

# Formal Verification of Web Navigation by Symbolic Model Checking

Hisashi MIYAZAKI, Tomoyuki YOKOGAWA, Kouichi SEKO,  
Yoichiro SATO, and Michiyoshi HAYASE

Faculty of Computer Science and System Engineering, Okayama Prefectural University, Japan

111 Kuboki, Soja, Okayama 719-1197, Japan

Tel: +81-866-94-2405, Fax: +81-866-94-2199

E-mail: miyazaki@radish3.cse.oka-pu.ac.jp

Keywords: symbolic model checking, UML, statechart, web navigation

## Abstract:

Previously, it is general that users navigate from a web page to the other by clicking on a hyperlink. Recently web pages become dynamic with a variety of scripts and embedded client side programs. Such pages with dynamic navigation have a more complicated structure, so it is difficult to model and analyze.

In this paper, we present a method to verify systems which have dynamic web navigation. For this purpose, we model the navigation using a UML statechart diagram and translate the model to a boolean expression. Thus it becomes possible to verify the systems formally using symbolic model checking.

To demonstrate the proposed method, we apply the method to a system with dynamic web navigation and model-check the system using symbolic model checking tool called NuSMV [9]. As a result, we verified reachability to all pages of the system.

## 1. Introduction

Recently, it becomes difficult to develop and maintain web sites because of the growing size and complexity. Several studies have been made on modeling structure of web sites and supporting development and maintenance of them. The modeling of navigation in web sites can help in the analysis for the structure of web sites.

In general, users navigate back and forth through the web pages using a hyperlink. However, the progress of web technology has given us other way to control moving between web pages using various client side scripts and programs such as Javascript, JAVA, Adobe Flash, and ActiveX. Such complex and dynamic web pages are hard to analyze.

Leung [8] et al. proposed the method to model the structure of dynamic web navigation by extending description of UML statechart diagrams [5]. This made it possible to model dynamic aspects of navigation using scripts and programs. However, it is still impossible to verify whether the structure satisfies requirements of designers or not.

In this paper, we propose the method to verify the structure of dynamic web navigation using symbolic model checking [3]. We first provide the model of dy-

namic web navigation using a UML statechart diagram and show the translation method from the model to a boolean expression. Thus it becomes possible to verify the structure of navigation using symbolic model checking.

## 2. Web Navigation

Navigation on web pages are classified into following four types; *intra-page navigation*, *inter-page navigation*, *frame-based navigation*, and *script-based navigation*.

**Intra-page navigation** In this type of web navigation, only one web page is involved in the navigation. A hyperlink on a web page is used to link to a different position in the same page. The same effect can be obtained using the page scrolling.

**Inter-page navigation** The inter-page navigation is the most common form of web navigation. In this type of navigation, activating a hyperlink on a web page induces to jump into a target web page specified by the hyperlink.

**Frame-based navigation** The frame structure makes concurrent viewing of web pages possible. In the web page with frames, a browser window is divided into some frames and each frame contains a web page. The contained web page can be a web page with frames. Navigation within each frame is carried out independently. When navigation to the outside of framed web page is executed, the target web page replaces the whole framed page.

**Script-based navigation** In script-based navigation, a dynamic behaviour is enabled by client side scripts. In web pages with scripts, hyperlinks can be enabled or disabled, activated automatically, or change their targets dynamically. These behaviours are supported in various client side scripts, such as JavaScript and Java. These scripts or programs have their own states and operate interactively. Moreover scripts can affect the contents and navigations of the hosting web pages. If the hosting web page is replaced by an other web page or closed,

the scripts and programs will be terminated immediately.

### 3. Modeling Web Navigation

As the dynamic web navigation has a very complex structure, it is necessary to model the behaviour in detail. In this paper, we use the model of dynamic web navigation using UML statechart diagrams [5] based on the model proposed by Leung [8]. This model deals with three types of navigation except intra-page navigation from the four types stated above.

**Statechart diagram** A statechart diagram is composed of states and transitions combining states each other. A transition is labeled by event, guard, and action as a form of “*event[guard]/action.*” An event is a trigger that causes state transition. An action is an effect produced by a state transition. A guard is a condition that must be true before when a state transition occurs. A state can contain AND or OR substates: if the state is activated, all of AND substates are also activated and only one of OR substates is activated. Such a state is called an orthogonal state. An orthogonal state can contain orthogonal states again.

