFSP;
OS Homo sapiens (Human).
OC Eukaryota; Metazoa; Chordata; Craniata; Vertebrata.
OX NCBI_TaxID=9606;
RN [1]
RP SEQUENCE FROM N.A.RC TISSUE=T-cell;
RX MEDLINE=88125000; PubMed=3257574;
RA Gershenfeld H.K., Hershberger R.J., Shows T.B., Weissman I.L.;
RT "Cloning and chromosomal assignment of a human cDNA"
RL Proc. Natl. Acad. Sci. U.S.A. 85:1184-1188(1988).
RL Proteins 4:190-204(1988).
CC -!- FUNCTION: This enzyme is necessary for target cell
CC lysis in cell- mediated immune responses. It cleaves after

Fig. 3. An entry of Swiss-Prot database.

The Mapper contains a catalogue of data source schemes and a brief description. It is worth noting here that it contains the logical file name of the scheme associated with one or more physical file names (for instance EMBL databank has a relational, flat file and XML version corresponding each to a Mapper entry).

Data Source Ontology (DSO) classifies the data sources w.r.t. some features providing a unified conceptual level representation of its registered component resources.

In the following text we show how concepts in different ontologies are linked. As an example, the relation "polypeptide_chain(**is_composed**, SwissProt.sequence, PDB.sequence)" expresses the fact that polypeptide_chain is both a sequence in Swiss-Prot or in PDB. For the databases cited above we could consider the classification for protein as follows, where the first field is the relation and the other ones are related attributes:

protein (**has**, name, polypeptide_chain, function)
polypeptide_chain(**is_composed**, SwissProt.sequence, PDB.sequence);
PDB.sequence(**has**, PDB.3Dstructure);
Cath.code(**has**, Cath.domain_def);
PDB.3Dstructure(**is_composed,** Cath.domain_def)
SwissProt.sequence(**has**,SwissProt.description, SwissProt.keywords);
protein.function (**is_composed**, SwissProt.keywords);

protein.function (**is_composed**, Enzyme.class).
Enzyme.ECnumber(**has**,Enzyme.catal.,Enzyme.class);

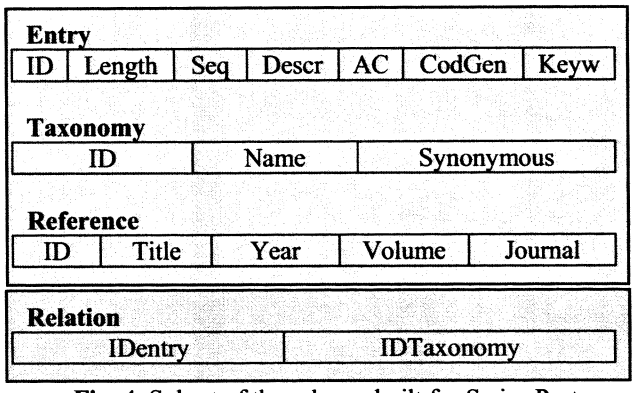

Fig. 4. Subset of the scheme built for Swiss-Prot.

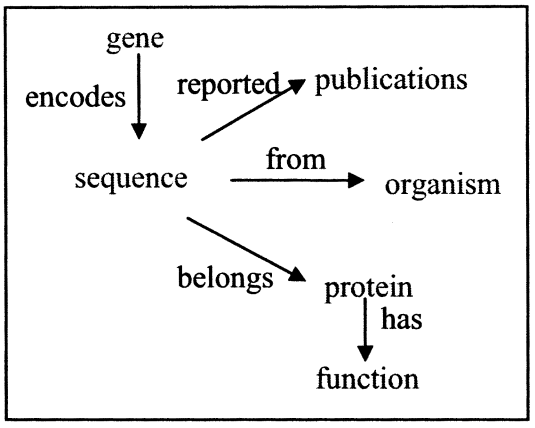

Fig. 5. Ontology for Swiss-Prot.

The Mediator accepts requests from the user and retrieves the information if the searched data (exploring the DSO) are semantically correlated. It is worth noting here that the Mediator should implement a logic having a definition of query with different abstraction levels (initially, we planned to use the SQL standard language but now we are considering other hypotheses, providing a request virtualisation layer). The Mediator engine coordinates the temporal activities of all of the components selecting those available on some nodes of a Computational Grid.

