

A Framework for the Simulation of Web Applications

Pedro Peixoto, K.K. Fung, and David Lowe

University of Technology, Sydney
PO Box 123 Broadway NSW 2007, Australia
{ppeixoto, kkf, david.lowe}@eng.uts.edu.au

Abstract. In recent years numerous Web application modeling languages have been developed and others improved. There has, however, been little research on how these languages may be simulated. Simulation of models constructed using design languages allows early evaluation and prevents unnecessary Web code development and implementation. It can therefore significantly reduce the design cycle time and cost. This paper introduces a Web application simulation model framework that was designed to be compatible with existing modeling languages. This was accomplished by specifically identifying the objectives of a simulation language and contrasting this with those of design models. The simulation model supports analysis of simulations from four key Web application perspectives (and hence the model is constructed around these perspectives) namely: presentation, navigation, functionality and content. We argue that with this approach substantial inferences about the quality of the design can be drawn from simulation of the Web application model.

1 Introduction

Web applications projects have reached a level of complexity that demand modeling techniques to tackle their design intricacies. Several modeling languages such as OOHDML [1], UML [2] and WebML [3] have been proposed as aids to Web application design, development, and implementation. Although these modeling languages provide an overview of the system, there is still the need for some coding to get a first glimpse of the interaction amongst the systems components.

A desirable approach would be to simulate the design without any implementation whatsoever. Simulation has been widely and successfully used in the hardware design field. In fact, Hardware Description Languages (HDLs) such as VHDL (Very High-Speed Integrated Circuit Hardware Description Language) [4,5], are considered an indispensable modeling tool. Simulation of VHDL models is performed without a physical implementation thereby reducing both cost and design time. But if modeling of Web applications has been thoroughly addressed, simulation of the models themselves is little researched. So far, simulation has only been used to evaluate Web systems performance [6,7,8].

We feel that Web modeling languages have reached a maturity level that permits the present attempt of defining a framework for Web application simulation. We aim to build the foundations of a simulation model based on a Web Description Language (WDL), developed specifically to assist on the simulation process, permitting the assessment of Web application designs regardless of the modeling language chosen.

The next section describes the WDL Simulation model, and in section 3 we outline the objectives, input stimuli, results, and analysis of the simulation. Finally Section 4 makes a brief summary and draws conclusions.

2 The WDL Simulation Model

Simulation consists on the observation of a system's response over time when a known set of stimuli is present at its inputs [9, 10]. In this paper, simulation is regarded as the evaluation of the extent to which the Web application design (represented using some modeling notation) supports interaction with the user, and aims to monitor and draw inferences of what occurs internally.

To simulate a design a description of its components has to be available. To deal with the specificities of Web applications a description language (WDL) was developed which captures both the structural and dynamics of each component, similar to VHDL, where for each component a definition of the interface and functionalities is explicitly declared. The main purpose of WDL is not to model the Web application but to enable the simulation.

For the designer the most important aspects of the application that need to be assessed are its graphical interface, the navigation network structure, the scripts interface and workflow, and data exchanged among the components. In fact, existing models such as the Dexter Hypertext Reference Model [11], HDM [12], and WebML [13, 14], tend to decompose systems along these lines [15] emphasizing the use of different layers to tackle distinct aspects of Web applications. For this reason, our simulation model focuses on four different perspectives corresponding to four different layers, namely: presentation, navigation, functional, and content, as Fig.1 shows. Also a *User Interaction Model* was considered to reflect the possible actions that may be undertaken by a user, restricting the simulation stimuli to a well known set. Each layer, being an orthogonal aspect of the design, enables analysis from a distinct point of view. Simulation on these four layers requires the ability of the modeling language to describe in detail components structure, workflow, and behavior, and, therefore, a set of requirements must be met for a high degree of accuracy simulation. If, however, the modeling language used does not encompass all the requirements, it does not automatically preclude simulation but will imply a lower simulation detail. The set of requirements the modeling languages should meet are enumerated and described in [16].

