

Recovery of User Interface Web Design Patterns using Regular Expressions

Khalid Mahmood^{*1}

^{*}Department of Computer Science

COMSATS Institute of Information Technology, Lahore, Pakistan

^{*}Division of Science and Technology University of Education,, Lahore, Pakistan

Faiza Tariq²

Division of Science and Technology
University of Education,
Lahore, Pakistan

Dr. Ghulam Rasool³

Department of Computer Science
COMSATS Institute of Information Technology
Lahore, Pakistan

Abstract—User Interface Web Design Patterns are standard solutions for the development of web applications. The recovery of these patterns from web applications supports program comprehension, reusability, reverse engineering, re-engineering, and maintenance of legacy web applications. The recovery of patterns from web applications becomes arduous due to the heterogeneous nature of web applications. Authors presented different catalogs and recovery approaches for extracting User Interface Web Design Patterns from source code in last one and half decade. There is still a lack of formal specifications for web design patterns, which are important for their recovery from source code. The objective of this paper is to specify User Interface Web Design Patterns (UIWDP) using semiformal specification technique and use these specifications for the recovery of patterns from the source code of web applications using regular expressions. 55 feature types are identified for the specification of 15 UIWDPs. We evaluated our approach on 75 randomly selected web applications and recovered 15 UIWDPs. The standard deviation, precision, recall and F-score measures are used to evaluate the accuracy of our approach.

Keywords—Design patterns; user interface patterns; web applications; web reverse engineering; regular expressions

I. INTRODUCTION

The World Wide Web has surprisingly affected many aspects of our life and it will continuously influence society. Due to increasing popularity and use of web applications (WAs) in all fields of life, the WAs are subject to continuous evolution, maintenance and re-engineering [1], [10]. WAs demand continuous evolution due to different reasons such as improvement in usability, quality, efficiency, correction of bugs, introducing the new functionality and modifications in legacy WAs [22], [27]. User Interface Web Design Patterns are standard solutions that are frequently used for developing web applications. The appropriate use of UIWEDPs not only creates consistency among different web pages of web applications, but it also establishes efficient layout of web pages [3]-[6]. The automatic recovery of UIWDPs supports comprehension, maintenance, evolution and reengineering activities.

Web application development too often had no design and modeling principles in early days of web application

development processes. Mostly, developers apply ad hoc approaches which make the process of extracting information from web applications difficult. The recovery of information from legacy WAs is a daunting task due to the heterogeneous nature of web applications. Web applications are developed using different technologies such as Java Scripts, HTML, DHTML, XHTML, XML, CSS, ASP, PHP, DOM, database, images etc. Single language based static analysis tools are not capable to extract information from multilingual aspects of WAs.

A major motivation for reverse engineering of web applications is to reuse the tested, reliable and valid artifacts of legacy applications in the development of modern applications. The recovery of UIWDPs helps to comprehend architecture of web applications. The UIWDPs are not just about buttons and menus; they are about the interaction between the users and applications or devices. The UI WDPs are used to create consistency throughout the web development process to give the best, attractive, user-friendly, most usable and effective layouts to the WAs. The different authors have presented various catalogs of UIWDPs in the last one and half decade such as UI pattern library [4], Welie patterns library for interaction design [5], 10 UI design patterns by Jovanovic [9] and 15 UI design patterns by Kayla Night [6]. All the authors of these lists focused only on the organization, usage, naming, problem summary and examples of UIWDPs.

Most previous contributions [11]-[15] in the field of UIWDPs focused on the collection, organization and applications of UIWDPs. The recovery of these patterns from WAs requires standard and formal definitions which are not yet available to the best of our knowledge. Some approaches recover different components and models from web applications, but they did not focus on recovery of specific patterns with complete information. The concept of recovering UIWDPs is applied in [31]. The applied approach only extract whether a pattern is present in a web application or not. The approach is also limited to recognize patterns with fixed tags and fails to recognize the same pattern when it is implemented by using different constructs. Our approach recognizes patterns from WAs along with complete information about the location

of patterns which is important for the analysis of source code. We propose a lightweight approach based on lexical analysis to search patterns by file name, file path, line number and its occurrences in the WAs. The regular expression patterns are used in this paper and are easily customizable while handling variations in the detection of UIWDPs.

Our work focuses on following contributions:

- Summarization of up-to-date state of the art work on web design patterns recovery;
- Semiformal specifications of UIWDPs;
- Automatic recovery of UIWDPs from web applications;
- Evaluation of approach.

The rest of this paper is organized as: In Section II, related work is discussed. Section III presents semiformal specifications of the web user interface design patterns used by our approach. This paper presents concept and architecture of our approach in Section IV. Section V presents the evaluation of approach and Section VI concludes the whole approach.

II. RELATED WORK

A number of approaches and tools are presented for recovery of information from web applications. The comparative overview on features of different approaches presented by different authors is given in Table I. The key factors for comparison are tool support, model/framework, source code languages, Analysis type, technique applied and experimental case studies. The important approaches are listed so far in Table I and discuss selected approaches in the following paragraphs.

According to the experiment of the Carlos et al. [28], the proper use of UIWDPs [7] in the development of web applications had positive impact on the quantitative/qualitative performance and usability of WAs [8]. Different libraries of UIWDPs such as Danish web developer [4], Welie (a pattern library for interaction design) [5], 10 UI design patterns by Jovanovic [9] and 15 UIWDPs by Kayla Night [6] are presented. The information about patterns in these libraries is available in informal languages.

Yingtao et al. [37] presented a reverse engineering approach to extract presentation layer from web applications. They specify recovered features in the form of WSDL and these features can be deployed through proxies accessing the original web server and parsing its responses. Authors recover functionalities of websites from the presentation layer instead of focusing on all the source code. The applied recovery process consists of five Components (page collector, pattern miner, pattern visualize, translator and service interface editor).

Amalfitano et al. [34] presented a tool for automatic reverse engineering of dynamic web applications using source transformation technology. Authors extract UML sequence diagrams from the execution traces generated by the resulting instrumentation. The result can be directly imported and visualized in a UML toolset such as Rational Software

Architect. The extracted results can be imported and visualized in any UML 2.1 toolset. Authors apply filter execution traces directly on information stored in a database that automatically eliminates redundant information which complicates the understanding process. The approach was limited only to reverse engineering of PHP-based web applications.

Bochiha et al. [15] proposed a semi-automatic approach for re-engineering of multi-language based WAs and recovered the conceptual model by using syntactic and semantic information hidden in the LWAs. The presented framework is based on two steps: web application reverse engineering and SWS forwarding engineering. Reverse engineering step extract class and activity diagrams and forward engineering process generates WSDL and WSMO semantic descriptions. As compared to our approach, authors did not focus on the features of UI patterns that are necessary for pattern's recovery, maintenance and re-engineering from LWAs [23].

Staiger [55] presented an approach for reverse engineering of GUI components from different applications using static analysis. He extracted control flow graph for the examined applications. Author maps source code constructs with GUI components and detects relationship between GUI components through event handlers and their callers. Bauhaus tool is applied by the author to extract GUI components from the source code of C/C++.

Norizan et al. [16] performed the survey to extract the list of user interface design patterns and their impact on the usability of WAs. In their approach, they tried to recover groups of UIWDPs that were used collectively and their impact on the quantitative/qualitative performance and usability of WAs with help of check list based survey. But in our approach, we recover a single or group of UI patterns from LWAs on the basis of pattern's features.

A number of approaches [10]-[13], [35], [36] recovered UML models for the comprehension of behavioral, structural and relational aspects of WAs but these approaches did not focus on the recovery of UIWDPs and their occurrences in WAs. The Marchetto et al. [14] proposed ReAjax tool to reverse engineer only Ajax based applications. Echeverria et al. [17] used MoDisco to reverse engineer only the Strut based WAs. Martin et al. [2] proposed reusing of software engineering tool Rigi as a means of analyzing and visualizing the structure of web applications. Chung and Lee [8] proposed an approach for reverse engineering of Websites and adopt Conallen's UML extensions [53] to describe their architecture. Rasool et al. [32] presented an approach to recover general artifacts from legacy software applications based on abstract regular expressions. Draheim et al. [56] presented a tool that constructs analysis from website based on the concept of from-oriented analysis. Bouchiha et al. [57] presented an ontology based approach for reverse engineering of web applications. Boldyreff et al. [18] proposed a system that exploits traditional reverse engineering techniques to extract duplicated contents and styles from websites in order to restructure them and improve their maintainability.

