統合プログラミング環境 ＯＲＡＧＡ（Ⅲ）
1E-8 ユーザインタフェース ObjectPeeper

阿部 雅彦　神田 陽治　田中 英彦　元岡 達

(東京大学 工学部)

1．はじめに

ObjectPeeperはＯＲＡＧＡシステムにおいて，エディタ，デバッガ等のユーザインタフェースを担当する。ここではこのObjectPeeperについて，デバッガとしての機能，実験システムの構成等について述べる。

2．ObjectPeeperのデバッギング機能

ObjectPeeperは，オブジェクトを表示する機能と，メッセージに関し，問合せを行う機能の２つの機能を持つ。オブジェクト表示はマルチウィンドウを用い，１オブジェクトに対し，１ウィンドウを開いて表示を行うものである。一方問合せ機能ハードウェア主に同期の検証を目的として計画しているが，現在はまだ実装されてはいない。この機能は，メッセージ送受について，該当オブジェクト，送受信時刻等を記録し，これについて，データベースの問合せ言語のような方法で調査を行うものである。オブジェクト指向言語では，メッセージの送受を調べることにより動作の様子を把握できるが，これを計算機で支援することにより，デバッグ作業の能率向上を期待している。

3．問合せ機能

どのオブジェクトからどのオブジェクトへ，どんなメッセージがいつ送信され，いつ受信されたかを，効率よく調べるための機能がこの問合せ機能である。今のところ詳細は未定だが，方法としてはＱＢＥ（Query By Example）のようなものを考えている。以下に例を用い，目ざしている機能について説明する。

(1) １レベルの問合せ

図１は最も簡単な問合せの例である。a）が問合せ，b）がそれに対する応答である。個々のオブジェクトはオブジェクトidによって識別される。左側が送信側，右側が受信側のオブジェクトである。

class name	id	message	t time	r time	class name	id
diningPhilosopher	p.1	left right id	p.10		Philosopher	p.2

a)

class name	id	message	t time	r time	class name	id
	12		8			7
	13		9			8

b)

図１　簡単な問合せ例

(2) メッセージ・パスの表示

並列に動くオブジェクトの間での同期の問題としては，図２のようなオブジェクトの共有が考えられる。このような場合，Ｃに副作用があると，Ａ，Ｂからのメッセージの到着順序が問題となる。もし，ＡとＢが互いに直接知っていれば，メッセージを交換して制御できる。一方直接知らない場合には，この共有を引き起すもととなったオブジェクトＸを捜し出し，同期の制御を行わせることになる。　以上のようなとき，Ａ，Ｂが直接知りあいかどうかは，オブジェクト表示において変数を調べればわかる。一方直接知らない場合は図３のような問合せによってメッセージ・パスを調べＸを捜し出すことができる。

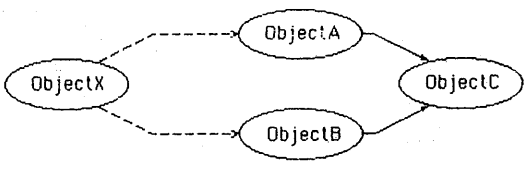

図２　オブジェクトの共有

class name	id	message	t time	r time	class name	id
p.classOfX	p.1	p.mes1 (2 L)			p.classOfA	2
classOfX	1	mes2 (3 L)			classOfB	3

a)

class name	id	message	t time	r time	class name	id
classOfX	11	mesToX'			classOfX'	
classOfX'	12	mesToA			classOfA	
classOfX	11	mesToB			classObB	

b)

図3　メッセージ・パスの問合せ

図3の場合もa）が問合せ，b）が応答である。a）の中でmessage の列にある（2 L）の指示はXからAの間に任意の数のオブジェクトが存在することを許す指示である。この結果えられたb）はXからA，Bへのメッセージ・パスを示している。

4．実験システムの構成

現在ObjectPeeperの実験システムをMacintosh 上に作製する作業をしている†。これはC言語で記述され，Macintosh のもつ画面制御機能を利用して作られている。今のところまだ一部しか実装されていないが，別に作製中のDinnerBellインタプリタと結合する予定で，図4のような構成をとる。動作の方法は次のようになる。まずホストのVAX 上でソースプログラムを作り，これをコンパイラで中間コードに落とした後Macintosh に転送する。Macintosh 上では，中間コードはインタプリタによって実行される。

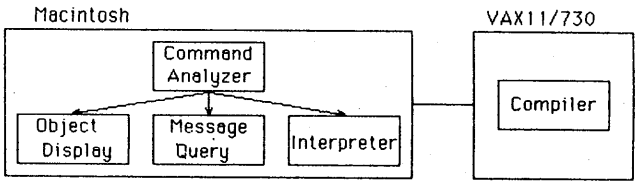

図4　システムの構成

Macintosh 上のシステムは4つの部分から構成されているが，このうちCommand Analyzerが各モジュールの実行を制御している。ユーザからのコマンドはここを通して各モジュールで受け付けられ，インタプリタでの実行，オブジェクトの表示，メッセージについての問合せ等が行われる。

5．今後の課題

(1) 時間の定義：問合せの中に時間の項目があるが，これをどのように決めるかは問題である。最初の実験はシングルプロセッサ上のインタプリタでの実装を考えているので問題は少ないが，ＯＲＡＧＡ計画はマルチプロセッサでの実現を目ざしているので，このときどのような時間を用いるかを考えなければならない。

(2) 問合せ仕様の詳細：今回示した例はまだ試案の段階であり，今後検討する予定

(3) 記憶しておく情報の選択：実行のすべての記録を記憶するのは負担が大きい。一方，不充分なデータからは満足のいくデバッグはできない。必要な情報だけを記憶しておく方法についての検討が必要である。

《参考文献》

〔1〕阿部　他『オブジェクト指向言語に向いたディスプレイ・エディタ』
情報処理学会第29回全国大会，4R-6 (1984)

〔2〕阿部　他『"DinnerBell+NameMaster+ObjectPeeper"による統合プログラミング環境
—ユーザインタフェース』
情報処理学会第30回全国大会，5T-5 (1985)

〔3〕C.J.Date『データベース・システム概論』

† Macintosh でのシステム開発にはStanford University Medical Centerで開発されたSUMEX MacC UNIX Development Kit を用いている。