

## プラント運転支援向け分散協調機構

坂下善彦, 渡部修介, 徳本修一, 三石彰純  
三菱電機(株) 情報技術総合研究所  
神奈川県鎌倉市大船5-1-1

### Distributed Autonomous Processing Mechanism for Plant Operation System

Y. Sakashita, S. Watanabe, S. Tokumoto, A. Mitsuishi,  
Information Technology R&D Center, Mitsubishi Electric Corporation  
5-1-1, Ohfunata, Kamakura, Kanagawa, 247-8501 Japan

{sakashita, swata, tokumoto, mitsuishi}@isl.melco.co.jp

#### 1. はじめに

プラント運転における制御手順には, 基本的な制御手法の他に, オペレータの経験に基づく判断, あるいは試行によるその結果の移推傾向を観察しながら次の策を探し出す場合がある. 更に, 制御システムの場合は, 何らかのフィードバック系が存在するので, 焦点を当てている問題領域より, 前方あるいは後方に, 副次的に焦点を設けて対策を検討する. この様に, 問題解決のアプローチは, 現象の変化の抽出, 変化における問題の有無, 状況の将来予測, 原因の推測, 現象を継続あるいは変化させるための対策, 等多岐に渡る対応が要求され, 対象の状況と共に柔軟に適用せねばならない[1][2].

更に, 従来の DCS(Distributed Control System)による制御操作支援システムに加えて, 過去の運転事例に基づく事例推論による支援[3], 対象プラントの計算可能な数理モデルなどに基づくモデリングによる支援[4] [5], 等が研究開発されている. このような, 様々な手法に基づく支援系が存在し, それらが互いに機能を補完しながら協調された制御操作支援システムへの期待が大きい.

本研究は, 各種の制御操作支援サブシステムを分散処理環境で協調的に動作させる分散協調の機構を, 石油プラントシステムを適用対象にして構築することを目的としている.

プラントシステムにおけるオペレータは, 対象としている実システムの運転状況に対して, 運転目標を設定し, その実現に向けた操作制御を行う. この際に, 先の目標を達成するために, この加えて変化が妥当であるか否かを確認する. この意図的な変化の加え方に注目すると, オペレータの知識と経験に基づき, 作戦→実行→観測という流れに沿って, 実行される.

問題解決のアプローチは, 対象の状況に柔軟に対応しなければならない. プラントの場合, 基本的な対処方針の類が存在し我々はそれをシナリオと呼ぶ[6]. しかし, 現実にはこのシナリオが想定する以外の事象が当然ながら発生する. この想定以外の事象に対しても, 情報処理システムの側からどのように対処・支援できるかが本論の課題である.

我々は, システムが対象としている対象世界を関係するサブシステムの共通課題として管理し, この課題に対応できる可能性があるサブシステムが自身の役割に応じて参加する機構を導入する[7]. このことにより, システムにおける制御の流れは, 存在する事象に対して, それぞれの役割を担う構成要素がメンバーとして参加して, 存在する課題を認識し, その解決のための戦略を立て, 役割を分担あるいは共有して互いに調整を行いながら, 目的を達成する環境を構成する[8][16].

## 2. 分散協調の基本的概念と機構

システムが対象としている対象世界を共通課題として管理し、この課題に対応できる可能性があるサブシステムがそれらの役割に応じて参加する機構を以下のように考える。図1に示すように、サブシステムの代理となるタスクエージェントと、サブシステムと通信を行いつつ複数のタスクエージェントの間で連携する環境となるタスクフィールドを導入する。これは、計算主体が局所的な情報を持ち、その部分あるいは全体の情報を共有する「場」となる情報フィールドの概念に基づくものである。この場では、契約ネットプロトコル[9]を基本としたタスク分担方式を定義し、サブシステムへの処理を依頼し、その結果を収集し、そしてその課題に対する評価を行い、統合した情報を解として応答する処理機構[10]を導入する。

