ISSN No. 0976-5697

Volume 2, No. 2, Mar-Apr 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Wrapper – Based Design Conversion in Database Re-Engineering

Dr. E. Kirubakaran
Sr. Deputy General Manager, Outsourcing,
B.H.E.L., Trichy-14, India
e_kiru@yahoo.com

S Manimekalai*
Research Scholar, MTWU, Kodaikannal.
Lecturer in Computer Science,
Government Arts College, Trichy-22, India
mega_somu@yahoo.com

Abstract: Now-a-days, all the organizations maintain their information through computers. To computerize the information, the database is needed. A database may contain multiple tables, which in turn, it contains lot of fields of different data types to organize the information. In order to store the information in these tables, one must be aware about the knowledge of databases, tables, fields and types.

In the Software Development Architecture, there exist a number of phases to develop the product. The first and essential step is to design the product based on the requirements. The next step is to develop code to implement the design. The designing process is done by a Software Architect, who is not necessary to know about the programming language in which the design implements. Most of the Software Architect may have the knowledge in HTML (Hyper Text Markup Language) which is easy to learn and understand. But the programmer must know and understand the design process and then they can convert it into implementation. Sometimes, there may arise some complexity due to inefficient design which was developed by a Architect who was unaware of the programming language.

To reduce this kind of complexity and to simplify the work, a special technique is incorporated in this paper, which is termed as "**Re-engineering**". This technique will automatically convert the table design into implementation code. In this paper, we 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 creates a table design with necessary fields and data using HTML. The programmer uses the Re-Engineering technique to transform it into database in Java. The snapshot about the conversion from HTML to Java is discussed in this paper.

Keywords: Computerize, Database, Design, Fields, Implement, Organize, Programming language, Re-Engineering, Snapshot, Software Architect, Software Development Architecture, Tables.

I. INTRODUCTION

Re-Engineering is the technique to convert the process from one form to another. In this paper, we implement the technique in databases. Databases are the ultimate source to store and maintain the data. It is also easy to retrieve the data from these databases than other source. A database may contain a collection of related tables with different fields of different data type. In this paper, to organize information in the database, we implement the concept of Re-Engineering.

In the Software Development Architecture, to develop a product, it needs a team work. The team members will develop a product with co-ordination. In a team, some members have done the design work for the product. And this design has to be converted to the implementation phase by some other members who are well-versed in the technology. The designer may not need to know about the technology, since they only concentrated on the design phase. The programmer has to convert the design into implementation.

This process becomes complex in some situation when there occurs a frequent changes in the fields or the types of the fields. This complexity arises to the programmer since, he has to update the database frequently to update the changes in the fields and types of the table. It becomes time consuming and it requires more time to do it by proper analysis of the data.

To reduce the complexity and to simplify the process, the re-engineering technique is incorporated. When we implement this technique, the conversion is done automatically. The Software Architect just designs the table with necessary formats such as what style they want and the name of the table, number of fields, specify data types for each field and the kind of database operations such as Add, Modify, Delete and View. They create the table like a template with sample data. They need not care about the programming language in which it is to be implemented, no database knowledge and no idea about the process of the table. After designing the table, when we apply the re-engineering, the design will convert into implementation by the following steps:

- The design template has to be analyzed.
- Create a table in any one of the databases such as Ms-Access, SQL Server, and Oracle and so on. This can be common to all kinds of process and so it is done by using Open DataBase Connectivity (ODBC).
- The fields are created in the table with their corresponding data types.
- The sample data are stored in the table.
- Code generates automatically using JDBC in Java.
- Based upon the Database Operations specified in the design, the operation performs automatically without any query written by the user.

These processes are generalized by specifying the type of operations performed, number of fields required in the output and the number of rows displayed in each page.

Using this information, the process is held automatically and the result is obtained.

