Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Application of XML Database Technology to Biological Pathway Datasets

Keyuan Jiang and Christopher Nash

Abstract— **The study of biological systems has accumulated a significant amount of biological pathway data, which is evident through the continued growth in both the number of databases and amount of data available. The development of BioPAX standard leads to the increased availability of biological pathway datasets through the use of a special XML format, but the lack of standard storage mechanism makes the querying and aggregation of BioPAX compliant data challenging. To address this shortcoming, we have developed a storage mechanism leveraging the existing XML technologies: the XML database and XQuery. The goal of our project is to provide a generic and centralized store with efficient queries for the needs of biomedical research. A SOAP-based Web service and direct HTTP request methods have also developed to facilitate public consumption of the datasets online.**

I. Introduction

BIOLOGICAL pathways represent our current understanding of biological system at the cellular level and play an important role in the study of how our life is developed and sustained as well as developing new pharmaceutical treatments to abnormalities. The National Institutes of Health has set a priority for the research in pathway discovery in its roadmap [1]. Like many other molecular biology databases, the biological pathway databases have not only grown in numbers, but also in the amount of data available. Today there exist more than 200 publicly accessible biological pathway databases [2], each with its own data schema, storage mechanism and access methods. The development of the BioPAX standard aims at creating a standard format to assist in the exchange of biological pathway datasets [3, 4]. Since the release of BioPAX Ontology Level 1 in 2005, a number of data hosts such as BioCyc, Reactome, KEGG and PathCase have published their pathway datasets based upon this new standard format. While this is a big leap forward towards the goals of the BioPAX effort, one major obstacle must be overcome. Although the data are in a standard format, the datasets provided are neither readily accessible nor easily processable by computer programs due to the different methods in which the datasets are organized and lack of a standard way computer programs to access the datasets, posing a challenge to the aggregation of data from different sources. For example, BioCyc places all the pathways for a particular organism in a single file, whereas KEGG generates a single file for each individual pathway. The flatfile format also contributes to the difficulty in processing the BioPAX compliant datasets. In addition, each provider makes their datasets available through different methods, which requires manual intervention when retrieving.

One of the motivations of the BioPAX effort was to provide a public accessible repository of biological pathway datasets in a standard format [5], along with which is an efficient way of storing the datasets. Without such an efficient storage method, processing datasets utilizing the BioPAX format with existing software tools can be quite challenging. First, many research tasks require focusing on a small amount of relevant data, instead of the entire collection of pathways. One may apply some filtering techniques to the datasets as that used in the BioDash demonstration, where semantic lenses are used [6, 7]. This solution works well if the datasets are small. A better solution to this problem would be to use an efficient query mechanism against a data store. Secondly, if a large raw BioPAX dataset is processed as in the case of datasets provided by BioCyc, where a single data file contains all the pathways of a particular organism and can be quite large in size (e.g., the size of *homo sapiens* pathway data file is about 20 MB), few existing XML parsers can efficiently process such a large amount data. The in-memory DOM method will require a significant amount of system resources to process large datasets, whereas the event-driven SAX approach will lead to a poor performance due primarily to the constant reading from the hard disk. Furthermore, for the datasets enclosing a single pathway within a data file, data can be processed efficiently by existing XML parsers, but this approach results in difficulty associating multiple pathways from different data files. More importantly, there has been a need by the authors of the BioPAX standard to store and process the BioPAX format data using existing and mature XML technologies [8]. Strömbäck has assessed the feasibility of using XML technologies to store pathway datasets in SBML and PSI-MI formats was assessed [9].

Although the BioPAX compliant datasets use the OWL (Web Ontology Language [10]) format, arguably it was not the initial intention of the BioPAX Working Group to implement the Semantic Web in BioPAX [3, 11]. While it is ambitious goal to achieve, many applications of the BioPAX datasets may not require this full implementation of the Semantic Web. On the other hand, the technology for this implementation has yet to mature [11, 12]. RDF triple stores

Manuscript received April 23, 2006. This work was supported in part by an eScience Application Grant from Microsoft Research (Award #: 12703).

