形式的手法と検査ツールによるモデル検査事例と考察

佐藤 祐太[†1]　本間 圭[†2]　高橋 薫[†3]
和泉 諭[†4]　阿部 雄貴[†3]　富樫 敦[†2]
[†1] 宮城大学事業構想学部　[†2] 宮城大学大学院事業構想学研究科
[†3] 仙台高等専門学校　[†4] 東北大学電気通信研究所/情報科学研究科

1. はじめに

論文 [4] では，画面遷移図とアプリケーション内部の状態を2つの有限状態オートマトンとみなし，画面入力値やシステム内部で保持するデータ値を変数で表現することによって，Webアプリケーションをモデル化する手法を提案している．本稿では，このモデル化手法に基づき，検査項目と具体的な検査事例について報告する．

2. 形式的設計モデル

画面遷移図はWebアプリケーションの設計における重要な要素である．画面遷移図は画面そのものを状態と捉え，画面遷移を状態遷移と見なすことにより有限状態オートマトンとして考えることが可能である．画面と同様に，システム内部の状態も重要である．画面により入力された情報を内部に保持し，その変数とその値の対の集合によってシステム内部の状態が決定される．また，画面に対するアクションと同期してシステム内部の状態も遷移する．このシステム内部の状態をシステム環境と呼ぶ．以下，変数付き画面遷移オートマトン M_G，同システム環境オートマトン M_e の定義を与える．システム全体のオートマトンは，これらの直積 $M_G \times M_e$ によって定義される．詳細な定義及び議論は，論文 [4] を参照されたい．

【定義 1】 変数付き画面遷移オートマトンは，6項組 $M_G = \langle Q_G, \Sigma, \delta_G, q_{0G}, \{R_i\}_{i\in[1,n]}, \{v_{0i} | i \in [1,n]\}\rangle$ である．ここで，各項目は次のように定義される．

1. Q_G は，状態（画面）の有限集合である．
2. Σ は，入力記号（画面に対するアクション）の有限集合である．
3. δ_G ($\subseteq Q_G \times (\Pi_{i\in[1,n]} R_i \rightarrow \{true, false\}) \times \Sigma \times Q_G$) は，動作関係（画面遷移）である．
4. q_{0G} ($\in Q_G$) は，初期状態（トップ画面）である．
5. R_i は，画面から入力された変数 x_i の有限な変域である．
6. v_{0i} ($\in R_i$) は，変数 x_i の表示される画面における初期値である．□

【定義 2】 変数付きシステム環境オートマトンは，6項組 $M_e = \langle Q_e, \Sigma, \delta_e, q_{0e}, \{R_i\}_{i\in[1,n]}, \{v_{0i} | i \in [1,n]\}\rangle$ である．ここで，各項目は次のように定義される．

1. Q_e は，状態（変数のその値の集合が表す状態）の有限集合である．
2. Σ は，入力記号（システムに対するアクション）の有限集合である．
3. δ_e ($\subseteq Q_e \times (\Pi_{i\in[1,n]} R_i \rightarrow \{true, false\}) \times \Sigma \times \Pi_{i_\in[1,n]}(\Pi_{j\in[1,n]} R_j \rightarrow R_i) \times Q_e$) は，動作関係（代入文による変数の値の変更）である．
4. q_{0e} ($\in Q_e$) は，初期状態である．
5. R_i は，画面から入力された変数 x_i の有限な変域である．
6. v_{0i} ($\in R_i$) は，各初期状態における変数 x_i の初期値である．□

