Model Based Testing for Web Applications: A Literature Survey Presented

Hasan Javed, Nasir Mehmood Minhas*, Ansar Abbas, Farhad Muhammad Riaz
Department of Computer Science, University Institute of Information Technology, PMAS-Arid Agriculture University Rawalpindi, Pakistan.

* Corresponding author. Tel.:+923335651973; email: nasirminhas@uaar.edu.pk
Manuscript submitted November 15, 2015; accepted January 25, 2016.
doi: 10.17706/jsw.11.4.347-361

Abstract: The World Wide Web has a great impact in the world of computing. Testing of web applications is becoming more challenging task with the tremendous growth, distributed nature, dynamic nature and heterogeneity of web applications. With the growing complexity and usage of web applications, there is a need of rigorous testing techniques for producing reliable applications. Model-based testing (MBT) is a promising paradigm for generating test cases from models of the system under test (SUT). Different techniques (based on model based testing) have been presented in the literature. The focus of this study is to present a survey of model based testing techniques with specific reference to web applications. Existing literature has been surveyed using a systematic literature review (SLR) approach. Applicability of MBT for web applications have been studied, comparison of existing approaches has been presented in the study, finally strengths and limitations of existing approaches have been highlighted.

Key words: Model based testing, system under test, web applications, systematic literature review, object oriented.

1. Introduction

There has been a remarkable growth in the usage of Web applications in our daily lives since the creation of the World Wide Web in the early 1990s [1]. The World Wide Web considers as the main aspect of the world of computing. The web has recognized as a powerful source for delivering software services over the Internet [2]. Now day's web applications are used in every field of life such as education, business, government, entertainment, industry and daily social life [2]–[6]. Web application can handle tasks that were handled with desktop applications before such as image editing or spreadsheet creation [7]. A web application typically comprises of front end (web pages) and a backend (database) with which user interact through the browser [1], [8]. There are two types of web applications; one is static in which users only view web pages while other is dynamic in which user can interact through input and modify content of web page [9], [10].

Web applications provide many advantages over desktop applications like: Cross platform nature, No need of installation [2], [11], accessibility around the world at same time [2], [11], [12], automatic up gradation with new features [2], [12]. Web applications are more useful as compared to desktop applications, but these advantages introduce new challenges for their quality assurance and testing. These challenges include:

1) Distributed nature of web application such as client/server architecture [11].

2) Dynamic nature which introduces several problems such as frequent technology change, changing nature of user requirement, testing methods and tools.
3) Heterogeneity; that these applications are developed in different programming language (HTML, JavaScript, CSS) for client side and (RUBY, PHP, Java) for server side [13].
4) Multiple user access.

Various studies show importance of web application testing. In 2003, Business Internet Group San Francisco (BIG-SF) [2] conducted a study in which they stated that 70% of web applications contain defects. Due to cross platform dynamic and multi lingual nature of web applications rigorous testing techniques are needed which are cost effective and efficient [13]. As most of web applications are critical to business operations so they should be tested thoroughly and frequently to prevent from defects, but it is difficult to perform testing with traditional testing techniques as these techniques are costly and time consuming [14], [15]. Researchers are working to find viable approaches for testing web applications. Model based testing (MBT) is possible approach for web application testing as it provide many benefits such as high fault detection and reduced cost and time [5], [16]. A model is an abstract representation of the system under test (SUT) [17] which is used to extract information from software related to our purpose. The tester uses her/his knowledge of the SUT in modeling, to design a test model [18].

Arilo *et al*. [14] provide a detailed survey on the MBT techniques. After that, a lot of work is presented in literature on various MBT approaches for web applications and we found no such type of survey in recent years. So, before moving towards new solutions and approaches, there is a need to synthesize existing work.

This paper presents a survey of model based testing techniques with specific reference to web based applications. Initially application of MBT in web domain has been investigated, then comparison of available techniques has been presented as a result the strengths and limitations of these techniques have been highlighted. The goal was to provide guidance to the testers to select a suitable MBT approach and provide a basis for future research in the domain.

The rest of the paper is organized as follows. Section 2 describes related work, which presents the previously conducted surveys, SLRs on MBT techniques. Section 3 highlights the research questions which provide basis of this survey. In Section 4 complete procedure of systematic literature review has been presented. In Section 5, results obtained from SLR have been discussed against each research question. Section 6 is dedicated for the analysis of MBT approaches for web applications. Finally conclusion and future work are placed in section 7 and section 8 respectively.

2. Related Work

MBT is a promising approach for generating test cases from models of the system under test (SUT). These models are the description of system behaviour. Model development is based on information acquired from requirement or specified document. Many authors work in MBT for web applications testing. Recently Serdar Dogan *et al.* present [2] SLR on web applications testing. They compare different tools and models like types of models and fault models, empirical studies and their implementation for web applications. In this survey they take fault models and taxonomies as primary studies. In test models most models relate to navigation models some other models are control and data flow models and DOM models. In fault models/taxonomies, they build a taxonomy of 50 faults related to browser incompatibility and synchronization. Metrics are also used by the authors to measure cost and effectiveness. 4 types of metrics are used like effort/test time, test-suite size, memory space and others. According to this SLR measuring effort/test time is mostly used in literature. They also discuss emerging trends in web applications testing, and provide future area for research to practitioners and researchers. The limitation of this SLR is it only

