Bioinformatics Data Distribution and Integration *via* Web Services and XML

Xiao Li and Yizheng Zhang*
College of Life Science, Sichuan University/Sichuan Key Laboratory of Molecular Biology and Biotechnology, Chengdu 610064, China.

It is widely recognized that exchange, distribution, and integration of biological data are the keys to improve bioinformatics and genome biology in post-genomic era. However, the problem of exchanging and integrating biological data is not solved satisfactorily. The eXtensible Markup Language (XML) is rapidly spreading as an emerging standard for structuring documents to exchange and integrate data on the World Wide Web (WWW). Web service is the next generation of WWW and is founded upon the open standards of W3C (World Wide Web Consortium) and IETF (Internet Engineering Task Force). This paper presents XML and Web Services technologies and their use for an appropriate solution to the problem of bioinformatics data exchange and integration .

Key words: biological data integration, eXtensible Markup Language (XML), web services, eXtensible Stylesheet Language (XSL)

Introduction

Recently, more and more genomes have been sequenced and annotated, and the data of proteins and gene interactions are accumulating. Biological data are mostly digital and stored in a wide variety of formats in heterogeneous systems. Biological data exist all over the world as various web services, which provide biologists with much useful information. However, when users actually make use of them, they need to access to web services (databases) one by one. If they want to compare many different kinds of data, they need to do cumbersome task. Actually, a large part of the work of biologists today consists in distributing local data, querying multiple remote heterogeneous data source, and integrating retrieved data manually. Therefore, distributing and integrating biological data is a very important task and has been recognized as a key component of today's genome biology research.

Many communities have devoted to a large amount of work on the exchange and integration of biological data (*1*). However, the whole problem of data integration is not solved satisfactorily. The difficulties in dealing with the bioinformatics data exchange and integration come from the following technical issues:

1) The volume of biological data grows at an exponential rate.

2) Data are disseminated in a myriad of different databases and managed by different DataBase Management System (DBMS).

3) Biological data from different sources have heterogeneous formats (*2*).

4) The platforms or systems for distributing data are different, and the interaction and independence is lacked among these platforms or systems.

5) Hypertext Markup Language (HTML), as a language used widely for database browsing, data publishing, gathering, submission and analysis, is not suitable for extracting and integrating data.

Data integration consists in wrapping data sources and either loading retrieved data into a data warehouse or returning it to the user. Nowadays, database federation is a main technology for solving data integration problem (*3*). Database federation offers the promise of a unified view of these disparate data and detailed query through a single easy-to-use interface available *via* the World Wide Web. There are two approaches for implementing database federation: concrete and virtual (or loose) federation. Some systems such as Entrez, SRS, DiscoveryLink, and DBGET (*4*), have been designed for the specific integration

*** Corresponding author.**
E-mail: yizzhang@scu.edu.cn

of biomolecular data. However, there are some shortcomings by using database federation technology for integrating data. First, the retrieved data by concrete federation are not always the latest and greatest. Second, because the retrieved data by virtual federation are web pages (HTML format), it is an arduous task to parse the result documents. Third, a client of virtual federation must be tied to the upstream web service directly. Changes of the web service interface make it difficult to maintain the federated database.

Web Services, a kind of service-oriented architecture, have been used worldwide to exchange and integrate data in e-commerce. However, few were introduced about the use in bioinformatics. Actually, the shortcomings described above are addressed in part by Web Services and XML (eXtensible Markup Language) technologies.

The flat file format (FF format) is the popular data format for distributing nucleotides data and other biological data. However, it is very difficult to parse the FF format for extracting the interesting information. XML has some features for overcoming the disadvantage of FF format. XML provided a generic way to represent structured and typed data, which makes it easy to write a script for parsing an XML document.

Web Services

A web service is a unit of business logic, located somewhere on the Internet, which is accessible through standard-based Internet protocols such as HTTP or Simple Message Transfer Protocol (SMTP). The core of today's Web Services technology is made up by SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) and UDDI (Universal, Description, and Discovery Integration; Figure 1).

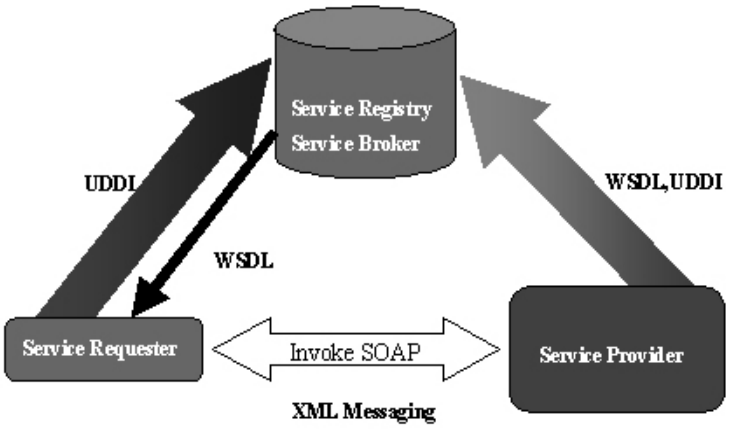

Fig. 1 The architecture of web services describes the relationship and working principle of these pieces (SOAP, WSDL, and UDDI).

