

## 発表概要

# JavaScript 仮想機械記述言語のコンパイラにおける 型に基づく最適化

平澤 祐太<sup>1,a)</sup> 岩崎 英哉<sup>1,b)</sup> 鷗川 始陽<sup>2,c)</sup> 高野 保真<sup>3,d)</sup>

2020年6月1日発表

我々は、組み込み機器上で動作するカスタマイズ可能な JavaScript 仮想機械を実現するツールキット eJSTK の開発を進めている。eJSTK は、仮想機械上で動作させるユーザプログラムで使用を想定するデータ型を、ユーザが指定してカスタマイズする。このことで、組み込み機器で必要とされる必要最小限の機能を持つコンパクトな仮想機械の構築を目指している。カスタマイズのため、eJSTK 内部では、第一級の値のデータ型に応じた分岐処理（型ディスパッチ）構文を持つ、静的型付けのドメイン特化言語 VMDL によって、仮想機械命令を記述する。VMDL のコンパイラは、型ディスパッチ処理コードを最適化するが、関数をまたがる型の解析を行っていなかったため、複数箇所から呼び出される VMDL 関数内部におけるディスパッチ処理を最適化することはできなかった。さらに、変数が実行時にとりうる型をコンパイル時に限定できるにもかかわらず、重複した不要な型ディスパッチ処理を行うコードを生成してしまっていた。これらの問題を解決するため、本発表では、VMDL 上の関数呼び出しの型依存関係を解析し、関数の引数となりうる実行時の型を求めることにより、不要なコードを削除する最適化を行った。さらに、VMDL 上の関数を呼び出す実引数の型を使ってディスパッチ先が一意に決定できる場合、インライン展開して重複する型ディスパッチ処理の削減を図った。

## Presentation Abstract

### Type-based Optimizations in Domain Specific Language Compiler for Describing JavaScript Virtual Machines

YUTA HIRASAWA<sup>1,a)</sup> HIDEYA IWASAKI<sup>1,b)</sup> TOMOHARU UGAWA<sup>2,c)</sup> YASUNAO TAKANO<sup>3,d)</sup>

Presented: June 1, 2020

We are developing a framework that enables the user to obtain customized JavaScript virtual machines (VMs) on embedded systems. The framework generates a VM on the basis of the user's specifications on possible datatypes used in the target program. In our previous research, we proposed a statically-typed domain specific language VMDL for describing parts of the source code of the VM, which has a syntax for type-based dispatching processes. The VMDL compiler can optimize the generating code for the type-based dispatching process on the basis of the results of its type analysis, but the eliminations of unnecessary type checking processes are insufficient because its type analysis is limited to intra-functional one. To resolve this problem, this presentation proposes design and implementation of two techniques adopted in our new VMDL compiler. The first is inter-functional analysis that computes type dependency relations among VMDL functions and eliminates unnecessary code by determining possible datatypes given to each function as its arguments. The second is inline expansions of auxiliary function calls when only a single branch is determined to be selected within the dispatching process of the auxiliary function.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

<sup>1</sup> 電気通信大学  
The University of Electro-Communications, Chofu, Tokyo  
182-8585, Japan

<sup>2</sup> 高知工科大学  
Kochi University of Technology, Kami, Kochi 782-8502,  
Japan

<sup>3</sup> 北里大学  
Kitasato University, Sagami-hara, Kanagawa 252-0373,  
Japan

a) 19yhirasawa@ipl.cs.uec.ac.jp

b) iwasaki@cs.uec.ac.jp

c) ugawa.tomoharu@kochi-tech.ac.jp

d) tyasunao@kitasato-u.ac.jp