

科学研究費助成事業 研究成果報告書

平成 29 年 6 月 26 日現在

機関番号：92628

研究種目：基盤研究(C) (一般)

研究期間：2014～2016

課題番号：26330099

研究課題名(和文) プログラミング教育向けビジュアルプログラミング環境へのアサーションの導入

研究課題名(英文) Introducing Assertions to Visual Programming Environment for Introductory Programming Education

研究代表者

小田 朋宏 (Oda, Tomohiro)

株式会社 S R A (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) 2,100,000円

研究成果の概要(和文)：ブロック式ビジュアルプログラミング環境は、プログラミング言語の構文規則やキーワードを暗記することなくプログラミングを体験することができることから、小学校から大学まで広くプログラミング入門に利用されている。本研究では、ブロック式ビジュアルプログラミング環境「Phratch」にアサーション機能を追加するソフトウェアを開発し、一般に公開した。アサーションは処理単位に対する事前条件および事後条件を定義したものであり、ソフトウェアの機能仕様、設計およびテストを系統的に行うことを可能にする。実際の大学学部レベルのプログラミング教育に「Assertch」を導入し、その効果を評価した。

研究成果の概要(英文)：Block-based visual programming environments are used for introductory programming classes at various levels of schools from elementary schools to colleges. Those visual languages allows learners to start programming without memorizing syntax rules and keywords. In this research, we developed a software that extends assertion blocks to a visual programming environment called "Phratch", and published it as an open source software. Assertions are preconditions and postconditions of functional units in programs, and enables systematic ways to specify, design and debug program codes. We also applied our software to an introductory programming class for undergraduate students and evaluated its effects.

研究分野：形式手法

キーワード：プログラミング教育 形式手法 ビジュアルプログラミング 契約プログラミング

1. 研究開始当初の背景

ICT教育の重要性への認識が高まるにつれて、小中学生を対象にしたプログラミング教育をおこなう学校やワークショップが広がりつつある。こうしたプログラミング教育では、キーボードへの依存が少なく、また、直感的な理解が可能であることから、プログラムを図形的に表現するビジュアルプログラミング環境が多く用いられる。

初等教育に導入されているビジュアルプロ

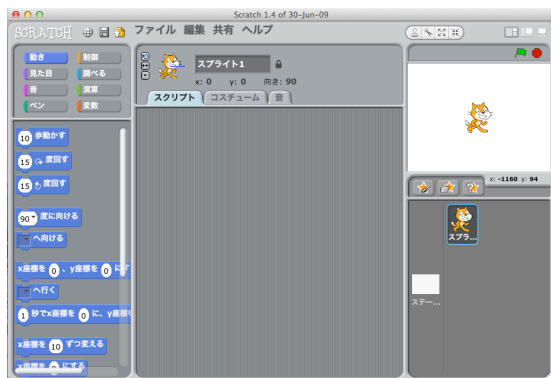


図 1: ビジュアルプログラミング環境 Scratch

グラミング環境の例として、米 MIT がプログラミングリテラシーを学習する環境として子供向けに開発した Scratch (図 1) が挙げられる。[1] Scratch は、ブロックと呼ばれるジグソーパズルのピースのような形をした部品を連結して並べることでプログラムを記述する LOGO ベースのビジュアルプログラミング環境である。日本国内では OtOMO, CANVAS などのグループが小中学校、高等学校に導入を進めている。大学においても、Scratch を導入した演習が報告されている。[2]

2. 研究の目的

Scratch は、非常に対話性が高く直感的なプログラミング環境を提供しており、プログラムの変更とその効果を視覚的に理解することを可能にしている。しかしながら、Scratch によるプログラミングでは、学習者がプログラミングを試行錯誤による発生的なプロセスとして理解してしまう危険をはらんでいる。すなわち、学習者が様々なブロックの組み合わせやパラメータを次々とて飯結果的に見栄えのする、自分の気に入った振る舞いを作ったらそれを残す、という方法でプログラムを記述している様子が観察されている。

