International Journal of Advance in Computer Science & its application – IJCSIA 2018
Copyright © Institute of Research Engineers and Doctors , SEEK Digital Library
Volume 7 : Issue 2 [ISSN : 2250-3765] - Publication Date: 25 June , 2018

The Integrated Role of XML and Java in Historical Data Processing

[Dr.Reshmy Krishnan]

***Abstract*— Basic requirements to process historic data are finding, collecting, structuring and retrieving. Key challenges facing by historians during transformation are heterogeneous data format, data exchange, cross platform, multiple languages etc. Conversion between source and recipient database is also a difficulty. During the transformation, data sharing and data security are the key concerns faced by historians. Data exchange, compatibility, extendibility and reuse of data becomes progressively important while collecting and processing data. Here combination of Java with XML is checked to resolve above concerns. How the features of Java and XML can be supportive for data sharing, data extension and cross platform issues are being discussed here. These paper insights into the historians everyday practices to manage constraints and concerns in the areas of data access, collection, sharing and security. This paper reveals how the usability of XML and Java can reduce the challenges and enhance data sharing and data security.**

Keywords*— *data sharing; data security; XML. Java Introduction

I. Historical Data Collection

A. The Importance of Historical Data Collection.

Social and rational perceptions of historical information are valuable in all fields of study. Historical research calls for "the use of scientific principles to study the interrelationship of social, economic, political, and psychological factors that influence ideas, events, institutions, and people"[Weller et.al,(2015)].The awareness about ancient data boosts our insight towards the finding solutions of currents problems. This also enhances us to develop the strategies to handle the present situations".

Since a huge volume of data is involved in the historical research, concept mapping is required to gather order and analyze data. Concept meaning is useful to find the semantically relationship between data. Since huge data is involved in historic research, a data gathering tool is required to gather, moderate, consolidate and process data.

Currently most of the data is getting from social media for historical research which remains hidden in majority cases normally [Weller et.al,(2015)].. Various challenges shade social media research such as heterogeneous datasets, different platforms and frameworks, various theoretical conventions etc.

Dr.Reshmy Krishnan
Muscat College
Sultanate of Oman
reshmy_krishnan@yahoo.co.in

Handling of these huge volumes of data and analysis are the key challenges in this topic. Two main phases of historical data research is data gathering and data sharing [Weller et.al,(2015)].. This can be done through direct access or through social media. The challenges faced by historians and researchers are mainly found in data collection and data sharing [Bruns(2003).

Issues related to data sharing can be summarized as follows:

1. Lack of clarity with data and inadequate documentation leads to the problems with quality of data issued by APIs [Bosak et,al,(1999)].
2. Data type from various providers and quantity of data are another key challenge. Limited quantity of data only can be collected through online.
3. Heterogeneous platform and data cause challenges in data collection. Change in data and platform make difficulty or loss of data when updates these to an API and then to website.

Technical complications such as inconsistencies in data collection infrastructure, crashing of coding, insufficient storage space etc

B. How sharing can improve historical research.

If data can be shared between end users, the huge pain in spending time for data collection can be avoided. An efficient data sharing can be effective in avoiding data cleaning..

Data sharing can play an effective role in availability of data. Availability of data can improve the speed of gathering data by making answers available to the research questions historians are adapting. Currently there are obstacles to access data as per the requirement and this creates a negative influence in the quality of data gathered. Another problem in data analysis [Bruns(2003)]. occurs because of data not suitable for analysis rather than shortage of most suitable data for analysis.

Data sharing can enable historical data research through the sharing of data sets for comparative studies, peer review of others work and extension of previous work etc. Sharing of survey questions can enhance research study to extend the work to the next stage and compare the new work with the previous study.

Data sharing faces the key issues such as

1.Lack of clear terms and conditions available.
2.Heterogeneous platforms of data.-As source data are from different platforms, data sharing will be difficult for cross platform.