3.2. Implementation

The Mediator component provides some methods, through a Web Services interface. The Web service server has been implemented in C, exploiting the gSOAP Toolkit [22], because it is well suited for the conversion of legacy application using SOAP and its main feature is a transparent SOAP API. To guarantee a secure channel to move biological data, we also used the Globus Security Infrastructure (GSI) support, available through our gSOAP plug-in [23]. So, the Mediator Web Service (server) and clients can establish a SOAP connection over a secure GSI channel exchanging X.509v3 certificates for mutual authentication/authorization and delegation. The Workflow editor has been implemented in Java so in this system the client to the Web Service has been realized using Apache Axis and GSS API.

Moreover, we are finishing the Wrapper APIs for the data banks cited above, to provide a set of primitives to get access to and interact transparently with different data sources. Finally, for high throughput applications we are investigating an approach based on our mechanism called SplitQuery which provides an efficient fragmentation of the biological data set and a protocol for retrieving the fragment, as described in [24].

Currently, we are exploiting the Globus Toolkit 3.2 pre-OGSI [25] as Grid middleware in our project.

4. Case study: using DAI in a Workflow for searching sequence similarity

Recently, many workflow languages have been defined such as Web Services Flow Language (WSFL) [26], Business Process Execution Language (BPEL) [27], and UML extensions. We use UML (Unified Modeling Language, [28]) activity diagrams as a workflow language specification. UML, as well as all of its extensions, is the most widely accepted notation for designing and understanding complex systems; it has an intuitive graphical notation, and UML activity diagrams support [29] most of the control flow constructs and are suitable to model workflow execution.

As an application of ProGenGrid, we present a workflow modelling the process of searching similarity matching among proteins. Figure 6 shows an activity diagram specification of the similarity search process. This process starts by supplying a target protein < IDProtein > or its FASTA format (in this example, the protein target is 1LYN), the search procedure accesses the database and all of the information about target protein is recovered from the Swiss-Prot database.

To date, we are using the SQL language like that for our experiment. In particular, given the input protein X (1LYN), and indicating with Y_i , i $\in$ (1, ... 200000) a set of sequences extracted from Swiss-Prot, the following query first selects all sequences from Swiss-Prot whose alignment score is greater than a threshold value *score*, and then, using the sequence Accession Number, it selects from PDB the structural information related to such sequences:

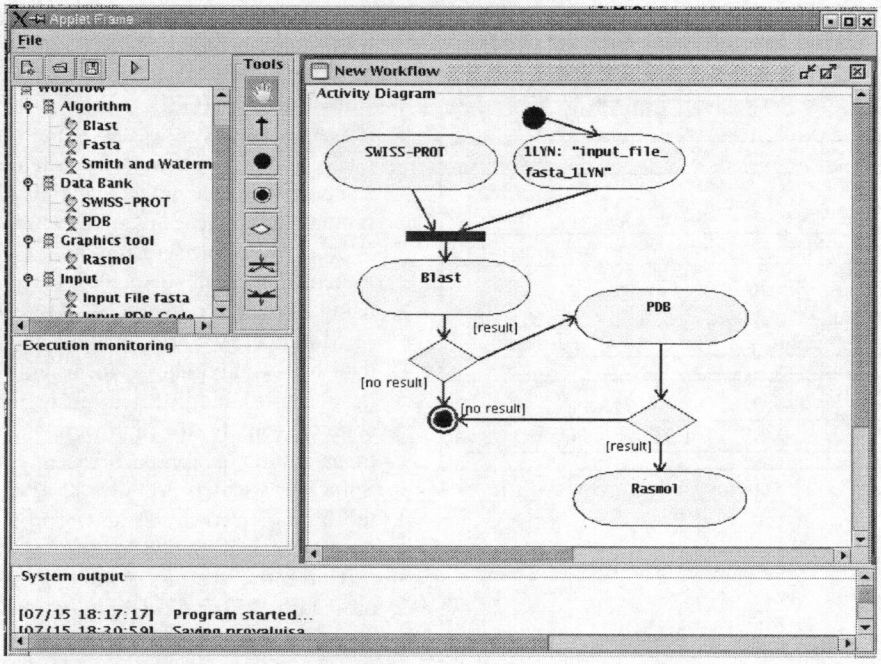

Fig. 6. Workflow of a bioinformatics experiment of sequence comparison.

