A new Approach to Capture Business Logic from UI with Automatic Code Generation and Database Creation

Sunil D. Rathod
Ph.D. Research Scholar, Department of Computer Engineering, JJT University, Jhunjhunu, Rajasthan, India
Dr. S. D. Joshi
Ph.D. Research Guide, Department of Computer Engineering,
Bharati Vidyapeeth University College of Engineering, Katraj, Pune, India

Abstract:
The Integrated Development Environment (IDE) based tools like .NET framework, Windows Builder, Net beans are popular for easier and comparably faster project development. These tools with the capabilities of Drag and Drop (DND) tool box help developer to design Graphical User Interface (GUI) or User Interface (UI) just with mouse. But, none of these utilities are capable of embedding "Business Logic" (BL) automatically in code. In routine development process it's really time consuming to do the repeated task of coding for same events. The vendor specific tools like Oracle Application Express (OAE) from ORACLE, VB .Net from Microsoft and some similar tools from IBM mainframe are used to design and develop the products for analysis and reporting purpose but these vendor specific tools strictly need the proprietary DB to develop the project. The OAE needs strictly ORACLE as a backend for its product development. Our tool - "Rapid Project Builder" (RPB) is aimed to avoid time consumed for coding the same business logic repeatedly. The tool will also perform Automatic Code Generation (ACG) in specific language like C++, Java, etc. The RBP helps developer to design the application's frontend GUI with DND along with the specification of form and fields. After adding control like add, delete, search or modify buttons to the application, RPB automatically embed code for the BL to these controls in language specified by developer. The automatic generation of DB tables is done with the help of specification given as the properties which are stored as XML semantics.

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

1. Introduction

In Software Industry the client demands are always very dynamic over the time which becomes a big challenge at the time of product development. Different stakeholders of the project focus on cost reduction, efficiency and reliability of product to be developed. Various tool and techniques of global standardization are already available to develop quality software products with reduced cost. The Object Management Group (OMG) has devised various internationally accepted standards to make the SDLC process more generic, platform independent and automatic with respect code generation and transformation.[12],[13],[16],[17].

The tools designed on the principle of UML specifications focus on design of the software projects using DND, with forward engineering the designed classifiers like Class diagram, Use Case, State Chart and Sequence diagrams are converted to code [1],[2],[5],[9],[10]. The special efforts are taken for transferring Sequence diagrams into code using XMI representation [19]. A lot of efforts are being put in the direction of automatic code generation directly from the UML classifiers but none of the approaches is able to generate complete code with embedded business logic in the code. The code generated for the given classifier in UML is just a code template with class structure, its attributes with data type and methods with just prototype declaration without any actual code in it.

In the generated template code, the BL code is explicitly written by developer using different logic for operations like Addition, Change, Insert and Delete (ACID) on date stored in Database (DB) tables. The code of BL depends on the domain specifications and constraints of BL. [26] [29] [35]

The connectivity of front end with back end DB is another issue to be handled by developer of the projects. The connectivity can be done either by connection wizards of the development environment or using code explicitly written in development language. Thus the entire project development cycle takes a lot of time in addressing the issues like code for GUI using DND, Code for different control likes text box, command button, check box, along with different events like mouse click or key press events, DB creation with ODBC connectivity issues and last is embedding BL in code.

Our tool Rapid Project Builder (RPB) address all these issues together just by designing the project UI using DND along with the specifications of different properties of the controls. The RPB stores the information given by developer as Meta data in XML standard form and automatically generate code, DB with tables, and BL in the code.

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

2. Related Work

The evolutionary efforts have been taken by researchers since many decades to make software development process as easy and fast as possible without losing the robustness, efficiency, ease of use and cost effectiveness of the product. Due to dynamic changes in product development, it is very much essential to use 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.

To meet all the above challenges, OMG's MDA approach is very suitable and is adopted at large in many industries as standard practice of product development [12], [13], and [17]. The process of 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],[26],[27].

