

SIMPOSのプログラミング環境  
-インスペクタ-

## 4D-4

金枝上敦史  
(財) ICOT)

石橋弘義

1.はじめに

逐次型推論マシン(PSI-Personal Sequential Inference machine)上に開発されたプログラミング/オペレーティングシステム(SIMPOS)の”インスペクタ”について述べる。

PSIのプログラム言語ESP(Extended Self-contained Prolog)はオブジェクト指向言語であり、SIMPOSおよびユーザー・プログラムはESPのクラスを基本単位として構成されている。そして、プログラムの実行時にオブジェクトが生成され、オブジェクトが実行の主体となる。

インスペクタはこのオブジェクトの内部状態を解析し表示する機能を持つ。

2.概要

ESPには論理型プログラミング言語の性質とオブジェクト指向プログラミング言語の性質があり、プログラムの流れは、論理の流れとオブジェクトの状態の変化として捉えることができる。そして、この論理の流れをユーザに見せるためのシステムとしてデバッガが提供され、オブジェクトの変化を調べるシステムとしてインスペクタが提供される。

オブジェクトを一言で表わすとデータに手続きを付加したものである。つまり、オブジェクトに対してある手続きを要求することにより処理をすすめてゆくことができる。オブジェクトにはスロットと呼ばれる変数があり、スロットの値の変化は、オブジェクトの状態の変化を意味する。例えば、スロットの値として制御用のテーブルが割り当てられている場合に、このスロットの値を変えることはこの制御テーブルを書き換えることになり、プログラムの流れが変わることになる。

ユーザはインスペクタを使ってオブジェクトの状態を見ることにより、プログラムが正常に流れているかを調べたり、オブジェクトの状態を変えることによりプログラムの流れを変えたりすることができる。この様にインスペクタはプログラムのデバッグやシステムの内部状態を調べるためのツールとして使われる。また、インスペクタはPSIの機械語であるKLOの各データを解析する機能も持っている。つまり、インスペクタはオブジェクト以外のデータに対し

てもそのデータを解析したり、書き換えをおこなったりすることができる。

3.機能、特徴

インスペクタの機能としてまずオブジェクトの解析機能とKLOデータの解析機能の2つがあげられる。オブジェクトをインスペクトした場合、そのオブジェクトが持っていたスロットのスロット名と値を表示する。KLOデータの構造型データ(例えば、ベクタ)をインスペクトした場合、そのデータを構成する要素を表示する。

今回開発したインスペクタの特徴として、データを見やすく表示するためにメニュー化したり、マウスを使用することにより操作を簡単にした点があげられる。データ表示は、各データのタイプをその属性と共に示し、構造型データの場合にはその各要素をメニューで表示する。それらの要素値をさらに解析するにはこのメニューに表示された各要素をマウスにより指定するだけで構造を順にたどることができる。

4.リスト・オブジェクトのインスペクト例

SIMPOS上にはいろいろな機能を持つオブジェクトがある。ここでは制御や資源の管理によく使われるリスト・オブジェクトについてのインスペクト例を示す。

リスト・オブジェクトは、データをキュー(Queue)の様に格納するためのオブジェクトである。そして、インスペクトの方法として3通りの方法がある。1番目はキューとしての見方である。この方法によりユーザはオブジェクトの構造を意識することなく格納された全データを調べることができる。そして、もっとも一般に使用される方法である。2番目はオブジェクトの構造を意識した調べ方で、スロットの値を調べてゆく方法である。一般にオブジェクトはこの方法で解析され、リスト・オブジェクトなどについては特に1番目の解析方法をサポートしている。3番目はオブジェクトを形成するスロット・テーブル、メソッド・テーブルの内容を調べる方法である。

以下に、リスト・オブジェクトのそれぞれの見え方を図示する。

① キュー

指定されたデータを表示したり、一度に全データを表示したりする。この時、ユーザは図1の様にデータがキュー状に格納されていると考えるだけでよい。

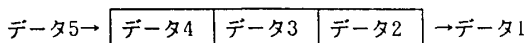


図1. キューとしてのリスト・オブジェクト

② オブジェクト

スロット名とその値が表示する。リスト・オブジェクトの構成は、最初のエントリを示すスロットと最後のエントリを示すスロットの2つのスロットによりなる。そして、各エントリはエントリ・オブジェクトにより形成される。エントリ・オブジェクトはエントリのデータを格納するスロットと次のエントリを格納するスロットで構成される。

