様式 C-19

科学研究費補助金研究成果報告書

平成２２年６月７日現在

研究種目：若手研究（Ｂ）
研究期間：2007～2009
課題番号：19700016
研究課題名（和文）　信頼性の高いソフトウェア開発に向けた「モデル-プログラム協調環境」の構築
研究課題名（英文）　Construction of "Model program cooperation environment" for reliable software development

研究代表者
田辺　誠（TANABE MAKOTO）
独立行政法人国立高等専門学校機構宇部工業高等専門学校・制御情報工学科・准教授
研究者番号：00353318

研究成果の概要（和文）：
モデル検査ツール UPPAAL によって安全性および信頼性の保証されたモデルから、これらの性質を保ったまま JAVA プログラムのソースコードを生成する技術を開発し、二種類の変換ツールを開発した。一方は JML(Java Modeling Language)記述の付加された JAVA ソースコード、もう一方はStateパターンによってモデル上の状態遷移を再現する JAVA ソースコードをそれぞれ生成するものである。
本研究の変換技術をチケット予約システムの設計および開発に適用することにより、本技術の有用性を検証した。

研究成果の概要（英文）：
A transformational technology is developed from models of a model-checking tool UPPAAL to JAVA program source codes that keep the properties guaranteed by model-checking process.
Two kinds of translation tools are developed. One generates JAVA source codes with JML (Java Modeling Language) description, and the other generates JAVA source codes where the state transition on the given model is reproduced by the corresponding state pattern description.
The effectiveness of this technology is verified by applying the transformational technology to the design and development of a ticket reservation system.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
２００７年度	900,000	0	900,000
２００８年度	700,000	210,000	910,000
２００９年度	900,000	270,000	1,170,000
総　計	2,500,000	480,000	2,980,000

研究分野：総合領域
科研費の分科・細目：情報学・情報学基礎
キーワード:プログラム理論、ソフトウェア検証、モデル検査、UPPAAL、Java Modeling Language, State パターン

１．研究開始当初の背景

ソフトウェアの規模が大きくなるにつれ、ソフトウェアの安全性及び信頼性に対する要求が高まってきている。これらの要求に応えるための技術としてモデル検査による形式的な検証技術が開発されてきた。

モデル検査とは、ある計算モデル上の状態遷移系として記述されたシステムの設計図が、時相論理などの形式的な言語によって与えられた仕様を満足するかどうかを網羅的に検証する技術である。しかし、ソフトウェア開発の現場にモデル検査の手法を取り入れる際、以下の欠点が導入の障害となる。

- 人的コスト：モデル検査終了後、モデルを参考にしながらプログラムを人手で作成する必要がある。
- 安全性の保証：モデルからプログラムへの変換に人的ミスが介在する恐れがあり、モデル上で保証された安全性が変換後のプログラムにおいても保証されるとは限らない。

これらにより、実際の開発現場において十分にモデル検査技術が活用されているとは言い難い状況であった。

２．研究の目的

本研究の目的は、モデルとプログラムの相互変換を自動的に行う技術を確立することである。具体的には、モデル検査によって安全性の保証されたモデルからプログラムの一部分を自動生成する技術の確立を目指した。この変換技術によるソフトウェア設計および開発のイメージを図１に示す。

これによって、上に述べた問題点が以下のように解決されると考える。

- 人的コスト:プログラムの自動生成により、モデルを参照しながらプログラムを実装する手間が軽減される。
- 安全性の保証:モデル上で確保された安全性を保ったまま変換がおこなわれるため、プログラムの安全性向上につながる。

３．研究の方法

研究にあたっては、研究の対象となるモデル検査ツールおよびプログラミング言語を選定した後、モデルからプログラムソースコードへの変換ツールを作成した。ツールの有用性を検証するため、システム設計・開発への有用性を検証した。

(1)モデル検査ツールおよびプログラミング言語の選定

研究の遂行にあたっては、モデル検査ツールとしてUPPAALを、また、プログラミング言語としてJAVAを以下の理由により採用した。

- モデル検査ツールUPPAALはスウェーデンのUppsala大学とデンマークのAalborg大学の共同プロジェクトにより開発されたツールであり、研究目的で自由に利用ができる。また、作成されたモデルがXML形式で保存されるため、既存のデータ変換技術を適用しやすい。
- JAVAプログラムのソースコード上に「契約による設計」条件を記述し、プログラムの安全性を高めるモデリング言語として、JML(Java Modeling Language)がある。UPPAALからソースコードを生成する際、JMLの条件記述も生成することにより、モデル上の遷移条件とソースコード上のメソッド実行条件とを対応づけることを目指した。