select Y.Structure *from* PDB *where* Y.AC in (*select* Y.AC *from* Swiss-Prot *where* align[blastP(X, Yi)] > score)

We have searched all of the ACs (Accession numbers) of the sequences of the Swiss-Prot that are similar to the input protein and hence satisfying a given score (applying *blastp* tool [30]). Since the ACs of the Swiss-Prot are present in the PDB, we have selected the corresponding structure for visualizing it with the Rasmol tool [31].
Given a protein, its graphical representation can be compared with respect to each similar protein produced by Blast.
We should express two considerations:

1. All of the tools used in the experiment are run on Grid nodes; for instance for the visualization, we have used GRB library [32] to redirect the output of Rasmol on our desktop, using all of the features of this tool;
2. In the above query we have used a semantic join for characterizing the relation between Swiss-Prot and PDB.

In a simple experiment such as that described above, our data access service is fundamental to access the Swiss-Prot and PDB data banks to retrieve the data. In particular an added value of our DAI service is related to the fact that, as protein sequences are retrieved from Swiss-Prot, their correspondent PDB versions (protein structures) can be recovered by using the information stored in the DAI schemes and ontologies, allowing querying PDB efficiently.

5. Conclusions

The large amount of data sets that today is available from geographically distributed storage sources, is making data integration increasingly important. Integration of data demands significant advances in middleware; distributed infrastructures such as Grids and Web Services can be used for data integration.
In particular coupling these with ontologies is a promising approach to model bioinformatics sources. In this paper we presented the architecture of a semantics-enriched Data Access and Integration service for biological databases. The proposed system extends the classical mediator approach in data integration by introducing domain ontologies in description of data sources and exposing services through the Web Services approach. Compared to other approaches, our system uses Grid protocols such as GridFTP and GSI for fast and secure exchange of data.

In our architecture wrappers are created manually and added to the mediator modifying its source code. We are now focusing our efforts to build a dynamic mediator through semantic mediation. It will allow using semantic information about data sources, such as query capabilities, data provenance, data scheme, etc. The main goal is to provide a method to add wrappers without

source code modifications. A secondary goal is a tool for automatic wrapper generation.

Future work will regard the full implementation of the system and its use inside ProGenGrid, a grid-based service oriented to software environment for bioinformatics applications.

6. References

[1] Fasman, K. H., Letovsky, S. I., Cottingham, R. W. and Kingsbury, D. T. (1996). Improvements to the GDB Human Genome Data Base. Nucleic Acids Res. 24, 57-63.

[2] B., Boeckmann, A., Bairoch, R., Apweiler, M., Blatter, A., Estreicher, E., Gasteiger, M. J., Martin, K., Michoud, C., O'Donovan, I., Phan, S., Pilbout, and M., Schneider. The Swiss-Prot protein knowledge base and its supplement TrEMBL. Nucleic Acids Research 31: 365-370 (2003). Site address: http://www.ebi.ac.uk/swissprot/.

[3] Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F., Brice, M. D., Rodgers, J. R., Shimanouchi, O. K. T. and Tasumi, M. (1977). The Protein Data Bank: a computer-based archival file for macromolecular structures. J. Mol. Biol. 112, 535-542.

[4] Özsu, M.T., & Valduriez, P. (1999). Principles of Distributed Database Systems, 2nd edition, Prentice Hall (Ed.), Upper Saddle River, NJ, USA.

[5] Rice, P. Longden, I. and Bleasby, A. "EMBOSS: The European Molecular Biology Open Software Suite" Trends in Genetics June 2000, vol 16, No 6. pp.276-277. Site address: http://www.ch.embnet.org/EMBOSS/.

[6] SRS Network Browser. Site address: http://www.ebi.ac.uk/srs/srsc/.

[7] WfMC. Workflow management coalition reference model. Site address: http://www.wfmc.org/.

[8] I., Foster, C., Kesselman: The Grid: Blueprint for a New Computing Infrastructure, Published by Morgan Kaufmann (1998).

[9] G. Aloisio, M. Cafaro, S. Fiore, M. Mirto, "ProGenGrid: A Grid Framework for Bioinformatics". Proceedings of International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB 2004), September 14-15 2004, Perugia, Italy.

[10] Kreger, H. "Web Services Conceptual Architecture.", WSCA 1.0. IBM, 2001.

