

# XML Markup Languages Framework for Programming in 21<sup>st</sup> Century towards Managed Software Engineering

**Khubaib Ahmed Qureshi**

Department of Information Technology,  
Hamdard Institute of Management Sciences  
Hamdard University, Karachi, Pakistan

**M Zeeshan Ali Ansari**

Faculty of Computing,  
The City APIIT, Karachi, Pakistan

**Abstract** - *It is highly expected that versatility of XML could play vital role in automated software development. Extensible Markup Language (XML) defines a set of rules that help to determine the structure of data and document. Usage of XML brings many advantages to people. However there are some deficiencies in XML that have identified. Nevertheless XML is effecting almost every aspect of computing among which is Programming. The paper initially attempts to analyze the different dimensions in which XML Markup Languages are affecting the realm of programming and attempts to predict the significance of each effect for the 21<sup>st</sup> Century. Finally on the basis of our micro analysis and capabilities of XML, the bases of our proposed XML based, XML Markup Language Framework is provided for managed software development.*

**Keywords:** Managed Software Development, XML based XML Markup Languages Framework, Software Code Management

## 1. Introduction

The only game in XML is the way it is used. Therefore covering the variety of advance dimensions exposed by XML, are extremely expected for proposing the basic framework of managed software development under one roof of XML. In general Markup Languages enable people to markup documents. Markup has been done since the beginning to describe or signify importance of certain parts of documents. XML is another markup language that offers much variety and varied applications. It defines rules that guide the people to create their own markup languages and there is also a range of its use. This is the prime reason for its popularity, but obviously not the only reason. There are several other reasons that have been highlighted in this paper.

Therefore the purpose of this paper is twofold, initially it attempts to analyze the different dimensions in which XML Markup Languages are

affecting the realm of programming and attempts to predict the significance of each effect for the 21<sup>st</sup> Century. Finally on the basis of our micro analysis and the capabilities of XML, the bases of our proposed XML based, XML Markup Language Framework is provided for managed software development.

These reasons, in conjunction with key terms associated with XML, advantages and drawbacks are discussed in the Section 2. The popularity of XML can be measured by the extent to which XML has penetrated in miscellaneous domains. Section 3 attempts to list some of the applications of XML in various fields. Among these is the application of XML in field of Programming. In 21<sup>st</sup> century XML is being used in the field of programming in a number of ways. These include XML based representation of source code to XML based syntax. From embedding of XML documents as first class types in Object Oriented languages to stimuli of shift from OO languages to functional programming. All this is discussed in the Section 4. Section 5 is an attempt to contribute some more ideas to the existing application of XML in programming. Finally Section 6 provides the basics of our proposed framework followed by conclusion and future research directions in Section 7 and 8.

## 2. Introducing XML

The initial attempt to develop XML was to facilitate large-scale electronic publishing [1]. XML defines a set of rules that enable its uses to semantically break a document into identifiable parts [2]. What makes XML different from other markup languages is that, it allows the people to create tags of their own and to define the sequence in which these tags can be nested. Thus it is said that XML allows people to define their own vocabulary and grammar. This flexibility has resulted in its wide popularity in every area where exchange, transfer or storage of information is required.

An XML file can be standalone or can be as per the rules of DTD or Schema. DTD stands for Document Type Definition and consists of rules which define the

vocabulary and grammar of the contents stored in a document.

Schema can be considered as a refined form of DTD and has the ability of describing vocabulary and grammar, more precisely.

A parser is a piece of code that checks the correctness of XML document. If the XML document is in conformance to its DTD or Schema, then it is considered as a Valid document. However if the document is in conformance with the basic rules of XML is considered as WELL-FORMED document.

DOM (Document Object Model) and SAX (Simple API for XML) are interfaces that allow the programmer to dynamically access and update XML data.

Another benefit of XML is that it allows separation of concern. While data described through tags is stored in one file, the formatting information is stored in a style sheet (similar to CSS and HTML) known as XSL.

X-Path is an expression language that allows navigation through the hierarchical structure of XML document, where as X-Query is a query language that has been designed with the objective of querying XML information sources.

The above was a glance at various technologies that are associated with XML and must be known to the reader of this paper.

These wide ranges of technologies that are associated with XML encourage the usage of XML however there are several other benefits that are briefly mentioned below [1].

- Specification of customized vocabulary and dictionary
- A simple data format
- Less chances of data corruption during transfer
- Separates contents from presentation
- Non-proprietary and platform independent.

