A Fast Software Project Development Tool with Generic XML Generation

Sunil D Rathod
PhD Research Scholar,
Computer Engineering Dept.
JJT University, Jhunjhunu
Rajasthan, India

S D Joshi, PhD
PhD Research Guide,
Computer Engineering Dept.
BharatiVidyapeeth University
College of Engineering
Katraj, Pune, India

ABSTRACT

The Integrated Development Environment (IDE) based tools are popular and madethe task of project development easier and comparably faster. These tools help developer to design Graphical User Interface (GUI) just by using Drag and Drop (DND) tool box with specification of the properties.Some of these tools are .NET framework, Windows Builder, etc to design the forms. But, none of these utilities embed “Business Logic” (BL) automatically in code. In routine development process it becomes very time consuming to do the repeated task of coding for same events. There are certain vendor specific tools like Oracle Application Express (OAE) from ORACLE and some similar tools from IBM mainframe to design the product and do analysis and reporting but these tools are too much vendor specific and strictly need the proprietary DB to develop the project. For instant, OAE needs strictly ORACLE as a backend for its development.Our tool - “Rapid Project Builder” (RPB) is used to avoid time consumed for coding the same business logic *repeatedly*. The tool will also performAutomatic Code Generation (ACG) in specific language of developer’s requirement. The developer can design the application’s frontend GUI using RBP with DND along with the specification of form and fields. After adding control like add, delete, search or modify buttons to the application, RPB adds the BL automatically to these controls along with complete language specific code.The automatic generation of backend DBMS is done with the help of specification given as the properties in the form design which is stored as XML semantics**.**

Keywords

IDE, GUI, DND, RPB, ACG, BL, ADC, OAE.

1. INTRODUCTION

Software Industries always face new challenges in development of new software products with very dynamic changing demandsfrom clients. The sole efforts of project development encircles around cost reduction, efficiency and reliability of product to be developed. The various tool and techniques are available with global standardization to help developers to improve the quality of product along with reduced cost while going through entire SDLC processes12],[13]. Various efforts have been made consistently on different fronts by OMG with its group to make these processes more generic, platform independent and automatic with respect code generation and transformation.[[16],[17].

The UML specifications focus on design of the software projects just by DND and with forward engineering, the designed classifiers like Class diagram, Use Case, State Chart and Sequence diagrams are converted to code [1],[2. Even though the efforts are being put in the direction of automatic code generation directly from the UML classifiers, none of the approaches is able to embed the business logic in the code],[5],[9]. The code generated from the given classifier in UML is just a code template with class, its attributes with data type of each attribute and methods with just prototype declaration without any actual code in it[10][19].

The BL code is explicitly written by developer as an event firing using different logic for operations like Addition, Change , Insert and Delete (ACID)on date stored in Database (DB) tables, depending on the domain specifications and constraintsof BL[24] [25][33].The connectivity of front end with DB is another issue to be handled by developer of the projects for which he needs to code the logic. Thus the entire project development will take a lot of time in addressing the issues like code for GUI or GUI using DND, Code for BL on different control likes text box, command buttons, check box with different mouse click or key press events, and DB creation with ODBC connectivity issues.

Our tool Rapid Project Builder(RPB) addressall these issues together just by designing the project UI using DND along with the specifications of data type, the events like click, key press or select, etc of the controls. The RPB will store the information given by developer as Meta data in XML standard form and will automatically generate code, DB tables, along with automatic addition of BL in the code.

The entire paper organization explores the capabilities of RPB. The next section is literature review/related work; subsequent chapters will cover the architecture of RPB, Methodologies used in achieving ACG, ADC and automatic embedding of BL in language specific code. The XML Meta file creation is main focus of the paper. Our work is concluded with some experimental results along with conclusion and future scope of the RPB tool.

