

Model Driven Development for Software Architect in Reengineering with Cloud Computing

E. Kirubakaran

*Additional General Manager
B. H. E. L
Trichy , India
e_kiru@yahoo.com*

S. Manimekalai

*Department of Computer Science
Government Arts College
Trichy, India
mega_somu@yahoo.com*

Abstract-Skill Development varies from person to person due to their way of thinking. Presently, most of the organizations are grappling to maintain their information with the older development tools, they trade – off into more modern and leverage to search for an efficient programming languages to develop a new one with a minimal cost. In the Software Development Architecture, there exist several phases to develop an application. To computerize the system first step is to design the application based on the requirements. For example, the software architect who is known in markup language (i.e., HTML) can create the design and further convert into implementation phase. Here, the design time will be reduced by a special technique called as Re-Engineering. After converting the application, the organization has to maintain the architecture of their work process. At that situation this Cloud Computing is used to integrate into our components. The cloud computer has many factors that contribute to the success and survival of the company during transition one of them are to assess learning curves of many different individuals. This paper aims to shed light on the realities of screen scraping and discuss some of the possibilities and limitations of automated language converters.

Keywords-Programming Language, Re-Engineering, Snapshot, Software Architect, Software Development Architecture, Cloud Computing

I. INTRODUCTION

This paper mainly focuses on Designing and Implementation. The Software Architect in IT Systems and Business Processes they are taking more time for designing the system. The reengineering technology reduces the effort by converting the design from one technology to other technology. Business Processes they develop a software application in different language for the academic trainees, they can just design the application in HTML and it converts what the implementation is required along with database operations. Likewise, the software architect of the IT Systems can use this MinSOR tool, when developing an application 85% of work has been reduced.

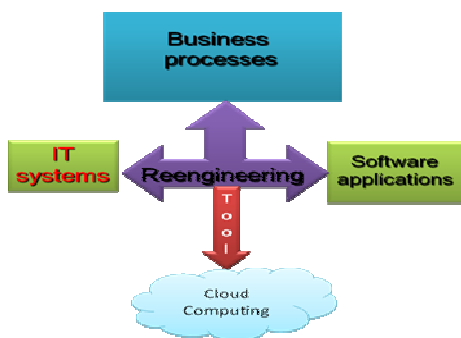


Figure 1. Process of Reengineering Tool

All the organizations are grappling to maintain their application, generally it takes 45%. Many large and important companies are throwing much money into this, and they will not see it's wasted. To avoid such complexity we progress to integrate the MinSOR tool into "Cloud Computing". Cloud Computing is suitable to deduce the enterprise software solution. It is adapted for requirements and strategies for detecting the solutions.

This paper has been organized as follow: Part- II, how or why we arrive this problem is discussed. Part- III, discusses about the methodology part of the problem domain. Part-IV, Experimental Results, Part-V represents

the comparison of existing software tool. Part-VI describes about the conclusion and future enhancement. Part-VII shows the reference papers related to this work.

II. RELATED TOOL WORKS

To implement the concept of re-engineering technology to convert the database from the design phase to implementation phase, we analyze the works done by other researchers which are related to Re-engineering. The papers we referred are given below, and there exist several graph-based reengineering tools.

The comparison with other projects following a graph-based approach, such as Rigi [12], Bauhaus [11], and GUPRO [4], shows that most of these tools lack the support provided by a high-level specification language. Hence, graph transformations cannot be specified in a declarative way. These projects also concentrate on reverse engineering and do not support software restructuring. The approach in shows how refactorings for object-oriented software can be defined by using graph rewrite rules using FUJABA and AGG [5] for tool validation. AGG is a general tool environment for algebraic graph transformation following the interpretative approach.

In the paper [10], they first report on what the assumptions are for adopting a model-based approach for forward engineering. Then, we describe a model-based approach for reverse engineering web pages implemented as an automatic or mixed-initiative process in the VAQUITA software, with an eye to applying forward engineering subsequently. Reengineering methods are then considered to produce new UIs for other contexts of use, thus creating a capability to rapidly produce UIs for different computing platforms, various access devices, etc

TaLE Tool - (Tampere Language Editor) it supports an object oriented programming environment. It is implemented in C++/Windows Environment. The user interface is partly graphical. The motivation of the system, the underlying conceptual model of language processing and the user interface issues.

