Volume 4, Issue 2, February 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

STAR Paradigm for Reengineering of Web Application

Abhilasha Sharma*
Assistant Professor, Deptt. of Software Engineering
Delhi Technological University, Delhi, India

Nishu Singh, Poonam Dhiman
M.Tech Scholar, Deptt. of Software Eng.
Delhi Technological University, Delhi, India

Abstract-The economy globalisation together with the need of new enterprise strategies has enormously promoted the development of web applications. As the development of web applications does not follow basic principles of software engineering methodology hence it lacks effective programming construct and testing techniques. Heterogeneity in tools/technique and indiscipline development has increased the necessity of web reengineering. Reengineering is useful in both web and software application but differs in various perspective from each other. This paper describes comparative analysis of reengineer process for software and web and the problem of reengineering of web applications, which leads to the need of STAR paradigm to define and implement a reengineering process that involves web applications and supporting tools. It offers a structured method for defining reengineering process and paradigm which provide systematic approach to reengineer any web application. By providing smooth transition of paradigm to web developers, we can help them to identify and understand various perspective of reengineering with minimum effort and time span.

Keywords-Web Applications (WA), Reengineering, Reverse engineering, Application Domain (AD), Web Application Connection Graph (WAG).

I. INTRODUCTION

The technological evolution of the last year has made the web service of the ideal platform for the appropriate support for their delivery and the development of web-based applications. According to research [1] and [2] the development of a web application is a multi-faceted activity, involving not only technical but also organizational, managerial and even social and artistic issues. Web application development refers a set of activities which applied in order to develop a web application of high quality having awaited characteristics, and to accomplish the development efficiently and coherently. Web engineering is the discipline which covers the overall cycle for the development and maintenance of web application. It is a significant and fast developing area, gaining more attention these days. Web maintenance and web reengineering both falls in the scope of web engineering. The World Wide Web has ability to ubiquitously provide and gather information to the economy globalization together with the need of new marketing strategies which has enormously boosted the development of WA. Software application is the backbone of the WWW infrastructure.

Why Reengineering?

Most of applications are developed under proper schedules and in a rapidly evolving environment. The development is often ad-hoc in nature and the applications are poorly structured and documented. Maintenance of such applications becomes problematic and increases the complexity in the application growth. Creating appropriate design and architecture models is the solution by managing this complexity and supporting evolution of applications. Researchers had identified the need to reengineer the system for already existing web applications into abstract design models.

When maintenance cost is not feasible, we go for reengineering the system. Reengineering makes the system new. It includes both the concept of forward engineering and reverse engineering.

Need of Reengineering

When system changes are confined to one subsystem, the subsystem needs to be reengineered, software and hardware support becomes obsolete and tools to support restructuring are readily available.

Reverse Engineering

In spanning of life cycle stages, reverse engineering covers a broad range starting from the existing implementation recapturing or recreating the design and deciphering the requirement actually implemented by the subject system. It include re-documentation, design recovering from code, restructuring like code to code transformation. Reverse engineering is a process to transform a code into model through a mapping from a specific implementation language [3].

Purpose

- Identify the system component and their interrelationships.
- Create representation of the system in another form or at a higher level of abstraction.
- Involves extracting design artifacts.

Forward Engineering

Forward Engineering process moves from higher level of abstraction to lower level where application gets integrated according to new design. It is the traditional process of moving from higher level abstraction to lower level and logical implementation to the physical implementation of a system. Forward engineering follows a sequence begins from requirement through designing its implementation.

The detailed study of all these areas has been done later in this paper which comprises of five sections. Section 2 illustrates the comparative analysis of reverse engineering and reengineering in terms of both software and web application that surrounds various intrinsic and extrinsic parameters. Section 3 covers an overview and conceptual knowledge of the different levels of reengineering for software and web application and also describes some of their differences. Section 4 introduces STAR paradigm and explains how it can be used to define and implement web application reengineering process while section 5 finally provide the concluding remarks.

II. REENGINEERING PROCESS VS REVERSE ENGINEEING PROCESS

This section focuses on some of the intrinsic and extrinsic differences of reengineering and reverse engineering [4]. Intrinsic comparisons: e.g. web page processing which were designed to be stateless whereas software processing language make use of states. Extrinsic parameter: e.g. cost and results. Some of differences in the process of reengineering and reverse engineering are listed in table 1.

