

## 学位論文題名

A Study on Construction Method for Simulation  
Debugging Environment of Embedded Computer Software(組み込み型計算機ソフトウェアのシミュレーション  
デバッグ環境の構築方法に関する研究)

## 学位論文内容の要旨

近年、組み込み型計算機EC (Embedded Computer) ソフトウェアのシミュレーションデバッグ環境SDE (Simulation Debugging Environment) が広範的に使われている。デバッグ環境が完成するまでEC応用ソフトウェアの開発は進められないので、ECソフトウェアデバッグ環境の構築方法は重要である。しかしながら、従来のSDEの構築方法では、手作業的な方法が用いられているため、種々の問題があり、効率的なSDE構築方法の開発への要求が高まっている。

このような観点から本論文では、SDEの構築に対する新しい方法を提案している。この方法は機能抽象記述と再利用技術に基づき、プログラム自動生成法を用いて、SDEを高い生産性で構築するための支援環境を実現するものである。本論文では、構築目標とするSDEの基本構造を定義するために、一般的なECの記述モデルと専用小規模の記述言語ESL (EC Specification Language) を提案している。さらに、実装システムの記述エディタ、プログラムジェネレータおよび再利用環境の実現法についても論じている。本論文は以下に示す7章から構成されている。

第1章は序論であり、まず本研究の概要を紹介した。次に、本論文の章節の構成を述べた。

第2章では、まずSDEの利用状況、開発方法および問題点を検討し、本研究の背景と意義を述べた。従来のSDEの構築方法では、新機種のECが誕生すると、それに対応するSDEを早期に入手することが困難である。また、作成したプログラムの改造と保守管理も困難である。このため、SDEの構築において生産性を高める環境が望めず、ECの種類増加にも対応できない現状を明らかにした。

次に、構築目標とするSDEの基本モデルの構造、内部の実現方法およびその利点を述べた。SDEモデルは三層 (Visual, Shell, Kernel) の基本構造を持ち、下層は上層をサポートして、上層は下層のプリミティブを引用する。汎用性を守るため、層間プリミティブは統一した関数で定義されている。このような構造は、SDEの基本構造が明確に表現され、再利用可能な部分と再利用の困難な部分も有効に分離できる。

さらに、提案法で実装したシステムの概要を紹介した。実装システムは主として機能記述部 (エディタ)、シミュレーションプログラム生成部 (ジェネレータ)、ソフト部品ベース (再利用ライブラリ) の三つの部分に分けられる。記述エディタは、目標ECの主要機能を記述して特定の記述ファイルを作成することができる。ジェネレータは、記述ファイルを翻訳して標準C言語型のシミュレーションプログラムに変換する。再利用ライブラリはSDEの構築をサポートする関数を保存している。これらによって、新たなECに対応した SDEが高い効率

で構築できるようになった。

第3章では、SDEの構築するため、ECの記述に関する機能記述モデル、記述言語および記述エディタについて詳細に述べている。まず、機能記述の目的と主要な方法を論じた。機能記述はジェネレータにECの規範記述を提供することを目的とする。ECの機能記述を行なうため、高位機能レベル(RTL-Register Transfer Level)で、機能族(C)、機能体(E)、属性(A)、操作(O)に基づき、記述モデルを定義した。また、本研究では一般的なECの各種機能体の属性と基本操作の抽象化して定義を行った。この抽象化は、基本的な機能族の分類、C言語型で機能族の機能体の属性値類型の定義、属性の構造体の定義、および、機能体が有する基本操作の類型と、対応する操作関数の定義を含んでいる。

次に、記述モデルと対応する小規模な専用記述言語ESLと記述エディタの開発方法を紹介した。従来のテキスト入力方法は間違いが起り易く、また、記述した内容の修正、検索等が面倒で、正確さの保持が困難である。このため、本研究では、ウィンドウ画面上での操作を主とする専用のスクリーンエディタを開発した。このエディタの中では、ウィンドウで各機能族に特別なテンプレート画面を提供する。一種類のテンプレートは一種類の機能体に対応しており、機能体の記述は各テンプレートにより行なう。テンプレートの役割は直観的で、簡潔な入力手段を提供し、これによりユーザは個々の表を完成することで、各機能体を記述することができる。テンプレートのもう一つの役割は正確な記述フォーマットによる入力処理で、機能体の検索が容易になる。

第4章では、本研究で開発したプログラム自動ジェネレータに関して、その機能、特徴および実現方法を述べている。このジェネレータの役割は記述エディタで作成したESL記述ファイルをC言語型の機能シミュレーションプログラムに変換することである。ジェネレータはESLファイルを二段階で処理する。ジェネレータはまず各機能族の有する機能体のC言語型の定義と構造データを作成し、それから、各種機能体の動作をシミュレートするためのC言語プログラムが生成される。同様に基本操作をテストするためのプログラム、ヘルプファイル、Makefileファイル、警告メッセージファイルも作成される。ある特殊機能については、ジェネレータはメインフレームを生成するようになっており、これを利用して手作業で対応することができる。メインフレームは、具体的な処理を除いて主要な処理の流れを表現するプログラム関数で、ユーザはメインフレームに付属するヘルプとコメントメッセージを参考し、再利用関数を用いて、手作業で作成したコードで、対応する関数を完成できる。

