

## 5D-7

オブジェクト指向型言語への事象駆動型  
プログラミング機能の導入の一方式について芳賀博英, 大藤淑子, 古賀明彦, 吉浦裕, 中所武司  
日立製作所システム開発研究所

## 1. 始めに

知識情報処理システムの記述を目的として、オブジェクト指向型プログラミングと論理型言語を融合した知識処理言語S-LONLIの研究、開発を行ってきた[1], [2]。現在までに、処理系のプロトタイプが完成し[3]、それを用いていくつかのプログラムを記述した。S-LONLIはオブジェクト指向型プログラミングをベースとしているため、個々のコンポーネントが外からの命令によって動くようなシミュレーションの問題は、モデリングもしやすく、プログラムも比較的簡単に作成することができた。しかし、(オブジェクト指向型言語一般の問題でもあるが)同じシミュレーション的な問題でも相手の状態を監視しながら自分が動くというような問題や、事象駆動(Event Driven)的なプログラム例えばプロダクション・システムは、必ずしもうまくプログラミングできない。本報告ではこれらの記述実験をふまえて、オブジェクト指向型言語に状態監視的あるいは事象駆動的なプログラミングを可能にする機能を導入する方式の一案について報告する。2ではS-LONLI言語の概要を簡単に述べ、3で記述実験の概要について述べる。4ではオブジェクト指向型プログラミングではうまくプログラムできない点を示し、それに関する一つの解決策を提案する。

## 2. S-LONLI言語の概要

S-LONLIは、オブジェクト指向プログラミングをベースにし、それに論理型プログラミングの機能を融合した言語である。S-LONLIの基本概念はオブジェクト指向プログラミングのそれとほぼ同じである。処理の対象となる実体(Entity)はオブジェクトと呼ばれる。オブジェクトには変数宣言とメソッド定義が可能である。メソッドは論理型言語で記述される。プログラムの実行はオブジェクトの間でのメッセージ交換によって行われる。図1にS-LONLIのソースプログラムの一例を示す。このプログラムは、内部にカウントを示す変数を持ち、アップ、ダウンなどのメッセージに従い内部の状態を変化させるカウンタを記述している。

## 3. 記述実験

現在までにS-LONLIで記述したシステムのうち、典型的な2つの例について簡単に説明する。

## 3.1 在庫管理システム(比較的うまくできる例)

このシステムは、情報処理学会誌[4]に掲載されていた問題である。S-LONLIでこの問題を記述した時には、倉庫係、受付係、コンテナ、在庫リスト、在庫不足リスト、その他、ようなオブジェクトを作成した。

```
define_frame counter ;
  attribute class ;
  supers root ;
  i_vars count(initial(0)) ;
  i_method up ;
    up :- New_count is count + 1,
          change(count, New_count), show ;
  end ;
  { method "down" is almost same as "up" }
  i_method set ;
    set(Value) :- change(count, Value), show ;
  end ;
  i_method show ;
    show :- write(' Value of count ==> '),
            write(count) ;
  end ;
end.
```

図1 S-LONLIのプログラム例

そして、受付係の仕事(メソッド)として入出庫依頼、在庫リストの表示などを定義し、倉庫係の仕事(メソッド)として、入出庫指示受付、その他のメソッドを定義した。システムは、端末から受付係に入庫、あるいは出庫の指示を送ったり、在庫の内容を問い合わせたりするメッセージを送ることによって動く。このシステムは実世界の状況に似ている動作をする。

## 3.2 ネコとネズミの追いかけっこ(あまりうまくできない例)

この問題は、次のようなものである。

ある大きさの部屋に一匹のネコと数匹のネズミがいる。部屋は密閉されていて外へは出られない。この状態でネコはネズミを追いかける。但しネコは近眼で自分を中心としたある領域しか見えない。この領域の中にネズミがいればそれを追いかける。いなければランダムに部屋を動く。ネズミはランダムに部屋を動く。この状況をシミュレートするプログラムを作成せよ。

この問題では、オブジェクトとしてネコ、ネズミ、そして部屋(もしくは空間)を表現するボードというオブジェクトを利用する。さらに全体を統括するシミュレータというオブジェクトを作り、制御をしている。

## 4. 記述実験の5の手裏

A method to Introduce Event-Driven Programming facility  
into Object-oriented Programming language

Hirohide Haga, Yoshiko Ohto, Akihiko Koga, Hiroshi Yoshiura, Takeshi Chusho  
Systems Development Laboratory, Hitachi, Ltd.

