

8Q-7 オブジェクトフロー型ソフトウェア開発方法への一提案*

平嶋 史武[†] 島田 馨[‡]
(株) ソフテック[†]

伊藤 貴雄[‡]
帝京平成大学情報学研究科[‡]

永井 正武[§]
帝京大学理工学部[§]

1. はじめに

本論文は、オブジェクトフロー型という概念を導入し、生産性向上のための並列処理を行うプログラムの開発方法を提案する。

2. オブジェクトフロー型

オブジェクトフロー型とは、階層化データフロー図[1]によるプログラムの設計法と同等の概念で出発し、データと機能で構成されたオブジェクト同士の関係を、可視表示手法の導入により処理フロー形式にし、ソフトウェアの再利用性も考慮したものである。

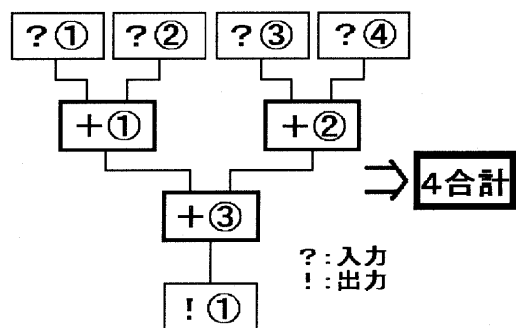


図1 4合計オブジェクトフロー

たとえば図1は4入力のデータの合計を求めるオブジェクトフローの簡単な例である。図示のように、入力オブジェクト4個、出力オブジェクト1個、加算オブジェクト3個から構成されている。ここで、加算オブジェクトとは入力2個、出力1個で、入力を加算演算したものを出力するオブジェクトのことである。

また、オブジェクト同士を接続している線を接続線と呼び、データの流れを表している。この結果として、本システムの場合は、C言語によるクラスレベルのプログラムリストを得ることができる。同時に入出力関係に着目したネットリストも得ることができる。

図1で表しているオブジェクトフローは、新たにオ

ブジェクトフローを作成する部品として使用することができる。

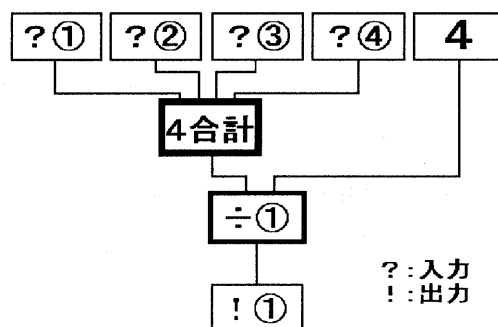


図2 4平均オブジェクトフロー

図2では、4入力の平均を求めるオブジェクトフローであるが、図1のオブジェクトフローを使い、ソフトウェア再利用性を実現した場合の適用例である。

このように、オブジェクトフローを再利用することにより、生産性の効率を上げることができる。また、開発言語の深い知識がなくてもプログラムを開発することができる。

3. オブジェクトフロー開発環境

3.1 システムの機能

このオブジェクトフロー型を開発・デバッグするためのシステムの機能は以下の通りである。

(1) マルチスレッド型並列処理機能

図1の加算①と加算②は同時に並列処理することができる。そこで、開発環境では、同時に処理できるオブジェクトについて、マルチスレッドを用いて並列処理をシミュレートしている。

