離散事象シミュレーション言語 O²SLについて

4J-2

佐伯晋弥　新井浩志　深澤良彰　門倉敏夫
早稲田大学理工学部

1. はじめに

離散事象シミュレーションのモデル化の方法としては、以下に示すものが広く利用されている。

a）待ち行列理論に基づくモデル化手法
b）待ち合わせ理論に基づくモデル化手法
c）メッセージ・パッシング方式に基づくモデル化手法

c）の手法では、要素間のメッセージ・パッシングに従ってシミュレーションが進行する。従って、要素の振舞いに注目した離散事象シミュレーションの記述が自然に行える。この手法を採用したものとしては、従来の手続き型言語にメッセージ・パッシングとそのバッファリング機能を追加する方式[1]や、オブジェクト指向型言語を用いる方式が考えられる。

しかし、従来のメッセージ・パッシングを用いた言語では、各シミュレーション要素の動作記述と発生イベントの時間管理の記述が完全に分離していなかった。このため、変更の容易性、試験実行の容易性について難点があった。

我々は上記の問題点を克服するため、オブジェクト指向型言語をベースとして、新たな離散事象シミュレーション言語O²SL（Object-Oriented Simulation Language）を定義し、その処理系を構築することとした。

2. 特徴

本来、シミュレーションでは、少しずつシミュレーション条件を変えながら、繰り返し試験的に実行することが多い[2]。このため、シミュレーション言語には、一般のプログラミング言語以上に、記述の部分的変更の容易性が求められる。

メッセージ・パッシング方式を用いて、ある離散事象を表現しようとする場合、大別して以下の二つの記述が必要となる。

1）事象記述・・時間軸管理、要素であるオブジェクトの生成・消滅、出力情報に関する記述
2）オブジェクト記述・・オブジェクトの動作記述

実際には、1）の記述を少しずつ変化させながらシミュレーションを行うことが多いため、1）と2）の記述は完全に分離していた方が望ましい。

O²SLでは、オブジェクト記述と事象記述を完全に分離するため、オブジェクト指向型言語smalltalk-80に、以下の機能を追加している。

a）ブロードキャスティング構文の採用
b）割り込み型メッセージの設定

3. 概要

シミュレーションは、入力メッセージ・バッファを持つシミュレーション要素（オブジェクト）が、事象記述に従って、メッセージ・パッシングを行うことにより進行する。

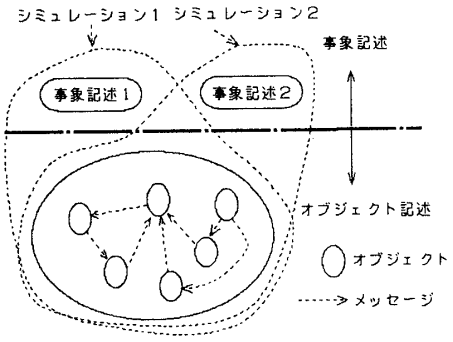

図1　O²SLの記述モデル

3．1 オブジェクト記述と事象記述の完全な分離

図1にO²SLの記述モデルを示す。事象記述を入れ替えることにより、同じオブジェクト群に対する様々な条件の下でのシミュレーションが可能になる。逆に同じ条件の下で、オブジェクトの処理内容を変えてシミュレーションを行うこともできる。O²SLで採用した事象記述、オブジェクト記述の内容を以下に示す。

＜事象記述＞
1）ヘッダ....シミュレーション名、最大時間、使用するオブジェクトのクラス、メッセージの優先度
2）出力情報....出力する変数、メッセージ
3）イベント....時間指定の発信メッセージ

＜オブジェクト記述＞
1）ヘッダ....オブジェクト名、使用する変数名
2）メッセージ....受信メッセージと動作の定義

図3に生産者－消費者モデルにおける生産者オブジェクトの記述の例を、図4にこのモデルの事象記述例を示す。このモデルには、複数の生産者と消費者、及び一つの市場が存在する。市場は、消費者からの要求に応じて在庫を管理し、生産者は、市場からの要求と他の生産者の生産状況に応じて生産管理を行う。

図4の（1）において、発生イベントを記述している。ここでは、各時刻で発生するイベントをオブジェクトに対するメッセージ・パッシング発信の形で記述する。図4の（1）における最後の文は、't=0' で生成された消費者クラスのオブジェクト c1 に対して、't=20' で 'consume' というメッセージを送っている。これによってオブジェクト c1 は、消費者クラスのオブジェクト記述で定義された 'consume' の動作を開始する。

