



## An Effective Regression Testing Technique for Web Application on Cloud Environment

**Diljot Singh Sethi**  
Mechanical Engineering, Dept  
Punjabi University,  
Patiala, India

**Raghu Garg**  
Research Scholar,  
Computer Engineering Dept,  
Punjabi University, Patiala, India

**Navjot Singh Sidhu**  
Computer Engineering,  
Asra College of Engg. & Technology,  
Sangrur, India

---

**Abstract:** - *Web applications are complex sophisticated programs that are based on novel computing technologies. Web applications are popular due to the omnipresence of web browsers and the convenience of using web browser as a client. The ability to update and maintain web applications without distributing and installing software on potentially large number of client computers is a key reason for its popularity. Modern web sites enables the capturing, processing, storage and transmission of sensitive customer data for immediate and recurrent use and that is made possible only through the web applications. Web applications are quickly deployed anywhere at no cost and without any installation requirements at the user end. Although powerful, these technologies bring new challenges for developers and testers. As a result, all entities of web application have to be tested. Due to market pressure and high configuration cost of software testing the web application functional testing is often neglected by the developers. No doubt, there a number of web application testing tools available but these tools use traditional techniques which are not suitable for complex structured web applications. The traditional techniques lack the additional features of testing such as regression testing, fixing errors, uniquely identifying hypermedia, testing database connection etc. In the proposed work CWEBTEST testing tool has been developed to overcome the problems of regression testing of web applications. CWEBTEST testing tool test web applications based on unique hypermedia. With the CWEBTEST tool hypermedia are identified from the compiled HTML code and then regression testing is performed. The developed tool improves the efficiency by reducing 10-25% test data size. Moreover, resource utilization is improved as compared to existing testing techniques. The tool has been published on Microsoft public cloud environment for experimental purpose and the performance has been found to be satisfactory.*

---

**Keywords:**- *Web Application Functional testing, Regression Testing, Cloud Environment*

---

### I. INTRODUCTION

The developments of web applications have received significant attention in past few years. Web applications are complex sophisticated programs that are based on novel computing technologies [1]. A web based application is an application which can be accessed and used over the network i.e. internet, intranet or extranet. Web applications have become more complex because of the increased use of distributed services, large number of hypermedia and stricter requirements of reliability, usability, inter-operability and security [2][3].

Testing is a process of providing correctness and quality of software. At the same time, the modifications that are incorporated into the application during testing must not adversely affect its basic functionality. This can be achieved with the help of regression testing [4]. Software testing is expensive, time consuming and labour intensive task. It requires 40-70% of software development cost. [5][6] Due to market pressure and high configuration cost of software testing the web application functional testing is often neglected by developers [7]. Cloud computing facility is one in which customers do not have to pay for infrastructure installation and maintenance cost [8][9]. As a user of cloud computing, one has to pay the service charges according to the usage of computing power and other networking resources [10]. Moreover, one need not worry about software updates, installation, email servers, anti-viruses, backups, web servers and both physical and logical security of the data. The web based applications are powerful and have the ability to provide feature rich content to a wide audience spread across the globe at an economical cost. Although powerful, these technologies bring new challenges for developers and testers. As, all entities of web application have to be tested [11].

More than 200 web application based functional testing tools exist either as online applications or on cloud environment but these tools use traditional techniques which are not suitable for complex structured web applications. The traditional techniques lack the additional features of testing such as regression testing, fixing errors, uniquely identifying hypermedia, testing database connection etc [12][13][14]. Regression testing is the process of selective retesting of a system or its component in order to ensure that modifications do not lead to undesired effects and that the system conforms to its desired functionalities. To retest a program after changes, regression testing technique enables selection of an adequate subset of the whole test suite on the condition that the selected subset will give the confidence about covering the changes. Regression tests often have very low yield in terms of finding bugs. Regression Testing is required

when there is a change in requirements and code is modified according to the requirement, when new feature is added to the software, for defect fixing and last but not the least for performance fixing issues.

In the proposed work CWEBTEST testing tool has been developed to overcome the problems of regression testing in the existing web application testing techniques. CWEBTEST testing tool test web applications based on unique hypermedia. With the CWEBTEST tool hypermedia are identified from the compiled HTML code and regression testing is performed on web application. To retest a program after changes, regression testing technique enables selection of an adequate subset of the whole test suite on the condition that the selected subset will give the confidence about covering the changes. CWEBTEST saves 10-25% test set size and more time is saved as compared to previous regression testing techniques. CWEBTEST has been published on Microsoft public cloud environment.

