

自動翻訳とオブジェクト指向リエンジニアリング

2018SE055 長岡 楓己

指導教員：横山 哲郎

1 はじめに

1.1 背景

今日、様々な物がデジタル化され便利な製品が開発されている。多くの人々が利用する地図や時計もデジタル化され、普及している。デジタル化されたこれらの物には、ソフトウェアが搭載されている。とくにこのような、製品に組み込まれたソフトウェアを組み込みソフトウェアという。

ソフトウェアを用いることで製品をより便利にしている。ソフトウェアの修正や拡張は、コードを書き換えたり書き加えればよい。ソフトウェアは、少ないコストで設計の自由度を高くすることができる性質がある。すなわち、ソフトウェアの品質が製品の品質に直接関わる。

ソフトウェアの設計ではプログラミングに C 言語を用いることがある。C 言語はコンピュータへの命令を記述する昔から多くのソフトウェアで使われているプログラミング言語であり、手続き型のプログラミング言語である。ソフトウェアの設計者は、手続き型の C 言語を用い、機器の制御したい命令を順番に記述することに集中すればよく便利である。しかし、制御が複雑であったり大規模なソフトウェア開発では、プログラムの量が多くなり、設計者がプログラムの内容をすべて把握することは難しくなる。

プログラミング言語 Python は、オブジェクト指向のプログラミング言語であり、システム（オブジェクト）の構造を記述、定義する。設計者は制御したいシステムの構造を記述することに集中でき、複雑、あるいは大規模なソフトウェア開発に有効、便利である [4]。

オブジェクト指向言語は 3 つの便利な機能（継承、カプセル化、多態性）がある。

1.2 C プログラムの再利用へ

オブジェクト指向プログラミング言語は高水準な言語であり、比べてこれらの機能が無い手続き型のプログラミング言語 C は低水準な言語 [7] であるとされている。C 言語でプログラムされた既存のソフトウェアを高水準なオブジェクト指向のプログラミング言語のソフトウェアに変更、移植することは慣行である。プログラマが手作業で大規模で複雑なプログラムを書き換えるには、かなりの労力が必要となる。

また、プログラミング学習者が新たにプログラミング言語を学ぶ場合において、一から学習する際に時間や費用、労力といった学習コストがかかるといった問題が生じる [8]。

このことから、手続き型である C プログラムをオブジェクト指向プログラムへ自動翻訳できる翻訳機が求められるようになった。

2 目的

本研究の目的は、手続き型の言語で記述されたコードを替え、オブジェクト指向言語のプログラムにし、オブジェクト指向にリエンジニアリングすることである。最終的には一連の過程を自動化する。

ここで研究の的を絞り、組み込みソフトウェアであるレゴマインドストーム EV3 で用いた C プログラムをオブジェクト指向で記述した MicroPython プログラムに変換することとする。以後、レゴマインドストーム EV3 を EV3 と書く。

2.1 研究課題

研究課題は以下の通り。

- EV3 で用いる手続き型 C プログラムを、手続き型で記述された MicroPython プログラムにし、オブジェクト指向リエンジニアリングを行う。

3 アプローチ

先行研究「オブジェクト指向自動翻訳: Automatic Translation and Object-Oriented Reengineering of Legacy Code」 [7] を参考に課題解決を試みる。元々、C2Eiffel は C プログラムを Eiffel プログラムへ自動翻訳するツールである。先行研究で開発されたオブジェクト指向リエンジニアリングを含む自動翻訳機は、C2Eiffel による翻訳の完全な自動化とオブジェクト指向リエンジニアリングの機能を追加実装したものだ。構図を下図に示す。

これを参考にした C から MicroPython プログラムへの変換の概要を下図に示す。

4 現状・結果

オブジェクト指向リエンジニアリングを含む自動翻訳の設計は未達成である。EV3 で正常に動作できる C プログラムを C2Python により翻訳された Python プログラムは、EV3 では同様な動作ができず、正しく翻訳できていないため、手動で Python プログラムを EV3 で正常に動作できる MicroPython プログラムへ書き換える必要がある。本研究では、アプローチの提案と未完成の翻訳までだ。（試作段階で作成された C, Python, MicroPython プログラムは本稿の付録に掲載した。）

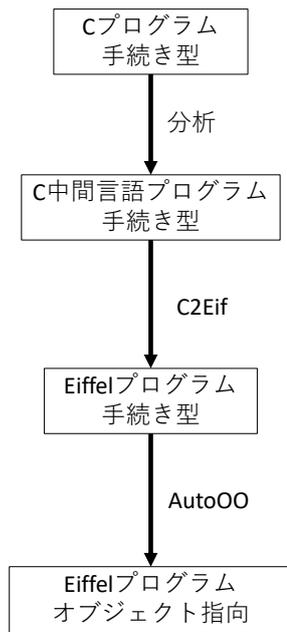


図1 C2Eifの概要

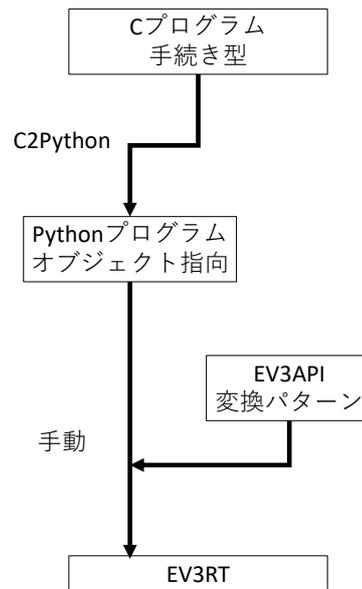


図2 CからMicroPythonプログラムの概要

5 おわりに

今後の課題として、生成された Python プログラムを EV3 で実用を可能にするには、C と MicroPython の EV3API の相違を対応づけることと、リエンジニアリングの自動化、未達成な課題を今後も研究する必要がある、ということだ。

参考文献

- [1] アフレル：Virtual Robotics で学ぶプログラミング基礎，Vol.1，アフレル（2020）.
- [2] アフレル：ロボットで学ぶ C 言語プログラミング基礎，Vol.3，アフレル（2018）.
- [3] アフレル：教育版 EV3 C 言語プログラミングガイド改訂版，Vol.3，アフレル（2016）.
- [4] アフレル：ロボットで学ぶ Java プログラミング基礎，Vol.1，アフレル（2018）.
- [5] 中山清喬，国本大悟：スッキリわかる Java 入門，Vol.3，インプレス（2019）.
- [6] 上田悦子，小枝正直，中村恭之：こらからのロボットプログラミング入門，Vol.1，講談社（2020）.
- [7] Trudel, M.: Automatic Translation and Object-Oriented Reengineering of Legacy Code, Doctoral thesis.
- [8] 久保井彩香，諏訪貴大，吉山大輔：初学者のためのレゴマインドストーム EV3 を用いた C プログラムから

Python プログラムへの翻訳器の設計.

- [9] Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Second Edition, MIT Press, ISBN 978-0-262-53381-2, 2017.