K. Jiang was with the Department of Computer Information Technology, Purdue University Calument, Hammond, IN 46323 USA. (phone: 219-989-2035; fax: 219-989-3187; e-mail: jiang@calumet.purdue.edu).

C. Nash was with Purdue University Calumet, Hammond, IN 46323 USA. (e-mail: chris@inash.net).

have been considered for the OWL-based datasets. The Pathway Knowledge Base (PKB) project at Stanford University [13, 14] uses the Oracle Spatial Network Data Model to store the BioPAX compliant datasets. Its approach of using RDF triple stores requires decomposing each pathway into a number of subject-predicate-object triples, generating a large number of records in the relational store. The PKB authors have acknowledged the slow performance of the system. The tests conducted by authors of this article indicated that regardless of the complexity of the queries they usually took about 20 seconds. This can be partially attributed to the fact that a significant number of queries is needed to reconstruct a pathway from triples.

To address the above issues, we have designed a storage system based upon several current XML technologies, and have applied them in the Gateway to Biological Pathways Web service. Our data store, implemented with an XML database, maintains the biological pathway datasets in the native BioPAX format in an XML datatype column, and supports efficient and sophisticated queries through the use of XQuery. To facilitate the access to the pathway datasets, we have implemented a collection of methods via a SOAP-based Web service as well as Web pages using HTTP requests.

II. Design and Implementation

A. The Data Store

We have observed that almost all of the applications proposed by the BioPAX Working Group require a substantial number of queries against the datasets. Although the decomposition of BioPAX compliant datasets into a collection of RDF triples results in the closest presentation for the Semantic Web, it suffers in query performance when a pathway object made up of multiple triples needs to be retrieved [13]. On the other hand, retaining the raw data without decomposition will also lead to poor performance when a specific component of a particular pathway is sought that is nested deep inside the pathway. To achieve a relatively good performance for both scenarios and ease the processing of retrieved data, the granularity of the data items to be stored in the database should lie between the RDF triples and the entire collection of the raw datasets. In addition, the data items should be as biologically meaningful and identifiable as possible.

Typically, a *pathway* object in BioPAX is composed of a number of other objects such as *pathway steps* and *physical entity articipants*. Using the *pathway* object as the storage unit in a database is not appropriate because (1) the data redundancy can occur since a single molecule may be involved in multiple pathways, and (2) queries can be inefficient if an object to be sought is embedded deeply in the tree hierarchy of the XML-based BioPAX data (a user or client program may be interested in objects made up of a particular pathway). To address the storage issue, we have chosen a set of objects of different types defined in the BioPAX Ontology as storage units. They include *pathway*, *protein*, *publicationXref*, *biochemicalReaction*, and so forth. Internally, each different type of object is a different type of XML document. That is, in BioPAX format, a *protein* object has a different XML presentation than a *publicationXref* object.

Our choice of data item granularity leads to multiple different data types to be dealt with. If we were to use a single database data type for each type of BioPAX entity, this will create a rigid design which could be quite costly if not completely broken, when any changes made in the BioPAX standard or the future levels of the BioPAX standard need to be implemented. The new XML datatype available in XML database was chosen to accommodate different data types in a single database column. This approach not only provides a generic way to store the data, but it also simplifies the query methods for a particular BioPAX object or across multiple BioPAX objects. To determine the type information about a particular object, the client software may access the BioPAX ontologies available at www.biopax.org [15, 16].

