ISSN 2250-3153

Software Support for Xml Schema Design Patterns and Pattern Matching of Xml Schemas

R. Bhuvaneswari, K. Kalai selvi

Department of CSE, Saveetha Engineering College, Chennai, India

Abstract- In current era, XML schema design patterns are used in vast extent to exchange data and to employ the modules for reusability. XML as in plain text format can be transmitted between different applications with different platforms, operating systems and browsers. Design patterns play important role in reusability of already existing solutions. The different XML schema design patterns namely GOF, abstract factory, template, singleton, MVP, MVC are used in varied applications. Softwares like Rational Rose support the design patterns in UML models which in turn can be converted to XML schemas. Rational Rose also supports comparison of two schemas. Softwares like NetBeans IDE and .NET also support XML schema design patterns. This paper focuses on conversion of XML schema design pattern into java code, so that a string can be used from client interface to pattern match the XML schema in servers. This reduces the whole XML schema or UML class diagram to be transmitted from client. XML schema can be converted into java code with help of EMF Framework and other aided plug inns which help in easy searching of existing design patterns for reuse.

Index Terms- XML schema design patterns, UML diagrams, EMF model, regular expression, GoF pattern

I. INTRODUCTION

In many real world applications today, XML becomes as the tool for sharing and transporting data between different applications. XML data is in plain text format which makes it easier to get upgraded in different operating systems, applications and browsers.

Even a simple java code can be able to read external XML file and process it. XML files can be converted into java code with the help of frameworks .Java is also platform independent which helps easy sharing of data between applications. In this paper, XML schema matching using different tools is discussed. Also the pattern matching using java classes and methods is discussed and a comparison of the best approach is accomplished. Data can be exchanged between incompatible machines over the internet easily with the help of XML.

II. LITERATURE SURVEY

The growing need for exchange of data between different applications over the internet creates a greater need for easy and comfortable format for transfer of data. Design patterns play important role in reusability of the already developed software. Combining the above two issues results in XML schema design patterns which are now implemented in most of the software platforms aiding in easy storage and transport of data.

In this paper fast and easy means of identifying and mapping the already existing XML schema design patterns is implemented. The already existing papers discusses on matching two xml schemas and the algorithms that exist for matching and comparing the schemas [2]. XML schema matching algorithms also exist which aid in pattern matching. In this paper matching of a regular expression pattern against XML schema is considered. By this approach the burden of clients to construct UML class diagrams which in turn to be converted to XML schema and transmission of whole schema is avoided. Instead only a string for example mentioning a main class like library for library management reusable pattern can be used which can be matched against the java code generated from corresponding XML schemas.

In java various pattern matching classes and methods are available which helps in easy and fast matching of patterns. Today various ADDONS and Plug inns are also coming up. In this paper various softwares in use are analyzed and the approach for easy comparison is also discussed.

Various GoF patterns are used and aid the developer in reusability of many software components. The best XML schema design pattern is chosen so that schema understanding and matching is easier.

In this paper the convenient means of matching a regular expression pattern given by a client to be matched against java code generated from xml schema design pattern is suggested.

A. Rational Rose UML to XML Schema design pattern generation and reverse engineering.

Rational rose supports generation of UML diagrams in which patterns can be inserted using stereo types which can be converted into corresponding XML schemas. Rose also supports reverse engineering that is generating class diagrams from XML schemas. The example of reverse engineering done is shown in the Fig.1 and XML schema below.

<! -- DTD for novel -->
<! ELEMENT novel. log
(preface,chapter+,biography?,criticalessa y*)>
<! ELEMENT preface (paragraph+)> <! ELEMENT chapter (title, paragraph+, section+)>
<! ELEMENT section (title, paragraph+)>
<! ELEMENT biography (title, paragraph+)>

<! ELEMENT critical essay (title,
section+)>
<! ELEMENT paragraph

```
(#PCDATA|keyword)*>
<!              ELEMENT              title
(#PCDATA|keyword)*>
<! ELEMENT keyword (#PCDATA)>

<! ENTITY % test "def IDREF #REQUIRED">
<! ELEMENT wfc EMPTY >
<! ATTLIST keyword      id            ID
#REQUIRED
name  CDATA             #IMPLIED
                        method      CDATA
#FIXED "123" >
```