2. RELATED WORK

Since many decades, the researchers have been striving hard to make software development process as easy and fast as possible without losing the robustness, efficiency and cost effectiveness of the product. Due to dynamic changes in requirement of the product it is very much essential to adapt the methods and methodologies with different paradigm to meet the challenges of making software product more generic, platform independent, portable, scalable and robust with less cost of development[12],[13]. To meet all these challenges OMG’sMDA approach is very suitable and is adopted at large in many industries as standard practice [17]. Various issues like rapid project development with reusability of code and other resources has already been simplified with the evolution of Object Oriented Programming (OOP) technologies which is again a part of UML standards [12],[24],[25].

The UML designing tools like IBM Rational Rose, free tools like star UML, and the like have been using many evolutionary techniques to generate the code automatically just by using state chart and class diagrams as classifiers.[1][6].

The author in their approach have even tried to extend the functionalities of existing UML tools by creating new tool called UJECTOR (UML to Java Executable Code generaTOR) forautomatic generation of executableJava code from UML diagrams. If aset of three UML classifiers like class diagram, sequencediagrams and activity diagrams are given as input to the tool; it automatically generates completely executable Javacode. UJECTOR builds Javaobject-oriented code class structure from class diagram, methods’flow of control through sequence diagrams, and object manipulations are includedby activity diagrams [3].

In [7] the authors have describeda program that allowsthe user or programmer to specify his/her code using an easy-to-understand, simple-to-write and more or less unchangeable pseudo code specification. The program checkserror in pseudocode and converts the algorithmic specification to a specified language like C, Java, or any other language.

The concrete system for automatic code generation, DB table creation, and some BL embedding in the code is seen in [23].The tool is developed in .NET framework and can be only used with only Microsoft platform.

The giant vendor Microsoft has VB .Net framework which uses DND approach for UI design with little code generation as templates but without BL and DB tablecreation[33].

The ORACLE Financial Management Analytics (OFMA)is designed with the intension of report generation as main concern but it too lags in generating source code of specific language and no BL embedding in code [30].

The similar features are available with eclipse Windows Builder, Microsoft Dynamics Customer Relationship Management (CRM)[31],[32].

Our tool RPB is intended to reduce the time and cost of design phase and coding phase (frontend as well as backend code generation) of Software Development Life Cycle (SDLC).

The RPB is an evolutionary approach towards reduction in the cost and time of development of software product. As a first step of development of this tool a concrete prototype is being developed for standalone applications with backend support but later on this tool can be extended for web based application development.

3. RPB ARCHITECTURE

The RPB has architectural flow as shown in figure below.

The RPB tool has an IDE for developer so that developer can design and develop the desired application. The tool provides similar facilities as that of VB .Net or Net beans or the like and developer can easily design UI. The developer can either create new application or open an existing one to customize it.The various components of RPB are described below in this section.

3.1 GUI Form designer

This is an editor which consists ofvarious panels like project explorer, property window, and tool box with controls. It helps developer to perform DND to design application form along with the specification of properties of each of the controls used on the form.

3.2 XML Metafile Generation

The file contains all the Meta information in XML format which is a useful for automatic code generation, DB table creation and embedding of Business Logic in code automatically.

3.3 Language Library

It is library of various languages like C++, JAVA, .NET, etc with its key words, syntax, semantics and different structures. This library basically helps RPB to take specification from XML and convert the GUI into working code in the specific language.