Authors in approaches [17]-[20], [24], [25] used Eclipse and WARE [21] tools to create the intermediate representations of LWAs. They integrate reverse engineering and model driven engineering to extract the conceptual models of LWAs and user interaction artifacts from web applications. All applied approaches did not address the locations and occurrences of UI patterns in the code that are necessary for maintenance, correction of bugs and abstraction. Key features of different approaches are summarized in Table I.

We see from Table I that most reverse engineering approaches presented for recovery of information from web applications start their process from the selection of a model or a modeling language that will generate an intermediate format of the legacy web application. These intermediate formats may be used by one or more tools for recovering different artifacts from the source code. Some approaches use transform techniques, queries and algorithms to recover different types of artifacts and their relationships from the web applications. Many recovery tools accept models as input and recover different types of information from examined applications. The web models act as main source for the comprehensive reverse engineering of legacy web applications.

III. SEMIFORMAL SPECIFICATION OF UIWDPs

The list of UIWDPs proposed by Kayla Night [6] has been used by our approach. We realize that patterns presented in this list are more generic which include maximum UIWDPs that are included in other lists as well. We take user interface graphical items, lexical items and feature types for the specification of patterns. Graphical items are specified with the help of different tags such as HTML, Form, DIV etc. Lexical items are captions/ labels that describe graphical items. For example, user login user name and password are lexical items that are used to describe graphical items. We use the concept of feature types presented in our previous work to specify all 15 UIWDPs [50]. The feature types presented in this paper are different from feature types used to specify GoF (Gang of Four) design patterns in previous work. Feature types in this approach are based on graphical and lexical properties of patterns. They refer to different characteristics of patterns implemented by developers using different tags. The worth and quality of our specifications are based on the appropriate selection of feature types that are used by web designers in the specific area with specific sequence to implement the UIWDPs in the web applications. We identified 55 feature types presented in Table II to specify all 15 UIWDPs and their variants. We don't claim that these 55 features can be used to specify User Interface design patterns of different other lists. These features are presented in Table II. We select four UIWDPs (Login, Navigation Bars, Bread Scrum, Lazy Registration) to demonstrate our approach with their intent, visual diagrams and specifications as given in the following subsections. The specifications of rest of patterns are available on our web source [29].

A. Login Pattern

Login Pattern is very common in web applications. The intent and specification of login pattern is given below:

A login pattern [26] is required when users need to identify themselves to either gain access to a restricted area or experience a more personalized user interface based on information provided previously. Fig. 1 below presents the screenshot of a Login pattern.

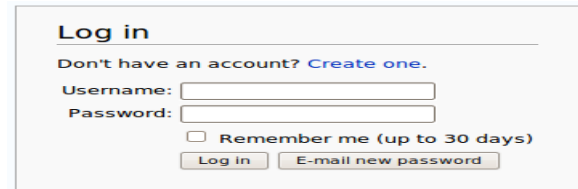


Fig. 1. Login pattern.

- Specification

Specification of login pattern described as feature types and diagrammatic notation is presented in Table III and Fig. 2. We consider variations used by developers while implementing these user interface web design patterns. Login pattern may have features (F11, F12, F13, F14, F15) or (F11, F12, F13, F14, F15, F16).

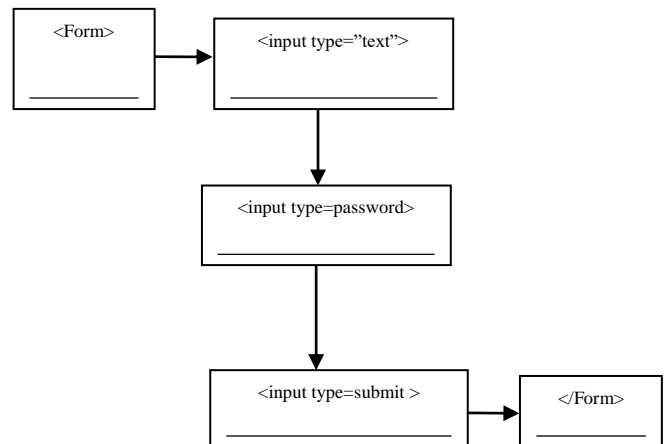


Fig. 2. Login patten.

B. Navigation Bar Patterns

Navigation Bar pattern has become a key feature of all web applications. The intent and specification of Navigation Bar pattern is given below:

Navigation Bar pattern is used when the user needs to locate contents and features necessary to accomplish a task. A vertical/horizontal navigation is a quite common layout that gives much emphasis to the vertical/horizontal orientation. Fig. 3 presents a screenshot of Navigation Bar pattern.

TABLE I. COMPARATIVE OVERVIEW OF FEATURES OF WEB RECOVERY APPROACHES

Reference/Tool	Model /Framework	Technique	Source Code Languages	Analysis Type	Experimental Case Studies
[38] (2001) OOHDM-Web (Framework)	OOHDMFrame. OOHDM-Web(UML)	Manual observation	OO code, ASP ISAPI, COM, Corba	Behavioral &.Object oriented analysis	Web application framework for conference paper review system.
[18] (2001) JavaCC parser	Rational Database	Perl, SQL	HTML, Java script	Code analysis Parse tree analyzer	Palatinate'99, SEG'99 Palatinate'00, Personal Web
[39] (2001) VAQUISTA	Presentation model XIML	Mapping tables matching techniques	HTML4.0	Static Analysis	<a href="http://www.acm.org/sigchi/ch
i2001/registration.html">http://www.acm.org/sigchi/ch i2001/registration.html).
[40] (2004) Saxon TidyR	(MultiLingual XHTML) UML	Page matching algorithm	HTML, PHP	Static analysis Syntax tree analysis	www.ien.it .
[41] (2002) WebRemUSINE	Task models	WebRemUSINE	HTML	Static analysis	<a href="http://glove.cnuce.cnr.it/sigch
i/index.html">http://glove.cnuce.cnr.it/sigch i/index.html
[10] (2001) ReWeb TeslWeb	UML Model Meta model	Test cases Node-Reduction algorithm	HTML	Static analysis Dynamic analysis	www.Wordnet.com www.Amazon.com
[13] (2012) WebLabUX	DOM, Three experiment patterns	Survey responses t-test.	N M	users' navigational behavior on a website	NM
[14] (2012) ReAjax	DOM GUI-based state models	Regular Expression FSM(finite state machine)	HTML, CSS Java script, PHP XUL,XML	Static analysis. Dynamic analysis Execution traces	[http://pafm.sourceforge.net], [http://tudu.sourceforge.net]
[15] (2010) OntoWeR	WSMO, SWS class & activity diagrams,UML Model	Transform Algorithms OWL	HTML, PHP JSP,	Syntactic and Semantic Analysis	Self made example
[42] (2008) SEASAT	MDE, Class diagram link-based models	Explanation based on 11 rules	HTML, Java script	Lexical analysis: Syntactic analysis	NM
[43] (2009) Kenyonweb,	Mining Software Repositories (MSR) DBMS	Classification algorithm & Regular expression	Java code	NM	Selected java code
[31] (2008) Google search engine, SPSS	Web mining. User interface Web patterns.	Query with degree scale (+,-,?) Descriptive statistics	NM	Semantic analysis	Nokia mobile websites
[44] (2012) MoDisCo	Navigational models MVC WebFrameworks	ATL Algo Data/Control Flows	HTML, JSP XML, Java	Static analysis.	<a href="http://www.unex.es/eweb/mig
raria/cs/agenda">http://www.unex.es/eweb/mig raria/cs/agenda
[45] (2013) RE-UWA UWA-MDD	Model Driven Reverse Engineering(MDRE) UWA models	Query View Transformation - QVT, ATLrules	Java beans, JSP, HTML	Conceptual Analysis Object oriented analysis	NM
[46] (2013) WARE	Re MDWE Meta-Model	Clustering Algorithm, SQL Queries	HTML,JavaScrip t,XML	Static analysis Dynamic analysis	General Example
[20] (2013) MIGRARIA WARE	MDWE MVC-based web frameworks.	Atlas transformations ATL rules Querying	JSP, Java, XML HTML	Static analysis Dynamic analysis	<a href="http://www.eweb.unex.es/ewe
b/migraria/cs/agenda/">http://www.eweb.unex.es/ewe b/migraria/cs/agenda/
[47](2010) JWebTracer JBPRcovery.	GUI-based Reverse Engineering, UML	Clustering techniques, BPMN, BPEL, Petri Nets	FSS, CSS, APTS	Dynamic analysis	www.softslate.com/
[34] (2009) WARE, PHP2XMI	Database model	ER model, SQL Queries	PHP,SQL HTML,	Static and dynamic analysis	NM
[21] (2007) WSDL2OWLS	MDRE, Model Driven Architecture (MDA)	(QVT) Algo [10] Owl ATL algo	WSDL files	Static, Dynamic and Semantic analysis	NM
[48] (2001) MySQL ,PHP interpreter, JSQLParser,	Application Model, System Overview, Database	SQL queries	PHP, Java	Dynamic analysis	Wordpress, Drupal and Gallery2.
[49](1996) PhpModeler, ReWeb	Reverse engineering (RE) process	Dependency analyzer Algorithm SQL	HTML,CSS, PHP,ASP.NET, Java,JavaScript,	Static analysis	Simple login page iForestFire
[54](2012)	DOM GUI-based State models	Execution traces on State modes	HTML, CSS Java script, PHP XUL,XML	Dynamic analysis	[http://pafm.sourceforge.net], [http://tudu.sourceforge.net]
[11](2004)	XMI	Mutational Techniques	HTML, ASP	Static and Dynamic analysis	XML node construction application