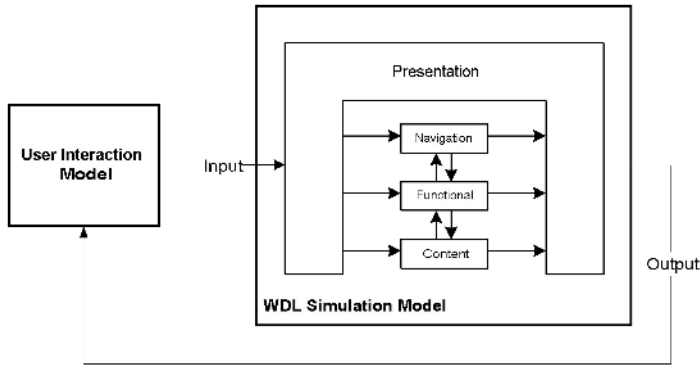


Fig. 1. The WDL Simulation Model

3 Layer Definition

The Presentation Layer – Simulation relies on its stimuli to evaluate the system under analysis. Since the system is a Web application, its change in state is essentially due to user inputs through the graphical interface – the Presentation layer. Our simulation model reflects this by wrapping the Presentation layer around the remaining preventing the user to directly interact with them.

This layer deals with the description of the page's look and feel, namely the user data input components, information components, and hyperlinks. However, it does not attempt to interpret the user actions; its semantics are dealt with on the remaining layers. As a consequence, user actions that potentially alter the state of the system are initiated on the Presentation layer but acquire significance and meaning on the remaining layers. Stimuli of this layer come mainly from the user interaction through interface components – concrete and clearly identified HTML elements with well-defined structure and behavior.

Results from the simulation are essentially the rendering of the model's graphical user interface description by displaying the active pages, and the interaction with the remaining layers – the navigational consequences of the user browsing, the functional script outcomes, and the data included and displayed. Assessment of the user interface is this layer's simulation main purpose, namely the graphical realization of hyperlinks and data input elements. Furthermore, the interface may pose some interaction limitations, namely on a data input level, which must be properly assessed by the designer.

The Navigation Layer – On this layer, pages are considered as macro blocks or abstractions of the pages displayed by the Presentation layer – they are simply containers of navigation components. Simulation of the navigation path through the Web pages set due to the user interaction with the interface via the Presentation layer is performed. It is on this layer that some of the signals that trigger script processing have their origin. Simulation performed on this layer is driven by user interaction with the displayed components of the Web application. The user interacts with the pages on the Presentation layer

and the possible actions performed belong to a well-defined finite set - following hyperlinks, mouse actions or buttons selection. Stimuli that drive the simulation are a generalization of the inputs in stereotyped categories such as clicking on links or the action of submitting data to the server, and script actions that imply navigation changes. However, the specific data input contents from the user are not considered in this layer - that will be dealt with on the Functional layer. Throughout the simulation the designer may observe the different navigation paths the user has available to chose from due to changes in the active pages. Different windows are opened in response to diverging navigation trails and, consequently, several new links become available. The designer may trace which user actions led to the opening, activation or closing of a specific window, and the correspondingly set of navigation elements available. One of the goals of simulating this layer is the verification of the navigation flow and assertion of the navigation path due to user and script actions.

The Functional Layer – It is on this layer that the application acquires its dynamic aspect and where the client and server-side scripts are managed. Furthermore, this layer interacts with the Presentation layer for dynamic page construction, and with the Navigation layer for internal changes in the active pages. Data input in Web forms as well as any state information are the principal sources of information obtained from users. Scripts process this data from the Presentation layer and stored data from the Content layer, and are triggered by other scripts or by user action via the Navigation layer. Resulting from the simulation, data flow exchange between the script components and components on the remaining layers are observed. The data used by a specific script and its results are also displayed. What scripts are running at a particular point in time and their contribution to the overall Web application dynamics are another of the simulation results. Furthermore, the assessment of the process workflow assures that the Web application dynamics is in agreement with the design requirements.