The contributions can be summarized as:

Analysis of web application based testing tool has been performed. During the analysis it has been observed that the traditional testing technique doesn't support regression testing.

Research papers on regression testing have been studied thoroughly to get a better idea about regression testing. These papers helped to get a clear vision of what regression testing is and where it can be best deployed for testing.

Problems of the regression testing tools and techniques has been studied thoroughly .The studied regression testing tools consumed more time for testing and test set size were also quite big. To overcome the problems of existing regression testing tool a new web application based regression testing tool CWEBTEST have been developed.

CWEBTEST has been implemented using C# Programming language using ASP.NET technology on Microsoft Windows Azure. The CWEBTEST tool reduces test set size by 10-25 percent and testing time is also saved by an extent.

The tool has been published on Microsoft public Cloud environment for experimental purpose and the performance of tested web applications has been found satisfactory.

## II. ANALYSIS OF WEB APPLICATION TESTING TOOL

No doubt a number of web application based functional testing tools exist either as online applications or on cloud environment but all these tools deploy traditional testing techniques. To study the behavior of existing tools analysis of twenty web application testing tool has been done. In the analysis phase comparison of some existing web application testing tools that are being used by MNC's is also carried out. Analysis of ten different cloud based testing tools have been done and the data is represented in Table 1 and other ten tools that has been analyzed are those that are online web application tools and the data analysis of these tools is represented in Table 2. Out of these twenty web application functional testing tools not even a single tool supports unique hypermedia regression testing.

Table 1: Analysis of Cloud Based Web Application Testing Tools

	Test Objective							Test Activity	
	Functionality Testing	Usability Testing	Interface Testing	Database Testing	Compatibility Testing	Performance/Load Testing	Security Testing	Scalability	Cloud Based
Soasta [20]	✓	✗	✗	✗	✗	✓	✗	✓	✓
Parasoftsoates t [21]	✗	✗	✗	✗	✗	✗	✓	✓	✓
Cloudtesting[ 22]	✓	✗	✗	✗	✓	✗	✗	✗	✓
Gcloud [23]	✗	✗	✗	✗	✗	✓	✗	✗	✓
Practitest [24]	✗	✗	✗	✗	✗	✓	✓	✗	✓
Blazemeter [2 5]	✗	✓	✗	✗	✗	✓	✗	✗	✓
Loadstorm [26]	✗	✗	✗	✗	✗	✓	✗	✓	✓
Keynote [27]	✗	✗	✗	✗	✓	✓	✗	✓	✓
Testmaker [28]	✓	✗	✗	✗	✗	✓	✗	✓	✓
Clap [29]	✗	✗	✗	✗	✗	✓	✗	✗	✓

Table 2: Analysis of Web Application for Functional Testing

	Test Static Hyperlinks	Test Unique Links	Test Entire Web App	Test Dramatic Web App	Test Image URL	Test External Links	Regression Testing	Graphical View of Report	Provide as a Services on Cloud Environment	Fixed Error
SpringTrax [30]	✓	✗	✓	✓	✓	✓	✗	✓	✗	✓
LinkTiger [31]	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗
Link Check Pro [32]	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗

Link Alarm [33]	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗
Alert Linkrunner [34]	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗
Link Scan [35]	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗
Link Audit [37]	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗
CyberSpder Link Test [38]	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗
Web Link Validation [39]	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗
W3C Link Checker [39]	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗

### III. PRELIMINARIES AND DEFINITION

In this section, first of all the web application regression testing is explained and then the definition of CWEBTEST steps is given and lastly the objective of the developed tool is defined.

#### A. Web Application Regression Testing

Regression testing of web application is a software testing activity that aims to uncover any new bugs, or regressions, that could arise after changes in the functional or non-functional areas of a system, such as patches, enhancement or configuration/environment changes. There are main three types of regression testing elaborated as:

**Bug-fix** as a process is aimed to ensure that fixed error had been really fixed.

**Old bug-fix** is an action to ensure that previously fixed and verified defect won't be reproduced in the system.

**Side-effect** is an attempt to check whether any other parts of the system hadn't been broken by the recent changes in the code.

As regression testing for web applications is an internal part of web development, web application should be tested against various regressions and new bugs. Thus, regression testing is essential part of the way delivering high quality and bug free software product to end-customer .

In the current period, it is hard to imagine successful business without websites. It doesn't matter what kind of services you offer to the clients (banking, tourism, healthcare, sport, etc.) as regression testing is just a stage of testing process – a stage that must to be included.

