B-029

WeB アプリケーション設計の UML アクティビティ図に対するセマフォア導入とモデル検査

Introduction of Semaphore for Activity Diagrams in UML of Web Application Design and Model Checking

小林 巧 †　山田 豊 ††　和崎 克己 †††
Takumi Kobayashi　Yutaka Yamada　Katsumi Wasaki

1　はじめに

UML アクティビティ図 [1] は，振る舞い図に属するダイアグラムで，処理の実行順序や条件，制御などを表現できる．他方，ソフトウェアの上流設計仕様を形式言語でモデル化し，自動検査ツールを用いて動作検証あるいは反例を検出するというアプローチが注目されている．SPIN [2] [3] はモデル検証ツールの一つであり，検証対象の振る舞いの全状態を探査し，検証条件（仕様）を充足するかを検査する．

本報告では，並列処理と同期機構に対応するように拡張された，従来研究で開発中の UML アクティビティ図から PROMELA に自動変換する変換器 [4] を用いて，Web アプリケーションに適用した結果について述べる．基本的な同期機構であるセマフォアを導入し，複数ユーザが更新対象とするファイルロックの排他制御に利用した．

2　SPIN モデル検査ツール

2.1　概要

SPIN は，G.J.Holzmann を中心に開発されているモデル検査ツールである．実際の開発現場でも広く普及しており，並列システムの振る舞いの検証などに多く用いられている．SPIN は，専用の仕様記述言語である PROMELA 言語で記述されたモデルから，無限長の語を扱う有限オートマトン（ Buchi オートマトン）を生成し，検証器と呼ばれる C 言語で書かれたプログラムを生成する．C コンパイラを使ってコンパイルした後，検証器を実行すると，検証器はあらゆる状態遷移を生成しながら，モデルが所定の性質を満たしているかを検証する．

2.2　表明と線形時相論理

SPIN では，モデルの正当性を表現するために，表明（assert）と線形時相論理（LTL）を用いることができる．表明では，assert （条件式）文を記述することで，その時点において，モデルが取ってはならない反例を示すことができる．線形時相論理は，命題論理式に，時の概念を表現できる時相演算子を加えた論理体系である．これにより， PROMELA で記述したプロセスが時系列に沿って成り立つべき性質を論理式の形で指定できる．線形時相論理では， LTL 式で性質を記述することで，モデルがその性質を満足するか否か，満足しない場合にはその反例を検査することができる．

† 信州大学大学院工学系研究科，Graduate School of Science and Technology，Shinshu University.
†† （株）プラグマティック・テクノロジーズ，Pragmatic Technologies Co. Ltd.
††† 信州大学工学部，Faculty of Engineering，Shinshu University.

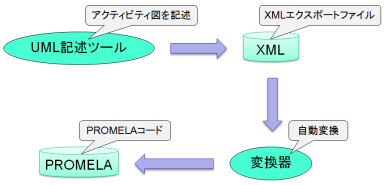

図 1　変換の流れ

3　UML アクティビティ図から PROMELA への変換

3.1　UML アクティビティ図

UML アクティビティ図を使用し，システムをモデリングする．UML アクティビティ図は，UML2.0 で独立して定義された振る舞い図の一つで，モデリング対象が行う個々の活動をアクションとして記述し，アクション間の順序関係を制御フローで連結する．並列処理の開始を表すフォークとその同期化を図るジョイン，条件付き振る舞いを示すディシジョン，マージなどを使用することで多様な処理を表現できる．

3.2　変換の流れ

図 1 に，UML から PROMELA への自動変換 [4] の流れを示す．まず，検証したいモデルを UML アクティビティ図で記述する．アクティビティが扱うオブジェクトの構造はクラス図で記述しておく．UML 記述ツールは，（株）チェンジビジョンの astah* Professional を使用した．ツールの機能によって，UML の情報を XML 形式のファイルとして出力する．変換器は，XML ファイルを受け取ると，アクティビティ図に関する要素を抜き出し，中間データを構成する．その中間データからアクションノードを骨組みにして，必要な情報を PROMELA の形式で出力する．PROMELA の表現で，アクティビティ図の要素では表せないものについては，UML の「タグ付き値」を利用している．「タグ付き値」は， UML の拡張メカニズムの一つで，モデル要素に任意の情報を定義することができる．今回，セマフォアに関するタグ付き値が新たに追加された．表 1 に，PROMELA 変換用に定義された「タグ付き値」の一覧を示す．

4　排他制御とセマフォアの導入

従来研究で開発中の変換器が拡張され，フォーク・ジョイン要素を用いた並列処理に対応するようになっ

Copyright © 2012 by
The Instiute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

