A Survey of Analysis Models and Methods in Website Verification and Testing

Manar H. Alalfi, James R. Cordy, and Thomas R. Dean

School of Computing, Queens University
Kingston, Ontario, Canada
{alalfi,cordy,dean}@cs.queensu.ca

Abstract. *Models are considered an essential step in capturing system behavior and simplifying the analysis required to check or improve the quality of software. Verification and testing of websites requires effective modelling techniques that address the specific challenges of web applications (WAs). In this study we survey 21 different modelling methods used in website verification and testing. Based on our survey, a categorization, comparison and evaluation for such models and methods is provided.*

1 Introduction

Web applications (WAs) are evolving rapidly, using many new technologies, languages and programming models to increase interactivity and usability. This inherent complexity brings challenges for modelling, analysis, verification and testing. Some of these challenges are:

- WAs interact with many components that run on diverse hardware and software platforms. The integration of such components is extremely loose and dynamically coupled, which provides powerful abstraction capabilities to the developers, but makes analysis for testing and verification extremely difficult.
- WAs are heavily dynamic, due to dynamically generated components, dynamic interaction among clients and servers, and continual change in web technologies.
- WAs may have several entry points, and users can engage in complicated interactions that the WA cannot prevent. WAs are often interfaced to database systems and provide the same data to different users. In these cases, access control mechanisms become an important requirement for safe and secure access to WA resources, and the process of implementing and applying such rules is considered a great challenge.
- Some information in WAs is transmitted using hidden fields and special channels, due to the stateless behavior of the HTTP protocol. It's a challenge to provide a precise analysis for WAs that takes this information into account.

Most of the early literature concentrates on the process of modelling the design of WAs, using forward engineering-based methodologies designed to simplify

the process of building a highly interactive WAs. Other research uses reverse engineering to extract models from existing WAs in order to support their maintenance and evolution. This survey studies a range of different analysis models that are currently applied in the field of verification and testing of WAs. Design modelling methodologies are outside the scope of our study.

While reviewing different methods, we found that some methods focus on modelling the navigational aspects of WAs. Others concentrate on solving problems arising from user interaction with the browser in a way that affects the underlying business process. Still others are interested in dealing with static and dynamic behavior. In this paper, we attempt to categorize these methods according to the level of WA modelling - navigation, behavior, and content. In each category, methods are sorted according to the kind of notation employed. A comparison and evaluation of 21 different methods is described. The rest of this paper is organized as follows: Section 2 lists the desirable properties of WA modelling. Section 3 describes the set of comparison and categorization criteria used in our study, and gives a comparative analysis of 21 different modelling methods. Conclusions and open problems are discussed in Section 4.

2 Desirable Properties for Website Modelling

We can think of WAs from three orthogonal perspectives: web navigation, web content and web behavior. Desirable properties of WAs can fall within these three dimensions and can be classified into:

- *Static navigation properties.* Most of the early literature on web analysis and modelling concentrates on dealing with static links, treating WAs as hypermedia applications. It addresses checking of properties such as broken links, reachability, consistency of frame structure, and other features related to estimating the cost of navigation, such as longest path analysis.
- *Dynamic navigation properties.* This kind of analysis focuses on aspects that make the navigation dynamic, where the same link may lead to different pages depending on given inputs. The inputs could be user inputs transferred via forms, or system inputs depending on a state in the server such as date, time, session information, access control information or hidden fields.
- *Interaction navigation properties.* This kind of analysis focuses on properties that are related to user navigation that happens outside the control of the WA, such as user interaction with the browser. This includes features such as the back button, the forward button, and URL rewriting.
- *Static content properties.* Consistency of the web page content with respect to syntax and semantics.
- *Dynamic content properties.* This analysis requires the ability to check the syntax and semantics of dynamically generated content that results from the execution of scripts by the server. Some technologies are able to generate new connections, which may be to a remote site. In addition, new web components could be generated at run time, and these must also be analyzed.

