

令和 3 年 6 月 28 日現在

機関番号：92628

研究種目：若手研究

研究期間：2018～2020

課題番号：18K18033

研究課題名（和文）モデル規範型形式手法を支援するバーチャルマシンの研究

研究課題名（英文）Research on Virtual Machine to Support Model-based Formal Methods

研究代表者

小田 朋宏（Oda, Tomohiro）

株式会社SRA（先端技術研究所）・先端技術研究所・研究員

研究者番号：00580383

交付決定額（研究期間全体）：（直接経費） 2,600,000円

研究成果の概要（和文）：実行可能形式仕様記述言語VDM-SLの実行を主目的とするバーチャルマシンを、データモデル、メモリモデル、レジスタモデル、バイトコードのインストラクションセット、実行コンテキスト（アクティベーション・レコード）およびコード領域の形式仕様をVDM-SLにより記述し、C言語による実装でx64プロセッサおよびarm32およびarm64プロセッサの3種類のプロセッサ向けにビルドし、実行させた。VDM-SL仕様とC実装の両方に同じユニットテストを適用することで、C言語のユニットテストでのエラーの原因が仕様レベルであるか実装レベルであるかを容易に判別することができ、C言語による実装の生産性が良好だった。

研究成果の学術的意義や社会的意義

現在多くのプログラミング言語処理系がバーチャルマシンを利用している。バーチャルマシンの開発には多くの職人芸的な高度な設計実装の技術が求められている。本研究は、バーチャルマシンの開発に形式仕様を導入することで、バーチャルマシンの開発の敷居を下げることを目標にしている。

研究成果の概要（英文）：A virtual machine to execute VDM-SL specification was developed using VDM-SL. Data models, memory models, register models, instruction set, activation record, and code pages are specified and implemented. The implementation was written in C and compiled into native binaries for x64, arm32 and arm64. By porting unit tests written in VDM-SL into C, the implementation in C went very productive because the same unit tests were applied to the both VDM-SL and C, and it made very easy to identify the cause of the failure.

研究分野：ソフトウェア工学

キーワード：形式手法 実行可能仕様 バーチャルマシン

## 1. 研究開始当初の背景

システム開発では、各アプリケーションドメインに特化したドメイン特化言語(Domain Specific Language, DSL)を始め、多くの独自言語およびその処理系の開発が求められることが多い。言語処理系の実現において、構文論に関する部分はパーサジェネレータなどツールの支援が手厚く提供されている。しかし、意味論に関する部分はツール支援が十分とはいえない。

現在、少なくないプログラミング言語処理系がバーチャルマシンアーキテクチャを採用している。JavaVM など、多くのハードウェアアーキテクチャ上で同一の実行環境を提供し、かつ、バーチャルマシンの命令セットを公開することで、多言語の実行ターゲットとして利用可能になっている。一方で、バーチャルマシンの開発では処理速度、信頼性、メンテナンス性など多くの特性を高度に両立させる必要があり、その設計および実行には高度な専門技術が要求される。

バーチャルマシンの設計および実装を、職人芸ではなくシステムチックな手法に基づいて行うことで、独自のバーチャルマシンの設計および実装の敷居が下げられることが期待されている。

## 2. 研究の目的

形式仕様記述言語 VDM-SL は形式手法の中でも歴史が長く、国内でもモバイル FeliCA チップの開発に適用されるなど成功事例がある。VDM-SL の大きな特徴として、実行可能なサブセットの存在が挙げられる。VDM-SL のほとんどの言語機能は、プログラミング言語同様にインタプリタ実行が可能であり、仕様記述に対してデバッガによるステップ実行やユニットテストにより誤りを発見し修正することができる。

本研究では、バーチャルマシンの開発に形式仕様を適用することで、信頼性およびメンテナンス性を確保しつつ、実用になりうる処理速度を目指した。また、バーチャルマシンは、VDM-SL を実行することを主目的とした。

## 3. 研究の方法

本研究では、実行可能形式仕様記述言語 VDM-SL の実行を主目的とするバーチャルマシンを、データモデル、メモリモデル、レジスタモデル、バイトコードのインストラクションセット、実行コンテキスト(アクティベーション・レコード)およびコード領域の形式仕様を VDM-SL により記述し、C 言語による実装で x64 プロセッサおよび arm32 および arm64 プロセッサの 3 種類のプロセッサ向けにビルドし、実行させた。

VDM-SL 仕様の記述は、仕様記述環境 ViennaTalk 上で行った。ViennaTalk はユニットテストなど、VDM-SL 仕様の実行を中心にした、仕様との柔軟な対話性を重視した仕様記述環境である。まずは、データモデル、メモリモデル、レジスタモデルをそれぞれ VDM-SL のモジュールとして実行可能な形式で記述し、それぞれのモジュール毎に多くのユニットテストを定義し、テスト駆動開発による形式仕様記述を行った。十分に信頼できるデータモデル、メモリモデル、レジスタモデルを確保した上で、アクティベーションレコード、コード領域、および命令セットを同じくテスト駆動開発のスタイルによって定義した。ユニットテストには、メモリからレジスタへのデータ転送のような細粒度な処理から、再帰関数の評価実行を行うプログラムをバーチャルマシン上の命令セットで実現し動作させる大きな粒度のものまで、さまざまな粒度のテストを記述した。