A part from these there are several drawbacks of XML. These are mentioned below [3].

- No difference between part and type
- No identification of type of literals
- Many forms of representation for a single document [4].
- XML is detailed and Verbose.
- Consumes greater capacity and bandwidth

### 3 Applications of XML

There is a broad spectrum of domains in which XML is being used. XML has been embraced by leading

solution providers like Microsoft and Sun. The .NET framework extensively makes use of XML and provides full support to it. Similar is the case with Sun Microsystems which uses XML in its various technologies, primarily for software configuration and other reasons. However computer science is not the only domain where XML is being applied. One of the popular applications of XML is CML (Chemical Markup Language) [1]. CML attempts to organize information about complex chemical objects in a format that is easily understandable and searchable by computer systems. Another such applications is MathML [1]. As the name suggests, this markup language is used to represent mathematical equations in computer systems. NASA (National Aeronautics and Space Administration ) is also taking interest in using XML. The ADC (Astronomical Data Center) of NASA is analyzing benefits of using XML to store their astronomical data [5]. In the field of Graphics, major giants have come on table to develop an XML based format for Image representation known as SVG (Scalable Vector Graphics) [6]. In the same field of graphics, a specification is being developed by W3C consortium for interactive simulations within the World Wide Web [7]. Open Financial Exchange (OFX) is being used by Microsoft Money and Quicken to provide online banking, stock trading and other electronic business [1]. VoXML stands for Voice Markup Language. VoXML is identified by Motorola as 'The Markup Language for Voice Applications' [8]. Another application of XML is WebServices. These are Objects that communicate with each other via an XML based protocol rather than propriety protocols. XML is also effecting the programming world in a number of ways. Following is an overview and analysis of the dimensions in which programming arena is being shaped by the XML.

### 4 XML and Programming Languages:

After research we have identified that great interest in the industry is being shown by the people to integrate XML into current programming techniques. A lot of predictions are being made and practical steps are being taken. The effects of this interest and actions can be categorized into four areas.

1. XML as a syntax of programming language
2. XML stimulated expanded type systems.
3. XML promoting other programming techniques
4. XML as a representation of source code
5. XML for separation of concerns

## 4.1 XML as syntax for programming language

XML is being widely used in the current software approaches. The usage includes, “Using XML to store information” and “Using XML to exchange and communicate data”. It has been observed that a lot of code has to be written for this data marshalling [9]. This is basically due to the impedance mismatch between XML and Programming Language [9]. To remedy this situation different solutions have been proposed, one of which is to develop languages that are completely XML based i.e. from syntax to type system. The examples of such languages include XL [9], Water [10], OXML [11]. It has been suggested that future programming languages should be extensible [12], [13]. Since the language evolve by formalizing best practices, thus providing extensibility to the programmer will enable him to formalize his own best practices [13].

*Comments:* Such languages will minimize marshalling but there are a few problems that will hinder the popularity of such languages. Programmers are more comfortable with the syntax of traditional languages i.e. they don't have problems with syntax but the extensive use of APIs. Secondly marked up programming languages are unnecessary lengthy.

Water however uses Concise XML that has been specially developed to address the drawbacks of XML like lengthiness etc. However, till now not much excitement has been shown by the industry for concise form of XML or such XML based languages.

## 4.2 XML stimulated expanded type systems

To address the same issue, i.e. to avoid marshalling and to reduce unnecessary and extensive usage of APIs, it has been recommended that the type of system of language should be extended for native support of XML [14]. In other words XML should be supported in as simple manner as strings, integer and boolean values are supported. This will definitely result in reduced code size. Also more interest can be found in the industry for this approach. Microsoft is working on a research project called XEN [14]. XEN proposes an extended type system for C# in order to support native XML processing. This is not the only example, there are other for example Xact, Xatic [15]. A slightly simpler approach is followed by XJ [15] in which just syntax is extended to support XML however in an additional step before compilation the

extended part of syntax is converted to corresponding DOM API calls.

*Comments:* In our opinion this is a much better approach and will gain much popularity in the industry, this is because there will be no challenge for the programmer of familiarizing with new syntax. Most importantly the objective of code reduction will be achieved.

## 4.3 XML promoting less common programming techniques

As mentioned before impedance mismatch between OO Languages and XML is what's irritating the programmers most. Some people even have predicted that this will eventually cause a shift to other programming paradigms like Functional Programming [16]. The reason that has been given is the nature of some programming languages is very similar to that of XML. Functional programming is declarative and XML is declarative too. Both have support for similar data structures.

