A Generic Open Source Framework for Auto Generation of Data Manipulation Commands

Dr. Poornima.G. Naik[1], Dr. Kavita.S.Oza[2]

[1]*(Assistant Professor, Department of Computer Studies, CSIBER, Kolhapur, India.)*
[2]*(Assistant Professor, Department of Computer Science, Shivaji University, Kolhapur, India)*

Abstract: *Free and open source technology has gained a tremendous importance in software development due to its low cost and holds numerous other compelling advantages both from the perspective of software user and software developer. J2EE and PHP technologies have availed a highest market share. All types of web sites ranging from medium to large sized applications employing these technologies include ample master tables for storing data which are subject to seldom changes. In application development, there is a lot of code repetition in manipulation of data stored in these tables. Further, globalization demands thorough testing of web applications for variety of web browsers depending on the client base. There is a monotonous sequence of tasks involved, from the task of code generation, deployment, setting runtime environment to testing. Hence the automation of these tasks enables a tremendous decrease in development time and cost. Further, customer demands vary in choice of technology, web server and back end database management systems. Also, validation is an integral part of any application which not only ensures that the data is present but also the valid data is stored in the database. Any well written application should enable easy switching on and off of the validation tests as per the requirements. To cater all these needs, in this paper, the authors have designed and implemented a generic framework for automatic generation of data manipulation commands for variety of web servers, web browsers and database management systems (DBMS). All the configuration information pertaining to backend DBMS information, technology information, validation information is stored in an XML file which is parsed using Microsoft XML Parser. Seven test cases are generated for testing the tool in various runtime environments involving various cutting edge technologies. The model is generic and susceptible to future modifications for incorporating new technologies.*

Keywords: *Open Source, JBOSS, Tomcat, JSP, Servlet, PHP, XML Parser.*

I. Introduction

Software development is the core part of software engineering. It requires efficient planning for use of technologies and human resources to enhance the quality. Software market is highly inhabited with new technologies emerging every day. These technologies need different techniques and tools to be implemented. Each technology has its own tool and implementation syntaxes. Thus organizations need to hire the expertise for each technology, thus increasing the manpower and decreasing the profit. Sometimes it is the demand of the client for particular technology thus increasing the overheads for the organizations.

Most of the technologies are used only for data manipulation and retrieval. Commonly used data manipulation commands are viz. insert, update, delete , sort etc. These operations are vital part of any software project development. Implementation for theses operations on the database changes as per the type of back end being used. Same code with little syntactical changes is written for these operations causing redundancy of code. Thus to reduce the code redundancy there is need for generic interface to databases which can provide access to any web servers, web browsers and database management systems (DBMS) as per clients requirements. Here a generic interface for data manipulation has been generated. which will provide the common platform for data manipulation for different backend technologies.

Paper is organized as follows: Section 2 puts light on related work in the domain, section 3 discusses the market position of different technologies, section 4 describes the proposed generic framework, Section 5 focuses on results and discussions and finally section 6 talks about conclusions and future work.

II. Literature Survey and Prior Art

Generic platform development is an emerging research topic . Not much work has been carried out in this domain due to new technologies coming every fortnight. It is difficult to incorporate future technologies. A generic data access layer for .Net called DALC4NET has been discussed by [1] It allows data access from databases like SQL Server, Oracle, MySQL, Microsoft Access, and Microsoft Excel. A Model driven engineering based framework has been proposed by [2]. They have provided information integration through linked open data for systematic deployment and evaluation of different models. They have also discussed the relationship between different models and software tools. A generic framework for modeling physical as well as

logical resources has been proposed by [3]. Standard means have been provided to transfer UML models to real time systems. There are many user interface description languages. a detail analysis of the salient features of these languages has been carried out by [4] to predict where they should be used. A user centric access control approach based on annotation based avccess has been proposed by [5], to help users to select appropriate contacts to grant access to resources for annotating collaborative relationships. [6] has described a generic framework used in the MR based AMIRE (Authoring MixedReality) project. The proposed framework can be used in every 3D based application for easy communication between the objects (game objects). Generic tool for software quality assurance has been developed by [7], which can be used across multiple languages. Problems and challenges faced during development of generic framework to enable design, modeling and simulation of large scale heterogeneous systems is discussed by [8]. Design and implementation of Maltilda-based virtual machines for distributed software development has been discussed in [9]. SimGrid framework which is based on simulation for evaluating cluster, grid and P2P algorithms has been discussed by [10]. It also discusses the key features and benefits of using SimGrid framework.

III. Technology in Market

A report by W3Techs shows usage statistics and market share data of PHP on web. According to their survey, PHP version 5 has been used by 81.9% of all the websites for server-side programming.[11]. Fig. 1 shows the historical trend in the percentage of websites using PHP
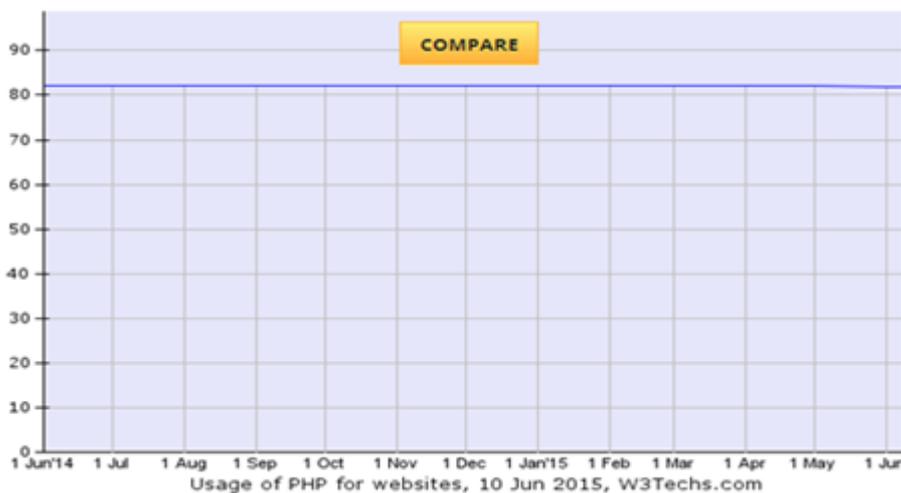
Fig. 1 Historical Trend of Websites using PHP

IV. Market position

Fig. 2 shows the market position of PHP in terms of popularity and traffic compared to the most popular server-side programming languages.Our dedicated market survey shows more server-side languages market data.

