

# XML as Import and Export of Data

**Binjal Gohil<sup>1</sup>, Tejal Chavan<sup>1</sup>, Vedant Kulkarni<sup>1</sup>, Sudarshan Kumar<sup>1</sup>**

8th Semester, Department of Computer Engineering, Dr. D Y Patil Institute of Technology, Pune<sup>1</sup>

**Abstract:** This paper presents the survey of XML technology in data integration, import, and export of data. In this paper, we study the XML technologies and design a new data integration structure for data import and export. This study shows that it has a great effect on improving the efficiency in importing and exporting the data from one system to another.

**Keywords:** Data integration, data import, and export, file transfer, performance factors.

## 1. INTRODUCTION

XML (extensible MarkupLanguage) is extended version of SGML and SGML (Standard Generalized Markup Language) is simplified version [1]. It is designed for data interchange and data representation on the web.

Maintenance and the natural growing of information systems imply many times; information is not stored in the most convenient structure or in the desired platform: then it becomes necessary to export it into another system or to transform its structure. How to manage the data efficiently?

Previously XML was accepted as the neural format for information representation. This is due mainly two factors. One is XML documents are purely textual files, structured and independent of any hardware or software platforms. Other is more and more public domain tools are available, to help us for transforming XML documents.

## 2. XML

XML is a subset of the Standard Generalized Markup Language (SGML) .it is defined in the ISO standard 8879:1986 [2]. XML was designed to make it easy to interchange structured documents over the Internet.

XML files always particularly mark the start and the end of each of the element(called logical part) of an interchanged document and XML defines how Internet Uniform Resource Locators can be used to identify component parts of XML data streams.

Defining the role of each valid element of a text using a formal model, known as Document Type Definition (DTD), XML users can check each component of a concrete document occurs in a valid place within the interchanged data stream.

An XML DTD allows computers to check users do not accidentally enter a third-level heading without having entered first a second-level heading and it is something that cannot be checked using the HyperTextMarkup Language (HTML).

SGMLrequire the presence of a DTD while XML does not require the presence of a DTD [3]. If DTD is not available, either because all DTD or part of DTD is not

accessible over the Internet or because the user failed to create DTD, an XML system can assign a default definition for undeclared components.

XML allows providing processing control information to support programs, such as document validators and browsers. It brings multiple tiles together to form compound documents and identify the places where illustrations are too highly incorporated into text files, and the format used to encode each illustration.

XML does not have a predefined set of tags (same as an HTML) to markup all the existing documents. XML was not designed in a standard way for coding text.it is impossible to devise a single coding scheme that would be suitable for all applications. XML is a formal language and that can be used to pass information about the component parts of a document [4]. XML is a meta language to define special-purpose markup systems. XML is flexible language that allows the description of any logical text structure (a form, letter. report, book, dictionary or database. etc.).

## 3. DOM

The DOM is standard for accessing and manipulating documents. It represents XML document as a tree-structure. All XML elements are accessed by the XML DOM.

The XML DOM is:

- A standard object model for XML
- A standard programming interface for XML
- Platform-independent and language-independent
- A World Wide Web Consortium(W3C )standard

The XML DOM is a standard for how to get, change, add, or delete XML elements.

### 3.1 Programming Interface:

The DOM models in XML is a set of node objects. The nodes can be accessed either with JavaScript or other programming languages. In DOM the programming interface is defined by a set standard properties and methods.



3.2 XML DOM Methods:

DOM API contains interfaces. Interfaces represent different types of information that can be found in an XML document, such as elements and text [5]. It includes the methods and properties which are necessary to work with this kind of objects. Properties define the characteristic of the node and methods give the way to manipulate the nodes [6].

- n.getElementsByTagName(name) - get all elements with a specified tag name
- n.appendChild(node) - insert a child node to n
- n.removeChild(node) - remove a child node from n

The Document Object Model (DOM), it is an application programming interface (API) for XML and HTML documents. It defines the logical structure of documents. It provided the way document is accessed and manipulated.

3.3 XML DOM Properties

These are some typical DOM properties [7]:

- n.nodeName - the name of n
- n.nodeValue - the value of n
- n.parentNode - the parent node of n
- n.childNodes - the child nodes of n
- n.attributes - the attributes nodes of n

Note: In the list above, n is a node object.

XML DOM is work as standard object model for XML. In XML document hierarchy of informational units is called as nodes. DOM describes a standard programming interface for those nodes and the relationships between them [8].

As XML DOM provides an API that allows a developer to add, edit, move or remove nodes at any point on the tree. For the creation of an application.

3.4 Parser

Below is the figure for the DOM structure which shows that parser evaluates an XML document as a DOM structure by traversing through each node.

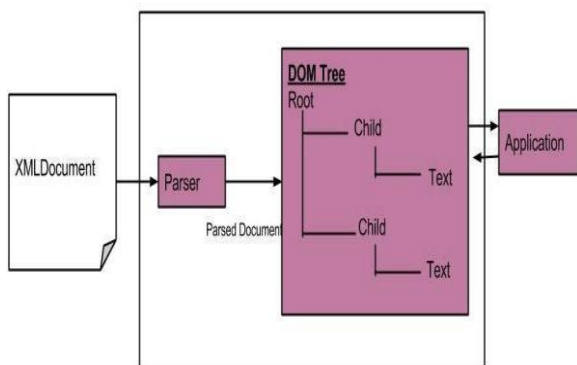


Fig. 1. Dom Structure

A parser is a software application that is designed to analyze a document. Some of the DOM based parsers are listed in the table below:

Parser	Description
JAXP	Sun Microsystem's Java API for XML Parsing (JAXP)
XML4J	IBM's XML Parser for Java (XML4J)

In a tree-based API of DOM, the parser traverses the XML file and creates the corresponding DOM objects. Then you can traverse the DOM structure back and forth.