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

The authors in [3] have even tried to extend the functionalities of existing UML tools by developing a new tool called UJECTOR (UML to Java Executable Code generaTOR) for automatic executable Java code generation from UML diagrams. If a set of three UML classifiers namely class diagram, sequence diagrams and activity diagrams are given as input to the tool; it will automatically generate complete executable Java code. UJECTOR generates Java object-oriented code class structure from class diagram, method and its flow of control through sequence diagrams, and object manipulations is done by activity diagrams.

In [7] the authors describe a system that allows the 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 system program checks errors, if any, in pseudo code and converts the algorithmic specification into code of specific language like C, Java, or any other language.

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

The giant vendor Microsoft has many language specific tools like VB6.0 or VB/ASP .Net framework which use DND approach for UI design. These tools generation code templates but no BL or automatic DB table creation is performed [35].

The ORACLE Financial Management Analytics (OFMA) is also one of the tools designed with the intension of report generation but it too lags in generating source code of specific language and no BL embedded in code [32].

The similar features like UI with report generation are available with eclipse Windows Builder, Microsoft Dynamics Customer Relationship Management (CRM) [33], [34].

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

First kind of such idea was proposed in [23], [24] in which the authors have given the concrete frame work of tool like RPB.

The RPB uses an evolutionary approach towards reduction in the cost and time of development of software product using software engineering processes. As a first step, this tool is incorporated with the facility to develop a concrete prototype for standalone applications with backend support but later on this tool can be extended with the functionality to develop web based applications.

3. RPB Architecture

The RPB architecture and process flow is as shown in figure fig-1below.

The RPB tool has standard IDE for developer to design and develop the desired application using DND. The tool provides similar project form design facilities as that of VB .Net or Net beans using which a developer can easily design UI. The developer can either create new application from the scratch or open an existing one to customize it.

The various components of RPB are described at length in this section.

3.1. GUI Form designer

This is an IDE editor environment which consists of various panels like project explorer, property window, and tool box. The developer can design application form with DND facility along with the specification of properties of each of the controls used in form design.

3.2. XML Metafile Generation

This file contains all the Meta information automatically generated in XML format which is useful for automatic code generation, DB table creation and embedding of BL in code.

3.3. Language Library
Language Library consists of key words, syntax, semantics and different structures of various languages like C++, JAVA, .NET, etc. This library is basically used by RPB to generate working code of GUI form in the specific language by taking specification from XML.

3.4. Database Library
Database library consists of DDL, DML, DCL commands, syntax and semantics of Database systems like ACCESS, MySQL, and ORACLE. RPB invokes it to convert the GUI specification into Database tables using XML Meta information.

3.5. ODBC Connectivity Component
ODBC component creates connectivity interface between front end and back end of the application.

3.6. Language Specific code generation
This component creating front end code of the application in various languages like C++, Java, .NET, etc.

3.7. Database Specific table generation
This component creates back end of the application.

3.8. Validation Engine
This component performs form validation. Validation engine automatically makes validation of the text fields, text area, etc of the frontend UI forms.

