World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 68-72, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.29.csea.2230

Designing ODA for Protein Databases

R. Udayakumar, K.P. Kaliyamurthie, Khanaa, and A.V. Allin Geo

School of Computing Science, Bharath University, Chennai-73, India

Abstract: Ontological thinking is coming to occupy a central role in bioinformatics. Most of the bioinformatics applications would benefit from comparing proteins based on their biological role instead of their sequence. As biological ontologies have inexorably grown larger, they are facing many issues; one of such issues is distributed architectures. This paper introduces issues involved in applying distributed architectures for protein databases using ontology approach. We are proposing ODA (Ontology-based Distributed Architecture) for Protein Databases that scales to very large ontologies. This ODA helps in maintaining backward compatibility and provides more reliability. We highlight some of the practical difficulties in developing ODA and suggest that the experiences with these architectures have implications for the development of ODA. Based on these we conclude that the proposed ODA is a helpful to support the replacement of older 80-column PDB files.

Key words: Bioinformatics · CORBA · Distributed Computing · Ontology · Protein Sequences

INTRODUCTION

CORBA and RMI are technologies have been used in many bioinformatics projects. CORBA, Common Object Request Broker Architecture is a standard that provides an intermediate object-oriented layer that handles access to the data [1]. For biological data analysis the scientific community has been using web-based application platforms. Biological data can be downloaded from institutes such as SWISS-PROT [2], EBI [3], NCBI [4], etc. But, these raw data sources usually come in different formats. So, software components need to be written to convert these different data formats into desired formats and import data into in-house database systems.

Based upon the functionality of the applications, sometimes it is necessary to build application specific indexed databases from the existing raw data sources. For example, the BLAST [5] program requires to run 'formatdb' on raw databank files like SWISS-PROT, EST etc. This returns a sorted binary index files that can be used by BLAST. These things show that they need additional programs or components for data format conversion of output. Additional programs require testing process and they may be tending towards errors, it requires complete software development process. Parsers convert raw data into a format that can be read by programs such as ClustalW [6]. Again this output will have to be parsed into another module to be processed into web ready formats, such as HTML, XML. Middle level process can be eliminated with the help of XQuery.

Comparing with different technologies, OODBMS [7] has the advantage of offering a very rich data model and well suited to biology. It provided many features like scalability, indexing, clustering, query optimization etc. Data definition and query languages have been standardized. OODBMS guarantees integrity, security, consistency and reliability. Only thing to use this it requires a high computer science background than XML. Another thing data returned from OODBMS should be parsed to show on the web. Weaknesses of XML overcome by combining the features of OODBMS and XML.

Queries on the database would then return XML formatted results that could be exchanged between users, databases or browsers. The use of XML would be efficient only if all databases share common DTDs. XQuery was designed as a general-purpose query language for XML. The processing of Xquery may be

Corresponding Author: R. Udayakumar, School of Computing Science, Bharath University, Chennai-73, India.

quite different; still the needs of domain are quite similar on the query-language level. In some ways this is analogous to the different uses of SQL, like processing, execution of queries.

Developing a CORBA environment is a heavy task that requires highly skilled computer scientists. It requires understanding of technologies like Interface Definition Language (IDL), protocols like Inter-ORB protocol etc. Java RMI (Remote Method Invocation) provides mechanism to invoke methods on Java objects, which can be resided in a local or a remote server [8]. CORBA based on two different programming languages, where as in RMI programming both the client and the server must be written in Java.

ODA: Ontology Based Distributed Architecture for Protein Databases: Here we are proposing the ODA model, which is used to extract information efficiently. This model is based on technologies like CORBA, JSP (Java Server Pages), EJB (Enterprise Java Beans), XML, XQuery etc. ODA consists of Remote databases, Internet Services, Web Services and Local XML databases. ODA architecture is shown in Fig 1. ODA scales to very large ontologies and it includes more reliable code. Main advantages of ODA are transfer between representations and helps in maintaining backward compatibility.