次に、この考えに基づき、分散・並列処理におけるフィールドの概念による計算モデルを検討する。分散・並列処理の領域においては、特定の仕事を特定のプロセッサにアサインする場合には、与える入力情報と条件そして仕事である処理仕様を与えて実行させる。図式的には、図2に示すように、入力情報を”In-Tuple”へ格納し、処理を”M(G)”に委ねる、そしてその処理結果が”Out-Tuple”に出力される。本論においては、処理操作の目的は、プラントの現状の一部が入力情報を”In-Tuple”へ格納され、目標となる処理結果が出力情報”Out-Tuple”に目標値として置かれる[11]。この際に、”In-Tuple”に存在するどの入力情報に注目して、最適な処理”M(G)”を捜し出すが、が課題となる。

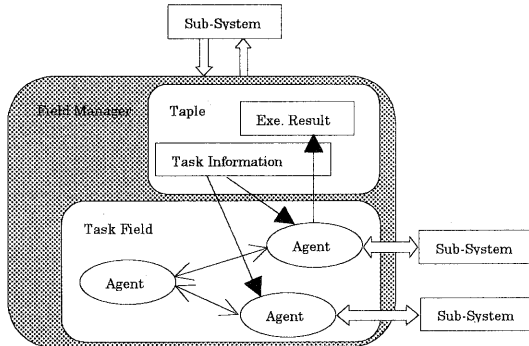


図1. 情報フィールドの概念

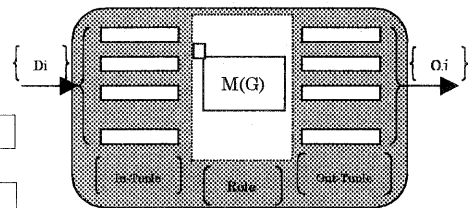


図2. 提案する計算モデル

プラント運転では、大枠の大目標を定めその目標を達成するための個別の操作を行うというトップダウン型の手法を採用しながら、目的を達成するための要件を求めてその要件に沿った処理を捜し出して処理を実行する、という特徴がある。提案するこのアプローチは、上述の場の包含関係を備えた階層構造により実現することにより、トップダウン型の対処方法に適合すると考えられる。

調整の機構や判断の基準やその実施部分は、当事者であるソフトウェアモジュールとは独立にエージェントとして備え、その機構環境を共通的なプラットフォームとして活用できる場とすることにより、この場自身が分散システム環境の実行環境として融合していく、と考えられる。本論では、この場を構成する共有メモリ空間を制御対象の動作空間を表す場として捉え、その場の機能を備える機構により、協調動作の制御を実現することを目的とする。

## 3. 分散協調機構

提案する機構は、協調連携する仕掛けを分散処理におけるフィールド（場）に設け、協調戦略を場のマネージャが司る方式とする[12]。この場は、情報や交信の局所性を持たせ

るために階層的に構成し、且つ関連する階層の間で場を包括的にグループ化することが可能となる。分散型ネットワークシステムの基本的な基盤では、計算主体が局所的な情報を持ち、その部分あるいは全体の情報を共有する「場」となるのが情報フィールドの概念である[15]。この類の代表的な計算モデルに論理的なパターン照合の手法を用いて互いに通信を行う Linda[10]がある。このモデルは均質な環境への適用が可能であるが、情報や関係の局所性あるいは大規模なシステムへの適用には、不必要な通信が生じられ通信の無駄が生じたり、特定の相手への伝達手段の機能において難がある。それを改善する階層化や局所化の研究も多い。

我々は、図 3 に示すように、それぞれに関連する階層関係にある場の中に包含関係を導入して、前述の役割を分担あるいは共有する制御を、場を介した情報の伝達により行う機構とする[11]。