The process of the conversion is shown graphically:

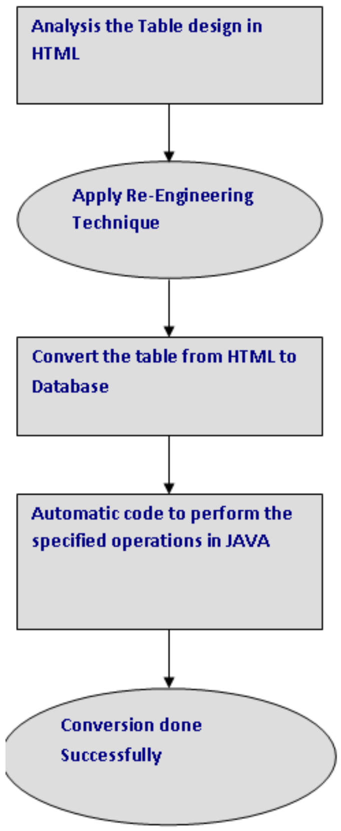
Analysis the Table design in HTML
Apply Re-Engineering Technique
Convert the table from HTML to Database
Automatic code to perform the specified operations in JAVA
Conversion done Successfully

Figure-1 Conversion of Database using Re-Engineering

This Re-Engineering technique also simplifies the process of the database operations using some implementations. Only the requirements are produced, the query will automatically generate and the result is produced. Thus in this paper, how the database are converted and how the information are retrieved from the table automatically without any query written by the programmer are analyzed using re-engineering.

II. RELATED WORK

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:

In paper [1], Gray and Fiddian states that, Multimedia applications (MMA) of database management systems (DBMS) pose significant problems because they push the technology to its limits by adding requirements that are hard to satisfy by any computing system. Examples of database-related problems include: supporting new abstract data types (“3 5-mm slide”, “video-frame”), complex description expressions (“video frame of aircraft landing at sunset”), the handling of new data objects by a query language, and the delivery of binary large objects (BLOBS) subject to varying optimization criteria (real-time video, cost effective download for later consumption, and so on). The aim of that paper is to break up the requirements into those that are specific to multimedia (and specific to certain types of multimedia) and those that are specific to DBMSs. They propose an architecture that separates three major sets of requirements: firstly, the environment needed for authoring, indexing and storing multimedia applications; secondly, the environment needed for searching multimedia information systems; and thirdly, the delivery requirements of certain types of multimedia applications. Their contribution to that discussion is to show which aspects of MMAs are covered by standards, methods and techniques already in place for DBMSs in general, and which aspects need specific consideration in the MMA context.

An MTS uses that principle by applying rules that convert parsing information into canonical expressions, and by matching the canonical expressions with attributed, abstract syntax trees from which target language Statements are constructed. The major ingredients of an MTS are a canonical model of the application domain in which translation is to be done, and a generic mapping specification language to declare the transportation rules of an appropriate AG. An excellent book on the theory of AGs is in paper [3]. A report by the authors of that paper, explaining the formal relationship between AGs and MTSs is in preparation [2].

In the MIPS research [4], a knowledge based system was used to connect domain specific terminology with appropriate multimedia data types [6]. The importance of authoring tools was acknowledged and a flexible set of templates based on SGML/HyTime was defined to attach multimedia data to application domain specific presentational primitives [5]. The meta-knowledge about the target data was expressed in IRL, a proprietary internal representation language. A query language extending the capabilities of SQL was also defined over the IRL. A basic interface between relational databases and the KBS was developed, but there was insufficient separation of low-level database access versus high-level querying at the domain-specific “knowledge level”. However, the MIPS architecture illustrates very well the major ingredients of any multimedia environment.

In the paper[7], they reports on a recent data reengineering research, the goal of which was to migrate a large CODASYL database towards a relational platform. The legacy system, made of one million lines of COBOL code, is in use in a Belgian federal administration. The approach followed combines program transformation, data reverse engineering, data analysis, wrapping and code generation techniques.

