# 共同作業と協調作業への考察

坂下善彦，池田峰次，太田賢，水野忠則

三菱電機 (株) 情報技術総合研究所
静岡大学 情報学部情報科学科

分散問題における共同作業と協調作業の違いを論じ，協調作業を分散コンピューティング環境において制御する方式を検討する．　分散問題解決で実施される通信のパラダイムは，グローバルメモリの共有とメッセージパッシングである．グローバルメモリの共有で，良く利用されるブラックボード・モデルにに基づき，メッセージを書き，解をポストし，情報を見つけるためのブラックボードを共有メモリと看做すことになる．本論では，この共有メモリ空間を，制御対象の動作空間を表す場として捉え，この機構により，協調動作の制御を実現することを目的とする．このための基本的な適用対象として，鬼ごっこ・御輿担ぎ・ゴミ集めを選び，それぞれの特徴と実現の方法について論ずる．

# A Study of Collaboration and Coordination

Yoshihiko SAKASHITA,
Minetsugu IKEDA, Ken OHTA, Tadanori MIZUNO

Information Technology R&D Center
Mitsubishi Electric Corporation
5-1-1, Ofuna, Kamakura, Kanagawa 247, Japan
Department of Computer Science
Faculty of Information
Shizuoka University
3-5-1, Shirokita, Hamamatasu, Shizuoka 432, Japan

Describing the differences between collaboration and cordination, we argue the mechanism of communication for mutual coordination on the distributed computing environments. The forcussed paradigm of communication for the distributed computing systems may be common grobal memory and message passing. Based on the concept of Blackboard-Model, we consider a blackboard in which the software module will send the messages, post the possible solutions, and find the information as a common memory. In this paper, we consider the common memory space as a field for reflecting the work space for control objects. We intend to examine the coordination mechanism based on this idea for appling to Onigokko-play, Mikoshi-Carrying on the shoulders, and Garbage-collection.

## 1 はじめに

情報処理システムのサービス形態，あるいは対象としている作業の形態は，予め作業の手順や処理の方法が分かっているものと，手順や処理の方法が予め特定できず，発生している状況や情報の内容に依存した対処が要求されるものがある．今日までは，主として前者の方を主な対象として情報処理システムは発展して来た．

今後は，企業活動の意思決定にも係わる領域にも情報処理システムが浸透していくことは自然な流れである．その領域では，従来のような予め処理の形態が予測されることよりも，受信した情報や発生しているシステムや系の状態に依存して，対応が求められることが必須となる．

同様なことは，制御の分野でも起こる．単なる搬送の制御や制御する対象の状態が従来の制御装置によるプログラム制御ではなく，産業システムにおける群制御のように，複数の制御部分が連携してあるいは組みして，物を持ち上げる・回転させる・塗装する等の作業を行う場合は，対象の状態を把握して目的に沿った対処・制御が必要となる．

本論は，産業システムを対象とした場合の複数制御要素間での連携を協調動作という視点から観て，分散処理の研究を基にして発展している「場」の概念を用いて[1]，制御システムにおける協調動作を実現することを研究の目的として，基本機構の適用の検討と，基本的な振舞いの例に対する適用を紹介する．

## 2 自律型情報処理システムの要件

このような予測のもとで，将来情報処理システムが備えなければならない要件を，次の２つの視点から概観する．

### 2.1 制御の分割問題としての共同作業

基本的には，従来からの分散システムにおける著名な課題である．これは，並列処理の領域で多くが研究されて来ている．

ここでは，制御対象を観察分析して，構成する機能の単位部分を括り出し，そのそれぞれに対して該当する処理作業を割り当てて実施させる．基本的には，それらの処理は並行して実行され，その結果が集められて目的とする作業が完遂されることになる．制御対象に対する機能と振舞いは予め知れていて，それらを実現する作業部分は，個別独立に定められたことを実行することとなる．

ここでは，処理実行単位が対象の一部分を作業単位として任され，個別に他との交渉関係も無く実施する形態を，共同作業と呼ぶ．一般的のこの類の作業形態では，指揮を司る役目がどこかに存在し，その管理の下に作業をする．メンバーの間で何等かの調整が必要になった場合は，その指揮を担うメンバーに依頼する形態で実施される．

### 2.2 制御の伝搬問題としての協調作業

先の共同作業における交渉の側面に焦点を当てる．交渉は本質的にはメンバー自身で処理が出来なくなり関連する当事者メンバーの間で交渉を直接に行って，課題を解決していくことを目指す．即ち，交渉をその都度指揮を司るメンバーに依頼するまでもなく，関連する当事者は，課題を抱えている近辺のメンバーに限定される，という性質があるからである．