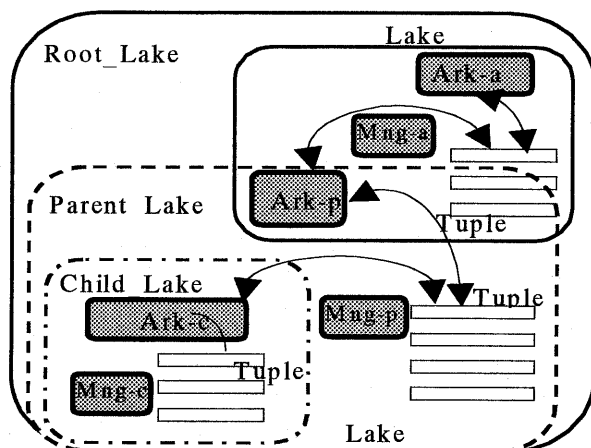


図 3. 包含関係を備えた場のアーキテクチャ

このモデルでは、協調するアプリケーションが”協調の場”を観ながら自律的に参加する形態を採用している。図中 Lake は場に相当した動作環境であり、複数の Ark、及び Lake 毎に存在する Mng と Tuple から構成される。Lake は Parent\_Lake と Child\_Lake のように階層構造を成して環境の継承関係を構成する。Ark は、アプリケーションの機能処理単位であり、場の状態変化を検知して、変化に応じた処理をメッセージ掲示用リスト Tuple を介してメッセージを交換して行う。Mng は、その都度場に与えられた役割を保持し、場が制御する対象の状態情報を Tuple を介して得て、更に関連するメンバーからの対応メッセージを取り込み、与えられている役割に基づき選択及び判断を行って制御を行う。即ち、Lake に存在する資源を用いて、役割に沿った制御を行い、且つ場自身を管理する管理機構である。また、Ark-p のように複数の場に包含関係を備えて所属し、場の状態情報を他の場の状態情報として伝達する機構を備える。これにより、局所化された場の状態情報が他の関連する場へ伝わり、関係のグループ化が可能となる。

## 4. 実装

### 4.1 ソフトウェア構成

ソフトウェア構成を図 4 に示す。Java 環境の上にオブジェクト管理機構を分散オブジェクト管理技術 ORB に基づく HORB[13]を実装して実現した[14]。協調処理サーバ(CPS: Cooperative Processing Server)を介してタスクの処理を依頼し、データ通信を実現する分散協調 API を定義している。

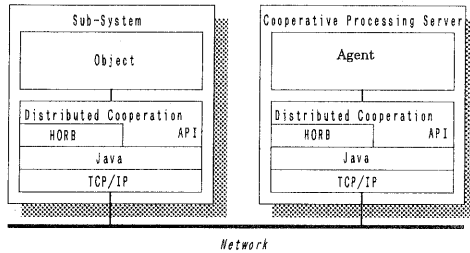


図 4. ソフトウェア構成

#### 4.2 サーバ・クライアント間の構成

サーバ・クライアント間の構成を図 5 に示す. 実際の TCP/IP の通信処理を Proxy/Skeleton が内部に隠蔽するので, 本質的なサーバおよびクライアントとしての機能の実装が容易になる.

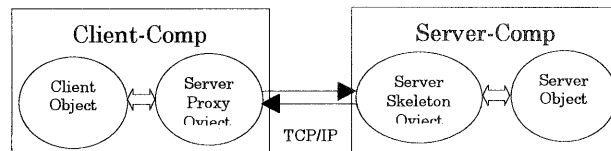


図 5. サーバ・クライアント間の構成

通信処理の流れは図 6 に示すように, 次のようになる.

- 1) SubSystem-A は分散協調 API Demand() を用いて CPS に処理を依頼する,
- 2) CPS は, タスクエージェント間の交渉を行い, 要求された処理に対応できる SubSystem-B に分散協調 API Request() を用いて処理を依頼する,
- 3) 処理結果を分散協調 API SendResult() を用いて SubSystem-B から受取り, 依頼側の SubSystem-A に分散協調 API ReceiveResult() を用いて結果を返す.

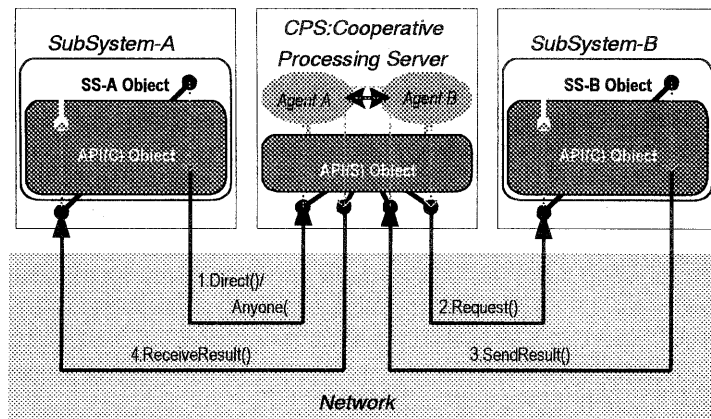


図 6. 通信処理

#### 4.3 タスクエージェント機能

タスクエージェントの本体は各サブシステム(API(C)Object)側に置き, Demand 等によってタスクエージェント間の交渉処理が必要になった場合のみ, CPS が Jini Lookup Service によりタスクエージェントを持つサブシステムからダウンロードして実行する. これによ

