Reverse Engineering Web Applications

Porfirio Tramontana

Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II
Via Claudio, 21, 80125 Napoli, Italy
ptramont@unina.it

Abstract

The heterogeneous and dynamic nature of components making up a Web Application, the lack of effective programming mechanisms for implementing basic software engineering principles in it, and undisciplined development processes induced by the high pressure of a very short time-to-market, make Web Application maintenance a challenging problem. A relevant issue consists of reusing the methodological and technological experience in the sector of traditional software maintenance, and exploring the opportunity of using Reverse Engineering to support effective Web Application maintenance.

The Ph.D. Thesis presents an approach for Reverse Engineering Web Applications. The approach include the definition of Reverse Engineering methods and supporting software tools, that help to understand existing undocumented Web Applications to be maintained or evolved, through the reconstruction of UML diagrams. Some validation experiments have been carried out and they showed the usefulness of the proposed approach and highlighted possible areas for improvement of its effectiveness.

1. Introduction

Web Applications are complex software systems providing to users access to Internet contents and services. In the last years they have had a large diffusion due to the growing of the diffusion of World Wide Web: nowadays the quantity of information and services available on the Internet is very remarkable, technologies have a continue evolution and existing Web Applications show many aging signs.

The complexity of the functions provided by a Web Application has also increased since, from the simple facility of browsing information offered by the first Web sites, a last generation Web Application offers its users a variety of functions for manipulating data, accessing databases, and carrying out a number of productive processes.

The increased complexity of the functions implemented by Web Applications is achieved with the support of several different technologies. Web Applications generally present a complex structure consisting of heterogeneous components, including traditional and non-traditional software, interpreted scripting languages, compiled languages, HTML pages, databases, images and other multimedia objects. A Web Application may include both 'static' and 'dynamic' software components. 'Static' components are stored in files, whereas 'dynamic' components are generated at run time on the basis of the user inputs.

The high pressure of a very short time-to-market often forces the developers of a Web Application to implement the code directly, using no disciplined development process, and this may have disastrous effects on the quality and documentation of the delivered Web Application. Poor quality and inadequate documentation have to be considered the main factors underlying ineffective and expensive maintenance tasks, burdened by the impossibility of applying more structured and documentation-based approaches.

Reverse Engineering methods, techniques and tools have proved useful to support the post delivery life-cycle activities of traditional software systems, such as maintenance, evolution, and migration. The software community is seriously addressing the problem of defining and validating similar approaches for Web Applications. Reverse Engineering allows to recover and abstract documentation from an existing Web Application, to achieve comprehension, to assess quality factors, and so on.

The remaining part of the synopsis is organized as follows: in section 2 the aim of the Ph.D. Thesis is traced; in section 3 the main contributions of the Ph.D. Thesis are reported; section 4 contains indications about the current and future extensions of the work.

2. Aim of the Thesis

The aim of the Thesis is to propose and realize an approach for the Reverse Engineering of Web Applications by extracting information and abstracting documentation describing the physical and conceptual structure of the application.

Reverse Engineering Web Applications is a complex task, due to the variety of languages and technologies used together. However, the benefits that can be obtained are remarkable: the availability of documentation at different abstraction levels will provide a useful help in maintenance interventions, migration and reengineering processes, and it may contribute in reducing their costs and risks and improving their effectiveness.

Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM'05)
1063-6773/05 $20.00 © 2005 **IEEE**