3.Different data formats-If collected data are of different formats, merging will create problems.

4.Incompatible data sets-To compare different data sets or import previous data sets to the current platform becomes difficult due to the incompatibility of data sets [Kinder-Kurlanda et.al,(2014)]. The new users have limited control over the previous data sets in this case. Various approaches in data gathering and different tools used make the datasets incompatible..

II. XML

Majority of the data representation problems can be avoided using XML standard. Since XML addresses only content, data representation can be done independent. As the data is handled separately from the cascading style sheet or Extensible style sheet language, sharing of data can be done independently with more security.

Data extendibility can be enhanced as XML provides freedom to applicants to use own tags needed for each application. This flexibility leads to generation of huge number of data labels or tags and thus increases the freedom to extend data and work with other organizational data to promote interoperability.

The features like Document Type Definition(DTD) and XML schema which describes the structure of XML make sure the validity of XML applications.XML applications with correct syntax is well formed and those who validated with XML schema or DTD is called well-formed and validated[Seligman et.al,(2001)]. A validated XML application permits other applications to import data which can be validated with XML schema or DTD.

Besides of the above mentioned features, XML provides several benefits such as support for multiple views of the content. Extensible style sheet language (xsl) is helping the developer to mention various styles and rules to transform an XML document to different presentable format. As a result, content of the XML can be managed separately and independently from the presentable views and same content can be displayed in various formats and styles as per the vision and requirement of the applicants. Since structure of data is specified separately in DTD or XML schema, alternative representation of content is also possible with XML document. A minor change in DTD or schema helps in making same content with different layouts or tags in case of making multiple applications of same XML content.

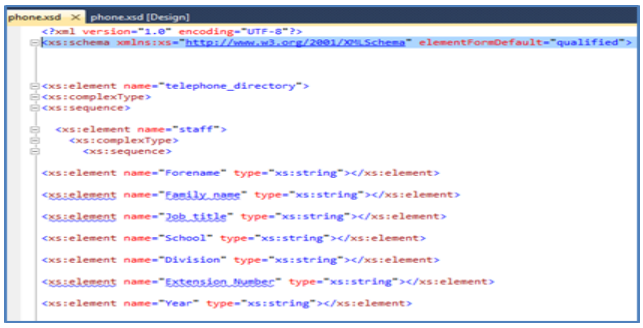

Figure.1. schema and corresponding XML

Selective (field-sensitive) query over the Internet and intranets etc. Previously, separate input was needed for getting display in different formats. As XML keeps the data separately from format, same content can be displayed in multiple outputs and it provides more benefits to the developers and organizations. XML makes sure that fields are retrieved for specified query only since XML provides tags for each data separately. If we mention only a name for searching, XML will not provide data unless it was within author tag. This ensures data security in a large amount.

Visibility of data can be enhanced through the semantic nature of XML data.XML has a standard infrastructure for exchanging data and information. Freely available parsers are there which can validate XML with DTD. The utility of XML is improved by various related standards and hence data sharing and management characteristics of XML

One of the major features of XML, which is document object model (DOM), is a standard for accessing and manipulating XML document. All elements and values of XML are considered as objects and properties in this object model of XML and methods are defined to access these objects. DOM acts as the programming interface for XML and is also with the features language and platform independent. It provides sufficient information to execute a parser without mentioning the data structure or types of operations of the result. DOM presents XML document as tree structure which is an abstract model of the hierarchical structure. Each element of the XML document is considered as nodes in the tress structure. Parent element is the root node and sub elements of the same element are the siblings of the root node. Two types of nodes are represented in tree structure which is internal nodes and external nodes. Internal nodes are the elements with sub elements while external nodes display elements without sub elements. As the elements are represented as nodes in a tree structure, the visualization on the structure of XML documents is clearer and thus increases the accessibility of nodes. Moreover this tree-based application programming interface (API) provides creation, modification and deletion of nodes through the methods such as getParentNode (), getChildNodes (), etc. The relation between nodes can be explicitly represented in DOM.

