BIO-BROKER: A BIOLOGICAL DATA AND SERVICES MEDIATOR SYSTEM

Aldana, José F.
Languages and Computer Science Department, University of Malaga
ETSI Informática, Campus de Teatinos, 29071 Málaga, Spain

Hidalgo-Conde, Manuel
Computer Architecture Department, University of Malaga
ETSI Informática, Campus de Teatinos, 29071 Málaga, Spain

Navas, Ismael
Languages and Computer Science Department, University of Malaga
ETSI Informática, Campus de Teatinos, 29071 Málaga, Spain

Roldán, María del Mar
Languages and Computer Science Department, University of Malaga
ETSI Informática, Campus de Teatinos, 29071 Málaga, Spain

Trelles, Oswaldo (*)
Computer Architecture Department, University of Malaga
ETSI Informática, Campus de Teatinos, 29071 Málaga, Spain

(*): To whom correspondence should be addressed

ABSTRACT
Diversity, dispersion and heterogeneity in data and services strongly constrain the integrated exploitation of biological data. Bio-Broker, an XML-based architectural framework, aims to assist systems developers in the construction of mediator-services. The unique characteristic of the platform is its ability to manage software tools and algorithms to allow dynamic, flexible and intuitive “wiring” of new services expanding the functionality and enabling the easy incorporation of new procedures to customize the system for specific concerns. The mediator offers a view of the system as a single data source where EVASs (Extended Valued Added Services) are readily available for enhancing query processing. A diverse set of applications that combine gene expression and genomic data are used to demonstrate the usefulness of the mediator system.

KEYWORDS
Mediation, Bioinformatics, EVAS, Integration, Architectures

1. INTRODUCTION
The accumulated biological knowledge needed to produce a more complete view of any biological process is disseminated around the world in the form of sequences, motifs, 3D-structures, pathways, gene-expression data, etc. Unfortunately, this valuable information is often dumped in proprietary data models and specific services are developed for data access and analysis, without forethought to the potential external exploitation and integration of such data.
The problem of heterogeneous data integration has been addressed by using wrappers to translate the data sources structure into a common model (Roth et al. 1997; Sahuguet et al. 1999), or by using mediators for encapsulating the knowledge needed to evaluate a query over multiple wrappers. The wrapper-mediator approach sets up an interface for a group of data sources, amalgamating their local schemas into a global one and integrating the information of the local sources. As result, the views of the data that mediators offer are coherent, producing semantic reconciliation of the common data model representations. Some examples of the wrapper-mediator systems are Tsimmis (Garcia-Molina et al. 1995) and Garlic (Haas et al. 1997). Some examples in the biological field are: Tambis (Stevens et al. 2000), BioDataServer (Lange et al. 2001), Kind (Gupta e al. 2000), BioZoom (Ling et al. 2003), and DiscoveryLink (http://www.ibm.com/discoverylink). With the advent of XML some of the mediator systems evolved toward the standard (Amos and Tsimmis), whilst other projects were initially XML-based like MIX (Baru et al. 1999). Some of them, -i.e. BioMOBY (http://plantgenome.sdsc.edu/mobyed2/white_paper.html )- provide a way to create web services, as well as a central registry with information on all the available services although in general are not designed to provide mediation services.

However, in the bioinformatics context, data analysis often involves the use of a variety of interconnected tools, defining in some way a processing pipeline that define complex virtual bioinformatic machines (Butler et al. 2002). These workflows are built mainly by hand, which may be a source of errors and involves a time cost. Thus, it is important both automate the task and allow dynamic conection of tools. We propose the use of EVAS for making the workflow construction flexible and simple. Each EVAS encapsulates a software tool as a black box that may be used as a building block which can be connected to another EVAS building the desired workflow in almost automatic way. The dynamic and flexible use of EVAS is an outstanding characteristic of our system which is not provided by traditional or biological mediators systems.

As an example of the potential utility of Bio-Broker we present the integration of gene-expression data with genetic information and the integration of various processing mechanisms developed by different groups through a graphical interface that allows fast and intuitive “wiring” of EVASs components, expanding the functionality of the individual tools and enabling the customization of the system. Uniform access to different data sources will be shown as well as integrated access to several processing services.