*Comments:* In the absence of native support for XML in OO languages, this could have been seriously considered however once native support gets matured in the widely used languages of today, people will find less reason to ship from one paradigm to another

## 4.4 XML as a source code representation format

This type of application of XML is one that is the focus of greatest interest. A lot of work is being done in this domain due to that benefits that could be achieved through this approach. Some of the benefits are listed below [12], [13], [17], [18], [19], [21], [22].

1. Ability to view source code at different levels of abstraction
2. Ability to customize the view of code
3. Ability to embed code and documentation together
4. Ability to query source code
5. Ability to add additional information (some additional data collected after parsing)
6. Ease of activating and deactivating parts of code
7. Ability of linking other information within the source code e.g. images, voice etc.
8. Ability to transform between syntax using XSLT

There are number of efforts that are being made to develop a versatile format for source code representation. Some of the examples are JavaML [20], srcML [18] and CppML [19].

*Comments:* In our opinion this approach is going to gain wider acceptance in the days coming ahead. This is due to the fact that this approach offers great benefits and it does not need any tuning of habits from the programmers behalf. Most of the current efforts in this domain dictate the programmer will be writing code in the regular way where as a tool will be converting that into a marked-up format.

#### 4.5 XML for separation of concerns

This approach is one which is already adapted in the industry. The examples are JSP, ASP.NET and CFML. These languages allow users to create there own set of tags, define functionality of the tag in a language and map them to each other. The responsibility of executing the page containing the tags is that of the application server. The application server when confronts a custom tag, it executes the functionality to which it is mapped.

*Comments:* As mentioned such approach already has enough support and presence in the market. It enhances separation of concerns are reusability. However debugging such implementation is little complex than that of the normal applications.

### 5 What more can be done?

In today's programming world we can see that languages are first transferred to some intermediate form e.g. Java to byte code , .NET to IL, and then this form, on the fly, is converted by the runtime in an executable format. The features of performance, security etc are actually the features of the runtime of that language. Thus the .class and .exe of java and .Net are actually binary versions of the code that is typed, with some added information. In .NET we have seen that languages like C# and VB.NET communicate easily since they both are converted to same intermediate language. This provokes a thought in our mind, the details of that are mentioned below.

It has been observed that the popular languages of today offer similar features to their end user but use different syntaxes. W3 Consortium is currently looking at possible uses of *XML binary* which will be a lighter version of the text based XML of today. If such efforts result, then this XML binary can be used as a standard intermediate format. If all languages are converted to a standard format and follow a standard

type system, then the code of one language can be called by that of another. However vendors will be providing runtimes for the intermediate form. These runtimes will be offering a different level and set of functionalities to its users.

## 6 XML Based, XML Markup Languages Framework

As it has been already discussed that the use of XML as a source code representation format is of valuable interest, in addition to others. However to achieve full efficiency a complete XML Based Framework is needed to combine different supported functionalities. Therefore three layers XML Based, XML Markup Languages Framework is presented in figure 1. The framework makes maximum use of XML capabilities (as discussed in Section 4).

The first layer provides orchestration facility through multiple XML designers provided in every modern native languages, which basically represent the native language code in XML form. The middle layer provides the common tag library defined for XML Markup Language for code management to provide multiple level of abstraction etc. The last layer relate the project code to the metrics with the help of XML tag used, this can be done through code clustering [23], different computed values of the project metric can provide ways to highlight better software development and optimizing throughput.

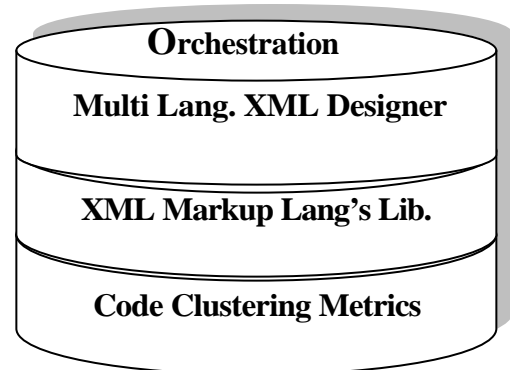


Figure1. Three Layer XML Based, XML Markup Languages Framework

## 7 Conclusions

XML is a markup language for describing new markup vocabularies. It offers some good advantages to its users, however there are some drawbacks. Due to its advantages a great support for xml is found in the industry and one can find enormous applications of XML in heterogeneous fields. XML is also affecting

the way programming is done currently, in the following ways.