ここでは、3で述べたもの以外も含めた記述実験を通して得られた知見について述べる。

#### 4. 1 オブジェクト指向型言語の問題点

##### (1) 記述実験での問題点

在庫管理システムでは大きな問題点は無かったが、ネコとネズミの問題をプログラムしている時に、「見える」という状況の表現が問題となった。「見える」という行為は、命令されておこなうものではなく、常に行われているものである。このように、

オブジェクトがメッセージ送信以外の手段で情報を得ることが自然な状況

を表現することが、オブジェクト指向型言語では難しい。これは「自分の行動はすべてメッセージによって支配される」というオブジェクト指向型言語の本質的な問題であると考えられる。

##### (2) 知識処理システム記述における問題

S-LONLIの対象である知識処理の分野では、この問題は、現在の知識処理の主流になっているプロダクションシステム(P S)において典型的に表れる。P Sは次のようなサイクルを繰り返す。

①ワーキングメモリ(WM)と呼ばれる共通領域に書かれた情報に従って、実行可能なルールが発火して必要な処理をおこなう。

②実行の結果をWMに書き込む

③①に戻る。

WMに何かある事象が書かれた時に、発火可能なルールが自ら発火するという制御方式をとるのが事象駆動型のプログラミングの本質である。

現在のオブジェクト指向型言語によるプログラミングでは、全体を統括するオブジェクトを作成し、これがWMの内容を常にチェックして、ルールの発火の制御をするというプログラミングを利用者が直接記述せざるを得ない。これは、全体を知っている「神様」のようなオブジェクトを導入することになる。

##### (3) 記述実験での解決策

上記の問題の解決策として、記述実験では「見える」という状況を表現するのに「ボード」というオブジェクトを利用した。つまり、「ボード」というオブジェクトが神様オブジェクトの役割を果たした。ボードは部屋(あるいは光を伝える媒体としての空間)を表現するオブジェクトである。一定の周期ごとにネコはボードに対して自分のセンスできる範囲にネズミがいるかどうかを尋ねる。いればその位置を教えてもらい、いなければ次の周期までランダムに動く。別の例では、トロッコの動きを制御するのに「レール」が「神様オブジェクト」となった。しかし、できればこのような不自然なオブジェクトを利用者がプログラムしなければならないことは避けたい。

#### 4. 2 解決策の提案

4. 1の問題の解決策の一つとして、オブジェクト指向型言語にこれらの機能を導入する方法の一つを以下に提案する。

システム組込のオブジェクトとして、上で述べた「神様オブジェクト」を提供する。このオブジェクト中ではメッセージ送信なしに他

のオブジェクトの情報がさわれる。またこのオブジェクトの中は事象が漂っている。各オブジェクトは、自分に固有のアンテナを持っており、どのような事象に同調するかを規定したチューナ(同調器)を通して自分の中にながっている。各アンテナは常に事象を監視しており、自分に同調する事象が出現すればそれを検知して教える。プロダクション・システムのプログラミングをする利用者は、チューナの同調条件とその時の処理をあらかじめ記述しておき、実行時に必要なオブジェクトを「神様オブジェクト」の中に入れる。

システムが提供するこの「神様オブジェクト」を利用することによって、利用者は特別の「神様オブジェクト」を導入することなく、3. 2のような問題を記述することができる。たとえば、ネコとネズミの問題では、神様オブジェクトの中にネズミを放しておき、ネズミが移動するごとにその位置を情報として含む「事象」を放出する。ネコの持つチューナの同調条件として、自分を中心としたある大きさの領域にネズミが入ってきたら、それを知らせて捕獲行動に移るようにしておく。「神様オブジェクト」の中には「事象」が漂っており、アンテナは自分の同調条件に合う事象を見つけると、それを捕まえる。このようにして、事象駆動型プログラミングがおこなえる。(図2参照)

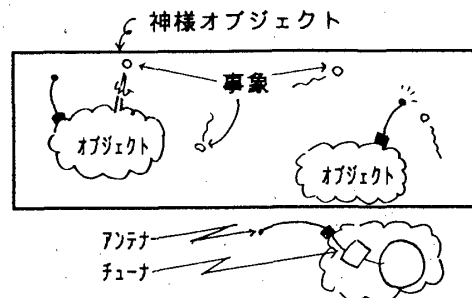


図2 神様オブジェクトの様子

#### 5. 終わりに

知識処理言語S-LONLIの記述実験をおこない、オブジェクト指向型プログラミングの問題点と、その解決策の一つを提案した。

##### [参考文献]

- [1] 芳賀他: "知識情報処理システム記述言語S-LONLIの提案", 日本ソフトウェア科学会第1回全国大会2D-2('84)
- [2] 中所他: "対象指向型言語と論理型言語の融合に関する考察", オブジェクト指向(鈴木則久編), 共立出版('85)
- [3] 芳賀他: "On the prototype processor of knowledge information processing system description language S-LONLI", 日本ソフトウェア科学会第2回全国大会4B-5('85)
- [4] 山崎: "共通問題によるプログラム設計技法解説", 情報処理vol. 25, No. 9('84)