Table 3 Notations

Symbols	Physical Meaning
$W_{URL}$	Web Application URL
$S_{URL}$	Web Application URL in String
H	Hypermedia
$H_{Images}$	Images
$H_{Hyperlinks}$	Hyperlinks
$H_I$	Internal Source Hypermedia
$H_X$	External Source Hypermedia
$H_U$	Unique Hypermedia
$H_W$	Working Hypermedia
$H_D$	Dead Hypermedia

Notations are enlisted in Table 3 that are must to be defined before describing the CWEBTEST web application testing tool.

CWEBTEST definition consists of five steps:

**GetURL( $W_{URL}$ )**→In this step web application URL is taken as the input. It outputs a Web application page HTML script in string format ( $S_{URL}$ ).

**GetHypermedia( $S_{URL}$ )** →In this step  $S_{URL}$  is taken as input. Scanning of string is done for identifying the hypermedia .HTML tags are used for identifiyng the hypermedia(H). For example, for identifying a  $H_{Images}$  <img>..</img>is used and for the  $H_{Hyperlinks}$  <a>...</a> is used.

**CompreHypermedia( $H_{Hyperlinks}$  , $H_{Images}$ )** →In this step hyperlinks and image are taken as input. Then, identify unique hypermedia The output of this step is unique hypermedia ( $H_I$ ) .

**IdentifyHypermedia ( $H_I$ )** →In this step Unique Hypermedia is taken as input. Once hypermedia is identified, the source of hypermedia (internal or external) is checked. The output of this is a internal hypermedia ( $H_I$ ) or external hypermedia( $H_X$ ).

**TestHypermedia(H<sub>T</sub>)** → In this step Unique Hypermedia is taken as input. In this step working hypermedia is tested, if it is working then set status is working hypermedia otherwise dead hypermedia. The output is a working hypermedia (H<sub>W</sub>) or dead hypermedia(H<sub>D</sub>)

**SaveHypermedia(H<sub>T</sub>)** → In this step Unique Hypermedia is taken as input. In this step hypermedia is saved in a XML database with tags internal hypermedia (H<sub>I</sub>) or external hypermedia(H<sub>X</sub>) and working hypermedia (H<sub>W</sub>) or dead hypermedia(H<sub>D</sub>).

**B. Definition of Objectives**

A number of web application based testing research papers have been thoroughly studied. While going through those papers it has been noticed that previous testing tools lack regression testing and as a result of that they don't serve the purpose of testing in a better way. The tools that have deployed regression testing don't produce accurate results and more to that time consumed by them for testing is quite high. Keeping in mind the limitations of previous testing tools a new web application based regression testing tool has been developed that not only consumes less time for testing the web applications but also provides accurate results.

The objectives of the research work are formulated as summarized below:

Identification and analysis of the problems associated with web application tools.

Analysis of the problems associated with regression testing

Improve the efficiency of web application testing tool in terms of reduction in test data and testing time.

To achieve research objectives CWEBTEST testing tool has been developed and published on Microsoft Public cloud environment

**IV. CLOUD WEB TEST**

In this section, the detailed construction of CWEBTEST tool is delineated.

**A. Overview of the developed Tool**

With the emergence of new technologies web applications are being widely accepted all over the globe. They are gaining popularity at a high pace because more and more services and information is being made available over the Internet and Intranet, web portals have also become extraordinarily complex for performing functional testing. These web portals usually have thousands of hyper links, images, multimedia files, data files, audio files among many others. Web pages are modified frequently viz. adding links, making user specific customizations, adding new features and much other functionality in this way web applications are being modified constantly with additions of new features. Therefore it becomes necessary to test the web applications that are being modified. The modified web application can be tested by using regression testing. Regression testing is the process of selective retesting of a system or its component in order to ensure that modifications do not lead to undesired effects and that the system conforms to its desired functionalities. The previously existing tools that deploy regression testing don't produce accurate results and more to that time consumed for testing is quite high. Keeping in mind the limitations of existing testing tools a new web application based regression testing tool has been developed. The Proposed web application functional testing tool CWEBTEST is a unique hypermedia regression testing tool for testing web applications.