1. XML as a syntax of programming language
2. XML stimulated expanded type systems.
3. XML promoting other programming techniques
4. XML as a representation of source code
5. XML for separation of concerns

Among these, the first three are in response to the problem that, accessing XML through a language requires a lot code resulting in too many lines of code, secondly there is too much marshalling in between that takes time. However in our opinion the first technique doesn't resolve the issue. This is due to the fact that writing language primitives in XML will take excessive lines of code, thus nullifying the size reduction caused by removal of API calls to access XML data. Water recognizes this problem and tries to eliminate them through the use of concise XML but we think that in the presence of extended type system with native support of xml, developers will not bother to get familiar with yet another syntax of language and markup. Thus if the second option is taken seriously then the chances of recognition of the first and third will reduce, thus will not cast great effect on 21<sup>st</sup> century programming. The 4<sup>th</sup> and 5<sup>th</sup> applications are the one that are gaining (Former) and have gained (Later), widespread recognition in the 21<sup>st</sup> century. Representing source code in XML offers a large number of benefits that includes, multiple views of same code, querying capabilities, and transformation from one syntax to another on the basis of XSLT. The example of the last are JSP, ASP.NET and CFML in which custom defined tags are mapped to language code, thus enabling users to create their own tags and specify the actions that will result in using such tags.

## 8 Future research directions

Investigate different ways and means of XML and XML like languages to develop Automated Software Engineering, ranging from Automated Software Specifications to Knowledge Management. The similar idea which is found in Semantic Web using Web Services.

## 9 References

[1]. about XML, <http://www.w3.org/XML>  
[2]. Harold. R. H, XML Bible, IDG Books, 2<sup>nd</sup> Edition

[3]. Fry, C. The Trouble with XML, [www.waterlanguage.org/TroubleXML.html](http://www.waterlanguage.org/TroubleXML.html)  
[4]. Features of ConciseXML, <http://conciseXML.org>  
[5]. <http://xml.coverpages.org/nasa-adc.html>  
[6]. Boye, J., SVG Brings Fast Vector Graphics to Web, *Internet Related Technologies*, <http://tech.irt.org/articles/js176/>, 29th July 1999  
[7]. <http://www.w3.org/MarkUp/VRML/>  
[8]. <http://xml.coverpages.org/voxXML.html>  
[9]. Florescu, D., Grünhagen, A, Kossmann, D, 'An XML Programming Language for Web Service Specification and Composition', *ETH Zurich Databases and Information Systems 2003*.  
[10]. <http://www.waterlanguage.org/about.html>  
[11]. o:XML Overview, <http://www.o-xml.org/documentation/whatis-o-xml.html>  
[12]. Wilson, G. Extensible Programming for the 21st Century. *ACM Queue*, November/December 2004-05.  
[13]. Wilson, G. XML-Based Programming Systems, *Doctor Dobb's Journal*, March 2003.  
[14]. Meijer, E., Schulte, W, Bierman, G., Programming with Circles, Triangles and Rectangles, *Microsoft Research*  
[15]. Harren, M., Raghavachari, M, Shmueli, O, XJ: Integration of XML Processing into Java, *The Thirteenth International World Wide Web Conference 2004*  
[16]. Parsia, B, 'Functional Programming and XML', *XML.com, O'Reilly Media*.  
[17]. Simic, H., Topolnik, M. 'Prospects of Encoding Java Source Code in XML.', *ConTEL, 7th International Conference on Telecommunications, June 2003*  
[18]. Maletic, J., Collard M. & Marcus, A., 'Source Code Files as Structured Documents', *IEEE Computer Society, Proceedings of the 10th International Workshop on Program Comprehension 2002*  
[19]. Evan, M., Kostas, K. 'Towards Portable Source Code Representations using XML. *Working Conference on Reverse Engineering*', 2000.  
[20]. Badros, G., 'JavaML: A Markup Language for Java Source Code.' *International World Wide Web Conference, 2000*.  
[21]. Gregory, M., John, M. & Siu, N. 'An Extensible Tool for Source Code Representation Using XML', *IEE Computer Society, Proceedings of the Ninth Working Conference on Reverse Engineering, 2002*.

[22]. Al-Ekram, R., Kontogiannis, K., ‘An XML-based Framework for Language Neutral Program Representation and Generic Analysis’. , *IEEE Computer Society, Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004*

[23]. Nadeem Asif, “Software Code Clustering” International Conference on Mathematics and its Applications-ICMA, Dec. 06, COMSAT Lahore.