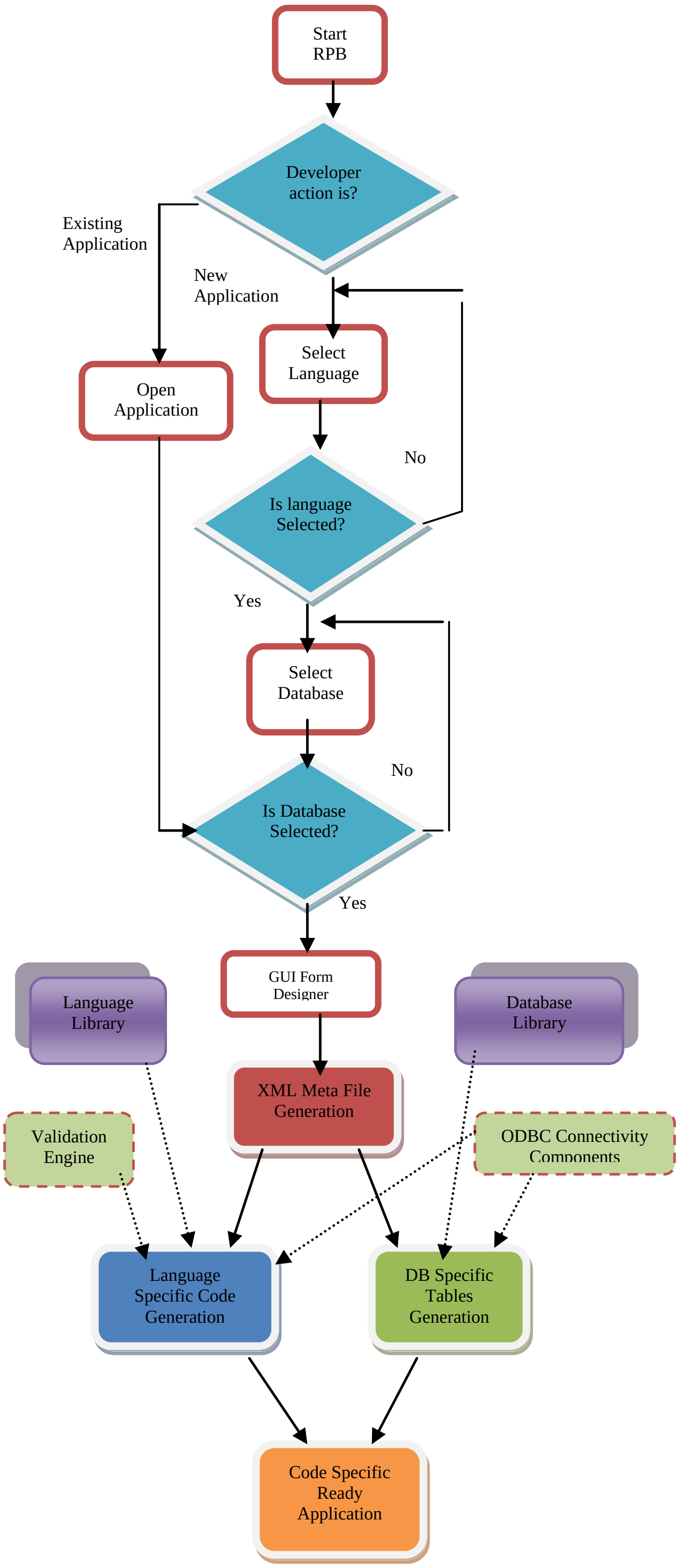


Fig. 1. RPB Architecture with Flow

3.4 Database Library

It is Database library of ACCESS, MySQL, ORACLE, etc with its DDL, DML, DCL commands, syntax and semantics. It is invoked by RPB convert the GUI specification into Database tables.

3.5 ODBC Connectivity Component

This component creates interface between front end and back end of the application.

3.6 Language Specific Code Generation

This component is module of RPB for creating front end of the application

3.7 Database Specific Table Generation

This component is module of RPB for creating back end of the application.

3.8 Validation Engine

This component is use for form validation. The Validation engine will automatically make the validation of the text field values of the frontend UI forms.

RPB also helps developer to even customize the code and can make the application redesigned as per the new requirements.

4. methodologies

The methodologies used to design and develop RPB are discussed in this section.

4.1 Form Creation using DND

The Form can be created using DND tool box. The property window allows developer to specify the data type of each field so that is used for creation of CLASS with attributes and corresponding fields for DB table. This is depicted in figure-2 below.

The controls like Label, Text Box, Buttons, etc will have separate property window wherein the properties for each of these control can be set.