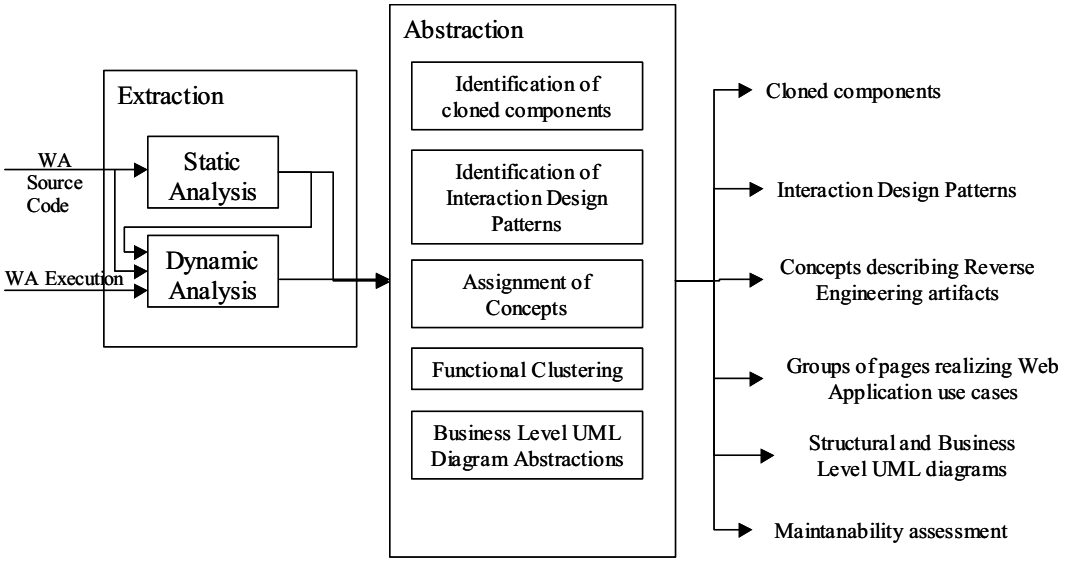

Figure 1: Web Application Reverse Engineering process

3. Main contributions

The Thesis tackles with many problems in the field of Reverse Engineering of Web Applications, providing significant contributions.

The main topics discussed in the Thesis are summarised in Figure 1 that also depicts the Reverse Engineering process defined in the Thesis.

Modelling Web Applications

A first problem to solve in a Reverse Engineering process is to define a model of the application. Thus, a first contribution of this Thesis is the model describing the structure of a Web Application [6] that represents it by a UML class diagram, modelling the application's pages, the relationships among pages, and the inner sub-components of pages. The proposed model extends the one of Conallen [1].

This model has been considered as the reference model to drive the Reverse Engineering process of a Web Application.

Extraction process

The proposed Extraction process consists of the analysis of the source code of Web Application static components (client and server pages, script modules). Static and dynamic analysis are executed to extract the needed information.

It is supported by a tool, called WARE (Web Application Reverse Engineering, [2]), which architecture is shown in Figure 2. This tool contains parsers that analyse the source code of Web Applications (ASP, PHP, HTML, Javascript, VBScript, JScript languages are supported) and stores the extracted information in a relational database (according to the proposed Web Application model). Static analysis is automatically executed by WARE, while the tool just provides a support for dynamic analysis (human intervention is required in this case).

One of the outputs of WARE is the UML class diagram depicting the structure of the Web Application (according to the model defined in [6]).

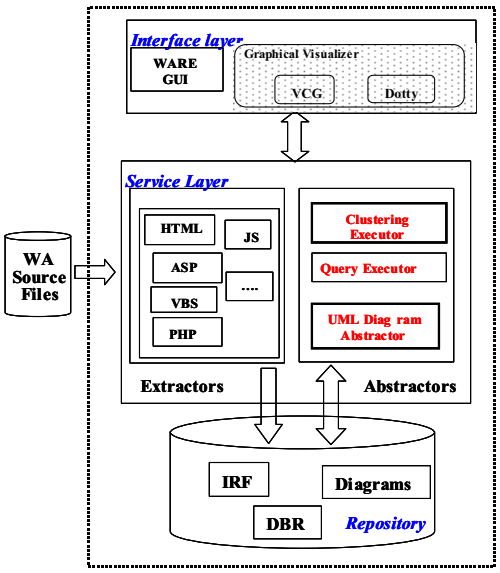

Figure 2: Architecture of tool WARE

Functional Clustering

A methodology is proposed to cluster the pages of a Web Application in subsets, maximizing the subset cohesion and minimizing the coupling between the subsets [3]. A clustering criterion has been defined by taking into account the typology and the topology of the relationships between the pages of the Web Application. A heuristic agglomerative hierarchical algorithm, similar to the algorithm proposed by Mancoridis et al. for the clustering of components of Object Oriented applications [12], has been developed.

Each cluster of pages is, then, associated to a use case provided by the Web Application. The tool WARE supports the automatic execution of the clustering algorithm. It also provides the visualization of diagrams showing the resulting clusters. Experiments have proven the usefulness and validity of the clustering algorithm in the identification of subsets of pages that collaborate to implement use cases of a Web Application.