第5章では、本研究では、SDEを構築する際のソフトウェア再利用の方法について述べている。本章ではまず、一般的な再利用システムの特徴、基本構成と応用現状を検討した。従来の再利用システムの大部分はビジネス用にしか適用できず、SDEのようなソフトウェアの開発するための再利用への研究はなされていないことを指摘した。

次に、SDEの開発するために再利用の重要性と再利用環境の実現方法を述べた。本研究では、従来のような単純なコード再利用ではなく、オブジェクトコンセプトに基づいて、クラス再利用モデルで再利用部品の表現、検索および追加方法を提案した。開発した再利用部品ライブラリはSDEのシミュレーションカーネル、シェル及びビジュアル層でよく使われる再利用可能な関数を含んでおり、プログラム自動生成時や手作業によるコーディングの際に利用される。汎用性、標準性を守るため、ライブラリはANSI規格に準拠して開発されている。さらに、開発中に作成した再利用可能な関数もライブラリに追加することができる。

第6章では、実際のシステムと実験結果について述べた。提案した方針に従って、SUN社のSparc/4上で、XウィンドウシステムとC言語を用いて実験システムを実装した。ソースコードのレベルで評価を行った結果、一つのSDEを構築するソースコード中、40%のソースコードが自動生成によって、54%のソースコードが再利用ライブラリから作成された。残りの手作業によるソースコード量は全体の約1/15にとどまった。これらのことから、提案した方法により、SDEの開発効率が従来法より向上することが示された。

第7章は結論で、本手法、およびその実装システムの長所について述べるとともに、今後の課題について言及している。

# 学位論文審査の要旨

主 査 教 授 青 木 由 直  
副 査 教 授 栃 内 香 次  
副 査 教 授 北 島 秀 夫

学 位 論 文 題 名

## A Study on Construction Method for Simulation Debugging Environment of Embedded Computer Software

(組み込み型計算機ソフトウェアのシミュレーション  
デバッキング環境の構築方法に関する研究)

近年、組み込み型計算機EC(Embedded Computer)ソフトウェアのシミュレーションデバッキング環境SDE(Simulation Debugging Environment)が広範囲に使われている。ここで、EC応用のソフトウェアは、デバッキング環境が完成するまで開発を進められないのため、ECデバッキング環境の構築方法が重要視されている。従来のSDEの構築方法は、手作業による方法が主体で種々の問題点が指摘されており、効率的なSDE構築方法の開発要求が高まっている。このような状況において、本研究は新たな組み込み型計算機ソフトウェアの早期開発を可能にするため、対応するシミュレーションデバッキング環境の生産性の高い構築方法について検討を行ったものである。本論文では、シミュレーションデバッキング環境の新たな構築方法について提案するとともに、機能抽象記述、プログラム自動生成及びソフトウェア再利用技術における実装システムの実現について検討しており、その主要な成果は以下の点に要約される。

(1) 構築の目標であるSDEの基本階層構造モデルとして、一般的な組み込み型計算機を記述するための機能を備えた記述モデルと記述言語を提案している。さらにウインドウ画面での操作のための専用スクリーンエディタを開発した。提案した方法による記述環境を構築することにより、記述した内容に対する容易な修正、検索および正確さの保持が可能であることを示した。

(2) 従来の手作業にとる開発方法に替えて、プログラムの生産性を向上させるため、自動プログラムジェネレータを開発した。ジェネレータは記述したファイルを翻訳して、標準C言語型のシミュレーションプログラムに変換する。これによりジェネレータで大量のプログラムの自動的生成が可能となり、手間のかかる手作業が減少し、プログラムの可読性と保守性を向上させた。

(3) 重複あるいは複雑なコーディング作業を減らすために、ソフトウェアの再利用環境を開発した。これは単純な関数再利用ではなく、提案したクラス再利用モデルに基づき、再利用部品の表現手法とシミュレーションデバッキング環境の構築を支援する再利用ライ

ブラリを作成し、利用している。さらに、再利用ライブラリを効率的にアクセスするため、対応する再利用部品を検出および追加のためのツールも開発した。

(4) 提案した方法により、新たな組み込み型計算機ソフトウェアのシミュレーションデバッグ環境が、高い効率で構築できるようになった。ソースプログラムレベルに基づく評価により、40パーセントのソースコードが自動生成によって、54パーセントのソースコードが再利用ライブラリから作成された。残りの手作業によるソースコード量は全体の約1/15にとどまった。これらのことから、応用開発までの期間を短縮できることを実証した。

以上、本研究は、シミュレーションデバッグ環境構築にかかわる生産性を向上するための方法を考案し、組み込み型計算機応用システムの開発に貢献しており、計算機工学・メディア工学の進歩に寄与するところ大である。

よって著者は、北海道大学博士（工学）の学位を授与される資格あるものと認める。