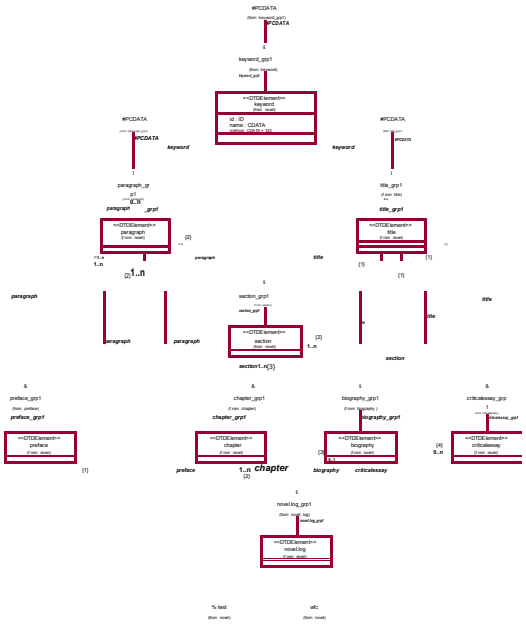

Fig.1 Class Diagram Generated from XML schema using Rational Rose.

B. NetBeans Enterprise Pack XML schema design pattern

The XML tools in NetBeans Enterprise Pack can be used for creating xml schema design patterns and also aids in switching from one design pattern to another design pattern.[5]

Design patterns can be applied to xml schemas by selecting the corresponding xml schema .xsd file.

C. .NET and design patterns

Design patterns improve reusability of business logic by separating the three components required to generate and manage a specific user interface (such as a single Web page). The Model contains the data that the View (the Web page) will display and allow the user to manipulate. The Controller or Presenter links the Model and the View, and manages all interaction and processing of the data in the Model [4].

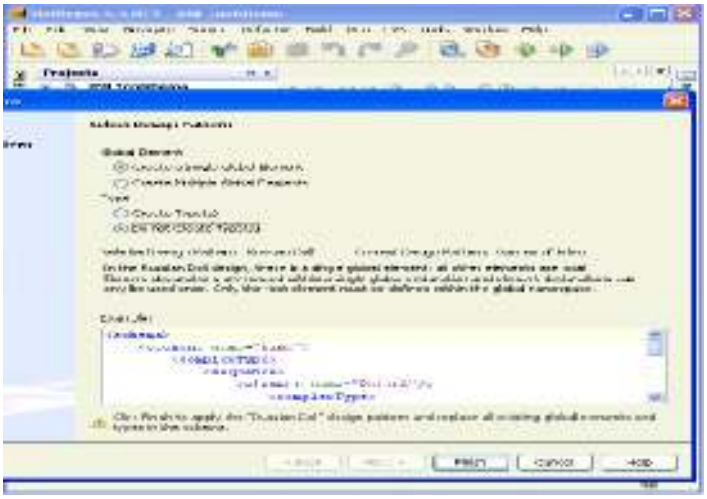

Fig. 2 Selecting Design Patterns in NetBeans IDE

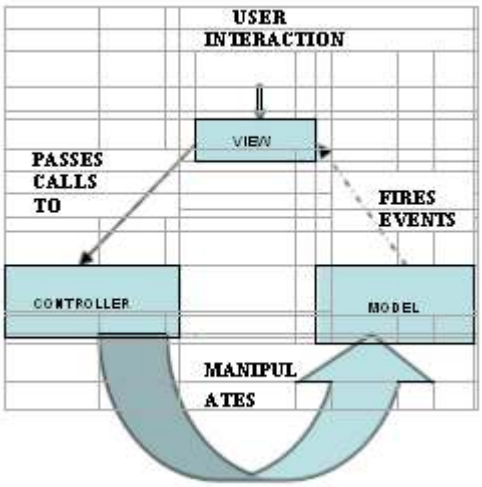

Fig.3 The Model View Controller Pattern

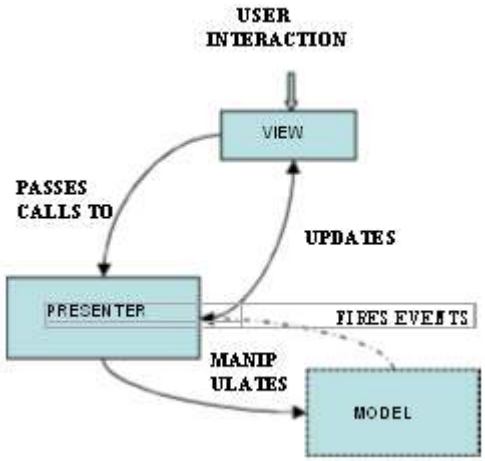

Fig.4 The Model View Presenter Pattern

This introduces dependency between the Model and the View. To avoid this, the MVP pattern uses a Presenter that both updates the Model and receives update events from it, using these updates to update the View. The MVP pattern improves

