ＰＳＩ上のプログラムブラウザについて

2H-1

明石孝祐　青柳龍也　田中英彦
（東京大学　工学部）

１．はじめに

システムに組み込まれたライブラリはオブジェクト指向言語にとって大きな構成要素である。しかし、各モジュールとしてのオブジェクトは独立性が高いために、逆にプログラムの再利用などを目的として理解しようとする場合、相互の関連が捕らえにくくソースレベルでの把握は困難である。Smalltalk などではデバッガ・インスペクタを使って逐次的に追うことになる。

本稿では、このような事態を少しでも改善するために、トレースの結果を元に、大域的な情報を提供する試みについて報告する。

２．集める情報

ここでは、大域的な情報として、プログラムを読む際に必要となる情報を考える。

①　メソッド・スロットのタイプ

オブジェクト指向言語ではダイナミックなメソッド呼出しの利点を利用するために、静的にタイプを付加することは少ない。これはプログラミングを柔軟にするいっぽう、ソースを読むときには障害となる。タイプインファレンスの研究もあるが、十分な成果はまだ得られてない。とりあえず、実行結果からタイプ情報を得ておくことは有用であろう。

②　スロットでのオブジェクトの共有

Smalltalk には共有変数という概念があるものの、動的には異なるオブジェクトのスロットに同じオブジェクトを保持することが生じ得る。このようなプログラムは読みにくく避けるべきであろうが、共有変数がない言語では特に問題となる。これはトレースの結果からそのスロットを見つけるしかない。

③　引数にくるオブジェクトの発生元

メソッド呼出しにおける引数渡しは一種のcall by reference であるから、オブジェクトは状態を持ち歩いているとも考えられる。メソッドの実行はそれをスロットに持つオブジェクトの状態を変える可能性があり、オブジェクトの発生元を知ることは、影響範囲を調べる手掛かりとなる。

④　引数のデータ依存性

ここで言うデータ依存性とは、データフローではない。オブジェクトは副作用を持つのでどんなメソッドを受けたかが重要である。それを順に集めてきたものがデータ依存性である。

３．PSI 上の実現

２にあげた情報を抽出するプログラムをPSI 上のオブジェクト指向言語であるESP を対象に作った。

そのデバッガは、dummy-debuger というクラスを継承し書き換えることでユーザがカスタマイズできる。これを使ってトレースの情報を得た。また、ソースの表示にはエディタPMACS を使った。

３．１　ソース・ビューア

これは２，①のタイプを表示させるためのツールである（図１）。タイプ、あるクラス名で表わす。引数及びスロットの値として可能なクラスのうち継承関係の中で一番上位のクラスを調べることになる。メソッド名・スロット名をマウスで選択することでタイプビューを表示する。

３．２　デバッガ

トレースをとること、及び２，②、③、④に対するビューを表示するためのツールである。

②のスロットの共有はトレースを終り次第表示する（図２）。ESP には共有変数がないので、この情報はある種のプログラムをよむときには便利である。

③の発生元とは、オブジェクトが引数に最初に束縛された位置をいう。オブジェクトが出力変数に束縛された

Program-browser on PSI
Takahiro AKASHI, Tatsuya AOYAGI, Hidehiko TANAKA
Tokyo University

場合と、スロットから参照した場合とを捜し、それを実行したオブジェクトとともに示す。

④はトレースの途中でメソッドの一引数を選択することでメソッドを集めて列挙する（図３）。

４．検討

今回作ったツールはSIMOS のエディタ・デバッガに付加したものであり、どういう形で情報を蓄積し、わかりやすい表示インターフェースを与えるかは今後考えなければならない。

オブジェクト指向という概念は、言語の持つ他の要素と比較的独立であると思われるが、実際に情報を抽出するとなると無視できない。ESP の場合、オブジェクト以外のリストやベクタのデータ構造があること、prologの持つ制御構造、引数に状態を引きずるプログラムの書き方などは、どういう形の情報を取り出すべきか難しくしている。

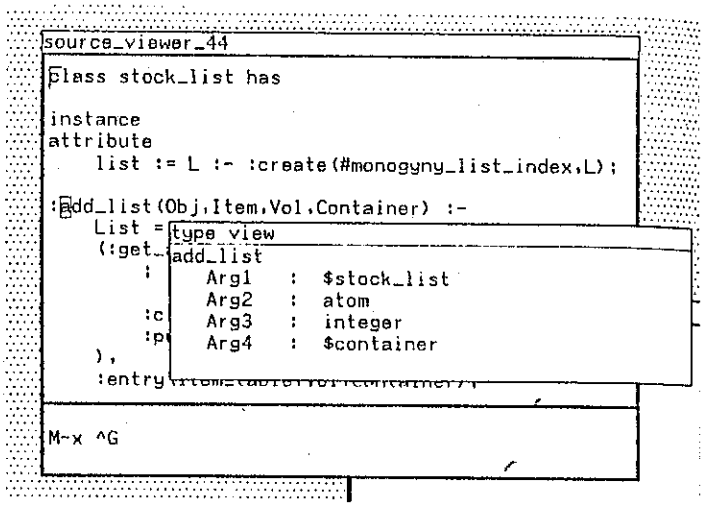

図１．　ソースビューアとタイプ情報

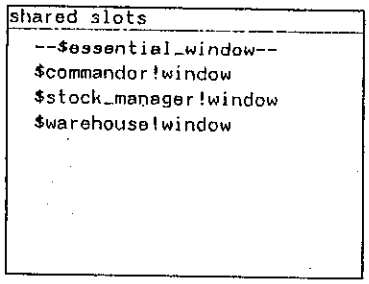

図２．　オブジェクトの共有

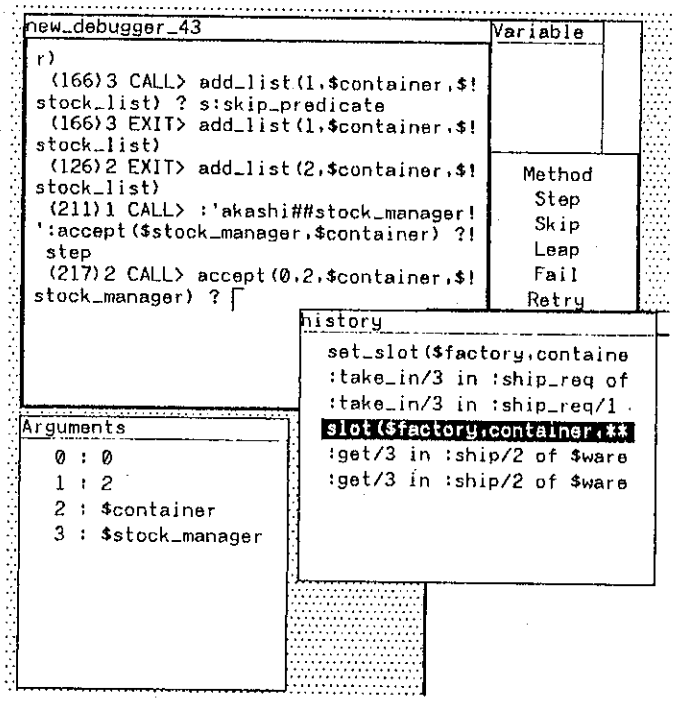

図３．　デバッガと
引数オブジェクトの発生元・依存性