(2)モデルからプログラムソースコードへの変換ツールの作成

UPPAALのモデルからJAVAソースコードを生成するツールを作成した。作成にあたっては、UPPAALモデルの動作をプログラムに対応付けることができるように変換の設計を注意深く行った。

(3) 変換ツールのソフトウェア設計・開発への応用

(2)で作成した変換ツールをソフトウェア設計・」開発に適用することにより有用性の検証を目指した。

本ツールの作成により、ソフトウェアの設計・開発者はモデル検査ツール上で設計の妥当性（安全性および信頼性）を検証後、変換ツールを利用することによりJAVAプログラムのソースコードを得ることができる。このプログラムにソフトウェアの詳細部分を実装することにより、安全性を保ったままソフトウェアの開発を行うことができる。

４．研究成果

(1)UPPAALモデルからJAVAソースコードへの変換ツールの作成

本研究では、モデル検査ツール UPPAAL のモデルから JAVA ソースコードへの変換ツールを二種類作成した。

① JAVAソースコードおよびJML記述を生成するツール

UPPAAL モデル上の状態遷移に対して、JAVAメソッド呼び出しを対応づける変換規則を定め、この規則に基づいた変換ツールを作成した(表 1)。モデル上の遷移の事前条件および事後条件が JML 記述に変換されるため、JML の条件チェックを行うことにより実装の安全性を確かめることができる（表1)。

② Stateパターンに基づくJAVAソースコードを生成するツール

JML が JAVA1.4 以前の API にしか対応していないため、JML を用いると最新の JAVA API での実装ができないという問題

点があった。また、実装者に JML に関する知識を要求するため、実装者の負担が大きいという問題も生じた。そこで、UPPAAL 上の状態遷移を State パターンを用いて JAVA 上に対応づけることにより、この問題を解決した（表 2)。

(2)チケット予約システムの設計・開発による有用性検証

チケット予約・発券システムの設計および開発を例題として、変換ツールの有用性を検証した。

変換ツールを用いた設計・開発の手順は以下のとおりである。

・**設計：** UPPAAL を用いてシステムの設計を行う。システムのモジュールごとにテンプレート（状態遷移図）を作成し、システムの安全性や信頼性など、システムに要求される性質を時相論理 CTL 式を用いて記述する。

・**検証：** システムの安全性検証を行う。UPPAAL のモデル検査機能を用い、設計されたシステムが要求される性質を満たすかどうか検査する。

・**変換：** 本研究で開発された変換ツールを用いて JAVA ソースコードを生成する。ソースコードはモデル内の状態遷移図と同じ動作を行うため、UPPAAL 上で検証された性質を満たす。

・**実装：** 生成されたソースコード内のメソッドの詳細部分を実装し、システムを完成する。

参考文献[1]に挙げられた航空券の予約システムに対し、この手順に沿って設計および開発を行った。

① システムの設計を UPPAAL(Ver4.0)を用いて行った。システムの主要コンポーネントである旅行者、航空会社、およびエージェント(予約システム)のそれぞれを、UPPAAL のテンプレート（拡張された状態遷移図）としてモデル化を行った。また、システムに要求される安全性および応答性を CTL によって記述し、UPPAAL の property として設定を行った。モデルが property を満たすかどうか UPPAAL のモデル検査によってチェックを行った結果、[1]に挙げられたモデルの通りに設計を行うとデッドロック等の不具合を起こすことが確認できた。モデルを修正することにより、デッドロックフリー等の安全性およびシステムの応答性などの仕様が満たされるモデルの設計に成功した。

② ①で得られた UPPAAL モデルから、本研究で作成した二種類の変換ツールを用いて JAVA ソースコードを生成した。ソースコード内のメソッドにデータベースへのアクセスなどの詳細を実装することにより プログラムの開発を完了した。

JAVA ソースコードおよび JML 記述を生成するツールについては、JML チェッカーが JAVA の 1.4 版以下にしか対応していないため、J2SE1.4.2 に準拠した実装を行い、JMLSPECS5.6 版にて動作の確認を行った。また、JAVA の State パターンに基づいたソースコードを生成するツールにおいては、J2SE6 に準拠した実装を行った。