Clone analysis

The existence of groups of cloned pages in a Web Application may suggest the reengineering to eliminate page duplications. Clone analysis techniques have been adopted to evaluate the degree of similarity between Web Pages [7]. Some different metrics have been proposed, such as metrics based on comparing between the HTML structures of the pages (specializing Levenshtein distance metric [11]) and metrics based on the Euclidean distance between appropriate feature vectors. A tool has been developed to automatically evaluate these metrics on the basis of the information extracted by the tool WARE.

Clone analysis has revealed its usefulness in different tasks, such as: the identification of plagiarisms, the identification of cloned components, the comparing between different versions of the same Web Application.

Web Interaction Design Patterns identification

The problem of identification of patterns that recur in the implementation of Web Application user interfaces have been addressed proposing a methodology to identify the presence of some common Web Interaction Design Patterns (a catalogue of this patterns is reported in [14]). The proposed methodology [10] is based on the identification, in a client page, of features characterizing the patterns. These features regard either structural aspects of the interface (such as the presence of tables, frames, forms, anchors and so on) or semantic aspects (such as the presence of particular words and textual expressions that are commonly used implementing a specific pattern). The identification process is based on continuous training: the system learns about the correlation between features and patterns analysing a set of pattern samples. A tool, exploiting the WARE repository, has been implemented. The methodology has been validated on a number of Interaction Design Patterns and it has shown a good degree of precision. The identification of Web Interaction Design Patterns has revealed its usefulness in the comprehension and for the reengineering of Web Applications.

Concept Assignment

To achieve further comprehension about the semantic of the artifacts resulting from the Reverse Engineering of Web Applications, the Concept Assignment problem has been addressed. The textual contents of static and dynamic client pages have been analysed using Information Retrieval techniques [8] in order to recognise concepts included in them. In particular, concepts retrievable in a Web Page have been ranked on the basis of their editing format, used to visualize the text in a browser, and of their frequency of occurrence. The recovered concepts are, then, exploited to support the assignment of concepts to Reverse engineering artifacts.

A tool has been produced to support this methodology: the results obtained in some experiments showed the validity of this automatic support.

Business Level UML Diagrams Abstraction

Methodologies have been proposed to abstract UML use case, class and sequence diagrams at business level [5].

To recover business level class diagrams, attributes, methods and relationships are identified by analysing the data a user inputs by a form, the data exchanged between Web Application pages, the data flow between the application and the databases. Class methods are identified by analysing the functions implemented by cluster of pages (recovered applying the proposed functional clustering technique). Relationships between classes are identified analysing the associations and the data flow among pages.

Experiments carried out have shown that UML diagrams abstracted using this technique are very similar to the ones produced along the development process of analysed Web Applications.

Maintainability assessment

The pieces of information extracted and abstracted have also been used to evaluate some quality characteristics of Web Applications. In particular, a model for the assessment of the maintainability of a Web Application has been proposed [9]. This model is an adaptation of the one proposed by Oman and Hagemeister [13] for traditional applications. This model comprehends a set of software metrics evaluating the maintainability of a Web Application under different aspects.

The proposed set of metrics may be automatically evaluated: a tool evaluating them on the basis of the information stored in the WARE repository has been developed. The proposed maintainability model has shown its usefulness by providing information about the complexity of a maintenance intervention on a Web Application.

4. Works in progress

Current work is addressing the problem of a better integration of the methods and of the tools that have been developed.

In this Thesis, the automatic extraction of information from the source code of Web Applications is mainly based on static analysis techniques. The problem of the automatic extraction of information from the analysis of the execution of a Web Application is now addressing. Tools have to be developed to provide the needed automatic instrumentation of the source code of Web pages (the

information extracted by static analysis can be a useful support to this task) and to recover information from the execution of the instrumented Web Applications.

Another current work is the definition of methods and techniques to support the migration or the integration of Web Applications to/with Web Services. Also in this case, the information abstracted and recovered with the methods proposed in this Thesis, is a good starting point to define reengineering and migration processes.

The Reverse Engineering approaches and tools presented in this Thesis have been used to support the testing of Web Applications [4]. Nowadays, some other testing tools are under developing, supporting the automatic testing of Web Application.

