

検証およびテスト生成の自動化を指向した Web ナビゲーションのモデル化

横川 智教^{†1} 佐藤 洋一郎^{†1} 有本 和民^{†1}

本稿では、画面遷移図を用いて設計された Web アプリケーションにおけるナビゲーション構造を UM のステートマシン図によりモデル化する手法を提案する。

Model of Web Navigation for Automatic Verification and Test Generation

TOMOYUKI YOKOGAWA,^{†1} YOICHIRO SATO^{†1}
and KAZUTAMI ARIMOTO^{†1}

We proposed a modeling method of dynamic web navigation using UM state machine diagrams. We also showed formal definition of navigation structure which is described using a page flow diagram and translation framework from the structure into our model.

1. はじめに

本稿では、Web アプリケーションの設計において重要な要素の 1 つである Web ナビゲーション構造を、ステートマシン図によってモデル化するための手法を提案する。本稿では画面遷移図で記述される情報の構造を形式的に定義し、そこから提案するステートマシン図によるモデルを生成するための手続きを示している。

本手法では、Web ナビゲーションの (1) ページ移動、(2) 認証管理、そして (3) ユーザ操作の要素を全てステートマシン図によってモデル化し、メッセージによって相互通信することにより Web ナビゲーション動作を表現する。これにより、ステートマシン図に対するモデル検査³⁾ による検証²⁾ やテストケースの自動生成⁴⁾ のように、既存の様々なモデルベースの技術を応用して検証やテスト生成を行うことが可能となる。

2. Web ナビゲーション

2.1 概要

本稿においてモデル化の対象とする動的ナビゲーションでは、ハイパーリンクなどの静的なナビゲーションと異なり、同じユーザ操作に対して、それ以外の要因に従って異なるページ移動が行われる。ナビゲーションに影響を与える要因には様々なものが考えられるが、本稿では、ユーザの認証状況および Web ページ上のフォーム部品に入力された値に依存してナビゲーション

が行われるものとする。

2.2 構造

Web アプリケーションのナビゲーション構造は、画面遷移図によって記述される。本稿では、画面遷移図によって与えられたナビゲーションの構造 $Navigation = (P, O, F, A, C, N, U)$ を以下のように定義する。

- P は、Web ページの集合である。
- O は、ユーザによる操作の集合である。
- F は、フォーム部品の集合であり、 F の要素 $f = \langle p, id, type, dom \rangle$ は、設置されている Web ページ p 、名前 id および型 $type$ 、そして定義域 dom によって定義される。
- A は、ユーザの認証状況であり、 $A = \{logon, logoff\}$ はユーザが認証されているか否かを表す。
- C は、ナビゲーション条件の集合であり、ユーザの認証状況およびフォーム部品への付値で構成される。
- N は、ページ移動の集合であり、 $N \subseteq P \times P \times O \times (C \cup \{\perp\}) \times (A \cup \{\perp\})$ である。 N の要素 $n = \langle p, p', o, c, a \rangle \in N$ は、ページ p において条件 c を満たしておりかつユーザが操作 o を行ったとき、ページ p' に遷移し認証状況が a となるページ移動を表す。ここで、ナビゲーション条件 c をもたない場合および認証状況 a が変化しない場合をそれぞれ \perp で表す。
- $U: O \rightarrow 2^P$ は、ユーザ操作が実行可能な Web ページを表す。 $p \in U(o)$ であるとき、操作 o は Web ページ p で実行可能である。

^{†1} 岡山県立大学

Okayama Prefectural University

3. ステートマシン図を用いたモデル化

3.1 モデル化の枠組み

動的な Web ナビゲーションをモデル化するためには、Web ナビゲーションを構成する要素であるページ移動、認証管理とユーザ操作の振る舞いに加え、ナビゲーションの制御に用いられるユーザの認証状況とフォーム部品の値を表現する必要がある。