Web services allow the XML database system to access the various bioinformatics databases. Top most layer is remote database layer is an abstract layer for the existing modules, each module has a corresponding function call. This layer consists of bioinformatics operations like sequence alignments, motif search, sequence manipulations etc. Sequence alignments include

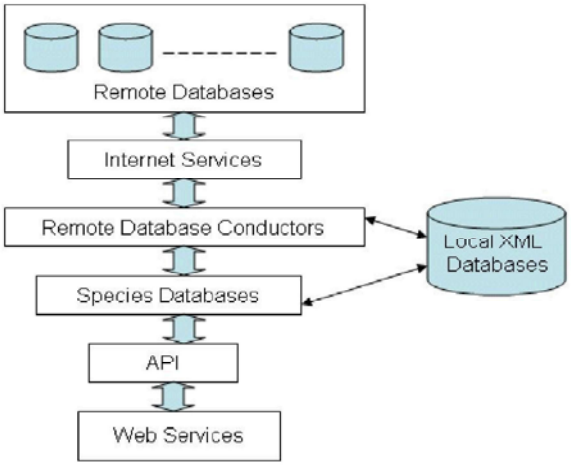

Fig 1: ODA Model

alignments between DNA sequences, protein sequences or Expressed Sequence Tags (ESTs). ODA provides a general framework for data extraction and integration for building ontology based biological data stores. Application Programming Interfaces are used to process XML templates for constructing XML bio-entities, XML bio-entity templates, XQuery based queries and extraction logics.

XQuery Coding for Protein Databases: The need for a query language for XML becomes obvious due to rapid increase in the content of Web. Generally from web only uncertain search queries can be formulated and the human reader must extract and integrate the information. The availability of huge amounts of data on the Web is responsible for several issues. Unfortunately most of the problems does not address by XML. XQuery is a general purpose query language for XML data [9]. For that reason much effort is spent for developing a query language for XML that allows SQL-like queries over XML documents.

Query languages are required for XML for data extraction, data filtering, data conversion, data integration and construction of XML documents. Transformation with XSLT (XML Stylesheet Language Transformation) requires pattern sophisticated extraction language like XPath.

Xquery is designed to meet the requirements identified by W3C. Key importance of XQuery is extraction of required documents from large documents. XQuery is derived from existing XML query language called Quilt, which extracted features from several other languages like XPath, XQL, SQL, OQL etc. According to W3C definition XQuery operates on the abstract, logical structure of an XML document, rather than its surface syntax. This logical structure is also known as the data model. XQuery Language includes keywords like let, result, for, where, sortby etc. Path expressions is a new feature in XQuery that are used to bind variables in the 'for clause'. Several built-in functions like distinct( ), document( ) simplifies the job of query. Figure 2 shows the sample code of XQuery that generates Locus information from two PDB files.

Ontologies and Bioinformatics: Ontologies enable the developer to practice a higher level of reusability. Ontologies allow developers to share application domain knowledge using a common vocabulary and enable the developer to concentrate on the structure of the domain [10]. The main purpose of ontologies is they work as a