Fig. 2 Market Position of Open Source Technologies

V.   Proposed Method

5.1 Generic Application Framework

In the current work, the authors have designed and implemented a generic framework for automatic generation of data manipulation commands for variety of web servers, web browsers and database management systems (DBMS). All the configuration information pertaining to backend DBMS information, technology information, validation information is stored in an XML file which is parsed using Microsoft XML Parser.The technology information involves persentation tier technology used, web server and web browser employed etc., The end user can select among one of the presentation tier technologies JSP, Servlet or PHP. The choice of the web server and web browser can be one of Tomcat, JBOSS or XAMP and Mozilla FireFox, Internet Explorer or Google Chrome, respectively. Presently, we have explored only two database management systems, viz., MS-Access and MySql. The user can either switch off validation tests or switch on client-side validations. Server-side validations are currently not incorporated. The first iteration of the prototype only supports 'RequiredField' and 'Range' validation criteria. The HTML file is parsed using text parser to locate control information. ResultSetMetaData class available in java.sql package is employed for locating column relatd information such as column name and column type. The mapping information is generated on fly and stored in mapping.xml file. To set up the runtime environment a batch file is generated dynamically in each case. Fig. 3 reveals a framework for auto generation of data manipulation commands. The corresponding control flow logic is depicted in Fig. 4 a)-Fig. 4 d).
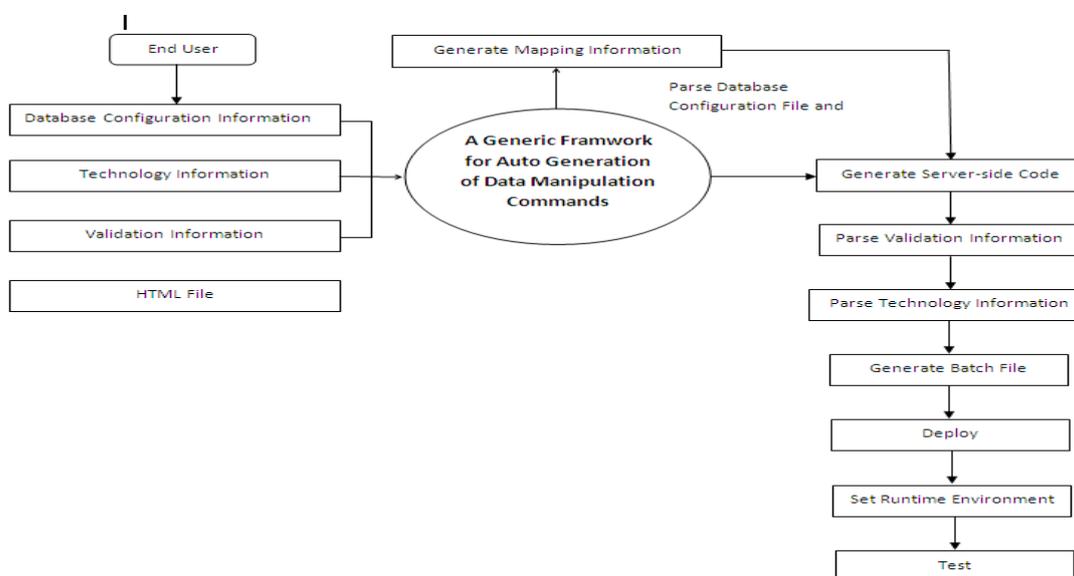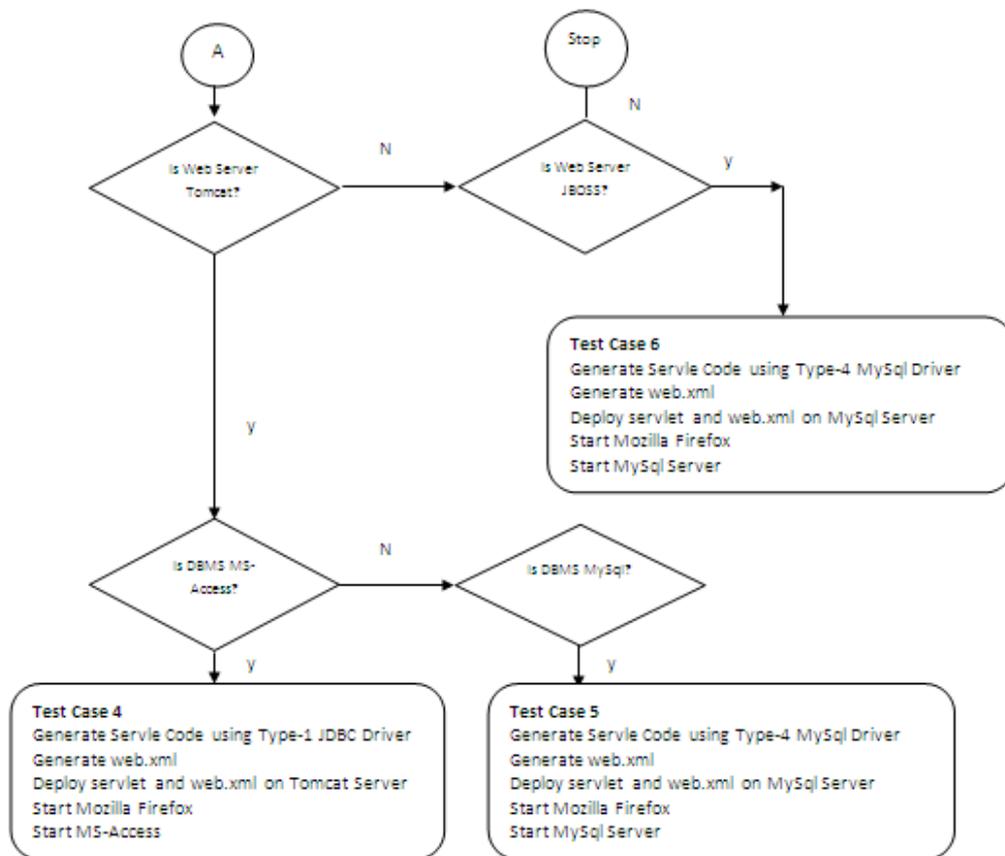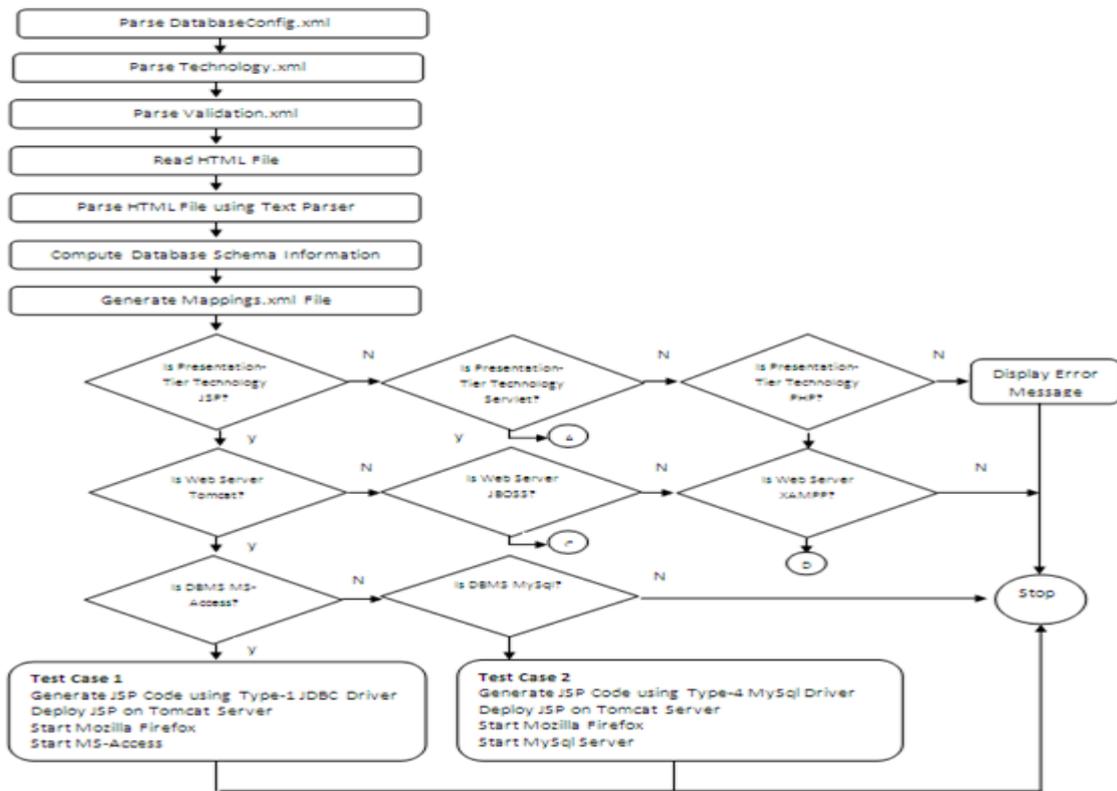

