Databases

Design of a description language for generating wrapper to collect biological data

Myungeun Lim*, Myunggeun Chung, Myungnam Bae and Sunhee Park
Bioinformatics Research Team, Electronics and Telecommunications Research Institute, 161, Gajeong-dong, Daejeon, Republic of Korea.

ABSTRACT

Summary: The biological data are scattered in various areas with various formats and they are changing continuously. Therefore, data integration becomes an important issue to provide researchers with a dynamic access of data. In the data integration process, the method of extracting heterogeneous data dynamically from the data source is an essential part. Data extraction method using wrapper can provide flexibility and extensibility to an integration system.

Contact: mellim@etri.re.kr

1 INTRODUCTION

The specific needs of biologists have led to produce numerous databases that provide various data with heterogeneous formats and they are changing continuously. Biological data integration is an important topic because data are scattered in various sources, preventing researchers from efficiently obtaining the information required to perform their work (Lacroix and Crichlow, 2003).

A lot of research on data integration in bioinformatics has been done. Well-known systems such as SRS (SRS documentation, http://srs.ebi.ac.uk/), DiscoveryLink (Hass et al., 2001), TAMBIS (Stevens et al., 2000), and BioKliesli (Davison et al., 2001) are successful results of this research. All the same, more systems are developing because requirements of data in biological research are quite different. Our assumption in data integration is that the system can provide various views reflecting real-time data from heterogeneous data sources, can adapt well to the change of data source and support easy-to-use interfaces for biological users.

We premise that data integration system goes through three steps—analyze the query, collect data and organize the results. In query analyzing step, the query given by the user is parsed and divided into local queries or terms. Then the data are extracted from the source in the collection step. The results are organized in the integration step. The topic of this paper is primarily focused on the second stage, collecting data, which makes it possible to adapt the data source and extract automatically for a given query term.

To implement dynamic data translation from heterogeneous databases, we designed a wrapper-based data extraction system. By using wrappers, data scattered in various areas can be collected and viewed by the users. To generate wrapper dynamically, we define a scripting language, wrapper description language (WDL). Using WDL, users can design wrapper to operate as they intended to. As the specification of WDL is simple, biological researcher who has used bioPerl or bioJava can use it well for their purpose.

2 SYSTEMS AND METHODS

Wrapper is a module or a program that extracts data automatically from target data source in a given format to a given query (Kushmerik et al., 1997). Wrapper is useful when the format and schema of the data source are changing frequently. Our extraction system uses wrappers to collect biological data. Wrappers are generated automatically from wrapper generating module by interpreting WDL document. We first introduce WDL and then explain how the extraction system works.

2.1 Wrapper description language

WDL is a kind of scripting language which is used to describe a type, structure and extraction rule of a data element extracted by a wrapper. In general, wrappers are generated automatically or manually by interpreting some clues written by the user, such as WDL document.

To define the function of the wrapper, WDL has to be able to describe basic information—type and definition of data, a rule of extracting data from the source, method of querying to source and constraint of data being satisfied. All of these are included in the WDL specifications.

The type and hierarchy of the data are defined in data definition part. The term ‘ELEMENT’ indicates the data element. The type and name of the element follow next. Most primitive types used in XML are supported. The nesting is allowed using ‘{’ symbol among elements to express the hierarchical relation of data. Variable can be defined using the term ‘VAR’ to deal with temporary data used in the document. The term ‘LINK’ can be used to define the cross-reference of databases. As biological databases have many relationships among them, a method to express these relationships is needed. In the ‘TARGET’ field followed by ‘LINK’, the name of the reference database and its primary field has to be described. ‘LINK’ is presented as an attribute of its parent in the result file, as showed in Figure 1b. The form of the result document is determined in data definition.

Figure 1a shows an example document described by WDL. It is a partial example of NCBI nucleotide wrapper; (ii) is the example of data definition part.

The rule of extracting data from target data source is defined in extraction rule part, as shown in (iii) of Figure 1a. Rule is expressed by Regular Expression which is used in bioPerl or bioJava to provide convenience in expressing rules to the biological user. The matching target of the rule can be variable or other elements. User-defined function can also be used in describing rules. The user can import user-defined function coded by Java (i) of Figure 1a.

As WDL aims at simplicity in the expression, it is necessary to provide a more complex way of data manipulation. User-defined function can be an alternative of providing extensibility to describe the rule.

The operation definition part describes how the wrapper accesses the data source and collects the content. Since the target source can be varied (e.g. web, flat file and DBMS), the system has to provide the user a way of describing how to access the database and how to query to them. Basically, we define two

*To whom correspondence should be addressed.
types of querying method—(1) query by ‘KEY’ and (2) query by ‘SEARCH’. If the user knows the key field, which is a unique field that identifies unique data, he can query by a special key and get just one item. Otherwise, the user has to query by terms to a certain field defined in ‘PARAMETER’ section. The user can define fields as parameters that can be a target of query. In the ‘QUERY’ section, the real query is assigned to be transmitted to the target source. If the target is on the web, query might be GET or POST type URL. Or else if the target is in DBMS, query might be SQL or other DBMS query.

In the constraint part, the restriction to the result data is described. When the user wants to restrict a specific element in the results, he can do it by describing ‘CONSTRAINT’. Since it is a powerful querying method to manipulate XML document, constraint is written by simplified XQuery. Each element or attribute in the document can be restricted to a certain condition. Constraint is applied after the result XML document is built.

2.2 Extraction system

We implemented wrapper-based biological data extraction system. The system has two components—wrapper generation module and data extraction module. When the user writes out the WDL document, WDL parser parses it and generates the wrapper as its function is described in the document. XML Schema is also generated which contains the schema of the results to check whether the extracted result is valid or not. If the user queries to the wrapper, it analyzes the query term and query type and executes it to the target database. Then, the wrapper organizes the result document as an XML format. If there is a constraint, wrapper interprets query and applies it to the result document. XQuery parser is embedded in the wrapper. We use Saxon v7.0 as XQuery parser.

2.3 Experiment

In the experiment, we made five wrappers of some web-based open databases—NCBI nucleotide, Gene, Uniprot, GO and SwissProt. From the genebank ID, given from the experimental data provided by LiveChem®, we traced reference information and ontology data using these wrappers. The sample result of NCBI nucleotide is shown in Figure 1b. As a result, we saw that the data in different sources are dynamically related using several wrappers.

3 CONCLUSIONS

We have suggested a description language which is used to describe wrappers function and biological data collection system that can create wrappers dynamically and extract data using them. It might be helpful to the biological researcher who wants to get data from various data sources and wants to analyze them. Our final purpose is to construct a biological data integration system and this extraction system can be used as a basic module in developing integration system. We are developing mediator–wrapper based integration system that can be operated in web service environment. The user might access the service without any customization of the integration system.

ACKNOWLEDGEMENTS

We are appreciative of the support from the Electronics and Telecommunication Research Institute (ETRI). This work was supported in part by Ministry of Information and Communication (MIC) grants (A1–03–00047–00) and Ministry of Health and Welfare (MOHW) grants (0412–MI01–0416–0002).

Conflict of Interest: none declared.

REFERENCES


