【 50 】

氏名	安松一樹（やすまつかずき）
博士の専攻分野の名称	博士（工学）
学位記番号	第 12729 号
学位授与年月日	平成 8 年 10 月 29 日
学位授与の要件	学位規則第 4 条第 2 項該当
学位論文名	A Translation Method from Smalltalk into Interoperable C Code (Smalltalk から相互運用性の高い C コードへの変換方式に関する研究)
論文審査委員	(主査) 教授 井上 克郎 (副査) 教授 首藤 勝　教授 都倉 信樹　教授 菊野 亨 奈良先端科学技術大学院大学教授 鳥居 宏次

論文内容の要旨

本論文は，純粋オブジェクト指向プログラミング言語である Smalltalk 言語に着目し，その応用プログラムの構築と配布に関する 2 つの問題点，すなわち，従来の環境で動作可能な実行形式の生成の困難性，および，従来の言語との協調実行（相互運用）の困難性，とを解決するため，Smalltalk から相互運用性の高い C コードへの変換方式を提案し，提案した方式を用いたシステムを実装し評価した研究をまとめたものである。

Smalltalk 言語は，関数閉包，並列実行，例外処理，ゴミ集めといった高度な言語機能を備えている。一方，C 言語はこれらの言語機能を備えてはいないが，従来の環境で動作可能な実行形式の生成が可能で，他言語との相互運用性が高い。Smalltalk コードを相互運用性の高い C コードへ変換することにより，応用プログラムを従来の環境で動作させることが可能となり，また，応用プログラムを複数の言語を用いて記述することが可能となる。しかしながら，Smalltalk と C とでは言語機能に大きな差があるため，Smalltalk の高度な言語機能を保存しつつ，かつ相互運用性の高い C コードを生成することは困難である。本論文では，Smalltalk の高度な言語機能を保存しつつ，かつ相互運用性の高い C コードを生成するため，次の 2 つの手法を提案している。

まず，Smalltalk の実行モデルに依存したクラスと同様の機能・インタフェースを持ち，通常の OS などの従来の環境で直接実行可能な実行時代用クラスライブラリ（runtime replacement classes）を提供するという手法を提案している。実行時代用クラスライブラリの基本的な考えは，Smalltalk の実行コードオブジェクトを機械コードに，実行環境オブジェクトをスタックフレームに，それぞれ対応付けることである。実行時代用クラスライブラリにより，Smalltalk と C との実行モデルの違いを安全かつ効率よくカプセル化し，関数閉包，並列実行，例外処理といった Smalltalk の言語機能を保存しつつ，相互運用性の高い C コードの生成を可能としている。

次に，ゴミ集めを提供しない言語を含む複数の言語で記述したプログラムを組み合わせる場合にも，安全に効率よく実行可能で，かつ，オブジェクト指向応用プログラムに適した世代別ゴミ集め手法を提案している。ゴミ集めの基本的な考えは，間接参照により保守的ポインタ判別と同時にオブジェクトの移動を可能とすることである。このゴミ集め手法により，Smalltalk と C とのデータモデルの違いを安全かつ効率よく吸収し，ゴミ集めを行いつつも Small-

talk のデータと C のデータとの共存を可能としている。

これら 2 つの手法を用いた SPiCE と呼ぶ Smalltalk から C への変換システムを実装し，実際の応用プログラムを用いた評価を行った。評価により，提案した変換方式は，Smalltalk の高度な言語機能を保存しつつ，かつ相互運用性の高い C コードを生成することが確認された。Smalltalk の高度な言語機能を保存していることは，既存の複数の応用プログラムの変換により示された。生成された C コードの相互運用性は，Smalltalk を含めた複数の言語による応用プログラムの構築により示された。生成された C コードの効率は，最速の Smalltalk システムとの比較により，ほぼ同等であることが示された。

論文審査の結果の要旨

本論文は，純粋オブジェクト指向プログラミング言語である Smalltalk 言語に着目し，その応用プログラムの構築と配布に関する 2 つの問題点，すなわち，従来の環境で動作可能な実行形式の生成の困難性，および，従来の言語との協調実行（相互運用）の困難性，とを解決するため，Smalltalk から相互運用性の高い C コードへの変換方式を提案し，提案した方式を用いたシステムを実装し評価した研究をまとめたものである。

Smalltalk 言語は，関数閉包，並列実行，例外処理，ゴミ集めといった高度な言語機能を備えている。一方，C 言語はこれらの言語機能を備えてはいないが，従来の環境で動作可能な実行形式の生成が可能である。本論文では，Smalltalk の高度な言語機能を保存しつつ，かつ相互運用性の高い C コードへ変換するため，次の 2 つの手法を提案している。

まず，Smalltalk の実行モデルに依存したクラスと同様の機能・インタフェースを持ち，通常の OS などの従来の環境で直接実行可能な実行時代用クラスライブラリ（runtime replacement classes）を提供するという手法を提案している。実行時代用クラスライブラリにより，Smalltalk と C との実行モデルの違いを安全かつ効率よくカプセル化し，関数閉包，並列実行，例外処理といった Smalltalk の言語機能を保存しつつ，相互運用性の高い C コードの生成を可能としている。

次に，ゴミ集めを提供しない言語を含む複数の言語で記述したプログラムを組み合わせる場合にも，安全に効率よく実行可能で，かつ，オブジェクト指向応用プログラムに適した世代別ゴミ集め手法を提案している。このゴミ集め手法により，Smalltalk と C とのデータモデルの違いを安全かつ効率よく吸収し，ゴミ集めを行いつつも Smalltalk のデータと C のデータとの共存を可能としている。

これら 2 つの手法を用いた SPiCE と呼ぶ Smalltalk から C への変換システムを実装し，実際の応用プログラムを用いた評価を行った。評価により，提案した変換方式は，Smalltalk の高度な言語機能を保存しつつ，かつ相互運用性の高い C コードを生成することが確認された。生成された C コードの効率は，最速の Smalltalk システムとほぼ同等であった。

以上の研究成果は，オブジェクト指向言語の相互運用技術の発展に貢献しており，本論文は博士（工学）論文として価値あるものと認める。