Fig.3 Generic Framework for Auto Generation of Data Manipulation Commands
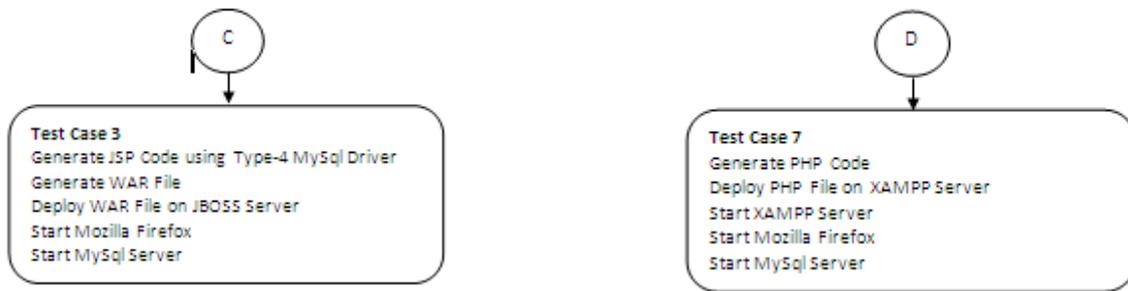
5.2 Control Flow Logic

Fig. 4 a)-d) Control Flow Logic for the Generic Framework

5.3 Pre-requisits for Rendering the Tool Operable
1. HTML file containing a Form element with proper action attribute pointing to the server resource processing form's data and the requisit form controls.
2. Database configuration file containing the back end DBMS related information with the following structure and Document Type Definition. The tags common to all the DBMS are displayed in bold.

Structure of databaseconfig.xml file	DTD of databaseconfig.xml file
<databaseInfo>	<?XML version="1.0" ?>
<type>MySql</type>	<!ELEMENT databaseInfo (type database table dsnameinstallDirectorydriverClassNamedrive rClassPath authentication)>
<database>library</database>	
<table>student</table>	<!ELEMENT type (#PCDATA)>
<dsnname></dsnname>	<!ELEMENT database (#PCDATA)>
<installDirectory>C:\Program Files (x86)\MySQL\MySQL Server 4.1\bin</installDirectory><driverClassName>com.mysql.jdbc.Driver</driverClassName>	<!ELEMENT table (#PCDATA)>
	<!ELEMENT dsnname (#PCDATA)>
	<!ELEMENT installDirectory (#PCDATA)>
<driverClassPath>c:\genericframework\mysql-connector-java-5.1.15-bin.jar</driverClassPath>	<!ELEMENT driverClassName (#PCDATA)>
<authentication>	<!ELEMENT driverClassPath (#PCDATA)>
<username>root</username>	<!ELEMENT authentication (username password hoststring)>
<password>pgn</password>	
<hoststring></hoststring>	<!ELEMENT username (#PCDATA)>
</authentication>	<!ELEMENT password (#PCDATA)>
</databaseInfo>	<!ELEMENT hoststring (#PCDATA)>

The following table depicts the tag names and values specific to particular DBMS system.

DBMS	Tag Name	Tag Value
MS-Access	<database>	library
	<dsnname>	library
MySql	<database>	library
	<driverClassPath>	c:\genericframework\mysql-connector-java-5.1.15-bin.jar
	<username>	root
	<password>	pgn
Oracle	<driverClassPath>	
	<username>	exam01
	<password>	exam01
	<hoststring>	Orcl

3. Technology configuration file containing description of presentation tier technology used in the application development with the following structure and Document Type Definition.

