B-012

SPINによる検証支援ツールの開発*

野口 拓也 †
茨城工業高等専門学校‡
産業技術システムデザイン工学専攻§

小飼 敬¶
茨城工業高等専門学校
電子情報工学科‖

滝沢 陽三**
茨城工業高等専門学校
電子情報工学科

1 はじめに

近年，ソフトウェアに対する信頼性や安全性に対する関心が高まり，形式手法が注目を集めている．形式手法とは代数論，集合論やグラフ理論など数学理論をベースとした手法のことで，要求されている仕様を数学的に厳密に記述することができ，作成されたソフトウェアの性質を数学理論によって検査・検証することができる．

形式手法は大きく分けて形式仕様記述と検査・検証ツールの二つに分けることができる．しかし，仕様記述は数学的表記が中心で自然言語から遠いことから理解に時間がかかり，ツールの利用には形式手法のメカニズムの基本知識が必要であるため広くは普及していない．

そこで本研究では検査・検証ツールのSPIN[1][2]について詳しく知らなくてもSPINを使った検証を行えるようにする支援ツールの開発を目的とする．

2 SPINとは

SPINはGerard J. Holzmann博士が中心となって開発したモデル検査に基づく自動検証ツールで，振舞い仕様と呼ぶシステムのひとつの側面に限定することで自動検証を可能としている．また，Promela(Prosess Meta Language)という言語で分散システムを状態モデルとして記述し，検証条件はLTL(Linear Temporal Logic)で記述される．モデル検査だけでなくシミュレータの機能も持ち実行履歴を取ることができ，1980年代から20年以上にわたって研究，開発が進められており，産業界でも多数の適用事例が報告されている[3].

3 ツールの開発

3.1 概要

SPINを使ってモデルの検証を行うにはPromela記述を書く必要がある．しかし，Promelaについて一から勉強したのでは時間が掛かるのでどうしても導入の敷居が高くなってしまう．

そこで振舞い仕様は状態遷移図で表現できるので，ユーザは状態遷移図を描くだけでPromelaの記述を書かずに自動で検証を行うツールを開発した．なお本研究では，状態遷移図をJUDEのステートマシン図描画機能で描くことにする．

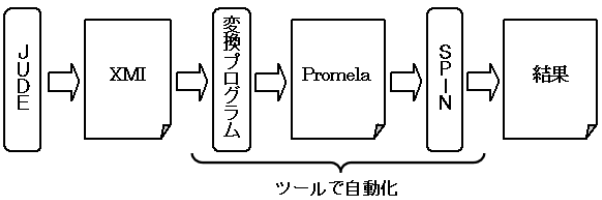

図 1: システム構成

*Development of Model Checking Tool by SPIN
†Takuya Noguchi
‡Ibaraki National College of Technology
§Advanced Course for Information
¶Kei Kogai
‖Electronic and Computer Engineering
**Yozo Takizawa

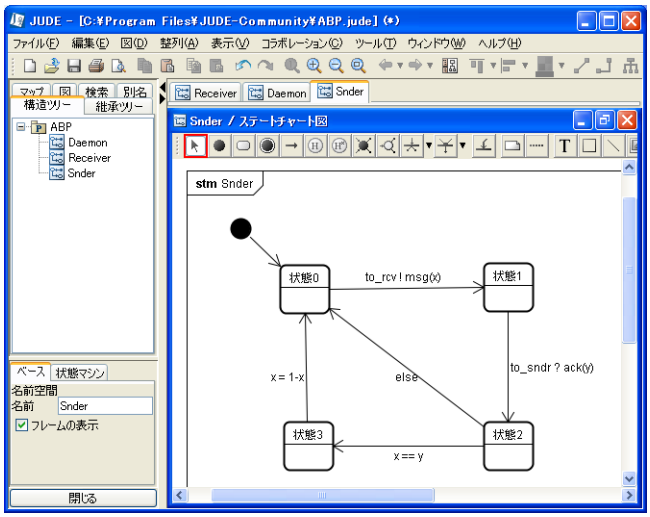

図 2: JUDE で描いたステートマシーン図

```
<UML:Transition xmi.id="e12-fvka0gn9-24a6ev--wuur07-
  f70d7c30d1cca2f246dd91047da8ae40" name="to_rcvr+%3F+mag%28y%29"
  version="0" unSolvedFlag="false">
  <UML:Transition.trigger>
    <UML:Event xmi.id="qmm-fvka0gn9-24a6ev--wuur07-
      f70d7c30d1cca2f246dd91047da8ae40" name="to_rcvr+%3F+mag%28y%29"
      version="0" unSolvedFlag="false" />
  </UML:Transition.trigger>
  <UML:Transition.source>
    <UML:StateVertex xmi.idref="dgm-fvka0gn9-24a6ev--wuur07-
      f70d7c30d1cca2f246dd91047da8ae40" />
  </UML:Transition.source>
  <UML:Transition.target>
    <UML:StateVertex xmi.idref="dsm-fvka0gn9-24a6ev--wuur07-
      f70d7c30d1cca2f246dd91047da8ae40" />
  </UML:Transition.target>
  <UML:Transition.stateMachine>
    <UML:StateMachine xmi.idref="d5u-fvka0gn9-24a6ev--wuur07-
      f70d7c30d1cca2f246dd91047da8ae40" />
  </UML:Transition.stateMachine>
</UML:Transition>
```

図 3: XMI サンプル

3.2 構成

ユーザは検証したいモデルのステートマシン図を JUDE で作成し XMI 出力機能を用いて XMI ファイルを出力させる．このとき出力された XMI ファイルの一部を図 3 に示す．

出力された XMI ファイルを変換プログラムが読み込み，遷移先の xmi.id 属性を辿りながら name 属性の値を元にして Promela の記述に変換し Promela ファイルを作成する．

ユーザにどんな検証を行いたいのかを選択させ，それを受けてツールが SPIN のコマンドを実行しシミュレーションもしくは検証器を使った検査を自動に行う．

4 課題と発展

作成したツールは図 2 の様な簡単な状態遷移図ならば自動で変換して検証することが出来る．しかし，コンポジット状態を含むような状態遷移には変換プログラムが対応できていないのでこれに対応させることが今後の課題となる．

また，本来ユーザが SPIN を意識せずに使うには「検証した結果，どう評価すればいいのか？」という情報をユーザに示す機能が必要になるのだが，今回はそこまで実装することができなかった．得られた結果をユーザにどう理解させるかも検討する必要がある．

5 まとめ

本ツールは，SPIN についてあまり知らなくても SPIN での検証が行えるようになっている．しかし，複雑な Promela 記述は作ることができないので，今後の開発では変換プログラムを改良し，より複雑な Promela 記述を作れるようにして実際に使えるように完成度を上げていきたい．

参考文献

[1] Spin - Formal Verification
http://spinroot.com

[2] Basic Spin Manual
http://www.asahi-net.or.jp/~hs7m-kwgc/spin/Man/Manual_japanese.html

[3] 中島震『SPIN モデル検査』(近代科学社, 2008)