Selenium tool (selenium IDE is a plug-in to Firefox) to record and playback tests (like WinRunner, QTP). Export the recorded test in most language e.g. html, Java, .net, Perl, ruby etc. The exported test can be run in any browser and any platform using "selenium remote control".

TidyCOM tool is the first step in the process is to clean up your HTML pages so that a XSLT (XSL Transformations) or a DOM or SAX parser can work with the documents. Dave Raggett's HTML Tidy is a good tool for converting your untidy (non well-formed) HTML to well-formed documents and also XHTML and XML. For a review on HTML Tidy read HTML Tidy: Keeping it Clean. This is command-line tool and will not be of much help if we are considering a Web -based interface. André Blavier has developed a COM Wrapper for HTML Tidy and also a Windows-based GUI front end called TidyGUI. TidyCOM can be used from scripting.

WebCream automatically converts Java Swing and AWT applications into fully-functional HTML and AJAX websites. WebCream can be thought of as a runtime Java to HTML converter that migrates Java graphical user interface to a browser-based front end. It then emulates user actions as GUI events for the Swing application to ensure that the business logic executes correctly. WebCream is unique in that it requires no modifications to existing applications, and does not require programmers to learn any APIs other than AWT/Swing. The same application can be run as thick Swing client or as a browser-based AJAX rich client. This allows companies to leverage the time and money invested into building Swing applications, while enjoying the benefits of thin-client deployment.

III. METHODOLOGY

The Main Motivation of this paper is to convert the Design from one technology to other technology, along with the database operation. Design is a process of converting system requirements into a completed product. This design is done by a Software Architect what the Project Manager instructs. And the Software Architect can also do the validation process. Software Architect has to submit the design to the Project Manager. The Project Manager instruct to the Business Requirement Document (BRD) to prepare the key note for the process of conditions, given by the Business Analyzer. Work progress can be allotted to the programmer. It takes only 15% of work, why the Software Architect cannot process this because of the Business Logic, each concern has different format of the operations.

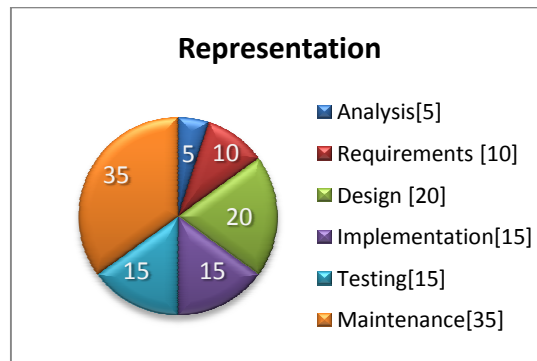


Figure 2. Graphical Representation of Reengineering Tool

From this representation, now 45% of work has been saved. To integrate the MinSOR tool into the cloud computing it saves 80% of work is reduced.

The Software Architect or Learners are well trained in HTML, since it is easy to design. It is not necessary for the designers to understand the programming language carried out in the development stage. Sometimes, both may be entirely different (i.e.,) the designing language is different with the implemented language. In that situation, the re-engineering technique helps the programmer to re-design the software by analyzing the tools used by the designers which are in HTML language. Each language has its own set of controls to design and from that design the implementation can be proceeded.

We also implement this technique to convert the design of the table which is designed in HTML by the Software Architect to database in Java. The Software Architect just enter a table name, it automatically retrieves and creates the design with the necessary fields and data type using HTML. The user has to choose the common DSN name in which the required table is placed. The Database operations (Insert, Update, Delete, and Select) can be performed by clicking the option.

This work is based on the domain “Cloud Computing”. Cloud Computing is suitable to deduce the enterprise software solution. It is adapted for requirements and strategies for detecting the solutions. It builds ontology for enterprise software to re-engineer the design and then partition the ontology to decompose the enterprise software into potential service candidates.