- *Security properties.* This issue is related to access control mechanisms that can be employed on the web content or web links. It can also involve the backend when the database contains data reserved to specific users. These properties are also tied to session control mechanisms.
- *Instruction processing properties.* This includes client- and server-side execution. Client-side execution is any process that changes the state of the application without communication with the web server. Server-side execution refers to all instructions processed on a web server in response to a client's request. A modeling method should to be able to model these features and recognize whether execution is done on the server or on the client.

3 Comparison and Categorization Criteria

In our study we reviewed 21 different modelling methods that are applied in the field of testing and verification of WAs. Following is a brief description of the main comparison criteria used in our review:

1. *Feature Type.* We note the WA features that are being captured by the proposed models, and the properties that the modeling methods are capable of checking. These features are categorized into:
   - *Static Features.* This includes static properties of WAs, and is mainly concerned with links that connect an HTML page to other HTML pages. When the user clicks on a static button or link, a request is sent to the server to fetch a page. The server responds by retrieving the required page from its storage and sending it back to the client. Properties to be checked in this category relate to static navigation and static content.
   - *Dynamic Features.* These features include both dynamic links and dynamic content. Dynamic links describe the connection between HTML pages and server code that must to be executed to generate the required information, build it into an HTML page, and return it to the client. The processing done by the server may depend on user or system inputs. User inputs are usually sent by filling a form or by hidden fields in the HTTP request. System inputs depend on the server state, such as server time, or on interaction with other resources such as database servers and web objects. The output could be constructed as new content, or as a link to a new HTML page. Properties in this category are those related to dynamic navigation properties, dynamic content properties, security properties, and instruction processing properties.
   - *Interaction Features.* This includes properties related to user interaction with the browser. The browser's influence on the navigation behavior of the WAs should be taken into consideration while modelling or analyzing WAs, as the web browser provides the interface to the WAs, and can change the navigation behavior while a user browses a WA.
2. *Notation.* Modeling methods use different notations; some of them are formal, while others are either semi- or informal. The main notation used by each method is noted.

3. *Level of Modelling.* WA modelling can be viewed from different perspectives. We compare the modelling methods here according to three basic levels: content, structure (navigation), and behavior. These three levels in turn could have a static or a dynamic flavor.
4. *Application of the Model.* In our study we focus on methods that are concerned with modelling WAs for the purpose of testing or verification; this also could include design verification.
5. Is Source Code required?. Modeling methods may require doing a white-box or a black-box analysis. This determines whether or not the existence of the source code is required for the analysis. The kind of analysis for each reviewed method is specified.
6. *Model Optimization.* Complex systems in general may have a state explosion problem or generate a large complex model. Such models require some kind of optimization. In WAs, this problem becomes a major challenge to the success of any method that attempts to model a scalable web system.
7. *Tool Support.* We list if the method being described is supported either with a proposed tool, or with a pre-existing tool.

Our study resulted in two different views of the methods we surveyed, a general categorization by modelling level, and a detailed comparison by property coverage. Table 1 summarizes the first one, where the 21 methods are categorized according to the level of WA modelling. A second comparison between methods was done based on the more specific details of methods in the same category in particular, and other methods in other categories in general. This second comparison is based on a combination of feature type and the level of WA modeling, using the comparison criteria outlined in Section 3 as desirable properties for website modeling. The second comparison could not fit in this short paper and can be found in our technical report [1].

4 Conclusions and Open Problems

Little work has been done to compare different modelling methods in the field of web development. To the best of our knowledge this is the first study which focuses on a comprehensive review and comparative study of modelling methods that are currently applied in website verification and testing. Previous work has focussed more on the development process in general, and on the design phase in particular. Our analysis is based on two sets of criteria, with results summed up in two concise tables. We found that this field is still in its infancy. While much has been done, up until now there is no complete modelling method that is able to capture all of the desirable properties of WAs at all modelling levels. An integration of different modelling methods may be required in order to generate a new complete model that could be verified using model checking. There is also a need for work on security modelling techniques that are able to deal with the complex, distributed structure of WAs, taking into account concurrent access to web servers and other shared resources.