(3) 考察および今後の展望

実装されたプログラムについては部分的正当性が保証された。すなわち、モデル化の対象となる抽象度に対応する動作に関しては、自動生成されたプログラムは正当性を保つ。しかしながら、例えばデータベース呼び出しに対する応答が返ってこないなど、そもそもモデル作成による設計時には記述の対象外であった部分に関する応答性は保証できない。したがって、プログラムの動作をモデル側の状態遷移として観察した場合、状態遷移系との対応はとれているが、状態遷移そのものが途中で止まる可能性がある場合が確認された。全体的正当性の確保については今後の研究課題である。

また、プログラムからモデルへの逆変換技術については今後の課題である。

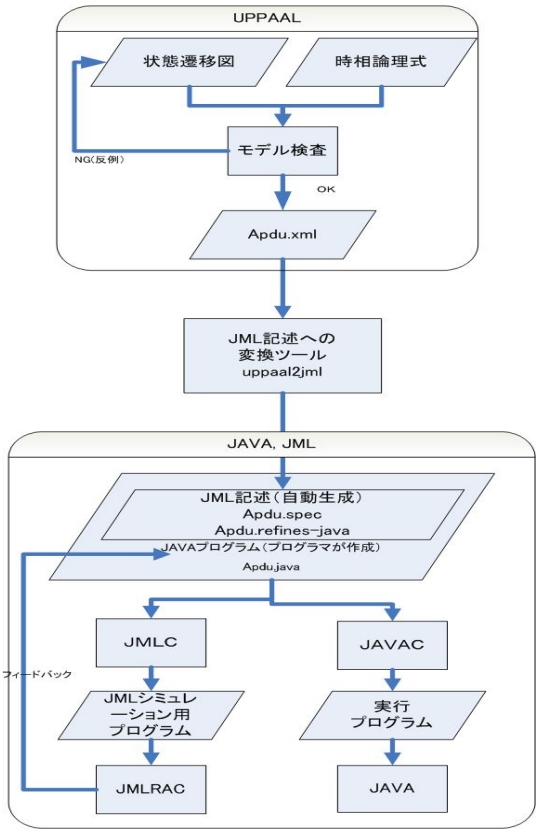

図 1 変換ツールを用いた設計開発イメージ

表 1:モデルとプログラムの対応付け(1)

UPPAAL モデル	Java ソースコード+ JML
グローバル変数 V	UPPAAL クラスのメンバ変数 V
テンプレート T	クラス T
テンプレート T 内の状態 S	クラスTのインスタンス変数「state」が取る値 S
テンプレート T の遷移 t	クラスTのメソッド t
遷移 t の条件 c	メソッドt呼び出しの事前条件(JML の require 節 c)
遷移 t の処理 a	メソッドt実行の事後条件（JML の ensures 節 a)
遷移 t1 と遷移 t2 の同期	メソッド t1 からメソッド t2 への呼び出し
テンプレート T のローカル変数 V	クラスTのインスタンス変数 V

表 2:モデルとプログラムの対応付け(2)

UPPAAL モデル	Java ソースコード（State パターン)
グローバル変数 V	UPPAAL クラスの静的変数 V
テンプレート T	テンプレートインタフェイスを実装したクラス T（テンプレートインターフェイスにはステートパターン実装用のメソッドが宣言されている)
テンプレート T 内の状態 S	状態インターフェイスを実装したクラス S （状態インターフェイスには、状態遷移に応じて呼び出される メソッド enter, exit, execute が定義されている)
状態S1から状態S2への遷移 t	クラス S1 内の、遷移クラスを継承したクラス「S1toS2」
遷移 t の遷移条件 c	対応する遷移クラス内のメソッド「canTransition」
遷移 t の事後処理 a	対応する遷移クラス内のメソッド「transition」内に記述。
遷移 t1 と遷移 t2 の同期	同期クラスを継承したクラスを用い、t1 と t2 に対応する遷移クラス内のメソッド「transition」内にそれぞれ記述。
テンプレート T のローカル変数 V	クラス T のインスタンス変数 V

参考文献
[1] Gregorio Diaz, Juan-Jose Pardo, Maria-Emilia Cambronero, Valentin Valero, Fernando Cuartero, “Verification of Web Services with Timed Automata”, Electronic Notes in Theoretical Computer Science, Volume 157, 2006, Pages 19-34.

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計０件）

〔学会発表〕（計０件）

６．研究組織
(1)研究代表者
田辺　誠（TANABE MAKOTO）
独立行政法人国立高等専門学校機構宇部工業高等専門学校・制御情報工学科・准教授
研究者番号：00353318