

平成 21年 6月 5日現在

研究種目：若手研究（スタートアップ）

研究期間：2007～2008

課題番号：19800007

研究課題名（和文） 機能の着脱が可能なスクリプト言語処理系の構成法に関する研究

研究課題名（英文） A Study on Plug-able Script Language Processor Construction

研究代表者

笹田 耕一（SASADA KOICHI）

東京大学・大学院情報理工学系研究科・講師

研究者番号：10436561

研究成果の概要：

本研究は、最近広く利用されているスクリプト言語処理系の核となる部分を抽出し、核部分以外の諸機能を、処理系の構築時、もしくは実行時に柔軟に着脱可能な処理系構築手法の解明を目的として、スクリプト言語 Ruby の処理系を用いて研究を行った。成果として、核となる機能の抽出を行い、機能の着脱が可能な言語処理系を開発した。プログラミング言語 Ruby 処理系の可能性を広げるといふ、実用性の観点からも社会的効果のある研究成果を達成した。

交付額

（金額単位：円）

	直接経費	間接経費	合計
2007年度	1,360,000	0	1,360,000
2008年度	1,350,000	405,000	1,755,000
年度			
年度			
年度			
総計	2,710,000	405,000	3,115,000

研究分野：工学

科研費の分科・細目：ソフトウェア

キーワード：言語処理系，構成法，Ruby，ソフトウェア

1. 研究開始当初の背景

近年、プログラム言語として、Java や C++ などといった静的コンパイル型言語ではなく、Perl, Python, PHP, Ruby に代表されるスクリプト言語を選択することが多くなった。この理由として、スクリプト言語によるプログラム記述の容易性や、提供されている機能・ライブラリの豊富さが挙げられる。これは、特にウェブアプリケーション開発において顕著であるが、プログラム記述の容易性から、アプリケーションを拡張するための

言語として採用されることも多い。

現在よく利用されている Perl, Python, PHP, Ruby のようなスクリプト言語処理系は、主に機能の拡張、拡大を目的として開発が進められており、その規模は増加の一途を辿っている。このような開発方針は、スクリプト言語処理系を単体で利用する際には便利である。しかし、例えば特定用途向けアプリケーションを拡張するための、いわゆるマクロ機能のためにスクリプト言語処理系を組み込む際には、不要な機能まで搭載することになる。不要な機能の搭載は複雑性を増す

ことから、組み込みの実装コストを増大し、セキュリティ上の脆弱性をもたらす危険性がある。また、移植性も悪くなる。たとえば POSIX 環境に特有の機能を利用した機能が組み込まれていた場合、POSIX 環境以外への移植は困難となる。

2. 研究の目的

以上の問題を解決するため、本研究では既存のスクリプト言語処理系の核となる部分を明らかにし、核部分以外の諸機能を、処理系の構築時、もしくは実行時に柔軟に着脱できるようにすることを目指す。以下でも述べるが、本研究はスクリプト言語 Ruby を対象とする。

まず、スクリプト言語 Ruby に最低限必要な機能（各部分の機能）を抽出する。その上で、核部分でないとした機能を容易に着脱可能とするための処理系の構成手法について探求する。ライブラリやクラス構成などの取捨選択は直感的には自明だと思われるが、シグナルやスレッド、ガーベジコレクタの機能など、処理系の構成に深く依存する機能を着脱可能にする実現手法は自明ではない。機能の着脱は、処理系の構築時、もしくは実行時に可能にする。これらを実現する手法はいくつかあるが、どの機能についてどの方式が適しているかを明らかにする。

上記構成手法を適用したスクリプト言語処理系 Ruby を、実際に資源制限があるような組み込み機器へ移植を行い、移植容易性を検証する。また、具体的なアプリケーションに対してスクリプト言語を拡張言語として組み込む実験を行い、拡張言語としての有用性を検証する。

3. 研究の方法

本研究は、松本、笹田（研究代表者）らが開発しているスクリプト言語 Ruby の処理系（Ruby インタプリタ）について、実際に最小構成を見極め、機能の着脱を可能にする仕組みを実装し、実際にアプリケーションに組み込む実験を行い、PDA などの OS への移植の容易性について検証する。

初年度は Ruby インタプリタに求められる最小構成（核部分）について明らかにし、その上で各部分からもれた機能を容易に利用可能とする機能付加の仕組みについて考察する。次年度は、この仕組みを搭載した Ruby インタプリタを、実際にアプリケーションへ組み込み、本研究の有効性について検証する。また、PDA のような環境の異なる計算機へ最小機能（核機能）を搭載した Ruby インタプリタを移植し、移植容易性について検証する。

笹田（研究代表者）は、スクリプト言語

Ruby の処理系の開発、とくに仮想マシン部分の開発に積極的に関わっており、Ruby 処理系自身への理解が深いため、本研究を円滑に進めることができる。また、処理系の開発は多数のボランティアによって行われているため、本研究について開発者の方々から、主に実装面で多くの助言が期待できる。

4. 研究成果

本研究は、最近広く利用されているスクリプト言語処理系の核となる部分を抽出し、核部分以外の諸機能を、処理系の構築時、もしくは実行時に柔軟に着脱できるようにすることを目指す。

