6H-03

複数タスクから成る組込み制御ソフトウェアの振る舞い検証

岩崎 友哉† 俞 明連† 横山 孝典†
†東京都市大学

1. はじめに

近年，組込み制御ソフトウェアの開発量は増大しており，開発効率を上げる必要がある．その手法としてモデルベース開発があり，制御ロジックを記述したモデル（制御モデル）をシミュレーションにより検証した後，ソースコードを自動生成することで開発者の負担を減らすことができる．しかしながら制御ロジック以外の，タスク構成やその優先度を考慮したソフトウェアの動作を記述したモデル（ソフトウェアモデル）の振る舞い検証は効率化できていない．

我々はソフトウェアモデルにおけるデータの整合性を，モデル検査ツール SPIN[1]を用いて検証するための環境を開発した[2][3]．具体的には，UML 記述されたソフトウェアモデルを入力し，SPIN 用のコード（PROMELA コード）を生成するツールを開発した．出力された PROMELA コードを SPIN に通すことでデータの整合性の検証を行う．

しかしながら，上記ツールはタスク内で発生するデータの不整合のみを対象としていたため，検証できる範囲に制約があった．そこで我々は，複数のタスク間の依存関係を考慮したデータの不整合を検出できるように拡張した．本論文では，その検証手法と，拡張した PROMELA コード自動生成ツールについて述べる．

2. 検証手法

依存関係が存在する複数のタスクから成るシステムの例を図 1 に示す．本システムは，データ A を算出する Task A，A を入力としてデータ B を算出する Task B，A を入力としてデータ C を算出する Task C，B と C を入力としてデータ D を算出する Task D から成る．優先度は Task A > Task B > Task C > Task D とする．

このシステムの動作例を図 2 に示す．Task B が起動し A の値を読み出した後に，より優先度の高い Task A が Task B をプリエンプトし，A の値を更新している．そして，Task B が B の値を更新して，終了した後，Task C が起動して A の値を読み出して C の値を更新している．最後に，Task D が起動し，C と B の値を読み出して D の値を更新している．図には A，B，C，D の値の更新回数も記してある．この場合，Task D において算出に使われる B と C の算出に使われた A の更新回数が異なっているため，値を正しく算出できていない．これがデータの不整合である．そこで，B と C の算出に使用した A の更新回数をそれぞれ保存し，B と C を使って D を算出するタイミングで，B と C の算出に使用した A の更新回数を比較すれば不整合を検出できる．そこでそれらの更新回数が常に一致することを検証するための PROMELA コードをツールで生成し，SPIN で検証する．

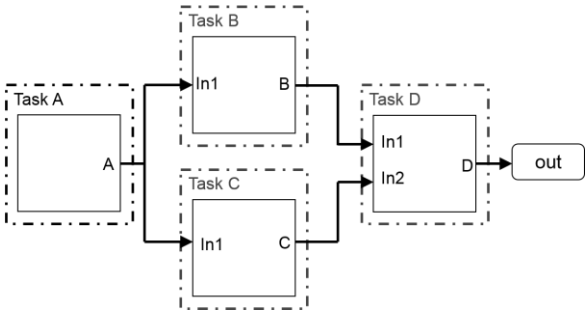

図 1 依存関係があるタスクから成るシステム

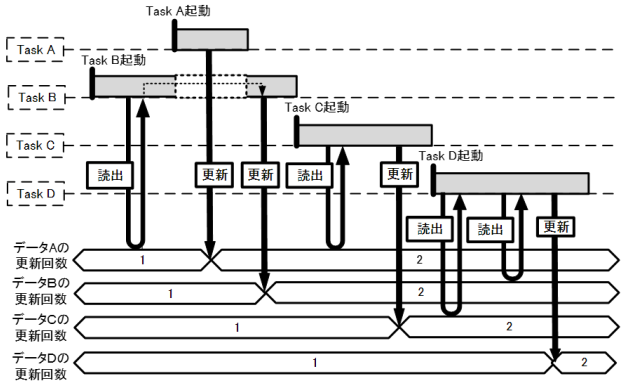

図 2 システムの動作例

3. PROMELA コード自動生成ツール

開発したツールの内部構成を図 3 に示す．本ツールは XMI 形式で保存された UML モデルを入力して解析し，検証用の PROMELA コードを出力する．UML モデルと PROMELA コードの対応を図 4 に示す．変換の際，不整合を起こす可能性のあるデータ間の依存関係を抽出して解析し，必要最小限のデータの更新回数の比較で検証できる PROMELA コードを出力する．

図 1 の例を用いてデータ間の依存関係の抽出方法を説明する．図 5 は図 1 のシステムのシーケンス図で，タスクごとに記述されている．Task A は操作

A Data Consistency Verification Environment for Embedded Control Software Development
† Tomoya Iwasaki, Myungryun Yoo, Takanori Yokoyama, Tokyo City University