The paper [7] also states the main purpose of the refactoring phase is to facilitate the migration phase itself by isolating the data manipulation primitives from the legacy source code. Indeed, no further alteration of the refactor legacy programs will be necessary at the time of migrating the database. The system refactoring phase is supported by two distinct tools. The first one is a plug-in of DB-MAIN [8] that generates the additional procedures from the source IDS/II schema.

The second tool, implemented on top of the ASF+SDF Meta-Environment [9], allows the transformation of the legacy programs. In the second phase (system migration), the legacy database is migrated and the additional procedures are regenerated such that they now access the target database through inverse wrappers [10]. These wrappers are dedicated data access components providing the legacy programs with a legacy API to the migrated data.

The implicit constructs are identified through legacy code inspection. Dataflow analysis techniques are used to understand the way applications use and manage the data, which allows recovering such implicit constraints as foreign keys, finer-grained record decomposition, and more expressive names for records and fields [11]. Obviously, the implicit constructs recovered by program analyzers are only candidate integrity rules that still have to be validated. It can be done via user validation or data analysis. The target relational schema is then derived from the recovered conceptual schema using a standard database design methodology [12]. It design phase is supported by DB-MAIN, offering a set of assisted schema transformations.

In paper [1], they discussed the main drawback of that approach concerns performance. The resulting programs are between 3 and 6 times slower than the original application, depending on the type of database access. In paper [13], to develop a flexible intranet-based cardiology information system, they reengineered that existing information management and database system using computer aided systems engineering (CASE) tools. Initial&, abstract data and function modeling was employed to create entity- relationship descriptions and business process definitions within a central CASE repository. Database design wizards were then used to automatically generate a prototype database description. After design adjustments, a generator automatically compiled and executed a set of consistent DDL statements that generate the database instance according to specific vendor format. Business functions were transformed into application Reengineering the information system with CASE technology has allowed us to (1) increase flexibility for changes in the database design, (2) reduce client administration tasks (3) facilitate system maintenance by adopting intranet technology and (4) reduce requirements for user support through the use of well established user interfaces.

Different strategies that stress functional design have recently been applied successfully in the business world. Among these, business reengineering which aims at a "radical redesign of business processes to bring about dramatic improvements in performance" [14] has gain wide popularity. System development was performed according to the CASE*METHOD [15, 16] which is a structured approach with detailed specifications of development phases, associated tasks and deliverables. CASE*METHOD pursues an integrated design philosophy targeting both structure and function of a system.

Standard hardware and software components were used to implement the information system. The central production system is based on an ORACLE 7.3 database under Windows NT 3.5 1. Client modules were generated for Windows and Macintosh computers using ORACLE Forms as presentation service. A print server on the central server permits generation of reports independent of the type of client that requests the report. An interface to the hospital information system to transfer patient data was implemented using ODBC data access. While Web clients in principle can already be generated, the graphical options in the current version are still limited in comparison to proprietary graphical clients. However, when a multi-tiered system architecture will become available, the system can quickly be adapted to provide its full functionality also within the Web standards which is increasingly used for medical information systems. [17, 18, 19]

In paper [20], Web Applications are subject to continuous and rapid evolution. Often it happens that programmers indiscriminately duplicate Web pages without considering systematic development and maintenance methods. This practice creates code clones that make Web Applications hard to maintain and reuse. This paper presents an approach for reengineering Web Applications based on clone analysis that aims at identifying and generalizing static and dynamic pages and navigational patterns of a web application. Clone analysis is also helpful for identifying literals that can be generated from a database. A case study is described which shows how the proposed approach can be used for restructuring the navigational structure of a Web Application by removing redundant code.

The speed of development of WAs means that they often have been produced with scant regard to established software engineering principles and are subject to continuous evolution [21, 22, and 23]. As a result, the current situation is somewhat similar to the early development of software systems: like legacy systems, these applications generally result in very poor quality products, with a "spaghetti-like" structure and no documentation. This makes them hard to test, maintain and reuse [24, 25, 21, 26, and 27].

