3R-7 並列オブジェクト指向言語DinnerBellの概要ーその動作

金子誠司 河野真治 神田陽治
田中英彦、元岡達 (東京大学工学部)

1. はじめに

並列オブジェクト指向言語DinnerBell[1]のマルチ・プロセッサ上への実装の概要について述べる。本言語はActor model[2]に基ずく並列性を、よく記述できることを目的とした言語であり、ほぼActor modelに近いモデルをもっている。

2. DinnerBellのオブジェクトのモデル

DinnerBellにおけるオブジェクトのモデルを図1に示す。DinnerBellのオブジェクトは、通信系、メッセージ・キュー、そしてExecution, Event 2種のプロセス群より成っている。このオブジェクトは2種のメッセージ、動メッセージと静メッセージの2種を受けつける。静メッセージは、Execution側のプロセス群により処理されるが、このときExecution側で2つ以上のメッセージを同時に処理できるオブジェクトと、そうでないものの2種がある。普通は前者で、後者は排他記述[1]により生ずる。普通のメッセージは静メッセージで、動メッセージはオブジェクトと実時間とのインターフェイスを与えるためのもので、到着後、すぐに実行される。また、動メッセージを介してシステムを実現する下位のレベルを操作する手法が与えられている。動メッセージは、デバッガや、Secretary[1]の実現等に用いる。

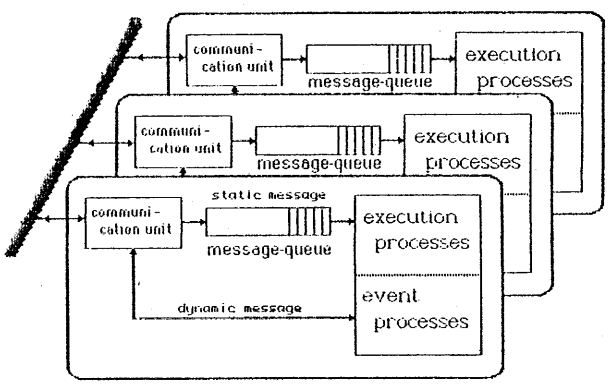

図1. オブジェクトのモデル

3. DinnerBellのオブジェクトの実行環境

DinnerBellでは、Smalltalk-80で言う、MetaClassはもっていない。DinnerBellでは各オブジェクトのクラスに対し、1つインスタンスを用意し、そこへ'new'というメッセージを送って新しいインスタンスを作る。オブジェクトの初期化のための手順は、動メッセージのメソッドとして書けばよい。DinnerBellでは、メソッドは、Super Classのものも含めて、セレクタにあてはまるすべてのものが実行されるのが原則であり、それに暗黙のOverrideの宣言が行なわれているという形となっているためである。次にDinnerBellでは、或るオブジェクトをNewによって作ったオブジェクトが実行を終了した場合は、そのオブジェクトの実行も終了し、領域の割りつけを解く。また、Compilerが充分にCheckして、Garbage Collectionを最小に押えるようになっている。(これは、クラス・オブジェクトの動メソッドとして実装されている。)

4. DinnerBellの実装

DinnerBellでは、Smalltalk-80よりもメソッドサーチの負荷が重い。また、オブジェクト=プロセサの対応では、その割り付けのアルゴリズムが明解なものとならない。そこで、DinnerBellでは、NameMaster[3]によるUnique Idを利用して、オブジェクトの発火チェックをオブジェクト自体より分離し、オブジェクトではなく、メッセージ単位=プロセサの対応で割り付けを行なう。

4a). テンプレート

セレクタに対応するものとしてUniqueなIdをもったテンプレートを考える。各オブジェクト・インスタンスは、発火可能なテンプレートの組み合わせの表(テンプレート・テーブル)をもっている。テンプレートはCompile時に生成されるものと、動的に作られるものとの2種あり それを区別するための1bitをもっている。