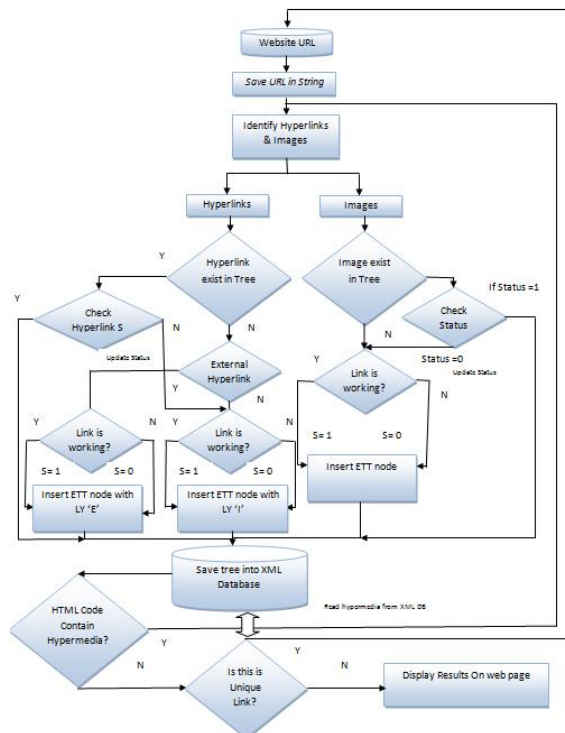


Figure 1 Proposed Model

In the proposed work, first of all the tool has identified the unique hypermedia like hyperlinks and images and then an Event Test Tree(ETT) has been generated and then unique hypermedia regression testing has been performed by comparing newly added hypermedia with ETT. Outputs of tool have been saved in the XML file. The web application with its interaction components and dependencies by the Event Test Tree and the relationship between two pages without loop has also been depicted.

**B. Algorithm for CWEBTEST**

Suppose a U has a web application’s URL. U has n number of hypermedia(H)  $U=(H_1, \dots, H_n)$ . In web applications hypermedia are updated frequently by web admin due to feature updates or user preference changes. Sometime these updates change the functionality of web application. Developer and tester need to perform regression testing on their web applications whether the updations have changed the functionality of web application or not. Regression testing is the process of selective retesting of a system or its component in order to ensure that modifications do not lead to undesired effects and that the system conforms to its desired functionalities.

Each hypermedia contains n number of images and hyperlinks  $H=(H_{image}^1, \dots, H_{images}^n, H_{hyperlinks}^1, \dots, H_{hyperlinks}^n)$ . Every web applications contain web pages. CWEBTEST test each page individually.

CWEBTEST definition consists of five steps:

**GetURL(W<sub>URL</sub>)**→In this step web application URL(W<sub>URL</sub>) is taken as the input. It outputs a Web application page HTML script in string format (S<sub>URL</sub>).

**GetHypermedia(S<sub>URL</sub>)** →In this step S<sub>URL</sub> is taken as input. Scanning of string is done for identifying the hypermedia .HTML tags are used for identifyg the hypermedia(H). For example, for identifying a H<sub>Images</sub> <img>..</img>is used and for the H<sub>Hyperlinks</sub> <a>...</a> is used. In this step all hyperlinks are identify of single web page.

$$H = \sum_{k=0, l=0}^{n, m} H^{Images(k)} + H^{Hyperlinks(k)}$$

$$H(T) = \sum_{p=0}^q H(p)$$

H is number of hypermedia of individual page, n is number of images and m is number of .hyperlinks on individual single page. H(T) is total number of hypermedia of web application, q is number of web pages in web application.

**CompreHypermedia(H<sub>Hyperlinks</sub>, H<sub>Images</sub>)** →In this step hyperlinks and image are taken as input. Then, identify unique hypermedia The output of this step is unique hypermedia (H<sub>i</sub>). To identify unique hypermedia each entry compare with previous entries of ETT(Event Test Tree).to identify hypermedia use a searching techniques example binary search or Linear Search

**Identify hypermedia(H<sub>i</sub>)** →In this step Unique Hypermedia is taken as input. Once hypermedia is identified, the source of hypermedia (internal or external) is checked. The output of this is a internal hypermedia (H<sub>i</sub>), H<sub>i</sub> or external hypermedia(H<sub>x</sub>). External links meaning hyperlinks that’s redirect to other web application web pages.

