トップエスイー修了制作

Promelaにおける割り込み制御処理のフレームワーク作成および デッドロック原因推定について

中山　仁

開発における問題点

- 組込みシステムの**割り込み処理をPromelaの通常の記述方法**で記述してモデル検査 を行うことができない。
- 検証結果の反例から**原因特定に時間を要する**場合がある。

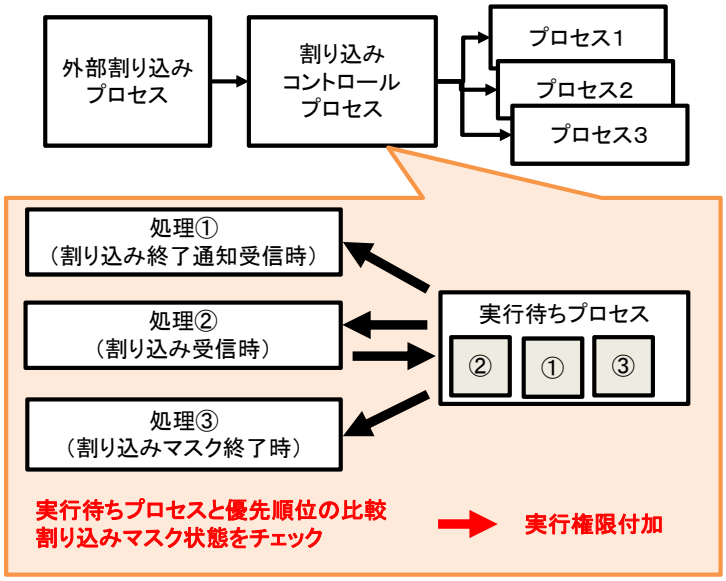
手法・ツールの適用による解決

利用者の負担を減らすために下記を提案する。

- Promelaにおいて簡単に割り込み制御処理を記述できる**フレームワーク**
- 反例からデッドロックの**原因推定ツール**

割り込み制御処理 フレームワーク

割り込みコントールプロセスで割り込みの各プロセス実行制御
「多重割り込み」や「割り込みマスク」にも対応



テンプレートファイルの決められた部分に記述して利用

テンプレートファイル

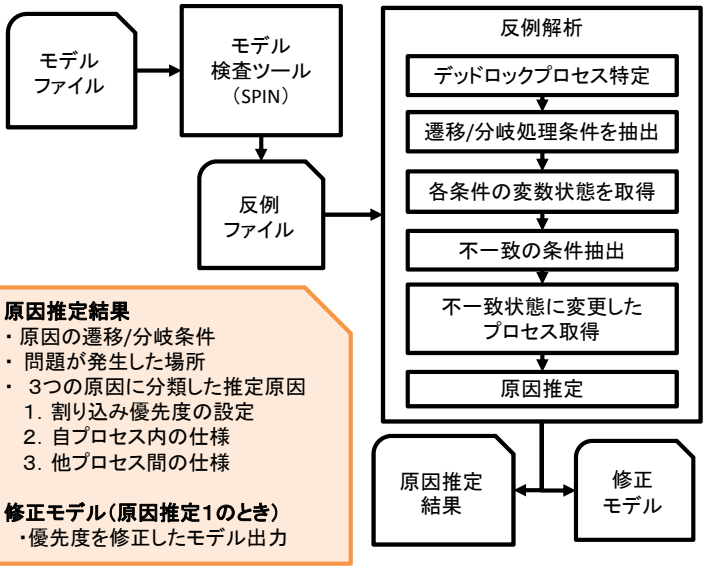
```
/* 割り込みコントロールプロセス*/
proctype Scheduler(){
<プロセス管理情報を定義>
.....
/* 各タスクのプロセス */
<各プロセスのモデルを記述>
```

【各プロセス管理情報】

項目	説明
優先度	プロセスの実行する優先度
実行フラグ	実行権限を与える変数（”provided”句に使用）
状態	「待機中」、「実行中」などのプロセス状態
開始ステータス	プロセスの開始ステータス
ステータス	プロセス内で定義するステータス変数に使用

デッドロック原因推定ツール

反例の実行ログより**遷移/分岐処理条件**の不成立になっている条件を明確にして原因推定



原因推定結果
- 原因の遷移/分岐条件
- 問題が発生した場所
- 3つの原因に分類した推定原因
  1. 割り込み優先度の設定
  2. 自プロセス内の仕様
  3. 他プロセス間の仕様

修正モデル（原因推定1のとき）
- 優先度を修正したモデル出力

まとめ・課題

●まとめ
- 割り込み処理モデル作成の簡易化
  フレームワークを利用することで、割り込み制御処理部分を意識することなくモデル作成が可能
- 単一問題（2つのプロセス間）の原因推定
  単一の問題によりデッドロックが発生しているモデルに対しての問題箇所と原因の推定が可能

●課題
- 複数問題（複数のプロセス間）の原因推定
- 実務での利用した試験と試験結果検証

NII 国立情報学研究所 National Institute of Informatics
トップエスイー ～サイエンスによる知的ものづくり教育プログラム～