この交渉は，その結果その時の関係者の域を越えて影響を及ぼすことが充分に想定される．その場合は，次の段階としての交渉を開始する．この現象を本論では，協調作業における制御の伝搬と呼ぶ．

一般に，作業に関係するメンバーは，目的に応じた動作環境に置かれていて，その枠組みの中で振舞うことになる[2]．メンバー間の交渉が発生する場合，直接にメンバー同士でメッセージを交換して交渉すればよいが，本論で提案する方式は，自らが置かれている動作環境をある場として捉え，この場とメンバーとが連携して，即ちメッセージの交換も含めて行う．メンバーは他のメンバーの特性などを意識せずに，ただその場とのやり取りのみに集中すればよいことになり，処理機構としても極めて汎用的なメカニズムになると期待される．

ここでは，サービス要素を明示的に指定して協調させる機構ではなく，ある「場」に適応できるサービス要素が自律的に参加することにより[3]，作業としての制御が伝搬して，その結果として作業が「協調される」形態を求める．

### 2.3 特徴

1. 共同作業あるいは協調作業のメンバーの間で直接にメッセージを交信して調整するのではなく，メンバーが置かれている環境及び制御の対象となっているオブジェクトの状態を介して，メンバーは自律的に振舞う．

2. BlackBood Model/Contract Net による制御

方式による場を介した制御の実現を目指す.

3. 分散環境での自律型の協調作業は，協調動作が必要になるまでは自立的に振舞いを継続するので，メンバー間でのメッセージの交換は行われなく必要時にのみおこなわれるので，結果として総体では通信量は少ないことが期待される.

4. 分散環境における代行処理が類似のメンバー (Processor, SubSystem,etc) により可能なために，システム全体の対故障性を向上させることが期待される.

## 3 自律システムへの期待

産業向けあるいは制御システムを対象とした自律システムの研究が盛んに行われている．そこでは，自律システムに対する要件やそれを実現するためにコンピュータシステムに要求される機能なども検討されている [4].

最近は特にネットワークに関係する技術，あるいはそれとコンピュータが連携して実現する技術が飛躍的に発展していることも背景にした制御系全体を一つのシステムとして扱う試みもなされている.

種々の機能を供えた制御サブシステムの複合体である制御システムにおいて，個別サブシステムの間で制御を連携させることへの期待は大きい，と予想する.

分散問題解決で実施される通信のパラダイムは，グローバルメモリの共有とメッセージパッシングである [5]．グローバルメモリの共有で，よく利用されるブラックボード・モデルに基づき，メッセージを書き，解をポストし，情報を見つけるためのブラックボードを共有メモリと看做すことになる [6].

協調すべき当事者どうしが直接にメッセージを交換して交渉する従来の調整の形態では，全てのメンバーが調整能力を備え個別事象に対する調整に必要な判断基準を持たねばなくなる．このような形態では，均質な一貫性のとれた調整や判断の実現が極めて難しい，と予想される．また，特に分散型のシステムでは，メンバーあるいは実行主体であるソフトウェアが追加されたり，除去されたり，あるいは改定されたりして，いわゆる動的な性質を特徴的に備えている.

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の外側に備え，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.

本論では，この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 [7]，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．その場の基本的な枠組みを図 1に示す．Ark は，アプリケーションの機能処理単位であり，場の状態変化を認知して，それに応じた処理を Tuple を介して行いその結果を場に返す．このモデルは，協調するアプリケーションが" 協調の場" を監視しながら自律的に参加する形態となっている.

## 4 協調制御

本章では，制御システムにおける基本的な協調の形態を述べ，提案する協調機構がどのように作用するかを概説する.

### 4.1 初歩段階

単純な役割を与えることによる実現．遊びは，特徴的な協調作業の一例である．個々のメンバーの役割は明確であり，行動はメンバーが存在している物理的位置によって，具体的にどのように振舞うかが決められる.

「鬼ごっこ」は図 2に示すように，限定された空間中で 1 人の鬼と複数の他による遊びを実現する.

- 鬼の役：自分以外のメンバーを捕まえる.
- メンバー：鬼から逃れる.

場の条件

- 自分が何処に位置していて，鬼を含む他のメンバーが何処にいるかは，場の情報として，誰にでも見える．限定された空間は，例えば将棋の盤とする.
- 同じ位置に複数のメンバーは存在できない.
- 単位時間経過後に鬼以外は移動してなくても良い.

