A Framework for Molecular Biology Data Integration

Sérgio Lifschitz, Luiz Fernando Bessa Seibel and Elvira Maria Antunes Uchôa
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
<lifschitz,seibel,elvira>@inf.puc-rio.br

Abstract: Molecular biology data are placed in different databases, repositories and flat files, usually distributed over the web. Distinct data models with schemas that are often changing implement these heterogeneous data sources. It is very important to gather information about these data sources, including schemas and ontology. The usual approach to handle this information integration problem is to use a single model that captures all the needed data and related methods. Instead, this work proposes the use of a domain specific framework for molecular biology data access and applications. This way we can capture multiple schemas and preexisting data sources, besides having a tool for schema evolution maintenance and database instantiation.

1. Introduction

The main tools used by researchers in molecular biology are associated to files and databases containing information on nucleotide sequences, proteins, or on other biological data related to one or various organisms [13, 25, 20, 12, 2, 21, 10, 18]. There are many projects aiming to raise the genetic code of different organisms. This fact has drawn the proliferation of the data sources applied to the molecular biology. An extensive list of molecular biology data sources as well as their contents is presented in [3].

Some of these sources of information are specialized, i.e., store data of specific organisms or cells; or even focus in a particular biological function. Moreover, some try to trace the mutations and differences found in a gene or group of genes. These data sources often differ in the way they store data, especially sequence, and in the relevant information considered by a research project. Such data sources are associated to applications that also differ in the services offered to the scientific community, such as data visualization (e.g., chromosomes, sequences, other relevant biological information), search tools, sequences alignments and comparisons, among others.

Each research group usually work independently from others, using different data models to represent and manipulate mostly the same kind of information. There are systems implementations that store information related to the genetic code in text files (*e.g.* GenBank [13]), in relational databases (*e.g.* Swiss-Prot [25]) and in object-oriented databases (*e.g.* AceDB [2]).

Since it is a domain in constant evolution, it became important to permit updates on schemas already implemented, suggesting the adoption of more flexible data models. This happens because new biological information and descriptions emerge frequently and it is essential that such information be stored and manipulated in the current data repositories.

Even though it is not clear which data model is the most adequate to represent the biological information, one of the most important aspects refers to its flexibility with respect to schema updates. So, implementations that consider the object-relational model (e.g. AatDB [1]) and the semi-structured model, have been suggested [19].

One of molecular biology research goals is to allow interactions between users and several data sources, so that the latter are viewed as a unique repository. User interactions are related to web-based data access and ad-hoc queries to specific and complex biological objects.

The usual approach to handle this information integration problem is to use a single model (e.g., OPM [6]) that captures all the needed data and, sometimes, related methods. Instead, this work proposes the use of a domain specific framework for molecular biology data access and applications, with the following properties: (i) schema definition related to a specific biology research based on different, preexisting data sources; (ii) schema evolution maintenance; (iii) definition mechanisms for ontology used in different data sources; (iv) appropriated tools for data access; (v) database instantiation related to the defined schema; and (vi) data availability for different applications.

This paper is organized as follows. In the next Section we present some molecular biology applications that will be integrated within the framework. Then in Section 3 we discuss some of the existing approaches and related work, besides explaining the underlying idea of an integration approach based on a framework. Section 4 gives an overview of the proposed framework and we finis h with conclusions, ongoing and future work in Section 5.

2. Molecular Biology Applications and Tools

There are many tools in the web, for immediate use or even source code download, available to molecular biology researchers and a very extensive list can be found in [22]. These tools may be organized in depuration and sequences submission systems, applications for sequences analysis, databases and forms for use through the Web.

Some research groups that perform frequent sequencing tasks use systems for depuration and sequence submission (e.g. [23]). These systems criticize the output of the sequencers, according to a pre-established criteria, identify and drop the vectors used in the biological reaction, and submit the sequences to generic databases (e.g., GenBank [13]) automatically, in order to verify if the sequence obtained already exists in the database. If there is a negative answer, the sequence may be stored in the database. The research group or even the single researcher that submitted the sequence becomes responsible for the quality of the given information. Otherwise, the sequence is rejected.

The applications for sequences analysis are tools largely spread, and implement important functions such as sequence comparisons with BLAST-like (or FAST) algorithms, sequences alignments, search on genes (Gene Finder), among others. Several of these applications make their source code available. The set of the most used applications is in fact a powerful research support tool.

