マルチエージェント指向プログラミング言語April*

4L－4

高田裕志　Francis G. McCabe　和田裕二
（株）富士通研究所 情報社会科学研究所
{yuji,fgm,wada}@iias.flab.fujitsu.co.jp

1　はじめに

近年ネットワークの発達にともない、マルチエージェント技術が注目を浴びるようになってきた。マルチエージェント指向のプログラミング言語も、Telescript, Joule, ACL, HotJavaなど、実用を目指したさまざまな言語が複数のベンダーによって開発・提供されている。

April [1] はネットワーク上でのメッセージ交換やマルチプロセスなど、分散システムの開発に必須な機能を備えた記号処理プログラミング言語であり、「分散人工知能」や「分散データベース」などにおける分散協調指向のシステム、特に、インターネットなどの広域ネットワーク上での分散協調システムの開発に適している。本稿では、独自の名前づけによるグローバルなプロセス管理、パターンに基づくメッセージ交換、強力なマクロ機能といったAprilの主な機能と特徴について述べる。

2　Aprilのプログラム例

図1はプログラミング言語の紹介によく用いられる"*hello world*"プログラムをAprilで記述したものである。これを実行するには、まずAprilコンパイラによりマシン独立な中間コードを生成する。生成された中間コードはAprilインタプリタによって実行される。

```
program {
  hello() {
    string?user := inline(stdin);
    writef(stdout,"hello",□);
    [done,user] >> creator();
  };
  main() {
    fork hello;                  /* spawn sub-process */
    { [done,string?user] ->      /* wait for child */
        writef(stdout," world, %S\n",[user])
    }
  }
} execute main
```

図1: "*hello world*" プログラム

このプログラムを実行すると、通例のものとは異なり、まずキーボードからの文字列の入力を要求（inline(stdin) の部分）する。入力された文字列は"*hello world,* "の文字列に続いて出力される。たとえば、入力"*Bob*"に対して、"*hello world, Bob*"と出力される。

3　プロセス管理

"*hello world*"プログラムはhelloとmainのふたつの手続きからなり、実行されると、まずexecuteで指定された手続きmainが起動される。次に、mainは手続きhelloをAprilのプロセスとしてforkする（fork helloの部分）。このAprilプログラムによってforkされたプロセスはApril独自のものであり、Unixなどの OS で管理されるプロセスとは異なるものである。実は最初のmain自身もAprilのプロセスであり、"*hello world*"プログラムに対して、ふたつのAprilプロセスが生成されることになる。

Aprilでforkされたプロセスは Aprilインタプリタと April独自の"nameserver"によってネットワーク上でグローバルに名前管理が行なわれる。通常はインタプリタによって適当な名前が各プロセスに割り当てられるが、明示的に名前を与えることも可能である。メッセージ交換はプロセスの名前を用いて行なわれる。他のプロセスにメッセージを送るには、そのプロセスの名前を送り先に指定する。

Aprilのプロセス管理のひとつの特徴は、コミュニケーションの信頼性である。つまり、通信中にあるプロセスがダウンしてしまっても、また同じ名前で別のプロセスを立ち上げれば、コミュニケーションを回復することができる。

4　メッセージ交換

メッセージの送信は、文

pattern >> *receiver*

で行なわれる。図1の例では、記号doneと変数userの値のペアを関数creator()で指定される受信先に送信している。ここで、関数creator()はそのプロセス

---

* A multi-agent oriented programming language April
Yuji Takada, Francis G. McCabe, and Yuji Wada
Institute for Social Information Science,
FUJITSU LABORATORIES LTD.
1-9-3 Nakase, Mihama-ku, Chiba-shi, Chiba 261, Japan

をforkしたプロセス（この場合main）の名前を値として返す。

メッセージの受信は、文

pattern -> *statements*

で行なわれる。*pattern*に“パターン・マッチ”するメッセージを受信したとき、*statements*が実行される。パターン・マッチによって、*pattern*に含まれる変数がマッチしたデータに束縛される。図1の例では、変数userがマッチしたデータに束縛される。このようにメッセージはパターン・マッチによって処理されるので、手続き的に解析することなしに複数のメッセージを統一的に処理することが可能である。この例の場合、記号doneと任意の文字列のペアであるメッセージはすべてひとつの受信文で処理される。

Aprilでは、記号、文字列、リストなどのさまざまなデータ構造が利用可能であるが、メッセージにも任意のデータ構造を用いることができる。プログラムやパターン自身をも抽象化によってデータとし、メッセージに含めて送受信することが可能である。この機能を用いて、遠隔でプロセスをforkすることができ、マルチエージェント技術のひとつの特徴である“移動エージェント”を実現することができる。

5 マクロ機能

Aprilには表現力の高いマクロ機能を用意しており、文脈依存性も表現することが可能である。このマクロ機能を用いて、April上で「知識表現言語」などの高級／抽象言語を構築することも可能である。

たとえば、マクロ

```
#macro ?T where (?P in ?S) ->
        setof {for P in S do elemis T};
```

は、集合Sの中からパターンPにマッチする要素をすべて求めるAprilのプログラム

```
setof {for P in S do elemis X}
```

を、

```
X where P in S
```

のように、知識表現言語風に記述することを可能にする。マクロを組み合わせることで、

```
X where X in male_set and
        [X,symbol?Y] in partners and
        not Y in male_set
```

のような、より自然な表現も記述することができる。また、ACLのような抽象度の高い言語を、マクロを用いてAprilで実装することも可能である。

6 エージェント指向技術

現在、Aprilをマルチエージェント技術を生かした総合的なシステムとするべく、さまざまな関連ツールを開発中である。

AdBはエージェント指向のデータベース・サーバで、各Aprilプロセスからはメッセージ交換によってデータベースをアクセスすることができる。AdBのひとつの特徴は、データベースの変更を“監視”する機能である。この機能により、データベースに変更が行なわれた場合、AdBは他のプロセスにメッセージを送ることで変更を通知することができる。

DialoXはエージェント指向のグラフィカルユーザインタフェース・サーバである。各Aprilプロセスはメッセージ交換によって、さまざまなウィンドウをDialoXに表示させることができる。したがって、Dialoxを用いれば、クライアントがインタフェースを事前に知らなくても、ウィンドウの仕様を直接サーバが送ることでインタフェースを指定することが可能になる。

7 おわりに

本稿ではAprilの主な機能と特徴について述べた。Aprilのプロセス管理はコミュニケーションの信頼性を保証する。例えば、Adbによるデータベース・サーバがダウンしても、クライアントを再起動することなく、サーバを再起動するだけでサービスを回復させることができる。また、April独自の名前管理によって名前指定によるメッセージ交換が可能であり、プロセス間のコミュニケーションが容易である。メッセージ交換はパターンによって行なわれるので、手続き的解析なしに複数のメッセージを統一的に処理することが可能である。マクロ機能を用いてApril上で高級／抽象言語を構築することも可能である。

クライアント-サーバ・システムや分散人工知能で用いられる「黒板システム」は、Aprilによって比較的容易に構築可能である[1]。今後、Aprilシステムによって、次世代グループウェアなどの広域ネットワーク上での応用システムを開発していく予定である。

参考文献

[1] Frank G. McCabe and Keith L. Clark. April - Agent PRocess Interaction Language. In M. J. Wooldridge and N. R. Jennings, editors, *Intelligent Agents, Lecture Notes in Artificial Intelligence, 890*, pages 324–340. Springer-Verlag, 1995.