



Reverse Engineering Of Web Applications:- An Emerging Trend

Reema Sandhu

Assistant Professor(Computer Science), Department of Higher Education

Priyadarshini Indira Gandhi Govt. College For Women

Jind(Haryana), India

reemasandhu111@yahoo.com

Abstract:- Reverse Engineering is defined as the process of duplicating an already existing component, subassembly of a product without the aid of drawings, documentation of a computer model by learning the design of an object and studying its implementation details. It is the process of discovering the technological principles of a device, object or system through analysis of its structure, function and operation. The aim of this research paper has been to highlight various methodologies used in RE applications. The focus has been on four important methodologies i.e. Specific ASP.Net methodology, UML and Ontology based approach used for restructuring the Web Applications. The use of different tools in Reverse Engineering of web applications like JSPick, VAQUISTA, ReWeb, TERESA and Revangie are highlighted which provides the means for recapturing lost information, restructuring complex systems or transforming old systems to a new systems, more maintainable and providing them better architecture. Traditionally the development uses forward engineering approach which is the traditional process of moving from high-level abstractions and logical designs to the physical implementation of a system. Reverse engineering of web application helps researchers to examine the strength of web pages and identify their weaknesses in terms of performance, security, and interoperability.

I. INTRODUCTION

The process of Reverse engineering is very common in such diverse fields as software engineering, entertainment, automotive, consumer products, microchips, chemicals, electronics, and mechanical designs. The goal of reverse engineering is the improved documentation of how the original product works by uncovering its underlying design. The discipline of reverse engineering was first proposed to address the “inherited” problems associated with legacy software so as to provide users of such systems with the option of updating and extending such software systems rather than allowing them to become redundant[1]. One reason design engineers are increasingly using it is that its hardware and software have become more affordable, helping engineering companies, to speed up development and cut production costs. In the case of WWW applications the “state-of-the-art” is still in its infancy[2]. Web Applications (WAs) have become popular in all enterprises as they provide the underlying engines that improve a company’s image and act as useful resources for increasing a company’s overall share in the market compatible with modern systems with which it needs to interface, e.g. databases, internet, etc. In the case of WAs the poor structure and lack of documentation is a function of commercial pressures to “get the system up and running”. WA reverse engineering should also not be confused with design reuse (see for example Schwabe et al., 2001) rather it is the process of using a basic foundation design and using this to engineer many WAs [3].

A. WA Reverse Engineering Objectives:

According to Tiley(1998) there are many different aspects for reverse engineering environments and he also suggested various requirements for its implementation. He had defined some objectives for traditional reverse engineering process [4]:

- Pattern Abstraction** achieved at three levels through i)Programme Analysis ii) Plan Recognition iii)Concept Assignment
- Redocumentation:** It is the oldest form of reverse engineering in which the source code is transformed to pseudo code i.e. it is the process of generating correct documentation from existing or undocumented software.
- Architecture Recovery:** This is the process through which an understanding of the structural aspects of a system’s architecture is done.

Tiley has described three characteristics of RE process which are as follows :-

- Data Gathering:-** It is the first process of RE through which data is gathered using various techniques which include (i) systematic examination(static analysis of source code) (ii) scanning of document (iii) capturing of experience
- Knowledge Management:-** It is the process of capturing the knowledge and its organisation regarding the problem domain.
- Information Exploration:-** In this the majority of programme understanding takes place as part of the information exploration process.

The reverse engineering process of WA’s has concentrated more on static analysis of source data than dynamic analysis of WAs. There are three different areas identified by Schwabe et al. (2001) where forward engineering approach of WAs is equally applicable to reverse engineering approach of WAs:-

- Application behaviour.
- Navigational modelling.
- Interface design.

II. DIFFERENT METHODOLOGIES USED IN REVERSE ENGINEERING OF WEB APPLICATIONS

There are many methodologies available which are typically dedicated to the RE of Web application, some of which are described as under[5]:-

- a. Specific ASP.NET Methodologies
- b. UML Based Methodologies
- c. Ontology Based WA reverse engineering.