TABLE I
REVERSE ENGINEERING VS REENGINEERING

Parameters		Reverse Engineering	Reengineering
Objective		To drive the design or specification of a system from its source code.	To produce a new and more maintainable system.
Definition		Reverse Engineering is finding out how a product works from the finished product.	Reengineering is examining the finished product and builds it again in better way.
Process	**Software Engineering**	1. It is trying to recreate the source code from the compiled code and trying to figure out how a piece of s/w works given only the final system.	1. It is creating a new piece of s/w with similar functionality as an existing one but improving the way it was build.
		2. It is important in s/w maintenance due to effectiveness & analysing the consistency between design and implementation.	2. It is concerned with the reimplementation of the legacy system to make them more maintainable.
	Web Engineering	Reverse engineering extract information from the web application and allow more abstract representation (model) to reconstruct.	Reengineering reconstruct the model view which is extracted by reverse engineering and generating semantic and syntactic descriptions.
Companies Perspective		Companies follow reverse engineering to copy and understand parts of a competitors products, which is illegal, to find out how their own product work in the event that the original plans were lost, in order to effect repair them.	Companies follow Reengineering to adapt generic product for a specific environment.
Cost		Continuous refactoring will decrease the total cost.	Cost is higher compared with reverse engineering.
Result		Reverse engineering improves the existing structure.	Reengineering create the whole new system with different structure and different behaviour.

Table 1 reflects that reverse engineering is used during the software reengineering process to recover the program design, which engineers use for better understanding of a program before reorganized its structure. However, reverse engineering need not always be followed by reengineering. The activities of reengineering process involves *source code translation* which converts program from an old programming to modern version and in data reengineering the data processed by the program is change to reflect program changes [5], *reverse engineering* is used to analyse the program and information extracted from the program which helps in document its organization and functionality and to improve *program structure that* controls the structure of program and modified to make it easier for understanding, *program modularization* is a part of reverse engineering that used to group together related parts of program.

III. OVERVIEW OF REENGINEERING

There are many definitions to describe the reengineering process, each is slightly different but all have same overarching theme, "the radical redesign of business process to achieve dramatic improvement in productivity and performance". The two terms are radical which means getting rid of existing processes, procedure and inventing new ways and dramatic improvement means a quantum leap in performance [6]. Reengineering is the main process in which organization becomes more modernize and efficient and to fulfil the demands for quality service and low cost, process must made simple, it transform an organization in a way that directly affects performance.

3.1 Software Reengineering

The objective of software reengineering process is to improve maintenance, improve reliability, adaptation of new platform and enhance the functionality of the software system. Some problems are faced due to lack of well-designed structure through which any change will affects the entire system which leads to complex and expensive software system. The organization do not ruin the system because it was built for many subsidiaries of group which, if destroyed will result in the application process to be lost. Therefore reuse of logic and component so that initial cost of developing logic and component of the software system should not waste. The challenges of software reengineering are to use existing system and add the good features and attributes. Developers created new software system where goal is to maintain required function in the application of new technologies. Reengineering is used to prepare for advance functionality rather than to enhance existing function. There are 3 levels of software reengineering [7]:

a) Domain Level,
b) System Level and
c) Component Level

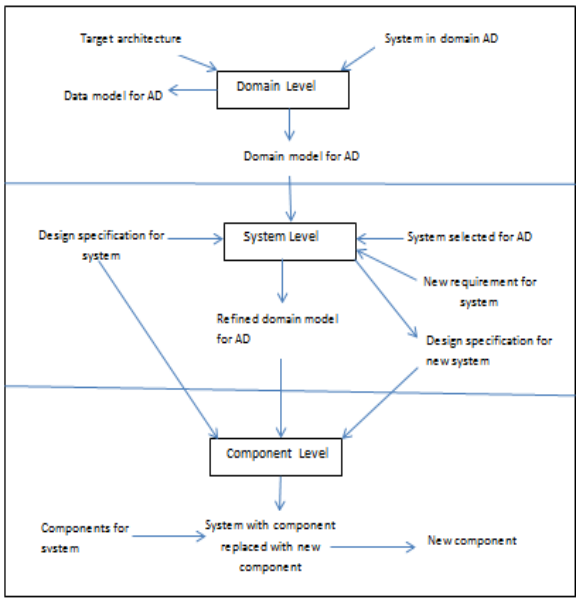

Fig.1 Level of Software Reengineering

Domain Level objective is to understand the system in application domain, prepare architecture framework for new system and analysis of domain [8].*System Level* produces the complete logical design specification for a selected system in application domain, these design specification are produced for both the original and target system. *Component level* is the incremental reengineering that selects system, component by component. *Approaches of Software Reengineering:* To access the software reengineering there are three different approaches [4]:

3. Big bang Approach: This approach is used when the entire system need to be replaced at a time. This method is often used for projects that require being resolved immediately. It consumes too many resources and large amount of time before generating the desire system. The risk of this approach is very high.
4. Incremental approach: In this approach the part of the system is reengineered and incremental updates have been made as the new version of the system requires meeting the new goals. The advantages of this approach is that component of the system are produced faster and easily control the errors when new components are clearly defined. This approach has lower risk than the big bang approach.
5. Evolutionary approach: In this approach part of the original system is replaced by the newly redesigned system and the parts are selected based on their function instead of existing system structure.

3.2 Web Reengineering

Web reengineering service helps to re-conceptualize and redesign the existing website and application service. Its objective is to restructure the web application and its pages are modified to control and access the standards and policies [9]. The 3 levels of the web reengineering [10] are:

a) Design Recovery
b) Analysis and Evaluation
c) Redesign

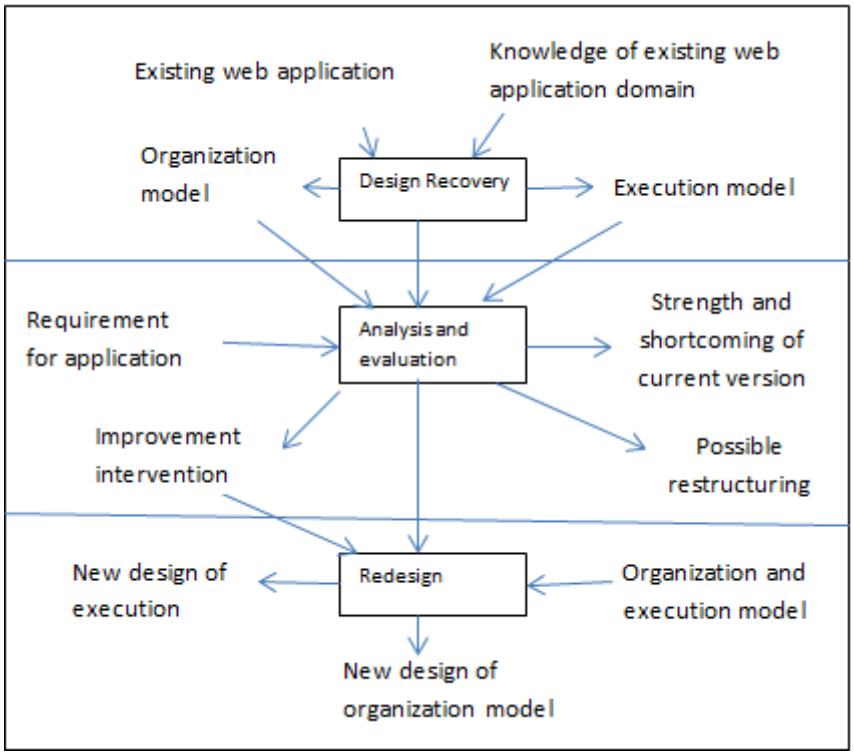

Fig. 2 Levels of Web Reengineering

Design Recovery: For existing web site goal is to revise the models with data obtain by direct inspection and analysis of site's content and structure [11]. Model can be recreated using three steps of design recovery procedure: 1) formalization of transaction (execution and organization model) 2) creation of execution model 3) construction of organization model. Automated reverse engineering process is used to improve the efficiency of the process. *Analysis and Evaluation*: The result of design recovery procedure is to model a web application transaction using revised version of the execution and organization model. The next step of this process is to perform the user-oriented analysis and evaluation of the recovered execution and organization model. The objective is to define all the possible restructuring for the current design and implementation of the considered transaction addressing the strength and shortcoming highlighted by this level. *Re-Design*: The objective of this level of reengineering is to redesign the execution and organization model and introduces the changes defined during analysis phase which further translated to recover the transaction design model that produces new design.

As we already discussed that reverse engineering and forward engineering subset of the reengineering process, so the development community of model based approach has shown the interest for forward engineering with many models such as domain model, dialog model, presentation model and application model [12], [13]. Today several environments support the forward engineering of web pages e.g. Allegro Serve is a web server that dynamically generate and enable existing web application with a browser front end. The goal of reverse engineering is to recover the presentation and dialog model of the web pages. Table 2 compares software reengineering with web reengineering with respect to the various parameters as mentioned below.