**TestHypermedia(H<sub>i</sub>)** →In this step Unique Hypermedia is taken as input. In this step hypermedia is tested , if it is working then set status is working hypermedia otherwise dead hypermedia. The output is a working hypermedia (H<sub>w</sub>) or dead hypermedia(H<sub>d</sub>). Testing time of hypermedia is

$$T = \sum_{n=0, m=0}^{p, q} (t_1 H^{Images(k)} + t_2 H^{Hyperlinks(k)})$$

$$T(Total) = \sum_{n=1}^r T(n)$$

T is time taken to test a individual page, p is number of images and q is number of .hyperlinks on individual single page. t1 is time taken to test images and t2 is time taken to test hyperlinks. T(Total) is time taken to test all page, of web application, r is number of web pages in web application.

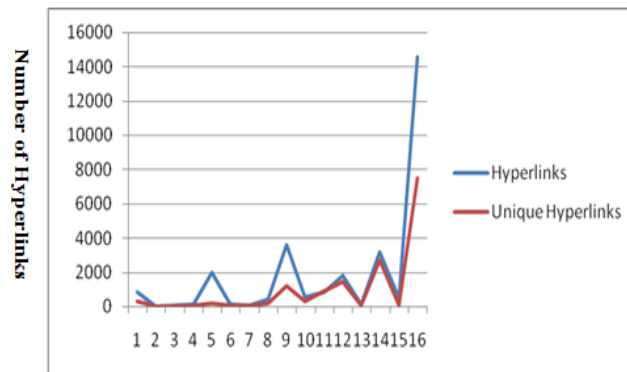
**SaveHypermedia(H<sub>i</sub>)** →In this step Unique Hypermedia is taken as input. In this step hypermedia is saved in a XML database with tags internal hypermedia (H<sub>i</sub>) or external hypermedia(H<sub>x</sub>) and working hypermedia (H<sub>w</sub>) or dead hypermedia(H<sub>d</sub>)

**V. RESULTS**

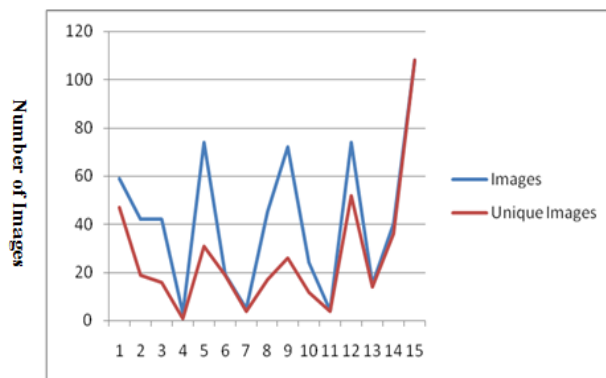
The CWEBTEST has been published on Microsoft public cloud environment for the experimental purpose. Microsoft cloud environment provides a Single Virtual Machine with 1.66 GHz CPU, 1.75 GB RAM and 2 GB bandwidth free of cost. CWEBTEST has been developed in C# 4.0 programming language using Microsoft cloud framework for ASP.NET web development technology in Microsoft Visual Studio 2010 IDE (Integrated Development Environment).

Some web applications that contain rich hypermedia have been considered for experiment for testing the performance of CWEBTEST web application regression testing tool. Parameters that have been taken for experiment are shown in Table 4. In the experiment first step that has been performed includes the identification of unique hypermedia. Hypermedia is used as a logical extension of the term hypertext in which graphics, audio, video, plain text and hyperlinks intertwine to

create a generally non-linear medium of information. But in this experiment hyperlinks and images have been taken. Figure 2 shows uniquely identify hyperlinks and Figure 3 shows uniquely identify images. In case of hypermedia 51% hyperlinks are unique and in case of image 65% images are unique.

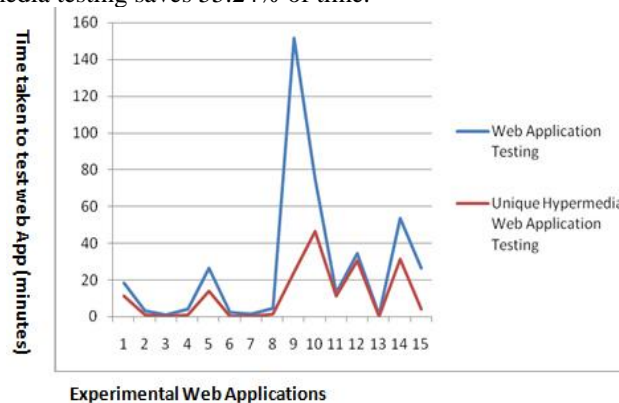


Experimental Web Applications  
Figure 2 Uniquely Identify Hyperlinks



Experimental Web Applications  
Figure 3 Uniquely Identify Images

The Table 5 represents the variation in execution time of entire website in different situations. In Figure 4 graph is representing the variation in execution time of web application hypermedia and unique web application hypermedia testing. In this experiment, testing of web application hypermedia and unique web application hypermedia are performed in same cloud environment. In web application hypermedia testing 6:49:3(HH:MM:SS) time is taken and unique web application hypermedia has taken 3:3:04 (HH:MM:SS) time to test experimental web applications. In the case mentioned unique web application hypermedia testing saves 55.24% of time.