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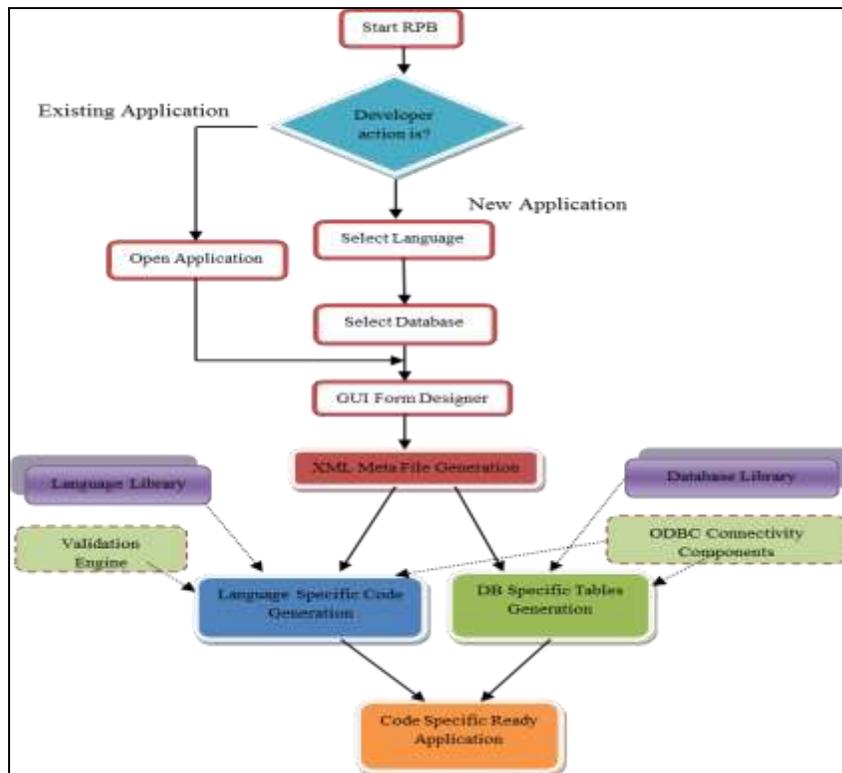

Figure 1: RPB Architecture with Flow

4. Methodologies
The methodologies used to design and develop RPB with its various components are discussed in this section.

4.1. Form Creation Mechanism using DND
The developer creates form with DND using controls in tool box. Through property window developer specifies various properties of controls like Label, text field, etc. The properties like data type of each field, size, name, fonts, once set are used for creation of CLASS with attributes and corresponding fields for DB table. This is depicted in figure fig-2 below.
The controls like Label, Text Box, Buttons, etc. have separate property window using which the properties for each of these control can be set by developer.

4.2. Conversion of GUI to XML Meta information
Once the form is designed RPB automatically generates XML Meta file and also the DB tables in backend DBMS. The form object is given XML converter module. This module extracts proper information from java form object. One of the modules of converter will generate XML Meta file. This is depicted in figure fig-3 below.

Customer Registration

Name:
Age:
Sex:
City:
Pin:

Ok   Cancel

Figure 2: Form creation using DND

4.3. Conversion of XML Meta Class to JAVA Class
The RPB automatically creates a JAVA class from the above generated XLM Meta class file using Language library syntax set file. The Language library syntax set file has all the relevant language key words, conditional/loop control, class, interfaces, design pattern, and syntax of specific language. Each language has separate file with all above information stored in it.
The SAX PARSER will scan the XML form file and parse the standard TAGs defined by 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 JAVA is the default language in which RPB generates the code automatically but developer can specify any OO language like C++ or C# supported by RPB.

4.4. Conversion of XML Meta class to Database creation
The conversion of XML Meta file information to Database creation is done using SAX parser in Java and RPB conversion module. The UI designed by developer using RPB is stored as XML Meta information. It is parsed to extract the Database table name, its attributes with data type and size of each attribute.
This extracted information is used in Conversion Module of RPB and a DB Table is generated using DML syntax of specific DBMS like MS-Access as follows.
CREATE TABLE Customer (Name Text, Age Int, Sex Char(5), Pin Long);

Here the Database with Project name and .mdb extension is first created once workspace and project in that workspace is created.

```
<FORM NAME = "Customer Registration">
<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" />
</CLASS>
</FORM>
```
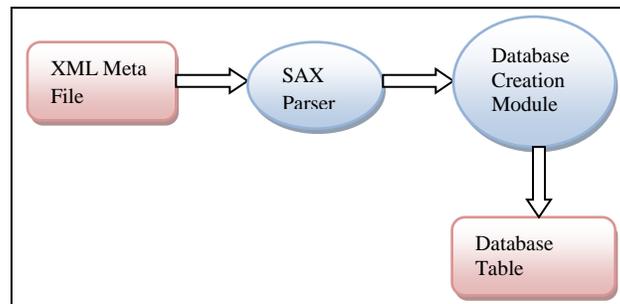
Figure 3: XML code generated by RPB