covers functional testing of web applications. The other limitation is it caters MBT for web applications on a limited scale. In another Mahesh Shirole and Rajeev Kumar [19] presented a survey on UML behavioral models. In this survey different MBT approaches are classified into groups theoretic (tree based, graph based), formal specification (OCL) and UML specification languages (Sequence Diagrams, state chart, activity diagrams). In this survey authors do cover web applications testing but narrowly. Arilo Claudio Dias Neto and Guilherme Horta Travassos [20] presented a survey on characterization of attributes like (behavioral model, complexity level, indication of MBT approach, indication of supporting tools) etc. for model based testing. This characterization is based on two aspects like observing in which adequacy of attributes is checked against MBT approaches and in second step relevancy of these attributes checked. Giuseppe A. Di Luccaa, and Anna Rita Fasolino presented [11] a comparative survey of traditional testing and web applications testing. Functional testing is a focus area in this survey. They provide a list of relevant contributions in the area of functional web application testing. They provide future directions for survey such as models survey, which is our motivation for this research. Vahid Garousi *et al* [21] presented a systematic mapping study on testing web applications. In this study they surveyed techniques, tools and levels testing in web applications. They mapped the research results according to five aspects like Types of papers by contribution facet (based on contribution), Types of papers by research facet (solution proposals), Type of testing activity (code or model coverage), test locations (testing side like client or server) and testing level (which level require more testing), these aspects are their research questions for their mapping study. Limitation of this study in our context is this study covers Model Based testing on limited scale. Mohamed Mussa *et al* [18] presented a survey on model driven techniques. In this survey comparison is made according to different parameters like modelling language (UML), system design artefacts, test case generation (conditions need for automatic test case generation), testing target (target area like in models or in implementation) and tool support. This survey neglects web applications. Eddy Bernard *et al* [22] presented a survey of MBT from UML models. The UML models which are used in their survey are class diagrams, instance diagrams, state machine diagrams and object constraint language. Test selection criteria is made according to transition based, decision based and data oriented. The limitation of this paper is it covers only some behavioural models and neglects other models like FSM, EFSM, Graphs and algorithms which we cater in our survey. Recently Yuan-Fang Li *et al* [1] presented a comprehensive survey of latest Web testing advances and discuss their goals, targets, techniques employed, inputs/outputs and stopping criteria. In this survey they surveyed different testing techniques like model based testing, scanning and crawling techniques, search based testing, concolic testing and random testing used for testing web applications. In this survey they also discussed different testing techniques for ensuring application functions consistently with specification like graph based and FSMs and Probable FSMs. This survey is comprehensive survey which covers testing of web applications, but the limitation of this survey regarding our context is that it only covers model based testing as finite state machines and probable FSMs and neglect other models like UML (activity diagrams, state diagrams, class diagrams). Arilo *et al* [14] provide a detailed systematic review on the MBT techniques. In this systematic review author covers MBT approaches for different domains like OO, COTS, critical systems and web applications. However, before moving towards new solutions and approaches, there is a need to synthesize existing work. From the above discussion it is concluded that authors have presented surveys and SLRs on model based testing, some of the authors covered Web application testing but not as a major focus. Our work is unique in a sense that, it is only focusing the model based testing for web applications. The aim of this survey is to provide an opportunity for researchers (academic and domain experts) to have a closer look on the progress in the research area and to propose new ideas for future research related to web applications.

3. Research Methodology

The goal of this research is to provide a survey of MBT approaches for web applications, the basis of this survey is based three research questions. These research questions define the scope of this study and provide a way to extract desired information about the MBT approaches for web applications. RQ1 aims to extract the extent of applicability of model based testing for web applications. RQ2 is to investigate various MBT approaches for web application testing. Finally RQ3 is formulated to study the characteristics, strengths and limitations of these approaches. Research questions are given as under:

RQ1: What is the extent of applicability of MBT for web applications?

RQ2: Which model based testing approaches are being used for web application testing?

RQ3: Which area these approaches addresses and what are the limitations of these approaches?

4. Systematic Literature Review

We perform a systematic literature review for the answers to our research questions. As the systematic literature review is "a mean of identifying, evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest" [23]. Our purpose for using this methodology was to conform to our goal. Web applications gained interest in every field so research studies are scattered in different sources, this is why we perform a systematic literature review to find the answer of our research questions. This systematic literature review is performed on the basis of principles describe by (Mikael Svahnberg *et al.*) [23].

4.1. Search Strategy

This section elaborates the overall search strategy of the systematics literature review, Items that have been displayed in the coming sections are, search string, digital research databases, data selection, data extraction, and data synthesis.

4.1.1. Search string

Search strings were formulated and applied to search in different databases as title, abstract and keywords and full text for collection of publications in all databases.

("model based testing") or ("model driven testing") or ("specification based testing") and (approach or method or methodology or technique) and ("web application") or ("web services") or ("WWW") or ("internet based applications").

