

# Automatic Conversion of Desktop Applications to Java Web Technology

Matija Tomašković

Evolva d.o.o.

Zagrebačka 94, 42000 Varaždin, Croatia  
matija.tomaskovic@evolva.hr

Ruben Picek

Faculty of Organization and Informatics  
University of Zagreb  
Pavlinska 2, 42000 Varaždin, Croatia  
ruben.picek@foi.hr

**Abstract.** *This paper analyses the problem of automatic conversion of legacy desktop applications to Java web applications with emphasis on how such conversion methodology can help in migration of legacy desktop developers to modern web development. Authors explain how automatic converter can enable desktop developers to start delivering new web applications immediately without knowing complex web technologies.*

*Finally, the paper will show that after conversion of legacy applications to web technology, it is also possible to enrich converted applications with modern web and business functionalities.*

**Keywords.** Desktop2web converter, business web application, web framework, Java web development, rapid application development (RIA), desktop application, migration, converter, conversion, automatic code generation

## 1 Introduction

Authors of this paper explored the benefits of automatic conversion in the process of migrating legacy desktop applications and legacy developers to web development. The migration refers to teams that develop desktop applications, either having graphical user interface (GUI) or text mode user interface (TUI), but want to develop in a web environment.

When considering strategies of legacy system modernization to web development, it is important to preserve advantages of the legacy system, such as its high business value and good quality code, but also to take care of other issues like maintenance and upgrading. The most appropriate strategy that meets these objectives is the migration strategy.

The migration strategy is the approach that moves the entire legacy system and its core framework to the new environment while preserving the original system's data and functionality. [2]

Authors propose Evolution Framework as a concrete Java web framework that must exist for the purpose of the automatic conversion. Framework-

based software development has been proven as useful technique to develop an application. However, the development of a framework itself is considered complex due to its large size and the vague requirements. [3]

## 2 Desktop vs Web Development

Before the World Wide Web was established, desktop applications ran locally on computers and users had to manage their installation for any update, download, installation, etc. Later, accessibility through the web began to be used to develop applications that support business processes – those applications are called business web applications.

Web application is a client-server application that uses a web browser as its client program. It delivers interactive services through web servers distributed over the Internet (or intranet). A web application can present dynamically tailored content based on request parameters, tracked user behaviors, and security considerations. [4]

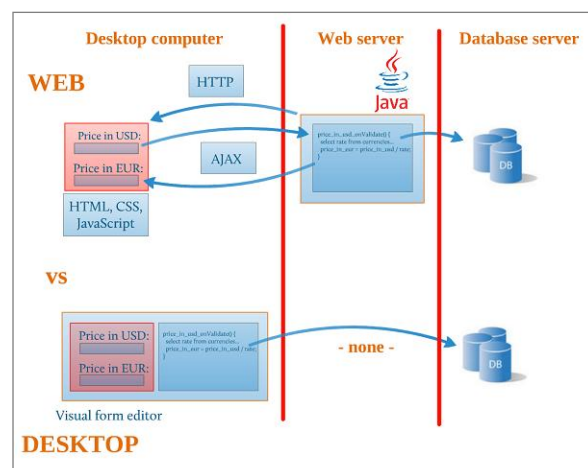


Figure 1. Technical difference between desktop and web development

The biggest difference between developing desktop and web applications is in the way the user interface is built and how it works, as seen in Fig. 1. Many authors argue the importance of user interface design within the overall application development. [6] [7]

## 2.1 The architecture of desktop user interface

User interfaces for enterprise desktop applications are designed in a visual editor provided by some of the desktop technology: Visual Basic, Delphi, Visual C++ or some 4GL tools like Panther (Prolifics).

When it comes to writing the code for the form, developer's code typically handles events on the form. The very first event that developer handles is usually the one that happens when the form is displayed to the user for the first time and the developer wants to initialize it. Later on, the user will click on a button, or double-click a row in the grid, or press a keyboard shortcut, and the code should handle these events interacting with a database or external services in a multi-tier systems.

## 2.2 Characteristics of web user interface

When developing web applications, designing the user interface is much more complex. Developing a web-based user interface can be quite tedious, it is a distributed program which has to run in a hostile environment. The web application user interface is the way through which the web user communicates to the web system. Web applications are essentially client-server applications - there is always a web client. [8]

When web applications are developed with web framework that doesn't provide a visual editor, programmers have to write the form layout character by character, tag by tag, using HTML, CSS and JavaScript code. Some web frameworks do provide visual editors that generate HTML code, but they are just partly helping to make HTML code while programmer still needs to touch up generated HTML code manually and after bigger manual changes some visual editors cannot load modified HTML pages again.