アクティビティ図全体に付与するタグ	
Spin.process	プロセス名（省略時は P1,P2,…）
Spin.number_of_processes	プロセス数（省略時は 1）
Spin.ltl_spec_pattern	仕様パターン（省略可）
アクションに付与するタグ	
Spin.label	プロセス中のラベル（省略可）
Spin.pre_condition	事前条件（省略可）
Spin.post_condition	事後条件（省略可）
Spin.semaphore	P または Pn（n=1,2,3…） V または Vn（n=1,2,3…）
シグナル送信アクションに付与するタグ	
Spin.channel_input	チャネルに送信するメッセージ
Spin.channel	チャネル名（省略時は ch0,ch1,…）
Spin.channel_buffer_size	バッファサイズ（省略時は同期通信）
Spin.channel_message_type	メッセージタイプ（省略時は Spin.channel_input の型で生成）
イベント受信アクションに付与するタグ	
Spin.channel_output	チャネルから受信したメッセージの出力先

表 1　PROMELA 変換用に定義されたタグ付き値

```
#define P(s) atomic { s > 0 -> s-- }
#define V(s) s++;
bit s = 1;
        .
        .
P(s)
  goto action17;
        .
        .
V(s)
  goto junction26;
        .
        .
```

図 2　PROMELA コードの例

た．並列処理を取り扱うことを考えると，モデリングにおいても資源の同時アクセスについて考慮する必要があり，一つの資源に対し二つ以上のプロセスが同時に扱う，という状況を作らないように，排他制御をかけるべきである．今回は基本的な同期機構であるダイクストラのセマフォアを導入した．

セマフォアとは OS が用意するグローバルなフラグのようなもので，そのフラグを待っているプロセスが複数あったとしても，セマフォアから実行の許可が得られるプロセス一つに制限させることができる．UML アクティビティ図中でセマフォアを表現するにあたり，クリティカルセクションの前後にあるエッジに対して，P/V 操作を行うための「タグ付き値」を付与する．図 2 に，変換器から出力された PROMELA コードの一部を示す．

5　CMS 向けオーサリングシステムへの適用

提案手法の有用性を確認するため，ある教育用 CAI 課題オーサリング援用システムの画面遷移設計を取り上げ，実装されている変換器を用いて適用実験を行った．この援用システムでは，編集したい科目，章を選択し，章タイトルの編集，または問題の新規作成，既存の問題の修正，問題のプレビューを行うことができる．援用システムには，シラバスファイルと問題ファイルが存在し，編集を行う際に，それぞれに対してロックをかけている．これらのファイルロックに関して，バイナリセマフォアで実現した．図 3 にオーサリングシステムの例を示す．

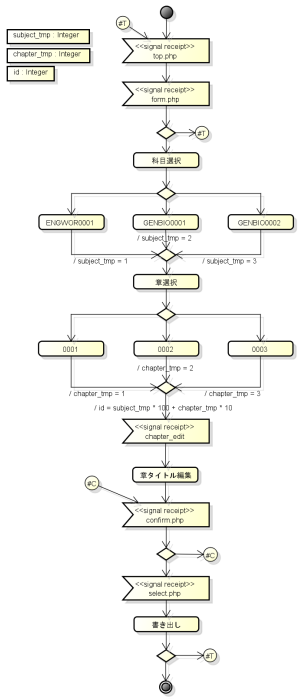

図 3　オーサリングシステム設計のアクティビティ図（抜粋）

6　まとめと今後の課題

今回は，従来研究で以前に試作された UML アクティビティ図から PROMELA で記述された検証用モデルに自動変換する変換器から，拡張された機能のうち，同期機構に着目し，アクティビティ図で記述した CAI 課題オーサリングシステムのモデルに，セマフォアを導入した．今後は，ファイル名のように文字列で扱うものや，問題のように数が固定的ではなく可変的なものに関して，より実際のシステムに近いモデルを作成し，検証できるかどうか，検討していく．

謝辞　本研究の一部は科学研究費 (23500174）の助成を受けたものである．

参考文献

[1] J. Rumbaugh, et.al : The Unified Modeling Language Reference Manual (2nd Edition), Pearson Higher Education (2004).
[2] G.J. Holzmann: THE SPIN MODEL CHECKER, Addison Wesley (2003).
[3] 中島 震: SPIN モデル検査, 近代科学社 (2008).
[4] Y. Yamada, K. Wasaki: Automatic Generation of SPIN Model Checking Code from UML Activity Diagrams, IJACT, Vol.3, No.8, pp.189-197 (2011).

Copyright © 2012 by
The Instiute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.