1E-9

統合プログラミング環境 ORAGA (IV)
並列オブジェクト指向アーキテクチャ OragaItSelf

河野真治 神田陽治 田中英彦 元岡達
東大 工学部

OragaItSelf は，Oraga-project の中核言語である DinnerBellを直接に実行するアーキテクチャである。オブジェクト指向言語としてはSmalltalk- 80 があるが，並列処理向きとしてはActor 理論〔1〕が知られている。OragaItSelf は，Actor 理論に基づきオブジェクト指向言語を，メッセージの受信を表すイベント単位に並列に実行する。そのアーキテクチャは，オブジェクトの持つ多様なデータ構造に対応するように考えられている。

1) 実行モデル

オブジェクト指向言語には次の様な特徴がある。

・データと手続きを一体化した統一性
・メッセージセンドによる仕様と実装の分離
・メッセージセンドの独立性からくる並列性
・インヘリタンスによる拡張性

オブジェクト指向アーキテクチャは，これらの特徴を活かすものでなくてはならない。OragaItSelf は，これらをイベントに対応させた三組コンテクストの縮約という単純な実行モデルにより統一的に実現する。コンテクストは，プロセステーブルの一般化である。三組コンテクストは，以下の構成である。

M メッセージ
D 送り先オブジェクト
R 答を入れる変数

この三つは，全てActor である。コンテクスト自身は，Actor であっても良いしなくてもよい。Actor の持つレパートリー，またはメソッドは，コンテクストからコンテクストを導出するルールとみなすことができる。

OragaItSelf では，Actor の実行をコンテクストの縮約とコンテクストスケジューリングにより行う。縮約機はコンテクストの導出ルールを全て持っている。オブジェクトはスケジューラの中でコンテクストの指すデータ構造として表現される。頻繁に生成消滅するActor の性質がスケジューラの性質と対応している。

2) アーキテクチャの構成

オブジェクトはあるメッセージへの応答のみが規定されていればよく，実装の方法が外から見える必要はない。最も高速な実装はハードウェアにより直接に実装する事である。OragaItSelf は，オブジェクトを専用の縮約機を設けることにより高速に実行する。つまり縮約機は複数台存在し，並列に動作する。

Actor には，副作用を持たないものと，副作用を持つ

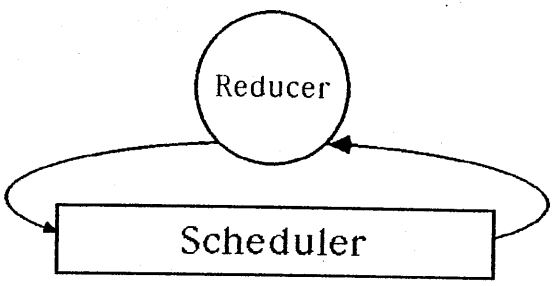

Fig.1 OragaItSelf の基本構成

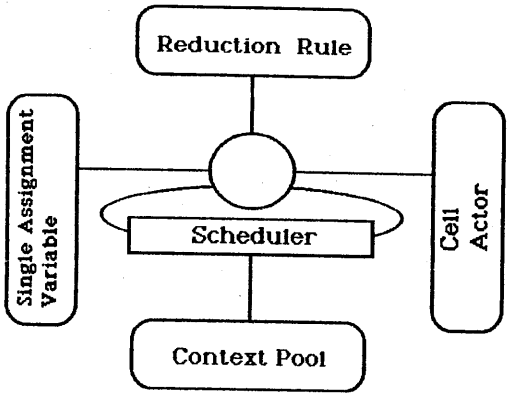

Fig.2 4つのメモリからなるノード

ものの二つがある。後者は，通常のメモリに相当する，cell Actor がその代表である。DinnerBellでは，前者は単一代入則により実行される。これはメッセージ送信を全て未来型〔1〕に限ることに相当する。つまり，その送信の答が必要となるまで，他の計算は並行して実行される。答が必要となると，その答が返るまで，その計算は，中断される。OragaItSelf ではこれらは二つの別なメモリとして実装される。

DinnerBellでは同期の問題は，単一代入則と排他制御により解決される。従って単一代入変数に対する代入が，それに引続く幾つかのコンテクストを発火させる条件となる。これは単一代入変数用のメモリがその変数に対する待ち行列を持ち，そのキューによりスケジューラに対して信号を送ることにより実現される。

排他制御はセマフォにより実現するが，これは副作用を持つ変数に対する待ち行列として実装される。単一代入変数用のメモリと cell Actor 用のメモリは，ともに待ち行列を持つ。単一代入変数用のメモリは，生成時に未定義を意味する値に初期化される。未定義な値に対す