Another important aspect of web development is communication. Web architecture requires the use of HTTP to send an initial form layout to a web browser. Then, when a user clicks a button, AJAX technology has to be used in the web browser to call the event code on the web server to interact with the database. Finally, developers have to return the update information from the web server to the web browser to update form layout, e.g. the data grid with new data rows or input fields with new values. AJAX (Asynchronous JavaScript and XML), which supports an immediate dynamic interaction between a user and a web application (similar to the interaction between a

user and a desktop application), is one of the most popular technologies for web applications. [4]

Creating a user interface in a way that complete code is written from scratch is appropriate for small web applications, but for enterprise applications which consist of hundreds or thousands of different forms such development is a very slow process prone to errors and, in the end, very expensive and hard to learn. Significant improvements in software development efficiency are achieved by automatic code generation which ultimately results in higher productivity and uniformly high quality. [1]

## 3 Business aspects of migration to web development

During the process of migration there are certain business aspects that must be considered by development department and its management:

**Aspect 1: a protection of business know-how.** Outsourcing partners is a common case during the process of migration, but outsourcing the development of business functionalities implies giving away knowledge about company processes and plans. Therefore, it is important to protect the business know-how.

**Aspect 2: a protection of existing workforce.** The existing development team has many advantages that a company wants to retain:

- they have a huge business know-how gained through many years of working in the company,
- they have a good relationship and communication with application users across the company,
- they are the only ones who are versed in existing business applications and can fix or upgrade them quickly,
- they are very efficient in desktop development and excellent as a team.

**Aspect 3: continuous delivery.** As company processes are evolving, the existing development team has to maintain current enterprise applications and spend time on upgrading them constantly. This means they don't have enough time to learn new complex web technologies from scratch. This brings into question the need to learn web technologies quickly and start delivering new web applications without slowing down current development and delivery. Hiring new people is often not an option because it represents an additional cost.

**Aspect 4: reuse of existing applications.** Over time, current enterprise desktop applications is difficult to maintain because the old technical knowledge will be lost and old software or old hardware support will be discontinued. Usually, there are thousands of desktop forms in action. Some applications are still being upgraded very often and the migration process must allow to continue improving and upgrading them on web technology.

**Aspect 5: a complete framework for the efficient development.** It is recommended that web development framework is powerful, built for enterprise environment and has a visual editor for forms. Also, if framework includes popular features that are required for modern enterprise web applications, this additionally contributes to the efficiency of development. These days users demand not just grids with data and forms with input fields but also document management, workflow and task management features, reporting to different file formats such as PDF or Excel, integration with e-mail and web applications which must also work on mobile devices and tablets.

**Aspect 6: focus on solving business issues, not technical issues.** Business programmers must be focused on solving business issues and not on solving technical problems in the web environment. The web framework must hide the complexity of web architecture and web technologies.

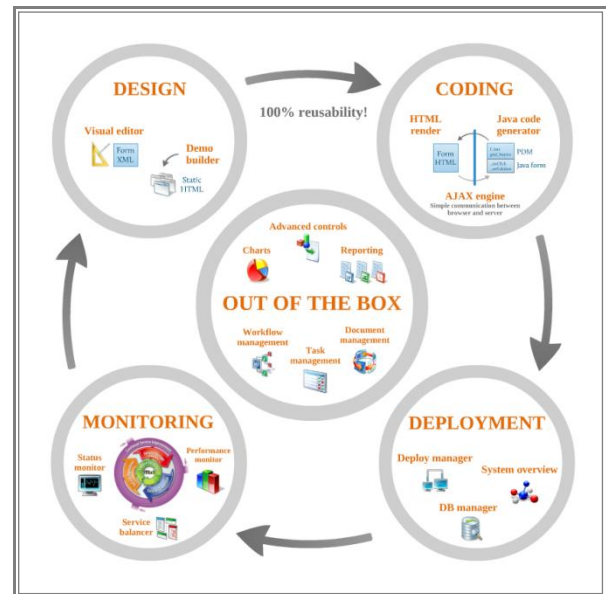


Figure 2. Evolution Framework tools

## 4 Methodology of the migration process

Keeping all mentioned aspects in mind, it is necessary to make two important decisions at the beginning of the migration process: choosing technology and choosing framework.