Experimental Web Applications  
Figure 4. Hypermedia Testing and Unique Hypermedia Testing

In Figure 5 graph represent web application regression testing and unique hypermedia web application hypermedia regression testing. In this experiment testing of both web application regression testing and unique hypermedia web application hypermedia regression testing are done on same cloud environment. In web application regression testing 1:47:5(HH:MM:SS) time is taken and unique hypermedia web application hypermedia regression testing has taken 1:17:45 (HH:MM:SS) time to test experiential web applications. In the case mentioned unique hypermedia web application hypermedia regression testing saves 18.05% of testing time.



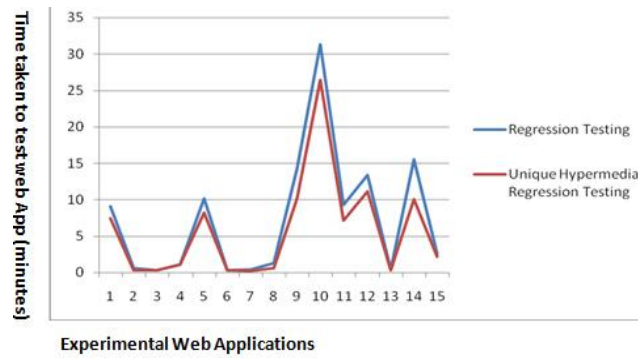


Figure 5. Regression Testing And Unique Hypermedia Regression Testing

## VI. RELATED WORK

In this section the regression testing based research papers have been discussed. These papers implemented regression testing but none of these papers have implemented regression testing based on unique hypermedia. The brief description of papers is as below:

Lian Yu, and Wei-Tek [15] have discussed Testing as a Service (TaaS) model for providing better testing capabilities to end users. Users can save the cost of complicated maintenance as well as software upgrade efforts as the service providers can upgrade their services automatically. Due to uneven volumes of concurrent requests, it is important to address the elasticity of TaaS

Platform in cloud environment. Scheduling and dispatching algorithm have been developed to improve the utilization of computing resources

Lei Xu and Baowen Xu [16] have modeled web applications as System Dependent Graph (SDG) and have introduced slicing along with regression testing technique for web application. Although the SDG has increased the workload and cost of the testing process but the slicing technique has provided simplified contents and thus has improved the working efficiency. But the model is not suitable for testing the web applications that contains hypermedia in tones.

Ricca Tarhini, Fouchal and Mansour [17] have modeled a web application and its component behavior as a two level abstract model. The model has taken into consideration timing constraints and has been represented by a Timed Labeled Transition Systems (TLTS). Further the author has suggested a safe regression testing algorithm for testing web applications. The proposed work has not been implemented in the real world scenario therefore nothing can be predicted about its outcome.

Tarhini, Fouchal and Mansour [18] have proposed a safe regression testing selection technique based on Event Dependency Graphs (EDG), however, cyclic redundancies have complicated the testing process visibly. Investigated by the challenges related to the diversified web applications and identified need to develop simplified techniques for automatically comparing test case output, the author has proposed an Event Driven Test case selection paradigm for regression testing of web application. The author has stated that the model saves 44-99 % time.

Akshi Kumar and Ruchi Goel [19] have discussed the event driven techniques, the author has generated an event dependency graph of the original and modified web application and then the original and modified web application graph has been converted into event test tree. Then the comparison of both trees has been done to identify affected and potentially affected nodes. As a result of this, selection of test cases has been enabled. For regression testing web applications has to reduce the test set size. The major limitations of the proposed work are that hyperlinks are not uniquely identified, repeated links are retested and a lot of time is wasted in the process of retesting the same links.