本手法では、ページ移動、認証管理、そしてユーザ操作をそれぞれステートマシン図としてモデル化する。ページ移動のステートマシン図では、状態が Web ページを表し、遷移がページ間の移動を表す。認証管理のステートマシン図では、状態がユーザの認証状況を表し、遷移が認証状況の変化を表す。そして、ユーザ操作のステートマシン図では、状態はユーザの状況を表し、遷移はユーザによる操作の実行を表す。提案するモデルでは、ユーザ操作をメッセージとして表し、その上でユーザ操作を表すメッセージをユーザ操作のステートマシン図からアクションとして送信し、ページ移動のステートマシン図でイベントとして受信することで、ユーザ操作に伴う Web ページ間の移動を表現する。

ユーザの認証状況は、前述の通り認証管理のステートマシン図において表す。フォーム部品の値はフォームが所属する Web ページ内のみにおいて有効であるため、ページ移動のステートマシン図において、Web ページに対応する状態のサブ状態として表す。

本稿では、ページ移動のステートマシン図によるモデル化についてのみ述べる。

3.2 ページ移動のモデル化

Web ページを状態として表し、ページ移動を状態間の遷移として表すことで、ページ移動をステートマシン図としてモデル化する。Web ページ間の移動のきっかけとなるユーザ操作は遷移のイベントを用いて表す。

ページ移動 $n = \langle p, p', o, c, a \rangle$ は、Web ページ p および p' を表す状態間の遷移として表す。ここで、ページ移動がフォーム部品の値や認証状況に依存せず認証状況を変化させない（すなわち $c = \perp$ および $a = \perp$ となる）場合は、ユーザ操作を表すイベントのみをもつ遷移となる。

例として、トップページからリンク LIST をクリックすることで商品リストページへと移動するようなページ移動を考える。このページ移動 $\langle \text{TOP}, \text{LIST}, \text{click link LIST}, \perp, \perp \rangle$ は、Fig.1 に示すようなステートマシン図としてモデル化される。トップ

ページおよび商品リストページを表す状態 TOP および LIST を、リンク LIST をクリックするというユーザ操作を表すイベント click_LIST をもつ遷移によって接続することで、リンク LIST のクリックによりトップページから商品リストページへと移動することを表す。

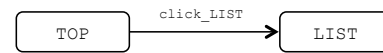


図 1 ステートマシン図によるページ移動のモデル

4. 適用実験

提案法を例題アプリケーションへと適用してステートマシン図によるモデル化を行い、得られたモデルに対して遷移システム解析の統合環境である SAL¹⁾ を用いて動作検証およびテストパターン生成を行っている。この例題アプリケーションは 27 のナビゲーションと 15 のユーザ操作、4 のフォーム部品をもつ。対象システムが小規模であるため、ほとんど時間を要さず検証およびテスト生成が可能であった。本稿では詳細は省略するが、検証によってアプリケーションのもつ誤りを正しく検出できており、また、生成されたテストパターンは全てのナビゲーションを網羅することが可能である。

5. おわりに

本稿では、UML のステートマシン図による Web ナビゲーションのモデル化手法を提案した。また、得られたモデルを用いて SAL 動作検証ならびにテストパターン生成を行い、正しい結果が得られることを確認している。

参 考 文 献

- 1) Bensalem, S., Ganesh, V., Lakhnech, Y., Munoz, C., Owre, S., Rues, H., Rushby, J., Rusu, V., Saidi, H., Shankar, N., Singerman, E. and Tiwari, A.: An overview of SAL, *Proc. of Fifth NASA Langley Formal Methods Workshop (LFM 2000)*, pp.187-196 (2000).
- 2) Chan, W., Anderson, R.J., Beame, P., Burns, S., Modugno, F., Notkin, D. and Reese, J.D.: Model Checking Large Software Specifications.
- 3) Clarke, E.M., Grumberg, O. and Peled, D.: *Model checking*, MIT press (1999).
- 4) Hamon, G., Moura, L.D. and Rushby, J.: Generating Efficient Test Sets with a Model Checker, *Proc. the Second Int'l Conf. on Software Eng. and Formal Methods (SEFM 2004)*, pp.261-270 (2004).