Molecular biology databases are important tools for storage and query of the data resulting from the research projects, such as nucleotide sequences, genes, mutations, protein structures, taxonomy and other relevant biological annotations and experiments conditions. Some of these databases use commercial database management systems (e.g. [25, 15]), others are data repositories with their own format (*e.g.* GenBank [13]), and some use databases systems specifically developed for a given genome project (*e.g.* AceDB [2]).

Most of these databases (actually not always database systems but rather simple data sources) are inserted in a broader tool that also makes available to the users some of the previously mentioned functionalities, in particular those for sequences analysis. Particularly, AceDB [2] presents an ad-hoc interface of its own to visualize the data through a graphic interface that provides a special form of presentation of relevant sites of a chromosome (chromosome map). There is also a *drill down* like functionality where from a simple mouse click in a chromosome region, the map of relevant biological information sites inserted in that region is exhibited, and like this successively, until one reaches the most basic information, which is the sequence itself.

There are many tools for molecular biology research available for use via the Web. In this case the search tools refer to a database, or even to pre-defined databases, where a form is exhibited for the definition of particular queries. These are sent to the databases and the answers are grouped for presenting the final result. Several of these tools have pre-defined input data format. Thus, if the data set can be converted to the appropriate format, the tools may be used with no need for changing the source code.

3. Why Integration through a Framework?

The main tools used in molecular biology are associated to files and databases that store information about nucleotides, proteins, and other biological data related to one or many organisms. As there are multiple projects in this area, multiple and different data sources have been created and it becomes difficult to access all of this information.

Existing approaches to integrate molecular biology information are based in specific tools that consider a single data model and schema. There is often a lack of functionality and/or performance, as only part of the data and related functionalities are usually considered.

Some integration tools deal with hypertext navigation, that allow the users to jump through registers of different data sources either through existing links among them [9] or navigation systems that create the links among different data sources [26]. Thus, in a first movement, the user accesses a data source register and, in what follows, the user asks for a link to another data source where the desired information is.

There exist integration tools that implement queries to the different pre-existing data sources (much like multidatabases). These queries may be formulated through special languages [4, 7, 8] that allow representing complex data types, with an access driver implemented for each data source to be accessed. Another strategy consists of using a mediator that is responsible for determining the data sources that participate in the query, creating an access plan, translating concepts and syntax, assigning the queries to the distributed environment and integrating the results [17]. Implementations of these strategies present quite slow results in distributed environments such as WWW.

Some integration tools deal with the implementation of a data instance that collects the biological information available in several sources (like data warehouses do) [16]. In this case, there is a good performance for processing queries; although it becomes quite bad in the aspect of auditing the schemas of the data sources. This happens very often, which demands frequent maintenance of these tools.

We have chosen here an approach that is based on object-oriented frameworks, due to its flexibility and extensible architecture. A framework is, in a few words, an incomplete software system to be instantiated. A framework contains many basic pre-defined components (*frozen spots*) and other that must be instantiated (*hot spots*) for the implementation of the desired and particular functionality.

The basic idea for choosing a framework approach is that we have been looking for a single data model and architecture that could be adopted in the molecular biology databases context but all possible approaches may fail in data representation. Moreover, it becomes hard to deal with schema evolution and the execution of new applications and methods. With our framework, it is possible to capture distinct schemas from different data sources in order to define a general schema for specific manipulations. Also, multiple schemas can be created and the related databases instantiated and used in particular applications.

Our proposed framework may be considered specific to the application domain, in our case, the one related to molecular biology and genome projects [5, 11]. In the next section we will briefly describe and discuss the framework functionalities and architecture.

4. Framework Overview

This section describes the main characteristics of the proposed framework, besides its general architecture. Frameworks offer an adequate infrastructure to fulfill the requirements of integrating the data sources and making them available to the different applications. Since a framework is a semi-complete software system, it is created with the aim of being instantiated. A framework defines an architecture for a family of systems, and provides pre-defined basic components, together with others that must be instantiated to reach the desired functionality.

Our proposed framework may be described (due to space limitations) through its main functionalities. We can start with the ability of capturing distinct schemas from several of the existing molecular biology data sources, so to give an integrated view of all the information required.