In the following both, the architecture design and the development mechanisms needed to build up the mediator will be described. Bio-Broker will be then introduced and selected exercises on gene-expression data will be used to illustrate the system.

2. SYSTEM AND METHODS

2.1 An XML based Architecture for Mediators Design.

The mediator system architecture (Aldana et al. 2002) is based on the use of XML standards. The key characteristics of the architecture and its main components are depicted in Figure 1:

1. The architecture is XML-based. XML (http://www.w3.org/TR/2002/CR-xml11-20021015) is itself a standard format for the representation, interchange and validation of data and, together with the XQuery query language (http://www.w3.org/TR/xquery/) and the XML Schema (http://www.w3.org/TR/xmlschema-(0, 1, 2)/), forms a complete data model for dealing with information on the web. XML Schema and RDF Schema (http://www.w3.org/TR/2000/CR-rdf-schema-20000327/) are used for describing all the metadata necessary for query processing in this architecture as well as for expressing the data resulting from the queries.

2. Xquery: all the clients connected to a mediator must send a valid XQuery query to the mediator and will receive an XML document. The client can be any sophisticated query generation mechanism that must send a valid XQuery query independent of how this query is generated.
3. The *query processor* receives a query and produce a sub-query for each of the data sources involved in the query. It analyses the data requirements of the query, identifying the data sources and generating a global plan for evaluating the query. The global plan contains information on the various alternatives for solving the same data requirement.

4. *Metadata* are used to express all information needed for the above process. This includes information both on the logical schemas produced by the wrappers and on the semantic equivalence of the terms produced by the data sources (the integration schema). We also store information about the query processing capabilities of these wrappers and about the location and availability of the data sources for optimising access to the data. Specific user information like privileges, EVASs usage and preferences is also stored.

5. Extended Value Added Services (EVASs) are all those processes that allow the user to further filter or post-process the queries. In a sense EVASs are modelling arbitrarily complex repetitive processes appearing on the data obtained from the data sources and shared by several users. Obviously these processes (e.g. BLAST, AnaGram, Frags) being application specific are not efficiently incorporated into the query language. On the other hand, being repetitive tasks performed by several users their automation and optimisation should be assisted by the mediator.

6. Wrappers translate the data sources into the common data model (XML/Xquery). Wrappers hide the internal complexity of the data sources and offer, in a controlled manner, their query capabilities. There is one wrapper for each data source integrated into the mediator. Wrappers receive XQuery queries and produce XML documents which conform to the XML schema (integration schema).

### 2.2 From Architecture to Mediators.

A framework has been developed for simplifying the mediator construction task. The mediator instantiation process involves the construction of a particular application of the framework by the development of wrappers and the use of metadata for the configuration of the different mediator components. Wrappers are the means of translation between the reference model of the architecture (XML) and the data source model. Once the mediator has been constructed, the mediator user can dynamically manage the EVASs adding or removing them as necessary.

The main tasks involved in mediator configuration are: (a) the wrapper construction; one is needed for each data source (wrappers for the more frequent types of data sources are available: relational databases, XML documents and HTML documents); (b) *Mediator configuration through metadata*: specifically they will configure both; the *Location scheme*–to link wrappers to the mediator- and the *Integration scheme* which specifies the data translations between the data model used by the client and that offered by the wrappers.

An EVAS is a software tool adapted for being connected to the mediator. EVASs are not in the mediator core, but are linked dynamically during the use of the mediator system. Therefore, although the EVASs configuration is made using metadata, the framework offers to the mediator-users a management tool to connect, arrange and remove EVASs without any information on the mediator internal architecture, thus, enabling a wide range of processing alternatives to be specified for relating—in different ways—diverse sources of biological data.
The integration of EVAS with the mediator consists of two main phases: (a) creation of the EVAS and (b) connection of EVAS to the mediator. Step (b) involves: (a) publish the EVAS as a web service; (b) register the web service in the system; (c) obtain a DLL library, which performs a call to the web service, and (d) register the library in the system.

When a set of EVAS has been developed and connected to the system, users can create workflows, taking advantage of a graphical tool provided by the system. Finally, once the mediator has been developed and managed, it is ready for use by the final clients. The clients can use the mediator either through a programming interface or through a graphical interface. The framework offers a graphical interface development tool.

