Mediator-Based Architecture for Integrated Access to Biological Databases

Teh Chee Peng, Wahidah Husain, Rosni Abdullah,
Rosalina Abdul Salam, and Nur'Aini Abdul Rashid

School of Computer Sciences, 11800 Universiti Sains Malaysia, Penang, Malaysia
cp.teh@plexus.com,
{wahidah, rosni, rosalina,nuraini}@cs.usm.my

Abstract. The amount of biological information accessible via the internet is growing at a tremendous rate. The biologists often encounter problems in accessing huge amount of widely spread data due to disparate formats, remotely dispersed and varying implementation on different platforms and data models. Besides that, the custom browsing and querying mechanism implemented in different data sources requires the users to uniquely query each individual database. Instead of having different custom interfaces to these public bioinformatics databases, an intuitive common integrated data access method is proposed to provide uniform transparent access to disparate biological databases. The proposed mediator-based architecture consists of three conceptual layers namely Query Formulation, Query Transformation and Query Execution.

1 Introduction

With the advancement of the microbiology technologies such as genome sequencing, microarray gene expression and mass spectroscopy, various public-accessible bioinformatics database has been developed. Simple retrieval of data from individual biological databases by multiple sequential steps is no longer practical for modern biological research. A specific biological research may require data analysis against multiple data sources and the results are fed into various application programs such as functional domain, protein structural prediction, and motive identification for various biological research purposes. The objective of this work is to create an integrated tool that access to multiple biological databases. This is implemented by applying a mediation framework that implements a transparent access to the heterogeneous data sources [1]. One main requirement of integration is that, the owner of the data maintains autonomy of each data source.

2 Proposed Design Architecture Framework

In the mediator-based architecture, the strongest feature is that it does not store any data and does not require a copy of database to reside in the local storage [2]. The queries are executed remotely in the respective data source and the results returned to

K.-M. Liew et al. (Eds.): PDCAT 2004, LNCS 3320, pp. 13–16, 2004.

end user. In this work, an integrated access to bioinformatics databases is devised based on a mediator-based architecture. The integration structure consists of three major components:

(i) GUI-based Query Formulation
(ii) Query Transformation
(iii) Query Execution

As illustrated in Fig.1, the three major components listed are mapped into a mediator-based architecture where the system first gets a general declarative query from the user. The declarative query is formulated by navigating the visual components provided in the GUI interface. The system then collects the information from the GUI and formulates the query to be fed into the next process to construct a source-dependent query. The process of query transformation is encapsulated in the mediator layer in mediator-based architecture. Query transformation is required to construct a meaningful query based on the declarative query from the GUI layer. The meaningful query is referred to as fully parameterized query that contains dependent information on data source to generate a detailed execution plan. The objectives of the query execution layer are to facilitate the physical execution of query against the data source and handling the communication with the remote database. Its responsibilities include getting the data and responses from different database sources.