Recently, researchers have extensively studied the problem of defining reverse engineering techniques and
Tools for analyzing WAs and help during their maintenance [24, 28, 29, 30, 26]. Ricca and Tonella [26] propose ReWeb, a tool for analyzing the structure and the evolution of static Web sites. They define a conceptual model for representing the structure of a Web site and several structural analyses relying on such model, from flow analysis to graph traversal algorithms and pattern matching. They also represent the evolution of a Web site by using colors as time indicators. This is useful to determine how the original structure of a Web site degrades during its life. To fully understand the behavior of a WA it is necessary to detect the dynamic interactions among its components. For the reverse engineering of Was approaches and tools have been proposed in [29, 30, 31]. It enables to extract the Conallen

extension [32] of UML diagrams from existing WAs, analyzing both static and dynamic contents.

III. METHODOLOGY

A. *Proposed Method*

The proposed method of this paper is discussed below which explains about the process of conversion of table from HTML to Database such as Ms-Access, SQL Server and Oracle. It also explains about how the database operations are performed automatically without using any query.

The Software Architect designs the table in HTML which is easy for them, since they had no knowledge about the technology. This design is analyzed and transformed into database. This conversion does not eliminate anything that was designed by the designer.

The operations are performed automatically in spite of writing the query for each operation. So, when the fields or types are changed, there is no need to update the code, since it is generated automatically. The process is done as follows:

- ➢ A list box contains a list of Back-ends.
- ➢ On selecting an item from the list, a list of databases from the selected item are added and displayed in another list box automatically.
- ➢ Upon choosing the database, another list box containing a list of tables is generated.
- ➢ On selecting the table, the types of database operations are shown to the user to choose the type of operation in which they want to perform such as insert, update, delete and select.
- ➢ After selecting the operations, the list of fields in the selected tables are shown to them. From this, they can select the fields and types to carry out the operations selected.
- ➢ After that, the values are provided to perform the selected operations.

The pictorial representation is shown below in figure 2.

Thus the process is carried out automatically and graphically without the need of any kind of query written by the programmer or user. The proposed method contains some algorithm to implement the technique and process the operations in the database.

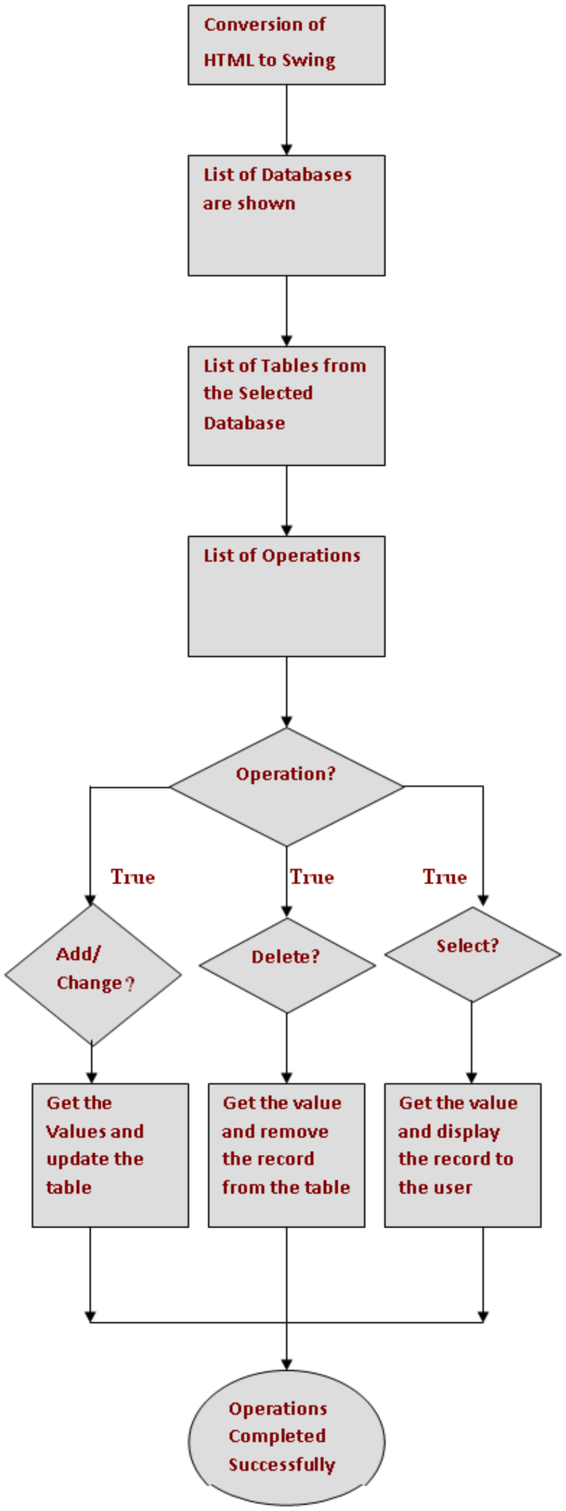
Conversion of HTML to Swing
List of Databases are shown
List of Tables from the Selected Database
List of Operations
Operation?
True
True
True
Add/ Change?
Delete?
Select?
Get the Values and update the table
Get the value and remove the record from the table
Get the value and display the record to the user
Operations Completed Successfully