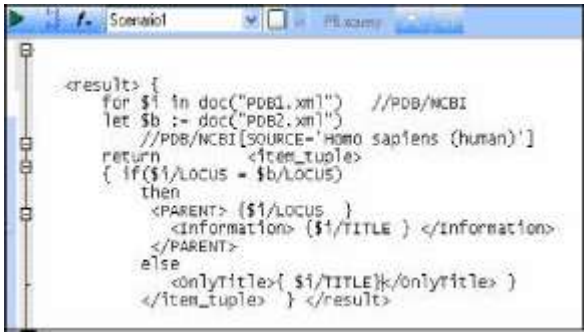

Fig 2: XQuery for extracting data from PDB files

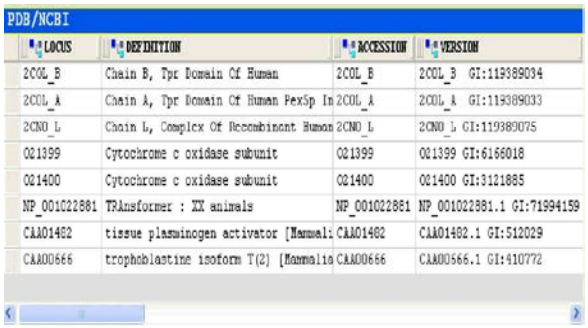

Fig 3: PDB File in Grid format using ODA model

basis for communication between people or between agents in near future. Some of the applications of ontologies are storing data, search process, retrieval, software development, classification of data resources, policy enforcements. Semantic Web based on ontologies that provide shared and common understanding of a domain that can be communicated across people and application systems. ODA provides grid format that shows protein flat files in row-column format as shown figure 3.

Many bioinformatics applications would benefit from comparing proteins based on their biological role rather than their sequence. In most biological databases, proteins are already annotated with ontology terms. XML provides semantic information for defining the structure of the document. It prescribes a tree structure for documents and the leaves are well-defined tags. XML schemas [11] provide a rich set of data types that can be used to define the values of elementary tags. The RDF (Resource Definition Framework) provides a means for adding semantics to a document without making any assumptions about the structure of the document [12]. OWL (Ontology Web Logic) provides more accurate web searches, knowledge management and intelligent agents. In general, ontology concepts and represented by terms [13]. Bioinformatics concepts might be represented by bio entities like proteins, locus, title, organism, comment etc. [14, 15]. The following figure shows the ontological representation of protein database.

XQuery Model for ODA: Primarily XQuery processing comprises of several processing domains namely external processing, query processing and schema import processing. External processing includes generation of instance documents and serialization. Query processing domain that includes static analysis and dynamic evaluation phases.

XML data model should be generated before a query processed, for that data generation steps are used. Documents are parsed with the help of XML parser and generates in XML Information set, this is preserved after validation against schemas. Mapping between source document and processed query is required. The mapping scheme in ODA model is shown in figure.

The Information Set may be transformed as Data Model instances. XQuery is defined in terms of XML data model without placing any constraints on how these instances are constructed. Based on the given protein data file, mapping scheme will be generated. According to the diagram left hand side shows the elements of NCBI document. In the diagram right hand side shows the target document generated from the given query. Based on the Common LOCUS information from two PDB files result is generated. Before generating result, data is compared with the source 'Homo Sapiens (Human)'. Finally the matched documents are printed with required information [16].

Schema import processing includes steps for generating in-scope schema definitions with the help of XSD or from direct generation. Query Processing defines two phases of processing static analysis phase and dynamic evaluation phase [17]. In the static analysis phase query is parsed into an internal representation that