As regards the quality assessment, an extension of the proposed maintainability model is needed to assess the aging of a Web Application and to assess the efficacy of reengineering intervention aiming at the improvement of Web Application maintainability.

Finally, a methodology and a tool to address the evaluation of Web Application accessibility, on the basis of information extracted with the Reverse Engineering process, are under developing.

Acknowledgements

I would like to thank peoples that supported me during this period: my Tutors, Professors Ugo De Carlini, Giuseppe Di Lucca and Anna Rita Fasolino. They advised me continuously, with constancy and patience. Finally, I would to thank Fabio Pace that shared with me the Laurea Degree Thesis work, and the ideation of the first fundamental contributions of this Ph.D. Thesis.

References

[1] J. Conallen, "*Building Web Applications with UML"*. Addison Wesley Publishing Company: Reading, MA, 1999.

[2] G.A. Di Lucca, A.R. Fasolino, U. De Carlini, F. Pace, P. Tramontana, "*WARE: a tool for the Reverse Engineering of Web Applications"*, in *Proceedings of 6th European Conference on Software Maintenance and Reengineering* – CSMR 2002, IEEE C.S. Press, pp. 241-250.

[3] G. A. Di Lucca, A.R. Fasolino, U. De Carlini, F. Pace, P. Tramontana, "*Comprehending Web Applications by a Clustering Based Approach*", Proc. of *10th IEEE Workshop on Program Comprehension,* IWPC 2002, IEEE C.S. Press, pp. 261 - 270

[4] G.A. Di Lucca, A.R. Fasolino, F. Faralli, U. De Carlini, "*Testing Web Applications*", in *Proceedings of International Conference on Software Maintenance,* ICSM 2002, IEEE C.S. Press, pp. 310-319.

[5] G. A. Di Lucca, A.R. Fasolino, U. De Carlini, P. Tramontana, "*Abstracting business level UML diagrams from web applications*", Proc. of 5th *IEEE Workshop on Web Site Evolution*, WSE *2003*, IEEE C.S. Press, pp. 12-19

[6] G.A. Di Lucca, A.R. Fasolino, P. Tramontana, "*Reverse Engineering Web Application: the WARE approach*", Journal of Software Maintenance and Evolution: Research and Practice, Volume 16, Issue 1-2, Date: January - April 2004, Pages: 71-101

[7] G.A. Di Lucca, A.R. Fasolino, P. Tramontana, U. De Carlini, "*Identifying Reusable Components in Web Applications*", IASTED International Conference on Software Engineering, SE 2004, pp.526-531

[8] G.A. Di Lucca, A.R.Fasolino, P.Tramontana, U.De Carlini, "*Supporting Concept Assignment in the Comprehension of Web Applications"*, Proceedings of the 28th IEEE Annual International Computer Software and Applications Conference, COMPSAC 2004, IEEE C.S. Press

[9] G.A. Di Lucca, A.R.Fasolino, P.Tramontana, C.A.Visaggio, "*Towards the definition of a maintainability model for web applications*", Proceedings of the Eighth IEEE European Conference on Software Maintenance and Reengineering, CSMR 2004, IEEE C.S. Press, pp. 279 - 287

[10] G.A. Di Lucca, A.R. Fasolino, P. Tramontana, "*Recovering Interaction Design Patterns in Web Applications*", Proceedings of the 9th IEEE European Conference on Software maintenance and Evolution, CSMR 2005, IEEE C.S. Press, pp. 366-374

[11] V. I. Levenshtein, "*Binary codes capable of correcting deletions, insertions, and reversals"*, *Cybernetics and Control Theory* 10 (1966), 707-710.

[12] S. Mancoridis, B.S. Mitchell, C. Rorres, Y. Chen and E.R. Gansner, "*Using automatic clustering to produce high-level system organizations of source code*", *6th International Workshop on Program Comprehension,* IWPC 1998, IEEE C.S. Press

[13] P. Oman, J. Hagemeister, "*Metrics fo Assessing a Software System's Maintainability"*, Proceedings of IEEE International Conference on Software Maintenance, ICSM 1992, IEEE C.S. Press

[14] M. van Welie, G. C. van der Veer, "*Pattern Languages in Interaction Design: Structure and Organization*", Proceedings of Ninth International Conference on Human-Computer Interaction, Interact 2003, pp. 527-534