りタスクエージェントがサブシステムのノード上でサブシステムプロセスの状態を把握することができ、また起動中のサブシステムの定期チェックが不要となる。

#### 4.4 Demand 処理

サブシステムから CPS への通信後、CPS で利用可能なタスクエージェントを各サブシステムから収集する。具体的には、図 7 に示すように CPS が Lookup Server より、登録されているすべてのサブシステムの Stub を取得し、それらすべての Stub の getAgent メソッドによりタスクエージェントのインスタンスをシリアル転送して取得する。

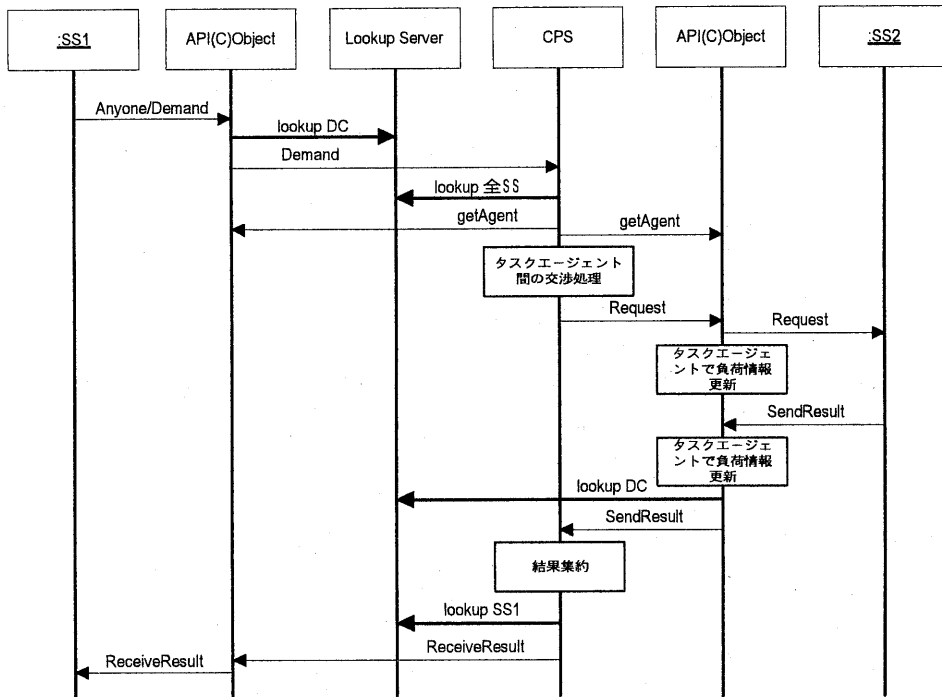


図 7. Demand 処理

#### 4.5 通信時間

クライアントがサーバへデータを送信し、受信したサーバがクライアントへデータを返信するまでの時間を表 1 に示す。また、クライアントがサーバのメソッドを呼び出し、その戻り値を得るまでの時間を表 2 に示す。[環境] OS:Windows NT4.0、NET:10Base-T

表 1. データ送受信時間

CPU:Pentium 120MHz、

msec	非分散	分散環境
データグラム	2.3	1.8
ストリーム(バッファ付き)	1.8	2.0
ストリーム(バッファ無し)	399.8	301.0

表 2. メソッド操作時間

CPU:Pentium II 266MHz、

	時間 : msec
C Socket	0.30
Java Socket	0.30
HORB	0.40

### 5. まとめ

従来の情報処理システムでは、予め定められている業務処理および制御処理の流れが定義されており、特定の業務や制御を指定することにより実行される。また、その過程で人による指示や判断処理、あるいはイベント処理がなされるが、それらは既知のものである。

しかし、一般には、業務の流れや制御の流れは、必ずしも全て予め発生しうる事象が既知であることは無い。今後の情報処理システムは、このような予め知る事の出来ない事象に遭遇しても対応できる能力を備えることが、そのシステムの能力を高めるためにも、あるいはシステムの適用幅を広げるためにも期待される。