testability, as all the logic and processing occurs within the Presenter, but it does add some complexity to the implementation because updates must pass from the Presenter to the View [4]. The other design patterns supported by .NET are The Service Agent, Proxy, Broker Patterns, The Repository Pattern, The Singleton Pattern etc.,

III. XML SCHEMA DESIGN PATTERN CHOOSING

Russian doll, Venetian blind, Salami, Garden of Eden is some of the XML schema design patterns available. Venetian blind is considered best because the name complexity is highly reduced because there is only single root element, one global element and all local elements defined inside the global namespace. This also favours the easy matching of schemas [3].

Venetian Blind

Example: <xsd:schema>
<xsd: simpleType name="Husband"> <xsd: restriction base="xsd: string"> <xsd: minLength value="1"/> </xsd: restriction>
</xsd: simpleType> <xsd:simpleType name="Wife"> <xsd:restriction base="xsd:string"> <xsd: minLength value="1"/> </xsd: restriction>
</xsd: simpleType>
<xsd: element name="Couple">
<xsd:complexType>
<xsd:sequence>
<xsd: element name="Husband" type="Husband"/>
<xsd: element name="Wife" type="Wife"/>
</xsd: sequence>
</xsd: complexType>
</xsd:element>
</xsd: schema>

Other patterns like garden of eden,salima slice, Russian doll can also be used depending on the application which have their own advantages and disadvantages.

Design patterns like Russian Doll and Garden of Eden can be selected depending on the requirements like single global element or multiple global elements.

Venetian slice and Garden of Eden are considered as best as they support multiple global elements which will be useful when pattern matching in xml schemas are considered

IV. TOOLS THAT SUPPORT XML SCHEMA MATCHING

Many tools are available today for XML schema comparison .Some of them are AltovaXMLspy, Difflog,ExamXML, Eclipse hyper model etc., The result generated by comparing two xml files using Exam XML is shown below[9].

V. GENERATING AN EMF MODEL USING XML SCHEMA

ECORE model can be generated from xml schema stored anywhere in the workstation using EMF Framework. The xml schema can also be loaded from a rose class model [8].

There are two ways to generate an ecore model [8]

- From XSD to ecore
- Creating ecore models from mdl files

A. Java code can be generated from the ECORE model .

EMF generates Java files. In java java.util.regex package supports regular expression processing. A regular expression is a string of characters that describes a character sequence. This general description called a pattern can then be used to find matches in other character sequences. Regular expressions can specify a regular expression that represents a general form that can match several different specific character sequences [1].

There are two classes that support regular expression processing. Pattern and matcher. Pattern is used to define a regular expression. Match the pattern against another sequence using matcher. So in java finding the matching patterns is easy as instead to compare two xml schemas in case of rational rose and other softwares used to compare XML schema design patterns.

Java is also platform independent. So it is easy to use than other softwares. Also EMF supports both importing UML and converting to EMF model and then to java file. Also any XML schema can be imported and converted to EMF.

Few example sample code in java for Pattern matching is shown below

Simple example for Pattern Matching in Java

```
Import java.util.Regex.* Class Regex {
Public static void main (String args[])
{
Pattern pat; Matcher mat; Boolean found;
pat=pattern. compile (“library”); mat=pat. matcher (“library”); found=mat. matches (); System.out.println (“The matching pattern in this sentence is library”);
if (found)
System.out.println (“MATCHES”); Else
System.out.println (“NO MATCHES”);
}
}
```