Figure 4: SAX Parser conversion process

4.4. Capturing BL from UI

RPB will dynamically capture the details of the BL to be generated from User Interface. When developer drags and drops the button control from standard tool box, the property window of button control helps developer to specify all the operation specific business logic details like what action user wants to perform on the specific button, is it ADD, DELETE, SHOW or any other specific operation?

As these properties are stored properly in XML file, RPB BL insertion module will parse the essential details from XML meta information and will create the appropriate query as per the BL operation so that front end and back end will be synchronized and proper action will be performed by application at run time.

The overall process is as follows. RPB BL insertion module will parse the details like the data entered by user at run time, the operation of the button and the table in which the data is to be synchronized. Here data synchronization means performing the operation like INSERT, DELETE, UPDATE, DISPLAY, CANCEL, etc. The module then generates complete code for operation to be performed on button click.

For INSERT/ADD operation following BL code is generated by RPB.

"INSERT INTO Customer(Name, Age, Sex, City, Pin) VALUES('"+new String(Name.getText()) +"',"+new
Byte(Age.getText())+",'"+new String(Sex.getText()) +"','"+new String(City.getText()) +"',"+new Long(Pin.getText())+")";

The values entered by user in text field are converted into the format of underlying DB automatically.

5. Experimental Results

RPB is an open source tool and will be made available to open source community after its completion. The implementation of some of the modules of RPB is shown as figures below.
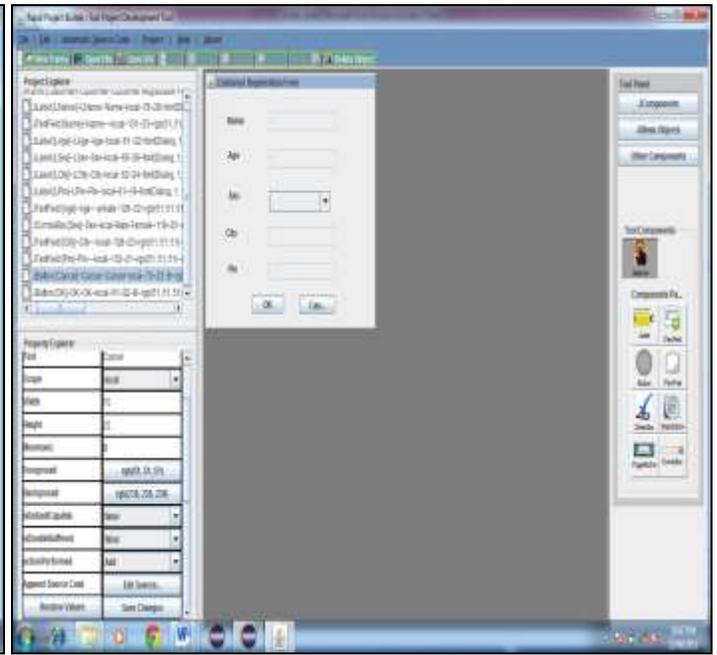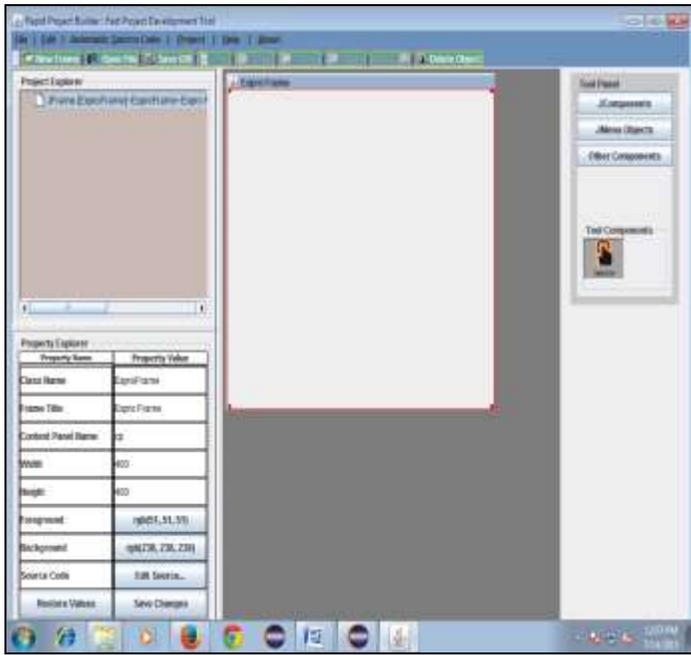
Figure 5: RPB Editor Environment with Form Design, Property window & Tool box   *Figure 6: Customer Form desing using RPB editor*