**Inter-page navigation** Each hyperlink from a web page to an other one is represented as a transition between states. The hyperlink activation is represented by an event triggering the transition. We define an event *jp(target.html)* which represents an activation of hyperlink which links the web page **target.html**.

Figure 1 shows a navigation between web pages: **index.html** and **target.html**.

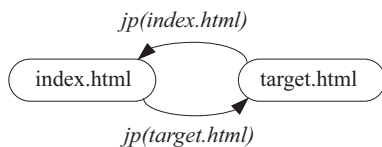


Figure 1. Model of inter-page navigation

**Frame-based navigation** A web page with frames is modeled as orthogonal AND state. Each AND substate of the orthogonal state represents each frame of the web page. Web pages within each frame is represented as OR substates in the AND substate describing the frame. Navigation within each frame is represented as a transition between its OR substates. Navigation to the outside of frame is represented as a transition into the state outside of the orthogonal state. The default web page loaded

in the frame is indicated as the initial state of the substate.

Figure 2 shows an example of navigation on the pages with frames. Optional state names, such as **frameA**, **frameB** and **frameC** in the Figure 2 can be added to each frame.

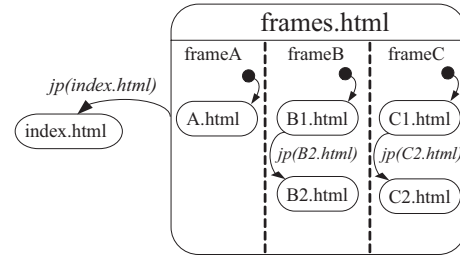


Figure 2. Model of frame-based navigation

**Script-based navigation** The dynamic behaviour by a script or a program in a web page is modeled as a statechart diagram. States in the statechart diagram represent the state of the script or program. Navigations in web pages with scripts or programs are controlled by guard conditions.

Figure 3 shows an example of navigation on the pages with scripts. The navigation which is represented by the transition labeled by *jp(menu.html)[JScript.in(agree)]/animation* is enabled only when the state of *JScript* is in *agree*.

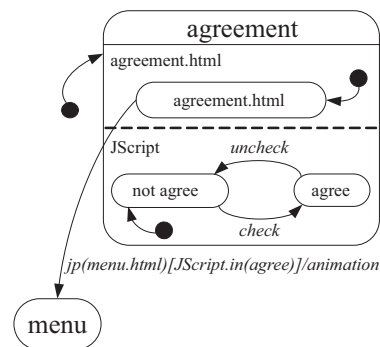


Figure 3. Model of script-based navigation

**User operation** Note that in order to model a structure of web navigation, it is necessary to model operations handled by users. Operations by user, such as clicking mouse, manipulating cursor, checking or unchecking a checkbox, and so on are modeled as a statechart diagram. Because user is supposed to operate any action at all times, the statechart modeling user has only one state.

Figure 4 shows an example of user operations. In this model, user may act these four types of operation: checking or unchecking a checkbox, clicking a mouse, or moving a mouse to some position.

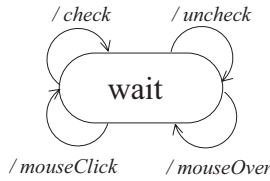


Figure 4. Model of user operation

#### 4. Model Checking of Web Navigation

In order to carry out symbolic model checking, a transition in a system needs to be represented as a boolean formula which holds iff the transition fires. We adopt the approach presented in [1] for translating statechart models into a boolean formula. We thus show translating transitions in a statechart diagram which describe navigations to a boolean formula representing them.

For example, the transition from the state *agreement.html* to the state *menu* in Figure 3 is represented as a boolean formula as follows. The transition is labeled by  $jp(menu.html)[JScript.in(agree)]/animation$ . This transition fires only when the system is in the state *agreement*, the substate of *agreement* is *agreement.html*, the event  $jp\_menu$  occurs, and the guard  $JScript.in(agree)$  evaluates to true, that is, the substate of  $JScript$  is *agree*. If the transition has occurred, the system will enter the state *menu*, the event  $jp\_menu$  is discarded, and the action *animation* will be executed. Therefore this transition is expressed as a formula  $S_{system} = agreement \wedge S_{agreement.html} = agreement.html \wedge jp(menu.html) \wedge S_{JScript} = agree \wedge S'_{system} = menu \wedge animation' \wedge \neg jp\_menu'$ . The primed variable (e.g.  $x'$ ) represents the value of it in the next state. Events and actions are represented by boolean variables.