Figure-2 Proposed Method using Re-Engineering

B. *Algorithm*

Start the process

Step-1: Analysis the design in HTML

Step-2: Convert the design into Database using Re-Engineering technique with same name and fields as defined in the design of HTML into Swing.

Step-3: To Process the Database

```
Dbcount = No. of databases in the selected back-end
For i = 0 to Dbcount-1
{
  Db[i] = Select the database names
  Add Db[i] to the Listbox1
  i++
}
Dbname = Selected database from the Listbox1
Identify and Open the Database from the list
Tbcount = No. of tables in the selected database
For j = 0 to Tbcount-1
{
  Tb[j] = Select the table names
  Add Tb[j] to the Listbox2
  j++
}
Tbname = Select the table from the Listbox2
Identify and Open the selected table
Fieldcount = No. of fields in the selected table
For k = 0 to Fieldcount-1
{
  Field[k] = select the field name
  Add Field[k] to the Listbox3
  k++
}
View the Field details with their datatypes
Display the list of database operations
Operation = select the operation from the list
If(Operation == "Add")
{
  Get the input values to all the fields
  Check the input with the datatypes
  Add into the table
}
Else if(Operation == "Change")
{
  Get the input values which we want to change
  Change the value in the table
}
Else if(Operation == "Delete")
{
  Get the input value which we want to remove
  Identify the record
  Remove the record from the table
}

Else if(Operation == "Select")
{
  Get the input value which we want to view
  Search in the table for the input value
  Retrieve the record and show it
}
Operations completed successfully
Stop the process.
```

C. *Algorithm Explanation*

The algorithm stated above will works as follows: First, the design from the HTML is converted into Database in Swing. The Re-Engineering technique is applied to list the database, tables, and fields to the user in the listbox. On selecting each one, the values are stored to carry out the process. Finally, the user can select an operation from the list of database operations. Based upon their selection, the operation will take place in the database. The operations may be to add a new record into the database (Insert), change the record in the database (Update), Remove the record from the database (Delete) or to view the record from the database (Select). Thus the re-engineering technique will simplify the process.

IV. EXPERIMENTAL RESULTS

The technique of Re-Engineering is used to convert the process from one form to another. In this paper, this technique is implemented to convert the table from HTML design to Java code.

The designer may be unaware of the programming language and so they will create the design of the table in HTML using **<Table>.** The design and the data are added using HTML tags by the designer. It is then converted into any programming language by the programmer. In this case, they use Re-Engineering technique. Thus the table is converted to **Swing in Java** in this paper. The table can be created by using the **JTable** class. Thus the table is created and designed as done in HTML.

To done the process in database, it also uses the Re-Engineering technique. Using this, the databases are listed to the user or programmer. On selecting the database, the list of tables in the selected databases is showed to the programmer and then it selects the fields in the selected table. Then the database operations are selected to carry out the process automatically using Re-Engineering. Thus the code written by the programmer is reduced and the process is done automatically.