我々は、プログラミングをリテラシーとして学習するためには、学習者が実現したいプログラムを合理的にデザインすることを学ぶ必要があると考える。本提案研究では、ビジュアルプログラミング環境 Scratch をベースに、アサーションブロックを追加し、プログラミングによって何を実現したいのか、配置したブロックについてそのブロックを選んだ理由は何か、といった、ソフトウェアの仕

様記述とデザインインテントをアサーションブロックを用いて記述し、それらをプログラムに結びつけるための仕組みを開発する。本提案研究では、アサーションブロックとして事前・事後条件ブロックと、普遍条件ブロックを導入する。アサーションブロックは、Scratch の制御構造ブロックと同様に、ネスとする形で他のブロックを任意の個数連結して保持する。また、ビジュアルプログラミング環境を拡張し、プログラム実行時にアサーションブロックが指定する事前条件、事後条件、普遍条件が満たされているかどうかを実行時検査し、違反を通知する仕組みを実装する。

アサーションブロックを導入することで、ビジュアルプログラミングに以下の概念が導入される。

- 仕様記述: 学習者が記述しているプログラムが何であるかを、ブロックの連結配置という実装やプログラムの実行時の振る舞いそのものでなく、プログラムの事前条件、事後条件、および普遍条件という形で記述することを促す。
- 合理的なデザイン: 事前・事後条件ブロックの内容となるブロック列をれケツ配置する際、どのようにして事前条件からブロックの実行終了時に事後条件を実現するのかを意識してブロックを選択することを促す。また、普遍条件を破らないためにはないが必要で何をしてはいけないかを意識してブロックを選択することを促す。
- テスト: プログラムの実行時に事前条件、事後条件、普遍条件をテストすることを促す。プログラムの振る舞いが事前に意図した通りであるかどうかを確認することができる。
- 検証: プログラムの各ブロックを個別に事前・事後条件ブロックで囲むことで、あるブロックの事前条件が普遍条件および直前のブロックの事後条件から導出を試みることで部分的正当性を検証することができる。

上記はいずれもソフトウェア工学において重要なトピックである。試行錯誤になりがちなビジュアルプログラミング環境に、2種類のアサーションブロックを新たに導入することで、ソフトウェア工学のエッセンスを学ぶ機会を提供することができる。

例えば、学習者がタートルを U ターンさせたいとする。学習者はまず [10 歩すすむ][10°右にまわる]の2つのブロックを[18 回くりかえす]制御ブロックで包むと、タートルが U ターンすることを経験的に学習する。アサーションブロックなしでは U ターンであることはどこにも記述されない。アサーションブロックを使って、例えば U ターンを描くブロック列を不変条件ブロックと事前・事後条件ブロックで包み、不変条件として「上の壁に触れていない」、事前条

件として「現在位置は上の壁から 100 歩以上離れている」、事後条件として「現在の向きは元の向きから 180°回転している」ことを記述することで、U ターンの目的である「壁に当たらずに 180°回転させる」ことを表現し、実行時にその目的が果たされているかテストすることができる。また、U ターンの仕様として壁を避けるためには壁にあたる直前ではなく回転のための余裕をもって U ターンを開始する必要があることを示すことができる。(図 2 参照)



図 2 : アサーションの例

3. 研究の方法

Scratch を拡張してビジュアルプログラミング環境にアサーションブロックおよび条件記述に必要な各種ブロックを導入する。ビジュアルプログラミングにおいて適切なアサーション結果の通知方法を設計し、ビジュアルプログラミング環境に実装する。アサーションブロックとして事前-事後条件ブロックおよび不変条件ブロックを実装する。小中学生向けのワークショップ、大学でのプログラミング導入教育、ソフトウェア工学演習において実際に学習者に使ってもらい、その様子を観察し、教育者からのフィードバックおよび作成されたプログラムを分析してアサーションブロックの効果を分析する。

4. 研究成果

Scratch と同様な機能を持つビジュアルプログラミング環境 Phratch にアサーションブロックを追加するアドオンとして Assertch を開発し、オープンソースソフトウェアとして一般に向けて公開した。[3]

開発した Assertch を山梨学院大学経営情報学部の講義に適用し、演習などを行い、評価した。[4]