The Source code and XML Meta information generated by RPB can even be saved separately in the work space. These files can be used at any point of time by developer. More specifically the source code file is file containing complete working code which can be executed on any machine having proper compiler of corresponding language in which source code was generated by RPB.
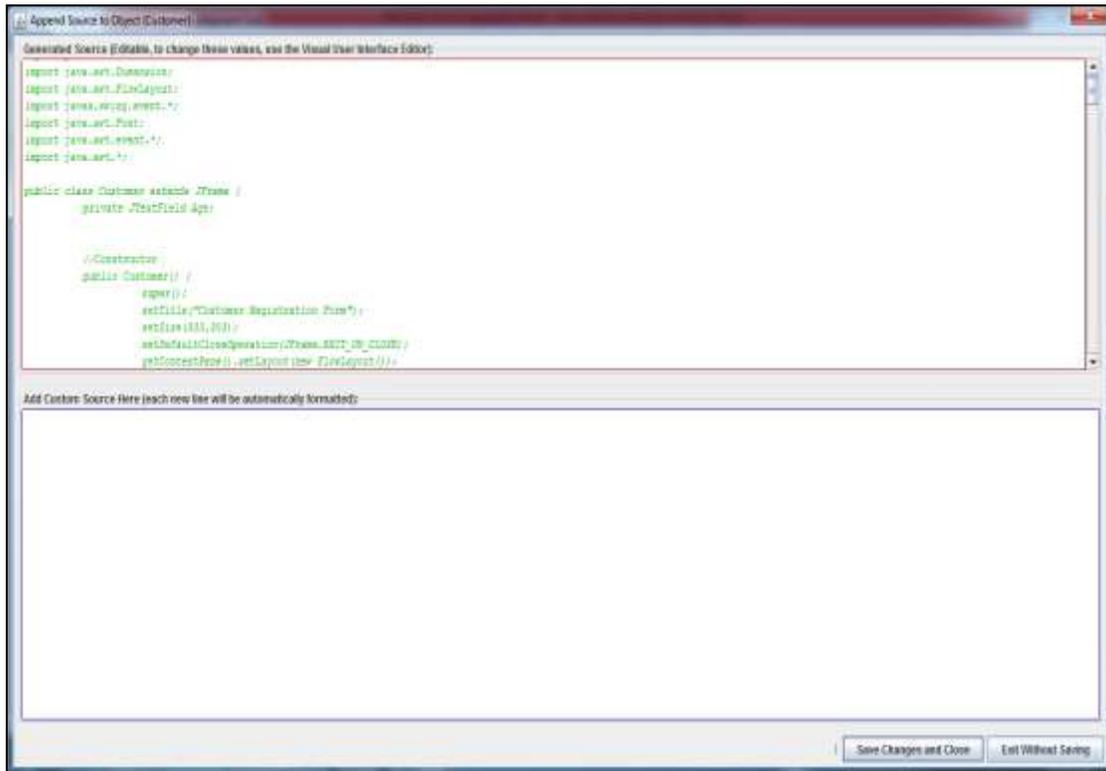
Figure 7: Source code of Customer Form using RPB

```
<?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_DATATYP
E>
<TEXTFIELD_DATASIZE>255BYTE</TEXTFIELD_DATASIZ
E>
</TEXTFIELD>
<TEXTFIELD>
<TEXTFIELD_NAME>Age</TEXTFIELD_NAME>
<TEXTFIELD_DATATYPE>INTEGER</TEXTFIELD_DATATY
PE>
<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_DATATY
PE>
</COMBOBOX>
<TEXTFIELD>
<TEXTFIELD_NAME>City</TEXTFIELD_NAME>
<TEXTFIELD_DATATYPE>STRING</TEXTFIELD_DATATYP
E>
<TEXTFIELD_DATASIZE>255BYTE</TEXTFIELD_DATASIZ
E>
</TEXTFIELD>
<TEXTFIELD>
<TEXTFIELD_NAME>Pin</TEXTFIELD_NAME>
<TEXTFIELD_DATATYPE>INTEGER</TEXTFIELD_DATATY
PE>
<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>
```

