

複数のプログラミング言語におけるオブジェクト指向

4C-6

プログラミングモデルの提案

下羅 弘樹* 西野 順二** 小高 知宏** 小倉 久和**

(*福井大学工学研究科 **福井大学工学部)

1 はじめに

オブジェクト指向は再利用、システム保守が容易であると言われている。このような理由からオブジェクト指向による開発が進められているところも多い。現在、そのコーディングは一つのプログラミング言語によってされることが多い。しかし、実際には現在多種多様なプログラミング言語が存在し、プログラムを開発するにあたって、複数のプログラミング言語を用いたほうが開発効率が上がる場合がある。これは、それぞれの言語に、得意領域、不得意領域があるためである。例えば、文字列処理部だけ文字列処理の得意な言語、リスト処理部だけリスト処理の得意な言語で処理したほうが開発効率が上がるであろうと考えられる。このような複数のプログラミング言語を用いての実装が手軽に出来れば開発効率が改善されるであろう。

また、オブジェクト指向は並列分散処理にも向いている。この並列分散処理に対する需要は、これからますます大きなものとなるであろう。その背景として、年々パーソナルコンピューターが高速かつ安価になってきており、また、マシン間のネットワークも高速になってきている。そして、それにともなって、大型計算機ではなくパーソナルコンピューターを、何台もネットワークで繋げて使用する環境も多くなってきている。現在、我々の研究室にもたくさんの個人使用のマシンがあり、ネットワークで接続されている。しかし、それらのほとんどのマシンの、CPUなどの計算機資源は全て使われているわけではない。実際、CPUがほとんど使

われていないコンピュータも多い。

一方、CPU等の計算機資源のかかる処理もあり、ニューラルネットワークや遺伝的アルゴリズムなどを用いたプログラムなどは、多くのCPU資源を必要とする。

現在のこのようなCPU資源を多く要求するプログラムがある一方で、計算機資源が余っている状況は、非常に非効率であるといえる。このような状況は、並列分散処理がもっと手軽に扱えれば改善されると思われる。

本研究では、計算機資源を抽象化して扱うことにより、このような複数の言語間でのやりとり、プログラムの並列分散処理が手軽に扱えるオブジェクト指向プログラミングモデルを提案する。

2 プログラミングモデルの提案

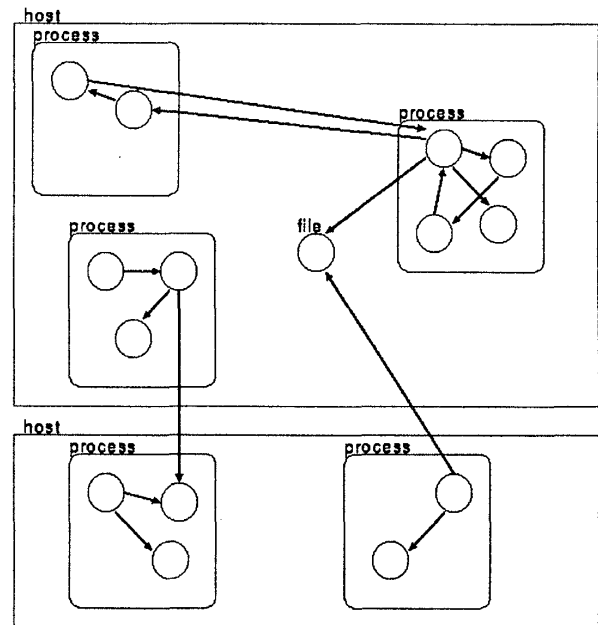


図 1: プログラミングモデルの概観

Proposal of an object oriented programming model in plural programming languages environment
 Hiroki Shimora* Junji Nishino**
 Tomohiro Odaka** Hisakazu Ogura**
 *Graduate School of Engineering, Fukui University
 **Faculty of Engineering, Fukui University

2.1 方針

本設計は、全ての計算機資源を統一的に扱うことを重要視して作成する。そうすることによって汎用のシステムとなるようにする。また、可能な限りシステムを簡略化する。

2.2 設計

扱う全てのものをオブジェクトというかたちで抽象化し、オブジェクト間の通信は全てコネクションとして抽象化する。オブジェクトとして扱うものは、整数値、文字列等のビットの並びで表現可能なものだけでなく、扱える限りのものをオブジェクトというかたちで抽象化する。例えば、ファイル、デバイス、処理の流れ等である。本モデルでは、簡略化のため非同期の通信のみをサポートする。また、本モデルでは複合オブジェクトは扱わない。これは、図2のようにB、Cを属性として持つAという複合オブジェクトは図3のようなオブジェクトのネットワークと等価であるため必要でないためである。

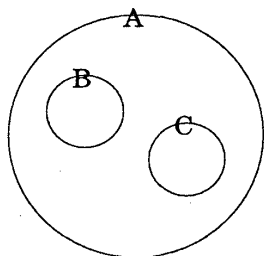


図2: 複合オブジェクト

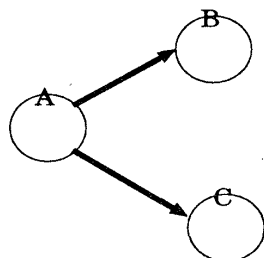


図3: 単一階層ネットワーク

3 プログラミングモデルの特徴

本モデルには、

- 複数の言語の組み合わせによるプログラムが容易になる。

- ネットワーク上でのプログラミングが容易になる。
- 並列分散処理、負荷分散のプログラムが比較的容易に作成出来、多くの計算機資源を比較的容易に確保出来るようになる。
- プログラムの動的再構成が容易になる。
- モデルがシンプルである。

等の利点がある。

3.1 プログラム動的再構成手法

本モデルでは、オブジェクトを実行中に置き換えることが可能である。そのため、プログラムの動的再構成が容易であるという特徴を持つ。ここでのプログラムの動的再構成とは、プログラム実行中にプログラムを書き換えることにより、プログラムを構成し直すことを示す。

このような手法を用いて、遺伝的アルゴリズムのプログラムの実行途中で、中間結果を見て、評価関数を書き直したり、遺伝子の一つ、ユーザの作成した遺伝子と置き換えたりすることが簡単に出来る。

また、最初はドライバやスタブだけで組んでおき、徐々にそれらを実行時に置き換えていくことによってプログラムを作成するという開発方法をとることも出来る。

4 まとめ

本研究では、全ての計算機資源を統一的に扱うことにより、複数の言語間でのやりとり、プログラムの並列分散処理が手軽に扱えるオブジェクト指向プログラミングモデルを提案する。

本モデルは CORBA[1] や PVM よりも広い範囲のものをオブジェクトとして扱う。本モデルは実行効率よりもプログラミング作業を効率化させることのほうに重点を置いている。

参考文献

- [1] Mark Roy Randy Otte; Paul Patrick. 分散オブジェクト指向 CORBA. プレンティスホール, 1996.