### 4.2 神輿担ぎ

同一の制御対象である神輿を担ぐという目的に対して，担ぐメンバーは大まかな神輿の動きを想定するが，個々のメンバーが置かれている物理的条件によって，その都度，対応する動きが異なる．この場

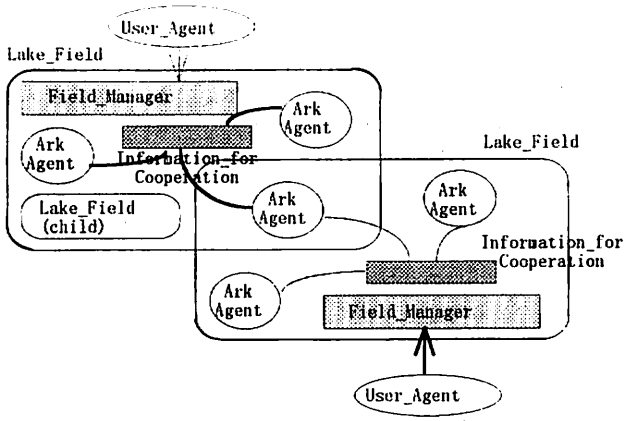


図 1: 場の計算モデル

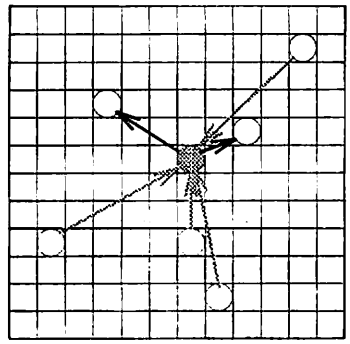

図 2: 「鬼ごっこ」

合は，神輿が置かれている状況に反応する形態で，協調作業が進行する．図 3.

役割

- 複数人のメンバーで神輿を持ち上げる

場の条件

- 神輿がどのような状態にあるかはメンバーに場の情報として見える．
- メンバーは互いに緩やかに同期して移動している

### 4.3 ごみ集め

この例は，隣接するメンバー間で自分の守備範囲に関する交渉が必ず発生する，ということで例に上げられる．拾う対象となるごみが複数のメンバーで拾う対象の領域に入って来た時点で，互いに交渉して，最終的にどのメンバーが注目しているごみを拾うかを決める必要がある．図 4.

役割

- 自分の近傍に散在するごみを拾う
- 他のメンバーと接触しない

特徴

- メンバー間での何等かの調整が必要となる作業．
- 他のメンバーが目に（ある域値に）入った時に，他のメンバーとの間での調整が初めて始まる．それまでは，メンバー間での調整はなく自立的に行動する．

## 5 制御の分散と伝搬

### 5.1 制御の安定性

一般に制御系では，制御対象が置かれている状態に依存して，複数の機能が独立に活性化されて振舞うことにより，目標に向かった制御が実施される．それぞれの機能の間，制御対象の状態を介して機能間で，互いの結果として影響を相互に与える．即ち結果的に制御が伝搬することになる．状況反応型・状態依存型・イベント駆動型，等の動きがこの伝搬の例であり．

この場合，全体としての動作・動き・が，時間とともに収束することが必須であるが，場合によっては発散する可能性もある．即ち，系全体としてはある時定数で収束するような協調制御が要求される．

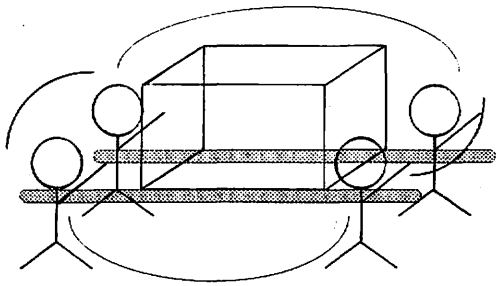

図 3: 「神輿担ぎ」

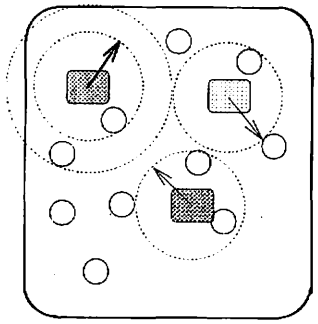

図 4: 「ごみ集め」

### 5.2 振舞い

その振舞いは，制御対象が置かれている現実世界に即するように制約が必要である．

1. 斜面を転がる，スキーで滑る
   - 山の斜面の凹凸に沿って物が移動し・飛び跳ね・方向を変える
   - スキーの板は斜面と並行に移動し・上下する
2. 車が道路を走る，鉄道が線路を走る
   - 路面に沿って車の車輪が接地面では滑らずに面回転しながら，凹凸に沿ってむ，
   - 線路に沿って，同様に車輪は回転しながら進行する
3. ボールを投げる，撃つ，蹴る
   - バットあるいは足によりボールは弾きかえされ，放物線運動に基づいて移動する，
4. 物と物がぶつかる，等など
   - 入射角度と反射角度の関係，反射係数，物の歪み変形，など