TABLE II. FEATURE TYPES

Feature No	Feature Tag	Feature Description	Feature No	Feature Tag	Feature Description
F1		List start point	F29	<Footer>	Footer start
F2		List end point	F30	</Footer >	Footer end
F3		Item start point	F31	<Label >	Label Text start
F4		Navigation item end point	F32	</Label >	Label Text end
F5	<a>	Anchor start	F33	<Section>	Sections start
F6		Anchor end	F34	</Section>	Section end
F7	<DIV>	Division start	F35	<article>	Sub section
F8	</DIV>	Division end	F36	</article>	Sub section end
F9		Navigation item area start point	F37	<header >	Sub section title/heading start
F10		Navigation item area end point	F38	</header>	Sub section title/heading end
F11	<Form>	Start of form	F39	<h>	Heading style start
F12	</Form>	End of form	F40	</h>	Heading style end
F13	<input type="text">	Text data field	F41	<p>	Para start
F14	<input type="password">	Password data field	F42	</p>	Para end
F15	<input type="submit">	Submit button	F43		Text Highlighter start
F16	User name /f-name/-l-name	Label/lexical item L1	F44		Text Highlighter end
F17	"Password"	Label/lexical item L2	F45	Digits	Digits
F18	"Login"/text	Label/text	F46		Order list
F19	<Table>	Start of table	F47		Un order list
F20	</Table>	End of table	F48	<Script>	Script language start
F21	<TR>	Start of row	F49	</Script>	Script language end
F22	</TR>	End of row	F50	<tbody>	Table body start
F23	<TD>	Start of column	F51	</Tbody>	Table body end
F24	</TD>	End of Column	F52	<!DOCTYPE>	Html Version
F25	<input type="hidden">	Submit with action	F53	<html><head><title>	Webpage head
F26	<button>	Button for action	F54	<linkrel="stylesheet">	CSS link
F27		Image icon	F55	<body>	Page body
F28	Id=Breadcrumb	Div tag Id which contain the path to this page			

TABLE III. FEATURES OF LOGIN PATTERN

UI graphical items	Lexical item	Feature types	Pattern definitions
<form>	Login	F11:Start of form F12:End of form F13:Text data field	F11, F13,*F14,F15,F12
<input type="text">		F14: Password data field F15: Submit button	OR
<input type="password">	Username	F16:Label/lexical item L1	F11,F16,*F13,F17,*F14,F15,F12
<input type="submit">	Password	F17:Label/lexical item L2	*: Means that feature type can repeat
</form>			



Fig. 3. Navigation pattern.

• Specification

Features of Navigation Bar pattern are given in Table IV and diagrammatic specification is presented in Fig. 4. “*” means that a feature type or group of feature types can repeat. The first variant of Navigation Bar pattern has repeating features (F3, F5, F4) as shown in Fig. 4. Features for other variants of Navigation Bar are given in Table IV.

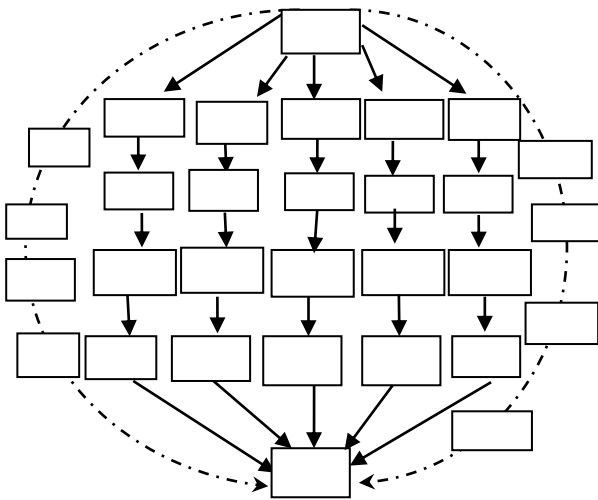


Fig. 4. Navigation bar.

TABLE IV. FEATURES OF NAVIGATION BAR PATTERN

Var.	UI graphical items	Feature types	Pattern definitions
1	 <a> 	F1:Navigation list start point F2:Navigation list end point F3:Navigation item start point F4:Navigation item end point F5:Click able link anchor start F6:Click able link anchor end	F1,(F3,F5,F4)* F2
2	<DIV> > <a> </DIV>	F7:Division or potion start that contain a list F8:Division or potion end that contain a list F9:Navigation item start point F10: Navigation item end point	F7,(F9,F5,F10)*,F8
3	<table> <tr><td> <a></td> </tr> </table>	F19:Start of table F20:End of table F21:Start of row F22:End of row F23:Start of column F24:End of Column	Horizontal F19,F21,(F23,F5,F6,F24)*F22,F20 Vertical F19,(F21,F23,F5,F6,F24,F22)F20 *: Means that feature type can repeat

C. Breadcrumbs Pattern

Breadcrumbs Pattern is mostly used when websites follow a hierarchical structure. The intent and specification of Breadcrumbs pattern is given below.

The web users require the tracking of complete browsing path form home to its current location in order to possibly switch back to a higher level in the hierarchy [3]. Fig. 5 presents a screenshot of Breadcrumbs pattern.

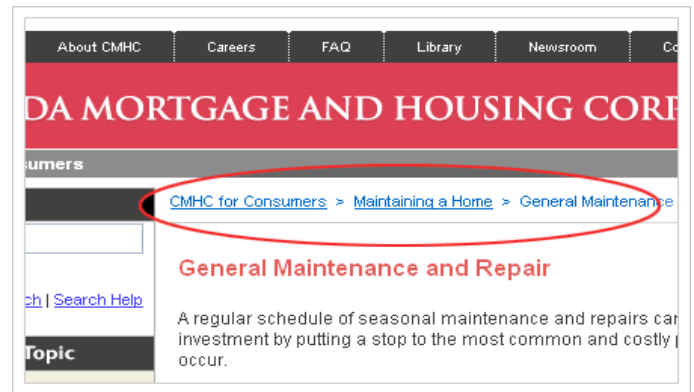


Fig. 5. Breadcrumbs patterns.

• Specification

Features of Breadcrumbs Pattern are given in Table V and diagrammatic specification is presented in Fig. 6. The sequence of features is important for specification and recovery of Breadcrumb. The standard code contains the following tags:

<DIV></DIV>, , , , <Nav></nav>

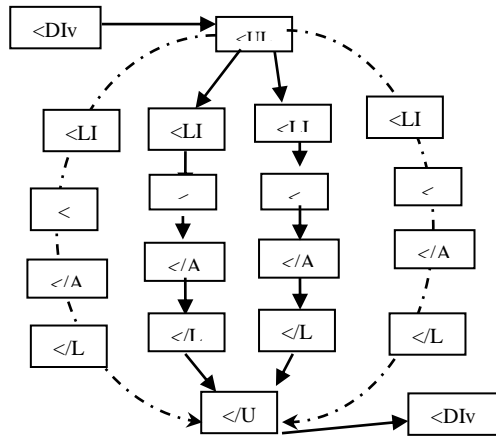


Fig. 6. Breadcrumbs pattern.

