

並列オブジェクト指向システム ORAGA  
—デバッグシステム ObjectPeeper

6F-2

阿部 雅彦 神田 陽治 田中 英彦  
(東京大学 工学部)

1. ObjectPeeper

ObjectPeeperはORAGAシステムのユーザインタフェースとなるもので言語DinnerBellによるプログラミング、デバッグを行うツールとなる。ObjectPeeperの持つデバッグ機能としてはオブジェクトの表示とメッセージ問合せの2つがある。現在実験システムを作製しているが、以下ではObjectPeeperの実行例およびインタプリタとの結合方法について述べる。

2. インタプリタ概略

インタプリタの実行は基本的には、コンテキストを取り出してはトランスフォームすることである。トランスフォームをすると新しいコンテキストができるので、それをまたプールに入れ、以下同じことを繰り返す。

この実行と、オブジェクトがメッセージを送り合うというユーザから見たモデルとの関係は次のように考えられる。

- ・コンテキストをプールから取り出してトランスフォームを始めることはメッセージの受信に対応する。
- ・トランスフォームを行うと新たに複数のコンテキストが生成される。これは、メソッドの発火に対応する。
- ・新しくできたコンテキストをプールに入れるのはメッセージの送信と考えられる。
- ・オブジェクトの生成の際にはメモリに itBlock変数の領域がとられる。
- ・メソッドが発火する毎にメソッド変数の領域が新しくとられる。
- ・メソッドが実行中かどうかは、発火によってできたコンテキストがプール内に残っているかどうかでわかる。全コンテキストがトランスフォームされれば実行は終了する。

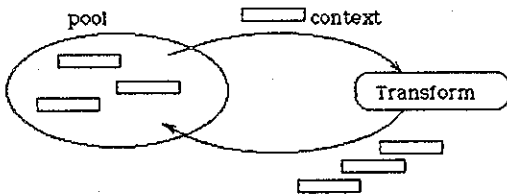


図1 インタプリタによる実行

3. オブジェクト表示

3.1 概略

オブジェクトはウィンドウの大きさを有効に活用するような処理をされて表示される。ObjectPeeperではクラスご

とにバレットと呼ばれるテンプレートを作り、オブジェクトの表示方法を指定することにより、ユーザにとってわかりやすい表示を実現する。

3.2 実験システム

Macintosh上にSUMacCを使ってC言語で実験システムを作製している。このシステムはオブジェクトのデータをファイルから読み込んで表示を行う。次にいくつかの例を示す。

図2はバレットである。このバレットに基づいてクラスForkに属するオブジェクトを表示したのが図3である。各変数名の隣の整数値は代入されているオブジェクトのidである。ウィンドウ最下段の2つの getは現在セクタ名が getのメソッドが2つ発火中であることを示している。オブジェクトの表示は、ウィンドウを開いた段階ではメソッド間で共有される itBlock変数のみが表示され、一方のメソッドを選択すると、新たにmethod変数、statement変数が表示される。

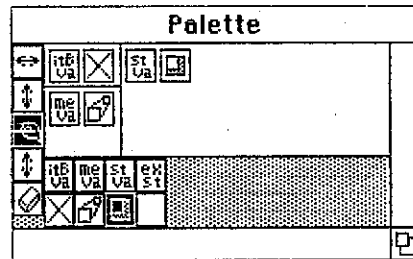


図2 バレット

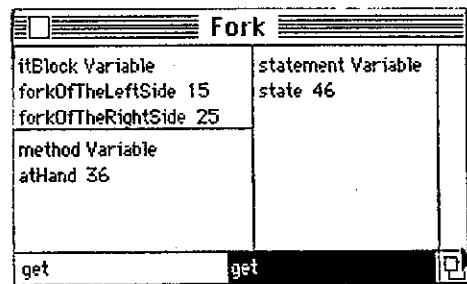


図3 オブジェクトの表示

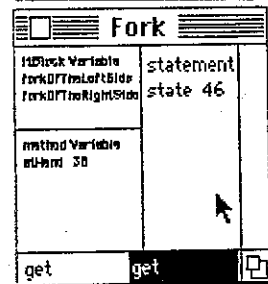


図4 圧縮の例

† SUMacCはStanford University Medical Centerで開発されたツールである

図4はウィンドウのサイズが小さかった場合である。部分ごとに違う圧縮方法がとられている。

### 3.3 インタプリタとの関係

このシステムとインタプリタの間はファイルを通してつながれている。インタプリタは実行中の適当な時点でファイルにオブジェクトの状態を書き出す。出力されるファイルは3種類あり、1つはクラスの情報、2つ目はオブジェクトとクラスの関係、3つ目は各オブジェクトの情報である。

## 4. メッセージ問合せ

### 4.1 概略

メッセージがオブジェクト間でどのように送受されたかを調べるにより、実行の様子を把握することを目指している。プログラムの実行時にデータを蓄え、そのデータに対し、問合せを出すことによって解析する。問合せ方法としてはQBE (Query By Example) のような方法を考えている。

### 4.2 実験システム

VAX上にC言語で作製している。現在、簡単な問合せに答えることができる状態にある。次に実行例を示す。これは図5のような場合のメッセージ送受について問合せを行ったものである。この図は例えば、クラスXのid1のオブジェクトから、クラスDateのid2のオブジェクトへ、メッ

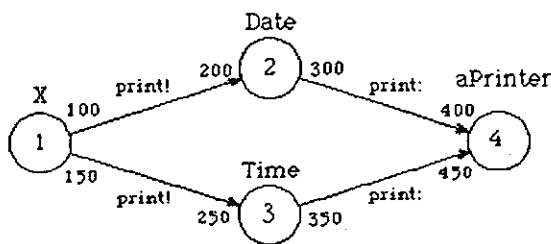


図5 メッセージ送受の例

Query	source class	id	message tran	rec	destinate class	id
	X	1	print!	100 200	Date	2
	X	1	print!	150 250	Time	4

図6 基本的問合せと答

Query	source class	id	message tran	rec	destinate class	id
	X	1	print!	100 200	Date	2
	Date	2	print:	300 400	aPrinter	3
Message pass 1	X	1	print!	150 250	Time	4
Message pass 2	Time	4	print:	350 450	aPrinter	3

図7 パスの問合せと答

セージ print! が時刻 100に送信され、時刻 200に受信されたことを示している。

図6は最も基本的な問合せで、オブジェクトid1のオブジェクトから送信されたメッセージを取り出しなさい、という問合せである。このほかに、受信側のみを指定したといあわせや、クラス名のみによる問合せも可能である。

図7はこのシステムで特徴的な問合せで、メッセージのパスを調べる。問合せ中の.1\_2がパス探索の指示になる。この例では図5の上側のパスと下側のパスがみつまっている。この機能により、並列実行時に問題となる、離れた場所でのオブジェクトの共有の発見が容易になる。

### 4.3 インタプリタとの関係

このシステムもインタプリタが出力したファイルを受け取って働く。インタプリタはコンテキストのトランスフォームのたびに、そのデータを内部に蓄えていく。実行が終了した時点で、これらのデータはファイルに書き出され問合せシステムへと渡される。

## 5. 今後の課題

- ・表示システムと問合せシステムの結合
- ・より図式的な表示
- ・動的デバッグ実現についての考察

## 参考文献

- (1) 情報処理学会第29回全国大会, 4R-6 (1984)
- (2) 情報処理学会第30回全国大会, 5T-5 (1985)
- (3) 情報処理学会第31回全国大会, 1E-8 (1985)