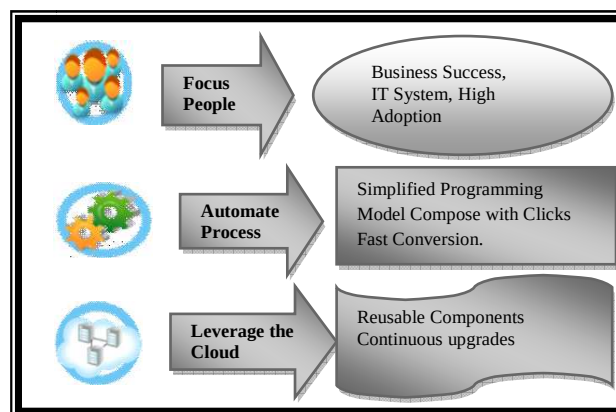


Figure 3. Process Automation

There are three steps in reverse engineering called re-engineering are as follows:

- Building Ontology for Source code, data and application.
- Integrating captured ontology.
- Deploying final produced ontology to obtain reverse engineering.

In this work, the ontology is built as a re-engineering tool is explained to convert the design from HTML to Java. But we also implement the provision to allow the user to choose the converted programming language. This work has been generalized to all kinds of software.

IV. EXPERIMENTAL RESULTS

The technique Re-Engineering is used to re-design the forms from one technology to another along with the database process, which helps the Software Architect to implement the code according to their platform. The designer just creates the tools in their platform. The aim of the proposed method is to convert the View Edition GUI Tools from HTML to Java AWT and Applet.

The proposed method has the three steps to follow: The first step of this conversion is that carefully observe the input objects in HTML. To extract the equivalent classes in Java for the input objects. Convert the input objects into appropriate tools available in Java. The tools are converted into appropriate programming language implemented to develop the process. On selecting the database, the list of tables and fields will be displayed. The database operations are selected to carry out the process automatically. The code has written by the programmer is reduced and the process is done automatically.

Proposed Approach of Deployment Cloud Computing

The main intension of this paper is to furnish a framework for developing MinSOR tool, which alleviate the deployment and development of Cloud based applications. This produces an unseamed environment, which conceals the Cloud implementation and deployment points. Mainly it focuses the problem domain of the Software Architect. This process requires as follows:

1. Developers of Software Architect
2. Prefer the application concerned to Design of the Software Architect.

Once the Design is produced by the Software Architect, the framework tolerates the MinSOR tool, which is a SOA, translates into executable Cloud specific code and automatically deploys it to the target Cloud platform.

Users can approach the deployed applications through browsers. The Key advantage of the demonstrated method is that the MinSOR appropriates the Software Architect to Design the applications using high level abstraction for symbolizing problem domains. In contrast to software development with a general- intension language such as Java Swing, implementation details are concealed from Software Architect.

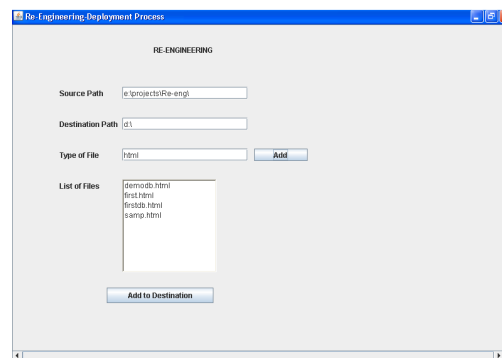


Figure 4. Deployment Process (HTML)

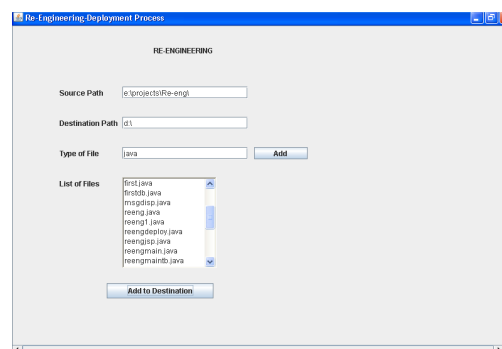


Figure 5. Deployment Process (Java)