TABLE II
SOFTWARE REENGINEERING VS WEB REENGINEERING

Parameters	Software Reengineering	Web Reengineering
Restructuring	Reorganize source code to perform some function more efficiently	Reorganize people, system and infrastructure to perform some basic functions in potentially more efficient ways.
Retargeting	Transport the source code and application system to new system	Adapt an existing business process to perform in new business functions
Reverse Engineering	Examine design of existing software system by deriving design from existing software code	Examine design of existing business process by extracting design from existing implementation

Forward Engineering	Develop new system design based on integration of new system requirements into existing system design.	Establish new business process design based on integration of new business requirement into existing business processes
Data Reengineering	Restructure the organization and format of stored information for use by software application.	Restructure the organization and format of stored information for use either more manual or automated processing activities.
Architectural Evolution	For software it generally requires centralised system, which is migrated to a distributed architecture. It is essential that the core of that architecture should be a data management system that can be accessed from remote clients.	For web application it evolves through Client Server to N-TIER, Legacy to web Services, and Client Server to SOA (Service Oriented Architecture), legacy to web enablement.

Web application must cope with an extremely short development evolution life cycle: A high level of flexibility, maintainability, and adaptability are actually necessary to compete and survive to market inflation. Unfortunately, to accomplish tight timing schedules to deliver web services, web applications are usually directly implemented without producing any useful documentation for their maintenance and evolution, and so those requirements are never be satisfied. In order to satisfy a growing market request for web applications and to deal with their increased technological complexity, we require specific methods and techniques which are able to support a disciplined and more effective development process. However, the high time pressure often forces the developers to implement the code of the application directly, without using disciplined development process, and this may have black effects on the delivered quality and documentation of the web application. The situation is same as one occurring for traditional software produced in a short time, without respecting software engineering principles and using no disciplined development process. Poor quality and documentation must be considered as prime factors which lead to abortive and expensive maintenance, unattainability of applying more structured and documentation-based approaches.

IV. STAR PARADIGM TO DEFINE A REENGINEERING PROCESS FOR WEB APPLICATION

Reengineering process is usually run to abstract and extract data, document them from existing software, and to unify these documents and data with expert knowledge and previous experiences that cannot be instinctively reconstruct from software. But the proliferation of various problems in reengineering of web application results in a pressing need to bring a paradigm for the characterization of key parameters of this area. Hence, STAR paradigm is introduced according to which a reengineering process is characterized by Situation, Tools, Application and Restructuring. *Situation* defines a set of views of the applications to be reverse engineered. *Tools* include techniques and tools to support the information recovery process. *Application* describes which particular types of application require to migrate from one technology to another. Lastly, *Restructuring* specifies the actual implementation of reengineering process and also decides which type of reengineering process (e.g. transaction reengineering, data reengineering, graphic design reengineering etc.) is suitable for a given application. Presented below are all the four characterising parameters of the paradigm for WA reengineering process.

Situation

In the field of web applications, situation explains the possible scenario in which requirement of reengineering has emerged. A reverse engineering process may aid assessment of the characteristics of an existing application, in order to be able to evaluate its quality attribute, including reliability, security or maintainability [14].

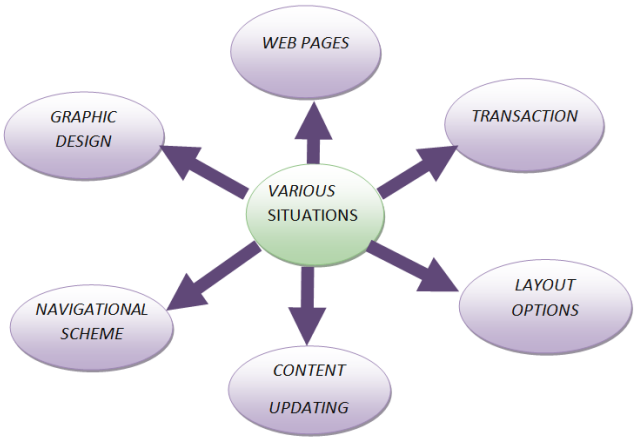

Fig. 3 Various Situations for Web Reengineering

Several situations and scenarios are there to reengineer a web application such as change in web pages by filtering the entity, tags and elements of the web pages that involve assemblage of any HTML objects with given properties that require keeping all control mechanism and eliminating the unwanted tags and elements from web pages. Reengineering of web pages can be accomplished by identifying and analysing the intercommunication between elements and then altering these elements for the modification of new platform itself and generating the source code into suitable programming language. *Transformation in the layout options* and relationships that include alignment balancing which depend upon the position of the objects on the page and *content updating* of the web pages according to the requirement changes, market evolution, usage and owner of website [15]. Reengineering is used to adapt a user interface to different format and to alter a navigational scheme that tells how web elements such as server page, client page, form, frame, email etc. are linked.