Method Name	Feature type	Notation	Level	Application	Source code required	Model optimization	Tool support
GFKF03[14]	Interaction	Abstract model, use lambda calculus	Interaction Behavior	WA interaction with the browser	No	No	Prototype
LK04 [3]	Interaction	WebCFG	Interaction Behavior	Verification	Yes	Yes	Implement a model checker
CZ04 [20]	Interaction +Static	Labeled transition	Interaction + static (Navigations)	Testing and verification	No	Yes	None
BA05 [2]	Interaction	UML(WS structure) OCL (behavior of the model)	Interaction Behavior	Verification for user interaction(Amazon + Orbitz bug)	No	Yes	UML2Alloy
ABF05 [16]	Static	Partial rewriting	Content	WS verification Tool(GVerdi)	Yes	No	GVerdi
Con99 [9]	Static	Extended UML	Structure (Navigation)	Analysis	No	No	Rational Rose Tools
BMT04 [22]	Static + dynamic	UML-meta Model + UML state diagram	Structure (Navigation)	Analysis & Testing	Yes	No	WebUML
RT00 [7]	Static	Directed graph	Structure (Navigation)	Analysis + can be use for verification & testing	No	No	ReWeb
dA01 [10] and dAHM01[26]	Static	Directed graph With Webnodes	Structure (Navigation)	Verification	No	No	MCWeb
SDMP02 [5]	Static + dynamic	Web graph	Structure (Navigation)	WA design Verification	No	No	AnWeb
SDM+05[6] and CMRT06 [23]	Static + dynamic	(WAG)WA graph + extension to Kripke structure	Structure (Navigation)	WA design Verification	No	No	WAVer + SMV tools
WP03 [25]	Static + dynamic	Extended StateCharts	Structure (Navigation)	Design Verification	Yes	Yes	SWCEditor
HH06 [19] FARNav	Static + dynamic	StateCharts	Adaptive Navigation	design and implementation Verification + testing	No	Yes	Existing SVM model-checking tools
SM03 [12]	Static + dynamic	SDL	Structure (Navigation)	Testing and verification	Yes	No	Existing SDL Support tool
KLH00 [21] WTM	Static + dynamic	Control flow graph, data flow graph, and finite state machines OSD(object state diagram)	Static and dynamic Behavior, Dynamic Navigation	Testing	Yes	No	None
BFG02 [11] Veriweb	static + dynamic	Directed graph	Navigation + Behavior	WS testing	Yes	Yes	VeriSoft + web Navigator + ChoiceFinder + SmartProfiles
HPS04 [8]	Static+ dynamic	System of communicating automata	Navigation + Behavior	WA Verification	No	Yes	Fame Work with GUI + network monitoring tool + analysis tool
AOA05 [17] FSMWeb	static + dynamic	hierarchies of Finite State Machines (FSM)	Navigation + Behavior	System level testing	No	Yes	Prototype
WO02 [18]	Interaction + static + dynamic	Regular expression	Interaction + dynamic Behavior	Can be used for testing + implementation + impact analysis	Yes	No	None
TR04 [13] And TR02 [15]	Static + dynamic	(model navigation layer) + CFG (client & server code)	Structure (Navigation)+ Behavior	Testing	Yes	No	ReWeb + TestWeb
KZ06 [24]	Static + dynamic	Extended UML (UWE)	Structure (Navigation) + Behavior	Design Validation and Verification	No	No	ArgoUWE + Spin or UPPAAL

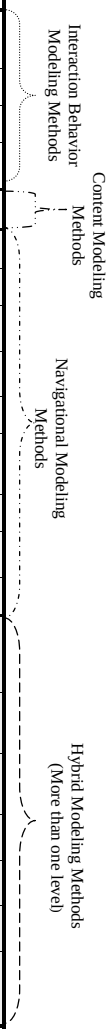

Table 1. Summary of Methods Categorized by Modelling Level

References

1. Alalfi, M., Cordy, J.R., Dean, T.R.: A survey of analysis models and methods in website verification and testing. Technical report, School of Computing, Queen's University (2007)
2. Bordbar, B., Anastasakis, K.: MDA and analysis of web applications. In: TEAA. Volume 3888 of LNCS., Springer (2005) 44–55

