テスト工程における形式手法の利用

矢田部俊介†、北村崇師†、Ling Fang†、
Do Thi Bich Ngoc†、大崎人士†

本講演では、自動車のマルチコア対応車載 RTOS 開発における製品テストの検証技術についての産業技術総合研究所組込みシステム技術連携研究体と東芝セミコンダクター社の装置提供型共同研究（平成 22 年度～平成 23 年度）の、平成 22 年度の成果を紹介する。本研究の目標は形式手法の応用により、負荷テストを中心に、テスト工程における説明力を増し、それによりテスト工程の品質の向上を可能にすることである。平成 22 年度は、テスト設計技法に関しては、ゴール分析によるテスト関心事の明確化のための記述法（FOT 記法）のプロトタイプ版を開発した。また、シミュレーション環境に関しては、ロック機構に的を絞り、便宜上コアが二つの環境でタスクの起動や終了･切り替え等を再現できるシミュレーション環境のプロトタイプ版を構築した。今後は、実機の動作ログの取得の困難さなどの問題点を解決するため、さらなる研究が必要である。

第１章　　背景

地球温暖化問題への関心が高まるにつれて、自動車産業界は、従来の化石燃料式のエンジンに代えて、燃費のよいガソリン-電気のハイブリッド式や電気式のエンジンシステムへと主力商品をシフトさせつつある。ハイブリットエンジンは、従来型エンジンよりさらにきめ細かい制御を必要とするため、こうした動きに合わせて、組込み産業分野でも次世代型車載機器の研究開発が進行している。

車載組込み機器は、各機器の制御ユニットがマルチコア化し、また機器同士がネットワーク接続され、複雑で大規模なシステムを形成している。組込み機器ネットワーク全体の振る舞いは、ネットワークを構成するユニットの数が増えるに従って、非常に複雑なものとなり、従来の組込みシステムの設計方法では対応できない。

特に深刻なのはソフトウェアのテスト工程である。従来のテスト手法は、自然言語ベースの仕様にもとづく人手によるテスト設計、実機（ハードウェア）を使った動作テストという属人的技術に頼るものがほとんどで、複雑で大規模な組込みネットワークシステムに対しては十分な品質保証や設計テストの自動化・効率化が難しい。

また、本質的にテストはシステム外部からの観測であり、テスト対象の詳細な挙動の情報は得られない。実際にテストをした結果も、異常停止をしないなどの大まかな結果はわかるものの、システム全体があまりに複雑なためそれ以上の分析が出来ないことも多く、既存のテストの説明力は不十分である。

テスト結果の分析手法に関しても、同様の説明可能性の不足という問題を抱えている。マルチコア上で OS を動かす場合、ランダムに他のコアから割り込みが入る可能性がある。「割り込みが入ってもある機能が問題なく動作するか」を検査する場合、その機能に関する長時間の動作試験（負荷テスト）

を行い、機器全体が異常停止するかしないかをチェックし、「長時間、多数の割り込みが入ったはずなのに、機器全体は異常停止しなかった」事を持って、その機能に関し深刻な問題は起こらなかったと見なすのが一般的である。しかし、異常停止するかどうかだけの基準はあまりに大まかすぎ、異常停止しなくても実は内部で異常動作が起きていた可能性は否定できない。従って、負荷テストはコストがかかる割には、その結果が何を説明するのかという説明力が弱いという点が問題である。

第２章　　目的

本研究は、自動車のマルチコア対応車載RTOS（AUTOSAR 規格（AUTOSAR）R4.0 SC2）開発における製品テストの検証技術についてである。対応するテストの種類は、OS の実装に依存しない負荷テスト環境を作るため、仕様ベースのテスト（ブラックボックステスト）とする。そして、「何を負荷テストで検査したかを説明可能にする」汎用性のある技術を開発することを目的とし、テスト設計手法と動作の正常性の確認方法を開発する。

テスト設計手法に関しては、テストと検査しようとする性質の対応関係を確立する。すなわち、テストを行う際の関心事（以後「テスト関心事」）を明確化し、あるテストケースがある関心事を満たしているとはどういうことかに関する基準を策定する。また、その基準を支える模範的テストケースを作成する。