III. SPECIFIC ASP.NET METHODOLOGIES

According to Katsimpa et al.(2006) ASP.Net technology is currently being used to develop many Web Applications. There are number of reported WAs using reverse engineering systems which are specifically directed at ASP.Net applications. It comprises of a collection of ASP.NET pages(aspx files) and forms supported by further code, configuration files and XML meta-data description files. The data is analyzed using static analysis approach with the help of two parsers[6]. The first parser is applied to the .aspx files which produces a tree for each page using the underlying process :-

- a. Pre-processing of .aspx files to ensure that they are well structured through different parameters like:-
 - i) Start tags must matched with the end tags appropriately
 - ii) There should not be cross nesting of tag pairs
 - iii) There should be inclusion of quotes for tag parameters etc.
- b. Preparation of a table/matrix of nodes of interest.
- c. Creation of an ASP.NET tag trees for each well structured aspx page.
- d. Start matching labels in the table with the nodes of the tree. Where a matching node is found update corresponding entry which includes an appropriate type unit corresponding to the tag and an object ID(unique in every .aspx page) which represents the node. The result obtained is mapping of HTML and ASP controls into WebML units.
- e. Processing of text files and analyzing corresponding source code. The aim is to identify SQL keywords such as SELECT, UPDATE, etc and then consequently specific database tables. Then traverse the application directories and represent this as another tree where leaf nodes in the latter contains information from the last step.
- f. Continue adding further edges to the final tree representing links between units and pages.

IV. UML BASED METHODOLOGIES

A research was conducted and it was found that most of the WA reverse engineering methodologies and tools are based on the Unified Modelling Language (UML). UML provides a stable, familiar environment for reengineers where they work for modelling components as well as upon the behaviour of an application, which shortens the learning curve involved while integrating such methods/tools during the analysis process. There are many examples of UML-based methods which can be found in the work of different researchers such as Chung and Lee (2000), Pu et. al. (2003) and Di Lucca et. al. (2004). One very well documented

example is the WARE (Web Application Reverse Engineering) tool developed by Di Lucca (Di Lucca et al. 2004, Di Lucca et al. 2001)[7]. ‘WARE-tool’ provides support for UML methodology, which offers an extensive package for reverse engineering of web applications. This methodology is based on Benedusi’s “Goals, Models and Tools” (GMT) paradigm (Benedusi et. al., 1992) and adopts the UML extensions proposed by Conallen (1999)[7] to extract package diagrams for detailing dynamic information. The modelling phase includes the use of:

- (i) Use-case diagrams of UML which specifies the functional requirements for WAs
- (ii) Class diagrams of UML which are used to describe the structure of WAs
- (iii) Sequence diagrams of UML which are used to document the dynamic interaction.

V. METHODOGY BASED ON ONTOLOGY APPROACH:-

The ontology based approach to reverse engineering of web applications has grown in the past few years because it “provides a common referenceable set of concepts for use in communication” (van Rees, 2003). The main emphasis of this method is the use of schemas for modeling of WAs. ‘OntoWer’, approach proposed by Benslimane et. al (2006, see also Bouchiha et al 2007) has the objective to enable conceptual schemas to be created. According to the foundation work of Benslimane et al[8], current research contributions on reverse engineering tends to “focus their discussion to a particular analysis technique, without expressing its relationship to the problem of RE in general”. The ontological approach highly caters for high-level analysis of WAs.

VI. TOOLS USED IN REVERSE ENGINEERING OF WA

- A. *JSPick* (Draheim et al. 2003) :- It is a reverse engineering tool which is directed at JSP (Java Server Page to extract “page signatures and form types”[9].
- B. *VAQUISTA* (Vanderdonckt et al. 2001, Bouillon and Vanderdonckt 2002):-This tool is directed at RE the user interface of WAs which help it to migrate to different platforms[10]. The VAQUISTA process starts with the static analysis of an HTML page and goes on translating this into a presentation model describing the different elements of the HTML page. The presentation model comprises of a hierarchy of presentation elements or objects. Vanderdonckt et al. has identified four categories of presentation object:
 - i. *Concrete Interaction Objects (CIOs)*:- It comprises of real elements in the HTML page which cannot be further decomposed example text, images, animations, push buttons, check boxes, etc.
 - ii. *Abstract Interaction Objects (AIOs)*:- It includes abstraction of CIOs identified by a name, attributes and other parameters.
 - iii. *Logical Window (LW)*:- It is the logical container for AIOs or a physical window (dialog box, check box, etc).