Table 1. Inclusion Criteria

Inclusion Criteria	
IC1	The article written in English language
IC2	The publication year of the paper is from 2006 to the point of conducting the search (2014).
IC3	The title or abstract discusses model based testing for web applications
IC4	The article is available in full text
IC5	The introduction discusses model, method, technique or tool related to MBT for web applications

Table 2. Exclusion Criteria

Exclusion Criteria	
EC1	The article is a duplicate of an already included article.
EC2	The article mainly discusses challenges and problems in MBT for web applications, but does not provide any beneficial solution or suggestion to solve such problem.
EC3	Books chapters
EC4	Studies that are only available as abstracts or PowerPoint presentations
EC5	Do not provide an empirical basis for their findings.

4.1.2. Databases

We use different databases which are in our access for data collection like IEEE Explore, ACM Digital Library, Springer-Link, Science Direct and search engine google scholar.

4.1.3. Data selection

Inclusion and exclusion criteria apply for studies selection to ensure accuracy and efficiency of data from identified studies. Inclusion criteria is presented in Table1. As seen from the table that we limit our research from 2006 to 2015, because most of the research concerning our area published during this tenure. Same as inclusion, exclusion criteria is also presented in Table 2 which helps us to exclude all those publications which are not lying into our selection process. Finally, all publications were read carefully to check against inclusion/exclusion criteria so that the final set of accurate data should be obtained.

4.1.4. Data extraction

We formulated a data extraction form, given in Table 3 to extract required information from selected publications to answer research questions as well as quality assessment. Data extraction for RQ1 is not shown in the data extraction form as studies found related to MBT for web application will automatically show their applicability. Before using this form in SLR, uncertainties were eliminated after carefully checking by all authors.

Table 3. Data Extraction Form

Data	**Value**	**RQ1**
Title of the paper		
Year of Publication		
Publisher		
Name of approach		RQ2
Behavioural Model		
Automation level	Low Medium High	
Tool Support	Yes No if yes mention name	
Algorithm Support	Yes No if yes mention name	
Functional Testing	Yes No	
Non Functional Testing	Security Performance Vulnerability Reusability Changeability Portability Interoperability Others	
Limitations	Focus Area, List limitations	RQ3

4.1.5. Data synthesis

After inclusion and exclusion criteria data collected is mostly in qualitative form. In this step we analyze that which different papers highlight same problem or provide a solution to the same questions and grouped them together.

4.2. Execution and Results

Search strings were applied to get related studies in different databases mentioned above. After applying search strings, we get 2892 studies in the first iteration. In second iteration we read the title and abstracts of these papers which clearly addressed model based testing for web applications and we got 207 papers. In third iteration, we downloaded these 207 papers and read all. We applied inclusion and exclusion criteria and finally selected 45 papers. Results from each source are presented in Table 4 whereas Table 5 represents our three iterative phases and total number of studies in each phase. Percentage of selected studies from each source is presented in Table 4.

Table 4. Amount of Selected Papers

Database	No. of Studies	No. of selected studies
IEEE Explore	1549	17
Springer Link	150	4
ACM	373	9
Science Direct	90	8
Others	730	7
Total	2892	45

Table 5. No. of Studies at Each Iteration

1st Iteration	2nd Iteration	3rd Iteration
2892	207	45

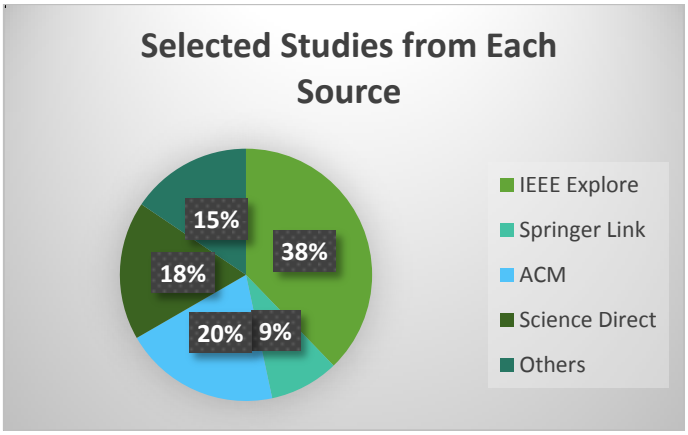

Fig. 1. Selected studies from each source.

5. Results Discussion

In this section we elaborate our findings against our research questions.

RQ1: What is the extent of applicability of MBT for web applications?

To justify our research question we present here different models and their applicability in web applications for different purposes like Simple Load Model [3], Work Load Model Using EFSM [24], On the fly testing Model [17], UML4MBT [25], Web Penetration test Model [26], for security and URMG [5], CMC, UCTM [11], Simulation Workload Performance Analyzer [27] for load management purposes, Simulation Workload Performance Analyzer [28] for performance and so on. Complete detail of these models according to their specific areas of applicability is shown in below Table 6.

RQ2: Which model based testing approaches are being used for web application testing?

In this question we performed a systematic literature review and pick different models used in web application testing. Our aim of choosing these models that, models which cover a specific area related to web applications. The name of these models which are taken in this survey are shown below. Models are named in the model used column like UML Class diagrams, activity diagrams, state diagrams other than

UML like Finite State Machines, Extended Finite State Machines (EFSM), Graphs, Formal languages like Z etc. It is clear from given table that most models are implemented in finite state machines and Graphs. Some models are implemented using ontologies, Z specification, some implemented using algorithms and some in UML diagrams.