この予め知る事の出来ない事象に対してどのように対応できるかという課題は、本論で対象としているプラントシステムでも、解決すべき大きな問題である。

本研究では、従来の事例推論の技術による対象事象への対応に加えて、対象システムの設計意図を備えた物理的計算モデルを備え持ち、そのモデルが導出する現象の理解と対応策候補の導出と連携させることにより、実オペレータに対する対応策案を提示して、解決案とそのアプローチを決定する方式を採用している。プラントシステムや制御システム等の場合は、これらの対象システムを物理モデルに基づく計算可能なモデルと表現できる可能性が高い。この類の対象の場合は、振舞いを予測できる可能性があるため、システム側が備える振舞いのモデルの基礎とすることが期待できる。

なお、本分散協調処理技術は、新エネルギー・産業技術総合開発機構委託研究「ヒューマンメディア(1996-2000年度)」の「石油プラント高度情報化システム技術研究開発」の一環として研究された。

#### 参考文献

- (1) 溝口, 五福, 山内, 坂下, “次世代プラント用ヒューマンインタフェースの研究開発”, システム制御情報学会 システム/制御/情報 Vol.42, No.5, 1998.
- (2) R. Mizoguchi, A. Gofuku, Y. Matsuura, Y. Sakashita, M. Tokunaga, “Human Media Interface System for the Next Generation Plant Operation”, 1999 IEEE International Conference on Systems, Man, and Cybernetics (SMC99), pp.V630-V635, 1999.10.
- (3) M.Lind, “Modeling goals and functions of complex industrial plants”, Applied Artificial Intelligence, Vol.8, No.2, pp.259-283, 1991.
- (4) E.A.Woods, “The hybrid phenomena theory”, Proc. IJCAI-91, pp.1138-1143, 1991.
- (5) 五福, “工学システムの機能モデルからの挙動の導出”, 人工知能学会誌, Vol.11, No.1, 1996.
- (6) Y.Sakashita, T.Ideguchi, F.Sato, T.Mizuno “Evolving Internet Application System based on Scenario-Role-Service Model”, 電子情報通信学会英文論文誌, Vol.E83-D, No.6, pp.1255-1265, 2000.
- (7) 坂下, 井手口, 佐藤, 水野, “役割に基づくアプリケーションの振舞い制御”, 電子情報通信学会論文誌, Vol. J-B-I, 1999.6.
- (8) S. Tokumoto, S. Watanabe, A. Mitsuiishi, Y. Sakashita, “Integrated Architecture for Distributed Cooperative Processing”, 1999 IEEE International Conference on Systems, Man, and Cybernetics (SMC99), pp.V654-V659, 1999.10.
- (9) R.G.Smith: “The contract net protocol: high level communication and control in a distributed problem solver”, IEEE Trans. on Computer, Vol.C-29, No.12, pp1104-1113, 1980.
- (10) Sudhir Ahuja, Nicholas Carriero, David Gelerder, “Linda and Friends”, IEEE COMPUTER, Vol.19, No.8, Aug.1986.
- (11) F.Sato, H.Kozuka, K.Miyazaki and H.Fukuoka: “Noah: An Environment for Autonomous Systems”, Proc. of 3rd International Symposium on Autonomous Decentralized Systems(ISAD'95), 1995.
- (12) 坂下, 金枝上, 鈴木, 上野, “分散システムにおける協調的業務とサービス”, 情報処理学会, マルチメディア・分散・協調とモバイル(DICOMO) シンポジウム, pp.227-233, 1998
- (13) Hiranano, S., “HORB: Distributed Execution of Java Programs”, World Wide Computing and Its Application, 1997.
- (14) Finin, T., et al., “DRAFT Specification of the KQML Agent-Communication Language plus example agent policies and architecture”, The DARPA Knowledge Sharing Initiative External Interfaces Working Group, 1993.
- (15) 桑野, 佐藤, 加藤, 永井, 本位田, “エージェント指向言語 Flage”, マルチエージェントと協調計算のワークショップ MACC'97, 1997.
- (16) 鶴林, 玉井, “ロール概念に基づく発展型移動エージェント”, コンピュータソフトウェア, Vol.18, No.21, 2001.