

並行オブジェクト指向言語 y の実行時環境

4 N-4

崔梗淦* 野呂昌満** 原田賢一*

* 慶應義塾大学理工学部 ** 南山大学情報管理学科

1 はじめに

分散オブジェクト指向計算モデルでは、オブジェクトを並行実行の単位として、オブジェクト間で通信を行ない、メッセージを受理することで対応する処理を行う。プログラミング言語 y はオブジェクト間通信を、一対一と一対多、通常と優先、同期と非同期、単方向と双方向の四種類に分類し、これらを任意に組み合わせて豊富な通信機能を表現可能な言語である。本稿では、y における並行オブジェクトの実現を支援するための実行時環境について述べる。

2 プログラミング言語 y

プログラミング言語 y [Koj93] は、構文の基本部分を C++ から採り入れたが、並行オブジェクトのための記述や継承 [Koj95] のための記述の点で大きく異なる。本稿では、並行オブジェクトを支援するための実行時環境について取り上げるので、継承についてはとくに触れない。

2.1 並行オブジェクトと逐次オブジェクト

y のオブジェクトには、同時に実行される並行オブジェクトと、単にデータ抽象を実現するための逐次オブジェクトとがある。並行オブジェクトは

- 局所的記憶
- メソッド
- シナリオ

からなる。シナリオは、メッセージの受理や送信を順路式の考えを取り入れて、オブジェクトの生成から消滅までの振舞いを記述したものであり、シナリオを持つことで、能動的に動作するクライアント型オブジェクトや、サーバ/クライアントの両方の性質を持つハイブリッド型オブジェクトを考えることができる。

現在の y ではメッセージの送受信のためには、次のような構文が用意されている。

同期メッセージ通信 同期メッセージ通信には、C++ のメンバ関数の呼出しと同様の構文を用いる。

```
<obj>.<message>({args})
```

Runtime environment of object oriented concurrent programming language y

Kyong-Rok Choi*, Masami Noro**, Ken'ichi Harada*

*Dept. of Computer Science, Keio University, 3-14-1 Hiyoshi, Kouhoku-ku, Yokohama 223, JAPAN

**Dept. of Information Systems and Quantitative Science, Nanzan University, 18 Yamazato-cho Showa-ku Nagoya 466, JAPAN

send 文 send 文は非同期通信時に結果を格納すべきオブジェクトを指定して、メッセージを送信する。

```
send <obj>.<message>({args}) #<dest>;
```

receive 文 send 文で指定した結果の待ち行列から最初の結果を取り出す。

```
receive <dest>;
```

discard 文 send 文で指定した宛先に向けて送られて来た結果を破棄する。

```
discard <dest>;
```

accept 文 メッセージを受理する。return 文の終了によって、メソッドの実行は終了し、シナリオの残りの部分の実行を継続する。

```
accept <message>({formal_arg_list});
```

return 文 結果を送信者に返す。

```
return {obj};
```

結果を必要とする非同期通信の場合、send 文の結果は宛先として指定されたオブジェクトに直接格納されるのではなく、そのオブジェクトに結びつけられた待ち行列が生成され（この待ち行列を宛先オブジェクトと呼ぶ）、そこに到着順に格納される。結果を利用するためには、receive 文によって、この待ち行列から明示的に取り出す。これは、ABCL/1 などの未来オブジェクトとほぼ同様の機能であるが、任意の個数の結果を受け取ることができるという点で異なっている。一対多通信において複数の結果を利用したい場合に用いられる。

3 並行オブジェクトのための実行時環境

y のコンパイラはソースコードから C++ コードを生成する。実行のためにはこの C++ のコードを更に、C++ でコンパイルして実行ファイルを得る。

前述のように y では、多彩な通信機能を並行オブジェクトに実装しなければならない。そのために、まず単一の計算機上で動作する実行時環境を実現した。この実行時環境上では並行オブジェクトはそれぞれ UNIX のプロセスとして実装されており、オブジェクト間の通信には BSD 系の UNIX のプロセス間通信の機能であるソケットを利用した。

UNIX 上では、プロセスが並行処理の単位であり、並行オブジェクトを UNIX のプロセスとすることは自然な方法である。また、プロセスとして並行オブジェクトを実現することで、資源の共有や並行オブジェクトのスケジューリングを実行時環境で支援する必要が無いという利点もある。