以上のような動きが可能になって初めて現実世界との写像関係が成立し，具体的な適用が現実化する．以上述べた現実世界のアナロジーを継承して制御対象が振舞うために，シミュレーションの分野で発展して来た World-Model の概念により実現する．その基本的な枠組みを図 5に示す．ソフトウェアモジュール（図中では Agent）はこのモデルにしたがった制約に従った動きをする．

### 5.3 制御系の動作要素

基本的な要件は次に示す通りである．

- ある状態を保つために，相手あるいは対象の状態に応じた振舞いをする．
- 対象の状態の変化に対して，要求された「状態」を保持する動きをする．

先に述べた World-Model の概念に基づき，制御系における動作を抽出・整理・クラス分けをする必要がある．その作業は別に譲り，ここでは，具体的

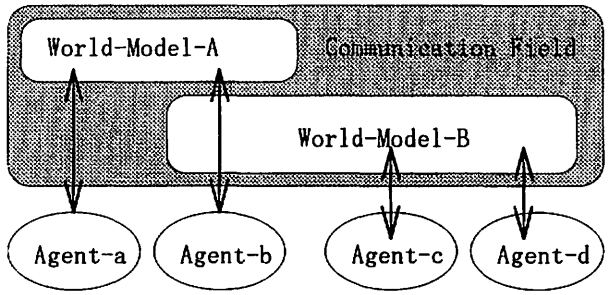


図 5: World-Model と動作

な事例を対象にした場合，どのような振舞いが存在するかを概観する．

1. 単一な状態の制御
   例：接する，結合する，滑る，転がる，流れる，浮く，回転，抜く，差す
   例：鬼ごっこ 鬼はメンバーを追いかける，メンバーは鬼から逃げる．

2. 複数の制御単位の組み合わせによる制御
   例えば，神輿担ぎのように，神輿を担ぐ制御と神輿の移動に沿った移動の制御となる．神輿を担ぐには，メンバーとの同期制御がある．完全に一致させる，意図的に不一致させてうねる，前方・後方を持ち上げる・下げる，左右を上げる・下げる，等々である．
   例：自転車 ペダルを踏んで前進し，ハンドルで方向を制御する． （バランスを保つことは除外して）

3. 制御の要素が時系列的に変化して制御する
   イベント駆動の概念を取り入れて対応し，各イベント間内で単位制御が完了する．即ち全体としての制御は，イベント列とイベント間内での振舞いを要素とした制御の枠組み・流れにより実施される．この時に，いわゆる時変要素をどのように定義すべきか，という課題は残る．

## 6 おわりに

本論では，共有メモリ方式に基づく場のアーキテクチャを基にした自律協調の機構を産業システムへの適用を課題にして検討し，その基本方式をロボット制御の分野で特徴的な形態の対象に適用することを提案した．基本原型に対する適用の可能性が見えて来たが，今後は，より高度で柔軟な振舞いが要求される現実システムへの適用を更に検討する．

## 参考文献

[1] K.Mori: "Autonomous Decentralized Systems: Concept, Data, Field Arcitecture and Trends", Proc. of 3rd International Symposium on Autonomous Decentralized Systems (ISAD93), pp28-29 (1993).

[2] N.Yoshida: "A Modeling Framework for Group Behaviors and Its Application to Cooperative Problem Solving", Proc. of 3rd International Symposium on Autonomous Decentralized Systems (ISA D93), pp70-76 (1993).

[3] 坂根茂幸: "分散協調ロボットシステム", 情報処理学会誌, Vol.36,No.10, (1995)

[4] C.Hewitt and B.Kornfeldd: "Message passing semantics", SIGART Newslett., p.48, (1980).

[5] R.G.Smith : The Contract Net Protocol ; High Level Communication and Control in a Distributed Problem Solver", IEEE Trans. on Computers, Vol.C-29, No.12, (1980)

[6] 宮崎一哉，福岡久雄，辻順一郎，坂下善彦，: "プロダクションシステムに基づいたユーザインタフェース管理システム", 情報処理学会 文書処理とヒューマンインタフェース研究会報告, 23-1 (1987).

[7] 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).