In order to retrieve data from XML and non XML documents, various XML languages are developed.XML query language is one among them which is having the features for extracting data from diverse sources. This simple nature language is flexible to retrieve data and information from all types of diverse sources such as documents, databases etc. [7].

Data and documents differences can be eliminated by the usage of XML documents as they are structures. They consider historical data as documents and allow documents to be treated as data sources.XML DTD and schema are for data management and to model data.DTD or schema describe the structure of XML document in detail by including root element, other elements, and hierarchy and data type. In addition to this schema explains relationships between elements and constraints also.

A. *Why XML for Historical Data?*

B. *XML for Well Structured Data*

.To share data semi structured data is more suitable since it can be handled automatically. The required data management capabilities are to focus on complicated queries, data integrity, expansion, updates etc. on data. To address these issues, object oriented data base is the best choice. By offering XML data management layer on top of the existing tools, these can be achieved. The existing data bases with support by XML can be utilized for further long-term use with expansion, extension and interface.XML is the best tool to make existing data into a semi structured form.XML supports hierarchical model and hence make data manipulation easier. Searching can be made easier since XML provide the feature of tag names and path operations. Operators can locate the data more easily using the tag names or manipulation paths at any depth. Eg. Find all books that have price<50$ and category ‘database’Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.

C. *XML and Data Sharing*

Recent research signaled that XML can play a vital role in data sharing. Jon Bosak [[Bosak et,al,(1999)].3] indicates that XML (together with XSL) will bring “complete interoperability of both content and style across applications and platforms. A predefined structure is not insisted by XML and hence provides more flexibility to the users. The contents need not have predetermined attribute list, type description and missing attributes can be represented as null. This irregular structure supports more suppleness for data sharing since documents differ in their structure as they are collected from various sources. When data are integrated and added, there is a possibility to change the structure also. To interface with web resources also this semi structured model of XML will be helpful. As XML provides well-ordered data sections, the complexity in representing sequence can be reduced and this improves the query processing. For data sharing, the architecture of source data should be required to change to new request of architecture. There are many challenges in data sharing such as distribution, heterogeneous data structure, heterogeneous languages, different attribute representations and semantics, different schemas, object identifications, data security, different standardization in data value etc.

One of the key challenges in data sharing is distribution of method invocations and results across the distributed networks. It can be enhanced by providing techniques to request remote function among web. Much architecture like SOAP use XML as a mechanism to invoke parameters, return values, exceptions etc.XML along with many protocols such as CORBA, DCOM, and HTTP can act as a method for invocation mechanism. Besides of this, data sharing demands mechanisms to create, send and read interchange files. XML avoids variance in syntaxes and semantics by providing middleware protocols which are layered on top of XML

Differences in data structures and languages are another difficulty in data sharing. Since features in middleware products and other standards for connectivity (ODBC) are hidden in the middleware tier, complexity to handle will be increased and efficiency will be reduced. This complexity can be reduced by the use of XML through the use of graph structured syntax for nested, tagged elements with links. These data structures can be accessed by DOM or a query language. It is found that a common model of data can be provided by XML to integrate heterogeneous data bases.

Same concept will be represented in different contexts in various sources. The more difficulty is the variance in semantics in different sources. The change in semantics can be resolved through transformations only in normal cases. As automated transformation is difficult, this issue keeps remains as a challenger for researchers and historians. This can be addressed by XML and its infrastructure by providing enough features. Since detailed information about data can be given as attributes in XML, this challenge can be resolved up to a limit [13].The user can specify the details in attributes and can avoid the ambiguity in semantics. Even though source and recipient has common understanding of semantics, representation can be different. This is due to the lack of standard to determine the semantics of a particular word. The previous procedure for standardization was not effective due to the insufficient tools to notify about the concepts to the stakeholders. Through the feature of namespaces, XML can provide an easy way to disambiguate names from various sources.