- iv. **Presentation Unit (PU)**:- It provides the presentation environment for undertaking an interactive task. Transformations are undertaken using mapping tables that can be used in a variety of ways as are directed by the user. (See also Vanderdonckt and Bouillon, 2002)[11].
- C. **ReWeb**: (Ricca and Tonella, 2000, 2001,2002) This tool undertakes traditional source code analysis of WAs by representing the WA as a graph structure and undertaking various types of analysis such as reachability, flow and traversal analysis. This tool can download and analyse various WAs. There are many search and navigation tools included in it and there is also an option to illustrate the evolution of WAs using a colour coding scheme. Popup windows appears to display the outcome of the analysis[12].
- D. **TERESA** (Paganelli and Paterno, 2002):-It is a statistical analysis tool the which analyses source code and then produces a task-oriented model of a Web application[13].
- E. **Revangie** (Draheim et al.2005). :- It is a black box approach to Reverse engineering of web applications which is independent of source code. It uses “typed bipartite state machines” which is a form-oriented user interface model that uses client pages and server actions. The models are graphs that include the relationships between server-side actions and pages [14]. Revangie has three modes of operation depending on the degree of user interaction:
- (i) Crawl mode:-It operates on the client side.
 - (ii) Snoop mode:- It operates at points within the communication line between server and client.
 - (iii) Guide mode:- It Is a combination of the crawl and snoop modes.

VII. CONCLUSIONS

The creation of better designs and the interoperability of existing products often begin with reverse engineering. Although a substantial amount of work has been undertaken and reported in the literature on “traditional” reverse engineering, little work has been done on the reverse engineering of WAs . What is also clear is that the field of WA reverse engineering is still in its infancy with little agreement on standardised approaches or methodologies. In doing so, it uncovers as much information as possible about the design ideas that were used to produce a particular WA. The reversed engineering of WAs requires different procedures than that required for other software systems. This is because WAs are internet based and therefore are developed using a specific set of technologies not found in “stand alone” software systems. Another distinction is that the objective of reverse engineering is to reengineer the system as applied to other applications which tends to be in the context of legacy is usually not the case with WAs which are more recent and therefore fully operational, the aim here is to find out how it works (with a view to future

maintenance) because the WA was not properly documented when it was first developed

VIII. REFERENCES

- [1]. T. Munakata, “Knowledge discovery”, Communications of the ACM, 42, 11, pp. 26-29, November 1999.
- [2]. Chikofsky, E.J. and J.H. Cross, "Reverse Engineering and Design Recovery: Taxonomy in IEEE Software". IEEE Computer Society: 13–17, 1990.
- [3]. Benedusi, P., Cimitile, A., De Carlini, U. (1992). Reverse Engineering Process, Document Production and Structure Charts. Journal of Systems and Software, Vol. 19, pp225-245.
- [4]. Tilley, S. (1998). A Reverse Engineering Environment Framework. Tech Report, Carnegie Mellon Software Engineering Institute. (CMU/SEI) Technical Report
- [5]. <https://www.csc.liv.ac.uk/research/techreports/tr2007/ulcs-07-017.pdf>
- [6]. Katsimpa, T., Panagis, Y., Sakkopoulos, E., Tzimas, G., Tsakalidis, A. (2006) Application Modelling using Reverse Engineering Techniques. Proceedings Symposium on Applied Computing (SAC'06), pp1250-1255.
- [7]. Di Lucca, G.A., Di Penta, M., Antonniol, G. and Casazza, G. (2001). An Approach for Reverse Engineering of Web-Based Applications. Proc. 8th Working Conference on Reverse Engineering, WCRE'01, IEEE, pp231-240.
- [8]. Conallen, J. (1999). Building Web Applications with UML. Addison Wesley. ISBN: 0-201-61577-0.
- [9]. Benslimane, M. S., Malki, M., Bouchiha, D., Benslimane, D. (2006). OntoWer: An Ontology Based Web Application Reverse Engineering Approach. International Review on Computers and Software, 1(1), pp52-58.
- [10]. Draheim, D., Fehr, E. and Weber, G. (2003). JSPick – A Server Pages Design Recovery. Proc7th IEEE European Conference on Software Maintenance and Reengineering, LNCS, pp230-236.
- [11]. Bouillon, L. and Vanderdonckt, J. (2002). Retargeting of Web Pages to Other Computing Platforms with VAQUISTA. Proc 9th Working Conference on Reverse Engineering, WCRE'02, IEEE, pp339-348
- [12]. Tonella, P., Ricca, F., Pianta, E. and Girardi, C. (2002). Restructuring Multilingual Web Sites. Proc. 18th IEEE Int. Conf. on Software Maintenance (ICSM'02), pp290-299.
- [13]. Paganelli, L. and Paterno, F. (2002). Automatic Reconstruction of the Underlying Interaction Design of Web Applications. Proc. 14th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE'02), ACM, pp439-445.
- [14]. Draheim, D., Lutteroth, C. and Weber, G. (2005). A Source Code Independent Reverse Engineering Tool for Dynamic Web Sites. Proc. 9th European Conference on Software Maintenance and Reengineering (CSMR'05), pp168-177.