There are two major computing trends in cloud computing. The first one is Virtualization defines a separation of the Operating System (i.e., my machine) from the underlying platform resources (i.e., cloud computing system). This removes much of the memory, storage, and network limits of a physical desktop system. It also seeks to simply the user interface, to protect the individual user from the complexity of all these background necessities of the computing world. The second is Effectiveness computing a packaging and

reselling of computer resources. Though in many ways similar to traditional utility services and it is better understood when compared to a web-hosting site. While many companies cannot afford to purchase and maintain sufficient hardware and software as to host their own web site, an effectiveness computing service will provide, on a rented basis, all the virtual software and hardware that may be required.

Another area needed for enhancement was the task of deploying applications into the cloud in a timely fashion and being able to update the application components rapidly and in the operational theater. In the application model, from a software perspective, the shape of the Web and SOA applications changed daily. Before to the enhancement effort, applications were deployed as approximately a desktop application; the deployment process was slow and inefficient.

In the proposed deployment model, operator-less processing was achieved by using our MinSOR tool. The resulting deployment architecture is not customer-facing; it is a view of the application architecture from a system administrator's perspective.

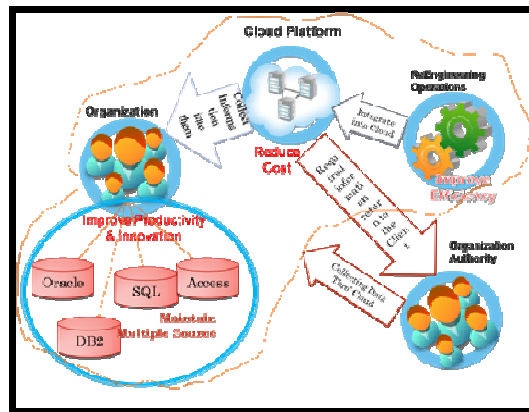


Figure 6. Components Integrate into Cloud

V. COMPARISON REPORT

The Re-Engineering technology is used to derive the JAVA design from the HTML design. Also, it is used to perform the database operations with the validation process. With the help of these techniques, the programmer and the designer has better benefits in developing the software in much lesser time. Also, only by providing the attribute values, the database operations have been handled efficiently by the programmer without much effort. Thus our study provides much better performance in the conversion process.

Thus it appears that cloud computing and information technology share the same fundamental functions. Each fills the consumer need for manipulating, storing, and retrieving various forms of information. Each has the need for technical expertise and administration. And though some of the current applications and tools are beginning to reveal specific differences, both methods require file servers, bandwidth, routers, and all manner of other hardware/software combinations. Each also carries an expensive start up and operating cost.

Table 1. Comparison of Tools

Features	MinSOR	o-Code	Selenium	Simo View	Web Cream
Conversion from one language to another	yes	yes	No	yes	yes
HTML to Java(Swing) Conversion	yes	No	No	No	No
HTML to JSP Conversion	yes	yes	No	No	No
HTML to .Net Conversion	yes	No	No	No	No
Validation on data	yes	No	No	yes	No
Database Conversion Automated	yes	No	No	No	No
Deployment	yes	No	yes	No	No
Database Operation Restriction	yes	No	No	No	No
Migrating the Converted File	yes	yes	No	No	yes
Support Multi-Web Server	yes	No	No	No	No

VI. CONCLUSION

The aim of the paper is to reduce the time for the programmer to convert the design from one technology to another. The cost of delivery by increasing the efficiency of the process. Creating 100% of an automated conversion tools. The conversion is made by creating and maintaining the design tools for each programming language. Conversion process is carried out by analyzing and identifying the appropriate controls from the destination language such as AWT, .NET, JSP, C# and Applet, which are suited for the control in the Source language such as HTML.

In future, the re-engineering technology can be carried out to convert the code from one language to another language is developed to improve the progress of Re-Engineering. It also implements the database operations that have to be done in the selected fields in the selected tables of selected database. Thus, when the field type or table changes, not necessary to change the code, it is done automatically using this technique.