(図2参照)

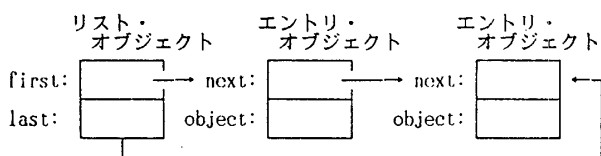


図2. リスト・オブジェクトの構造

③ 構造体(ベクタ)

PSI 上では、オブジェクトを(ヒープ)ベクタにより実現している。オブジェクトは状態を表わすスロットのテーブル(スロット・テーブル)と操作を表わすメソッドのテーブル(メソッド・テーブル)で構成されている。

実際には、オブジェクトを示すヒープ・ベクタにスロット値を持ち、さらにオブジェクトを記述するスロット・テーブル、メソッド・テーブルへのポインタを持った構造体を形成している。リスト・オブジェクトを生成した場合には\*のベクタだけが新たに生成される。(図3参照)

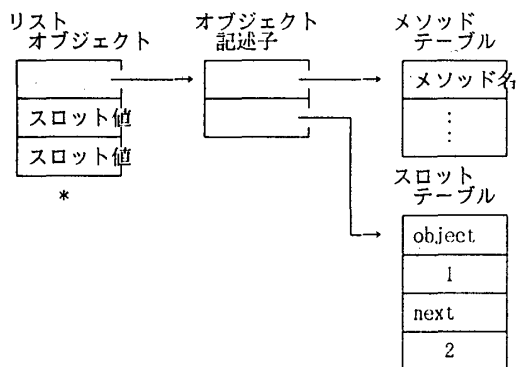


図3. オブジェクトの内部表現

これらの3つの見方はこのリスト・オブジェクトに対して何を調べたいかというそれぞれのユーザの目的によって違ってくる。単にデータの格納に使うだけならば①の解析方法で十分である。②のような解析の方法をおこなうのは

プログラム開発者がオブジェクトをデバッグする時である。③の解析方法はよほど特殊なユーザに限られる。例えば、この方法で、メソッド・テーブルに登録されたコードを調べることができる。

SIMPOSにはリスト・オブジェクトの他にデータを格納するためのいろいろなオブジェクトが用意されている。例えば、ハッシュをおこなうことによりデータの検索スピードを高速化したハッシュ・インデックスがある。このオブジェクトを解析する場合には解析している人がハッシュ値を知らないでも①の方法ですべてのデータを調べることができる。

5. デバッグ・ツールとしてのインスペクタ

PSIはオペレーティング・システムからプログラミング・システムまでを単一言語ESPで統一し、さらにユーザ言語として広く用いている。このことはSIMPOSがユーザに対して開かれたシステムであることを意味する。よって、ユーザは容易にSIMPOSの機能を使ったり拡張したりすることができる。さらに、インスペクタによりシステム資源の状態をオブジェクトという単位でSIMPOS、ユーザ・プログラムの区別なく調べることができる。つまり、デバッグ中に対象となるオブジェクトをインスペクタでたどることができれば、SIMPOSの内部状態を調べたり、書き換えたりすることが可能となる。

ここで、インスペクタによるデバッグの例をあげる。あるシステムでリスト・オブジェクトをデータ格納のために使用し、この格納データを他のシステムから参照したり、加工したりする様なプログラムを考える。

この場合のデバッグは、デバッガでプログラムをステップ実行しているだけでは全体のデータの流れがわからない。そこでインスペクタを使うことによりリスト・オブジェクトを解析しデータの出入りや変化を知ることができる。

前にも説明した様にデバッガは論理の流れを調べる役目があり、そしてインスペクタはオブジェクトの状態変化を調べる役目がある。つまり、お互いが補完しあひながらはじめて完全なデバッグが可能となる。

6. おわりに

現在のインスペクタはデバッガとの間に密接な関係を持ち、ESPによるプログラミングにおいて強力なデバッグ・ツールとして働き、欠くことのできない存在となっている。

そして、この様なインスペクタはESPだけではなく他のオブジェクト指向言語においても重要な位置付けにある。