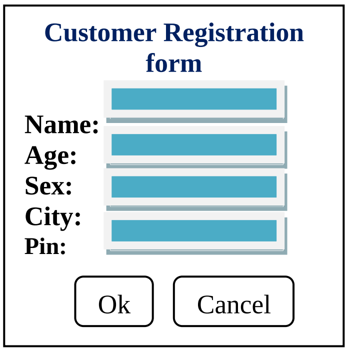


Fig. 2. Form creation using DND of RPB

4.2 Conversion of GUI to XML Syntax

Once the form is creation designed using RPB, it automatically generate XML Meta file and also the DB tables in the DBMS specified by developer. The properties set by developer will be included in XML. The field with its data type will help the code converter module and DB table generator module for creating code and DB tablesrespectively. This is depicted in figure-3.

4.3 Conversion of XML Class to JAVA Class

The Application will then automatically create a JAVA class from the above generated XLM class using JAVA library syntax set file.

The JAVA library syntax set file is having all the relevant language key words, conditional/loop control, class, interfaces, design pattern, etc syntax already stored in it.

The PARSER will scan the XML form file and parse the standard TAGs used by our RPB application development tool. It will transform the XLM TAGs into JAVA class definition as follows:

```
Class Customer

{
            Private String Name;

             Private int Age;

            Private List Sex[]={“Male”, Female”};

             Private String City;

            Private int Pin;


            Public Ok_onClick() {

                        //Business Logic for saving the fields

                         //in DB table is embedded here.

            }

            Public Cancel_onClick() {

                        //Logic for Cancelation of the action.

                        }

}
```

The default language in which RPB generates the code automatically is JAVA but developer can specify any OO language supported by RPB like C++ or C#.

```
<FORM NAME = “CustomerRegistrationForm”>
<CLASS NAME =”Customer”>
< LABEL NAME=”Name” />
<TEXT TYPE= “String”/>

< LABEL NAME=”Age” />
< TEXT TYPE= “Integer”/>

< LABEL NAME=”Sex” />
< TEXT TYPE= “List”  VALUES =”Male, Female” />
< LABEL NAME=”City” />
< TEXT TYPE= “String”/>

< LABEL NAME=”Pin” />
< TEXT TYPE= “Integer”/>
< CBUTTON NAME=”Ok” />
< CBUTTON NAME=”Cancel” />
```

Fig. 3. XML code generated by RPB

4.4 Conversion Process

The conversion from java object to XML and vice versa can beachieved using **J**ava **A**rchitecture for **XML** **B**inding (JAXB). JAXB provides methods for un-marshaling (reading) XML instance documents into Java content trees, and then marshaling (writing) Java content trees back into XML instance documents in very easy way as depicted in figure below.

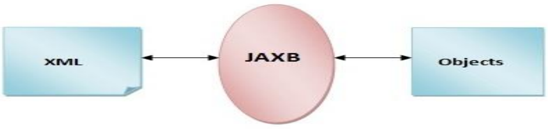


Fig. 4. JAXB conversion process

Alternatively the conversion can even be achieved using SAX or DOM parser in Java.

In our approach we have used the direct conversion method as our project GUI is stored as '.jframe' object and it is manipulated using a special function which is part of RPB.

4.5 Embedding of Business Logic in Code

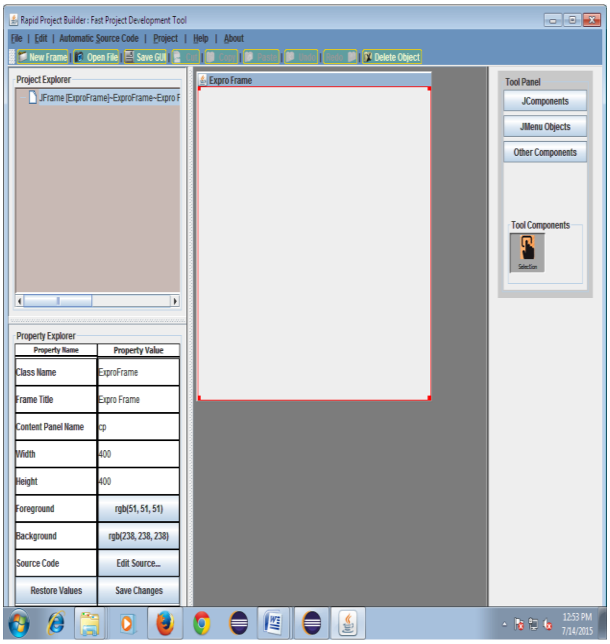
The business logic in code is embedded using words of **business domain vocabulary** from XML Meta file. The Object Data Base Connectivity (ODBC) component is required connect front end and backend DB.