一方、機能の正常さの確認するための方法として、シミュレーション技法を採用し、「シミュレーション環境」を開発する。シミュレーション技法では、モデル検査技法のように状態爆発を起こすことなく、特定のテストケースが入力された際の実際の動作が仕様通りであったかどうかを確認できる。

想定される本技術の使用法は以下の通りである。実機にテスト設計手法を用いて設計したテストケースを入力し、動作ログを出力させる。同時に、シミュレーション環境にも同じテストケースと実機ログを入力し、実機の動作が再現可能であるかを検査する。もしも実機の動作がシミュレーション環境において再現できない場合、実機の動作は仕様書通りではないことになる。この再現可能性の検査により、実機の動作の詳細について正常性が判定できる。

以上の検証技術により、負荷テストを行った際にその結果が製品のどのような性質を証明したのかを明確にすることができ、テスト工程の品質を向上することができる。このことは、テスト効率の費用対効果の上昇を意味する。従って、この方法を応用することで、将来的なテスト工程のコスト削減を進めることが出来ると期待される。

第３章　　研究成果

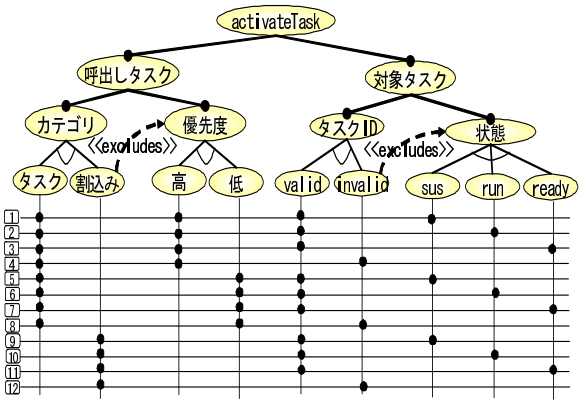
本章では、平成 22 年度の研究成果について述べる。

第１節　　テスト設計手法

テスト設計手法に関し、平成 22 年度は、ゴール分析によるテスト関心事の明確化のための記述法（FOT 記法）を確立し、またその形式的意味論のプロトタイプ版を策定し

た［北村崇師，2011］。また、FOT 記法による構造木の記述を支援するツールの基本設計を行った。本節では、FOT 記法の説明を行う。なお、技術的詳細に関しては文献［北村崇師，2011］を参考にされたい。

FOT 記法では、ゴール分析手法に基づき、テストを行う際の関心事を分析する。そのため、分類木手法（Classification Tree Method; CTM）をゴール分析手法とドッキングさせる。分析は、ゴール分析手法を用いて大きなテスト関心事を細かいゴールに分解し、さらに、それぞれのゴールについて、そのゴールを満たすためにはどのような状況でテストを行えばいいかという分析を CTM でサポートする。



上図は、RTOS である OSEK/VDX の API 函数 activateTask （suspend 状態のタスクを起動する函数）の機能テスト分析の見本である。

図上半分は、分類木によるゴール分析および状況分析である。下部の表は、上の分類木分析によるテスト関心事を網羅するためのテストケースを列挙している。表中の横線が一つのテストケースに相当し、よって、図では、この分類木から 12 個のテストケースが得られることを示す。

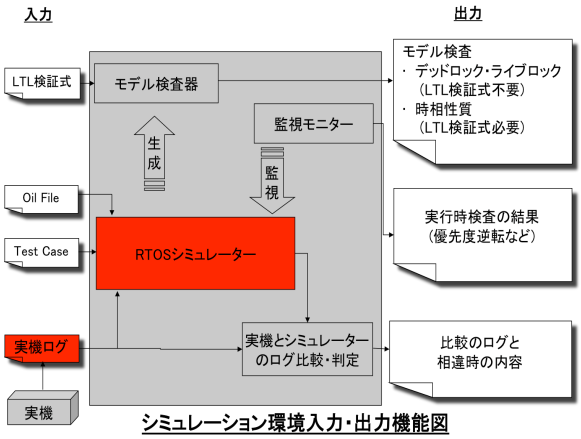
分類木手法はテスト分析・設計に有効なアプローチであるが、その問題の一つは形式的な議論が欠けていることである。分類木手法はその基本的な考え方が示されているのみで、その正確な定義も意味論が与えられていない。従って、計算機上で扱う際には、定義の欠如によりデバッグ機能などを装備できず、また意味論の欠如によりテストケースの自動生成も難しい。