Tools

The recovery of information from an existing web application and the production of models, documentation of its relevant features that cannot be effectively accomplished without the support of suitable techniques and *Tools* that automate the web application analysis. However, the diverse and dynamic nature of objects producing the application, and the lack of effective mechanisms for implementing the basic software engineering principles in web applications, that makes analysis process more complex and make it necessary to address specific methodological and technological problems. Some transcoding tools [16], [17], [18], [19] automatically transform a UI code from the original platform to a target platform.

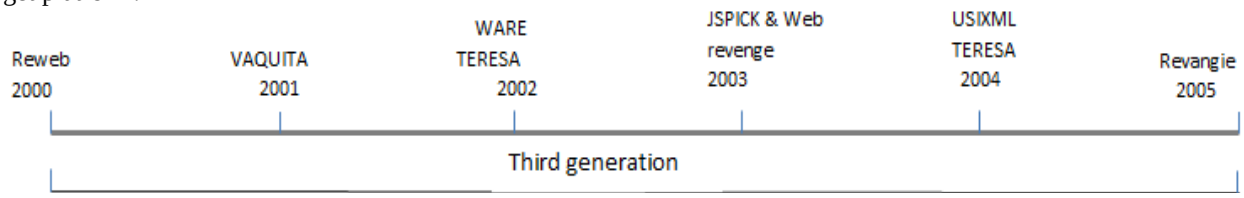

Fig.4 Evolution of Web Reverse Engineering Tools and Methodologies.

More precisely, heterogeneous software components developed with different technologies and languages require techniques and tools for multi-language analysis. The existence of 'dynamic software components' in a web application, such as pages created at runtime depending on user input, will impose the application of dynamic analysis techniques and static analysis of the code in order to obtain more precise information about the web application behaviour. In addition, the absence of effective mechanisms for implementing the software engineering principles of modularity, encapsulation, and separation of concerns, will make the use of suitable analysis approaches, such as program slicing which is necessary in order to localize more cohesive parts in the web application code. Finally, on the basis of the situations, tools / technique and application identified the sequence of activities composing the reengineering process, their input/output and responsibilities will be precisely set out. The reverse engineering process can be executed with the support of various engineering tools proposed as reweb [20], vaquita [21], WARE [22], [23], tersa JSPICK [24], web revenge and revangie tool.

Applications

Web application development within an organisation depends upon several factors. The number and the importance of web applications are increasing rapidly over year by year. At the same time, the quantity and impact of security vulnerabilities in such applications have grown as well [25] .The motivation depends upon the initial purpose of web usage, the customer's expectations and the competitive environment. The drive to systematise the development is subject to overall view of the web and conscious policy decisions within the organisation. For example, a low level view of the web is lead to ad hoc. Initially we need to understand the problem domains that currently addressed by web. Organizations that started their web development early may also have followed a similar order in the past. However it is possible to start web development with applications in any category. Table 3 presents categories of web applications which are useful to explain to organisations with modest presence on the web how they might improve or benefit from incremental exposure, thus keeping the risks to the minimum.

TABLE III
CATEGORIES OF WEB APPLICATIONS

Category	*Examples*
Workflow	Planning and scheduling systems, inventory management, status monitoring
Collaborative work Environments	Distributed authoring systems, collaborative design tools
Informational	Online newspapers, product catalogues, newsletters, service manuals, classifieds, e-books

Interactive ▪ **User-provided information** ▪ **Customized access**	Registration forms, customized information presentation, games
Online communities, Marketplaces	Chat groups, recommender systems, marketplaces, auctions
Web Portals	Electronic shopping malls, intermediaries
Transaction	E-shopping, ordering goods and services, banking
Web Services	Enterprise applications, information and business Intermediaries

Migration of applications to the newer technologies can provide leading edge to business by removing inefficient workflow and processes while preserving original objectives, model and investment. We can help enterprises in migration of the legacy systems from old technologies to present day platforms. Reengineering strategically designed to overcome the cross platform compatibility challenges. Due to upcoming advance technology and growing business states, there is need for the migration of legacy software systems to new technologies and environments. There are different kind of legacy system reengineering services that includes language and database migration, platform-to-platform porting and system redevelopment.