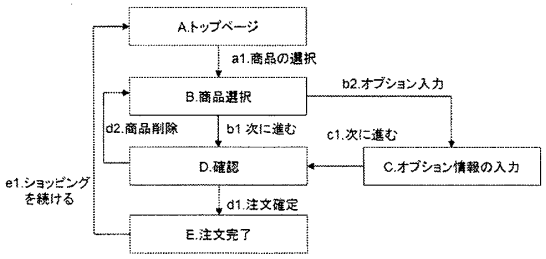

図1　画面ベースの遷移概要図

3. 検査項目

通常のWebアプリケーションでは，以下の4つの条件が成り立つように設計される．

1. 初期画面から到達可能な画面には，遷移可能な画面が必ず存在する．
   Webアプリケーションの操作中に，必ず遷移先画面が存在することをデッドロックフリーと呼ぶ．つまり，この条件はWebシステムはデッドロックフリーであることを要求している．
2. 初期画面から全ての画面へ遷移可能である．
   Webアプリケーションは，初期画面からいずれかのルートを経由して全ての画面へ到達可能である．
3. 全ての画面から初期画面へ遷移可能である．
   Webアプリケーションに終了画面はないが，繰り返し処理を行うためには一連の処理を行った後に初期画面に遷移できることが必要である．
4. 変数は設計されている変域に含まれる．
   設計されている変域はあるが，アプリケーションで稼働させた場合に変数の値が変域に入っていることが必要である．

4. モデル化と検査事例

モデル化と検証の適用例として，代表的なWebアプリケーションである商品注文システムを用いる．図1に画面ベースの遷移概要図を示す．本アプリケーションを，変数付き画面遷移オートマトン M_G2 と変数付きシステム環境オートマトン M_e3 の直積として表す．

x_1, x_2 は，「B. 商品選択」画面で入力される変数，x_1 は

Examples and Discussions of Model Checking using Formal Method with Tool

Yuta SATOH[†1], Kei HOMMA[†2], Kaoru TAKAHASHI[†3], Satoru IZUMI[†4], Yuki ABE[†3], and Atsushi TOGASHI[†2]
[†1]School of Project Design, Miyagi University
[†2]Graduate School of Project Design, Miyagi University
[†3]Sendai National College of Technology
[†4]Research Inst. of Elec. Comm. / Graduate School of Info. Sci., Tohoku University, Tohoku University

Copyright © 2010 Information Processing Society of Japan.
All Rights Reserved.

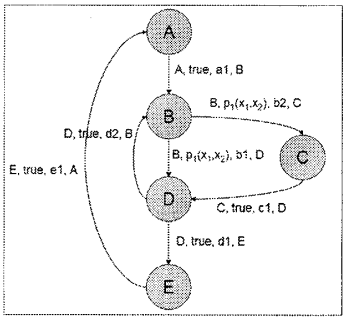

図 2　変数付き画面遷移オートマトン

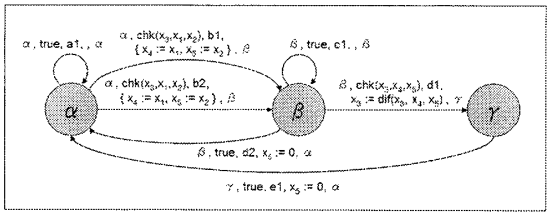

図 3　変数付きシステム環境オートマトン

商品名，x_2 は商品の数量を表す．関数 p_1 は真偽値を返す関数で，引数の値がその変数のドメインに入っていれば true，入っていなければ false を返す．p_1 は，アプリケーションの実装における入力チェックに相当する．

x_1, x_2 は，画面遷移オートマトンと同様に「B. 商品選択」画面で入力された変数で，x_1 は商品名，x_2 は商品の数量を表す．x_3 は内部に保持するデータベースの表で，連想配列である．x_4, x_5 は，「B. 商品選択」画面で入力された値を内部に保持するための変数を表す．図 3 において，$dif(x,y,z)$ は差分を求める関数で，$dif(x,y,z) = x(y) - z$ を表す．これは，連想配列 x から商品 y の数量を取得し，数量 z の値を差し引いた結果を返す関数であり，今回のモデルでは，在庫の引当を行うために利用している．また，$chk(x,y,z)$ は真偽値を返す関数で，$x(y) > z$ ならば true を返す．これは，連想配列 x から商品 y の数量を取得し，数量 z の値と比較し，連想配列の値が大きければ true となる関数で，画面から入力された商品名と数量について，テーブルの値と比較し，数量が十分であることや，内部で保持している商品名と数量について，テーブルの値と比較し，数量が十分であることを確認する関数である．