Table 6. Applicability of MBT

Approach	Paper ID
Load Testing	[3], [10], [11], [16], [29]
Performance	[11], [22], [25], [27], [30], [31]
Security	[4], [9], [12], [24], [25], [32], [33]
Verification & Validation	[26], [34]
Navigation	[6], [26], [32], [35]
Conformance	[22]

Table 7. Models Used

Paper ID	Title	Model Used
[3]	Model-based testing for Web applications	Finite State Machines
[6]	Analysis of navigability of Web applications for improving blind usability	Graphs like shortest path and weighted directed graph
[9]	Automatic Model Inference of Web Applications for Security Testing	Z specification
[10]	An Automated Model Based Approach to Test Web Application Using Ontology	Graphs, test case generation using ontologies
[13]	Scalability issues with using FSMWeb to test web applications	Finite State Machines
[15]	Model-driven testing for web applications using abstract state machines	Finite State Machines, Abstract State Machine
[16]	LTF: A Model-Based Load Testing Framework for Web Applications	Activity Diagrams, Finite State Machines
[24]	Dynamic test input generation for web applications	Finite State Machines, Algorithm
[25]	Model-Based Vulnerability Testing for Web Applications	State diagrams, Class Diagrams
[32]	A model based testing technique to test web applications using state charts	State charts
[34]	Model-based Web Components Testing: Prioritization Using MIDS and Centrality Measures	Greedy algorithm applied to Undirected graphs, shortest path graphs
[35]	A model-based approach for crawling rich internet applications	Finite State Machines, Algorithms
[36]	Mutation Analysis of Magento for Evaluating Threat Model-Based Security Testing	Threat Models (Threat Tree, Threat nets)
[37]	A model-based approach for testing the performance of web applications	Extended Finite State Machines
[38]	A study of usage-based navigation models and generated abstract test cases for web applications	Directed Graphs
[39]	Model-based testing of web service compositions	Algorithm, Finite State Machines
[40]	Automatic generation of test drivers for model inference of web application	Parametrized Finite State Machine, Extended Finite State Machine
[41]	Towards Specification Based Testing for Semantic Web Services	Ontologies, Finite State Machines)

RQ3: Which areas these approaches address and what are the limitations of these approaches?

In this question our aim is to identify gaps in these approaches and provide a future direction for research in these lacking areas and produced better approaches in terms of time and cost. Focus Areas and limitations are shown in given table 8. These identified limitations are complex model [3] [16] [26],

annotation cost [6], accuracy [6], lack in tool maturity [10], State space explosion [13], scalability issues [16], error prone [28], time consuming [32], [36], [42], multi components with the same degree [34], performance issues [35], lack of automation [41], quality assurance [41] etc. Focus areas and limitations are shown in given Table 8.

Table 8. Focus Area and Limitations

Paper Id	Technique	Focus Area	Limitations
[3]	SLM (Simple Load Model) RUM	Generate realistic load for load testing.	RUM support only simple parameter, Complexity of Model with more branches add. SLM does not support think time.
[5]	URMG	Optimize evaluation and execution process.	Not suited to performance evaluation, loss of characteristics in characterization.
[6]	Landmark-oriented nonvisual navigation model	Navigability of real world applications	Accuracy, Author annotation cost,
[9]	Automatic and Vulnerability-driven Model inference approach	Vulnerability detection, data handling.	Complex Vulnerabilities, scanner doesn't traverse all accessible pages.
[10]	Structural Model using ontology	Automation of test cases for form filling, evaluation of dynamic feature	Difficulty in form filling, lack of maturity in tool
[12]	Model Checker	Security	Lacks in Bridging the gap between an abstract attack trace output by a model-checker and a penetration test on the real web application
[13]	FSMWeb Model	Scalability	State space explosion.
[16]	CBSAM (Context-based Sequential Action Model)	System workload for performance testing	Do not model request wait times explicitly, lacks in controlling the external factors
[26]	Probalistic Model	Navigability, quality	Complex Model, Complexity in probability table building
[28]	CFG	Data flow analysis and testing	Understanding and testing of Jsp's are difficult, does not have compiler checking so error prone.
[34]	ORD design Model	Verification and validation	Multi components with same degree, same BCs, attribute addition.
[35]	Model Based Crawling	Navigation	Performance as test case execution is not satisfactory.
[36]	Threat Models (Threat trees, threat nets) for automated test generation	Security	Security creation manually time consuming, requires in depth understanding about functionalities, requirement, source code.
[39]	ESG (Event sequence Graph) for Web services Composition	Web Services Composition	No distinction between orchestration and choreography
[41]	WSMO Web Services Modeling Ontology	Web Service Testing	Lack Of automation, quality assurance

[42]	NuSMV Model	Performance	Time consuming as DSL require hierarchical model, Lack in paths specified by user.
[43]	Navigational Behavioral Model	Automate the detection of vulnerability, accuracy and precision	Application discovery, Generation of many false positive results, Model-based fuzzing
[44]	Work Load Model using EFSM	Performance	Limited support to inter dependency & data dependency.
[45]	MBT using state charts.	Performance	Lack in front end modelling. Browser Compatibility

6. Analysis of MBT Approaches for Web Applications

In this section we analyze Model based testing approaches for web applications against identified parameters, these parameters are Models Type, Automation Level, Algorithm Support and Test Coverage. These parameters are important as when we apply MBT approaches for web applications issues can occur [14], so by taking these parameters into consideration these issues could be catered.