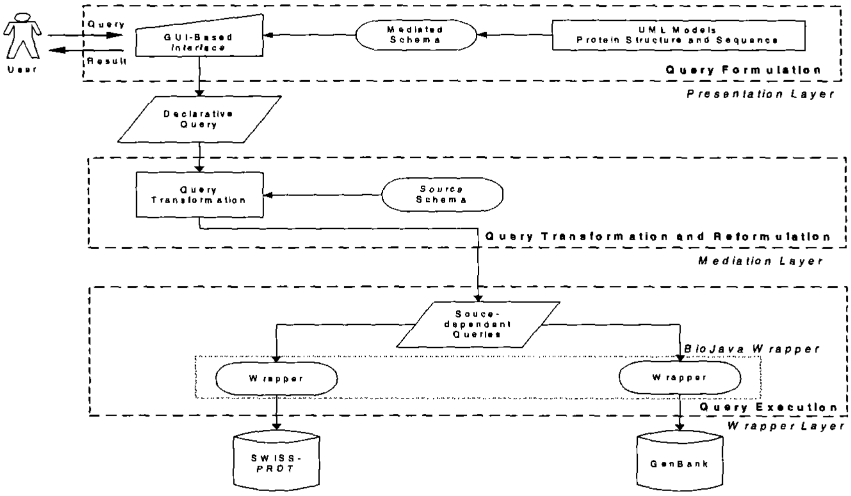

Fig. 1. Mediator-based Architecture

3 Implementation and Results

In the implementation of the mediator-based architecture in accessing Swiss-Prot and GenBank, Java's J2EE framework is adopted. The prototype is implemented as Java servlets that are executed in J2EE web application server. The servlets are used to generate GUI forms from the global schema and also to perform mediation on the queries submitted by the user. The global schema and local schemas for each individual databank are stored in a relational data model (MySQL). Each data source is implemented as a single relation in the global schema and the wrapper provides an interface that can accept queries on the data source relation; process the queries against the actual data source and return the required result. The web data sources integrated into the prototype are Swiss-Prot [3], a protein sequence bank and GenBank, a popular nucleotide sequence database. The query form consists of the source dependent fields that are relevant to the desired database. However, the user does not need to specify the field name for the searched field in the individual database as the fields in the query form are rewritten in terms of source fields during the query execution in the wrapper [4]. In order to analyze the performance of the tool, a search for organism field containing "`human coronavirus`" in Swiss-Prot is performed in each integration method to compare the query response time. Fig. 2(a) shows the response time of mediator against other integration approaches.

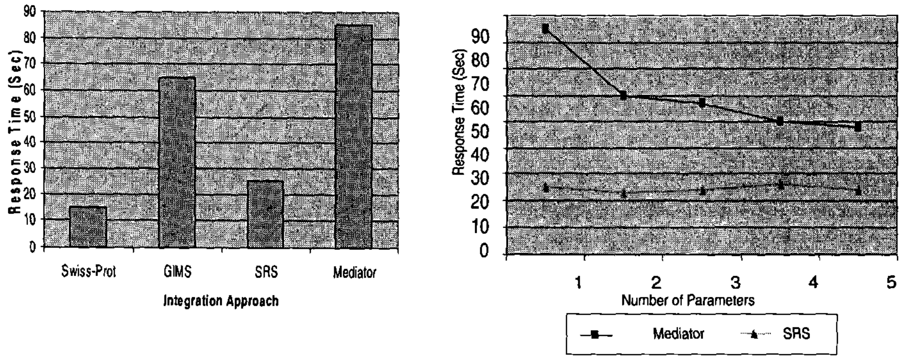

Fig. 2. Response Time for Querying Organism "`Human Coronavirus`" from Swiss-Prot for various integration methods; Comparison of Response Time for mediator and SRS for searching with various no of parameters

The results shown that GIM [5] and SRS have faster response time compared to mediator-based approach. The response time is slower in the mediator-based because of the delay from query translation, the process of parsing and filtering the results in the wrapper components and the network propagation time for submitting the query and getting back the result. However, as observed in Fig. 2(b), the mediator-based approach shows that the response time for querying Swiss-Prot with the less number of parameter consumes more execution time than a complicated query that has more parameters. This is because less result is returned by the data source with the specification of more search criteria. As for SRS, the response time required to generate the result does not depend on the number of parameters.

Although the mediator-based approach is slower in terms of query response time, the result generated by the mediator-based approach has high accuracy rate and the return results are also consistently up-to-date compared to other methods. This is because the mediator is querying the data sources directly and the updates of the data source are done autonomously by the owner of the data source.

4 Conclusion

In conclusion, the contribution of this work is to demonstrate the web-based bioinformatics application that utilizes different stand-alone bioinformatics database, implemented using mediator-based integration approach. This work contributes to the effort of integrating various types of biological databases and further improvement in query execution plan by engaging query optimization and dynamic execution plan generator. As for improving the response time of the mediator approach, the future work is to parallelize the result parsing in the wrapper module. The parallelism of parser could bring tremendous improvement response time especially in the situation where the query result is large.

References

1. The Mediagrid Project, A Mediation Framework for a Transparent Access to Biological Data Sources, Proceedings of the ECCB 2003 Conference Poster Session, Paris (2003).
2. Peter D. Karp. A Strategy for Database Interoperation, J Comput Biol 2(4): 573-86 (1995).
3. Magrane M., Apweiler R. Organisation and standardisation of information in SWISS-PROT and TrEMBL, Data Science Journal 1(1): 13-18. (2002).
4. Chia-Hui Changy, Harianto Siekz, Jianss-Jyh Luz, Jen-Jie Chiou, Chun-Nan Hsuz. Reconfigurable Web Wrapper Agents for Web Information Integration, Proceedings of Information Integration on the Web, Mexico (2003).
5. Mike Cornell, Norman W. Paton, Shengli Wu, Carole A. Goble, Crispin J. Miller, Paul Kirby. GIMS – A Data Warehouse for Storage and Analysis of Genome Sequence and Functional Data, 2nd IEEE International Symposium on Bioinformatics and Bioengineering, Maryland. (2001).