The Commonly used actions like Update, Delete, Add, Ok, Cancel, etc are coded easily just by parsing the corresponding XML and connecting the form fields to the DB tables.

The domain specific actions require the specification of business logic in XMLMeta file so that proper BL can be embedded in code automatically.

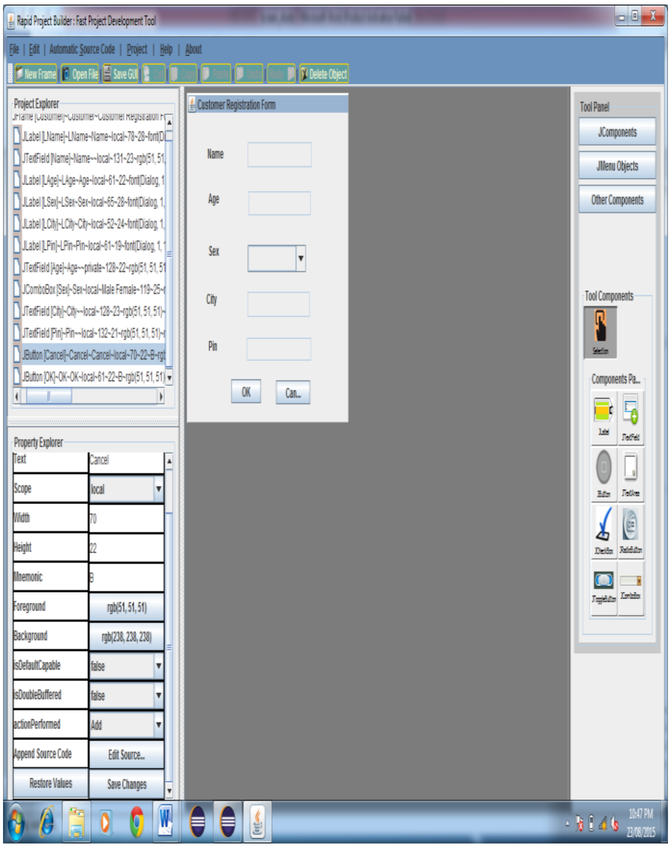
5. EXPERIMANTAL RESULTS

The RPB is designed and developed using JAVA. Being an open source tool and will be made available to open source community. Some of the implementation modules of RPB are shown as snapshots.



Property Name	Property Value
Class Name	ExproFrame
Frame Title	Expro Frame
Content Panel Name	cp
Width	400
Height	400
Foreground	rgb(51, 51, 51)
Background	rgb(238, 238, 238)
Source Code	Edit Source...
Restore Values	Save Changes

