6F－6

自律協調分散システムNoahにおけるエージェントの制御機構

小塚 宏，佐藤 文明，宮崎 一哉，福岡 久雄
三菱電機(株) 情報システム研究所

1 はじめに

ダウンサイジングの潮流の下，多数の計算機を有機的に結合した分散システム環境が注目されている．分散システムでは，高いコストパフォーマンスとシステム構築の柔軟性が期待されるが，システムが複雑になるにしたがって，そのシステム全体の運用・管理が困難になり，またエンドユーザにシングルシステムイメージを与えるためにさまざまな工夫が必要となる．

Noah (Network oriented applications harmony)[1]は，このように複雑になる一方の分散システムの利用 / 運用 / 管理を容易かつ柔軟に行なうために，複数の自律的な計算主体(エージェント)が，ある系の中で協調しながら処理を進める，分散システムのモデルを提供し，大きくタプルスペース型の通信部，協調処理プロトコル部，エージェントの監視 / 制御部の3要素から構成される，分散システムの構築基盤である．本稿では，Noahにおけるエージェントの制御機構の概要を述べる．

2 Noahにおけるエージェント

Noahにおけるエージェントは，自律協調動作をするために用意された系(フィールド)の中で，Linda[2]タプルスペース(TS)型のメッセージ・プールに能動的にアクセスし複数のエージェント間でメッセージ交換を行ない，そのフィールドに設定したポリシーにしたがって協調動作を行なう．

ポリシーの定義や管理は，基本的にフィールド自体の計算主体であるフィールド・マネージャ(フィールド・マネージャ自身もエージェントとして実装される)が，そのフィールド内の情報を操作することによって行なわれる．エージェントは，この操作された情報を非同期に受信し解釈することによって，フィールド内部で一貫性のある振舞いをする．通常，フィールドあるいはエージェント自身に対して強制的な制御が行なわれることはない．

しかしながら，直接フィールドあるいはエージェントの起動，停止，再起動，内部情報の取得 / 変更等の強制的な監視 / 制御をエージェントの自律性を一部阻害して行なう場合がある．また，Noah の環境で動作するように作られていない既存のアプリケーションに対しても，Noah 環境での協調動作への参加を許し再利用するためには，これらのアプリケーションに対して疑似的にNoah のエージェントと同等に振舞わせるための機構を提供する必要がある．

3 エージェントの制御機構

Noah では，エージェントへの監視 / 制御あるいは，既存のアプリケーションの Noah 環境への取り込みを実現するために，協調処理機構の他にエージェントの監視 / 制御を行なう機構 NAM (Noah Agent Manager) を提供する．この監視・制御を可能とするために，エージェントは図1に示す形で利用される．

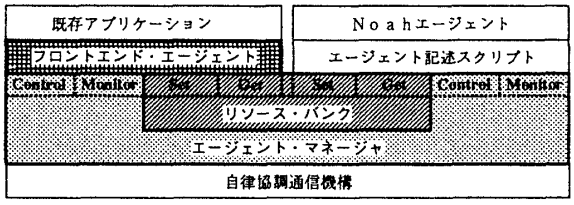

図 1: エージェントの監視・制御機構の構成

主な構成要素は，以下の3つの部分である．

1. リソース・バンク (RB)：

   エージェントの監視・制御を行なうために必要となるデータ / メソッドを管理する部分．

2. エージェント・マネージャ (AM)：

   Noah の非同期通信機構及び，監視 / 制御用に用意される同期型の通信機構と通信を行ない，リソース・バンクの情報を介してエージェントの監視・制御を行なうインタプリタ・プロセス．

Agent Control Mechanism for Autonomous Cooperative Distributed System: Noah

Hiroshi KOZUKA, Fumiaki SATO, Kazuya MIYAZAKI, and Hisao FUKUOKA
Computer & Information Systems Laboratory, Mitsubishi Electric Corporation
5-1-1, Ofuna, Kamakura, Kanagawa 247, Japan

3. フロントエンド・エージェント (FEA) :

   Noah 環境用に作成されていない既存のアプリケーションを疑似的に Noah 環境のエージェントと同等な振舞いを行なわせるために, 既存のアプリケーションと Noah 環境を媒介する機構.