D. Lazy Registration Pattern

Lazy Registration Pattern lets the user browse the website without formal registration. The intent and specification of Lazy Registration pattern is given below.

Signup forms have long worked the casual visitor. During the process of discovery, nobody wants to stop and fill out details before they can “unlock” the rest of the site’s potential. As web users become more and more fickle, signup forms are becoming an increasingly large barrier that repels many prospective visitors from great sites. Fortunately, there is a new signup system in web designing that is making it much easier for the visitor to interact with the site and it increases signups. Fig. 7 presents a screenshot of Lazy Registration pattern.



Fig. 7. Lazy registration.

- Semiformal Specification

Features of Lazy Registration Pattern are given in Table VI and diagrammatic specification is presented in Fig. 8.

IV. DETECTION APPROACH

A lightweight and customizable approach for recognition of UIWDPs from WAS has been presented. Our approach can handle variations in patterns by customizing specifications and regular expression patterns. A number of regular expression parsing tools are available, but PowerGrep [33] has been selected due to its excellent features and free availability. It can match patterns through large numbers of files/folders in multiple formats. PowerGrep [33] is also capable to map one or more than one patterns at the same time. State of the art approaches [16], [21], [23], [31] discussed in Section II only

indicate the presence or absence of patterns by using SQL, checklist survey and descriptive analysis of WAs. The proposed approach takes semiformal specifications of patterns, source code and regular expression patterns as input and recognizes patterns with detailed information. We also plan to automate the process of writing regular expression patterns directly from specifications in the future. The architecture of the approach is given in Fig. 9.

TABLE V. FEATURES OF BREADCRUMBS PATTERNS

Var.	UI graphical items	Lexical item	Feature types	Pattern definitions
1	<Div> <a> text </Div>	<Div> tag contain attribut e breadcr umb > >> /	F7: Division or potion start that contain a list. F1: Navigation list start point F3: Navigation item start point F5: Click able link anchor start F6: Click able link anchor end F4: Navigation item end point F2: Navigation list end point F8: Division or potion end that contain a list	F7,F28,(F1, F3,F5,F6,F4,F2)*,F8
2	<Div> <a> text </Div>		F7: Division or potion start that contain a list. F9: Navigation item area start point F5: Click able link anchor start F6: Click able link anchor end F10: Navigation item area end point F8: Division or potion end that contain a list	F7,F28,(F9, F5,F6,F10)*,F8
3	<Div> <a>... > </Div>		F7: Division or potion start that contain a list. F5: Click able link anchor start F6: Click able link anchor end F9: Navigation item area start point F27: Navigation item area end point F8: Division or potion end that contain a list	F7,F28,(F5, F6,F9,F27)*,F8 *: Means that feature type can repeat

A. Regular Expressions

We develop regular expression patterns for 15 UIWDPs discussed in [6]. There are few false matches due to limitations of regular expressions. We will handle false matches by integrating our present approach with other searching techniques. Table VII below presents the regular expressions for 15 selected UIWDPs. All regular expression patterns are independent of any specific tool and they can be used in any regular expressions based editor. We list variant regular expression patterns for different patterns in column 3.

The above regular expression patterns are based on the compulsory features of UIWDPs. The worth and quality of the

recovery process is dependent on identification of correct feature types, their sequence and regular expression patterns which must be free from backtracking and false matches. The presented approach is capable to detect the presence of specified UI pattern in a given WA. It can also recover filename, complete path and line number on which it was implemented. However, we want to clarify that presented approach can recover the UIWDPs according to specified sequence of feature types from the web source codes, but, if any application has implemented a pattern with same feature types according to our semiformal specifications, but in different order, then our approach is unable to recover such types of patterns.

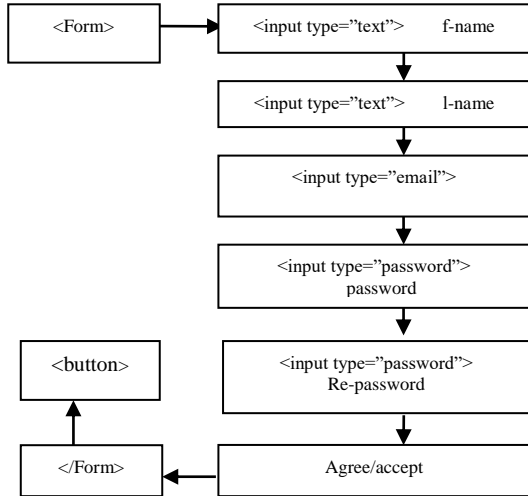


Fig. 8. Lazy registration.

TABLE VI. FEATURES OF LAZY REGISTRATION

Var.a	UI graphical items	Lexical item	Feature types	Pattern definitions
1	<code><form></code> <code><input type="text" name="f-name"></code> <code><input type="text" name="l-name"></code> <code><input type="email/text" name="email"></code> <code><input type="password" name="password"></code> <code><a---</code> <code><input type="submit">/Button</code> <code></form></code>	First name Last name Email Password Agree/accept term and condition	F11:Start of form F12:End of form F13:Text data field F14:Password data field F5: Click able link anchor start F6: Click able link anchor end F15:Submit button F16:Label/lexical item L1 F17:Label/lexical item L2	F11,F12,*F13,*F14,F5,F6 , F16 ,F17 *: Means that feature type can repeat