Another challenge facing by researchers and historians are heterogeneous schemas. A standard interface schema can be defined by XML DTD or schema so that a structure can be connected external or internal to an application without changing the application. As a DTD can be developed for a common community and this can be connected to the application, multiplicity in interfaces can be reduced and data sharing can be more easier.XML DTD or schema enables interoperability through mapping between XML elements and models. The issues in model standardization can make simpler through the creation of XML Schema and hence it is possible to map different organizational schemas to the standards.

If two objects refer to the same object in the real world, object identification issues will be happened.XML makes it easy to join ambiguous elements to any output. Data value uncertainty is also another concern in data sharing. As data value reconciliation depends on the metadata attached to the

data, XML can support to attach annotations by return a set of alternative elements for an uncertain value.

D. *XML and Databases*

Current relational database management systems can be easily integrated with XML data since it can be very well store in the database. Many tools are developed to create XML extracts of databases and utilities are imported to accept XML.As the output of XML contains its own schema, data in XML is self-describing with the use of tags. All the records in relational databases can be easily represented using XML

E. *How important is Java*

Interoperability feature which is required for the integration of resources can be supported by the use of Java programming. It is used to make different APIs interoperate smoothly by transforming legacy collections to modern collections. Interoperability makes the system to work with other systems without special effort. The programming difficulty of the programmers can be reduced since automatic memory management, garbage collection, etc. are supported by Java language, and it is simple to use and type-safe in design [Gong(1997)]. Range checking on arrays also will be done by Java and help the developers to make robust code. Many modifiers are there in Java to limit the entry to the class implementation and can be allocated in classes, methods and fields. Data security can be ensured using the three access levels such as private, public and protected.

Various levels of restriction can be obtained by the use of appropriate access levels. The most restrictive one is private modifier which will not allow members to be accessed outside the class[Borgman(2012)]. Any subclass method or other class members in same package can access members of a class if it is protected. Classes in the packages are allowed to access in the same package. The easiness of the programming is more by using different features like inheritance, overriding and polymorphism. The feature – Inheritance allows the subclasses to include attributes and operations from superclass and add more and may redefine them. Java allows any class to have more than one method with same name [14]. The merit of this feature-overriding- is the ability of a subclass to implement a parent class method up on requirement [Chakrabarti (2000)]]. In java an object can be appeared in many forms through polymorphism. A parent class reference can be used to refer a child class object. Many objects can be assigned a same reference variable provided type of the reference variable would determine the methods to be invoked by that object [12].

Java programs are transformed by the compiler into a byte code which is machine independent. The verifier in byte code ensures the reliability of the codes which are executed during runtime. It also makes sure that byte codes adheres Java language specifications, rules and name space regulations. Besides of these modifiers check memory management violations stack underflows or overflows, and illegal data typecasts[Boyd et.al,(2012)]. Java runtime take the preparation after byte codes have been verified by the modifiers.

Overall data security is imposed through different mechanisms in Java. Mainly two aspects are supporting for data security. A ready built platform (JDK) is there in Java on which Java based applications can be executed which provides data security to Java programs. In addition to this, security tools and services are there in Java to enable data security. Java language supports developer to make safer code through the language features like automatic memory management, garbage collection, range checking on strings and arrays. Language security at run time is guaranteed by byte code verifier together with Java virtual machine (JVM).Another feature to ensure that an untrusted applet cannot affect other Java programs is local name space.

III. How integration of Java and XML can help?

The API used by Java based applications which is JDBC can be able to deal with XML also. Moreover JDBC has the capability to create XML data. The use of Java API for XML processing (JAXP) along with JDBC helps to manipulate relational databases and XML. Since JDBC now supports all SQL defined data types, such as CLOB (Character Large Object), XML documents can be comfortably stored. Through this feature XML and JDBC API can work with each other. Gathering Meta data is also another successful feature of JDBC which is effectively capable of creating XML data from a table name. The class Java.sql.ResultSetMetaData which is a part of JDBC API helps to make XML from the table name.