Table 9. Coverage Parameters

Parameter	Importance for MBT
Model Used	Characteristics which are tested by MBT approach are represented by models. These models are adapt in specific domains like some support UML based implementation, some implemented using EFSM, FSM and some support Graph based implementation.
Automation Level	The main aspect which focus in MBT is automatic test case generation using some tool, main reason for this is to minimize cost, time and effort for test case generation. If tool generates all test cases its support is consider as high and if it supports semi-automated test case generation is consider as medium and if test cases generation is manual then support is low.
Algorithm Support	Algorithm support is another challenging task for choosing correct approach. A model is either supported by an algorithm or not. The models which have algorithm support are easily understood.
Testing Coverage	This phase is consider as difficult phase in choosing which type of testing is cover in specific approaches used in this survey. Both types like functional and non-functional testing are analysed in this step.

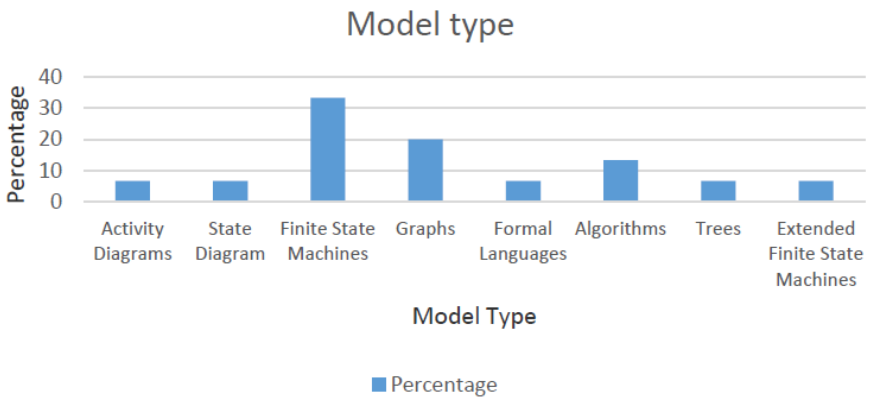

Fig. 2. Model type.

6.1. Model Type

This is a key issue in choosing the appropriate approach of MBT for web applications, for generating test cases, selection of correct approach is important [14]. Each approach has its limited scope, i.e. which specific area it covers and in which area it lacks. These approaches are applied according to their specification. These approaches must be integrated into the software domain [14] and then web applications, for designing and coding purpose building of separate models is time consuming activity, so

models must be integrated to support design and code. We classify these models according to their implementation like UML supported models (Class diagrams, state chart, activity diagrams) and other like using z specification (Z languages, ontologies), graphs, trees, (Extended) Finite State Machines.

6.2. Automation Level

MBT approaches feasibility can be determined by automation levels. Tools must support in automating (by models) and non-automated (by generation) way [14]. Complex and non-automated approaches are unfeasible [16]. Level of complexity of approaches is depends on automation of test cases, as manual has high complexity semi-automated has medium and automated has low complexity. Tool support is also an important aspect in choosing the correct approach, as without tool support implementation of approach is difficult. Manually generation of test cases with any approach is not efficient way and caused time and cost overhead. Approaches which are supporting tools generate test cases automatically so time and cost reduce by integrating tools with approaches. We characterize three types of criteria in our survey low, medium and high level support of automation. If tool support is available for a model and test cases are generated from the tool completely, support is rated as high, if test cases are generated by some graph or algorithm and have not a tool support rated as medium and if an approach not have tool support and test cases are generated manually rated as low. By observing Fig. 2 we clearly get idea that approximately 58 % approaches have a medium level of test case generation, such as these approaches have some tool support, about 39% approaches have full tool support and 3% have not any tool support.

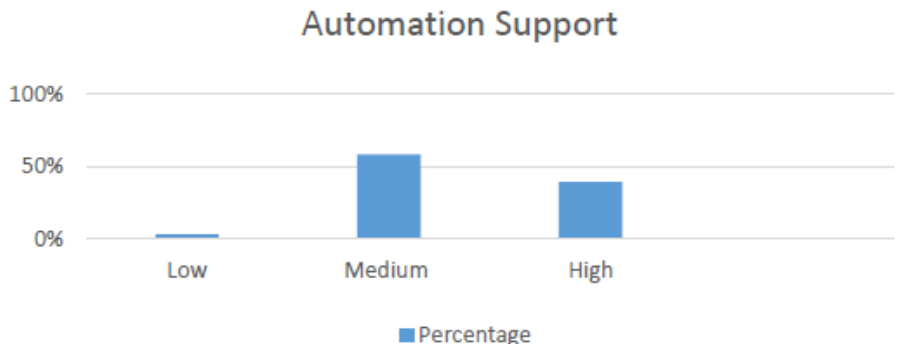

Fig. 2. Automation support.

6.3. Algorithm Support

Algorithm support is another challenging task for choosing correct approach. A model is either supported by an algorithm or not. The models which have algorithm support are easily understood for a person who is responsible for test case generation as compared the model which don't have algorithm Support. The Algorithm provide step wise information for test case generation which is simple and easily understood for tester. The analysis for algorithm support is presented in given Fig. 3. It is clear from the figure that approximately 65% approaches used in this survey have not algorithms support. Only few approaches provide stepwise implementation by algorithms like 35%.