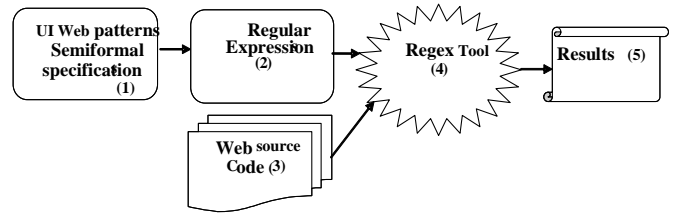


Fig. 9. Architecture of approach

TABLE VII. REGULAR EXPRESSION PATTERNS

D	Vr.	Regular Expressions
Login	1	$((\langle FORM \rangle (?: (?: \langle ORM \rangle \langle pe = "password" \rangle \langle pe = "hidden" \rangle \langle pe = "submit" \rangle)) * type = "text" (?: (?: \langle pe = "text" \rangle)) * type = "password" (?: (?: \langle pe = "text" \rangle \langle pe = "password" \rangle)) * type = ("submit" "hidden")))) ((\langle ? \rangle \langle form [\wedge] * \rangle) ((\langle ? \rangle \langle ! ORM \rangle \langle pe = "password" \rangle \langle pe = "hidden" \rangle \langle pe = "submit" \rangle)) * type = "text" (?: (?: \langle ! ORM \rangle \langle pe = "hidden" \rangle \langle pe = "submit" \rangle)) * type = "password" (?: (?: \langle ! ORM \rangle \langle pe = "password" \rangle \langle pe = "hidden" \rangle \langle pe = "submit" \rangle)) * type = ("submit" "hidden")) ((\langle ? \rangle \langle ! for \rangle) * \langle ? \rangle \langle / form \rangle))$
Navigation	1	Using <code></code> and <code></code> cannot be nested $(\langle ? \rangle \langle ul [\wedge] * \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! ul \rangle) * \langle li [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / li \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! ul \rangle) * \langle li [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / li \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! ul \rangle) * \langle li [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / li \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! ul \rangle) * \langle li [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / li \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! ul \rangle) * \langle li [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / li \rangle)$
	2	Using <code><Div></code> and <code></code> $(\langle ? \rangle \langle div [\wedge] * \rangle) (\langle ? \rangle \langle ? \rangle \langle span [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / span \rangle) (\langle ? \rangle \langle ? \rangle \langle span [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / span \rangle) (\langle ? \rangle \langle ? \rangle \langle span [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / span \rangle) (\langle ? \rangle \langle ? \rangle \langle span [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / span \rangle)$
	3	Using <code><table></code> cannot be nested $(\langle ? \rangle \langle table [\wedge] * \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! table \rangle) * \langle td [\wedge] * \rangle (\langle ? \rangle \langle ! table \rangle) * \langle a [\wedge] * \rangle (\langle ? \rangle \langle ! table \rangle) * \langle / a \rangle (\langle ? \rangle \langle ! table \rangle) * \langle / td \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! table \rangle) * \langle td [\wedge] * \rangle (\langle ? \rangle \langle ! table \rangle) * \langle a [\wedge] * \rangle (\langle ? \rangle \langle ! table \rangle) * \langle / a \rangle (\langle ? \rangle \langle ! table \rangle) * \langle / td \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! table \rangle) * \langle td [\wedge] * \rangle (\langle ? \rangle \langle ! table \rangle) * \langle a [\wedge] * \rangle (\langle ? \rangle \langle ! table \rangle) * \langle / a \rangle (\langle ? \rangle \langle ! table \rangle) * \langle / td \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! table \rangle) * \langle td [\wedge] * \rangle (\langle ? \rangle \langle ! table \rangle) * \langle a [\wedge] * \rangle (\langle ? \rangle \langle ! table \rangle) * \langle / a \rangle (\langle ? \rangle \langle ! table \rangle) * \langle / td \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! table \rangle) * \langle td [\wedge] * \rangle (\langle ? \rangle \langle ! table \rangle) * \langle a [\wedge] * \rangle (\langle ? \rangle \langle ! table \rangle) * \langle / a \rangle (\langle ? \rangle \langle ! table \rangle) * \langle / td \rangle)$ Using <code><table></code> can be nested $(\langle ? \rangle \langle table [\wedge] * \rangle) (\langle ? \rangle \langle ? \rangle \langle td [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / td \rangle) (\langle ? \rangle \langle ? \rangle \langle td [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / td \rangle) (\langle ? \rangle \langle ? \rangle \langle td [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / td \rangle) (\langle ? \rangle \langle ? \rangle \langle td [\wedge] * \rangle) (\langle ? \rangle \langle a [\wedge] * \rangle) (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / td \rangle) (\langle ? \rangle \langle ? \rangle \langle table \rangle)$
id	1	$((\langle ? \rangle \langle Div [\wedge] * \rangle (?: \langle = rumb [a-z] * \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! Div \rangle) * \langle a [\wedge] * \rangle (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle (\langle ? \rangle \langle ! Div \rangle) * \langle a [\wedge] * \rangle (\langle ? \rangle \langle / a \rangle) (\langle ? \rangle \langle / Div \rangle)))$

V. EVALUATION

We evaluate our approach on 75 randomly selected websites. The source code for all these websites is available freely on web. We cannot find a standard benchmark system from the literature that can be used for comparison of our results. Due to this reason, we performed manual analysis on the source code which was very time consuming and daunting task. The results of manual analysis are compared with automated results on five selected websites for each pattern. The reason for performing manual and automated on five selected websites is due to the reason that we want to validate patterns definitions manual analysis of source code. Secondly, we performed an experiment on all websites for 15 UIWDPs and extracted results are shown in Table X. We want to clarify that we calculated accuracy of our approach on five selected websites for each pattern. The detail about automated and manual experiments is given below.

A. Automated Experiment

In order to validate our semiformal specifications and regular expression patterns, we performed experiments on 75 different websites using PowerGrep [33] tool and recovered instances of 15 selected patterns. PowerGrep tool has excellent capabilities for parsing source code by using regular expressions. The recovered results are presented in Table VIII.

B. Manual Experiment

Manual analysis of source code is very time consuming and daunting task. It was important for us to validate our semiformal specifications of UIWDPs in the source code manually. In order to recover a pattern, we start searching from folders, subfolders and files, line by line, that was very time consuming and very difficult to mark starting and end point of

the patterns. The multiple occurrences of patterns and finding the precise location of patterns in the code are very hard.

There exists slight difference between the manual and automated results due to the several factors. One of the major reasons is that different web designers implement UIWDPs according to their skill, nature of applications and frameworks. The manual detection involves personal experiences and understanding of multi-language source code, but our approach can recover patterns automatically by following the semiformal specifications and regular expression patterns.

The difference between automated and manual experiments with their Standard Deviation (S.T.D.) is shown in the following Fig. 10.

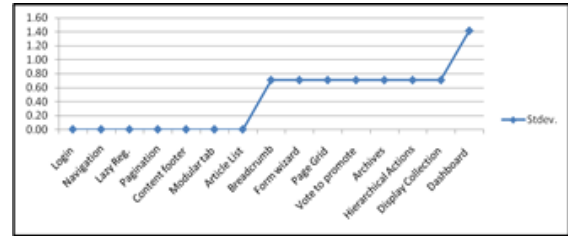


Fig. 10. Comparison between the manual & automatic recovery.

C. Accuracy of Approach

The accuracy in the universal statistical sense indicates the proximity of calculations or estimates to the accurate or exact results. To measure the accuracy of our approach, we compute the Precision, Recall and F-Score for our recovered results.

The data retrieval techniques can be evaluated by using the Precision and Recall metrics. This matrix method is popular for the evaluation of patterns extraction approaches.

TABLE VIII. EVALUATION RESULTS

Pattern	Website	Files		Results		Total		SD	TP	FP	FN	Pr	Rc	Fs
		Tot	Imp	M	A	TM	TA							
Login	Olx.com	80	1	1	1	3	3	0.00	3	0	0	1.00	1.00	1.00
	samshine.com	87	0	0	0									
	geomedia.com	103	0	0	0									
	www.fast.edu.pk	10	1	1	1									
	Netvib.com	85	1	1	1									
Navigation	Olx.com	80	1	2	2	9	9	0.00	9	0	0	1.00	1.00	1.00
	aiou.edu	107	2	2	2									
	geomedia.com	103	1	1	1									
	Ngaming.com	80	1	1	1									
	Bath_baby.com	80	3	3	3									
Breadcrumb	hec.gov.pk	17	0	0	1	3	4	0.71	3	0	1	0.75	1.00	0.95
	Ngaming.com	80	1	1	1									
	Bath_baby.com	80	1	1	1									
	Netvib.com	85	1	0	0									
	Vimeo.com	91	1	1	1									
Lazy	hec.gov.pk	17	0	0	0	2	2	0.00	2	0	0	1.00	1.00	1.00

Registration	sis.cuonlineatd.edu	95	0	0	0													
	aiou.edu	107	0	0	0													
	Netvib.com	85	1	1	1													
	Vimeo.com	97	1	1	1													
Pagination	drweb.de/magazin	73	1	1	1	8	8	0.00	8	0	0	1.00	1.00	1.00				
	d3.ru/popular	45	2	2	2													
	erweiterungen.de	35	2	2	2													
	subcide.com/articles	95	1	1	1													
	spacecollective.org	88	1	1	1													
Content Footer	www.cssbeauty.com	17	1	1	1	5	5	0.00	5	0	0	1.00	1.00	1.00				
	billyhughes.oph.gov	17	1	1	1													
	www.cityofgrace.com	35	1	1	1													
	www. Lastfm.com	84	1	1	1													
	viget.com	28	1	1	1													
Form wizard	club.nokia.com/	116	1	1	1	6	5	0.71	5	1	0	1.00	0.83	0.85				
	beanstalkapp.com/	86	1	1	1													
	ui-patterns.com/	87	2	2	2													
	mite.com	20	1	1	1													
	statementstacker.com	36	1	1	0													
Modular Tab	bestwebgallery.com	43	1	1	1	6	6	0.00	6	0	0	1.00	1.00	1.00				
	haveamint.com	10	1	1	1													
	mailchimp.com	11	1	1	1													
	www.cbs.com	107	2	2	2													
	viget.com/advance	16	1	1	1													
Page Grid	nps.gov/index	46	1	1	1	5	4	0.71	4	1	0	1.00	0.80	0.83				
	nytimes.com/	158	1	1	1													
	yeeaahh.subtraction.com	30	1	1	0													
	theguardian.com	131	1	1	1													
	yui.github.io	37	1	1	1													
Article list	alistapart.com	17	1	1	1	5	5	0.00	5	0	0	1.00	1.00	1.00				
	businesscatalyst.com	55	1	1	1													
	visitmix.com	23	1	1	1													
	www.popsci.com	106	1	1	1													
	valetmag.com	82	1	1	1													
Vote to promote	digg.com/	99	1	2	3	9	10	0.71	9	0	1	0.90	1.00	0.98				
	news.layervault.com	9	1	1	1													
	news.ycombinator.com	3	1	1	1													
	stackoverflow.com/	65	1	3	3													
	www.squidoo.com/	62	1	2	2													
Archives	astheria.com	9	1	1	1	6	6	0.71	6	0	0	1.00	1.00	1.00				
	ismaelburciaga.com	36	1	1	1													
	colly.com	6	1	1	1													
	www.cssmania.com	17	1	1	1													