For this research, Java has been chosen because it is a proven technology for development of enterprise-level business web applications, it is free to download and there is a huge Java open source community with many available solutions.

When it comes to choosing framework, all above objectives imply that desktop programmers must be able to skip the use of complex web technologies and quickly start with the delivery of web forms. Such an efficient development of modern enterprise applications can be achieved only by using a web framework that allows desktop-like web development. A framework of such kind is Evolution Framework whose tools were developed specifically for the migration process as described in this paper. An overview of Evolution Framework tools can be seen in Fig. 2.

Those high and specific requirements for framework leads us to the methodology of web engineering that promotes successful web application development through all phases of development, including migration of legacy system to web environments. [10]

Meeting the unique requirements of web-based applications during whole application lifecycle can bring the potential chaos in web-based system development under control, minimise risks, and enhance maintainability and quality. [9]

## 5 Desktop-to-web converter

When considering how to move the existing applications to the web, a software tool can convert old desktop forms to web forms automatically. With such a converter, thousands of old form layouts and thousands of lines of old code could be automatically converted to the web environment (Fig. 3.). This saves the developers significant amount of time needed for rewriting old code.

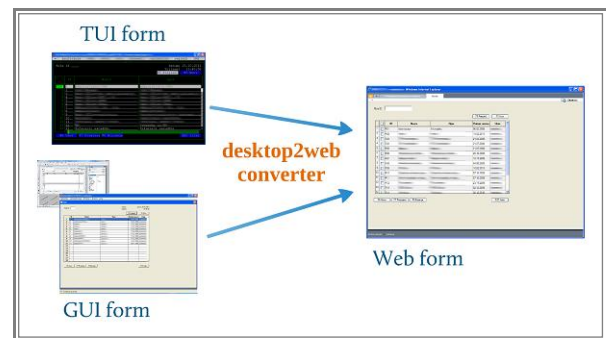


Figure 3. Desktop-to-web converter

The main purpose of the converter is to convert old code, but in this paper we particularly demonstrate how it enables a great migration strategy for developers.

## 6 The migration strategy when having a converter

When having a converter, authors propose that migration process is done in 3 steps:

### 6.1 Web development with converter

With the desktop-to-web converter it is possible to start the web development almost immediately. First, a developer has to setup a web development, testing and production environments. A visual system overview screen makes this much easier, without bothering with many configuration files.

Then, a programmer will write new forms in the old desktop technology, but when the desktop form is finished the programmer will use the desktop-to-web converter to convert the desktop form to a web form. This way, desktop programmers is able to start delivering web forms and web applications instantly.

### 6.2 Comparison of codes

The desktop-to-web converter is not just a tool for converting old forms to web forms. It is also a learning tool. After converting desktop forms, developers can compare the old desktop form code with the new Java form code and learn the syntax more quickly. A comparison of the old and new converter code can be seen at Fig. 4. Some authors propose certain methods to improve accuracy of code conversion. [12]

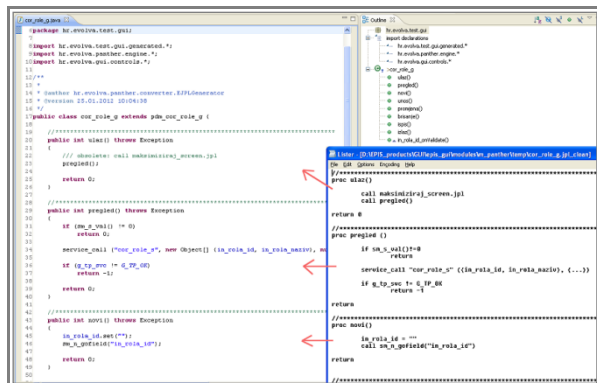


Figure 4. New converted Java code on the left, the old desktop code on the right

After the developers learn the new Java code syntax and get familiar with framework tools, they can start designing form layouts directly in visual editor and write Java code using e.g. Eclipse for Java as a popular IDE.

### 6.3 Conversion of old applications

Moving old applications to the web environment will be done simply by using the desktop-to-web converter. Usually, this step starts with top priority candidates for migration, the ones that will soon be out of old software or hardware support, or maybe those that they want to upgrade with new modern technical features.