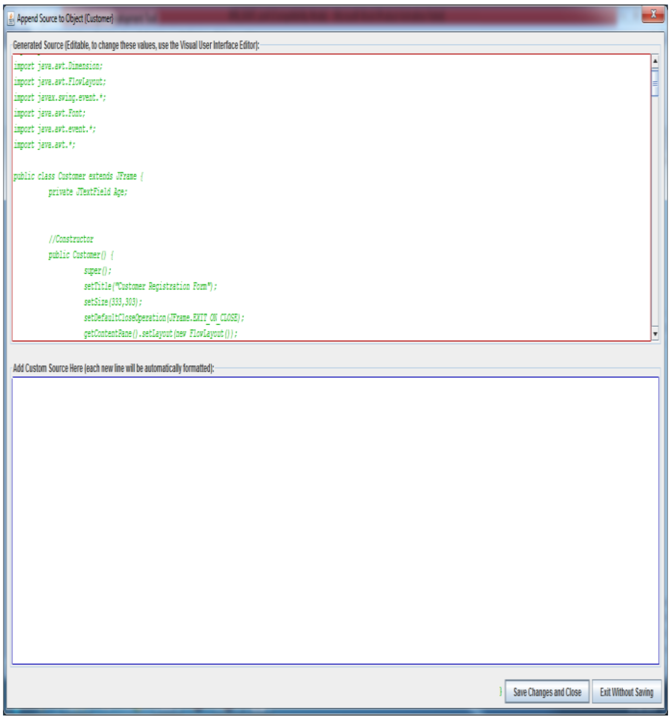
Fig. 5. RPB Editor Environment (Fom Design, Property window& Tool box)



```
JFrame [Customer]~Customer~Customer Registration F
JLabel [LName]~LName~Name~local~78~28~font(D
JTextField [Name]~Name~~local~131~23~rgb(51, 51,
JLabel [LAge]~LAge~Age~local~61~22~font(Dialog, 1
JLabel [LSex]~LSex~Sex~local~65~28~font(Dialog, 1,
JLabel [LCity]~LCity~City~local~52~24~font(Dialog, 1,
JLabel [LPin]~LPin~Pin~local~61~19~font(Dialog, 1, 1
JTextField [Age]~Age~~private~128~22~rgb(51, 51, 51
JComboBox [Sex]~Sex~local~Male Female~119~25~
JTextField [City]~City~~local~128~23~rgb(51, 51, 51)~
JTextField [Pin]~Pin~~local~132~21~rgb(51, 51, 51)~
JButton [Cancel]~Cancel~Cancel~local~70~22~B~rgb
JButton [OK]~OK~OK~local~61~22~B~rgb(51, 51, 51)
```

Text	Cancel
Scope	local
Width	70
Height	22
Mnemonic	B
Foreground	rgb(51, 51, 51)
Background	rgb(238, 238, 238)
isDefaultCapable	false
isDoubleBuffered	false
actionPerformed	Add
Append Source Code	Edit Source...
Restore Values	Save Changes

Fig. 6. RPB Editor Environment (Customer Form)



Generated Source (Editable, to change these values, use the Visual User Interface Editor):

```
import java.awt.Dimension;
import java.awt.FlowLayout;
import javax.swing.event.*;
import java.awt.Font;
import java.awt.event.*;
import java.awt.*;

public class Customer extends JFrame {
        private JTextField Age;


        //Constructor
        public Customer() {
                super();
                setTitle("Customer Registration Form");
                setSize(333,303);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                getContentPane().setLayout(new FlowLayout());
```

Add Custom Source Here (each new line will be automatically formatted):