Web Services have some features for solving the problems of information exchange, data integration and distributed application:

1) Web service is XML-based. All of web services use XML as the data representation layer.

2) Web service is loosely coupled. A client of a web service is not tied to the web service directly.

3) Web service is coarse-grained. The technology provides a natural way of defining coarse-grained services that access the right amount of business logic.

4) Web service supports Remote Procedure Calls (RPCs). It allows clients to invoke procedures, functions, and methods on remote object using XML-based protocol.

5) Web service supports a transparent exchange of documents to facilitate data and documents integration.

eXtensible Markup Language (XML)

XML (http://www.w3c.org/XML/; ref. *5*) has been developed to overcome the limitations of HTML. XML is a markup language that specifies neither the tag set nor the grammar for that language, and is a meta language used to define other language. XML allows one to define his own markup language, which consists of his own tags. Furthermore, the meaning of tags is essentially different between XML and HTML. Unlike HTML, the tags defined in XML indicate the

semantics of a document rather than display a document. The set of tags and grammar, or schema, for an XML language, describing admissible combinations of tags in a document, can be formalized and enforced by standard parsing tools. It is clear that XML has some interesting features addressing the problems of data exchange and integration introduced above. For one thing, XML provides an open framework for standard specifications in managing bioinformatics data. That is an important point because bioinformatics lacks standardization. For another, XML makes it very advantageous to exchange data among data sources if all of them adopt the same standardization in XML documents. Thirdly, it is convenient to extract interesting information from XML documents, which helps to data integration.

In life science, many commercial and academic communities are now adopting XML as a standard for their genome biology data management (http://scbi.scu.edu.cn/XML/).

Results and Discussion

Using Web Services, SOAP and XML, a simple web service client was constructed to retrieve XML nucleotides data. The web service provider is XML Central of DDBJ. The eXtensible Stylesheet Language (XSL) was used to transform XML data into HTML form. The web service client is available at http://scbi.scu.edu.cn/webservices/.

The new and exciting Web Services and XML technologies are drastically changing the way people conduct business, vastly altering the competitive landscape of information technology industries, and significantly improving enterprise efficiency. The influences of the Web Services and XML technologies are becoming increasingly visible, and their momentum will greatly become more significant over the next several years.

Under the Web Services, a single application can tap into the services of millions of applications scattered throughout the Internet. The potential of this is enormous. The promise of web services is to enable a distributed environment in which any number of applications, or application components, can interoperate seamlessly among and between organizations in a platform-neutral, language-neutral fashion. The interoperation brings heterogeneity to the world of distributed computing. Nowadays, Web Services are revolutionizing the Internet, and are used in e-commerce worldwide.

As a promising standard, Web Services and XML technologies will undoubtedly be a chance to exchange and integrate biological data on the bioinformatics community. It supports an integrated view for managing remote or local heterogeneous biological data sources with advanced data accessing. We propose to replace the flat file format by XML and to distribute data by web services, which would provide the programmers a uniform access to the biological data and a standardized interface for interoperable applications.

In this paper, the author examined only a simple application for integrating biological data using web service client and XML technologies, but an architecture of web services is obviously more complex than the application that makes the web service call directly (Figure 2).

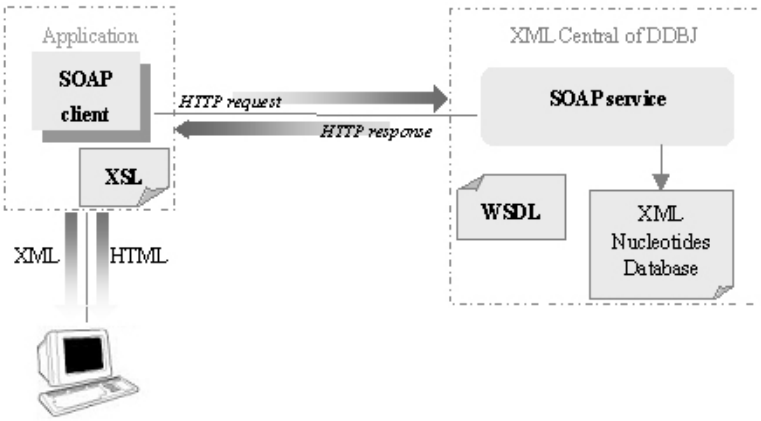

Fig. 2 Discrete components and interactions in the web services architecture.

Methods

Our service client is implemented by the hardware of Pentium IV 1.8 G with 512 Mb memory and 18 G hard disk. The set of software is RedHat Linux 8.0, Jakarta Tomcat 4.1.24, Java (j2sdk1.4.1_02), Apache Axis 1.1, and Apache 1.3.27.

Apache Axis (Apache eXtensible Interaction System, http://ws.apache.org/axis/) is an open-source implementation of the SOAP submission to W3C. It is essentially a SOAP engine—a framework for constructing SOAP processors such as clients, servers, gateways, *etc.* The current version of Axis is written in Java.