## VII. CONCLUSION AND FUTURE WORK

With the steady growth of web applications testing has become a tedious job. No doubt there a number of web application testing tools available but these tools use traditional techniques which are not suitable for complex structured web applications. The traditional techniques lack the additional features of testing such as regression testing, fixing errors, uniquely identifying hypermedia, testing database connection etc. Although regression testing is being widely accepted as it aims at detecting errors by comparing present behavior with the past behavior of the web application and it assures the reliability of software by providing information about the quality of an application but it suffers the limited use in the domain. In the thesis work CWEBTEST testing tool has been developed to overcome the problems of regression testing in the existing web application testing techniques. CWEBTEST testing tool test web applications based on unique hypermedia. CWEBTEST saves 10-25% test set size and more time is saved as compared to previous regression testing techniques. CWEBTEST has been published on Microsoft public cloud environment. The future work involves fixing errors that occur in the web applications.

Table 4: Parameters for Web Application Unique Hypermedia Regression Testing

S.No	Web Application URL	Hyperlinks	Unique Hyperlinks	Working Hyperlinks	Dead Hyperlinks	Images	Unique Images	Working Images	Dead Images
1	http://www.asp.net	864	296	291	5	59	47	46	1

2	http://www.computerscience mcq.com	30	5	5	0	42	19	19	0
3	http://www.csprojectsonline. com	68	9	8	1	42	16	16	0
4	http://www.google.com	127	87	83	4	3	1	1	1
5	http://www.csmcqs.blogspot.i n	2018	159	148	11	74	31	29	2
6	http://www.ppsc.gov.in	170	60	28	32	19	19	18	1
7	http://www.upsconline.nic.in	69	57	21	36	5	4	4	0
8	http://www.punjabpolice.gov. in	448	162	152	10	45	17	17	0
9	http://www.puchd.ac.in	3608	1173	1112	61	72	26	24	2
10	http://current-gk- today.blogspot.in	559	291	275	16	24	12	12	0
11	http://www.indianbloggers.or g	888	880	865	15	4	4	3	1
12	http://www.ubuntu.com	1840	1460	1454	6	74	52	52	0
14	http://www.gktoday.in	138	59	58	1	15	14	13	1
14	http://www.powerasp.net	3214	2713	2641	72	40	36	32	2
15	http://www.snapdeal.com	507	77	67	10	108	108	107	1

Table 5: Execution Time of Various Web Application Testing

S. No	Web Application URL	Test Hypermedia (MM:SS) I	Test Unique Hypermedia (MM:SS) II	Regression Testing (MM:SS) III	Unique Hypermedia Regression Testing (MM:SS) IV	Time Saved (I>II) %	Time Saved (IV<III) %
1	http://www.asp.net	18.43	11.39	9.11	7.36	37.69	23.77
2	http://www.computersciencemcq.com	3.11	1.25	.54	.31	55.44	74.19
3	http://www.csprojectsonline.com	1.12	.34	.31	.25	51.74	24
4	http://www.google.com	4.33	1.37	1.02	1.02	64.47	0
5	http://www.csmcqs.blogspot.in	26.42	14.11	10.11	8.16	46.85	23.89
6	http://www.ppsc.gov.in	2.36	.34	.28	.26	77.72	7.69
7	http://www.upsconline.nic.in	1.46	.46	.33	.19	56.78	73.68
8	http://www.punjabpolice.gov.in	4.43	1.53	1.22	.57	60.05	114.03
9	http://www.puchd.ac.in	151.43	24.33	14.23	10.11	83.88	40.75
10	http://current-gk-today.blogspot.in	75.00	46.34	31.33	26.43	37.90	18.53
11	http://www.indianbloggers.org	13.14	11.23	9.23	7.13	4.055	29.45
12	http://www.ubuntu.com	34.46	30.43	13.41	11.14	11.66	20.37
13	http://www.gktoday.in	1.03	.56	.40	.31	11.65	29.03
14	http://www.powerasp.net	53.43	31.40	15.53	10.09	41	53.91
15	http://www.snapdeal.com	26.28	4.48	2.56	2.13	81.82	20.18

## REFERENCES

- [1] W. Zhang, B. Xu, X. Zhou, D. Li, And L. Xu, Meta Search Engine Survey, IEEE International Computer Software And Applications Conference ,Pages 36-41, March 2001.
- [2] F. Ricca And P. Tonella, Analysis And Testing Of Web Applications, IEEE International Conference On Software Engineering, Pages 25-34, May 2001.
- [3] F. Ricca And P. Tonella, Web Site Analysis: Structure And Evolution, International Conference On Software Maintenance, Pages 76-86, October 2001.