VDM-SL 仕様で全ての命令セットを記述した後で、VDM-SL による記述を C 言語へ移植した。仕様で定義された各機能を移植するごとに、仕様に対するユニットテストも C 言語に移植して、十分なユニットテストによって信頼性を確保した。

## 4. 研究成果

本研究では、VDM-SL 仕様を記述して、その記述を元に、手で C 言語による実装を行った[1]。VDM-SL 仕様の記述にテスト駆動開発が有効であることが確認できた[2]。ViennaTalk に、仕様またはユニットテストのどちらかが更新されるたびに全テストケースをバックグラウンドで実行し、テスト失敗があれば直ちに通知する機能を実装した。結果、仕様の更新によるリグレッションを素早く検知して、誤りの除去を行うことができた。

VDM-SL 仕様と C 実装の両方に同じユニットテストを適用することで、C 言語のユニットテストでのエラーの原因が仕様レベルであるか実装レベルであるかを容易に判別することができ、C 言語による実装の生産性が良好だった。実装されたバーチャルマシンのパフォーマンスを、フィボナッチ数の計算をベンチマークとして計測して、Python3 を上回る結果を得た[2]。実装されたプログラムは、x64/arm32/arm64 の各プロセッサをターゲットにコンパイルして、実行した。

さらに、ターゲット言語である VDM-SL の適用対象を広げることを目的に、VDM-SL によるグラフィカルなユーザインターフェイスの記述を行う ViennaVisuals[5]や、HTML 中に VDM-SL 仕様を埋め込んでウェブブラウザ上で VDM-SL 仕様を実行させる仕組み ViennaDoc[3]を開発した。また、バーチャルマシンの仕様記述で得られた経験から、仕様記述における試行錯誤の履歴を集めて振り返りを行うツール[4]を開発した。これらのツールは現在のところまだ ViennaVM による実行には至っていないが、将来的に実行可能にする予定である。

- [1] Tomohiro Oda, Keijiro Araki, and Peter Gorm Larsen, “ViennaVM: a Virtual Machine for VDM-SL development, in Proceedings of Overture 16, pp.39-53, 2018.
- [2] 小田朋宏, 荒木啓二郎, “軽量形式手法 VDM によるバーチャルマシンの開発”, ソフトウェアシンポジウム 2019 論文集, pp. 46-55, 2019. (最優秀論文賞)
- [3] Tomohiro Oda, Keijiro Araki, Yasuhiro Yamamoto, Kumiyo Nakakoji, Hiroshi Sako, Han-Myung Chang, and Peter Gorm Larsen, “ViennaDoc: An Animatable and Testable Specification Documentation Tool”, in Proceedings of Overture 17, pp. 289-302, 2019.
- [4] 小田朋宏, 張漢明, 山本恭裕, 中小路久美代, 荒木啓二郎, “探索的仕様記述のための履歴ツールの提案と実装, ソフトウェアシンポジウム 2020 論文集, pp. 95-103, 2020.
- [5] Tomohiro Oda, Keijiro Araki, Yasuhiro Yamamoto, Kumiyo Nakakoji, Han-Myung Chang, and Peter Gorm Larsen, “Specifying Abstract User Interface in VDM-SL”, in Proceedings of Overture 18, pp. 5-19, 2020.

5. 主な発表論文等

〔雑誌論文〕 計0件

〔学会発表〕 計5件（うち招待講演 0件 / うち国際学会 3件）

1. 発表者名 小田朋宏, 荒木啓二郎
2. 発表標題 軽量形式手法VDMによるバーチャルマシンの開発
3. 学会等名 ソフトウェアシンポジウム2019
4. 発表年 2019年

1. 発表者名 omohiro Oda, Keijiro Araki, Yasuhiro Yamamoto, Kumiyo Nakakoji, Hiroshi Sako, Han-Myung Chang, Peter Gorm Larsen
2. 発表標題 ViennaDoc: An Animatable and Testable Specification Documentation Tool
3. 学会等名 The 17th Overture Workshop (国際学会)
4. 発表年 2019年

1. 発表者名 小田朋宏, 張漢明, 山本恭裕, 中小路久美代, 荒木啓二郎
2. 発表標題 探索的仕様記述のための履歴ツールの提案と実装
3. 学会等名 ソフトウェアシンポジウム2020
4. 発表年 2020年

1. 発表者名 Tomohiro Oda, Keijiro Araki, Peter Gorm Larsen
2. 発表標題 ViennaVM: a Virtual Machine for VDM-SL development
3. 学会等名 The 16th Overture Workshop (国際学会)
4. 発表年 2018年

1. 発表者名 Tomohiro Oda, Keijiro Araki, Yasuhiro Yamamoto, Kuniyo Nakakoji, Han-Myung Chang, Peter Gorm Larsen
2. 発表標題 Specifying Abstract User Interface in VDM-SL
3. 学会等名 The 18th Overture Workshop (国際学会)
4. 発表年 2020年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

<p>GitHub - tomooda/ViennaVM  <a href="https://github.com/tomooda/ViennaVM/">https://github.com/tomooda/ViennaVM/</a></p> <p>ViennaTalk  <a href="https://viennatalk.org/">https://viennatalk.org/</a></p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6. 研究組織		
氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------