The BioPAX compliant datasets are in a specific XML format and many components (objects) are embedded in the hierarchical fashion. To store various types of objects in our data store, the XML-based hierarchy was flattened by rearranging the BioPAX format data such that embedded object definitions were replaced with references and moved out of the composing object. When the composing object is retrieved, all its components are retrieved first and are placed in the same set. Although extra queries are needed to retrieve the components, the number of the components is significantly smaller than that of RDF triples, hence reducing the number of the queries needed.

```
SELECT objectID, BioPaxObject
FROM tblXML
WHERE ObjectID = @objectID;
```

Figure 1. The SQL SELECT statement retrieves a particular BioPAXObject with its ID equating to the value in variable @objectID. All BioPAX objects are stored in an XML datatype

```
SELECT objectID, T.C.value('.',
'varchar(500)') AS synonyms
FROM tblxml TX
cross apply TX.biopaxobject.nodes('
declare namespace
ns="http://www.biopax.org/release/biopax-
level1.owl#"; /child::*/ns:SYNONYMS') as T(C)
WHERE T.C.value('.', 'varchar(500)') = @name;
```

Figure 2. The XQuery code snippet for retrieving BioPAX objects with the synonym value equating to the value of variable @name.

TABLE 1
THE GATEWAY WEB SERVICE AND HTTP REQUEST METHODS

Web Service Method	HTTP Method	Description
getObjectByID	getObjectByID.aspx	Retrieves a BioPAX object (top-level description) of a given ID
getDetailedObjectByID	getDetailedObjectByID.aspx	Retrieves a BioPAX object with all of its components of a given ID
listObjectsByNameOrSynonym	listObjectsByNameOrSynonym.aspx	Retrieves a list of BioPAX objects that match either a given name or synonym
listObjectsByType	listObjectsByType.aspx	Retrieves a list of BioPAX objects of a given type
listObjectsNameAndIDByType	listObjectsNameAndIDByType.aspx	Retrieves a list BioPAX objects of a given type (names and IDs only)
listObjectsReferencingID	listObjectsReferencingID.aspx	Retrieves a list of BioPAX objects which contain a specific object
listObjectTypes	listObjectTypes.aspx	Retrieves a list of available BioPAX object types
listOrganisms	listOrganisms.aspx	Retrieves a list of available organisms
listSourceDBServers	listSourceDBServers.aspx	Retrieves a list of available source DB servers

For the implementation, we opted to store various types of pathway objects in a single column utilizing the XML datatype which can accommodate the variations. Microsoft SQL Server 2005 [17] was used to implement the data storage in order to meet our storage and query requirements. To improve query performance, key pieces of information such as IDs are extracted from the xml documents and added to auxiliary columns in the same record. Basic searches are achieved using standard SQL statements as shown in Figure 1, while sophisticated queries such as searching an object with a particular synonym require the use of XQuery statements as illustrated in Figure 2. Due to our "flattened" presentation of BioPAX data, a more efficient mechanism had to be used to improve the query performance when retrieving all the components involved in a given pathway. In addition to support of SQL and XQuery, Microsoft SQL Server 2005 also supports stored procedures written in CLR (Common Language Runtime) languages. Leveraging this feature, we developed a few CLR-based stored procedures in C# that allows us to efficiently retrieve all the components of a particular object.

B. The Retrieval Methods

To facilitate retrieving pathway data in the machine-processable manner, we implemented two access mechanisms for external consumption of the datasets: a SOAP-based Web service and direct HTTP request (both GET and POST) methods. The SOAP-based Web service provides a programmable interface for querying pathway objects, whereas HTTP request methods allow an application to access pathway objects via URLs. Both the Web service and HTTP request methods are available at http://jlab.calumet.purdue.edu/theGateway. Listed in Table 1 are Web service methods and their corresponding HTTP request Web pages. Interested readers can find online a collection of comprehensive documentation about these methods.

TABLE 2
AN INCOMPLETE LIST OF BIOPAX OBJECTS IN THE DATA STORE

Object	Number
BiochemicalReactions	2,721
Catalyses	4,223
ChemicalStructures	2,274
Complexes	607
Pathways	594
pathwaySteps	3,867
physicalEntityParticipants	8,779
Proteins	3,914
smallMolecules	3,627
Transports	140
unificationXrefs	22,082
Total # of objects	64,158

III. DISSCUSSIONS

Our approach of using the XML datatype to store biological pathway datasets has several advantages. First, it simplifies the database model by requiring a single XML datatype field to store different types of pathway components in a single column. When new changes, features and levels of BioPAX standard are introduced in the future, the same storage will seamlessly accommodate the new requirements. Secondly, the XML database supports many common database operations on the XML datatype, including insertion and update in addition to queries. Updates on the part of a pathway component are also supported, which allows a user to create and modify pathway data in the BioPAX format, eliminating the need to convert back and forth between the native data host format and the BioPAX format.