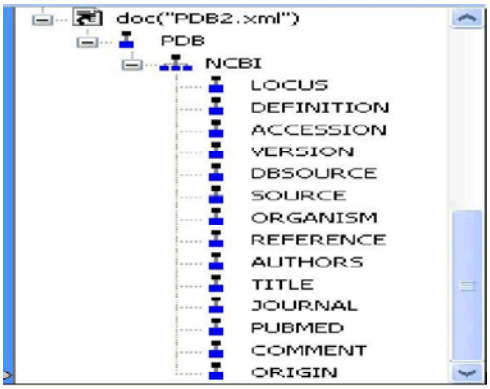

Fig: Ontological Structure

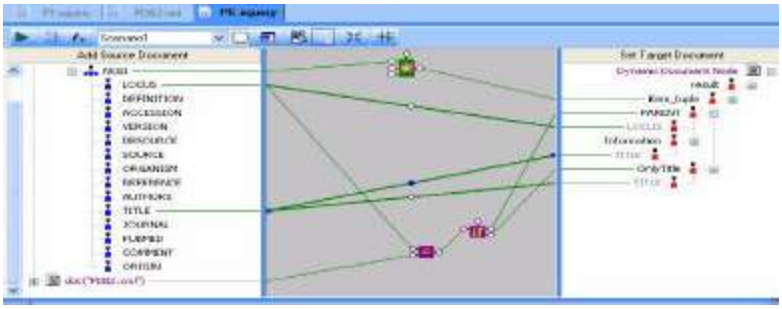

Fig: ODA XQuery Mapping Model

calls the operation tree. The operation tree is then normalized by using some explicit and implicit operations. If there are no errors in static analysis phase then only dynamic evaluation phase is started. The dynamic evaluation phase depends on the operation tree of the expression being evaluation, on the input data and on the dynamic context. This phase may create a new model values and it may extend the dynamic context [18]. Serialization is the process of converting an XML Data model instances into a sequence of octets.

CONCLUSIONS

The goal of this study is to identify challenges in the discovery, search and access of protein data sources that can be addressed by using the Semantic Web in conjunction with database technologies and techniques. We address these needs by providing an extraction method that supports acquisition of domain ontologies from textual sources. ODA model is proposed for addressing these issues. Evaluating the performance of our extraction is a complex issue, which we addressed from different perspectives. From a qualitative perspective, the biologist considered the extracted ontology as a helpful first step in the creation of bioinformatics ontology. We concluded that automated support for extraction and hierarchical structuring would provide a valuable benefit.

REFERENCES

1. Siegel, J., 1996. CORBA, Fundamentals and Programming, Wiley.
2. Junker, V.L., R. Apweiler and A. Bairoch, 1999. Representation of functional information in the SWISS-PROT data bank. Bioinformatics, 15: 1066-1067.
3. The EMBL outstation-European Bioinformatics Institute (EBI), http://www.ebi.ac.uk/
4. The National Center for Biotechnology Information (NCBI), http://www.ncbi.nlm.nih.gov/
5. BLAST, http://workingobjects.com/blastxml.dtd
6. Thompson, J.D., D.G. Higgins and T.J. Gibson, 1994. Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, Nucleic Acids Research, 22: 4673-4680.
7. Cattell, R.G.G., 1996 The Object Database Standard: ODMG-93, Morgan Kaufman, San Fanscisco.
8. Leser, S., U.W. Fleischman and R. Apweiler, 0000. DITtoTrEMBL: a distributed approach to high-quality automated protein sequence annotation, Bioinformatics, 15: 219-227.
9. XQuery An XML Query-Official Web site for Xquery, http://www.w3.org/TR/ xquery/.
10. Gruber, T.R., 1995. Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, 43(5/6): 907-928.
11. The Extensible Markup Language W3C Recommendation, 2000. http://www.w3.org/tr/ 2000/rec-xml-20001006.
12. Dave Beckett, 2002. editor: RDF/XML Syntax Specification (Revised). W3C Working Draft, March http://www.w3.org/TR/rdf-syntax-grammar/
13. Mike Dean, Dan Connolly, Frank van Hermelen *et al*., 2002. editors: *OWL Web Ontology Language 1.0 Reference Description*. Draft document, June 2002. http://lists.w3.org /Archives/Public/www-webont-wg/2002Jun/att-0124/01-owl-ref-proposed.html
14. Craven, M., D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigram and S. Slattery, 1998. Learning to Extract Symbolic Knowledge from the World Wide Web, AAAI/MIT Press.
15. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Detection of Material hardness using tactile sensor, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1713-1718.

16. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Blue tooth broad casting server, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1707-1712.
17. Saravanan, T. and R. Udayakumar, 2013. Comparision of Different Digital Image watemarking techniques, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1684-1690.
18. Udayakumar, R., V. Khanna, T. Saravanan and G. saritha, 2013. Cross Layer Optimization For Wireless Network (Wimax), Middle-East Journal of Scientific Research, ISSN: 1990-9233, 16(12): 1786-1789.