Retrieving XML nucleotides documents

XML Central of DDBJ is actually a SOAP server and web service. The server provides various servers, including BLAST, FASTA, ClustalW, and so on (*6*). The GetEntry is one of the appropriate servers for getting entries by specifying accession number. Using SOAP message, our service sends a request into the GetEntry server to call the method getXML_DDBJEntry. The information of the GetEntry server is obtained from the WSDL document (http://xml.nig.ac.jp/wsdl/GetEntry.wsdl; Figures 3 and 4).

```
Import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;
import org.apache.axis.utils.Options;
import javax.xml.rpc.ParameterMode;
import javax.xml.namespace.QName;
import java.net.URL;

public class DdbjXML {
      public static void main ( String args[] ) throws Exception {
            String wsdlURL = "http://xml.nig.ac.jp/wsdl/GetEntry.wsdl";
            String nmspace = "http://www.themindelectric.com/wsdl/GetEntry/";
            String srvname = "GetEntry";
            String fncname = "getXML_DDBJEntry";
            String query    =   "AB000001";
            QName serviceQN = new QName(nmspace, srvname);
            QName portQN       = new QName(nmspace, srvname);
            Service service = new Service(new URL(wsdlURL), serviceQN);
            Call call = (Call)service.createCall(portQN, fncname);
            String result = (String)call.invoke(new Object[]{query});
            System.out.println(result);
      }
}
```

Fig. 3 A simple Java program for retrieving XML nucleotides document by specifying accession number AB000001.

Fig. 4 The result, an XML document of accession number AB000001 nucleotides sequence, is displayed in web browse.

Transforming XML documents into HTML form

XSL transforms and translates XML data from one XML format into another. Consider, for example, that the same XML document may need to be displayed in HTML, PDF, and Postscript form respectively. Without XSL, the XML document would have to be manually duplicated, and then converted into each of these three formats. Instead, XSL provides a mechanism of defining stylesheets to accomplish these types of tasks. Rather than having to change the data because of a different representation, XSL provides a complete separation of data, or content, and presentation.

To transform XML documents into HTML form, an XSL file named "ddbjhtml.xsl" was designed and is now available at http://scbi.scu.edu.cn/webservices/ddbjhtml.xsl.

More information about XSL can be obtained from W3C website (http://www.w3c.org/Style/XSL/).

Extracting information from XML

It is a key advantage of XML to bioinformatics data integration that XML documents are parsed easily. SAX (the Simple API of XML), DOM (the Document Object Model) and JAXP (Java API for XML parsing) are three major APIs (Application Programming Interface) to parse XML documents. By parsing XML documents, data extracted from various XML documents can be dumped into a database, or integrated into an XML document, which is the major task of bioinformatics data integration (Figure 5). Moreover, application interoperability, standardizing the interfaces between stand-alone applications such as the data generated from one application flow as the input to another application, will be convenient.

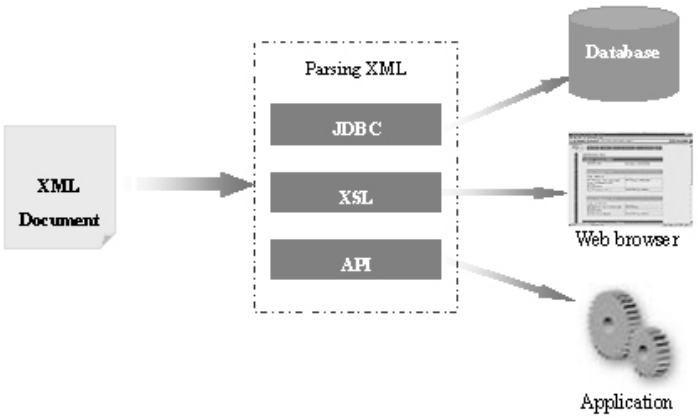

Fig. 5 Java DataBase Connectivity (JDBC) is the middleware that allows Java programs to access data from a relational database. JDBC provides a standard SQL database access interface.

Ackonwledgements

We thank Mr. Yang Shiming and Mr. Zhang Guojun of the UESTC (University of Electronic Science and Technology of China) for their support in the development of web service client.

References

1. Stein, L. 2003. Integrating biological databases. *Nat. Rev. Genet.* 4: 337-345.
2. Stein, L. 2002. Creating a bioinformatics nation. *Nature* 417: 119-120.
3. Haas, L.M., *et al.* 2002. Data integration through database federation. *IBM Systems Journal* 41: 578-596.
4. Fujibuchi, W., *et al.* 1998. DBGET/LinkDB: an integrated database retrieval system. *Pac. Symp. Biocomput.*: 683-694.
5. Achard, F., *et al.* 2001. XML, Bioinformatics and Data integration. *Bioinformatics* 17: 115-125.
6. Sugawara, H. and Miyazaki, S. 2003. Biological SOAP servers and web services provided by the public sequence data bank. *Nucleic Acids Res.* 31: 3836-3839.