Another Java API –Document Object Model (DOM) which is the cross platform and language independent in nature, defines XML with all objects, properties and methods. It represents XML as a tress structure where each node is an object representing a part of the document. To execute all operations such as access, insert, delete van be done in nodes using methods in XML DOM. Loading of XML document is needed before accessing and executing data[Dietrich-Felber(2004)]. Document builder returns a document object as the root/top of the tress after parsing an XML file. To find a mode at a specified place in DOM tree, XPath expression can be used. Another event based API is also available in Java SAX (Simple API for XML) which is straightforward and handles documents of any size. XML tree is displayed as a stream of events generated by parser instead of data structure. Generally this can be used for large XML document.

.

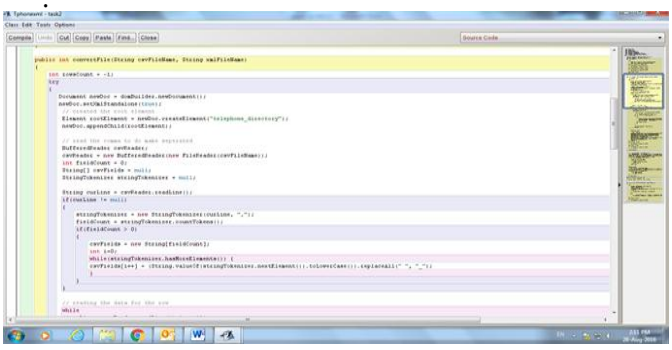

Figure.2. DOM API

References

[1] Weller, K., & Kinder-Kurlanda, K. (2015). Uncovering the challenges in collection, sharing and documentation: The hidden data of social media research. *Standards and Practices in Large-Scale Social Media Research: Papers from the*, 28-37.

[2] Bruns, A. (2013). Faster than the speed of print: Reconciling 'big data'social media analysis and academic scholarship. *First Monday*, *18*(10).

[3] Bosak, J., & Bray, T. (1999). XML and the second-generation Web. *Scientific American*, *280*(5), 89-93.

[4] Kinder-Kurlanda, K., & Weller, K. (2014, June). I always feel it must be great to be a hacker!: the role of interdisciplinary work in social media research. In*Proceedings of the 2014 ACM conference on Web Science* (pp. 91-98). ACM.

[5] Seligman, L., & Roenthal, A. (2001). XML's impact an databases and data sharing. *Computer*, *34*(6), 59-67.

[6] Gong, L. (1997). Java security: Present and near future. *IEEE Micro*, *17*(3), 14-19.

[7] Borgman, C. L. (2012). The conundrum of sharing research data. *Journal of the American Society for Information Science and Technology*, *63*(6), 1059-1078.

[8] Boyd, D., & Crawford, K. (2012). Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. *Information, communication & society*, *15*(5), 662-679.

[9] Dietrich-Felber, U. (2004). Using Java and XML in Interdisciplinary Research A New Data-gathering Tool for Historians Working with EuroClimHist.*Historical Methods: A Journal of Quantitative and Interdisciplinary History*,*37*(4), 174-185.

[10] https://en.wikibooks.org/wiki/XML_-_Managing_Data_Exchange/XML_and_JDBC

[11] https://www.w3.org/TR/xquery/

[12] http://www.tutorialspoint.com/java/java_polymorphism.htm

[13] http://www.tutorialspoint.com/java/java_overriding.htm

[14] http://www.tutorialspoint.com/java/java_inheritance.htm

About Author (s):


Image

Dr.Reshmy Krishnan presently working as associate professor in Muscat University college,Sultanate of Oman has finished Ph.D in 2005 in the area of semantic Information retrieval and Post doctoral in 2013 from university of Stirling.