Structure of technology.xml file	DTD of technology.xml file
<technology>	<?XML version="1.0" ?>
<frontendTechnology>	<!ELEMENT technology (frontendTechnology webserver webbrowser)>
<name>servlet</name>	
<projectFolder>test</projectFolder>	<!ELEMENTfrontedndTechnology(name projectFolderfileNamehtmlFilename)>
<fileName>ProcessStudent.java</fileName>	<!ELEMENT name (#PCDATA)>
<htmlFilename>processStudent.html</htmlFilename>	<!ELEMENT projectFolder (#PCDATA)>
</frontendTechnology>	

<webserver> <webservername>tomcat</webservername> <installDirectory>C:\Program Files (x86)\Apache Software Foundation\Tomcat 6.0</installDirectory> <exeFilename>Tomcat6.exe</exeFilename> </webserver> <webbrowser>IExplore</webbrowser> </technology>	<!ELEMENT fileName (#PCDATA)> <!ELEMENT htmlFilename (#PCDATA)> <!ELEMENTwebserver(webservernameinstallDirectoryexeFilename)> <!ELEMENTwebserver name (#PCDATA)> <!ELEMENT installDirectory (#PCDATA)> <!ELEMENT exeFilename (#PCDATA)> <!ELEMENT webbrowser (#PCDATA)>

The following table depicts the real-time configuration information for various combinations of presentation tier technology used and web server / application server.

Presentation Tier Technology	Web Server/Application Server	Format of technology.xml
JSP	Tomcat	<technology> <frontendTechnology> <name>jsp</name> <projectFolder>test</projectFolder> <fileName>processStudent.jsp</fileName> <htmlFilename>processStudent.html</htmlFilename> </frontendTechnology> <webserver> <webservername>tomcat</webservername> <installDirectory>C:\Program Files (x86)\Apache Software Foundation\Tomcat 6.0</installDirectory> <exeFilename>Tomcat6.exe</exeFilename> </webserver> <webbrowser>Firefox</webbrowser> </technology>
JSP	JBOSS	<technology> <frontendTechnology> <name>jsp</name> <projectFolder>test</projectFolder> <fileName>processStudent.jsp</fileName> <htmlFilename>processStudent.html</htmlFilename> </frontendTechnology> <webserver> <webservername>JBoss</webservername> <installDirectory>C:\jboss-4.0.2</installDirectory> <exeFilename>run.bat</exeFilename> </webserver> <webbrowser>Firefox</webbrowser> </technology>
Servlet	Tomcat	<technology> <frontendTechnology> <name>servlet</name> <projectFolder>test</projectFolder> <fileName>ProcessStudent.java</fileName> <htmlFilename>processStudent.html</htmlFilename> </frontendTechnology> <webserver> <webservername>tomcat</webservername> <installDirectory>C:\Program Files (x86)\Apache Software Foundation\Tomcat 6.0</installDirectory> <exeFilename>Tomcat6.exe</exeFilename> </webserver> <webbrowser>Firefox</webbrowser> </technology>
Servlet	JBOSS	<technology> <frontendTechnology> <name>servlet</name> <projectFolder>test</projectFolder> <fileName>ProcessStudent.java</fileName> <htmlFilename>processStudent.html</htmlFilename> </frontendTechnology> <webserver> <webservername>JBoss</webservername> <installDirectory>C:\jboss-4.0.2</installDirectory> <exeFilename>run.bat</exeFilename> </webserver> <webbrowser>Firefox</webbrowser> </technology>
PHP	XAMPP	<technology>

		`<frontendTechnology>`
		 `<name>php</name>`
		 `<projectFolder>test</projectFolder>`
		 `<fileName>processStudent.php</fileName>`
		 `<htmlFilename>processStudent.html</htmlFilename>`
		 `</frontendTechnology>`
		 `<webserver>`
		 `<webservername>xampp</webservername>`
		 `<installDirectory>C:\xampp</installDirectory>`
		 `<exeFilename>xampp-control.exe</exeFilename>`
		 `</webserver>`
		 `<webbrowser>Firefox</webbrowser>`
		 `</technology>`

4. Validation configuration file containing the type of the validation employed and validation rules with the following structure and Document Type Definition.

Structure of validation.xml file	DTD of validation.xml file

`<validation>`
 `<type>clientside</type>`
 `<control>`
 `<name>rollno</name>`
 `<validationRules>`
 `<validationRule chars="any">RequiredField</validationRule>`
 `</validationRules>`
 `</control>`
 `<control>`
 `<name>name1</name>`
 `<validationRules>`
 `<validationRule chars="alpha">`
 `RequiredField`
 `</validationRule>`
 `</validationRules>`
 `</control>`
 `</validation>`
	`<?XML version="1.0" ?>`
	 `<!ELEMENT validation (type control+)>`
	 `<!ELEMENT type (#PCDATA)>`
	 `<!ELEMENTcontrol(name validationrules)>`
	 `<!ELEMENT name (#PCDATA)>`
	 `<!ELEMENTvalidationrules(validationrule+)>`
	 `<!ELEMENTvalidationrule (#PCDATA)>`
	 `<!ATTLIST name CDATA #REQUIRED>`

The structure and Document Type Definition of mappings.xml file generated by the tool is shown below:

Structure of mappings.xml file	DTD of mapping.xml file

`<mappings>`
 `<mapping>`
 `<controlName>rollno</controlName>`
 `<columnName>rollno</columnName>`
 `</mapping>`
 `<mapping>`
 `<controlName>name1</controlName>`
 `<columnName>name1</columnName>`
 `</mapping>`
 `</mappings>`
	`<?XML version="1.0" ?>`
	 `<!ELEMENT mappings (mapping+)>`
	 `<!ELEMENT mapping (conrolNamecolumnName)>`
	 `<!ELEMENT controlName (#PCDATA)>`
	 `<!ELEMENT columnName (#PCDATA)>`

5. MS-Access or MySql database containing the requisit tables.

VI.   Results and discussions

The model presented above is implemented in VB6. The HTML file contains the controls for populating student table. The structure of HTML file and student table are depicted in Fig. 5 a) and Fig. 5 (b), respectively. The GUI for the auto generation of data manipulation commands is depicted in Fig. 6. The user can automatically generate a mapping document which maps controls in HTML file to the corresponding columns in the table. The mapping can be performed either on one-to-one basis or with a single command if and only if there is one-to-one correspondence between control names and column names. The GUI for creation of mapping document is depicted in Fig. 7. All the XML configuration files are parsed using Microsoft XML parser. Currently, only the client-side validation code is incorporated with only few validation rules for required field validation and range validation.