Thus the Re-Engineering technique is carried out in this research by converting the database from **HTML** to **SWING** and is implemented in the process of converting the databases

V. CONCLUSION

The aim of the paper is to convert the database from Programming language to another. In this paper, the conversion is done by converting the database from HTML to Swing. This process is done by using Re-Engineering technique.

In this paper, 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, it will not necessary to change the code, since it is done automatically using this technique.

In future, this can be enhanced by 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.

VI. REFERENCES

[1] "DATABASE INTEROPERATION SUPPORT IN MULTIMEDIA APPLICATIONS ARCHITECTURE AND METHODOLOGY" by W A Gray, N J Fiddian, W Behrendt, Department of Computer Science, Cardiff University, UK.

[2] [BehFidGra99] Bebrendt W, Fiddian N J, Gray W A, "Theory and Practice of Meta-Translation Systems", Technical Report, Department of Computer Science, Cardiff University, 1999, (in preparation).

[3] [KueVog97] Kuehnemann A, Vogler H, Attributgrammatiken - Eine grundlegende Einfuehrung, (in German), Vieweg Verlag, Braunschweig; Wiesbaden, 1997.

[4] [Wi195] Wilson M D, "Enhancing Multimedia Interfaces with Intelligence", In Vince J A, Earnshaw R A (Eds), Multimedia Systems and Applications, Academic Press, London, 1995.

[5] [Mac951 Macnee C A , Behrendt W, Wilson M D, Jefiey K G, Kalmus J R, Hutchinson E K, "Presenting DynamicallyExpandable Hypermedia", Information and Software Technology, 37,7, pp. 339-350, Elsevier Science, 1995.

[6] [Jef94] Jeffery K G, Hutchinson E K, Kalmus J R, Wilson M D, Behrendt W, Macnee C A, "A Model for Heterogeneous Distributed Database Systems", Proceedings of 12" British National Conference on Databases (BNCOD 12), Cambridge University Press, 1994.

[7] "Large-scale Data Reengineering: Return from Experience" by Jean Henrard and Didier Roland, ReveR S.A, 24, Boulevard Tirou, 6000 Charleroi, Belgium, {jean.henrard,didier.roland}@rever.eu and Anthony Cleve and Jean-Luc Hainaut, University of Namur, 21, rue Grandgagnage, 5000 Namur, Belgium, {acl,jlh}@info.fundp.ac.be

[8] DB-MAIN. http://www.db-main.be, 2006.

[9] M. van den Brand, A. van Deursen, J. Heering, H. de Jong, M. de Jonge, T. Kuipers, P. Klint, L. Moonen, P. Olivier, J. Scheerder, J. Vinju, E. Visser, and J. Visser. The ASF+SDF Meta-Environment: A component-based language development environment. In Compiler Construction (CC '01), volume 2027 of LNCS, pages 365–370. Springer-Verlag, 2001

[10] A. Cleve, J. Henrard, D. Roland, and J.-L. Hainaut. Wrapperbased system evolution application to codasyl to relational migration. In Proc. of the 12th European Conference on Software Maintenance and Reengineering (CSMR'08), pages 13– 22. IEEE CS Press, 2008.

[11] A. Cleve, J. Henrard, and J.-L. Hainaut. Data reverse engineering using system dependency graphs. In Proc. of the 13thWorking Conference on Reverse Engineering (WCRE'06), pages 157–166, 2006.

[12] T. J. Teorey, D. Yang, and J. P. Fry. A logical design methodology for relational databases using the extended entityrelationship model. ACM Comput. Surv., 18(2):197–222, 1986.

[13] Reengineering a Cardiology Information System for Intranet Use and Web Technology D Moertl, 0 Halter", T Dombrowski*, G Porenta Department of Cardiology, University of Vienna, Austria "Oracle Consulting, Vienna, Austria.

