

オブジェクト指向計算機のメカニズムに

6B-4

関する一考察・・・言語環境

金子誠司、神田陽治、河野真治

田中英彦、元岡達 (東京大学)

Smalltalk-80風のオブジェクト指向のプログラミング・スタイルは魅力的なものはあるが、言語の仕様が逐次形計算機向きと存、ていて、並列性の導入は難しい。本発表では、Smalltalk-80に対して制御抽象の考え方をとり入れ、並列向きとした言語Dcunerbellの概要を紹介する。

2. Smalltalk-80型の言語

関係のある点のみ抜き出した。

A) Smalltalk-80では実行はすべてSystemicに1つあるInterpreterが行なう。Interpreterは、各Objectの内部コードを実行し、メセージを送ることにより、相手は制御が移る。これは既存の言語の関数呼び出しと実質上同じで、Actor model [4] にいう並列性はとり出せていない。また、例外処理はInterpreterが受けるが、このためObjectが制御に関して独立とは、ていない。

B) Smalltalk-80ではObjectのTemplateとInstanceは同じ扱いを受け持っている。また、Inheritanceはテラクシ性による依存関係であり、わずかに異なる性質をもったObjectも他と同様にsubclassとして、性質による分類される。この方式ではObjectの種類が多くなるとともに、そのObjectを用いるObjectとの関係はわかりにくくなる。この問題を解決するために、Smalltalk系言語ではMultiple-Inheritanceを導入して、親子もあつたが、互いの関係問題に悩まされることになる。

ベルでのselectorにmatchするmethodが複数ある場合がある。いくつかの言語では記述順などの情報をもとにして1つに絞られているが、必然性には欠ける。

3. 本言語の概要

本言語はSmalltalkに似た、並列処理用言語である。Smalltalk同様、本言語でもSystem-Browserを用いてObjectの記述を行うため、表現の見やすさは追っていない[2]。並列は文単位のものとし、必要に応じて処理の順序を決めるための記述を加える形をとった。

Objectの記述は、その名前、Static記述部、Dynamic記述部よりなる。Static記述はStatic側のInheritanceとlocal classの宣言、Instance変数とmethodの記述よりなる。Dynamic記述はDynamic側のInheritanceとState変数の宣言、eventの記述よりなる。

A) Static記述とDynamic記述の相異

Static記述はObjectのmethodに対するふるまいを記述し、Dynamic記述はStatic記述による実行に対するメタ情報を与える。Actor modelにあてはめれば、Static記述は実行部に対する情報、Dynamic記述は、Bankerなどのサポートに与える情報に各々相当する。

B) Local Class

Local Classとは、各class2局所的に使用するclass2ある。単に何らかのObjectを用いるというものでなく、Multiple Inheritanceを用いてある特殊な性質をもつObjectを生成することが出来る。これにより、

このObjectを用いるときの状況を細かくObjectのclassに反映させることができ、また全系で共有しているクラスを不必要に増加させない。使い方のとしてはあるclassから共有が無意味な特殊なsubclassを作り出すことや、Dynamic記述によってそのObjectを用いるときの順序制御情報を与えるなどがあげられる。

c) Inheritance

Multiple inheritanceを採用する。前述のように、super階級のselectorが重複している場合の問題であるが、ここでは対応したmethodはすべて動かすという立場をとる。すなわち

- あるclassでmethod探索が成功したとき、そのより上位のmethodは探索しない。
 - searchはdepth-firstを行ない、発見されたmethodはすべて実行する。
- この方針は、あるselectorに対してこのようにふるまうというものがObjectの性質とみなす解釈、及び一つのClassに同じselectorに複数書いたときはすべて実行の自然である、などによる。この方針のもとに記述によってこれを実行するか決める。

d) Event 記述

実行時の制御情報としてEventの発生はMessageの到着と同じ扱い、記述も同様のsyntaxがある。Eventの例としては、そのInstanceの生成、外部からの割り込み、実行時のエラー、Messageの送交や変数の代入などがあげられ、初期化の手続きも記述できる。またStatic記述中からeventを発生させる。それによって順序制御を行なうことができる。また...

e) Metaclass 記述の削除

初期化手続きをDynamic記述側に行きかたため、Metaclassを各Objectにおく意味が薄れたため、削除しMetaclassは全系で一つ、つまりTempからInstanceを作る方法は一種とする。

4. 記述例

記述例を下に一つ示す。

```

class name
  RectangularArea
  static
  super class
  local class name
  rectangle
  static
  super class
  Object
  instance variable
  xpos ypos width height
  instance method
  xpos:x ypos:y width:w height:h ||
  xpos:=x ypos:=y width:=w height:=h
  show || ^self
dynamic
super class
Object
event method
New || self xpos:0 ypos:0 width:0 height:0
instance message
xpos:x ypos:y width:w height:h ||
rectangle xpos:x ypos:y width:w height:h
clear ||
<primitive>
<ok?> ifTrue: [^self Paramerr || ^self killself

```

5. おわりに

Smalltalkは並列性を導出したDinn-Bellの概念について述べた。本言語は現在170以上の用途にある。次にこの言語を効率的に実行する計算機システムの構成を検討していく予定である参考文献

- [1] 本大会 2R-6
- [2] 本大会 4R-6
- [3] Goldberg AND Robson, Smalltalk - 80 Addison and Wesley (1983) (3冊)