This formula holds iff this transition is enabled and the value of each variable in the next state is assigned as designed by this transition. The formula thus represents this transition fires.

#### 5. Experimental Result

We apply the proposed method to an example modeled as the statechart diagram in Figure 5. Figure 5(a) models user operations such as mouse click, mouse over, and checking or unchecking a checkbox. Figure 5(b) models the navigation generated by the user operations.

The transition relation described by the statechart diagram is translated a boolean formula and the reachability of web pages is checked by symbolic model checking tool NuSMV[9].

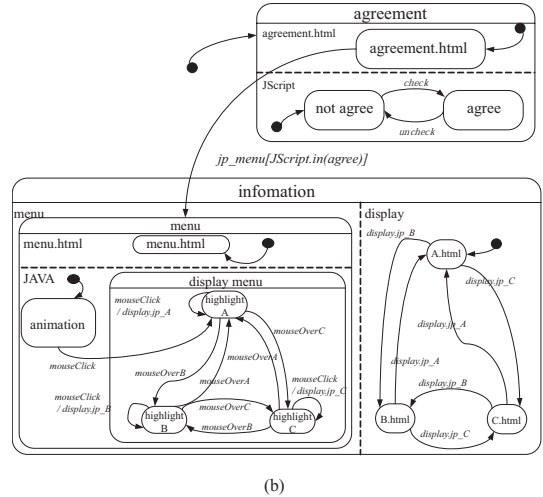
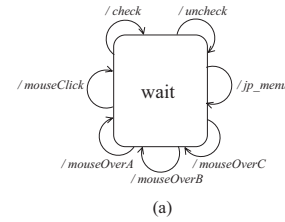


Figure 5. User operation and web navigation

The verification result is shown as Figure 6. Here  $EF$  is an operator of CTL[2]. The formula  $EFp$  holds in a state  $s$  if  $p$  holds some states along some sequences of states starting from  $s$ . As shown in the Figure, the reachability properties of web pages **menu.html**, **A.html**, **B.html**, and **C.html** are verified.

```
-- specification EF S_menu_html = menu_html is true
-- specification EF S_display = A_html is true
-- specification EF S_display = B_html is true
-- specification EF S_display = C_html is true
system diameter: 21
reachable states: 5178 (2^12.3382) out of 73728 (2^16.1699)
```

Figure 6. Verification result

#### 6. Conclusion

In this paper, we show the model for dynamic web navigation using UML statechart diagram. We also show the translation from the model into a boolean formula. We apply the method to an example describing dynamic web navigation and verify the reachability property of web pages.

## Acknowledgments

This work was in part supported by Grant-in-Aid from the Ministry of Education, Culture, Sports, Science and Technology of Japan (No. 19700030).

## References

- [1] W. Chan, R. J. Anderson, P. Beame, S. Burns, F. Modugno, D. Notkin, and J. D. Reese. Model checking large software specifications. *IEEE Transactions on Software Engineering*, 24(7):498–519, July 1998.
- [2] E. Clarke, E. Emerson, and A. Sistla. Automaric verification of finite-state concurrent systems using temporal-logic specifications. *ACM Trans. Programming Languages and Systems*, 8(2):244–263, 1986.
- [3] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.
- [4] M. C. F. de Oliveira, M. A. S. Turine, and P. C. Masiero. A statechart-based model for hypermedia applications. *ACM Transaction on Information Systems*, 19(1):28–52, January 2001.
- [5] O. M. Group. *Unified Modeling Language*. Object Management Group, 2001. <http://www.uml.org>.
- [6] D. Harel. Statecharts: a visual formalism for computer system. *Science of Computer Programming*, 8(3):231–274, 1987.
- [7] D. Harel, A. Pnueli, J. P. Schmidt, and R. Sherman. On the formal semantics of statecharts. In *Proceedings of the 2nd IEEE Symposium on Logic in Computer Science*, pages 54–64, June 1987.
- [8] K. R. P. H. Leung, L. C. K. Huia, S. Yiu, and R. W. M. Tang. Modeling web navigation by statechart. In *Proceedings of the 24th International Computer Software and Applications Conference (COMPSAC '00)*, pages 41–47, October 2000.
- [9] *NuSMV*. <http://nusmv.irst.itc.it/>.