2.3 Biological database integration: Bio-Broker.

The platform will be used to integrate gene expression applications. In general, in this type of application, clustering is the classical analysis and, once the data has been clustered, post processing analysis aims to extract relevant knowledge. In a sense, clustering analysis is incomplete without integrating it with functional information: functional annotations contained in nucleotide and protein databases; pathways, etc.

At present Bio-Broker integrates different type of data sources: the nucleotide EMBL (Wang et al. 2002) and protein SWISS-PROT sequence information (Bairoch et al. 2000), the worldwide repository of three-dimensional structures of biological macromolecules -the Protein Data Bank, PDB (Bourne et al. 2003) at http://www.rcsb.org/pdb/; the MiCrobial Advanced Database Organization (MICADO) -a relational database dedicated to microbial genomes (Samson et al. 2000) and functional analysis of Bacillus subtilis, the DIP™ -Database of Interacting Proteins- (Xenarios et al. 2002) and the BIND -Biomolecular Interaction Network Database (Bader et al. 2003) designed to store full descriptions of interactions, molecular complexes and pathways.

These databases have been selected on the basis of the difference in content, format, access mechanism, and geographical location. Our intention is to show how these very diverse data sources can be easily integrated using the proposed architecture and how the possibility of easily adding EVASs allows to mix different tools for obtaining enhanced data processing capabilities. Services requested of the mediator help to recover information associated with a given set of clustered genes.

Figure 2. Bio-Broker Architecture: Data access, pre-processing, integration, post-processing and user interfaces
3. RESULTS

3.1 Bio-Broker Architecture.

Bio-Broker architecture is shown in Figure 2. A user interface provides access to the services. The queries, expressed in terms of the integration schema, are sent to the server, which divides it into different sub-queries which are sent to the different databases. Each database has its query and XML result document construct mechanism. The service receives the sub-query results over which it is possible to apply an EVAS to process the sub-query results. A new EVAS can be applied to the first EVAS process result, and so on.

Then results of each sub-query, are integrate removing duplicates and inconsistencies. Removing duplicates can be performed taking advantage of the integration schema and the mappings between this schema and resource schemas. Data received by the mediator are expressed in terms of the integration schema, and the mappings allow to know if two data items are duplicated and to remove one of them from the integrated data.

Next, this integrated solution can be post-processed by a set of EVAS. Finally, the last result is sent to the user interface. It is worth observing that where and how these EVAS’s are implemented is completely irrelevant: they are just web services integrated into the mediator system.

The integration schema of Bio-Broker stores fields for which users can make queries, and their provenance (the database from which the data is obtained). This is an important issue due to the necessity of assessing the quality of data in the biological context. This quality is obtained taking advantage of well known biological web services, such as EMBL, PDB, etc. Thus, we can for example annotate the GeneName field with metadata showing that it can be obtain from EMBL, GenBank or MICADO databases.

Furthermore, addition of EVASs in the integration process has not penalty in the query processing. We only must to add the time of making use of these tools to the time required to retrieve data. Furthermore, the cost of using tools is always less than using them manually by the end-user once obtained integrated result.

That is, it is usual that users need to apply filters and transformation tools to data retrieved from data sources. This repetitive task, which is manually done for each data has a high cost that will be decreased if we apply them automatically. Besides, execution of workflows is automatically performed taking advantage of metadata, so the cost of maintenance is limited to definition of workflow that is done according to user needs.

Table 1. Requested services. Four examples are shown to describe different levels of complexity, and the different databases and services that can be requested and combined in a simple query. The following labels have been used for metadata information: <S> Specie, <G> Gene cluster, <P> Prefix length, BS=Bacillus subtilis.