Sometimes, the conversion of old applications is the first step in migration project and afterwards the delivery of new web forms can begin.

## 7 Architecture of converter

In desktop technology, every form has two main parts: the layout information and the form event code (Fig. 5.) Therefore, the converter must include the conversion of both form layout and form code.

Layout information for a single form is usually written as a binary or text file. Such a file contains a list of all controls like buttons, input fields and grids which appear on the form along with their initial property values like x and y position, enabled/disabled, visible/invisible, etc.

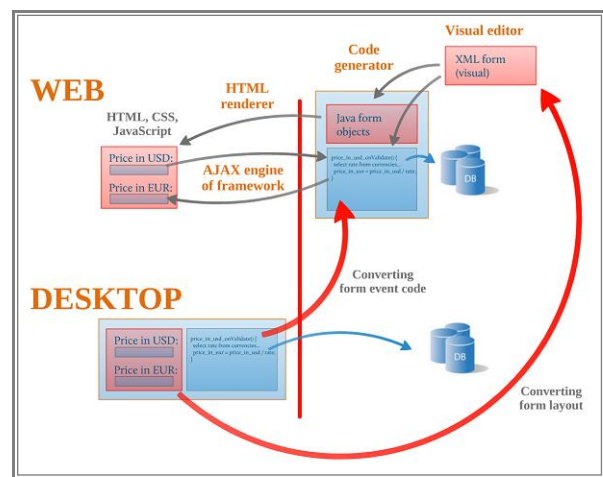


Figure 5. The process of conversion

The first step in creating the converter was to make it handle the conversion of this old layout information to the new layout information written as a visual editor form file. After the conversion, users can use the visual editor to modify the form layout and add some more controls to the form, even some advanced ones that didn't exist before in the old environment.

Desktop event code for old desktop forms is an actual code that consists of functions that handle specific events on the forms like form initialization, button click, double-click on the row, field validation after the focus is lost, etc. The second step in creating the converter was to make it handle the conversion of such event code. The converter must analyze the old code and make equivalent Java code.

Some old commonly used API functions, available in the old desktop framework, must be converted as well, so that the programmer has almost the same look and feel of the Java code in the web environment.

## 8 Example of conversion

The conversion described in this paper must be supported by a web framework whose web engine logic is very similar to the desktop engine's logic. Such a web framework must have the following features implemented:

- it must be capable of completely rendering the initial HTML form in the web browser, based on the layout information from the visual editor,
- it must render Java code for each form with all the controls on the form so that the programmer's form event code could easily communicate with the controls on the web server in Java,
- it must provide mechanisms that will send user events from the web browser to the web server directly via AJAX, so that the Java form event code could save the data entered by the users to a database and/or read the data from the database and update form controls,
- it must provide mechanisms to return event results from Java code on the web server back to the form in the web browser upon updating form content.

This way, programmers never have to bother with HTML, CSS, JavaScript, HTTP and AJAX, and they can focus on writing event code and interacting with databases and other external business systems.

In order to demonstrate how converter works, here are presented results of the conversion compared with the old forms.

### 8.1 Old desktop form

A sample of the old desktop form in a visual editor provided by a desktop technology is shown at Fig. 6..

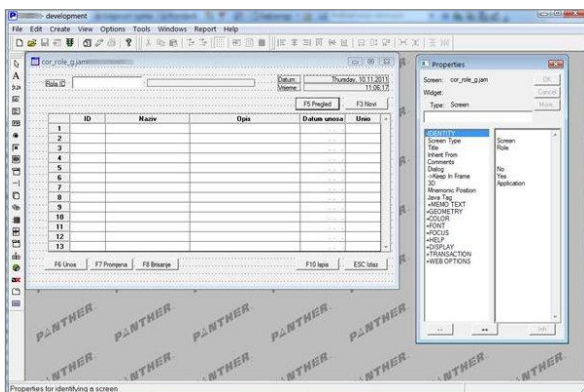


Figure 6. Old form in Panther IDE

Fig. 7. is showing same form in action having a text user interface, and the same form with a graphical user interface is shown at Fig. 8.

