

組込みソフト開発におけるスクリプト言語の利用とメソッドチェック手法 (Method checking mechanism for scripting language in embedded software development)

森田 亘[†] 田中 和明[‡]

九州工業大学大学院情報工学府[†]

九州工業大学大学院情報工学研究院[‡]

1. 緒言

近年 IoT(Internet of Things)の浸透に伴い、組込み技術への注目が集まっている。注目の 1 つとして、組込み技術に用いるプログラミング言語の多様化が挙げられる。従来の組込みソフトウェア開発には、C 言語が使用されてきたが、IoT のような高機能な組込みソフトウェアの開発には、可読性の高さ、機能追加の容易さが求められる。Java や Ruby といったオブジェクト指向型言語を用いた開発が増加している。しかし、組込みソフトウェアの開発は、実行環境であるハードウェアとの親和性を無視することができない。本研究では、Ruby の記述をそのままに、ハードウェアとの親和性が高い言語である mruby を利用し、組込みソフトウェア開発における生産性の向上を目的とする。

また、mruby のようなスクリプト言語の特徴として、実行環境の依存を受けない利点があり、テスト環境下で動作すれば、実行環境下でも動く保証が高い。しかし、実行前にメソッドのエラーをチェックすることができないため、コンパイルを終えたコードが正しく動作する保証がない。そこで、本稿では、コンパイル時にメソッドのエラーの有無をチェックする機能を mruby で作成し、エラーの原因特定を支援する機能におけるアルゴリズムおよび手法について述べる。

2. システム構成

2.1 組込みソフトウェア開発支援用 PaaS

mruby 組込みソフトウェア開発支援用 PaaS を開発中で、その一部に本稿で述べる機能が搭載

Method checking mechanism for scripting language in embedded software development

Wataru Morita [†]
Kazuaki Tanaka [‡]

Kyushu Institute of Technology Graduate School of Computer Science and Systems Engineering [†]
Graduate School of Computer Science and Systems Engineering[‡]

されている。図 1 に本システムの構成図を示す。この PaaS は、mruby のコードが Ruby と大部分が同じであることを利用して、開発者が Ruby のコードを書くことで、自動的に mruby のバイトコードを生成し、mruby を用いた組込みソフトウェアの開発を行うというシステムである。また、特定の組込み機器に対して、指定した組込みソフトウェアのダウンロード(デバイス管理)を行う機能も搭載している。これにより、開発環境の構築および組込み機器への機能搭載の時間が削減できるため、より効率的な組込みソフトウェアの開発が可能である。

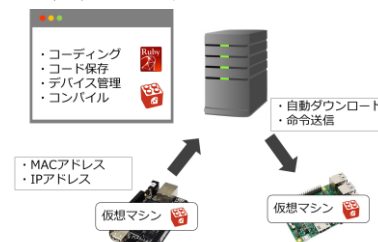


図 1: 組込みソフトウェア開発支援用 PaaS

2.2 mruby

mruby は 2010 年に公開されたオープンソースのプログラミング言語である^[1]。RiteVM 上で実行することにより、開発環境と実行環境における依存を払拭することができる。そのため、組込み開発における実装テストが早い段階で可能となり、組込みソフトウェア開発の効率化に繋がる。mruby の開発環境および実行環境の構成を図 2 に示す。

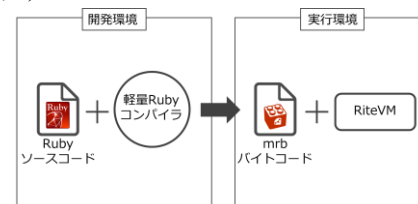


図 2: mruby の開発環境および実行環境の構成

3. 研究内容

3.1 スクリプト言語の問題点

スクリプト言語は、実行前にプログラムのエラーをチェックすることができない。現状として、スクリプト言語による組込みソフトウェア実装時に発生するエラーは、シンタックスエラーか、ハードウェア依存のエラーかを判別することが困難である。組込み技術のようなクロス開発において、実行環境下でシンタックスエラーを発生させないことが望ましい。

3.2 メソッドチェックのアルゴリズム

mruby のメソッドチェック機能概要を図 3 に示す。はじめに、チェックの対象となる Ruby のソースコードを軽量 Ruby コンパイラでコンパイルする。その際、「-v」オプションの追加により、Ruby のメソッドのうち、mruby でも使用可能なものを取得することができる。このオプションは、メソッドの取得以外にも、開発者の入力ミスや、作成した関数もメソッドとして認識するため、シンタックスのチェックも可能である。

一方で、Ruby の全てのクラスにおけるメソッドを書き出したものを mrb バイトコードへと変換し、初期状態の mruby で使用可能なメソッド一覧表を作成する。先ほど取得したメソッドとこのメソッド一覧表を 1 つずつ正規表現で比較する Ruby プログラムを作成した後、mrb バイトコードに変換して、実行する。

以上のフローにより、mrb バイトコードにおけるメソッドチェックが、mruby 自身のプログラムで実行できる。そのため、開発環境および実行環境に負荷が生じない上に、mruby gems (mruby ライブラリ) 内のメソッドにエラーが生じた場合、mruby gems がインストールされていないなどの環境におけるエラーを予期することもできる。

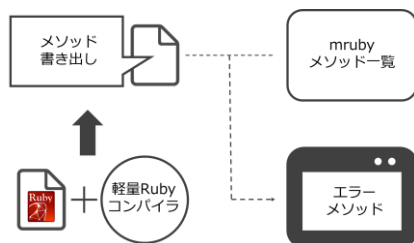


図 3: mruby のメソッドチェック機能概要図

3.3 実装

この機能を本研究で作成中の PaaS に搭載する際は、ユーザビリティを高めるため、Ajax によるメソッドチェックを実行するように設定している。これにより、開発者側は任意のタイミングでメソッドチェックが可能である。

4. 研究結果

メソッドチェック機能の動作の様子を図 4 に示す。PaaS 上のメソッドチェックボタンを押すと、Ajax による非同期接続で、ソースコードのメソッドチェックを開始する。非同期接続であるため、ソースコードを改変し、メソッドチェックを行うたびに、新たなエラーの抽出を行う。図 4 は、ソースコード中において、開発者が独自に作成した system 関数と output 関数、mruby でサポートされていない usleep メソッドにエラーが発生している。このエラー表示により、mruby のソースコードに含まれるシンタックスエラーおよび自作の関数名を実行前にチェックすることが可能である。

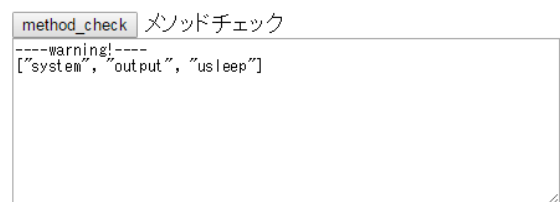


図 4: メソッドチェック機能の動作の様子

5. 結言

mruby を使用したメソッドエラーのチェックおよび組込みソフトウェア開発における生産性向上を支援する機能を開発した。これにより、組込み業界における mruby の有用性を証明し、他の業界から組込み業界への参入促進が期待できる。今後は、完成した PaaS を組込み技術者の教育の場に導入し、組込み技術者の増加を検討している。

本稿で紹介した mruby のメソッドチェック機能は、GitHub にて公開している。

(https://github.com/wataru-script/methods_table)

謝辞

この成果の一部は、国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務の結果得られたものです。

参考文献

- [1] K. Tanaka, Y. Matsumoto, H. Arimori, Embedded System Development by Lightweight Ruby, ICCSA Workshop 2011, pp.282-285, 2011