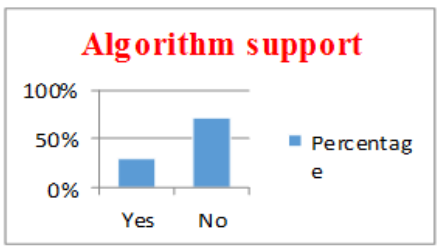

Fig. 3. Algorithm support.

6.4. Testing Coverage

Testing Coverage considers perhaps difficult phase. In this type we surveyed that which approach support which type of testing such as functional or non-functional. In this step we perform in depth analysis and analyze approaches that if an approach support non-functional testing, then which type such as security, performance, reusability and others, but for functional testing we only check whether a technique support a functional testing or not. These results are shown in Fig. 4. The results are clear from figure like 57% approaches support functional testing. In Fig. 5 we present detail of each non-functional testing and percentage of supporting approaches according to these non-functional testing. As web applications support portability so much work is done in this area. In this figure we show results according to each non-functional testing like security 10%, performance testing15%, vulnerability 10 %, reusability 15%, changeability 5%, portability 2%, interoperability 4% and we take others like adaptability, integrity, correctness, completeness etc. taken in other category which are 39%.

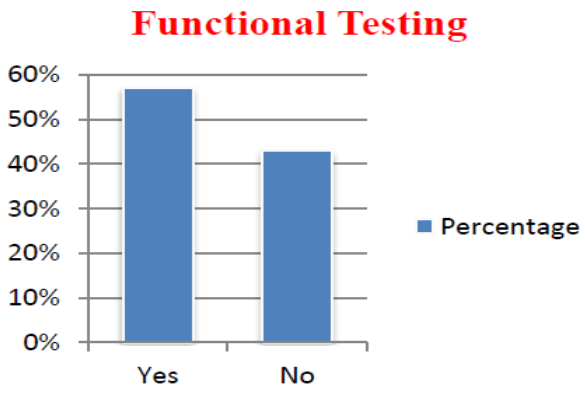

Fig. 4. Functional testing.

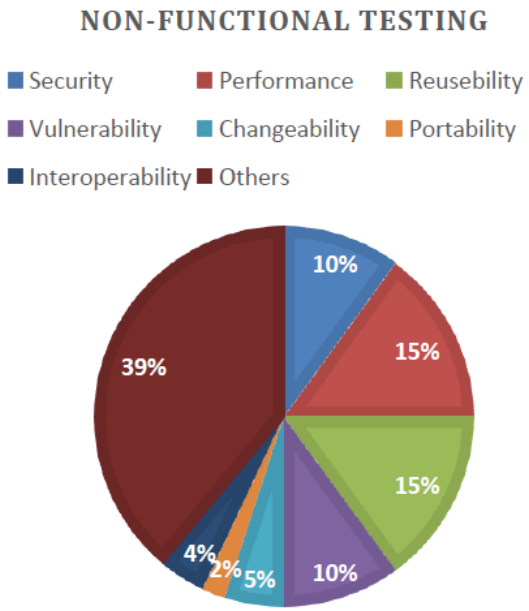

Fig. 5. Non-functional testing.

7. Conclusion

Web is the major source for delivering services from the internet. Testing in web applications is the main hurdle for developers due to its complexity. As most of web applications are critical to business operations so they should be tested carefully and regularly to prevent from defects. But it is difficult to perform testing

with traditional testing techniques as these techniques are costly and time consuming. Researchers are working to find viable approaches for testing web applications. Model based testing (MBT) could be a possible approach for web application testing as it provides many benefits such as high fault detection and reduced cost and time.

In this survey, we perform Systematic Literature review, which is based on three questions. In Q# 1 the focus was the applicability of Model based testing for web applications in different domains, there are some approaches which are applied in web applications domains like navigation, performance testing, load testing and security. In Q#2 we identified different models used in these approaches. The majority of models used in the identified approaches are based on behavioral models like, Finite State Machines, Graphs, UML Activity Diagram, State Diagram, Formal Languages (Z specification), trees and Extended Finite State Machines. The purpose of Q#3 was to identify areas which these approaches address and to find out the limitations of the identified techniques. The areas which are the focus of the identified techniques are navigation, security, performance, and load testing. While highlighted issues/limitations are complexity of models, accuracy, state space explosion, complex vulnerabilities, and lack in tool support, etc.

Our survey presents state of art of Model based testing approaches proposed for web applications. It provides future direction to researchers for analyzing existing research and planning work for different Web application areas like navigation, security and load testing.

8. Future Work

In this research our aim for performing a literature survey to identify different MBT approaches for web applications and identify which areas these approaches address. We also identify gaps in these approaches for future research. In future we are aiming to further enhance our research in this area and try to overcome the identified issues in the proposed model based testing technique for web application. As in this research we study different tools supporting different techniques have some issues, we have aim to develop a supporting tool which overcome stated limitations.

References

[1] Li, Y. F., Das, P. K., & Dowe, D. L. (2014). Two decades of Web application testing — A survey of recent advances. *Information Systems*, *43*, 20-54.