	www.darrenhoyt.com	31	1	1	1										
Dashboard	www.tripit.com	48	1	1	1	8	7	1.41	6	2	0	1.00	0.75	0.78	
	www.mint.com	23	1	1	0										
	www.weatherspark.com	10	1	1	1										
	www.udemy.com	77	2	4	3										
	www.last.fm	120	1	1	1										
Hierarchical Actions	www.fedex.com	40	1	1	1	5	4	0.71	5	1	0	1.00	0.83	0.85	
	www.twitter.com	30	1	2	2										
	www.paypal.com	29	1	1	1										
	www.barnesandnoble.com	22	1	1	0										
	www.reg.ebay.com	11	1	1	1										
Display Collection	www.smileycat.com	63	2	3	3	9	8	0.71	8	1	0	1.00	0.89	0.90	
	http://patternzap.com	34	1	3	3										
	www.webdesignpractices.com/	26	1	1	0										
	www.quince.infragistics.com	123	1	1	1										
	http://www.patternry.com	55	2	1	1										
Total/ Average		-	79	90	86	90	86	-	84	6	2	0.98	0.93	0.94	

SD: Standard Deviation, TP: True Positives, FP: False Positives, FN: False Negatives, Pr: Precision, Rc: Recall, Fs: F-Score, Tot: Total Files, Imp: Files in which pattern is implemented, M: Manual Result, A: Automated Result, TM: Total Manual Instances, TA: Total Automated Instances

The Precision and Recall have been used to evaluate the quality of different systems from the last few decades. This method can evaluate how many patterns retrieved are relevant and how many relevant patterns are retrieved [51]. The accuracy of any approach can be measured by finding the relationship between the Precision and Recall metrics. In the ideal situations, the Precision results of an approach should remain high when the Recall increases [52]. The following parameters are used to calculate the Precision and Recall for pattern recovery techniques as given in Table IX.

Precision: The fraction of retrieved documents those are relevant: $P=TP/(TP+FP)$

Recall: The fraction of relevant documents that are retrieved: $R=TP/(TP+FN)$

TABLE IX. PRECISION AND RECALL

TP= True Positives TN=True Negatives FN=False Positives, FN=False Negatives	Relevant	Non Relevant
Retrieved	TP	FP
Not Retrieved	FN	TN

F-Score: The accuracy of pattern recovery approaches can be effectively measured by using the Precision and Recall, although the combination of both factors yields a combined effect. This common factor for evaluating the Precision and Recall metrics for any recovery technique is addressed by Peterson et al. [30]. They proposed a standard solution by using the Precision and Recall of any approach. They defined F-score (Fw) as:

$$Fw = (1+W2)(PR)/(W2P+R)$$

The value of W is constant ($W = 2.28$). If the Precision and Recall of any approach are high, then F-Score obtained is also high. The Precision and Recall for manual and automated experimental results are given in Table VIII and presented in Fig. 11. The average Precision, Recall and F-Score is 98%, 93% and 94% respectively for all 15 UI WDPs detected from the 75 web applications.

D. Validity Threats

Validity is an important concern for empirical acceptance of results extracted by different approaches. A major threat to results of our approach is a lack of standard definitions for UIWDPs and their variants. These definitions are not available in literature to the best of our knowledge. Internal validity refers to the consistency of measurements across all methods and tools. It is affected by experimental biases. We tried to mitigate this threat by manually performing experiments on source code of 75 selected web applications. All the experimental results of our approach are presented in paper and researchers can validate our results. External validation is affected by generalization of results. We specified all 15 UIWDPs using semiformal specifications and these specifications are available on web for community. The source code of 75 web applications is also freely available. One possible threat to external validity of our results may be application of different constructs for the implementation of UIWDPs. Reliability validity affects the replicability of our results. All selected web applications and their source code are available on web for validation. Reliability validity threat is eliminated because our regular expression patterns can be validated with different regular expression parsing tools which are available on the web.

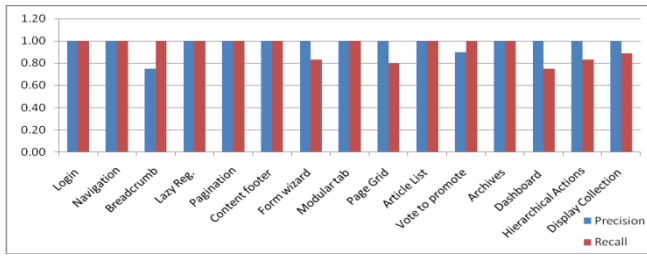


Fig. 11. Precision and recall for all patterns.

VI. CONCLUSION

UIWDPs recovery from WAs is a challenging task due to ever increasing applications of new technologies for the development of web applications. The recovery of information from web applications provides valuable information to maintenance, comprehension, refactoring, reuse and re-engineering disciplines. A number of techniques and tools are presented for recovering information from web applications, but they are not capable enough to deal with the heterogeneous nature of web applications completely. The recovery of

information from web applications is difficult due to number of technologies and external dependencies in web applications. State of the art approaches focused on extraction of UML models from WAs. In this paper, we present an approach that can recover UIWDPs from LWAs with their necessary attributes such as filename, line number, numbers of matches per file, etc. The deviation in the automated results extracted by our approach and manual results shows that there is no consensus on the definitions of UIWDPs from community. The implementation variations are another cause in the disperse results extracted by our approach. Our approach can handle multi-language source code partially for recovery of patterns directly from the source code. Moreover, the recovered UI pattern's information can be effectively used in the maintenance, abstraction, comprehension, upgradation, migration of applications from one framework to another and re-engineering of LWAs. In future, we plan to extend the scope of our automatic recognition of UIWDPs approach on other UI pattern libraries such as Yahoo pattern library, Weli pattern library, etc.

TABLE X. EVALUATION OF ALL WEBSITES FOR ALL PATTERNS

Pattern #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
www.olx.com	1	2	x	x	x	1	x	2	x	x	1	x	x	x	2
www.samshine.com	x	1	x	x	x	1	x	1	x	1	x	1	x	x	1
www.geomedia.com.au	x	1	x	x	x	1	x	2	1	1	x	2	x	x	x
www.fast.edu.pk	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x
www.netvibes.com	1	x	x	1	1	1	1	1	1	x	x	x	x	x	x
www.aiou.edu	x	2	x	x	x	x	x	1	x	x	1	x	x	x	1
www.ngaming.com	x	1	1	x	x	1	x	2	1	x	x	1	x	x	1
www.buybuybaby.com	x	3	1	x	x	1	x	x	x	x	x	x	1	x	x
www.hec.gov.pk	x	1	1	x	x	1	x	2	1	x	1	x	x	x	2
www.buybuybaby.com	x	x	1	1	x	1	1	x	1	1	x	1	x	1	1
www.netvibes.com	1	2	x	x	x	1	x	2	x	x	1	x	x	x	2
www.vimeo.com	x	1	x	x	x	1	x	1	x	1	x	1	x	x	1
www.sis.cuonlineatd.edu.pk	x	x	x	x	x	x	x	1	1	x	1	x	x	1	x
www.drweb.de/magazin	x	3	x	x	1	1	x	1	1	1	1	x	2	x	3
http://d3.ru/popular	x	3	x	1	2	2	x	1	x	x	x	1	x	1	1
www.erweiterungen.de	x	x	x	1	2	x	x	x	x	1	x	x	x	x	1
www.subcide.com/articles	x	x	x	x	1	x	x	1	x	x	1	1	x	x	x
http://spacecollective.org	x	x	x	x	1	1	x	x	x	x	2	x	x	x	x
www.cssbeauty.com	x	x	x	x	2	1	x	1	1	x	x	1	x	x	x
www.billyhughes.opf.gov	x	x	x	x	x	1	x	1	1	1	x	x	x	x	1
www.cityofgrace.com	x	x	x	x	x	1	x	x	1	x	x	x	x	x	x
www.last.fm	x	x	x	x	x	1	x	1	1	x	1	2	x	x	2
www.viget.com	x	x	x	x	x	1	x	1	x	1	x	1	x	x	1
www.club.nokia.com	x	x	x	1	x	1	1	1	x	1	1	1	x	x	5
www.beanstalkapp.com	1	x	x	1	x	1	1	1	2	1	1	1	x	2	1

