

## 7C-1

## 分散型問題解決シミュレータDPSSの概要

北村泰彦 奥本隆昭  
大阪市立大学 工学部

## 1 はじめに

疎結合、複数のエージェントによる協調的な問題解決を扱う分散型問題解決は並列性、拡張性、信頼性、分散アプリケーションへの対応といった利点のゆえに注目されており、すでに多くの報告がなされている[1]。本稿では分散型問題解決システムの実現とその評価のためのシミュレータであるDPSS (Distributed Problem Solving Simulator) について述べている。

## 2 DPSSの特徴

分散型問題解決シミュレータDPSSには以下のような特徴がある。

- 並列オブジェクト指向  
分散型問題解決におけるエージェント間の相互作用はオブジェクトとオブジェクト間のメッセージ伝達として表現され、オブジェクトは疑似並列的に動作する。このために、エージェントの追加や削除が容易であるとともに、現実の分散処理システムへのインプリメントも容易になる。
- 時間概念の導入  
各オブジェクトの内部処理やメッセージの伝達に要する時間を表現することが可能である。時間概念の導入により、通信速度や推論速度を変化させた分散型問題解決システムの評価が可能になる。
- 論理型推論  
DPSSはPrologで記述されており、その推論機能をそのままオブジェクト内で用いることができる。従ってオブジェクト内においても高度な問題解決機能を実現することが可能になる。

## 3 DPSS

DPSSにおけるオブジェクト(Object)は図1に示すように、メッセージキュー(Message Queue)とメッセージ解釈部(Message Interpreter)から成り立っている。メッセージキューには他のオブジェクトから送られてくるメッセージがタイムスタンプ(Time Stamp:TS)順に並べられている。また、メッセージ解釈部は各オブジェクト毎のクロック(Clock)に従ってメッセージを処理し、クロックを更新してゆく。

DPSSにおいてオブジェクトは以下のように記述される。

```
object(object_name,action_list).
```

object\_nameはオブジェクトの識別子、action\_listはアクション(action)と呼ばれるオブジェクトの動作が記述される。アクションには起動方法の違いにより、以下の2種類がある。

## 1. a\_action(frequency,goal\_list)

オブジェクトが自ら起動するアクションでgoal\_listには実行するゴール列、frequencyにはゴール列が実行される頻度を記述する。(DPSS内部では自分宛のメッセージが送られるように実現されている。)

## 2. p\_action(message,goal\_list)

オブジェクトが他のオブジェクトからのメッセージにより起動されるアクションで、goal\_listには実行するゴール列、messageには照合するメッセージを記述する。

ゴール列には一般的にPrologの述語を記述するが、DPSS特有の述語として以下のものが定義されている。

- send\_message(message,object,time)  
他のオブジェクトにメッセージを送る述語である。アクションが起動されたtime時間後にメッセージmessageがオブジェクトobjectに到着する。
- terminate(time)  
アクションの実行終了を宣言する述語である。オブジェクトのクロックはtime時間だけ更新される。
- assert\_data(data), retract\_data(data)  
dataで記述されるデータをオブジェクト内で追加や削除をする述語で、そのデータは他のオブジェクトからはアクセスできない。

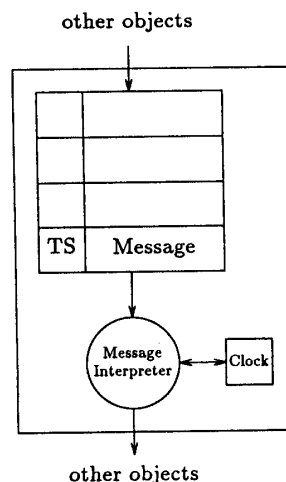


図1 オブジェクトの構造

## 4 オブジェクトの制御

DPSS では図 2 に示すように制御部 (Controller) が複数のオブジェクトを管理し、その実行を以下のアルゴリズムにしたがって制御している。ここではメッセージキューが空でないオブジェクトのことをアクティブオブジェクトと呼ぶことにする。