A Discrete-Event Simulation Language O²SL
Shinya SAEKI, Hiroshi ARAI, Yoshiaki FUKAZAWA, Toshio KADOKURA
School of Science and Engineering, WASEDA University

3．2　ブロードキャスティング構文の採用

あるオブジェクトにメッセージを送る場合には、その対象となるオブジェクトを識別する必要がある。しかし、オブジェクトの動作記述と、発生イベントの記述を分離するためには、オブジェクトの生成・消滅状況に依存せずにメッセージを送る機能が必要である。このためO²SLでは、特定のオブジェクトを指定せずにメッセージを送るために、以下のブロードキャスティング構文を採用している。

```
BSO:'<CLASS>'  <MESSAGE>.
(Broadcast to Sameclass Objects)
```

・・同じクラスのオブジェクトに対してメッセージをブロードキャストする場合に使用する。

```
BAO  <MESSAGE>.
(Broadcast to All Objects)
```

・・現存するオブジェクト全てに対してブロードキャストする場合に使用する。

図3の例では（2）において他の全ての生産者オブジェクトの生産状況を調査するために、ブロードキャスティングが使用されている。これは、生産者クラスのオブジェクトに対して、'produce' 状態にあるならば、生産量を返すように要求するメッセージである。返値は Ppq* という配列に納められ、以後の処理に使用される。

3．3　割り込み型メッセージの設定

従来の手法では、他のオブジェクトの内部状態を参照するために、共有変数や、各オブジェクトの状態を管理する特定のオブジェクトを用いていた。これらは、オブジェクトの生成・消滅状況を把握している必要があり、参照するオブジェクトも、このような変数や特定のオブジェクトを認識していなければならなかった。これに対し、O²SLでは、オブジェクト記述間の独立性を保つため、従来のメッセージに優先度を設定する方式[1]に加えて、割り込み型のメッセージを設定している（図2）。

割り込み型メッセージは、入力バッファに入らない。そして、このメッセージを受けたオブジェクトは、現処理を中断して割り込みメッセージの処理を行う。図3の（2）と（3）にこの例を示す。文の最後の | INT で、割り込み型メッセージであることが示される。図3の（3）は、生産者に対し、arg の状態にあるならば、現在の生産量を返すように要求するメッセージの定義である。

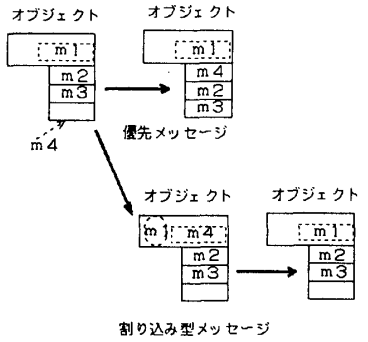

図2　割り込み型メッセージ

```
SimulationObject:Producer
superClass:SimulationObject
instanceVariables:'ability state produceTime
                   produceQuantity'
responses:
quantityAt:arg | INT                              --
     state = arg                                    (3)
      ifTrue:[ ↑produceQuantity ]                  --

investigate:arg
    | ms M Mcon Mzaiko Ppq* |
      :
      :
      Ppq* ← BSO:'Producer'                        --  (2)
               quantityAt:'produce'.               --
      :
      :
```

図3　生産者－消費者モデルにおける生産者のオブジェクト記述

```
SimulationName:Producer-ConsumerModel
MaxTime:t=200
SimulationObjectClass:#(Producer Consumer
                        Market)
MessageTypes:#('HP' '')
OutputMonitor:
traceMessages:#(Producer)
      :
      :
Events:                                           --
t=0 [p1 ← Producer createAbility:100              |
                  produceTime:8.                  |
     c1 ← Consumer create:5.                      |  (1)
     (M ← Market create) open | HP.]              |
t=20 [c1 consume | HP.]                           |
      :                                           |
      :                                           --
```

図4　生産者－消費者モデルにおける事象記述

4．　おわりに

O²SLを用いて幾つかの問題を記述してみた結果、事象記述を書き換えても、オブジェクト記述に影響しない事を確認した。従って、従来のメッセージ・パッシング方式のシミュレーション言語より、変更の容易性及び試験実行容易性が高まっていることが確かめられた。今後は、O²SLをさらに多くの問題に適用し、動作確認を行う予定である。

参考文献

[1] R L.Bagrodia, K M.Chandy, J Misra: "A Message-Based Approach to Discrete-Event Simulation", IEEE Trans.on Software Engineering, vol. SE-13, no.6, june 1987

[2] J A.Spriet, G C.Vansteenkiste: "Computer-aided Modeling and Simulation", Academic Press, New York and London, 1982