4. DTD

XML Document Type Declaration is known as DTD. It is a way to describe precisely the XML language. DTDs check the validity of structure and vocabulary of an XML document against the grammatical rules of the appropriate XML language.

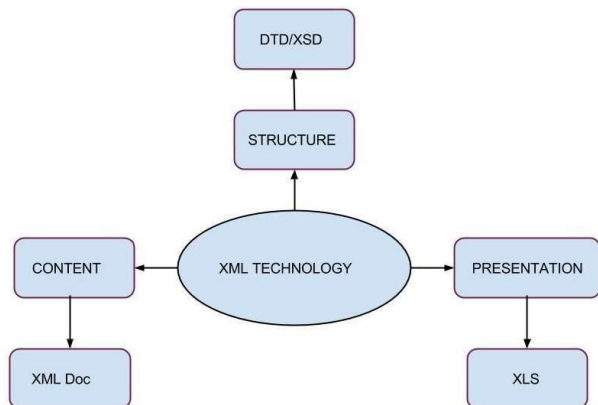


Fig. 2. DTD check for validity of structure

Classification of DTD based on XML documents

- Internal DTD
- External DTD

When a DTD is declared within the file it is called Internal DTD and if it is declared in a separate file it is called External DTD [9].

5. XSLT

XSLT was commonly known as Extensible Stylesheet Language Transformation. XSLT is a way to transform the XML document into other formats such as XHTML.

In HTML document, tags are predefined such as a table, div, and span; and the browser knows how to add style to them for display those using CSS styles. But in XML documents, tags are not predefined. In order to understand and style an XML document, World Wide Web Consortium (W3C) developed XSL. Which act as XML-based Stylesheet Language for XML. An XSL document specifies or guides how a browser should render an XML document.

An XSLT stylesheet is used to define a marked change in form rules to be applied to the target XML document. XSLT stylesheet is written in XML format [10]. XSLT Processor takes the XSLT stylesheet and applies the



transformation rules to the target XML document and then it generates a formatted document of XML, HTML, or text format. Formatted document is then utilized by XSLT format to generate the actual output which is to be displayed to the end-user.

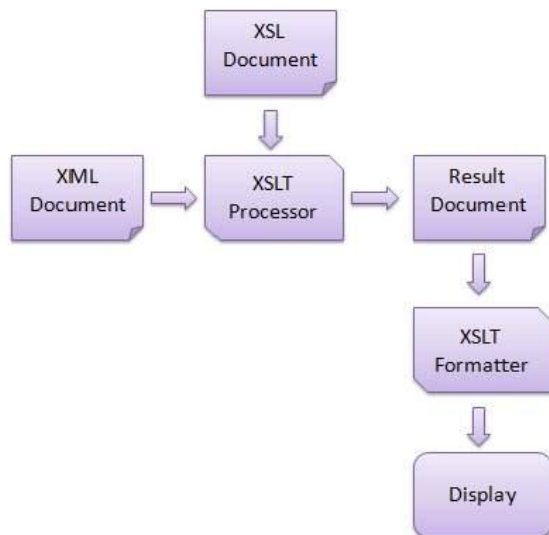


Fig. 3. XSLT change in form rules

## 6. CONCLUSION

One good characteristic about XML and the new tools or gadget built on XML is that we can build applications that are driven by a single information model instead of multiple data models accommodating each application function. By changing the underlying XML, we can change the behaviour and functionality of application programs rather than by changing code. Additionally, we can optimize performance by changing the way in which information is represented. Even in environments not fully leveraging XML as a central information model, it is important to design good XML for the purpose of maintainability and readability.

## REFERENCES

- [1] K. Ahmad, "A comparative analysis of managing XML data in relational database". Lecture Notes in Artificial Intelligence LNCS/LNAI 6591, 2015, pp: 100-108.
- [2] M.Atay, C. Chebotko, D. Liu, S. Lu and F. Fotouhi. "Efficient schemabased XML-to-relational data mapping". Inf. Sys., 3, 2007, pp: 458-476.
- [3] F.Fasseti and B. Fazzinga, "Approximate functional dependencies for XML data". Local Proceedings of ADBIS, 2007, pp: 86-95.
- [4] S.Shahriar and J. Liu, "On defining functional dependency for XML". International Conference on Semantic Computing, 2009, pp: 595-600.
- [5] T. Trinh, "Using transversals for discovering XML functional dependencies". Proc. of FoIKS, LNCS4932, 2007, pp:199-218.
- [6] P.Buneman, S. Davidson, S., Fan, W., Hara, and C. Tan, "Keys for XML. In Proceeding of the 10th World Wide Web Conference. 2001, pp. 201-210.
- [7] J. Qin, S. Zhao, S. Yang, S., Dou, "XPEV: A storage model for well-formed XML documents". FSKD. LN AI 3613, 2005, pp.360-369.
- [8] M. Yoshikawa, T. Amagasa, and T. Shimura, "XRel: A path-based approach to storage and retrieval of XML documents using relational database". ACM Transactions on Internet Technology, vol.1, 2001, pp.110-141.
- [9] P. Bohannon, J. Freire, J., P. Roy, and J. Simeon. "From XML schema to relations: A cost-based approach to XML storage". In Proceedings of the 18th International Conference on Data Engineering, pp.64-74. 2002.
- [10] Y. Chen, S. Davidson, and Y. Zheng, "Constraint preserving XML storage in relations". In Proceeding of the 9th International Conference of Database Theory, 2002, pp.7-12.