4P-2 並列オブジェクト指向言語A-NETLのプログラミング支援環境

加賀谷 学　浜田 正泰　吉永 努　馬場 敬信
(宇都宮大学工学部)

1．はじめに

並列オブジェクト指向言語においてはモジュールをオブジェクトと考え，その間のメッセージ交換で全体の処理が記述できるため，並列処理を実現するのに適している[1]．しかしその作成に当たっては，プロセス間の時間依存性・非決定性等に起因する並列言語固有の問題を解決しなければならない．

我々は，研究室で試作中の並列オブジェクト指向を核概念とするトータルアーキテクチャA-NET(Actors-NETwork)[2]を対象として，その言語A-NETL(A-NET Language)のプログラミングを支援するシステムを，設計・試作した．本稿では，その設計方針・基本機能について述べる．

2．A-NETLによるプログラミング

A-NETLは並列オブジェクト指向概念に基づき設計された言語であり，次のような特徴を持つ．

1)オブジェクト単位の並列処理の記述 ： プログラムは，オブジェクトと，動的に生成するオブジェクトを定義したクラスの集合で記述される．物理的プロセッサにはオブジェクト，クラス単位で割り付けられ，これが並列処理の単位となる．

2)メッセージ・パッシング ： メッセージにはABCL／1[1]と同様の過去，現在，未来の型がある．また，協調機構のためにメッセージマルチキャストや，複数メッセージの待ち合わせとそれに対する複数リターンがある．それらを利用することにより複数のオブジェクトを同時に活性化できる．

3)動的なオブジェクトの生成，消去，変更 ： 実行時にオブジェクトの生成・消去ができる．また，メソッド単位の追加・変更が可能である．

プログラミング支援環境に対する使用者の要望を明らかにするため，A-NETLのプログラミング演習を数人に行ってもらった．得られた意見は次の通りである．

- あるオブジェクトの記述中に他のオブジェクトの内容を見たい．
- オブジェクト間のメッセージのやりとりを，メッセージの種類も含めて図形的に表示してほしい．
- 未来型メッセージを送った場合，受け側がリターン文を記述していることを確かめたい．
- 各オブジェクトごとの受理可能なメッセージパタンのリストを見たい．

以下，これらの意見をもとに設計したシステムについて述べる．

3．支援環境

3．1 設計方針

A-NETL では，ユーザが並列実行を意識してプログラム記述を行う．本システムは，”並列”プログラムの作成支援という立場から，次のような設計方針のもとにシステムを構築した．

1)並列処理の単位であるオブジェクト単位で，テキスト，メッセージ等を入力・表示・操作する．

2)オブジェクト間の関係を考える上で中心となるメッセージに注目し，受理可能なメッセージパタン，メッセージの実行履歴等の表示や，オブジェクトとメッセージの図形表示を行う．

3)マルチウィンドウ，アイコン，マウス等を利用し，視覚的にプログラムの作成を助ける．

4)エディタ，コンパイラの起動，メッセージ検索，デバッガなどを統合したシステムとする．

システムはUNIX4.2BSD上にXウィンドウを用いて実現した．以下に実現した基本機能について述べる．

3．2 基本機能

1)受理可能メッセージの入力・表示

各オブジェクトごとに，受け付けられるメッセージパタンのリストを入力・表示する．オブジェクト指向の概念から，オブジェクトはその通信方法であるメッセージパタンのみを外部に公開すれば良い．現在は，作成者がオブジェクト作成時にメッセージパタンを入力しているが，将来は，支援システムがコンパイル時にメッセージをオブジェクトのテキストから読み取るように拡張する．

表示形式を図1に示す．”NOT”，”!”，”!<EXP>”は，それぞれリターン文なし，有り（式なし），有り（式有り）を表す．

object1	
NOT	mess1
	mess2 & mess3: arg1 & mess4 †
!	mess5: arg2
!<EXP>	3 * mess6: arg3 ††
	mess7 & mess8: arg4

†異なるメッセージの待ち合わせ
††同一メッセージの待ち合わせ

図1　ﾘﾀｰﾝの有無と受理可能ﾒｯｾｰｼﾞﾊﾟﾀﾝの表示

Programming Support System for a Parallel Object-Oriented Language A-NETL
Manabu KAGAYA, Masahiro HAMADA, Tsutomu YOSHINAGA, and Takanobu BABA
Utsunomiya University

2)転送先オブジェクトの入力・表示

オブジェクト間のメッセージ・パッシング関係を示すために，各オブジェクトごとにメッセージを送る可能性のあるオブジェクト名のリストを作り表示する．また図2に示すように，アイコンをオブジェクトとクラスに見立てて図形的に表示する．画面上に複数のオブジェクトと，それらのメッセージ送信関係を表示することにより，プログラマの並列的思考を助ける．図中，破線はメッセージマルチキャストであること示す．これも受理可能メッセージの入力と同様，支援システムがテキストからオブジェクト名のリストを作るように拡張する．