Fig. 7. RPB Editor Environment (Source code)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<FORM>
<FORM_NAME>Customer Registration Form</FORM_NAME>
<CLASS>
<CLASS_NAME>Customer</CLASS_NAME>
<TEXTFIELD>
<TEXTFIELD_NAME>Name</TEXTFIELD_NAME>
<TEXTFIELD_DATATYPE>STRING</TEXTFIELD_DATATYPE>
<TEXTFIELD_DATASIZE>255BYTE</TEXTFIELD_DATASIZE>
</TEXTFIELD>
<TEXTFIELD>
<TEXTFIELD_NAME>Age</TEXTFIELD_NAME>
<TEXTFIELD_DATATYPE>INTEGER</TEXTFIELD_DATATYPE>
<TEXTFIELD_DATASIZE>1BYTE</TEXTFIELD_DATASIZE>
</TEXTFIELD>
<COMBOBOX>
<COMBOBOX_NAME>Sex</COMBOBOX_NAME>
<COMBOBOX_ITEM>Male Female</COMBOBOX_ITEM>
<COMBOBOX_ITEM>TestItem</COMBOBOX_ITEM>
<COMBOBOX_DATATYPE>STRING</COMBOBOX_DATATYPE>
</COMBOBOX>
<TEXTFIELD>
<TEXTFIELD_NAME>City</TEXTFIELD_NAME>
<TEXTFIELD_DATATYPE>STRING</TEXTFIELD_DATATYPE>
<TEXTFIELD_DATASIZE>255BYTE</TEXTFIELD_DATASIZE>
</TEXTFIELD>
<TEXTFIELD>
<TEXTFIELD_NAME>Pin</TEXTFIELD_NAME>
<TEXTFIELD_DATATYPE>INTEGER</TEXTFIELD_DATATYPE>
<TEXTFIELD_DATASIZE>4BYTE</TEXTFIELD_DATASIZE>
</TEXTFIELD>
<BUTTON>
<BUTTON_NAME>Cancel</BUTTON_NAME>
<BUTTON_MNEMONIC>B</BUTTON_MNEMONIC>
<BUTTON_OPERATION>Add</BUTTON_OPERATION>
</BUTTON>
<BUTTON>
<BUTTON_NAME>OK</BUTTON_NAME>
<BUTTON_MNEMONIC>B</BUTTON_MNEMONIC>
<BUTTON_OPERATION>Add</BUTTON_OPERATION>
</BUTTON>
</CLASS>
</FORM>
```

Fig. 8. RPB Editor Environment (XML of Customer Form)

6. CONCLUSION

As discussed in various section of the paper our work is an innovation towards making the software project development more and more simple, efficient and robust through our tool RPB.The existing systems, vendor specific tools and the work done by various researchers have limitation of designing the software products with the functionality of DND, generation of Code template, data analysis and report generation.

The RPB is capable of generating code automatically in various languages like Java, C++, C#. If more language libraries are added, more will be the language support. At initial level DB support will be only for MS-Access and MySQL and later on more DBs will be added in it.

Our efforts are aimed towards the reduction of software development cost thus contributing towards the simplification of some of the processes of SDLC.

The scope of RPB can be extended for development web application like standalone Project development. As a future development RPB can be added with functionality of Model View and Architecture (MVC) framework so that Web application can be developed using all the features provided for standalone application.

7. REFERENCES

[1] Abdeslam Jakimi and Mohammed Elkoutbi, Automatic Code Generation From UML Statechart, International Journal of Engineering and Technology Vol. 1, No. 2, June, 2009 1793-8236, 165-168.

[2] Design and Implementation of an Autonomic Code Generator based on RTPA, Yingxu Wang, University of Calgary, Canada Xinming Tan, Wuhan University of Technology, China Cyprian F. Ngolah, Sentinel Trending & Diagnostics Ltd., Calgary, Canada, 44 International Journal of Software Science and Computational Intelligence, 2(2), 44-65, April-June 2010I.S. Jacobs and C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[3] Automatic Generation of Java Code from UML Diagrams using UJECTOR, Muhammad Usman, and Aamer Nadeem, International Journal of Software Engineering and Its Applications Vol.3, No.2, April, 2009

[4] Automated Analysis and Code Generation for Domain-Specific Models 2012 Joint Working Conference on Software Architecture & 6th European Conference on Software Architecture, George Edwards ,Yuriy Brun, Nenad Medvidovic.

[5] Generation of Java Code Structure from UML Class Diagram, Prajkta R. Pawde, Vikrant Chole, International Journal of Innovative Science and Modern Engineering (IJISME) ISSN: 2319-6386, Volume-2, Issue-7, June 2014

[6] Visual Composition and Automatic Code Generation for Heterogeneous Components Coordination with Reo. Herve Kabamba Mbikayi, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 1, Issue 8, October 2012