http://ui-patterns.com	2	x	x	2	x	2	2	2	2	x	x	x	x	2	2
www.mite.com	1	x	x	1	x	x	1	x	1	x	x	x	x	1	x
www.statementstacker.com	x	1	1	x	1	1	x	x	1	1	x	1	x	1	x
www.bestwebgallery.com	x	x	x	x	x	2	x	1	1	x	2	4	x	x	3
www.haveamint.com	x	x	x	x	x	1	x	1	x	x	x	x	x	x	x
www.mailchimp.com	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
www.cbs.com	x	x	x	x	x	1	x	1	x	x	x	1	x	x	1
www.viget.com/advance	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
www.nps.gov/index	x	x	x	x	x	1	x	1	1	x	1	x	x	x	3
www.nytimes.com/	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
ww.yeeaaah.subtraction.com	x	x	x	x	x	1	x	1	x	1	x	1	x	x	1
www.theguardian.com	x	x	x	x	x	1	x	1	1	x	x	2	2	x	2
www.yui.github.io	x	x	x	1	x	1	x	2	1	1	3	1	x	x	4
www.alistapart.com	x	x	x	x	x	1	x	x	1	x	x	1	x	x	1
businesscatalyst.com	x	x	x	x	x	1	x	1	1	1	x	4	1	x	3
http://visitmix.com	x	x	x	x	2	3	x	2	5	2	1	1	3	x	1
http://www.popsci.com	x	x	x	x	x	x	x	1	x	1	x	x	x	x	x
www.valetmag.com	x	x	x	x	x	1	x	1	x	1	x	1	x	x	2
http://digg.com	x	x	x	x	x	1	x	2	x	2	x	x	x	x	2
https://news.layervault.com	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
www.news.ycombinator.com	x	x	x	x	x	2	x	1	1	1	1	1	x	x	1
http://stackoverflow.com	x	x	x	x	x	2	x	2	x	1	1	x	x	x	3
http://www.squidoo.com	x	x	x	x	x	1	x	1	x	1	3	1	x	x	1
http://astheria.com	x	x	x	x	x	x	x	x	x	x	x	1	x	x	1
http://ismaelburciaga.com	x	x	x	x	x	2	x	1	1	1	1	1	x	x	2
http://colly.com	x	x	x	x	x	x	x	x	x	x	x	1	x	x	1
www.cssmania.com	x	x	x	x	x	x	x	1	x	x	x	1	x	x	1
www.darrenhoyt.com	x	x	x	x	1	1	x	1	x	x	x	1	x	x	x
www.tripit.com	x	x	x	x	x	1	x	x	1	x	x	x	1	1	1
www.mint.com	1	x	x	x	x	1	x	1	1	1	x	x	0	x	x
www.weatherspark.com	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
www.udemy.com	x	x	x	x	x	1	x	1	1	1	x	x	5	x	1
www.last.fm	x	x	x	x	x	x	x	1	x	1	2	5	1	x	x
www.fedex.com	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x
www.twitter.com	1	x	x	x	x	1	x	1	x	x	x	4	x	1	1
www.paypal.com	3	x	x	2	x	3	x	2	1	x	x	1	1	2	1
www.barnesandnoble.com	x	x	x	x	x	1	x	x	1	x	x	x	x	1	x
www.reg.ebay.com	x	x	x	x	x	1	x	1	1	x	1	1	x	x	3
www.smileycat.com	1	x	x	x	x	1	1	x	x	x	x	x	x	1	x
http://patterntap.com	x	x	x	x	x	3	x	3	2	2	x	2	x	x	1
www.webdesignpractices.com	x	x	x	x	x	1	x	1	x	x	1	1	x	x	1
www.quince.infragistics.com	1	x	1	x	1	1	x	1	x	1	x	x	1	1	1
www.patternry.com	1	x	x	x	x	x	x	x	x	x	x	x	1	1	1
Total	15	21	6	10	15	61	8	60	38	30	29	52	19	17	72

1: Login, 2: Navigation, 3: Breadcrumb, 4: Lazy Registration, 5: Pagination, 6: Content Footer, 7: Form wizard, 8: Modular Tab, 9: Page Grid, 10: Article list, 11: Vote to promote, 12: Archives, 13: Dashboard, 14: Hierarchical Actions, 15: Display Collection