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
<th>Process diagram</th>
<th>Result information</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Single service request</td>
<td>[Diagram of Bio-Broker architecture]</td>
<td>[Diagram of Bio-Broker architecture]</td>
<td>[Example of Bio-Broker architecture]</td>
</tr>
<tr>
<td>Description</td>
<td>Genes of a specific species from a specific database.</td>
<td>[Diagram of Bio-Broker architecture]</td>
<td>[Diagram of Bio-Broker architecture]</td>
<td>[Example of Bio-Broker architecture]</td>
</tr>
<tr>
<td>Procedure</td>
<td>Send a query to WEDO database.</td>
<td>[Diagram of Bio-Broker architecture]</td>
<td>[Diagram of Bio-Broker architecture]</td>
<td>[Example of Bio-Broker architecture]</td>
</tr>
<tr>
<td>Method</td>
<td>CDiO call</td>
<td>[Diagram of Bio-Broker architecture]</td>
<td>[Diagram of Bio-Broker architecture]</td>
<td>[Example of Bio-Broker architecture]</td>
</tr>
<tr>
<td>Data Description</td>
<td>Request sequences of a specific species.</td>
<td>[Diagram of Bio-Broker architecture]</td>
<td>[Diagram of Bio-Broker architecture]</td>
<td>[Example of Bio-Broker architecture]</td>
</tr>
</tbody>
</table>
3.2 Bio-Broker Services.

We present the integration of gene expression data with several biological databases and tools described previously. The set described by Kobayashi et al., 2001 will be used. This set is composed of the cDNA expression levels of 4005 genes of Bacillus subtilis two-component regulatory systems in which the overproduction of a response regulator of the two-component systems, coinciding with a deficiency of its cognate sensor kinase, affects the regulation of genes, including its target ones. The genome-wide effect on gene expression caused by the overproduction was analysed on 24 two-component systems (http://www.genome.ad.jp/kegg/expression).

The *engene* platform (García de la Nava, et al, 2003 at http://chirimoyo.ac.uma.es/bitlab) was used to cluster the gene expression data. The mediator system is available at http://uranos.khaos.uma.es/mediator. Table 1 describes four samples of implemented queries with different levels of complexity, and the different databases and services that can be requested and combined in a simple query.

The first service (Service A in Table 1) request the system for information about the “genomic context” of clustered genes, whose more straightforward definition is based on the physical proximity of genes in the chromosome. It is well accepted that neighbouring genes in bacteria are often functionally related, so the gene/cluster distribution provides additional information for interpreting expression data. This service works on MICADO, which contains detailed information on B. subtilis among other bacteria (see Figure 3).

![Figure 3](image3.png)

Figure 3. Graphical representation of output from service A. Red and green colours have been used to represent up- and down-regulated genes, and forward and backward gene direction is represented as out- and in-side the chromosome.

Position is counted clockwise. A clear proximity relatedness can be observed with potentially useful biological information.

The objective of the second request is, given a set of related DNA sequences clustered by their gene expression data, to end up with a set of over-represented fragments identified in the up-stream positions which could represent putative promoters or activation signals of those genes. In this case, there are two very simple EVASs in the pipe: Prefix and Repeats, the former to obtain the up-stream section of the DNA sequences, and the latter to identify over-represented strings of length K from those up-stream sequences.

The third exercise aims to supply information not only about the original sequences, but about sequences related to the original ones. In this case the input is also a collection of DNA sequence identifiers and the final output is a collection of Keywords to be used in an association rules discovering procedure. The workflow is: first, from EMBL get the accession numbers of the protein sequences that correspond to the collection of genes (e.g. /db_xref="SWISS-PROT:P37800"). Second, obtain the sequences and keywords corresponding to those proteins. Third, BlasP is launched to complete a database search for similar sequences using each of the original proteins as query sequences. Finally, the sequences, keywords and identifiers of the similar sequences are also obtained. These are used as input to the Frags procedure which is able to detect statistically significant patterns from the collection of sequences. Sequences, patterns and keywords are used as input to ASSRUL, a procedure to detect association rules (available in the engene platform) that correlate the presence of some patterns with functional keywords. This is a complete exercise, in which the ability to use EVASs in the pipeline is demonstrated. This allows the user to dynamically incorporate additional information into the gene expression data.

Finally, in the last service the possibility of combining gene-expression data with structural information is explored. The fragments obtained as partial output in service C, are seek to verify if these fragments –with
strong association at the sequence level- are also associated at structural level. A collection of PDB codes, start-end positions of the fragments conform the input and 3D alignments of such fragments are the output.