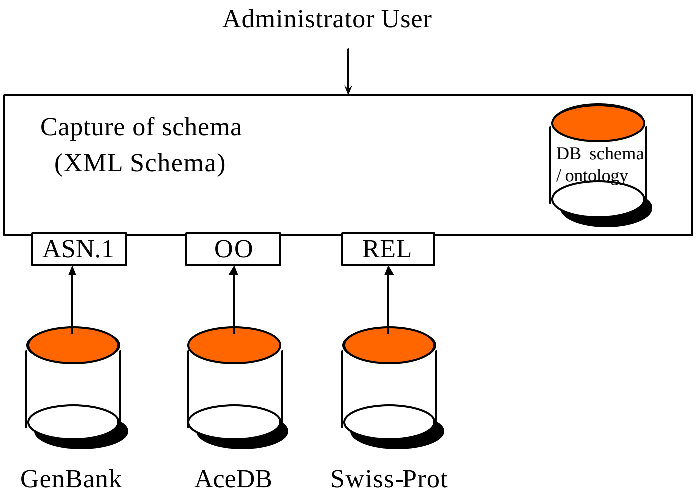

Figure 1 – Capture of Schemas

The basic idea is shown in Figure 1, where there is a repository/schemas database to store the desired information. There we give as examples 3 well known data sources: GeneBank (text, semi-structured), AceDB (object-oriented) and Swiss-Prot (relational). This process is supported by a basic ontology, also available, that is stored and added to the terms used in each distinct data source whose schema is captured. The representation of the schemas is based on the XML Schema pattern language. This choice is quite immediate due to the known advantages of XML and semi-structured data models in this context [14].

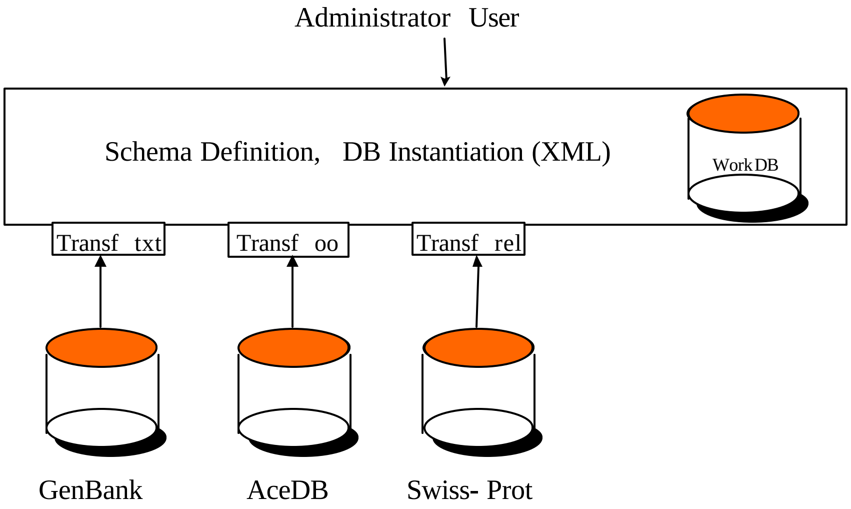

Figure 2 – Definition, manipulation and instantiation of a work database

With these schemas, it becomes possible to the users to define a *work database*, which is specific for their research and goals. Once defined, the instantiation of the work database is done with data coming from the pre-existent data sources that are relevant to the particular applications. This work database may suffer schema updates that are originated at the data sources that are part of the integrated schema, as well as from new knowledge informed by the users (Figure 2). The data in the work database is stored in XML format. The work database schema definition, as well as its updates, is supported by the ontology that is being created and manipulated by the users.

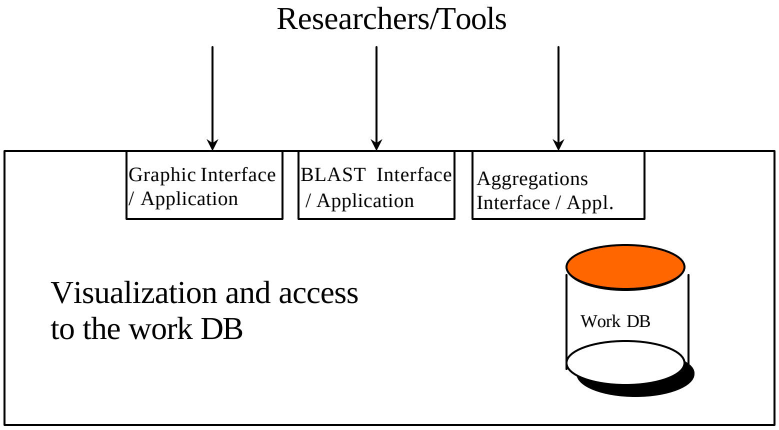
Researchers/Tools
Graphic Interface / Application
BLAST Interface / Application
Aggregations Interface / Appl.
Visualization and access to the work DB
Work DB