A web application must follows the enterprises standard and rules implemented in a legacy application, while transforming those to new business and architecture requirements, to produce a flexible, tested or validated modified system. Reengineering and application migration has benefits of saving time and effort, enhancements in operational efficiency and usefulness of the latest technologies and platforms.

Restructuring/Re-Conceptualisation

Reengineering is the analysis of existing software system and modifying it to constitute into a new form. Chikofsky and Cross define reengineering as 'the examination and alteration of a subject system to reconstitute it in a new form and subsequent implementation of that form [26]. According to IEEE Std. 1998, a system changing activity that results in creating a new system that either retains or does not retain the individuality of the initial system [27]. Techniques of static and dynamic analysis of the source code and dynamic data were taken into account.

Additional techniques for analysing the web application structure and identifying relevant subsets of its components were also considered where clustering techniques were defined to carry out this analysis. Finally, the specifications of the tools required to support these analyses could be defined.

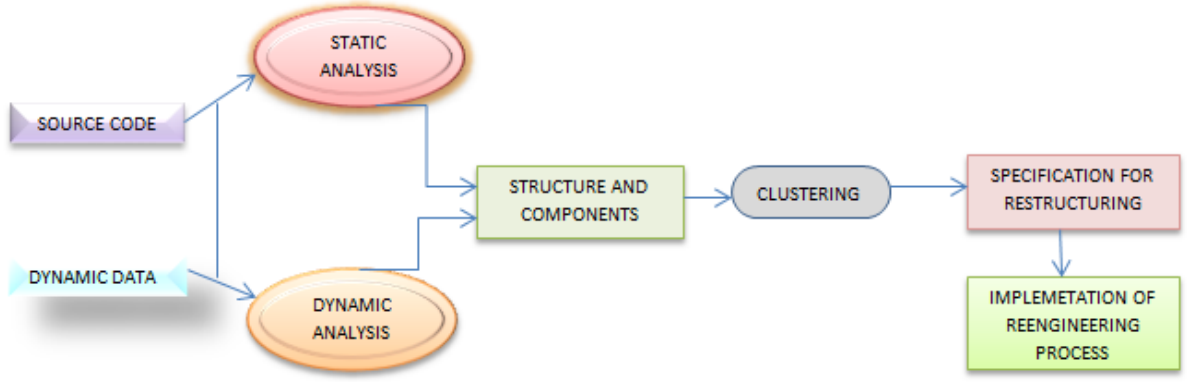

Fig. 5 Steps of Reengineering Process

In static analysis, all the information necessary to obtain the inventory of the web application entities (pages and inner pages) such as forms, scripts and other web objects are identified, the static relations between them is extracted from the code, and the statements producing *link*, *submit*, *redirect*, *build*, and other relationships are identified and localized in the code. The dynamic analysis phase is based on and uses static analysis results. The web application is executed and dynamic interactions among the entities described in the class diagram are recorded. Dynamic analysis is performed by *observing* the execution of the web application, and *tracing* any observed event or action to the corresponding source code instructions. Analysis of the execution is a task that can be carried out either automatically on the basis of the application code or manually by observing the page execution by a browser and recording the observed events.

The information about the web application obtained by static and dynamic analysis can be used to produce a graph whose nodes represent the set of Web application entities, and whose edges describe the different relationships among these entities. This kind of graph has been named as WAG; *Web Application Connection Graph* [28].WAG analysis may support the comprehension of the application. However, since this graph may be large (in terms of the number of nodes and edges) even in the case of small size web applications, in order to simplify the analysis of large WAG graphs, some kind of automatic clustering [28] can be used to decompose this graph into smaller cohesive parts. This clustering

approach evaluates the degree of coupling between entities of the application (such as server pages, client pages and client modules) that are interconnected by *Link, Submit, Redirect, Build, Load in Frame*, and *Include* relationships. After getting more abstract specification from clustering algorithms actual implementation of reengineering process is executed according to selected situation and application. Different approaches of reengineering processes based on the above three parameters (i.e Situation, Tools, Application) are listed below.