Table 2. XML string output for service C. This service request for the protein sequences and associated functional keywords of the genes belonging to a collection of clusters. The rules’ significance can be established upon a variety of parameters calculated by the system: confidence (probability of rule satisfaction), support (examples in data covered by the rule), coverage (examples in data covered by the antecedent of the rule), improvement (how much more frequent is the occurrence of the rule than normal) and leverage (additional examples covered by the rule above those expected).

<table>
<thead>
<tr>
<th>Query</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOR Iteration IN INTEGRACION WHERE (Iteration/Organism/data()='Bacillus subtilis' AND (Iteration/GeneName/data()='lplB' OR (Iteration/GeneName/data()='licC' (Iteration/GeneName/data()='opuBD'))))</td>
<td></td>
</tr>
<tr>
<td>Blast results for translated protein yesS gene</td>
<td>yesS E-value Swissprot Keywords</td>
</tr>
<tr>
<td>O31522 0. DNA-binding; Transcription regulation;</td>
<td>Q9KFJ6 2.6e-38 DNA-binding; Transcription regulation</td>
</tr>
<tr>
<td>O32071 8.1e-37 DNA-binding; Transcription regulation</td>
<td>O30502 1.6e-20 DNA-binding; Transcription regulation</td>
</tr>
<tr>
<td>Q9KE68 3.2e-16 DNA-binding; Phosphorylation; Sensory transduction; Transcription regulation;</td>
<td>Q9KBL6 1.5e-14 DNA-binding; Phosphorylation; Sensory transduction; Transcription regulation;</td>
</tr>
</tbody>
</table>

Let us see with some more detail the third service -a similar procedure to service B-. The objective is to obtain functional information both from the original sequences and from those sequences related to the original ones. Table 2 presents the query and results from the mediator. Firstly the mediator requests the protein sequences and its associated keywords corresponding to a given collection of genes. Results from this request are used to launch a Blast database search. Blast procedure retrieves a set of related sequences, and the mediator is used to extract the sequence and its associated keywords (see Table 2). A data mining procedure (Ffrags) is used to detect statistically significant patterns from the collection of sequences; and finally, the ASSRUL procedure detects association rules. From the implementation point of view, the Service C construction requires the creation of a wrapper for EMBL, and the creation and configuration of three EVAS (E-BlastP, E-Frags and E-ASSRUL). These EVAS are published as web services and interconnected in Bio-Broker using a graphical tool provided by the framework. The easy connection of EVASs allows to solve these examples without knowledge of the internal details of the connected services. This characteristic is especially suitable for allowing different users to share tools with which are unfamiliar.

4. DISCUSSION

In this work we have presented an architectural design that allows an easy and transparent integration of
heterogeneous sources of information. This architecture is especially suitable for collaborative environments in which users wish to include their own specific software tools (EVASs) with minimal intervention and expertise. The inclusion of EVASs in the data flow is the key feature of the architecture and allows filtering, restructuring and processing data without requiring any knowledge of the EVASs implementation. The framework provides a set of templates that can be readily adapted and refined to meet user needs, incorporating new databases and algorithms for integration.

Different use cases have also been developed to demonstrate how this architecture and its associated framework allow the rapid development of domain specific applications. The development of Bio-Broker is a proof of concept of the suitability of this kind of architecture for the bioinformatics domain. In particular, the usefulness of the mediator system is demonstrated by a diverse set of applications aimed at combining expression data with genomic, sequence-based and structural information, so as to provide a general, transparent and powerful solution that goes beyond traditional gene expression data clustering.

We are currently working on the integration of gene expression data with pathways information, which would be a significant development. There are several metabolic databases available that contain several hundred manually drawn pathway maps, but in some cases their static and separate diagrams do not provide flexibility enough to integrate the information.

ACKNOWLEDGEMENT

This work has been partially supported by grant “GNV5-Bioinformática Integrada” from Genoma-España and by the MCYT grant TIC2002-04586-C04-04. The authors would like to thank Dr. Andrés Rodriguez and Dr. Antonio Pérez for clever recommendations.

REFERENCES

Baru et al, 1999. “XML-Based Information Mediation with MIX”. In Exhibitions Program of ACM-SIGMOD.