The Content Layer – All data stored on the system is managed by this layer which is a main information resource of the Functional and Presentation layers. It manages all the data used by the Web application – whether from databases, files, or more volatile data such as Web session variables. The stimuli of the content layer simulation are the data write and read command set such as a database write operation or reading a client's cookie ID.

The information flow amongst the different layers and all the actions that involve data manipulation are displayed, enabling assessment of the Web application content management.

4 Conclusion and Future Work

A framework for the simulation of Web application design models is presented. By using a Web Description Language that maps the design into an enumeration of the structural and behavioral characteristics, the WDL Simulation model is able to evaluate the design from four orthogonal perspectives, namely: Presentation, Navigation, Functional, and Content, which relate to the four important

analysis a designer has to conduct to verify the Web application and validate its requirements. By simulating the application directly from the model, and without any code implementation, substantial project time reduction is achieved. A prototype is being developed to simulate designs using several modeling languages such as UML or WebML. Furthermore, the next logical step is to use WDL to automatically synthesize the code. This, as it happens in the hardware design field, would provide designers with powerful tools for the development of complex Web applications. A fuller description of the layers and especially the interfaces between them is available in an extended version of this paper published as an Technical Report [16].

References

1. Schwabe, D., Rossi, G.: The object-oriented hypermedia design model. *Communications of the ACM* **38** (1995) 45–46
2. Conallen, J.: *Building Web Applications with UML*. Addison Wesley Object Technology Series. Addison-Wesley (1999)
3. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: *Designing Data-Intensive Web Applications*. 1st edn. Morgan Kaufmann Pub. (2003)
4. Ashenden, P.: *The Designer's Guide to VHDL*. 1st edn. Morgan Kaufmann Publishers (2002)
5. Jerraya, A., Ding, H., Kission, P., Rahmouni, M.: *Behavioral Synthesis And Component Reuse With VHDL*. 1st edn. Kluwer Academic Publishers (1997)
6. Cohen, E., Krishnamurthy, B., Rexford, J.: Improving end-to-end performance of the web using server volumes and proxy filters. In: *SIGCOMM*. (1998) 241–253
7. Iyengar, A., MacNair, E., Nguyen, T.: An analysis of web server performance. *Global Telecommunications Conference, GLOBECOM '97* **3** (2000)
8. Hadharan, R., Ehrlich, W.K., Cura, D., Reeser, P.K.: End to end performance modeling of web server architectures. *SIGMETRICS Perform. Eval. Rev.* **28** (2000) 57–63
9. Schriber, T.J., Brunner, D.T.: Inside simulation software: inside discrete-event simulation software: how it works and why it matters. In: *Proc. of the 32nd conference on Winter simulation, Soc. for Comp. Sim. International* (2000) 90–100
10. Banks, J.: Simulation fundamentals: simulation fundamentals. In: *Proc. of the 32nd conference on Winter simulation, Soc. for Comp. Sim. Int'l* (2000) 9–16
11. Halasz, F., Schwartz, M.: The Dexter hypertext reference model. *Commun. ACM* **37** (1994) 30–39
12. Garzotto, F., Paolini, P., Schwabe, D.: HDM - a model-based approach to hypertext application design. *ACM Trans. Inf. Syst.* **11** (1993) 1–26
13. Ceri, S., Fraternali, P., Bongio, A.: Web modeling language (WebML): a modeling language for designing web sites. In: *Proc. of WWW9 Conf., Amsterdam* (2000)
14. Bongio, A., Ceri, S., Fraternali, P., Maurino, A.: Modeling data entry and operations in WebML. *Lecture Notes in Computer Science* (**1997**)
15. Fraternali, P.: Tools and approaches for developing data-intensive web applications: a survey. *ACM Comput. Surv.* **31** (1999) 227–263
16. Peixoto, P., Fung, K., Lowe, D.: *Simulating Web Applications - The Definition of a Framework*. Technical report, University of Technology, Sydney, Faculty of Engineering, Sydney, Australia (2004)
<http://www.eng.uts.edu.au/Research/Reports/UTS-Eng-TR-04-001.pdf>.