Applying regular expressions on the contents of a file

```
import java.util.regex.*; import java.io.*; import java.nio.*;
import java.nio.charset.*; import java.nio.channels.*;
public class CharBufferExample { public static void main(String[] args)
throws Exception {
// Create a pattern to match comments
Pattern p = Pattern.compile ("//.*$",
Pattern.MULTILINE);
// Get a Channel for the source file File f = new File(" LIBRARY MANAGEMENT.java"); FileInputStream fis = new
FileInputStream (f);
FileChannel fc = fis.getChannel ();
// Get a CharBuffer from the source
file
ByteBuffer bb =
fc.map (FileChannel.MAP_RO, 0, (int) fc.size ());
Charset cs = Charset.forName ("8859_1");
CharsetDecoder cd = cs.newDecoder ();
CharBuffer cb = cd.decode (bb); // Run some matches
Matcher m = p.matcher (cb); while (m.find ())
System.out.println ("Found comment: "+m.group ());
```

}
}

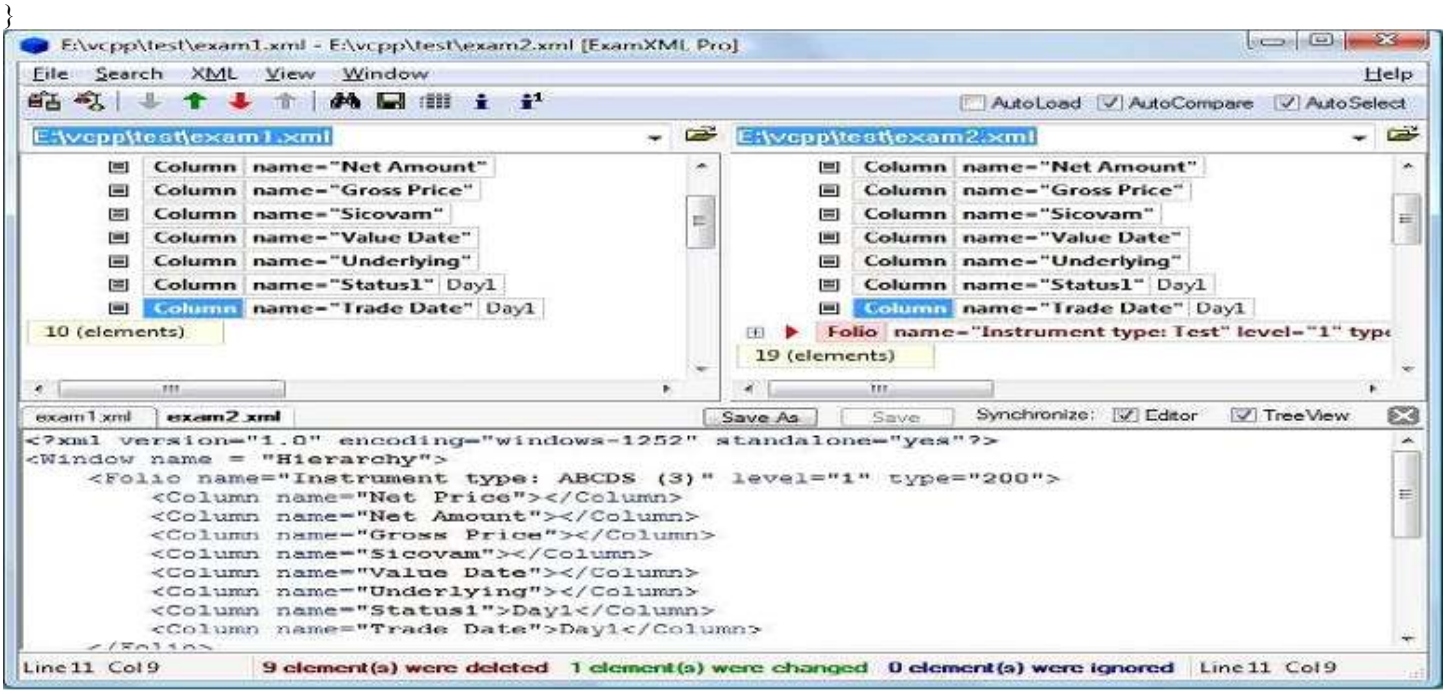

Fig. 5 XML file comparison using EXAMXML

VI. GENERATING AN EMF MODEL USING XML SCHEMA

A. XML Schema Size Vs Efficiency

As the size of the XML schema design pattern increases in size the efficiency is low as the time taken for comparison is ultimately more. If the number of lines of comparison is more it takes more time to compare, even the schema with which it is compared is of less size. But if a single string is used to match a pattern using regular expression classes and methods in java for example pattern and matcher classes the time taken for comparison is very less.