前述したモデルをモデル検査ツール SPIN を利用して検証する．SPIN は Promela という独自の仕様言語で記述した形式仕様を入力とし，検証器を生成する．まず，画面遷移オートマトンとシステム環境オートマトンをそれぞれ Promela のプロセスとして記述した．Promela のラベルで示した位置を状態とする表現法を用いて，各オートマトンの状態を記述し，オートマトンの直積は 2 つのプロセスをチャネルを用いて同期通信させることにより記述した．変数を用いた遷移条件は，Promela のガードコマンドを用い，代入は通常の変数への代入を用いて記述を行った．チャネルを用いて画面遷移オートマトンとシステム環境オートマトンを連動させた．

Web アプリケーションの検査項目を SPIN を用いて検査する方法を述べる．

1. 初期画面から到達可能な画面には，遷移可能な画面が必ず存在する：この検査には，Promela のガードコマンドの特徴を用いる．画面遷移オートマトンでは，各画面から送信可能な入力記号をガードコマンド（do … od）の中に記述している．これにより，記述されている入力記号のいづれかが送信される．同様に，システム環境オートマトンでは，各状態で受信できる入力記号の内容をガードコマンド（do … od）の中に記述する．送信された入力記号をシステム環境オートマトンのある状態で受信可能であれば，受信した内容に応じたシステム環境の遷移が行われる．

しかし，送信された入力記号が，システム環境オートマトンのある状態で受信できない場合，画面遷移オートマトンは送信した入力記号が適していなかったと判断し，ガードコマンド内に記述されている別の入力記号を送信する．

SPIN では，ガードコマンドに記述されている内容が全て実行不可能の場合，デッドロック状態と判断する．つまり，SPIN の到達性検査を用いると，到達した画面とシステム環境の対において，画面とシステム環境が同時に遷移可能な入力記号が存在しないとデッドロックが発生する．すなわち，SPIN でデッドロックが発生しないということは，3 節で定義したデッドロックフリーと同義となる．

2. 初期画面から全ての画面へ遷移可能である：この検査には SPIN の LTL 式を用いた検査を用いる．ただし，LTL 式で「全ての画面」を表現することはできないため，初期画面から任意の画面へ遷移可能であることを確認する．以下の LTL 式が成り立つということは，初期画面からパス上のある時点で任意の画面に遷移可能ということになる．

LTL 式：$p \rightarrow <> \ q$,
where p = 初期画面，q = 任意の画面.

LTL は CTL(Computation Tree Logic) と異なり，複数のパスが存在する場合に，「その中のあるパス」という検査ができない．そこで，上記 LTL 式の否定をとり，SPIN で検査を行う．遷移可能な場合にエラーとなり，その反例として初期画面からその画面までのルートが出力される．指定する画面をその都度変更し，全画面を検査することにより，初期画面から全ての画面へ遷移できることを確認することができる．

3. 全ての画面から初期画面へ遷移可能である：上記と同様の手法を用いて検査することが可能である．

4. 変数は設計されている変域に含まれる：上記 2. と同様に，この検査には SPIN の LTL 式を用いた検査を行う．アプリケーションの全ての状態で変数の値が変域の最大値 (x_{max}) 以下であり，変域の最小値 (x_{min}) 以上であることを検査する．

LTL 式：[] (p && q),
where $p : x >= x_{min}$ $q : x <= x_{max}$

5. おわりに

本論文では，画面遷移とシステム環境の遷移をシステムに対する入力記号によって結びつけることによりシステム全体のオートマトンを生成するという手法を用い，SPIN によって Web アプリケーションで成り立つべき性質を検査することが可能であることを示した．

参考文献

[1] Clarke, E.M., Grumberg, O. and Long, D.E.: Model Checking and Abstraction, ACM OPLAS, Vol. 16, No. 5, pp. 1512-1542 (1994).

[2] Holzmann, G.J.: The Model Checker SPIN, IEEE Transactions on Software Engineering, Vol. 23, No. 5, pp. 279-295 (1997).

[3] 崔銀惠, 河本貴則, 渡邊宏: 画面遷移仕様のモデル検査, コンピュータソフトウェア, Vol. 22, No. 3, pp. 146-153 (2005).

[4] 本間圭, 高橋薫, 和泉諭, 阿部雄貴, 富樫敦: 変数を用いた Web アプリケーションのモデル化と形式的手法による検査, 電子情報通信学会技術研究報告, Vol. 109, No. 298, pp. 31-38 (2009).

Copyright © 2010 Information Processing Society of Japan.
All Rights Reserved.