そのため、本研究では、分類木を命題論理式に翻訳する手順のプロトタイプ版を作成した。さらにそれを基に分類木から得られるテストケースの集合を定義した。また、関数型言語 Haskell を利用したテストケース自動生成機構のプロトタイプを作成し、ある程度のテスト自動生成を可能にした。ただし、現在の版は制約条件（特に優先条件）の記述に完全には対応していないため、来年度のさらなる改良が必要である。

第２節　　シミュレーション環境

シミュレーション環境に関し、平成 22 年度は、主にロック機構に的を絞り、モデル検査機 SPIN を用いたシミュレーション環境で実機の動作を再現し、また実機テストの結果とシミュレーターの動作を照合し内部の動作を推定するための機構のプロトタイプを開発した（Ling Fang, 2011）。

テスト生成器によって生成されたテストケースは、実機とシミュレーション環境に入力される。シミュレーション環境の中核であるシミュレーターは仕様に従って各状態の基準値を計算し、また実機の出力したログファイルを入力として受け取り、両者を比較し異同を判定する。



シミュレーション環境入力・出力機能図

シミュレーション環境は上図に示したように、「RTOS シミュレーター」、「監視モニター」、「モデル検査器」と「実機と RTOS シミュレーターのログ比較・判定器」からなる。本システムは仕様レベルと一部のコードレベルの欠陥を検出する。

本研究では、シミュレーション環境の中核部分となる RTOS シミュレーターを、モデル検査器 SPIN の記述言語 Promela によって、AUTOSAR 規格 RTOS の仕様を記述し、シミュレーション環境を構築した。補足すると、Promela を用いて記述したため、デッドロックや優先度逆転などの様相性質を検査することが SPIN の機能により可能である。

RTOS シミュレーターは、テストケースを入力された場合、仕様に従った各タスク、CPU、メモリ、割り込みなどの振る舞いを出力する。RTOS シミュレーターから生成されたモデル検査器は二つの機能がある。一つ目はデッドロック・ライブロックの自動検査である。二つ目は LTL 式による様相性質、即ち到達性、安全性など、またユーザが検証したい様相性質などの検証であるが、ただし LTL 式の自動検査は爆発問題があるため、すべての性質が検査できるわけではない。そのため、監視モニターを作り、RTOS シミュレーターの行動基準、例えば優先度逆転がないなどの規則を定め、RTOS シミュレーターの実行列を監視し、違反がある場合は報告する。「実機とシミュレーターのログ比較・判定」は比較・判定し、トレースと違い時の状況を報告する。

第４章　　今後の課題

本研究の結果、手法の有効性が証明されたが、一方で、実用化のための課題もいくつか明確になった。以下の課題を解決することが来年の急務である。

第１節　　テスト設計手法

FOT 手法の課題は、形式的意味論の洗練である。 [北村崇師, 2011]で定めた形式的意味論は、大枠において完成しているものの、制約条件に関し、排反条件・優先度記法などをサポートしていない。これらの記法の論理式への翻訳法を与えるのが重要である。また、実際の現場のテストケース生成に関し、現在の記法で十分であるかどうか、実例を持って検証する必要がある。

第２節　　シミュレーション環境

第１項 AUTOSAR v4.2 準拠

現在のシミュレーション環境は、プリエンティティブ・ポリシー、タスクの個数や優先度、ロックのアルゴリズムなど、環境設定や実装に依存する。今後、環境設定や実装に依存しない AUTOAR R4.0 SC2 準拠のシミュレーターになるように、シミュレー