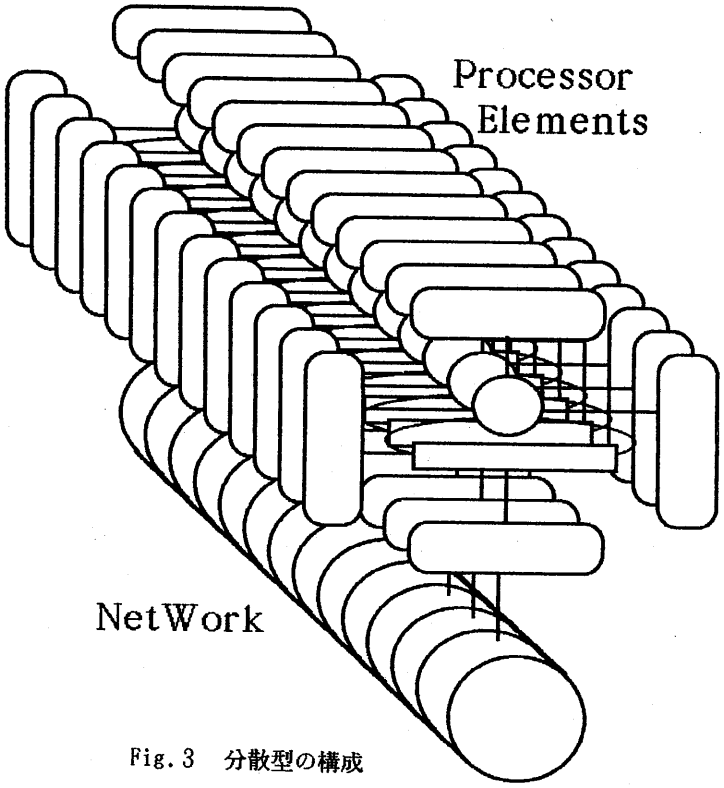

Fig.3　分散型の構成

る参照は全て待ち行列に挿入される。一旦，値が確定するとその値を同時に参照してもかまわない。単一代入メモリ上の確定した値は，自由にコピーしてよくコピーすることによりアクセスの競合が避けられる。一方 cell Actor では，全ての cell Actor に対する参照と代入は，データベースとして無矛盾である必要がある。これは通常，単一の cell Actor に対して，順序化することにより実現される。これは単一代入変数を用いて実現することもできるが， cell Actor 用のメモリそのものの機能とすべきである。 cell Actor 自身は，名前によりアクセスされるので，普通コピーは許されない。しかし，その名前はコピー可能である。

最終的に，OragaItSelf では，４種のメモリが存在する。それぞれ，縮約の為のルール，コンテクストのスケジューラ，単一代入変数， cell Actor を格納する。

3）階層的な構成

OragaItSelf は，この４種のメモリと，幾つかの種類の縮約機により，一つのノードを形成する。これらのノードは，さらにネットワークにより接続されて分散環境をつくる。ルールは全てのノードがコピーして持っている。 cell Actor は，各ノードに固定されている。その名前は，必要なところにのみ分配される。ガーベジコレクションなどで，cell Actorが移動する場合は，その間の無矛盾性を維持する必要がある。値の確定していない単一代入変数が移動することはないが，値の確定したものは，ノード間でコピーしつつ移動する。ネットワーク上を移動するのは，コンテクストであり，値のコピーはそれに伴って起きる。

独立性の良いコンテクストのみを他ノードに出すことにより，高いレベルでの並列性を維持することができる。答を返す必要のないコンテクストは，最も独立性が高い。あるノードの cell Actor の名前を多く含むものは，独立性が低い。このようにコンテクストの内容により，ある程度独立性を評価することができる。しかし，この点は，シミュレーションなどにより，実際の環境において，調整する必要がある。

一方，縮約機を多重化することにより，ベクトル化や，パイプライン処理等により同一のノード内で，より低位の並列実行を行うことができよう。この時，単一代入則が最も有効に働く。

このように，OragaItSelf には，縮約機の部分による下位の並列性と，各ノード間に生じる上位の並列性の二種類の並列性がある。

4）比較

慶大でのZoom〔3〕とConcurrent Smalltalk〔2〕は，逐次型のオブジェクト指向言語マシンの複数動作である。それに対しOragaItSelf は，高並列を狙う単調で階層的なアーキテクチャを持っている。またコンテクストの縮約は，従来の命令実行とプロセススィッチの中間の概念である。より高速なプロセススイッチと，より高機能な命令の両方を実現する。この構成は，リダクションマシン＋リアルタイムモニタのようにみることができる。リダクションマシンよりも豊富なデータ構造を取り扱うことにより，より容易に下位の並列性の実現することができると考えられる。

5）今後

完全結合型のマルチマイクロプロセッサによる，OragaItSelf の二つの並列性の内，上位の並列性のシミュレーションを計画中である。

参考文献

〔1〕Yonezawa,Y.,"An object Oriented Approach for Concurrent Programming",T.I.T., Research Report C-63,Nov.,1984

〔2〕横手靖彦，"Concurrent Smalltalk"，日本ソフトウェア学会第一回大会 2E -2

〔3〕Ishikawa, Y., "THE DESIGN OF AN OBJECT ORIENTED ARCHITECTURE ",Proc. of the 11th Int'l Symp. on Computer Architecture, Jun,1984

〔4〕Suzuki, N., "Sword 32：A Bytecode Emulating Microprocessor for Object-Oriented Languages ",Proc. of the Int. Conf. on fifth generation computer systems,Nov,1984