ビジュアルプログラミング環境が提供するライブプログラミングの機能に着目し、テキストベースの仕様記述言語である VDM-SL の環境をライブプログラミング環境上に構築することで、仕様記述の初期段階(探索的仕様記述)を支援するための理論的枠組みと設計指針および具体的な実装を示した。開発された VDM-SL 記述環境 ViennaTalk をオープンソースソフトウェアとして一般[3][5][6]に公開した。また、プログラミングと仕様記述の関係をより明確にするために、VDM-SL で記述された仕様から Assertch および

ViennaTalk の記述言語である Smalltalk 言語のプログラムに自動変換する機能を実現した。[7][8]これにより、ビジュアルプログラミング環境でのプログラミング導入教育から本格的な形式仕様記述言語および実装までを 1 つの繋がりを持つ技術として、1 つの環境の上で実現された。

[1] Karen Brennan and Mitchel Resnick, "New frameworks for studying and assessing the development of computational thinking", *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*. 2012.

[2] 伊藤一成, "プログラミング, 何をどう教えているか : Scratch を用いた授業実践報告", *情報処理学会誌*, Vol.52, No.1, pp.111-113 (2011.01)

[3] Tomohiro Oda, Keijiro Araki and Peter Gorm Larsen, ViennaTalk and Assertch: Building Lightweight Formal Methods Environments on Pharo 4, in Proc. Of International Workshop on Smalltalk Technology 2016.

[4] ウェア開発発注者育成のための形式手法を取り入れたプログラミング教育, *ソフトウェアシンポジウム 2017 論文集*, 2017.

[5] Tomohiro Oda, Keijiro Araki and Peter Gorm Larsen, A Formal Modeling Tool for Exploratory Modeling in Software Development, *IEICE TRANSACTIONS on Information Systems*, Vol. E-100D, 2017.

[6] 小田朋宏, 荒木啓二郎, 形式仕様工程の初期段階に着目した統合仕様記述環境 ViennaTalk, *コンピュータソフトウェア (accepted)*

[7] Smalltalk プログラムの自動生成, *ソフトウェアシンポジウム 2016 論文集*, 2016.

[8] Tomohiro Oda, Keijiro Araki and Peter Gorm Larsen, Automated VDM-SL to Smalltalk Code Generators for Exploratory Modeling, in Proc of the fourteenth Overture Workshop: Towards Analytical Tool Chains, 2016.

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 2 件)

Tomohiro Oda, Keijiro Araki and Peter Gorm Larsen, A Formal Modeling Tool for Exploratory Modeling in Software Development, *IEICE TRANSACTIONS on Information Systems*, Vol. E-100D, 2017.

小田朋宏, 荒木啓二郎, 形式仕様工程の初期段階に着目した統合仕様記述環境 ViennaTalk, *コンピュータソフトウェア (採録確定)*

[学会発表] (計 4 件)

小田朋宏, 荒木啓二郎, VDM-SL 仕様からの Smalltalk プログラムの自動生成, *ソフトウェアシンポジウム 2016 論文集*, 2016

Tomohiro Oda, Keijiro Araki and Peter Gorm Larsen, ViennaTalk and Assertch: Building Lightweight Formal Methods Environments

on Pharo 4, in Proc. of International Workshop on Smalltalk Technology 2016.
Tomohiro Oda, Keijiro Araki and Peter Gorm Larsen, Automated VDM-SL to Smalltalk Code Generators for Exploratory Modeling, in Proc of the fourteenth Overture Workshop: Towards Analytical Tool Chains, 2016.
伊藤栄一郎, 小田朋宏, 荒木啓二郎, ソフトウェア開発発注者育成のための形式手法を取り入れたプログラミング教育, ソフトウェアシンポジウム 2017 論文集, 2017.

〔図書〕 (計 0 件)

〔産業財産権〕

○出願状況 (計 0 件)

○取得状況 (計 0 件)

〔その他〕

ホームページ等

<https://www.sra.co.jp/ktl/assertch/index.html>

6. 研究組織

(1) 研究代表者

小田 朋宏 (ODA, Tomohiro)
株式会社 SRA・先端技術研究所・研究員
研究者番号：00580383

(2) 研究分担者

()

研究者番号：

(3) 連携研究者

()

研究者番号：

(4) 研究協力者

()