[7] Automatic Algorithm Specification To Source Code Translation, Suvam Mukherjee, Tamal Chakrabarti.

[8] Automatic C Code Generation for Parallel Compilation, Amit Barve & Brijendra Kumar Joshi

[9] Effort Reduction by Automatic Code Generation,Sukhvir Singh, Neeraj Kumar

[10] Transformation Of Uml Sequence Diagram To Java Code, Harshal D. Gurad, Prof. V. S. Mahalle

[11] A Code Generation Framework for Java Component-Based Designs Jeff Tsay, Christopher Hylands, Edward A. Lee.

[12] Automatic code generation by model transformation from sequence diagram of system’s internal behavior, EL BEGGAR Omar, BOUSETTA Brahim,GADI Taoufiq, International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 01– Issue 02, November 2012.

[13] An MDA Method for Automatic Transformation of Models from CIM to PIM, Abdelouahed Kriouile, Najiba Addamssiri, Taoufiq Gadi, American Journal of Software Engineering and Applications, 2015.

[14] An Approach to Code Generation from UML Diagrams, Harshal D. Gurad, V. S. Mahalle, International Journal Of Engineering Sciences & Research Technology.

[15] Model Based Software Development: Issues & Challenges, N Md Jubair Basha,Salman Abdul Moiz & Mohammed Rizwanullah, Special Issue of International Journal of Computer Science & Informatics (IJCSI), ISSN (PRINT) : 2231–5292.

[16] Comparison of Two Model Driven Architecture Approaches for Automating Business Processes, Moskitt Framework and Bizagi Process Management Suite, Oskeol Gjoni, Mediterranean Journal of Social Sciences MCSER Publishing, Rome-Italy, Vol 6 No 2 March 2015

[17] Definition Of A Semantic Platform For Automated, Code Generation Based On Uml Class Diagrams And Dsl Semantic Annotations , Andrés Muñetón Carlos Zapata, Dyna, year 79, Nro. 172, pp. 94-100. Medellin, april, 2012. ISSN 0012-7353

[18] Automating Reuse In Web Application Development Automating Reuse In Web Application Development, Josip Maras 2014, Printed By Arkitektkopia, Västerås, Sweden

[19] A Simple Implementation of UML Sequence Diagram to Java Code Generation through XMI Representation, George. T. Vadakkumcheril, , M. Mythily, M. L.Valarmathi

[20] A Novel Approach for Business Logic Evaluation Model in Web Service Environment, Divya.A, and V. Renuka, IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 9, November 2014

[21] The Effect of Automatic Code Generation on Developer Job Satisfaction, Stephen Cullum, Department of Computing,Faculty of Mathematics, Computing and Technology The Open University , Walton Hall, Milton Keynes, MK7 6AA ,United Kingdom

[22] Code Generation by Model Transformation. A Case Study in Transformation Modularity, Z. Hemel, L.C.L Kats, E. Visser, Software Engineering Research Group, Delft University of Technology, The Netherlands,

[23] Auto Generation Of Code And Table Tool, Mr. Sandeep Agarwalla, Ms. Priyanka Roy, IJCSMC, Vol. 4, Issue. 4, April 2015, pg.487 – 492.

[24] Unified Modeling Language User Guide, The Grady Booch James Rumbaugh Ivar Jacobson, Addison Wesley ,ISBN: 0-201-57168-4, 512 pages

[25] UML 2.0 in a Nutshell, By Dan Pilone, Neil Pitman, Publisher: O'Reilly, ISBN: 0-596-00795-7, Pages: 234

[26] Java 6 Platform Revealed, John Zukowski, Apress, ISBN-13 (pbk): 978-1-59059-660-9, Pages 239

[27] The Definitive Guide to Java Swing, John Zukowski, Apess, ISBN (pbk): 1-59059-447-9 , Pages 913

[28] Java and XML Data Binding, Brett McLaughlin, O'Reilly, ISBN 0-596-00278-5, Pages 214

[29] www.oracle.com/us/solutions/.../financial-mang-analytics-ds-501409.pdf

[30] http://www.oracle.com/us/solutions/business-analytics/performance-management/financial-close-reporting/financial-management-analytics/resources/index.html

[31] http://www.microsoft.com/en-in/dynamics/crm.aspx

[32] http://www.interdynbmi.com/microsoft-dynamics-crm

[33] https://msdn.microsoft.com/en-us/library/2x7h1hfk.aspx