Currently, pathways of three organisms (*homo sapiens*, *E. coli* strain K-12, and *E. coli* strand O157:H7) from BioCyc Common Databases are available in the Gateway data store. There are 64,158 BioPAX objects in the store with 594 *pathways*, 3,867 *pathwaySteps*, and 8,779 *physicalEntityParticipants*, among others (refer to Table 2 for an incomplete list of BioPAX objects in our data store). Datasets from other hosts/providers are being added.

Authors envision that a number of bioinformatics software applications can be developed using the Gateway Web service. The clients of the Gateway services do not need to store a large amount data locally. Any pathway objects needed can be retrieved programmatically from the Gateway. The client applications can leverage existing XML technologies to efficiently process small sets of relevant pathway data.

REFERENCES

[1] The National Institutes of Health: NIH Roadmap for Biomedical Research - Building Blocks, Biological Pathways, and Networks: [http://nihroadmap.nih.gov/buildingblocks/]

[2] The Pathway Resource List: [http://cbio.mskcc.org/prl/]

[3] J.S. Luciano: "PAX of Mind for Pathway Researchers." *Drug Discovery Today*, 10, 937-942, 2005

[4] The BioPAX Workgroup: BioPAX – Biological Pathways Exchange Language, Level 1, Version 1.4 Documentation, 2005 [http://www.biopax.org/release/biopax-level1-documentation.pdf]

[5] The BioPAX Workgroup: [http://www.biopax.org]

[6] E.K. Neumann and D. Quan: "Biodash: A Semantic Web Dashboard for Drug Development." *Proc. Pacific Symposium on Biocomputing*, 2006 [http://helix-web.stanford.edu/psb06/neumann.pdf]

[7] The BioDash Demonstration: [http://www.w3.org/2005/04/swls/BioDash/Demo]

[8] A. Ruttenberg: "Late 2005 Proposal for BioPAX Development/Response." 2005 [http://biopaxwiki.org/cgi-bin/moin.cgi/Late_2005_Proposal_for_BioPAX_Development/Response]

[9] L. Strömbäck: "Possibilities and Challenges Using XML Technology for Storage and Integration of Molecular Interactions." *16th International Workshop on Database and Expert Systems Applications* (DEXA'05), pp 575-579, 2005

[10] World Wide Web Consortium: Web Ontology Language [http://www.w3.org/2004/OWL/]

[11] A. Ruttenberg: "Experience Using OWL DL for the Exchange of Biological Pathway Information." Workshop on OWL Experiences and Directions, 2005 [http://www.mindswap.org/OWLWorkshop/sub37.pdf]

[12] M. Samwald: "Classes versus Individuals: Fundamental Design Issues for Ontologies on the Biomedical Semantic Web." Workshop on Foundations of Clinical Terminologies and Classifications (FCTC 2006), 2006 [http://www.imbi.uni-freiburg.de/medinf/fctc-2006/docs/sa_final.pdf]

[13] N. Kotecha, K.J. Bruck and W. Lu: "Pathway Knowledge Base: Integrating BioPAX Compliant Pathway Data." Submitted

[14] Pathway Knowledge Base: [http://pkb.stanford.edu]

[15] The BioPAX Ontology Level 1: [http://www.biopax.org/release/biopax-level1.owl]

[16] The BioPAX Ontology Level 2: [http://www.biopax.org/release/biopax-level2.owl]

[17] Microsoft Corp.: Microsoft SQL Server: [http://www.microsoft.com/sql/default.mspx]