TABLE IV
DIFFERENT APPROACHES OF REENGINEERING PROCESS

Data Reengineering	The process of analysis and re-organising the data structure and data values in a system to make it more understandable.
Transaction Reengineering	In transaction oriented website, the user executes a series of activities in order to carry out a specific task. Business processes are realized by means of transaction, in which this context can be interpreted as high level work flows corresponding to user tasks.
Application Migration	Migration of application to the new technologies will give business a leading edge by removing inefficient workflow and process while preserving original objectives, model and investment.
Graphic Design Reengineering	It is used to adapt a UI to another context. To change user interface developers do not necessarily want to start from scratch to design a UI for a new platform since UI already exists.
Reengineering of Web Pages	Reengineering of web pages can be accomplished by detecting and analysing the interaction of object and transforming these objects for the adoption of new platform itself and generating the source code into new language.
Business Process Reengineering	It is the analysis and redesign of workflow within and between enterprises. It defines all the process in an organization and prioritizes them in order to redesign urgency.

Following is an example illustrating the real motive of STAR paradigm, Suppose there is *Situation* of graphic design needs to be alter then we can have transcoding *Tool* TERESA[16] that automatically transform a graphic interface code from the original platform to a target platform with any of the *Application* say interactive websites (User-provided information Customized access). Hence, above three parameters itself implies that *Restructuring* can be performed using graphic design reengineering which transforms a final user interface into a logical representation that can be changed to direct forward engineering to allow a user interface from one computing platform to another one having minimum effort.

SITUATION	TOOL	APPLICATION	RESTRUCTURING
Graphic design	TERSA (Transcoding)	Interactive website	Graphic design reengineering
Web page	VAQUITA	Web Portals	Reengineering of web pages

In a similar manner, we can have various tuples of different permutation and combination of all the four parameters of STAR paradigm.

V. CONCLUSION

The diversity and economic pertinent of web application are rapidly increasing. Unfortunately, web applications are usually developed with lack of disciplined process and effective programming principles. Due to this, maintenance and evolution of web application will become more complex. Hence, therefore there is a huge necessity of reengineering of web application to enhance functionality, improve maintenance process and to access the new platform. This paper discusses the reengineering for both software and web application that conceptualizes the different levels of reengineering process. Comparative analysis between reverse engineering and reengineering and between software reengineering and web reengineering has been done that encircles around various parameters. We proposed a STAR paradigm: *Situation, Tools*, *Application* and *Restructuring* that support the comprehension of existing web application during maintenance. The aim of the paradigm is to reconstruct web application illustrating various aspects of web reengineering. It provides effective and easier way for the reconstruction of web application. Introduction of STAR paradigm into web reengineering process can increase maintainability and effectiveness as it led to better understanding of procedures and practices. STAR paradigm will save the time for re-constructing or restructuring the web application due to strong discerning and contrivance.

REFERENCES

[1] P. Fraternali, "Tools and Approaches for Developing Data-Intensive web Applications: a Survey", *ACM Computing Surveys*, 1999.
[2] S. Selmi, N. Kraiem, and H. Ben Ghezala, "Toward a comprehension view of web engineering*"* , 2005.