[2] Dogan, S., Betin-Can, A., & Garousi, V. (2014). Web application testing: A systematic literature review. *Journal of Systems and Software*, *91*, 174-201.

[3] Miao, H. K., Chen, S. B., & Zeng, H. W. (2011). Model-based testing for Web applications. *Jisuanji Xuebao(Chinese Journal of Computers)*, *34*(*6*), 1012-1028.

[4] Ran, L., Dyreson, C., Andrews, A., Bryce, R., & Mallery, C. (2009). Building test cases and oracles to automate the testing of web database applications. *Information and Software Technology*, *51*(2), 460-477.

[5] Xu, X., Jin, H., Wu, S., Tang, L., & Wang, Y. (2014). URMG: Enhanced CBMG-based method for automatically testing web applications in the cloud. *Tsinghua Science and Technology*, *19(1)*, 65-75.

[6] Takagi, H., Saito, S., Fukuda, K., & Asakawa, C. (2007). Analysis of navigability of web applications for improving blind usability. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *14*(3), 13.

[7] Nabuco, M., & Paiva, A. C. (2014). Model-based test case generation for web applications. In *Computational Science and Its Applications–ICCSA 2014* (pp. 248-262). Springer International Publishing.

[8] Hossen, K., Groz, R., Oriat, C., & Richier, J. L. (2014, March). Automatic model inference of web applications for security testing. *Proceedings of the Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh International Conference on* (pp. 22-23).

[9] Hajiabadi, H., & Kahani, M. (2011, September). An automated model based approach to test web application using ontology. *Proceedings of the 2011 IEEE Conference on Open Systems (ICOS)* (pp. 348-353).

[10] Wang, X., Zhou, B., & Li, W. (2013). Model-based load testing of web applications. *Journal of the Chinese Institute of Engineers*, *36*(*1*), 74-86.

[11] Di, L. G. A., & Fasolino, A. R. (2006). Testing web-based applications: The state of the art and future trends. *Information and Software Technology*, *48*(*12*), 1172-1186.

[12] Büchler, M., Oudinet, J., & Pretschner, A. (2012, June). Semi-automatic security testing of web applications from a secure model. *Proceedings of the 2012 IEEE Sixth International Conference on Software Security and Reliability (SERE)* (pp. 253-262).

[13] Andrews, A. A., Offutt, J., Dyreson, C., Mallery, C. J., Jerath, K., & Alexander, R. (2010). Scalability issues with using FSMWeb to test web applications. *Information and Software Technology*, *52*(*1*), 52-66.

[14] Dias, N. A. C., Subramanyan, R., Vieira, M., & Travassos, G. H. (2007, November). A survey on model-based testing approaches: a systematic review. *Proceedings of the 1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies: Held in Conjunction with the 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE) 2007* (pp. 31-36).

[15] Bolis, F., Gargantini, A., Guarnieri, M., Magri, E., & Musto, L. (2012). Model-driven testing for web applications using abstract state machines. *Current Trends in Web Engineering*.

[16] Zhou, J., Zhou, B., & Li, S. (2014, October). LTF: A model-based load testing framework for web applications. *Proceedings of the 2014 14th International Conference on Quality Software (QSIC)* (pp. 154-163).

[17] Ernits, J., Roo, R., Jacky, J., & Veanes, M. (2009). Model-based testing of web applications using NModel. *Testing of Software and Communication Systems*.

[18] Mussa, M., Ouchani, S., Al Sammane, W., & Hamou-Lhadj, A. (2009, August). A survey of model-driven testing techniques. *Proceedings of the 9th International Conference on Quality Software* (pp. 167-172).

[19] Shirole, M., & Kumar, R. (2013). UML behavioral model based test case generation: A survey. *ACM SIGSOFT Software Engineering Notes*, *38(4)*, 1-13.

[20] Dias, N. A. C., & Travassos, G. H. (2008, October). Surveying model based testing approaches characterization attributes. *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 324-326).

[21] Garousi, V., Mesbah, A., Betin-Can, A., & Mirshokraie, S. (2013). A systematic mapping study of web application testing. *Information and Software Technology*, *55(8)*, 1374-1396.

[22] Bernard, E., Bouquet, F., Charbonnier, A., Legeard, B., Peureux, F., Utting, M., & Torreborre, E. (2006). *Model-Based Testing from UML Models*.

[23] Svahnberg, M., Gorschek, T., Nguyen, T. T. L., & Nguyen, M. (2013). Uni-REPM: Validated and improved. *Requirements Engineering*, *18*(*1*), 85-103.

[24] Wassermann, G., Yu, D., Chander, A., Dhurjati, D., Inamura, H., & Su, Z. (2008, July). Dynamic test input generation for web applications. *Proceedings of the 2008 International Symposium on Software Testing and Analysis* (pp. 249-260). ACM.

[25] Lebeau, F., Legeard, B., Peureux, F., & Vernotte, A. (2013, March). Model-based vulnerability testing for web applications. *Proceedings of the 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 445-452).

[26] Malak, G., Sahraoui, H., Badri, L., & Badri, M. (2010). Modeling web quality using a probabilistic approach: An empirical validation. *ACM Transactions on the Web (TWEB)*, *4(3)*.

[27] Hussain, T. (2013, December). An approach to evaluate the performance of web application systems. *Proceedings of International Conference on Information Integration and Web-Based Applications & Services* (p. 692).