この目的を達成するために、平成 19 年度では次に示す課題に取り組んだ。

（1）スクリプト言語処理系 Ruby の核となる部分の機能の抽出

スクリプト言語 Ruby の処理系の核となる必須機能について検討した。検討の結果、通常の機能から大幅に機能を削減することになるが、処理系の評価器部分、およびオブジェクト（メモリ）管理部分が最低限必要な機能であることがわかった。その他の機能は利用するアプリケーションによって要、不要が別れる。その他の機能として、(a)パーサ・コンパイラ(b)スレッド(c)シグナル(d)正規表現エンジン(e)組み込みクラス(f)組み込みメソッドが挙げられる。これらが場合によっては不要であることを確認した。

（2）核とされなかった機能の着脱手法の検討と実装

（1）で述べた(a)～(f)の着脱手法について検討した。(e)、(f)は拡張ライブラリとして着脱が実現可能であることは自明である。しかし、容易に着脱可能な処理系を実現するためにトレース情報を用いた機能の着脱手法を検討した。(a)～(d)に関しては、処理系構築時、もしくは処理系起動時の着脱手法について検討した。また、(1)での必須機能としたオブジェクト管理部分についても、システムに依存した軽量の浮動小数点数表現手法を提案し、メモリ管理オーバーヘッドの削減を実現した。評価の結果、従来に比べて 50%の高速化を実現したことを確認した。

平成 20 年度では、次に示す課題に取り組んだ。なお、本研究はスクリプト言語 Ruby の処理系（以降、Ruby 処理系）を用いて研究を進めた。

(3) 言語処理系構築システムの提案と開発
Ruby 処理系は豊富な機能を有しているが、目的によっては、メモリ消費量、移植性などの問題が生じる。そのため、必要な機能のみで Ruby 処理系を構築するためのシステムを開発し、atomic-Ruby という名前を付けた。現在は、Ruby の組込クラス単位で必要、不要を選択可能である。実行トレースを元にしたレシピファイルの自動生成と、より粒度の細かい機能選択の実現が今後の課題である。

(4) ガーベージコレクションのためのライトバリアの自動挿入

Ruby 処理系は GC を備えているが、対象とする用途によっては GC アルゴリズム（例えばインクリメンタル GC、世代別 GC、並行 GC）を変更したい。つまり、GC 実装を着脱可能としたい。これらの GC はライトバリアを必要とするが、この追加を手作業で行うのは信頼性と開発効率の点で問題である。そこで、メモリ保護機能を利用してライトバリアが必要な箇所を自動的に検出するためのシステムを開発し、実際にインクリメンタル GC を実装した。

(5) 機能を手軽に追加するためのフレームワークの提案と開発

環境や用途によって Ruby 処理系を C で拡張する必要があるが、現状では拡張ライブラリを作成する必要があり、開発が難しい。これを容易にするためのシステムである Ricsin を提案し、開発した。Ricsin は Ruby スクリプトに自然な文法で C を埋め込むことを可能にし、Ruby だけでは困難な機能拡張、および性能改善を行う方法を提供する。

本研究では、プログラミング言語 Ruby の機能の着脱が可能な構成を明らかにした。また、実際に Ruby 処理系について、その機能の着脱可能なプロトタイプを実装し、その有効性を確認した。さらに、Ricsin という異なるプログラミング言語間をつなぐシステムを提案し、開発効率の高い言語処理系の可能性を示した。ただし、実際に組込システムなどへの適用実験を行うことができなかったため、今後実際に適用し、本研究を実際に利用できるソフトウェアとして社会へ還元していく。

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)

┆ 笹田 耕一, Ricsin: Ruby に C を埋め込むシステム, 情報処理学会論文誌 (PRO), 査読有り, Vol.2, No.2, pp.13-26 (2009.3).

┆ 笹田 耕一, Ruby 処理系での軽量の浮動小数点数表現, 第 10 回プログラミングおよびプログラミング言語ワークショップ 論文集, 査読有り, pp. 9-16 (2008).

[学会発表](計4件)

相川 光, 笹田 耕一, 本位田 真一, Ruby 処理系へのスナップショット GC の実装, 第 71 回情報処理学会プログラミング研究発表会, 平成 20 年 10 月 28 日, 鳥根県松江市

笹田 耕一, Ricsin: Ruby に C を埋め込むシステム, 第 71 回情報処理学会プログラミング研究発表会, 平成 20 年 10 月 28 日, 鳥根県松江市

奈須 翔太, 辻 良繁, 相川 光, 笹田 耕一, 竹内 郁雄, atomic-Ruby: 世界でたった 1 つだけの Ruby, 第 71 回情報処理学会プログラミング研究発表会, 平成 20 年 10 月 28 日, 鳥根県松江市

笹田 耕一, Ruby 処理系での軽量の浮動小数点数表現, 第 10 回プログラミングおよびプログラミング言語ワークショップ (PPL2008), 2008 年 3 月 5 日, 仙台市 秋保温泉 緑水亭

[その他]

本研究の研究の成果の一部を取り入れたウェブサイト:
<http://www.ruby-lang.org>

6. 研究組織

(1) 研究代表者

笹田 耕一 (SASADA KOICHI)
東京大学・大学院情報理工学系研究科・講師
研究者番号: 10436561