**Step.0 [初期化]** 全オブジェクトのクロックを 0 に合わせる。

**Step.1 [オブジェクトの選択]** アクティブオブジェクトのうち、クロック最小のものを選ぶ。

**Step.2 [メッセージの選択, 処理]** キューの先頭メッセージのタイムスタンプとクロックを比較し、タイムスタンプの方が大きければ Step.3 へ。小さければ、メッセージを削除し、そのアクションを実行する。Step.1 へ。

**Step.3 [クロックの更新]** キューの先頭メッセージのタイムスタンプと、自分を除いたアクティブオブジェクトのクロックの中で最小のものに自分のクロックを更新する。Step.1 へ。

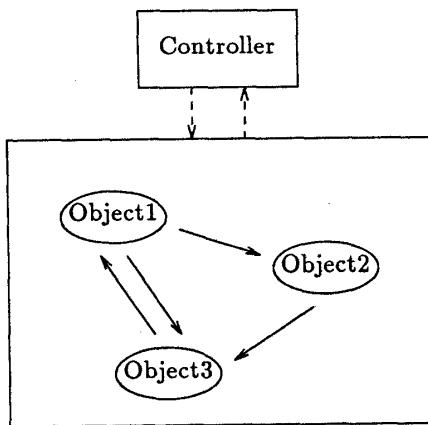


図 2 オブジェクトの制御

## 5 記述例

本節では DPSS による契約ネットプロトコルの記述例を示す。契約ネットプロトコル [2] は R.G.Smith により提案された分散型問題解決のための問題割り当て方式で、以下のように記述される。

```

object(manager, [
  a_action(constant(5, [0,100]), [
    create_task(TASK)
    send_message(task_announcement(manager, TASK),
      all, 10),
    send_message(expiration_time(TASK), manager, 30),
    terminate(1)]),
  p_action(bid(P_CONTRACTOR, TASK, ESTIMATION), [
    record_bid(TASK, P_CONTRACTOR, ESTIMATION),
    terminate(1)]),
  p_action(expiration_time(TASK), [
    choose_contractor(TASK, CONTRACTOR),

```

```

    send_message(award(manager, TASK), CONTRACTOR, 10),
    terminate(1)]])
  ]).

```

```

object(agent(X), [
  p_action(task_announcement(MANAGER, TASK), [
    calculate_estimation(TASK, ESTIMATION),
    send_message(bid(agent(X), TASK, ESTIMATION),
      MANAGER, 10),
    terminate(1)]),
  p_action(award(MANAGER, TASK), [
    execute_task(TASK, TIME),
    terminate(TIME)]),
  ]).

```

マネージャ (manager) ではタスク (TASK) が 0 から 100 単位時間の間、5 単位時間毎 (constant(5, [0,100])) に生成され、タスクの通知 (task\_announcement) メッセージが全てのエージェント (all) に向けて送られる。また、自分宛に入札締切 (expiration\_time) メッセージを 30 単位時間後に送ることにより入札締切時間が設定される。通知メッセージを受け取ったエージェントは見積りを算定し (calculate\_estimation)、マネージャに入札 (bid) メッセージを送り返す。入札メッセージを受け取ったマネージャはタスク毎に記録しておき (record\_bid)、入札締切時間が来るとそれまでに送られてきた入札をもとに契約エージェントを決定し (choose\_contractor)、判定 (award) メッセージを送る。判定メッセージを受け取ったエージェントはタスクを実行する (execute\_task)。なお、本記述例ではメッセージの処理に 1 単位時間、メッセージの送信に 10 単位時間かかるとしている。また、タスクの実行時間はタスクに応じて異なるとしている。(execute\_task で算出される。)

## 6 おわりに

本論文では分散型問題解決システム実現と評価のためのシミュレータ DPSS の概要と契約ネットプロトコルの記述例について述べた。DPSS は現在、UNIX ワークステーション上に暫定版が完成しつつあり、波及型推論方式 [3] や回覧板プロトコル [4] の評価のために用いている。さらに必要な機能として、デバッグ機能やウィンドウ表示機能などがあげられるが、それらは今後の課題としたい。

## 参考文献

- [1] Alan H. Bond and Les Gasser, editors, *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, California, 1988.
- [2] Reid G. Smith, The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Trans. on Comput.*, C-29(12), 1980.
- [3] 北村泰彦他, 分散型問題解決の定式化と一推論手法, 人工知能学会研資, SIG-FAI-8802-4, 1988.
- [4] 北村泰彦他, 分散型問題解決における問題割り当て, 情処学マルチメディアと分散処理研資, 42-1, 1989.