[28] Liu, C. H. (2006). Data flow analysis and testing of JSP-based Web applications. *Information and Software Technology*, *48(12)*, 1137-1147.

[29] Wang, X., Zhou, B., & Li, W. (2013). Model-based load testing of web applications. *Journal of the Chinese Institute of Engineers*, *36*(*1*), 74-86.

[30] Jiang, G., & Jiang, S. (2009, September). A quick testing model of Web performance based on testing flow and its application. *Proceedings of the WISA 2009 Web Information Systems and Applications Conference* (pp. 57-61).

[31] Shams, M., Krishnamurthy, D., & Far, B. (2006, November). A model-based approach for testing the performance of web applications. *Proceedings of the 3rd International Workshop on Software Quality Assurance* (pp. 54-61).

[32] Bansal, P., & Sabharwal, S. (2013, August). A model based approach to test case generation for testing the navigation behavior of dynamic web applications. *Proceedings of the 2013 Sixth International Conference on Contemporary Computing* (IC3) (pp. 213-218).

[33] Stepien, B., Peyton, L., & Xiong, P. (2012, March). Using TTCN-3 as a modeling language for web penetration testing. *Proceedings of the 2012 IEEE International Conference on In Industrial Technology (ICIT)* (pp. 674-681).

[34] Al-Herz, A., & Ahmed, M. (2011). Model-based web components testing: Prioritization using MIDS and centrality measures. *International Journal of New Computer Architectures and their Applications (IJNCAA)*, *1(4)*, 821-837.

[35] Dincturk, M. E., Jourdan, G. V., Bochmann, G. V., & Onut, I. V. (2014). A model-based approach for crawling rich internet applications. *ACM Transactions on the Web (TWEB)*, *8(3)*.

[36] Thomas, L., Xu, W., & Xu, D. (2011, July). Mutation analysis of magento for evaluating threat model-based security testing. *Proceedings of the 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW),* (pp. 184-189).

[37] Shams, M., Krishnamurthy, D., & Far, B. (2006, November). A model-based approach for testing the performance of web applications. *Proceedings of the 3rd International Workshop on Software Quality Assurance* (pp. 54-61).

[38] Sprenkle, S., Pollock, L., & Simko, L. (2011, March). A study of usage-based navigation models and generated abstract test cases for web applications. *Proceedings of the 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation* (pp. 230-239).

[39] Belli, F., Endo, A. T., Linschulte, M., & Simao, A. (2011, December). Model-based testing of web service compositions. *Proceedings of the 2011 IEEE 6th International Symposium on In Service Oriented System Engineering (SOSE)* (pp. 181-192). IEEE.

[40] Hossen, K., Groz, R., Oriat, C., & Richier, J. L. (2013, March). Automatic generation of test drivers for model inference of web applications. *Proceedings of the 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops* (pp. 441-444).

[41] Jokhio, M. S., Dobbie, G., & Sun, J. (2009, April). Towards specification based testing for semantic web services. *Proceedings of the ASWEC'09 Software Engineering Conference* (pp. 54-63).

[42] Torsel, A. M. (2013, March). A Testing tool for web applications using a domain-specific modelling language and the NuSMV model checker. *Proceedings of the 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation* (pp. 383-390).

[43] Marín, B., Vos, T., Giachetti, G., Baars, A., & Tonella, P. (2011, May). Towards testing future web

applications. *Proceedings of the 2011 Fifth International Conference on Research Challenges in Information Science* (RCIS), (pp. 1-12).

[44] Krishnamurthy, D., Shams, M., & Far, B. H. (2010). A model-based performance testing toolset for web applications. *Engineering Letters*, *18(2)*.

[45] Reza, H., Ogaard, K., & Malge, A. (2008, April). A model based testing technique to test web applications using statecharts. *Proceedings of the Fifth International Conference on Information Technology: New Generations* (pp. 183-188).

Hasan Javed was born in 1990. He is currently pursuing his master degree in software engineering from University Institute of Information Technology PMAS Arid Agriculture University Rawalpindi Pakistan. He earned his undergrad degree in computer science from UIIT, PMAS – AAUR Pakistan. His research area is requirement engineering and software testing.

Nasir Mehmood Minhas was born in 1973. He is in field of teaching for the last 16 years. He is currently working as an assistant professor of software engineering at University Institute of Information Technology PMAS-Arid Agriculture University Rawalpindi, Pakistan. He is currently pursuing his PhD from Capital University of Science and Technology, Islamabad Pakistan. His research interests are software process, software requirement engineering, and software testing with a special focus on Global Software Development Environment.

Ansar Abbas was born in 1990. He is currently pursuing his master degree in software engineering from University Institute of Information Technology PMAS Arid Agriculture University Rawalpindi Pakistan. His research area is requirement engineering**.**

Farhad Muhammad Riaz was born in 1984. He completed his BS(CS) degree from AIOU and MS degree from IIUI Islamabad, Pakistan in 2009 and 2012 respectively. He is in the field of teaching for the last 6 years. Currently he is also a visiting faculty member of University Institute of Information Technology (UIIT) PMAS-UAAR Rawalpindi. His research interest are social network, professional network and recommender systems, big data analysis techniques and Data mining techniques in requirement engineering.