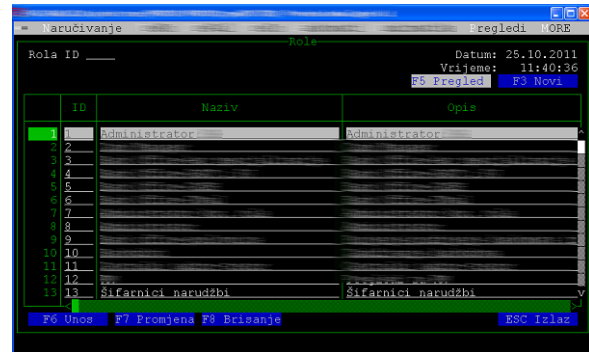


Figure 7. Old TUI form

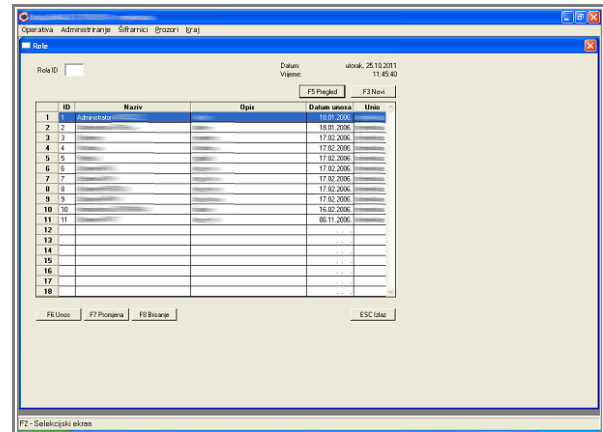


Figure 8. Old GUI form

### 8.2 Conversion of an old form

The legacy desktop IDE has an option to export a form to a single file which contains layout information and form event code.

The desktop-to-web converter tool is a part of Evolution Framework and it is a web based tool. This web application is run on a local web server and can access exported form files in a local folder, a sample can be seen at Fig. 9.:

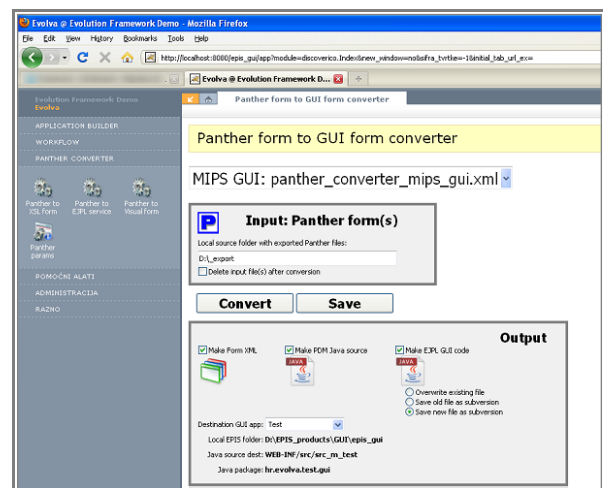


Figure 9. The desktop-to-web converter

When running the converter, it creates three things:

- first, it creates an XML file with the form layout which is an input file for the visual editor so that the form layout can be modified in the editor and later upgraded with new advanced controls;
- then it generates a Java code with all the controls on the form. This code is a Java "model" of the form that the programmer will communicate with either to read input field values, fill the grid or hide buttons. The programmer will never modify this code, it will be generated again every time the form is modified in the visual editor;
- and finally, the converter converts the original form event code to the Java code which will look almost the same because of the same API already implemented additionally into the framework.

### 8.3 The converted form

Fig. 10. is showing the final form in the visual editor. And the same form in action, after some redesign in the visual editor, can be seen at Fig. 11.