Figure 3 - Access and visualization of the data

The framework may create many interfaces between the work base and the multiple applications used in molecular biology. An application layer is made available for the users, as each application request a proper interface (Figure 3). For instance, the FASTA format is needed if one will be executing the FAST or BLAST similarity search family of algorithms. Transformations are then performed in the work database data, which are then draw to availability in order to feed the different applications.

The framework being proposed is divided in four modules: *Administrator, Captor, Driver and Converter*. Their relationship and an overview of the framework architecture is given in Figure 4:

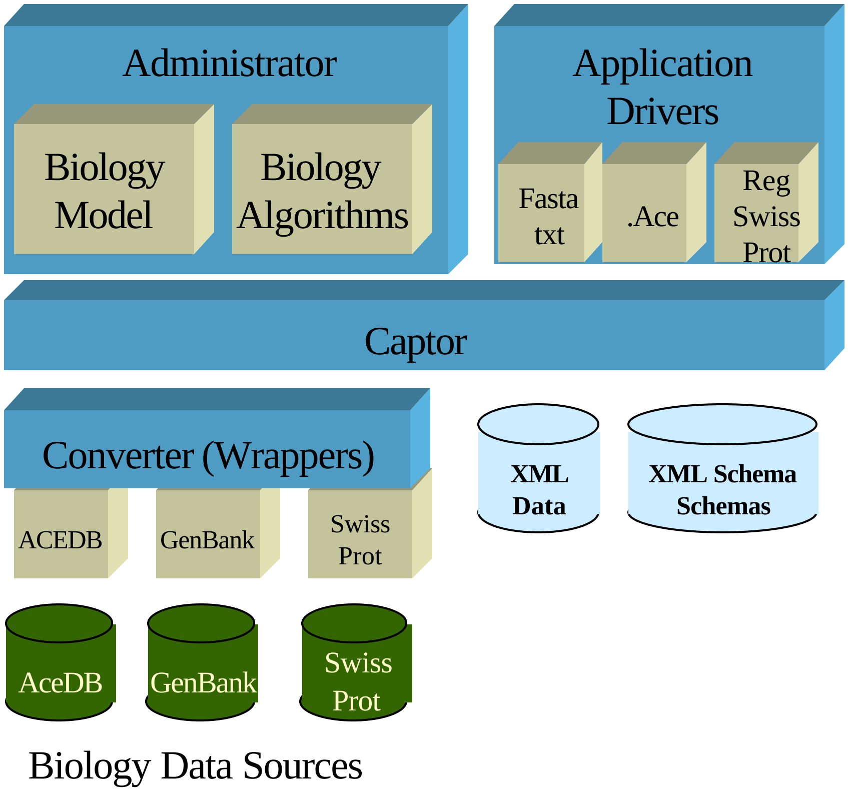

Figure 4: Framework architecture overview

The Biology Model and Algorithms, Wrappers associated to biology data sources and Application Drivers are the framework hot spots. When instantiated, they implement a particular functionality of the architecture, defining an application over the molecular biology application domain.

The Administrator module performs the interface with the users in a way to provide management of the biological data model; request for capturing schemas and/or data; and permit the execution of the algorithms instantiated in the framework itself. Therefore, this module contains a biology class model that is committed with the existent data sources, as well as with the methods that are associated to these classes.

The Captor module is responsible for the data and architecture schemas repository. The Converter implements the access to the biology data sources, making the translation of the data sources schemas to XML Schema and data for XML. The Drivers implement the interface generation between the biology applications and the *framework*. A more detailed presentation of our framework can be found in [24].

5. Conclusions

The framework proposed here is still under development. Some issues still remain to be solved, as well as many details concerning its implementation and its effective use. A prototype will be soon available with complete functionality, although with restricted access to data sources.