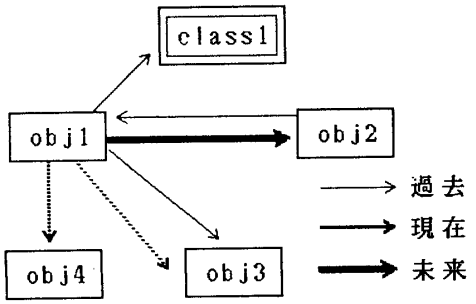

図2　メッセージ転送先オブジェクトの表示

3)既存プログラムの再利用

オブジェクトはモジュール性が高いため，その再利用が容易である．全ての情報をオブジェクト単位のファイル構成とし，また，プログラム・オブジェクト・クラス名をメニュー形式で表示することで，その管理・編集を支援する．その際，関係の深いオブジェクト群ごとにグループ化して表示する．（プログラム実行時には，関係の深いオブジェクトは同一または近傍のプロセッサに割り付けられる．）

4)説明文

プログラムのテキストを見ただけでは，その動作は分かりにくい．特にオブジェクトの再利用を図るときなどは，テキスト自体よりも大まかな内容を知りたいことがある．そこで，オブジェクト・クラスごとに，オブジェクトの説明文を入力・表示する機能を設けた．

5)メッセージの検索

オブジェクト間の関係を調べる場合，最も重要なのはメッセージパッシングの関係である．特定のメッセージパタンを持つオブジェクトを検索するコマンドを用意した．

6)コンパイル

オブジェクト・クラス単位でA-NETLコンパイラにテキストを送り，分割コンパイルを行う．

7)その他

ディスプレイ画面の制御，ユーザのディレクトリ変更，テキストの出力，一時UNIXのコマンドに戻る機能等を用意した．また，オブジェクトやクラスの作成の際には，ウィンドウ中の初期画面に必要な構文を表示し，テキストの入力を助ける．

3．3　デバッグ支援機能

並列プログラムのデバッグに関しては，プロセス間の相対的な速度が影響するため動作の再現が難しい．一方，オブジェクト指向プログラムでは，1)オブジェクト間の情報伝達手段はメッセージの送受信のみである，2)基本的にメッセージの受信順序により実行の順序が決定される，等の特徴がある．つまりメッセージのやりとりを調べることでプログラムの実行の概略をつかむことができる．そこで，実行時にオブジェクト間のメッセージの履歴を取り，動的なメッセージ・パッシング等の関係を調べてデバッグに利用することにした．基本機能を以下に挙げる．

1)履歴を取るオブジェクトの指定

高並列を目指していることから，多数のオブジェクトが同時に動作する可能性がある．履歴を取るオブジェクトの指定は，実行前から存在するオブジェクトについてはオブジェクト名で，動的に生成されるオブジェクトについてはクラス名で指定する．

2)履歴の表示

a) 図3の様に，指定したオブジェクト単位にメッセージ・パッシングが行われた双方のオブジェクト名，メッセージの種類（P:過去，N:現在，F:未来，R:リターン，@:マルチキャスト），メッセージの内容，通信時刻等をリスト形式で表示する．

b) ディスプレイ画面上に，アニメーションによる実行の様子を再現する．途中で止めて，転送中のメッセージの内容を表示することも可能である．

c) 特定のメッセージ・パターンを指定・検索・表示する．

3)メッセージ・パッシングの頻度を測定・表示する．これによって特定のオブジェクトにメッセージが集中するなどのボトルネックを指摘する．

obj1	P → mess1: arg	obj2	Time1
	@ P → mess2:	obj3	Time3
	R ← messR:	obj2	Time4
obj2	P ← mess1: arg	obj1	Time2

図3　メッセージ履歴の表示

4．おわりに

充実したプログラミング環境とするには，以上に述べた機能をベースに更に検討を重ねる必要がある．現在，プログラム実行時のデバッグ支援機能として，オブジェクトやメソッド単位に状態の表示や書換え等を行うデバッガを試作中である．システムができ次第プログラマに使って貰い，評価・改良する予定である．

本研究は一部文部省科学研究費（課題番号62550255オブジェクト指向高度並列ＡＩマシンに関する研究）の援助による．

[参考文献]

[1] 米澤：”並列オブジェクト指向言語ABCL/1による並列処理記述とその枠組みの研究”，信学論，Vol.J71-D(1988)

[2] 馬場，他：”並列オブジェクト指向トータルアーキテクチャA-NET”，並列処理シンポジウム '89,A4-1(1989)