3. Licata, D.R., Krishnamurthi, S.: Verifying interactive web programs. In: ICASE, IEEE Computer Society (2004) 164–173
4. Sciascio, E.D., Donini, F.M., Mongiello, M., Piscitelli, G.: Anweb: a sytem for automatic support to web application verification. In: ICSEKE. (July 14-19 2002) 609–616
5. Sciascio, E.D., Donini, F.M., Mongiello, M., Totaro, R., Castelluccia, D.: Design verification of web applications using symbolic model checking. In: ICWE. Volume 3579 of LNCS., Springer (July 27-29 2005) 69–74
6. Ricca, F., Tonella, P.: Web site analysis: Structure and evolution. In: ICSM. (2000) 76–86
7. Haydar, M., Petrenko, A., A.Sahraoui, H.: Formal verification of web applications modeled by communicating automata. In: FTNDS. Volume 3235 of LNCS., Springer (September 27-30 2004) 115–132
8. Conallen, J.: Modeling web application architectures with UML. Communications of the ACM **42** (1999) 63–71
9. de Alfaro, L.: Model checking the world wide web. In Berry, G., Comon, H., Finkel, A., eds.: CAV. Volume 2102 of LNCS., Springer (July 18-22 2001) 337–349
10. Benedikt, M., Freire, J., Godefroid, P.: Veriweb: Automatically testing dynamic web sites. In: IWWWC. (May 2002)
11. Syriani, J.A., Mansour, N.: Modeling web systems using SDL. In Yazici, A., Sener, C., eds.: CIS. Volume 2869 of LNCS., Springer (November 3-5 2003) 1019–1026
12. Tonella, P., Ricca, F.: A 2-layer model for the white-box testing of web applications. In: IWWE, IEEE Computer Society (2004) 11–19
13. Graunke, P.T., Findler, R.B., Krishnamurthi, S., Felleisen, M.: Modeling web interactions. In Degano, P., ed.: PLS. Volume 2618 of LNCS., Springer (April 7-11 2003) 238–252
14. Tonella, P., Ricca, F.: Dynamic model extraction and statistical analysis of web applications. In: IWWSE, IEEE Computer Society (2002) 43–52
15. Alpuente, M., Ballis, D., Falaschi, M.: A rewriting-based framework for web sites verification. ENTCS **124** (2005) 41–61
16. Andrews, A.A., Offutt, J., Alexander, R.T.: Testing web applications by modeling with fsms. Software and System Modeling **4** (2005) 326–345
17. Wu, Y., Outt, J.: Modeling and testing web-based applications. Technical report, George Mason University (2002)
18. Han, M., Hofmeister, C.: Modeling and verification of adaptive navigation in web applications. In: ICWE. (2006) 329–336
19. Chen, J., Zhao, X.: Formal models for web navigations with session control and browser cache. In: ICFEM. (2004) 46–60
20. Kung, D.C., Liu, C.H., Hsia, P.: An object-oriented web test model for testing web applications. In: COMPSAC. (2000) 537–542
21. Bellettini, C., Marchetto, A., Trentini, A.: Webuml: reverse engineering of web applications. In: SAC. (2004) 1662–1669
22. Castelluccia, D., Mongiello, M., Ruta, M., Totaro, R.: Waver: A model checking-based tool to verify web application design. ENTCS **157** (2006) 61–76
23. Knapp, A., Zhang, G.: Model transformations for integrating and validating web application models. In: Modellierung. (2006) 115–128
24. Winckler, M., Palanque, P.A.: Statewebcharts: A formal description technique dedicated to navigation modelling of web applications. In: DSV-IS. (2003) 61–76
25. de Alfaro, L., Henzinger, T.A., Mang, F.Y.: MCWEB: A model-checking tool for web site debugging. In: WWW. (2001) 86–87