Figure 8: XML Meta data of Customer Form

6. Conclusion

The existing research done by various researchers and the tools developed by various global giant like IBM, ORACLE, and Microsoft are limited to the design of UI and generation of code template in some vendor specific language.

The work presented in this paper is an innovation towards making the software project development more simple, efficient and robust through our open source tool RPB.

The RPB is aimed to generate code automatically in various languages like Java, C++, and C #. If more language libraries are added in RPB, more will be the language support. At initial level DB support is given only for MS-Access and MySQL.

The sole efforts of this research are to reduce software development cost thus contributing towards the simplification of some of the processes of SDLC.

The scope of RPB can be extended for development of web application along with standalone Project development.

7. References

i. 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.

ii. Yingxu Wang Xinming Tan, Cyprian F. Ngolah, Design and Implementation of an Autonomic Code Generator based on RTPA, 44 International Journal of Software Science and Computational Intelligence, 2(2), 44-65, April-June 2010

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

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

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

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

vii. Suvam Mukherjee, Tamal Chakrabarti, Automatic Algorithm Specification To Source Code Translation,

viii. Amit Barve & Brijendra Kumar Joshi, Automatic C Code Generation for Parallel Compilation

ix. Sukhvir Singh, Neeraj Kumar, Effort Reduction by Automatic Code Generation,

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

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

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

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

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

xv. 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.

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

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

xviii. Josip Maras, Automating Reuse In Web Application Development Automating Reuse In Web Application Development, 2014, Printed By Arkitektkopia, Västerås, Sweden

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

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

xxi. Stephen Cullum, Walton Hall, Milton Keynes, The Effect of Automatic Code Generation on Developer Job Satisfaction

xxii. Z. Hemel, L.C.L Kats, E. Visser, Code Generation by Model Transformation. A Case Study in Transformation ModularitySoftware Engineering Research Group, Delft University of Technology, The Netherlands

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

xxiv. Chandan P & Rathod S D, Automatic Generation of Business Logic using DND, MJRET, 288-294, 1(3), October 2014

xxv. Chandan P & Rathod S D, An Approach towards Automatic Source Code Generation and Embedding Generic Business Logic Using DND, IJARCSSE, 513-521, 5(6), June 2015

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

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

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

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

xxx. Java and XML Data Binding, Brett McLaughlin, O'Reilly, ISBN 0-596-00278-5, Pages 214

xxxi. www.oracle.com/us/solutions/.../financial-mang-analytics-ds-501409.pdf

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

xxxiii. http://www.microsoft.com/en-in/dynamics/crm.aspx

xxxiv. http://www.interdynbmi.com/microsoft-dynamics-crm

xxxv. https://msdn.microsoft.com/en-us/library/2x7h1hfk.