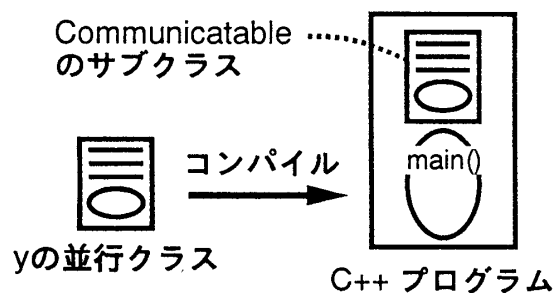


図 1: 並行クラスと生成されるコードの関係

これらを用いて、並行オブジェクトを実現するために、実行時ライブラリをC++のクラスとして定義した。また、一対多の通信を可能とするために、オブジェクトマネージャを実現した。

3.1 並行オブジェクト

並行オブジェクト、オブジェクト間通信をそれぞれUNIXのプロセス、プロセス間通信として実現するために、Communicatableクラスを定義した。Communicatableクラス(とその派生クラス)は、主にオブジェクト間の一対一の通信を支援するクラスであり、非同期のメッセージの受付と他のオブジェクトに対するメッセージの送信のための基本機能を持っている。yプログラムのすべての並行クラスはコンパイル時に、このクラスの派生クラスとしてC++のコードに変換される。その際にシナリオはそのクラスのある特定の名前を持つメソッドとして変換され、シナリオ中のメッセージの送受信、同期に関する操作はCommunicatableクラスのメソッドを用いて実現される。

yプログラムの、それぞれの並行クラスに対して、一つずつC++のプログラムが生成される。そのプログラム中ではインスタンスの生成と、シナリオに相当するメソッドの呼出しを行なう。実行時には、オブジェクトマネージャに対して、オブジェクト生成要求を送ることで、新たに並行オブジェクトをUNIXのプロセスとして生成することができる。

また、Communicatableクラスは到着したメッセージを格納する待ち行列を持つ。一対多通信の際に用いられる宛先には、後で述べる宛先オブジェクトが用いられる。

3.2 オブジェクトマネージャ

オブジェクトマネージャはすべてのオブジェクトを管理する。各ノードに一つずつ配置されたオブジェクトマネージャは、それぞれのノード上でのオブジェクトの生成・消滅の制御やオブジェクト間の一対多の通信の支援を行なう。ノードを跨るオブジェクト間通信はまずオブジェクトマネージャを経由して行なわれ、相手先オブジェクトに関する情報を得ることでそれから後は直接通

信が可能となる。

オブジェクトマネージャ自体も、Communicatableクラスの派生クラスとして実現されている。各並行オブジェクトからの要求は他のオブジェクト間通信と全く同等に行なわれる。

また、一群のオブジェクトをあるグループと関連付けてオブジェクトグループとして管理をすることもできる。あるグループにメッセージを送信するには、オブジェクトマネージャを中継点としてメッセージの送信を行なうことで、一対多の通信を行なう。

3.3 宛先オブジェクト

宛先オブジェクトもオブジェクトマネージャと同様、Communicatableクラスの継承する。宛先オブジェクトの機能をCommunicatableクラスから切り離すことで一対多通信の際の並行オブジェクトへの負荷を減少させることができる。

4 まとめ

本稿では、並行オブジェクト指向言語yの実行時環境の実現について述べた。UNIXのプロセスを並行オブジェクトとして機能させるために必要な機能を、特に通信に焦点をおいて、C++のクラスとして実現し、オブジェクトマネージャやその他の機能をこのクラスを用いて実現した。

今回、一般的なUNIXの機能だけを用いて分散プログラミングの環境を実行時ライブラリとして実現したが、分散プログラミング言語の実行時環境は、それ自体が分散プログラムの典型である。今後はこの環境上でyを用いて実行時環境を整備すれば、yの分散処理のための機能を評価することができるとと思われる。

参考文献

- [Yon86] 米澤明憲, 柴山悦哉, Briot J. P., 本田康晃, 高田敏弘: オブジェクト指向に基づく並列情報処理モデル ABCM/1 とその記述言語 ABCL/1, コンピュータソフトウェア 3, 3, 1986, pp. 9-23.
- [Koj93] Kojima, K. and M. Noro: The Design of the Object-Oriented Concurrent Programming Language y, in *Proceedings of the Joint Conference on Soft. Eng. '93*, IPSJ, pp. 59-66.
- [Cho95] 崔 梗涑, 小島 一人, 野呂 昌満, 原田 賢一: “並行オブジェクト指向言語yの通信機能,” 情報処理学会第50回全国大会講演論文集.
- [Koj95] 小島 一人, 野呂 昌満, 崔 梗涑, 原田 賢一: “並行オブジェクト指向言語yにおけるクラス階層の定義と継承,” 情報処理学会第50回全国大会講演論文集.