Fig. 5 a) Structure of HTML File

Fig. 5 b) Structure of Student Table
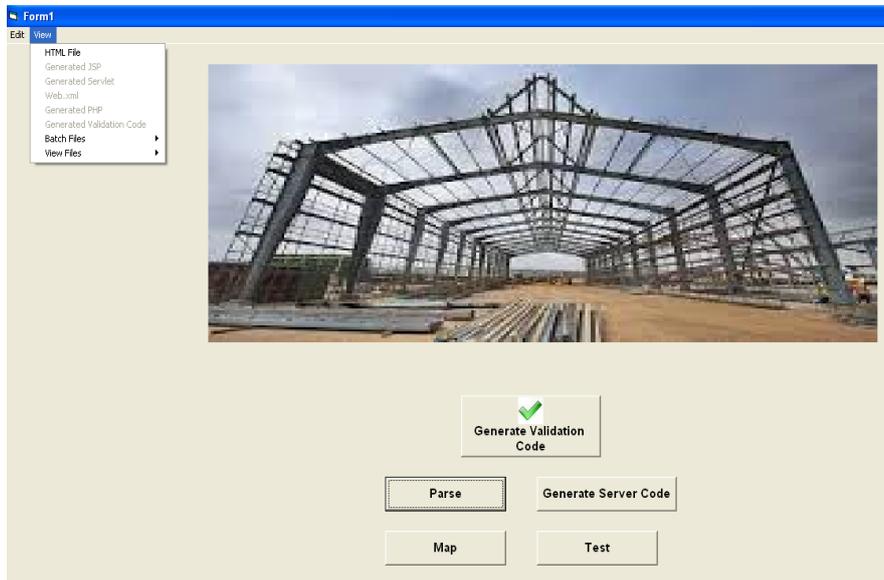
Fig. 6. GUI for the Auto Generation of Data Manipulation Commands
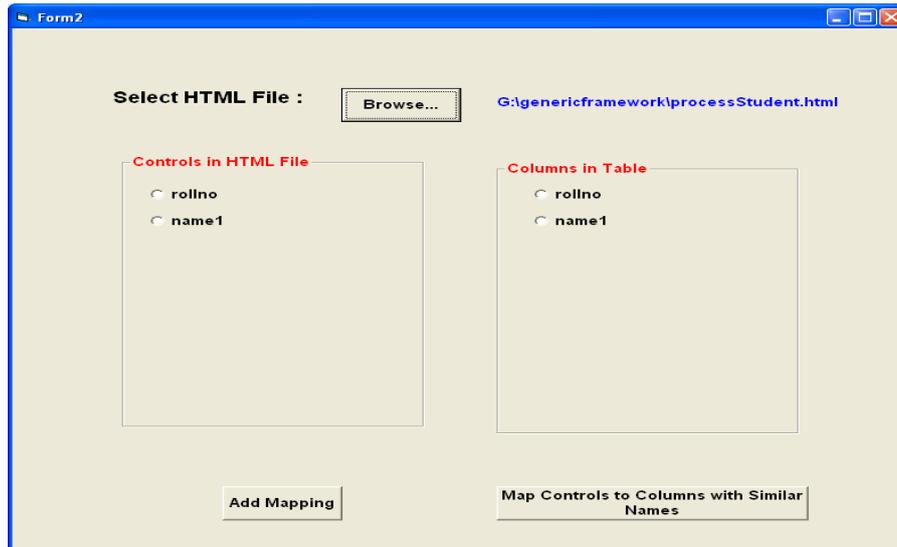
Fig. 7. GUI for the Generation of Mapping Document

The user can edit any of the XML configuration files by selecting Edit option from the main menu. Fig. 8 depicts editing of database configuration information by selecting Ed`it->Database Configuration Info from the main menu.

Fig. 8. Edit window for Configuration Files

The user can view all generated files by selecting view option from the main menu. The code generated for JSP,Servlet and PHP is shown in Fig. 9.

processStudent.jsp

```
<%@ page import="java.sql.*" %>
<%
int rollno = Integer.parseInt(request.getParameter("rollno"));
String name1 = request.getParameter("name1");
String oper = request.getParameter("oper");
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/library","root","pgn");
if (oper.equals("Insert")) {
String query="INSERT INTO student VALUES (?
?)";
PreparedStatement st=con.prepareStatement(query);
st.setInt(1
rollno);
st.setString(2
name1);
st.executeUpdate();
out.println("<font color=blue><h2>Record Inserted successfully...</h2></font>");
st.close();
}
if (oper.equals("Delete")) {
String query1="DELETE FROM student WHERE rollno =?";
PreparedStatement st1=con.prepareStatement(query1);
st1.setInt(1
```