これらの要素は, 個々のエージェントあるいはアプリケーションに対して外付けされる.

3.1 リソース・バンク (RB)

RB には, 各種のエージェントに対して共通な制御情報・操作をもつパブリックな部分と個々のエージェントに固有なプライベート部分とが存在し, パブリック部分は複数のエージェントで共用される. 個々のエージェントが特に監視・制御に関するデータ・メソッドを持っていなくても, RB を付加することでエージェントのプロセスの単位での起動 / 停止 / 再起動 / 優先順位操作 / 情報取得等の監視 / 制御が可能となる.

3.2 エージェント・マネージャ (AM)

AM 部分は, RB の内容が動的に変化しても, エージェント自身を停止することなく, 監視・制御を継続するためにエージェント・プロセスと Noah 通信機構との仲介を行なう. また, フィールドに対して動作する際には, 異なるフィールドのポリシーを侵すことなく他のフィールドとの情報交換を行なうための制約を行なうことになる.

3.3 フロントエンド・エージェント (FEA)

既存のアプリケーションの全てを Noah 環境に適合させることはできない. しかしながら, UNIX[1] 上の多くのフィルタ系のコマンド群や X ウィンドウ・システム上のグラフィカル・ユーザインタフェース (GUI) をもつアプリケーションの多くは, FEA を介在させることにより Noah 環境に取り込み, 疑似的にエージェントとして取り扱うことが可能となる. FEA は, 次の 2 種類に大きく分類される.

1. ファイル I / O 型 FEA:

   標準入出力, ファイル入出力, ソケット入出力, デバイス I/O 等に適用する. 図 2に, 標準入出力に適用した例を示す.

2. X イベント型 FEA:

   X ウィンドウ・システム上の GUI ウィンドウ間で送受されるイベントの通信, X サーバに格納される情報の利用に適用する (図 3).

[1]UNIX オペレーティング・システムは UNIX System Laboratories,Inc. が開発し, ライセンスしています.

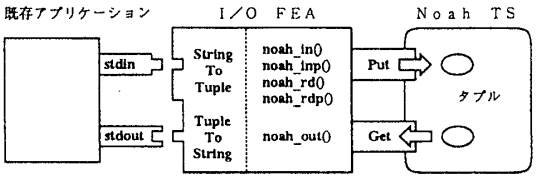

図 2: ファイル I/O 型 FEA の例

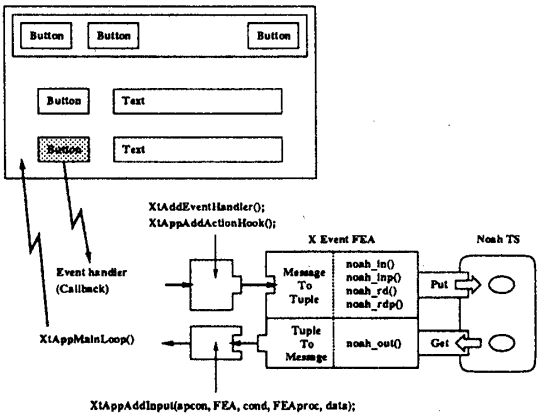

図 3: X イベント型 FEA の例

4 おわりに

Noah におけるエージェントの監視・制御機構を提供する Noah Agent Manager (NAM) の概要を述べた. NAM では, リソース・バンク (RB) とエージェント・マネージャ (AM) が, Noah 環境で自律協調動作をするエージェントに対し, 直接的な制御を可能とする. また, フロントエンド・エージェント (FEA) の介在によって, 元々 Noah 環境向けに作成されていないアプリケーションの Noah 環境への取り込みを支援する.

今後は, フロントエンド・エージェントを電子メールシステム等に適用し, Noah システム全体としての評価を行なう予定である.

参考文献

[1] H. Kozuka et al., "An Architecture of Distributed Computing Environment — NOAH: Network Oriented Applications Harmony," Proc. IEEE First International Workshop On System Management, 1993.

[2] Sudhir Ahuja, Nicholas Carriero, David Gelernter, "Linda and Friends," IEEE COMPUTER, Vol.19, No.8, Aug.1986.