

Title	リアルタイムJava検証のための検証支援環境に関する研究
Author(s)	曽我, 徹典
Citation	
Issue Date	2007-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/3611
Rights	
Description	Supervisor:片山 卓也, 情報科学研究科, 修士

リアルタイム Java 検証のための検証支援環境に関する研究

曽我 徹典 (510057)

北陸先端科学技術大学院大学 情報科学研究科

2007 年 2 月 8 日

キーワード: リアルタイム Java, 検証支援環境, パラメトリックモデル検査, プログラムストラクチャ.

本研究は、組込みソフトウェア開発の実行基盤となるリアルタイム Java に対して、プログラムの不具合を発見するための検証支援環境を提案する。

組込みソフトウェアにおいて、リアルタイム性の保証と資源の制約は高信頼性を持つために遵守されるべきである。しかし、開発期間が変わらないまま、肥大化していくソフトウェアに開発コストと開発者の負担は大きくなってきており、ソフトウェアの品質に影響を与えている。

近年、このようなソフトウェアの肥大化への対抗手段について、議論が盛んである。現在、このような問題を解決するために形式的手法が注目されている。その中でもモデル検査は必要な予備知識が比較的少なく、開発現場での適応が期待されている。また、リアルタイム性の保証と資源の制約は、プログラマが注意深く明示的にプログラムを記述するがある。この作業は、職人芸的な要素が強く、やはり開発コストとソフトウェアの品質の両面で大きな問題となっている。これらの作業を最大限に自動化し、プログラマによる指定を最低限に抑えるための実行基盤を用いることも対抗手段の一つである。

本研究の目的は、組込みソフトウェアに対するそのような取り組みをふまえ、組込みソフトウェアの開発に適応することができるための検証支援環境を提案し、その一部分を実験によって評価することである。本研究で提案する検証支援環境は、組込みソフトウェアがリアルタイム Java で開発されることを考えられており、作られたソフトウェアを開発者が大きな負担をかけることなく検証を行うことができるためのツールである。

本研究は、リアルタイム Java を検証するための検証支援環境の全体像を考え、最初の操作となるリアルタイム Java から検証に必要な状態遷移モデルを作りだすためのプログラムストラクチャを作りだすことができるかを確かめることである。ここでいうプログラムストラクチャとは、クラスファイルを解析してえられるバイトコード命令のうち、逐次実行をおこなう部分を区間ごとに分けた木構造である。この構造は、逐次実行される部分ごとにバイトコード命令が分割されている。これは、リアルタイム Java の実行環境での一つ

一つのバイトコード命令の実行時間を計ることで、プログラムストラクチャに時間を割り当てることができるようにするためである。

検証支援環境を用いた検証の全体の流れを説明する。

リアルタイム Java のクラスファイルを入力として、バイトコード処理ライブラリ BCEL を用いてプログラムストラクチャに変換する。リアルタイム Java の実行環境での一つ一つのバイトコード命令の実行時間を計った実行時間表を作成する。プログラムストラクチャと実行時間表を合わせることでプログラムストラクチャに時間を割り当てる。時間付きのプログラムストラクチャからパラメトリックタイムストラクチャを作る。パラメトリックタイムストラクチャとは、本研究で検証に用いるパラメトリックモデル検査の状態遷移モデルである。パラメトリックモデル検査は、時間を定数として決めておきたくない部分を変数としておき、検証した結果として時間の不等式を返す、モデル検査手法である。マルチスレッドのプログラムの場合、それぞれのスレッドでパラメトリックモデルストラクチャが作られるので、これを合成する。ここまでの操作でできたパラメトリックタイムストラクチャに、検証性質を書いて、パラメトリックモデル検査ツールで検証を行う。パラメトリックモデル検査での検証によって時間の不等式をえられる。その不等式から要求に合う適切な値を決めてプログラムを修正する。

このような検証支援環境ならば、開発者は設計・分析の段階を変更することなく検証を行うことができるようになる。

本研究では、提案した検証支援環境のうち、リアルタイム Java のクラスファイルからプログラムストラクチャを作りだすツールを実験的に実装した。

ツールは、リアルタイム Java に対して部分的な解析しかできなかったため、本研究では通常の Java のクラスファイルを解析の対象とした。実験の結果として、このツールによって、BCEL を使用して Java のクラスファイルを解析して、プログラムストラクチャを作りだすことができる、とわかった。これは、この実験に限って、検証支援環境を作るための最初の操作を行うことができるという意味である。また、プログラムストラクチャを作る上で、問題となる Native メソッドや Abstract メソッドなどを発見することができた。これらの問題点については解決策を考察した。本研究で作りだしたプログラムストラクチャは、パラメトリックタイムストラクチャを作りだすためには不完全である。このため、検証を行うためには、パラメトリックタイムストラクチャを作りだすことのできるプログラムストラクチャを作りだす必要がある。