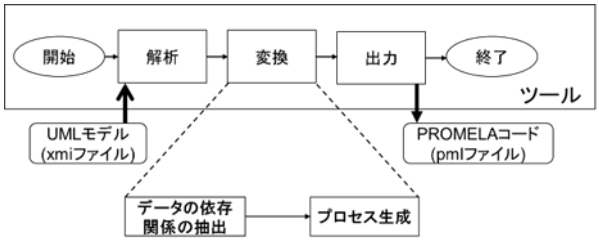

図 3　ツールの内部構造

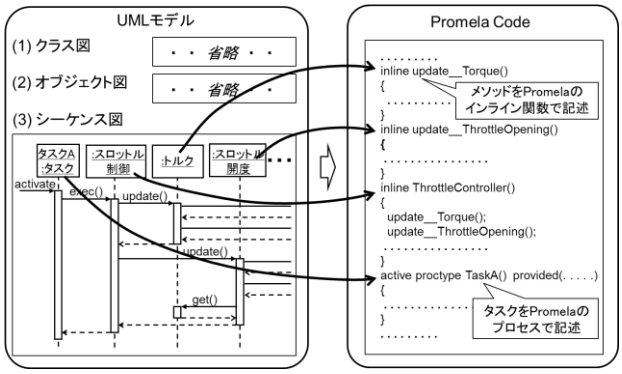

図 4　PROMELA コードへの変換

update で A の値を更新している．Task B は操作 update で A の操作 get を呼び出して算出に必要な A の値を取得し B の値を更新している．Task C は，Task B と同様に操作 update で A の操作 get を呼び出して必要な A の値を取得し，C の値を更新している．Task D は操作 update で B と C の操作 get を呼び出して B と C の値を取得し，D の値を更新している．

以上のシーケンス図中の操作 get の呼び出しを解析することで，データ間の依存関係を抽出する．

4. 評価

本研究で作成したツールの実用性を評価するため，実際の制御モデルを対象に PROMELA コードを生成し，SPIN で検証する実験を行っている．これまでに評価に用いたシステムの例として，ステッピングモータの Simulink モデルを図 6 に示す．この例での検証時間は 0.1 秒以下である．現在，より大規模なシステムを対象に評価実験を進めている．

5. おわりに

組込み制御ソフトウェアの開発効率向上を目的に，ソフトウェアモデルの振る舞いを検証する手法を提案するとともに，そのための PROMELA コードを自動生成するツールを開発した．

現在，本ツールがどの程度の規模のシステムまで対応できるか評価中である．

謝辞

本研究の一部は JSPS 研究費 JP15K00084 の助成を受けたものである．

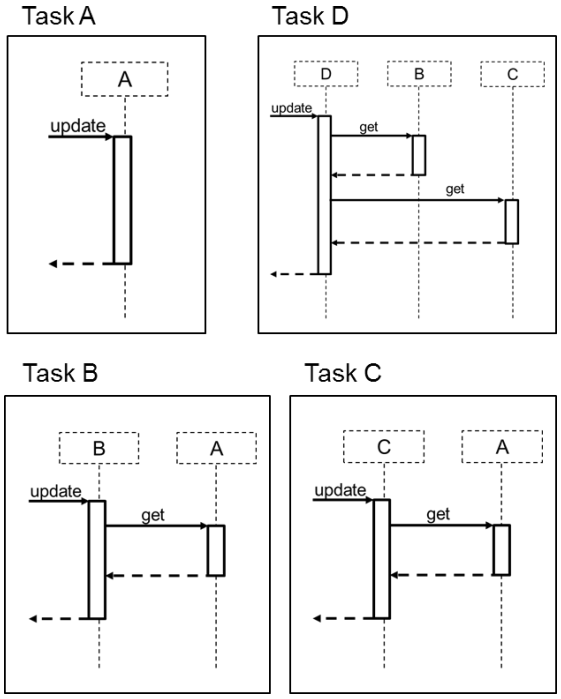

図 5 シーケンス図

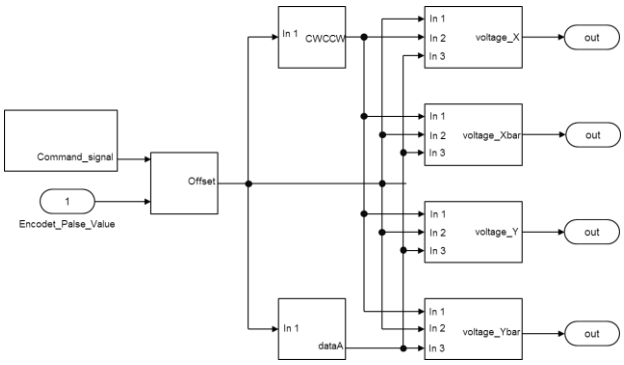

図 6 ステッピングモータ制御システム

参考文献

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

[2] 田村雅成，兪明連，横山孝典，モデル変換と振る舞い検証を活用した組み込み制御ソフトウェア設計法，情報処理学会研究報告，EMB，組込みシステム 2013-EMB-28, No.31, pp.1-6, 2013.

[3] Ito,T., Tamura,M., and Yokoyama,T., An Embedded Control Software Development Environment with Data Consistency Verification for Preemptive Multi-Task Systems, International Journal of Advances in Software Engineering & Research Methodology, Vol.2, No.2, pp.1-5, 2015.