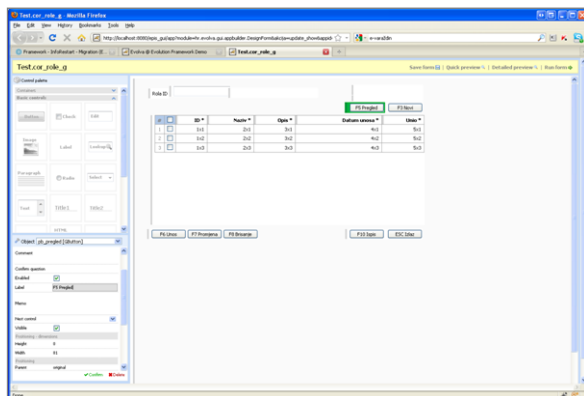


Figure 10. Converted form layout in Evolution Framework's visual editor

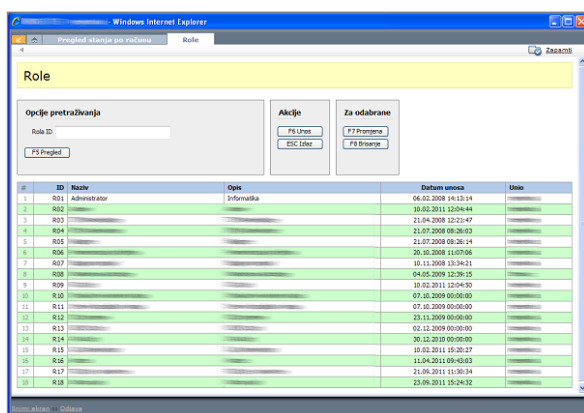


Figure 11. Web form running, after some redesign in Evolution Framework's visual editor

## 9. Reusability

The main objective of application framework is to promote the reuse of both design and code in the development of new applications. Reuse of existing components dramatically reduces the production cost and improve the quality.

The Rapid Application Development Framework, or simply RADF, aims to make application development more efficiently by means of architectural reuse. [3]

## 10. Conclusion

In this paper we have presented the role of the desktop-to-web converter in the process of conversion of legacy desktop applications to modern web applications as well as a value of such a tool in migrating a legacy development team to a web development.

Results of this research confirm the high relevance of the automatic conversion and high level web frameworks for overcoming the complexity of native web technologies. In order to ensure instant web development, it is crucial that a converter tool takes care of all specific characteristics of web architecture.

The automatic conversion has proven to be a secure way to preserve functionalities of old desktop applications but only a complete web framework will allow those applications to be further easily upgraded and maintained in new environment.

## References

- [1] Burke P. and Sweany P., *Automatic Code Generation Through Model-Driven Design*, 20th System and Software Technology Conference, Las Vegas NV, 2008.
- [2] Almonaies Asil A., Cordy James R. and Dean Thomas R., *Legacy System Evolution Towards Service-Oriented Architecture*, International Workshop on SOA Migration and Evolution SOAME 2010, Madrid, 2010.
- [3] Dongjin Yu, *Towards the Rapid Application Development Based on Predefined Frameworks*, Journal of software, vol. 6., no. 9., 2011.
- [4] Shklar L. and Rosen R., *Web Application Architecture: Principles, Protocols and Practices*, 2nd ed., John Wiley & Sons Ltd, Chichester, England, 2009.
- [5] Henrich V., Hinrichs E., Hinrichs M. and Zastrow T, *Service-Oriented Architectures: From Desktop Tools to Web Services and Web Applications*, Romanian Academy Publishing House, Bucharest, Romania, 2010.

- [6] Strahonja V. and Picek R., *User Interface Modeling Within Application System Development*, Proceedings of the 27th International Conference Information Technology Interfaces, 285-291, Cavtat, Croatia, 2005.
- [7] Strahonja V. and Picek R., *User Interface Modeling With RUP*, Proceedings of the 16th International Conference on Information and Intelligent Systems, 57-54, Varaždin, Croatia, 2005.
- [8] El-Bakry H. M., Riad Alaa M., Abu-Elsoud M., Mohamed S., Hassan A. E., Kandel M. S. and Mastorakis N., *Adaptive User Interface for Web Applications*, Publisher: WSEAS, 190-211, 2010.
- [9] Choo C.H. and Lee S.P., *Towards Persistence Framework-based Rapid Application Development Toolkit for Web Application Development*, Journal of Computer Science, vol. 4., no. 4., 2008.
- [10] Murugesan S., Deshpande Y., Hansen S., Ginige A., *Web Engineering: A new Discipline for Development of Web-based Systems*, in Web Engineering: Managing Diversity and Complexity of Web Application Development, Springer-Verlag, 2001.
- [11] Zhang W., Berre A.J., Roman D. and Huru H.A., *Migrating Legacy Applications to the Service Cloud*, in Proc. of Towards Best Practices in Cloud Computing, OOPSLA09, Orlando, USA, pp. 59-68, 2009.
- [12] Sudhakar P. and Sakthivel P., *Predicting Source Code Irregularities in Automated Code Conversion Systems using SRASG*, European Journal of Scientific Research, 67(3), 486-491, 2012.