REFERENCES

- [1] A. Rezazadeh, "Formal Patterns for Web-based Systems Design", PhD Thesis, http://eprints.soton.ac.uk/267101/1.hasCoversheetVersion/Formal_Patterns_for_Web-based_Systems_Design.pdf, 2007.
- [2] J. Martin and L. Martin, "Website maintenance with software engineering tools", In Proceedings of 3rd International Workshop on Web Site Evolution, pp. 126-131, 2001.
- [3] J. Wedekind, "Web Design Patterns A future approach?", Bachelor Thesis, www.patterns.jana-wedekind.de/downloads/BachelorThesis_Wedekind.pdf, 2008.
- [4] A. Toxboe, "UI Patterns," <http://ui-patterns.com>, 2008.
- [5] W. Van, V. Martijn, C. Gerrit, "Pattern Languages in Interaction Design: Structure and Organization", <http://www.welie.com/papers/Welie-Interact2003.pdf>, 2003.
- [6] K. Night, "15 UI Design Patterns Web Designers Should Keep Handy", 2010.
- [7] C. Alexander, "The Timeless Way of Building," Oxford University Press, New York, 1979.
- [8] Y. Chung, S. Lee, "Reverse software engineering with UML for website maintenance", In Proceedings of 1st International Conference on Web Information Systems Engineering, IEEE Computer Society Press: Los Alamitos, CA, Vol.2. pp. 157-161, 2001.
- [9] J. Jovanovic, "10 UI Design Patterns You Should Be Paying Attention To", <http://uxdesign.smashingmagazine.com/2009/06/23/10-ui-design-patterns-you-should-be-paying-attention-to>, 2009.
- [10] F. Ricca and P. Tonella, "Analysis and Testing of Web Applications", In Proceedings of the 23rd international conference on Software engineering, pp.25-34, 2001.
- [11] C. Bellettini, A. Marchetto, "WebUml: Reverse Engineering of Web Applications", ACM Symposium on Applied Computing, Vol. SAC(04), pp.1662-1669,2004.
- [12] C. Wei, E. Cuddihy, J. Barrick and J. H. Spyridakis, "Conducting Usability Research through the Internet", In Proceedings of the Usability Professional Association, 2005.
- [13] E. Cuddihy, J. H. Spyridakis, "From Web Designer to Researcher: Using Experiment Patterns to Research Web Design Decisions" In Proceedings of Professional Communication Conference, Vol. 7(12), pp. 1-10, 2012.
- [14] A. Marchetto, P. Tonella, F. Ricca, "ReAjax: a reverse engineering tool for Ajax Web Applications", IET Software, Vol. 6, pp.33-49, 2012.
- [15] D. Bouchiha, M. Malki, "Towards Re-engineering Web Applications into Semantic Web Services", In Proceedings of International Conference on Web and Machine Intelligence, pp.350-353, 2010.
- [16] W. W. Norizan, W. Hashim, N. Laila, M. Noor, W. Adilah, W. Adnan, "Web Aesthetic Interaction Design Pattern in Popular Clothing Brands Websites: An Initial Investigation", In Proceedings of International Conference on Systems, Man and Cybernetics, Vol. 978-1, pp. 2323-2327, 2010.
- [17] R. R. Echeverria, M. J. Conejero, J. Pedro, Clemente, D. M. Villalobos and F. S. Figueroa, "Generation of WebML Hypertext Models from Legacy Web Applications", In Proceedings of 14th IEEE International Symposium on Web Systems Evolution, pp.91-95, 2012.
- [18] C. Boldyreff, R. Kewish, "Reverse Engineering to Achieve Maintainable WWW Sites", In Proceedings of 8th Working Conference on Reverse Engineering IEEE Computer Society Press: Los Alamitos, CA, pp. 249-257, 2001.
- [19] De Lucia, A., Francese, R., Scanniello, G., and Tortora, G., "Reengineering Web Applications Based on Cloned Pattern Analysis", In Proceedings of the 12th IEEE International Workshop on Program Comprehension (IWPC'04), 2004.
- [20] R. R. Echeverria, M. J. Conejero, J. Pedro, D. M. Villalobos and F. S. Figueroa, "Model Driven Extraction of the Navigational Concern of Legacy Web Applications," ICEWE Workshop LNCS 7703, pp.56-70, 2012.
- [21] R. Patel, F. Coenen, R. Martin, and L. Archer, "Reverse Engineering of Web Applications: A Technical Review," <https://intranet.csc.liv.ac.uk/research/techreports/tr2007/ulcs-07-017.pdf>, pp.1-12, 2007.
- [22] G. Rasool, "Customizable Feature Based Design Pattern Recognition Integrating Multiple Techniques," Ph.D. Thesis, 2010.
- [23] J. Conallen. "Modeling Web application Architectures with UML", In Communications of the ACM, Vol. 42, pp. 63-70, 1999.
- [24] J. Maras, A. Petric, and M. Stula. "Reverse engineering legacy Web applications with php Modeler", Conference Paper at Mälardalen University Software Engineering Workshop (MUSE09), 2009.
- [25] S. Tilley and S. Huang, "Evaluating the reverse engineering capabilities of web tools for understanding site content and structure: A case study", In Proceedings of the 23rd International Conference on Software Engineering, pp.514-523, 2001.
- [26] P. Tramontana, "Reverse Engineering of Web Applications", Ph.D. Thesis, Chapter 10, 2005.
- [27] F. Ricca and P. Tonella. "Website understanding and re-structuring with the reweb tool", IEEE MultiMedia, Vol, 8, No. 2, pp. 40-51, April-June, 2001.
- [28] A. Carlos, L. Mmarc and Maldonda, "Do Common user Interface Design Patterns improve Navigation" In Proceedings of the Human and Ergonomics Society 46th Annual Meeting, Vol. 46, No 14, pp. 1315-1319, 2002.
- [29] <http://research.ciitlahore.edu.pk/Groups/SERC/DesignPatterns.aspx>
- [30] N. Pettersson, W. Löwe and J. Nivre, "Evaluation of Accuracy in Design Pattern Occurrence Detection", IEEE Transactions on Software Engineering, IEEE computer Society Digital Library, IEEE Computer Society, pp. 575-590, Volume 36, No. 4, July/August 2010
- [31] Karel, k., Jana, K., Ondrej, L., Milos, K., Vaclav, S., Hana, R., "Web Page Analysis: Experiments Based on Web Patterns", In proceedings of 4th International Conference on Innovations in Information Technology, pp. 16-20, 2007.
- [32] G. Rasool and N. Asif, "Software artifacts recovery using abstract regular expressions", In Proceedings of Multitopic Conference INMIC, pp. 1-6, 2007.
- [33] PowerGrep, Home page:<https://www.powergrep.com>
- [34] D. Amalfitano, A. R. Fasolino, and P. Tramontana, "Experimenting a reverse engineering technique for talian the talian of rich internet applications," In Proceedings of IEEE International Conference on Software Maintenance, pp. 571-574, 2009.
- [35] W. Sun, S. Li, D. Zhang, and Y. Yan, "A model-driven reverse engineering approach for semantic web services composition," In Software Engineering, 2009. WCSE'09. WRI World Congress on, 2009, pp. 101-105.
- [36] Djelloul, B., Mimoun, M., & El Kader, M. A., "Towards reengineering web applications to web services", The International Arab Journal of Information Technology, 6(4), 359-364, 2009.
- [37] Jiang, Y., & Stroulia, E., "Towards reengineering Web sites to Web-services providers", In Proceeding of Eighth European Conference on Software Maintenance and Reengineering, pp. 296-305, 2004.
- [38] D. Schwabe, L. Esmeraldo, G. Rossi, F. Lyardet, "Engineering web applications for reuse", IEEE Multimedia, 8(1), pp. 20-31, 2001.
- [39] J. Vanderdonck, L. Bouillon, N. Souchon, "Flexible reverse engineering of web pages with VAQUISTA", In Proceedings of Eighth Working Conference on ReverseEngineering, pp. 241-248, 2001.
- [40] P. Tonella, F. Ricca, E. Pianta, C. Girardi, "Restructuring Multilingual Web Sites", In Proceedings of International Conference on Software Maintenance, pp. 290-299, 2002.
- [41] L. Paganelli, F. Paternò, "Automatic Reconstruction of the Underlying Interaction Design of Web Applications", In Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, pp.439-445, 2002.
- [42] J. Pu, H. Yang, B. Xu, L. Xu and W. Cheng-Chung Chu, "Combining MDE and UML To Reverse Engineer Web Based Legacy Systems", In proceedings of IEEE International Computer Software and Applications Conference, pp. 718-725, 2008.
- [43] S. Kim, S. Shivaji, E. James Whitehead Jr., Kenyon "Web: Reconfigurable Web-based Feature Extractor", In Proceedings of IEEE ICPC, pp. 287-288, 2009.

- [44] R. Rodríguez-Echeverría, J. Conejero, P. J. Clemente, M. D. Villalobos and F. Sánchez-Figueroa, "Generation of WebML Hypertext Models from Legacy Web Applications", In Proceedings of 14th Symposium on Web Systems Evolution, pp.91-95, 2012.
- [45] M. L. Bernardi, G. A. Di Lucca, D. Distanto and M. Cimitile, "Model Driven Evolution of Web Applications", In Proceedings of IEEE Conference, pp. 45-50, 2013.
- [46] P. Tramontana, D. Amalfitano and A. R. Fasolino, "Reverse Engineering Techniques: from Web Applications to Rich Internet Applications", In Proceedings of 15th Symposium on Web Systems Evolution, pp.83-86, 2013.
- [47] D. Amalfitano, A. R. Fasolino, and P. Tramontana, "An Iterative Approach for the Reverse Engineering of Rich Internet Application User Interfaces", In Proceedings of Internet and Web Applications and Services Conference, pp. 401–410, 2010.
- [48] D. Lucca, G.A. Di Penta, G. Antoniol, G. and G Casazza, "An approach for Reverse Engineering of Web Based Applications", In Proceedings of 8th Working Conference on Reverse Engineering, pp. 208-207, 2001.
- [49] M. Moore, "Representation Issues for Reengineering Interactive Systems", ACM Computing Surveys, 28(4), pp. 199, 1996.
- [50] G. Rasool and P. Mäder, "A Customizable Approach to Design Patterns Recognition Based on Feature Types", Arabian Journal for Science and Engineering, pp. 1-23, 2014.
- [51] M. Voorhees, "The philosophy of information retrieval evaluation," In Evaluation of cross-language information retrieval systems, pp. 355-370, 2002.
- [52] C. K. Roy, "Detection and analysis of near-miss software clones," In Proceedings of International Conference on Software Maintenance ICSM 2009. pp. 447-450, 2009.
- [53] [53] F. Ricca, M. Di Penta, M. Torchiano, P. Tonella and M. Ceccato, "An empirical study on the usefulness of Conallen's stereotypes in Web application comprehension", In Proceedings of Eighth IEEE International Symposium on Web Site Evolution, pp. 58-68, 2006.
- [54] A. Marchetto A, P. Tonella and F. Ricca , " ReAjax: a reverse engineering tool for Ajax Web applications", IET Software Vol 6, No 1, pp. 33-49, 2012.
- [55] S. Staiger, 'Reverse engineering of graphical user interfaces using static analyses'. Working Conf. on Reverse Engineering (WCRE), Washington, DC, USA, 2007.
- [56] Draheim, D., Lutteroth, C., and Weber, G., "A Source Code Independent Reverse Engineering Tool for Dynamic Web Sites", In Proceedings of 9th European Conference on Software Maintenance and Reengineering (CSMR'05), pp. 168-177, 2005.
- [57] Bouchiha, D., Malki, M., and Benslimane, S-M., "Ontology Based Web Application Reverse-Engineering Approach", Journal of Computer Science (INFOCOMP), VOLUME 6-N. 1-. pp: 37-46, 2007.