In future, this can be enhanced by carrying out implementing the Re-Engineering technique in convert the code from one programming language to another, in which nobody knows source and destination language. Thus, the Programmer using the Re-engineering technique to simplify the process in all situations.

This paper attempts to lay out the various elements that collectively form the cloud computing platform. The primary goal of this paper is to discuss and clarify the concepts around cloud computing, while defining a framework to gauge the adoption of cloud as a platform with business imperatives. The secondary goal of this paper is to better understand the business drivers and organizational commitment which inspires the shared vision of a cloud computing platform. We would also emphasize that while infrastructure technologies such as hardware and software virtualization as a service form the individual building blocks of cloud computing platform.

REFERENCES

- [1] Andreas Wolff, Peter Forbrig "Model-based Reengineering of User Interfaces", University of Rostock Institute of Einstein Str. 21, 18059 Rostock, Germany Computer Science Albert.
- [2] Armburst et al, "Above the Clouds: A Berkley view of Cloud Computing". [On-line] Available [here](#); Retrieved: July 01 2009.
- [3] Bouillon L, Vanderdonckt J, Eisenstein J, "Model- Based Approaches to Reengineering Web Pages", in Proceedings of 1st International Workshop on Task Models and Diagrams for user interface design TAMODIA'2002, INFOREC Printing House, Bucharest, pp. 86-95, 2002.
- [4] Ebert J, Kullbach B, Riediger V, Winter A. "GUPRO – Generic Understanding of Programs: An Overview". Electronic Notes in Theoretical Computer Science72 (2), 2002.
- [5] FUJABA – From UML to Java and Back Again. 1999.
- [6] Gray W.A, Fiddian N.J, Behrendt W, "Database Interoperation Support In Multimedia Applications Architecture And Methodology", Department of Computer Science, Cardiff University, UK.
- [7] Kirubakaran E, Manimekalai S, "Re-Engineering for GUI Conversion from Markup Language to Programming Language", International Journal of Advanced Research in Computer Science, Vol 1, No.4, Dec-Nov-2010.
- [8] Kirubakaran E, Manimekalai S, "Wrapper-Based Design Conversion in Database Re-Engineering", International Journal of Advanced Research in Computer Science, Vol 2, No.2, Mar-Apr-2011.
- [9] Koschke R. "Atomic Architectural Component Recovery for Program Understanding and Evolution". Doctoral thesis, Institute of Computer Science, University of Stuttgart: Stuttgart, Germany, Stuttgart, Germany, pp 414, 2002.
- [10] Laurent Bouillon, Jean Vanderdonckt , "Model-Based Approaches to Reengineering Web Pages", Université catholique de Louvain Belgian Lab. of Computer-Human Interaction.
- [11] Mainetti, Paiano, Pandurini, "User-Centered reverse engineering: Genesis-D project", in proceedings of Web Maintenance and Reengineering 2006, CEUR Workshop Proc. 193.
- [12] Muller H.A, Wong K, Tilley S.R, "Understanding Software Systems Using Reverse Engineering Technology", in the 62nd Congress of L'Association Canadienne Francaise pour l'Avancement des SciencesACFAS 1994. Pp. 41–48. Montreal, Canada, May 1994.
- [13] Rathack, Wolff, Forbrig, "Using HCI-Patterns with Model-based Generation of Advanced User-Interfaces", Proc. of MDDAUI 2006, Models 2006, Genova, Italy.
- [14] Tucker, K. and Stirewalt, R, " Model based User Interface Reengineering", in Proc. 6th WCRE, 1999.
- [15] Van Gorp P and Grefen P, "Supporting the internet-based evaluation of research software with cloud infrastructure," Software and Systems Modeling, pp. 1--18, 2010.
- [16] Vanderdonckt J, Bouillon L, Souchon N, "Flexible Reverse Engineering of Web Pages with VAQUITA", in Proceedings of IEEE 8th Working Conference on Reverse Engineering WCRE'2001 (Stuttgart, 2-5 October 2001, IEEE Press, Los Alamitos, pp. 241-248, 2001.