[11] T.T. Wee, M.D. Silva, L.K. Siong, O.G. Sin, R. Buyya, and R. Godhia, "Asia Pacific BioGRID Initiative", Site Address: http://www.apbionet.org/grid/docs/.Presentation Slides at APGrid Core Meeting, Phuket. 2002.

[12] myGrid Project, University of Manchester. Site address: http://mygrid.man.ac.uk/.

[13] GridFTP Protocol. Site Address: http://www-fp.mcs.anl.gov/dsl/GridFTP-Protocol-RFC-Draft.pdf.

[14] M. Lenzerini. Data Integration: A Theoretical Perspective. In Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium of Principles of database systems (PODS), pp. 233-246. ACM Press, 2002.

[15] Widerhold G. "Mediators in the Architecture of Future Information Systems". IEEE Computer 1992; 25:38-49.

[16] Stevens et alt (2000). "TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources". Bioinformatics, 16:2 PP.184-186.

[17] Lange et alt. (2001). "A Computational Support for Access to Integrated Molecular Biology Data".
Site address:
http://www.bioinfo.de/isb/gcb01/poster/lange.html#img-1.

[18] Orengo C.A., Michie A.D., Jones S., Jones D.T., Swindells M.B., Thornton J.M. "CATH – A Hierarchic Classification of Protein Domain Structures". Structure 1997; 5: 1093-1108.

[19] Bairoch, A. (1993). The ENZYME data bank. Nucleic Acids Res. 21, 3155-3156.

[20] The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. Nature Genet. 25: 25-29 (2000).

[21] Aloisio, G., Cafaro, M., Fiore, S., Mirto, M.: The GRelC Project: Towards GRID-DBMS, Proceedings of Parallel and Distributed Computing and Networks (PDCN) IASTED, pp-1-7, Innsbruck (Austria) February 17-19 (2004). Site address: http://gandalf.unile.it.

[22] Van Engelen, R.A., Gallivan, K.A. "The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks.", Proceedings of IEEE CCGrid Conference, May 2002, Berlin, pp. 128-135.

[23] Aloisio, G., Cafaro, M., Lezzi, D., Van Engelen, R.A. "Secure Web Services with Globus GSI and gSOAP", Proceedings of Euro-Par 2003, 26th - 29th August 2003, Klagenfurt, Austria, Lecture Notes in Computer Science, Springer-Verlag, N. 2790, 421-426, 2003. Site address: http://sara.unile.it/~cafaro/gsi-plugin.html.

[24] G. Aloisio, M. Cafaro, S. Fiore, M. Mirto, "Bioinformatics Data Access Service in the ProGenGrid System". Proceedings of the First International Workshop on Grid Computing and its Application to Data Analysis (GADA 2004), October 25-29, Larnaca, Cyprus, Greece, OTM Workshop 2004, LNCS 3292, pp. 211-221, R. Meersman et al. (Eds.), 2004.

[25] I., Foster, C., Kesselman: Globus: A Metacomputing Infrastructure Toolkit, Intl J. Supercomputer Applications, Vol. 11, 1997, No. 2, pp. 115-128.

[26]IBM. Web services flow language -wsfl. Site address: http://www-306.ibm.com/software/solutions/webservices/pdf/WSFL.pdf

[27]IBM. Business process execution language for web services- bpel4ws. Site address: http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/.

[28]OMG. Uml- unified modeling language: Extensions for workflow process definition. Site address: http://www.omg.org/uml/.

[29]R. Eshuis and R. Wieringa. Verification support for workflow design with UML activity graphs. In CSE02. Springer Verlag, 2002.

[30] Altschul, Stephen F., Gish Warren, Webb Miller, Eugene W. Myers, and David J. Lipman (1990). Basic local alignment search tool. J. Mol. Biol. 215:403-410.

[31] Roger A. Sayle and E. J. Milner-White, "RasMol: Biomolecular graphics for all", Trends in Biochemical Science (TIBS), September 1995, Vol. 20, No. 9, p.374. Site address: http://www.umass.edu/microbio/rasmol/.

[32] Aloisio, G., Blasi, E., Cafaro, M., Epicoco, I. "The GRB library: Grid Computing with Globus in C.", Proceedings HPCN Europe 2001, Amsterdam, Netherlands, Lecture Notes in Computer Science, Springer-Verlag, N. 2110, 133-140, 2001.