- [4] C.H. Liu, D.C. Kung, P. Hsia And C.T. Hsu, Object-Based Data Flow Testing Of Web Applications, Conference On Quality Software, Pages 7-16, October 2001.
- [5] Marchetto A., Ricca F. And Tonella P, A Case Study-Based Comparison Of Web Testing Techniques Applied To Ajax Web Applications, Journal On Software Tools For Technology Transfer, Pages 477-492, December 2008.
- [6] Marchetto A., Tonella P., And Ricca, F., State-Based Testing Of Ajax Web Applications, IEEE Computer Proceedings Of The 23<sup>rd</sup> International Conference On Software Engineering, Pages 121-130, April 2008.
- [7] Mesbah, A And Van Deursen, A., Invariant-Based Automatic Testing Of Ajax User Interfaces, International Conference On Software Engineering (ICSE'09), IEEE Computer Society, Pages 210-220, May 2009.
- [8] Bu Sung Lee, Aggregating IaaS Service, 2011 Annual SRII Global Conference, Pages 335-338, February 2011.
- [9] B. Hayes, Cloud Computing, Communications Of The ACM (CACM), Pages 9–11, July 2008.
- [10] Koray Incki, Ismail Ari And Hasan Sozer, A Survey Of Software Testing In The Cloud, IEEE 6<sup>th</sup> International Conference On Software Security And Reliability Companion, Pages 18-23, May 2012.
- [11] Peng Zhenlong And Ou Yang Zhonghui, The Application And Development Of Software Testing In Cloud Computing Environment, 2012 International Conference On Computer Science And Service System, Pages 450 – 454, March 2012.
- [12] Xu L, Xu B, And Chen Q, A Scheme Of Web Testing Approach, Journal of Nanjing University, Pages 182-186, November 2004.
- [13] W. Zhang, B. Xu, X. Zhou, D. Li, And L. Xu, Meta Search Engine Survey, IEEE International Computer Software And Applications Conference ,Pages 36-41, March 2001.
- [14] S. Elbaum, G. Rothermel, S. Karre And M. Fisher, Leveraging User-Session Data To Support Web Application Testing, IEEE Transactions On Software Engineering, Pages 187–202, March 2005.
- [15] Philip Robinson and Carmelo Ragusa, Taxonomy and Requirements Rationalization for Infrastructure in Cloud-based Software Testing , Third IEEE International Conference on Cloud Computing Technology and Science, pages 454-461, December 2011
- [16] Philip Robinson and Carmelo Ragusa, Taxonomy and Requirements Rationalization for Infrastructure in Cloud-based Software Testing , Third IEEE International Conference on Cloud Computing Technology and Science, pages 454-461, December 2011
- [17] Tarhini, A, Fouchal H and Mansour N, Regression Testing Web Services-based Applications, ACS/IEEE International Conference, pages 163-170, December 2006
- [18] Tarhini A, Ismail Z and Mansour, Regression Testing Web Applications, International Conference on Advanced Computer Theory and Engineering, pages 902-906, October 2012
- [19] Akshi Kumar and Ruchi Goel, Event Driven test case selection for Regression Testing Web Applications, IEEE-International Conference on Advances in Engineering Science and Management, pages 121-128, March 2012
- [20] SOASTA CloudTest [Online] <http://www.soasta.com/>
- [21] Parasoft SOAtest [Online] <http://www.parasoft.com>
- [22] CloudTesting's [Online] <http://www.cloudtesting.com/>
- [23] GCLoads' Proxy Sniffer [Online] <http://gclload.com>
- [24] PractiTest [Online] <http://www.practitest.com/>
- [25] BlazeMeter [Online] <http://blazemeter.com/>
- [26] LoadStorm [Online] <http://loadstorm.com/>
- [27] Keynote [Online] <http://www.keynote.com/>
- [28] TestMaker [Online] <http://www.pushtotest.com/>
- [29] CLAP Tool [Online] <http://www.persistentsys.com/>
- [30] SpringTrax [Online] <https://springtrax.com/>
- [31] LinkTiger [Online] <http://www.linktiger.com/>
- [32] LinkCheckPro[Online]<http://www.link-checker-pro.com/>
- [33] Link Alarm [Online] <http://www.linkalarm.com/>
- [34] AlertLinkrunner[Online] <http://www.alertbookmarks.com/lr/index.htm>
- [35] Link Scan [Online] <http://www.elsop.com/>
- [36] LinkAudit[Online] <http://www.blossom.com/audit.html>
- [37] CyberSpderLinkTest[Online] <http://www.cyberspyder.com/cslnkts1.html>
- [38] Web Link Validation [Online] <http://www.relsoftware.com/>