The proposed architecture helps the molecular biology research in the following aspects: provide access to a heterogeneous environment of data sources, in which heterogeneity occurs in several levels; enable the representation of frequently changing data sources schemas that demand complex data; permit the execution of applications without depending on their associated database; and make the incremental integration of the information available in the data sources feasible.

Moreover, our framework deals with the schema evolution based on a meta-model, *i.e.*, its evolution is independent of each distinct data model used by the data sources. This issue is further investigated in [24].

Besides implementing the framework discussed here, we are currently working with the definition and representation of an specific ontology for the molecular biology area, effectively used in the existing data sources.

Acknowledgments: The authors would like to thank Wim Degrave and Antonio Miranda, researchers from FIOCRUZ Biology and Biochemical department, directly involved in the Tripanossoma Cruzi Genome Sequencing, for their continuous help with the application domain data and functional requirements.

References

[1] AatDB: http://weeds.mgh.harvard.edu/

[2] AceDB: http://genome.cornell.edu/acedoc/index.html

[3] M. Ashburner and N. Goodman, “Informatics – Genome and Genetics Databases”, Current Opinion in Genetics & Development 7, 1997, pp 750-756.

[4] P. Buneman, S.B. Davidson, K. Hart, G.C. Overton and L. Wong, “*A Data Transformation System for Biological Data Sources*”, ~~Buneman P., Davidson S.B., Hart K., Overton C.,~~ Proc~~.~~s of ~~–~~VLDB Conference, 1995, pp 158-169.

[5] G. Booch, "*Designing an Application Framework*", Dr. Dobb´s Journal 19(2), 1994, pp 24-30.

~~.~~

[6] I.A. Chen and V.M. Markowitz, “*An Overview of the Object Protocol Model (OPM) and the OPM Data Management Tools*”,~~Chen I.A., Markowitz V.M~~ ., Information Systems 20(5), 1995, pp 393-418.~~.~~

[7] CPL/Kleisli: http://www.cis.upenn.edu/~db/home.html

[8] S.B. Davidson, C. Overton and P. Buneman, *“Challenges in Integrating Biological Data Sources”*, Journal of Computational Biology 2(4), 1995, pp 557-572.

[9] Entrez: http://www.ncbi.nlm.nih.gov/Entrez/

[10] Enzyme: http://www.expasy.cbr.nrc.ca/enzyme/

[11] M.E. Fayad, D.C. Schmidt and R.E. Johnson, "*Building Application Frameworks*", Addison-Wesley, 1999.

[12] GDB: http://www.gdb.org

[13] GenBank: http://www.ncbi.nlm.nih.gov/Genbank/index.html

[14] V. Guerrinia and D. Jackson, "*Bioinformatics and XML*", On Line Journal of Bioinformatics, 1(1), 2000, pp 1-13.

[15] GSDB: http://www.ncgr.org/gsdb

[16] Integrated Genome Database (IGD): http://igd.rz-berlin.mpg.de/~www/lpi.html

[17~~]~~] P. Karp, “*A Strategy for Database Interoperation*”, ~~Karp P.,~~ Journal of Computational Biology 2(4), 1995, pp 573-586~~.~~

[18] Mitomap: http://infinity.gen.emory.edu/MITOMAP

[19] S. Navathe and A. Kogelnik, *"The Challenges of Modeling Biological Information for Genome Databases"*, Conceptual Modeling (Chen et al, eds), LNCS 1565, 1999, pp.168-182.

[20] PDB: http://www.rcsb.org/pdb

[21] PUBMED: http://www4.ncbi.nlm.nih.gov/PubMed/

[22] http://www.public.iastate.edu/research_tools.html

[23] S. Rozen, L. Stein and N. Goodman, *"LabBase: A Database to Manage Laboratory Data in a Large-Scale Genome-Mapping Project*", Rozen S, Stein L, Goodman N, IEEE Engineering in Medicine and Biology,IEEE Engineering in Medicine and Biology, 14(6), ~~nov/dec~~ 1995, pp 702-709.

[24] L.F.B. Seibel and S. Lifschitz, "A Genome Databases Framework", PUC-Rio Informatics Department Tecnical Report, 2001 (submitted for publication).

[25] Swiss-Prot: http://www.ebi.ac.uk/swissprot

[26] SRS: http://expasy.cbr.nrc.ca/srs5