4b) メッセージの受信

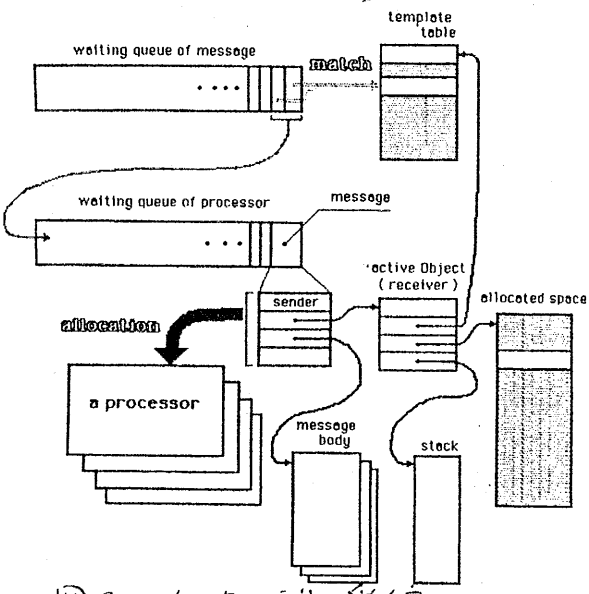

図2 メッセージの受信

手順を左図2に示す。システムに1つメッセージのキューがあり、メッセージはまずそこに入れられる。この内容は、その行き先のオブジェクトのテンプレートとつきあわされ、オブジェクトが発火可能とわかったときは、メッセージをプロセサ待ち行列(システムに1つ)に入れる。そして、そこから出したメッセージをプロセサに割りつける。プロセサはメッセージの行き先のポインタから、メモリに割りつけてあるオブジェクトを用いてメッセージを評価するという形をとる。前述の排他オブジェクトは、変数領域をロックすることにより実現している。

4c) メッセージの送信

変数領域のなかから必要な領域をcopyして、動的なテンプレートを加えたヘッダを付けて、メッセージとしてキューに入れるという手順となる。ヘッダはsenderの名もつけてあり、必要に応じて参照したり、返事のあて先にすることもできる。

4d) 変数の代入待ち (右図3)

DinnerBellでは、単一代入による同期が用いられている。代入はメッセージによって行なわれるがこのようなメッセージを待つために、送信のときのテンプレートをもったテンプレート・テーブルを作り値が代入されたら発火する手順を実現する。

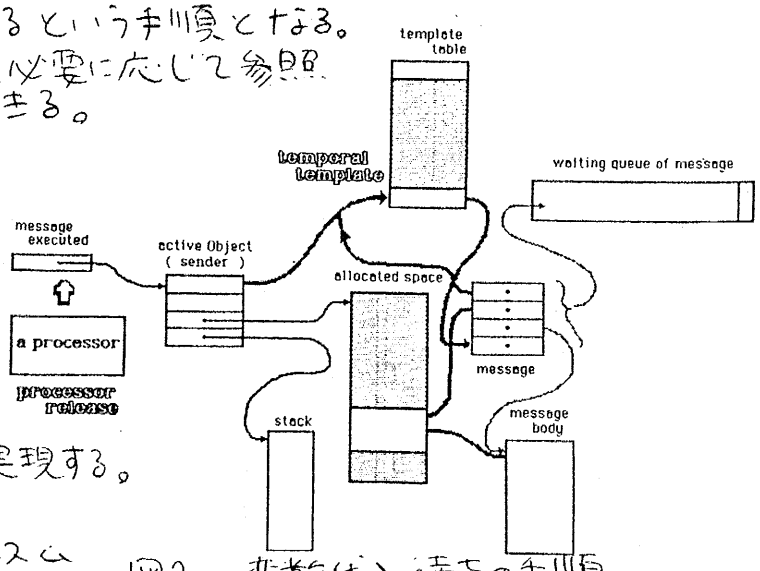

図3. 変数代入待ちの手順

5. おわりに

DinnerBellの下位の実行メカニズムについて述べた。これからシミュレータにより、量的な挙動について調べる予定である。

References

1) 神田.他,'並列オブジェクト指向言語DinnerBellの概要',ソフトウェア基礎論研究会. 11-3(1984)
2) A. Yonezawa, Specifying Software Systems with High Internal Concurrency Based on Actor Formalism', J. Inf. Proc, vol.2 Nr.4(1980)
3) 神田.他,'名前付けの統括管理システムNameMasterの構成',ソフトウェア科学会第1回大会,1C-3