[14] Hammer M and Stanton S. The reengineering revolution, HarperBusiness, New York, 1995.

[15] Barker R, Longman C. CASE*METHOD Function and Process Modelling. Addison-Wesley, New York, 1992.

[16] Barker R, Longman C. CASE*METHOD Tasks and deliverables. Addison-Wesley, New York, 1992.

[17] Klimczak JC, Witten DM 2nd, Ruiz M, Brilhart JG, Frankenberger ML. Proc AMIA Annu Fall Symp 1996:623-627.

[18] Willard KE, Hallgren JH, Sielaff B, Conelly DP. The deployment of a World Wide Web (W3) based medical information system. Proc Annu Symp Comput Appl Med Care 1995:771-775.

[19] Cimino JJ, Socratous SA, Clayton PD. Internet as clinical information system: application development using the World Wide Web. J Am Med Inform Assoc 1996;3:41.

[20] Reengineering Web Applications Based on Cloned Pattern Analysis Andrea De Lucia, Rita Francese, Giuseppe Scanniello, Genoveffa Tortora Dipartimento di Matematica e Informatica – Università di Salerno Via Ponte Don Melillo – 84084 Fisciano (SA) – Italy

[21] C. Boldyreff, M. Munro, and P. Warren. "The evolution of websites", In Proc. of 7th IEEE International Workshop on Program Comprehension, Pittsburgh, Pennsylvania, IEEE CS Press, 1999, pp. 178–185.

[22] D. Eichmann, "Evolving an Engineered Web", In Proc. International Workshop Web Site Evolution, Atlanta, GA, 1999, pp 12-16.

[23] K. Wong. "Toward Reusable and Evolvable Web Sites", In Proc. of 1st International Workshop on Web Site Evolution, Atlanta, GA, USA, 1999.

[24] G. Antoniol, G. Canfora, G.Casazza, and A. De Lucia, "Web Site Reengineering using RMM", In Proc. Of International Workshop on Web Site Evolution, Zurich, Switzerland, 2000, pp. 9-16.

[25] L. Aversano, G. Canfora, A. De Lucia, and P. Gallucci, "Web Site Reuse: Cloning and Adapting", In Proc. of 3rd IEEE International Workshop on Web Site Evolution, Florence, Italy IEEE CS Press, 2001, pp.107-111.

[26] F. Ricca and P. Tonella, "Understanding and Restructuring Web Sites with ReWeb", IEEE Multimedia, vol. 8, no. 2, 2001, pp. 40-51.

[27] F. Ricca, and P. Tonella, "Analysis and Testing of Web Application", In Proc. of International Conference on Software Engineering, Toronto, Ontario, Canada, 2001, pp. 25-34.

[28] C. Boldyreff. and R. Kewish, "Reverse Engineering to Achieve Maintainable WWW Sites", In Proc. of 8th IEEE Working Conference on Reverse Engineering, Suttgart, Germany, IEEE CS Press, 2001, pp. 249 – 257

[29] G. A. Di Lucca, M. Di Penta, G. Antoniol, and G. Casazza, "An Approach for Reverse Engineering of web-based applications", In Proc. of 8th IEEE Working Conference on Reverse Engineering, Suttgart, Germany, IEEE CS Press, 2001, pp. 231-240.

[30] G. A. Di Lucca, A.R. Fasolino, U. De Carlini, F. Pace, and P. Tramontana, "WARE: A Tool for the Reverse Engineering of Web Applications", In Proc of 6th European Conference on Software Maintenance and Reengineering, Budapest, Hungary, IEEE CS Press, 2002, pp. 241-250.

[31] G. Di Lucca, A. Fasolino, U. De Carlini, and P. Tramontana, , "Abstracting Business Level UML Diagrams from Web Applications", In Proc. of 5th IEEE International Workshop on Web Site Evolution, Amsterdam, The Netherlands, IEEE CS Press, pp. 12 – 19.

[32] J. Conallen, Building Web application with UML, Addison Wesley, 2000.