ProcessStudent.java

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;
public class ProcessStudent extends HttpServlet
{
PrintWriter out;
public void doGet(HttpServletRequest request
HttpServletResponse res) throws IOException
ServletException
{
try
{
res.setContentType("text/html");
out = res.getWriter();
int rollno = Integer.parseInt(request.getParameter("rollno"));
String name1 = request.getParameter("name1");
String oper = request.getParameter("oper");
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/library","root","pgn");
if (oper.equals("Insert")) {
String query="INSERT INTO student VALUES (?
?)";
```
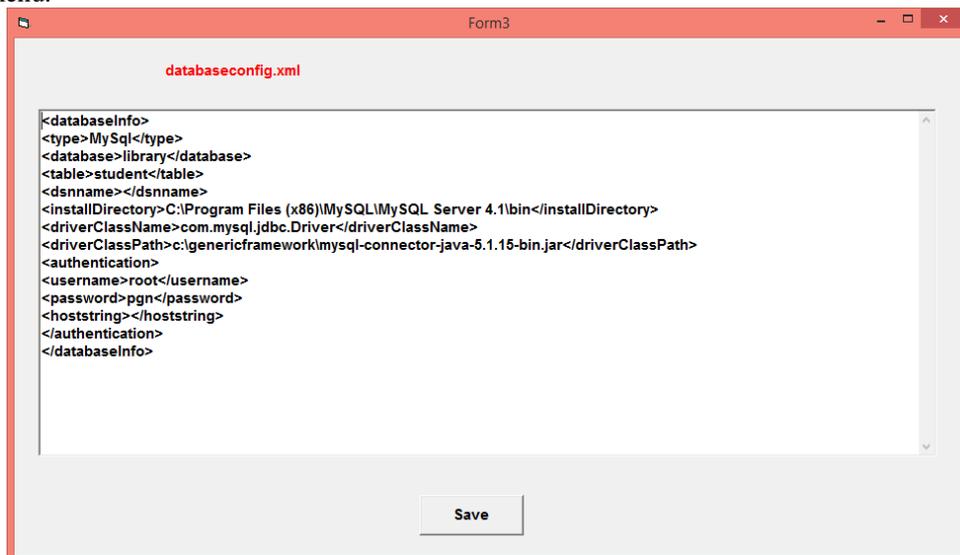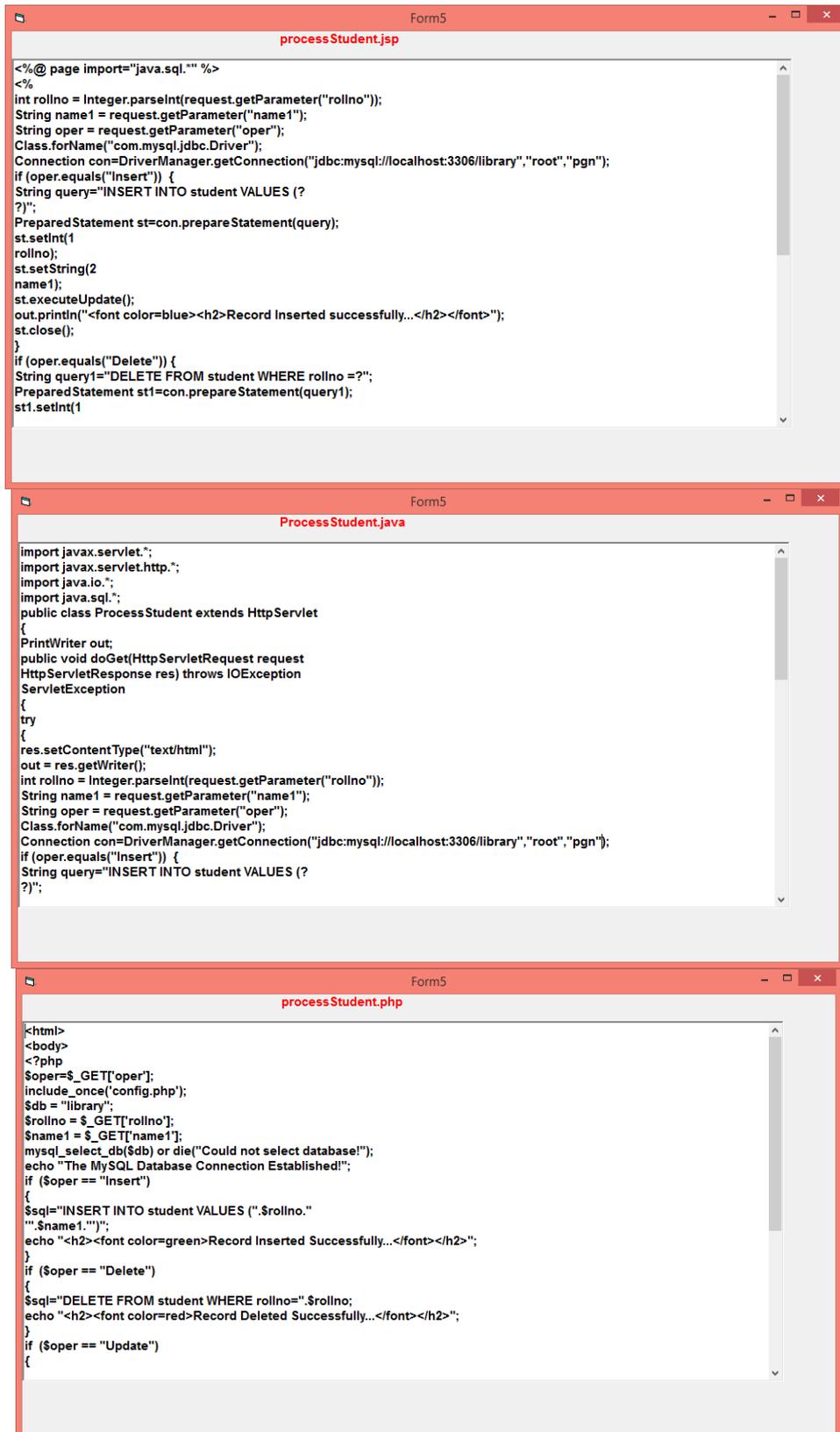
processStudent.php