ターをさらに改善する必要がある。

第２項 ログファイルの問題

本研究において、テストの説明可能性は、実機の動作についての十分に詳しい情報をログファイルとして取得し、それをシミュレーターで再現できることによって担保される。そのため、負荷テストについて、正しく十分に詳しいログファイルを取得できることは決定的に重要である。しかし、検討を続けるにつれて、ログ取得に関し、いくつかの問題があることが明らかになった。来年度は、この点を解決する必要がある。

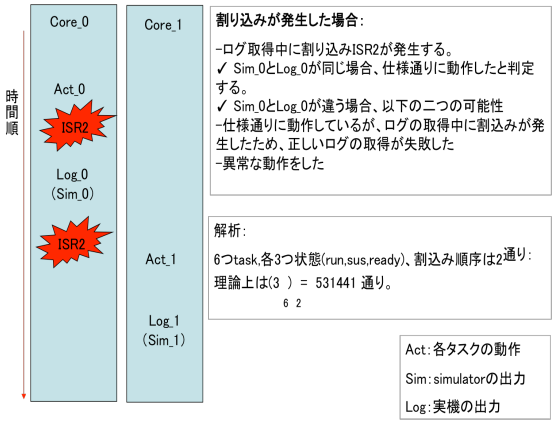
OS の制約（割り込み処理）

AUTOSAR OS では、内部状態の取得は、Print 文を使用し、内部状態を書き出させる。以下にその例を示す。

```
Task0 = {
  activate (Task3);
  PrintOut(core0, task0,activate, task3)
  activate (Task4);
  PrintOut(core0, task0,activate,task4)
  terminate;
  PrintOut(core0, task1,terminate)}
Task1 = {
  terminateTask;
  PrintOut(core0, task1,terminate)}
Task2 = {
  activateTask(Task5);
  PrintOut(core0, task2,activate,task5)
  activateTask(Task4);
  PrintOut(core0, task2,activate,task4 )
 terminateTask;
  PrintOut(core0, task2,terminate)}
 ｏｏｏ
```

ここで、青字で書かれた Print 文は、各コマンドの直後に必ず入り、コマンド実行後変化した状態をシリアル通信を通じて外部に書き出す働きをする。Print 文は通常のコマンドと同じ優先度を持つため、割り込みは全て Print 文よりも優先度が高い。

例えば上図 Task2 において、activateTask(Task5)を実行し Task5 が run 状態になり、その状態を Print 文を用いてログファイルに書き出したが、その後割り込みが入り、その結果 Task5 が終了してしまったとしよう。その場合、実際は Task5 は終了しているにもかかわらず、ログファイル上では run 状態のままである。



このように、割り込みの優先度が高いため、ログファイルに書かれた状態が実際の CPU の状態を正しく反映していない可能性がある。

ハードウェア的な制約

実機の内部状態をファイルに書き出すためには、シリアル通信によって実機状態を接続している PC に通信する必要があるが、シリアル通信によるログの出力には（通信ボーレートにも因るが）CPU 処理に対して桁違いに遅い。 このため、リアルタイムなログをシリアル通信にてリアルタイムに出力することは ECU の動作に大きく影響を与える。

第３節 テスト設計手法とシミュレーション環境の連携

研究初年度は、FOT 記法およびシミュレーターという要素技術の開発に注力し一定の成果を得、負荷テスト工程の説明力の強化

と最適化の可能性を示すことが出来た。一方で、ログファイルに関する困難のため、本来の目標であるところの要素技術を統合した機構を現場でどう使用するかを明確にし、それによってテスト工程の説明力強化への道筋を描くという点まで達し切れていない。今後は、不十分なログしかとれないという前提の下で、開発した二つの要素技術を結びつけ、それをどう開発現場で使用するかを明確化すること、またこの技術により解決できる問題の具体的な例などが必要になると思われる。

第５章　　参考文献

Ling Fang, Takashi Kitamura, Shunsuke Yatabe, Hitoshi Osaki (2011 年 6 月). Formal verification environment for an AUTOSAR multicore RTOS. Submitted to ICFEM 2011 .

北村崇師, ling fang, 矢田部俊介, 大崎人士. (2011 年 1 月). 形式分類木手法に向けて. (情報処理学会, 編) ウインターワークショップ 2011・イン・修善寺.

† 産業技術総合研究所 組込みシステム技術連携研究体