オブジェクトを2つのスレッドで実行した場合、図3のように並列処理できる部分はそれぞれのスレッドに振り分けて処理をする。

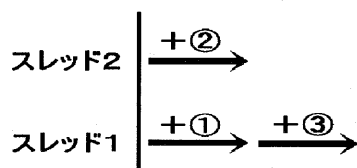


図3 スレッドイメージ

*Design and Execution of Programs Using Layered Object Flow

[†]Fumitake Hirashima and Kaoru Shimada: SOFTECH CO.,LTD

[‡]Takao Ito: Graduate School of Informatics, Teikyo Heisei University

[§]Masatake Nagai: School of Sciences and Engineering, Teikyo University

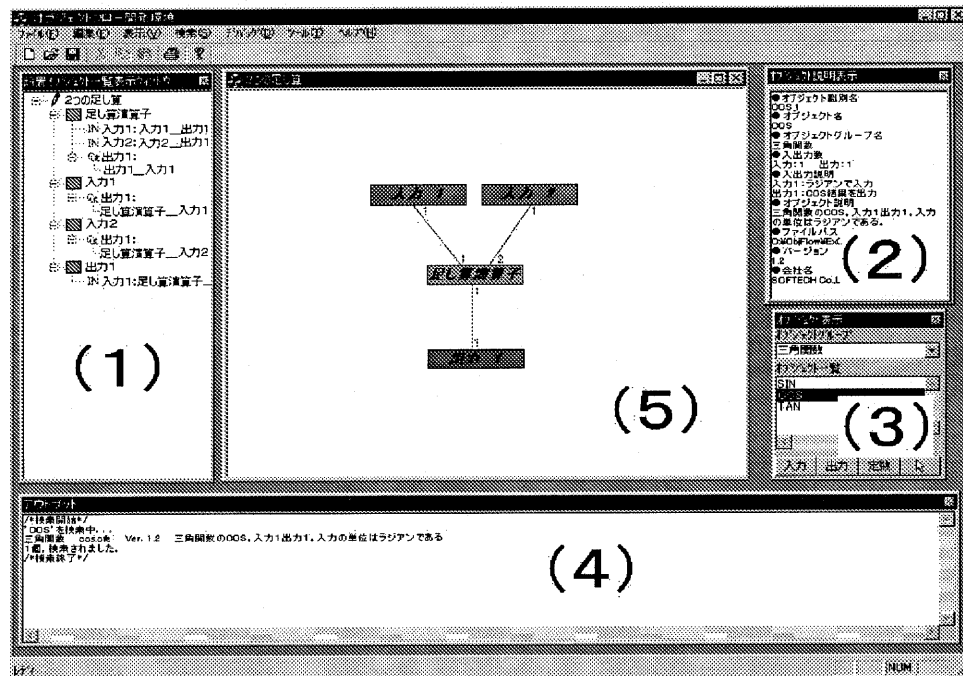


図4 メインウィンドウ

(2) 出力機能

作成したオブジェクトフローをCソースとしてファイルに出力する。よって、他のCコンパイラなどとの連携が容易になる。また、ネットリストも出力可能である。

(3) オブジェクトの検索機能

登録してあるオブジェクトフローから、ユーザが目的のオブジェクトを検索できるように、複数の項目について検索できる。

3.2 システムの画面

図4はオブジェクトフロー作成ツールのメイン画面例である。

(1) オブジェクト一覧ウィンドウ

作成しているオブジェクトフローで使用しているオブジェクトの一覧を表示する。また、オブジェクトの接続先も表示する。

(2) オブジェクト説明ウィンドウ

オブジェクトの説明を表示する。

(3) オブジェクトウィンドウ

標準に用意しているオブジェクトや、ユーザが登録したオブジェクトを表示する。ここで、オブジェクトを指定することにより、現在作成しているオブジェクトフローにこのオブジェクトを設置する。

(4) アウトプットウィンドウ

オブジェクトの検索結果や、エラーなどを表示するメッセージ欄である。また、指定したオブジェクトのCソースも表示する。

(5) オブジェクト設置ウィンドウ

オブジェクトが設置され、オブジェクト同士を接続する。

4. システムの評価

このシステムを実装・稼動[2]することにより、ソフトウェア生産性の向上を図ることが可能となった。また、ソフトウェア製作に必要なCソースと、ハードウェア製作に必要なネットリストを出力することにより、ハードウェアとソフトウェアのセマンティックギャップの短縮化が実現可能となった。さらに、並列処理部分をマルチスレッドでシミュレートすることで、処理の高速化が可能となった。

5. おわりに

システムの評価から、生産性向上のための並列処理を行うプログラムの開発方法に糸口をつかむことができた。今後は、さらにネットリストから論理式化し、最適化を図っていく予定である。

6. 参考文献

- [1] 川口進, 白須宏俊, 奥田哲: 階層化データフロー図によるプログラムの設計と実動, 電子情報通信学会論文誌 D-I, Vol.J82-D-I, No.10, pp.1265-1275, 1999
- [2] 伊藤貴雄, 平嶋史武, 黒坂功, 永井正武: オブジェクトフロー型概念によるソフトウェア開発技法への一提案, 情報処理学会第 62 回全国大会, 5Z-4, 2001