```
<html>
<body>
<?php
$oper=$_GET['oper'];
include_once('config.php');
$db = "library";
$rollno = $_GET['rollno'];
$name1 = $_GET['name1'];
mysql_select_db($db) or die("Could not select database!");
echo "The MySQL Database Connection Established!";
if  ($oper == "Insert")
{
$sql="INSERT INTO student VALUES (".$rollno."
'".$name1."')";
echo "<h2><font color=green>Record Inserted Successfully...</font></h2>";
}
if  ($oper == "Delete")
{
$sql="DELETE FROM student WHERE rollno=".$rollno;
echo "<h2><font color=red>Record Deleted Successfully...</font></h2>";
}
if  ($oper == "Update")
{
```

Fig 9. Code Generated for JSP, Servlet and PHP

The seven test cases generated are depicted in the tree structure shown in Fig 10.
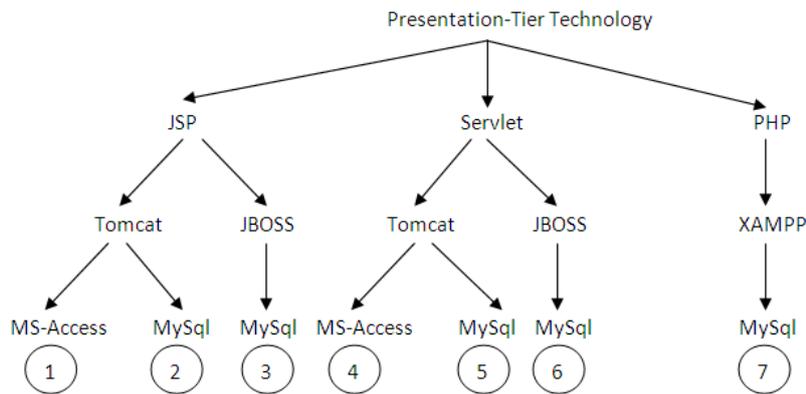
Fig 10. Tree Structure for Generation of Test Cases

Test Case 1: JSP technology with MS-Access as backend, deployed on Tomcat server tested using Mozilla Firefox. Runtime environment setting involves
i) Installation of Tomcat server
ii) Installation of MS-Office
iii) Installation of Mozilla Firefox.
Form element of HTML file has the structure shown below:
<form name="form1" action="http://localhost:8080/test/processStudent.jsp" method="get">
Deployment : The JSP page is deployed in a project folder specified by the user in one of the configuration files.
Format of run.bat file generated is shown below:

```
copy vprocessStudent.html "C:\Program Files (x86)\Apache Software Foundation\Tomcat
6.0\webapps\ROOT\test\processStudent.html"
 cd C:\Program Files (x86)\Mozilla Firefox
 start firefox.exe http://localhost:8080/test/processStudent.html
cd C:\Program Files (x86)\Microsoft Office\Office12
start MSACCESS.EXE C:\genericframework\library.mdb
cd c:\genericframework
copy viewstudent.jsp "C:\Program Files (x86)\Apache Software Foundation\Tomcat 6.0\webapps\ROOT\test"
copy processStudent.jsp "C:\Program Files (x86)\Apache Software Foundation\Tomcat 6.0\webapps\ROOT\test"
cd C:\Program Files (x86)\Apache Software Foundation\Tomcat 6.0
cd bin
start Tomcat6.exe
pause
```

Similar batch files are generated in other test cases.
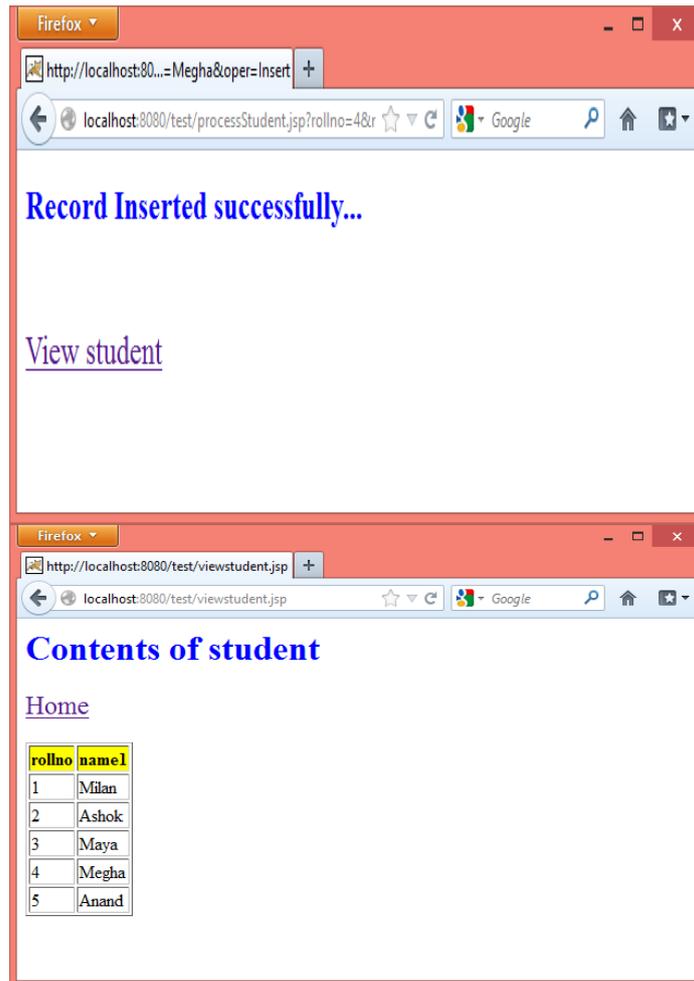The output generated is shown in Fig. 11(a)-(c).
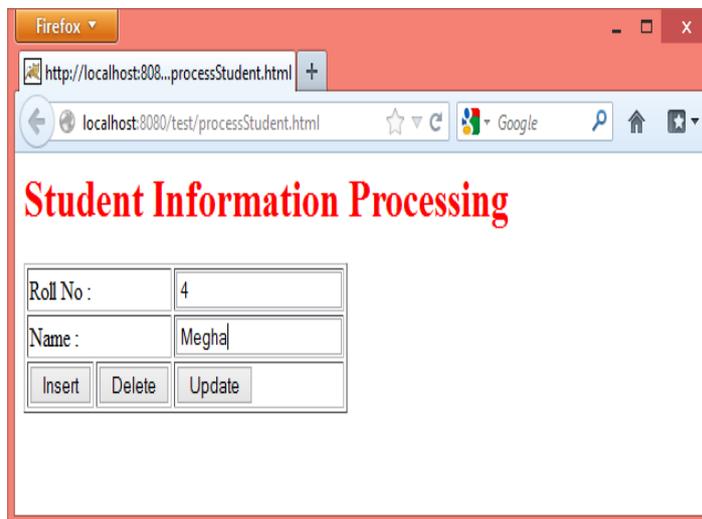
Fig. 11a)-c). Execution of Server-side code.

Test Case 2 : JSP technology with MySql as backend, deployed on Tomcat server tested using Mozilla Firefox. Runtime environment setting involves
i)   Installation of Tomcat server
ii)   Installation of MySql
iii)   Installation of JDBC driver for MySql
iv)   Installation of Mozilla Firefox.

Form element of HTML file has the structure shown below:

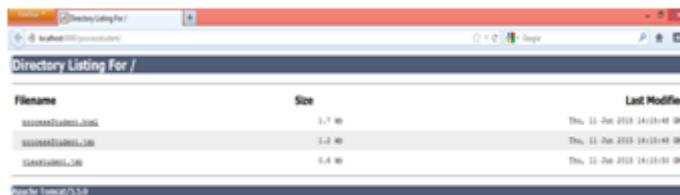<form name="form1" action="http://localhost:8080/test/processStudent.jsp" method="get">

Test Case 3: JSP technology with MySql as backend, deployed on JBOSS server tested using Mozilla Firefox. Runtime environment setting involves
i)   Installation of JBOSS server
ii)   Installation of MySql
iii)   Installation of JDBC driver for MySql
iv)   Installation of Mozilla Firefox.

Form element of HTML file has the structure shown below:

<form name="form1" action="http://localhost:8080/processstudent/processStudent.jsp" method="get">

Where processsstudent is the name of the WAR file deployed in JBOSS_HOME/server/default/deploy folder of JBOSS.

Test Case 4: Servlet technology with MySql as backend, deployed on Tomcat server tested using Mozilla Firefox. Runtime environment is same as above.

Form element of HTML file has the structure shown below:

<form name="form1" action="http://localhost:8080/examples/ProcessStudent" method="get">

Test Case 5 : Servlet technology with MS-Access as backend, deployed on Tomcat server tested using Mozilla Firefox. Runtime environment is same as above.

Form element of HTML file has the structure shown below:

<form name="form1" action="http://localhost:8080/examples/ProcessStudent" method="get">

Deployment : The servlet class is deployed in a /examples/WEB-INF/classes folder and the deployment descriptor for the servlet, web.xml is automatically generated and stored in /examples/WEB-INF folder.

Test Case 6: Servlet technology with MySql as backend, deployed on JBOSS server tested using Mozilla Firefox. Runtime environment is same as above.

Form element of HTML file has the structure shown below:

<form name="form1" action="http://localhost:8080/processstudent/ProcessStudent" method="get">

Test Case 7: PHP technology with MySql as backend, deployed on XAMPP server tested using Mozilla Firefox. Runtime environment setting involves

i)   Installation of XAMPP server
ii) Installation of MySql
iii) Installation of Mozilla Firefox.

Form element of HTML file has the structure shown below:

<form name="form1" action="http://localhost/processStudent.php" method="get">

Format of run.bat file generated is shown below:

The sample partial parse code employed for parsing XML files is enclosed in Appendix A. The validation code generated is enclosed in Appendix B along with GUI.

VII.   Conclusion and Scope for Future Work

In today's competitive era, the software development community is inclining towards open source technology owing to its incomparable advantages, with PHP 5.0 capturing highest market share according to a survey conducted by W3Tech. Keeping this scenario in mind authors have designed and implemented a generic framework for partial web application development focusing on maintenance of master tables. The common denominator in various web applications is separated out and the frequenctly changing information is spread out in various XML configuration files. The framework is generic and easily adaptable to new technological advancements. The focus is laid on common tools which distinguish one web application from another such as web server, DBMS employed etc. on the server-side and web browser on the client-side. Seven test cases are generated for thorough testing of the tool. All the generated code can be viewed by an end user, as such the tool serves as first hand tutor for the user interested in learning the basics of web technologies.

Our future work focuses on expanding validation rule base to incorporate regular expression editor and custom validation. Another area which is overlooked by us is server-side validation. We plan to incorporate this feature by dynamically generating the validation class on the server-side which can be employed by other classes/pages. Also, in future, we intend to render the tool intelligent enough for automatic selection of advanced technologies such as AJAX, JSON based on the task under consideration and enable the user to choose between 2-tier and 3-tier web applications.

References

[1].   http://www.codeproject.com/Articles/158042/DALC-NET-A-Generic-Data-Access-Layer-for-NET
[2].   Ivan Ruiz-Rube , Juan Manuel Dodero, Ricardo Colomo-Palacios, A framework for software process deployment and evaluation, Information and Software Technology 59 (2015) 205–221, Elsevier
[3].   Selic, B, A generic framework for modeling resources with UML, Computer, volume:33, Issue: 6, Page no. 64 – 69, 06 August 2002, IEEE

[4].   Guerrero-Garcia, J.; Univ. Catholique de Louvain, Louvain-la-Neuve, Belgium ; Gonzalez-Calleros, J.M.; Vanderdonckt, J.; Muoz-Arteaga, J., A Theoretical Survey of User Interface Description Languages: Preliminary Results, Web Congress, 2009. LA-WEB '09. Latin American, IEEE.

[5].   PeymanNasirifard,, VassiliosPeristeras, Stefan Decker, Annotation-based access control for collaborative information spaces, Computers in Human Behavior, Volume 27, Issue 4, July 2011, Pages 1352–1364, Elsevier Science Publishers.

[6].   Haller M., Hartmann W., Zauner J., A generic framework for game development, ACM/SIGGRAPH and Eurographics Campfire "Production Process of 3D Computer Graphics Applications – Structures, Roles and Tools", Snowbird (Utah), 2002.

[7].   Darryl Owens and Mark Anderson, A Generic Framework for Automated Quality Assurance of Software Models Supporting Languages of Multiple Paradigms, Science and Information Conference (SAI), 2013, pg no. 2191- 2196. , IEEE

[8].   JiEun Kim, Daniel Mosse, Generic Framework for Design, Modeling and Simulation of Cyber Physical Systems, ACM SIGBED Review - Special issue on the RTSS forum on deeply embedded real-time computing, Volume 5 Issue 1, January 2008, Article No. 1 , ACM New York, NY, USA

[9].   Hiroshi Wada, Junichi Suzuki, Adam Malinowski and Katsuya Oba, Matilda: A Generic and Customizable Framework for Direct Model Execution in Model-Driven Software Development, Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalisation, Chapter: 17, Publisher: IGI Global, Editors: M. Ramachandran, R. A. de Carvalho, pp.250 - 279

[10]. Henri Casanova, Arnaud Legrand, Martin Quinson. SimGrid: a Generic Framework for Large-Scale Distributed Experiments. 10th IEEE International Conference on Computer Modeling and Simulation - EUROSIM / UKSIM 2008, Apr 2008, Cambrige, United Kingdom. IEEE, 2008. <inria-00260697>

[11]. http://w3techs.com/technologies/details/pl-php/all/all