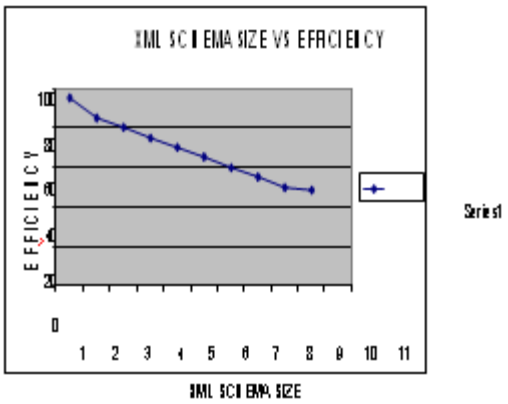

Fig.6 XML schema size vs. Efficiency

B. Time Taken For Comparison Vs Efficiency

Time taken for comparison and efficiency are inversely propotional.If the size of XML schema to be transmitted increases bandwidth increases, time taken for transmission over the network increases, time taken for comparison increases , thus increasing the total time taken and ultimately efficiency is reduced.

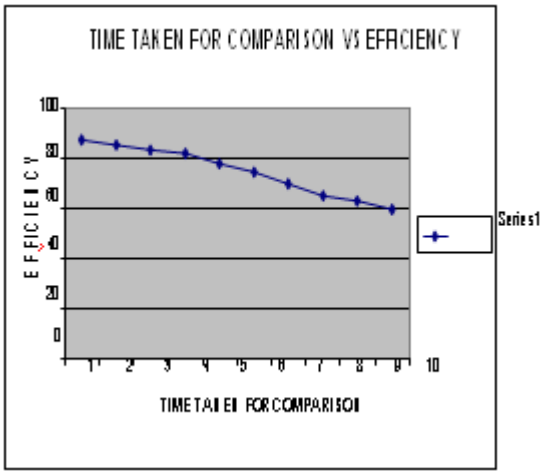

Fig.7 Time Taken for Comparison vs Efficiency

VII. CONCLUSION

Thus pattern matching using a string when the XML Schema or UML model is converted to java code is very simple and consumes less bandwidth memory space and time for comparison as compared to matching two xml schemas. So it is necessary whenever the MVC OR MVP type of models or applications implementing XML schema design patterns are being developed, all the mentioned issues have to be considered. XML schema design patterns or when the patterns are generated from any

UML models like Rational Rose there is facility for converting into java code with the help of EMF like frameworks and other ADDONS and plug-inns. Thus this paper discusses various issues regarding XML schema design patterns that aid in easy transport of data and reusability in many applications.

REFERENCES

[1] Herbert Schildt, Java The Complete Reference 7th Edition McGraw-Hill Osborne Media; 7 edition.

[2] A new algorithm for mapping XML Schema to XML schema http://ieeeexplore.ieee.org

[3] A novel method for measuring Semantic similarity for XML Schema matching Available online: www.sciencedirect.com

[4] W3C XML Schema Design Patterns: msdn.microsoft.com/en-us/library/aa468564.aspx

[5] ASP.NET Patterns every developer Should know http://www.developerfusion.com/

[6] XML Support in NetBeans IDE http://xml.netbeans.org/

[7] Eclipse Modeling Framework (EMF)–Tutorial. http://www.vogella.de/

[8] Innovative XML solutions http://www.a7soft.com

[9] Generating an EMF Model using XML Schema (XSD) http://help.eclipse.org/

[10] Metamodeling with EMF: Generating concrete, reusable java Snippets http://www.ibm.com

First Author – R. Bhuvaneswari**,** B.E and M. Tech. in Information Technology .Currently, she is an Assistant Professor in the Department of Computer Science Engineering Saveetha Engineering College, Chennai, India.

Second B. Author - K. Kalaiselvi, received her B. E. and M. E. in Computer Science Engineering. Currently, she is an Assistant Professor in the Department of Computer Science Engineering, Saveetha Engineering College, Chennai, India