[3] I. Jacobson, G. Booch and J. Rumbaugh (1999), "The Unified Software Development Process". Addison Wesley Longman, Inc, Reading: MA.
[4] B. Laurent, L. Quentin, V. Jean and M. Benjamin, "Reverse Engineering of Web Pages Based on Derivation and Transformation*"* , 27 August 2005.
[5] G. K.Tyagi and D. P. Ballou, "Examinig Data Quality*", communication ACM*, 41 (2), Feb. 1998.
[6] Hammer, Michael (1996), "Beyond Reengineering: How the Process-Centered Organization is Changing Our Work and Our Lives*"*, New York, NY: HarperCollins Publishers, Inc.
[7] Stanjarzabek, "Strategic reengineering of software:life cycle approach*",* Department of Information Systems and Computer Science National University of Singapore.
[8] S. Jarzabek, "Domain Model-Driven Software Reengineering and Maintenance*", Journal of Systems andSoftware*, January 1993, pp. 37-51.
[9] M. Giordano, G. Polese, G. Scanniello, and G. *Tortora, "*Visual Modelling of Role-Based Security Policies in Distributed Multimedia Applications*". In Proc. of IEEE6th International Symposium on Multimedia SoftwareEngineering*, Miami, FL, USA, IEEE CS Press, 2004, pp.:138 – 141.
[10] D. Distante, T. Parveen, and S. Tilley, "Towards a Technique for Reverse Engineering Web Transactions from a User's Perspective*"*, In *Proceedings of the 12th IEEE International Workshop on Program Comprehension* (IWPC 2004: June 24-26, 2004; Bari,Italy). Los Alamitos, CA: IEEE CS Press, 2004.
[11] S. Tilley, D. Distante, and S. Huang, "Design Recovery of Web Application Transactions*"*, Submitted to the 11thIEEE Working Conference on Reverse Engineering(WCRE 2004: Nov. 9-12, Delft, the Netherlands). June 2004.
[12] P. Szekely, P. Luo, and R. Neches, "Beyond Interface Builders: Model-Based Interface Tools*", Proc. of ACM Conf. onHuman Aspects in Computing Systems InterCHI'93*, ACM Press, New York, 1993, pp. 383-390.
[13] J. Vanderdoncktand P. Berquin, "Towards a Very Large Model-based Approach for User Interface Development",*Proc. of 1st Int. Workshop on User Interfaces to Data Intensive Systems UIDIS'99*, IEEE Computer Society Press, LosAlamitos, 1999, pp. 76-85.
[14] Offutt J, "Quality attributes of Web software applications",*IEEE Software* 2002; **19**(2):25–32.
[15] P. Diman, N. Singh and K. Kumar, "Unified V- Model Approach of Reengineering to reinforce Web Application Development", *IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume X, Issue X (Nov. - Dec. 2013), PP 01-00.*
[16] M. Moore, "Representation Issues for Reengineering Interactive Systems", *ACM Computing Surveys Special issue: positionstatements on strategic directions in computing research*, Vol. 28, No. 4, Dec 1996, article # 199, ACM Press, New York, NY, USA.
[17] M. Moore and S. Rugaber, "Using Knowledge Representation to Understand Interactive Systems", in *Proc. of the FifthInternational Workshop on Program ComprehensionIWPC'97* (Dearborn, 28-30 May 1997), IEEE Computer Society Press, Los Alamitos, 1997
[18] G. Mori, F. Paternò, C. Santoro, "Tool support for designing nomadic applications", *Proc. of the 2003 international conferenceon Intelligent user interfaces, Jan 2003*, (Miami, USA), ACM Press, New York, USA, pp141-148.
[19] F. Ricca, P. Tonella, I.D. Baxter, "Restructuring Web applications via Transformation Rules", *Proc. Of IEEE Workshopon Source Code Analysis and Manipulation SCAM.2001* (Florence, 5-9 Nov 2001), IEEE Computer Soc. Press, Los Alamitos, 2001, pp. 150-160.
[20] Vanderdonckt, J., Bouillon, L. and Souchon, N. (2001), "Flexible reverse Engineering of Web Pages with VAQUISTA*"*,Proc 8th Working Conference on *Reverse Engineering, WCRE'01, IEEE*, pp241-248.
[21] L. Paganelli, F. Paterno, "Automatic reconstruction of the underlying interaction design of web applications", in *Pro .Ofthe 14th international conference on Software engineeringand knowledge engineering*, (July 2002, Ischia, Italy), ACM Press, New York, USA, pp 439 – 445.
[22] Di Lucca, G.A., Fasolino, A.R., Pace F., Tramontana, P. and De Carlini, U. (2002a),"*WARE: A Tool for The Reverse Engineering of Web Applications"*, Proc. 6th European Conference onSoftware Maintenance and Rengineering (CSMR'02), pp241-250.
[23] D. Lucca, G.A., Fasolino, A.R. and Tramontana, P. (2004), "*Reverse Engineering WebApplications: The WARE Approach"*, Journal of Software Maintenance and Evolution,Research and Practice, Vol 16, pp71-101.
[24] D. Draheim, E. Fehr and G. Weber (2003), "JSPick – A Server Pages Design Recovery",Proc7th IEEE European Conference on Software Maintenance and Reengineering, LNCS, pp230-236.
[25] Jovanovic, N. ; Secure Syst. Lab., Tech. Univ. of Vienna ; Kruegel, C. ; Kirda, E., "a static analysis tool for detecting Web application vulnerabilities",Security and Privacy, 2006 IEEE.
[26] E. Chikofsky and J.H.Cross, "Reverse Engineering and Design Recovery: A Taxonomy", *IEEE Software Engineering journal*, (Jan. 1990), pp 13-17.
[27] IEEE Std 1219-1998, In IEEE Standards Software Engineering, 1999 Edition, Volume Two, Process Standards, IEEE Press.

