

カプリングによる問題解決とその適用

吉田 誠* 岩根典之**

*沖ソフトウェア株式会社中国支社 **広島市立大学情報科学部知能情報システム工学科

本論文では、問題解決の手法としてのカプリング方式を紹介するとともに、当該方式の適用について紹介する。カプリングモデル、カプリングルール、カプリングによる制御方式、問題解決事例について記述する。本方式の特徴は、ローカルな自律性を維持しながら、グローバルな問題解決を行えることにある。

多様化の時代、e-ビジネスの進展に伴いビジネスの形態も刻々と動的に変化している。仮想空間上で実現される各種仮想ロジスティクス・アプリケーションズ（仮想企業、仮想共同体、仮想工場、等）も普及しつつある。そこには、視野の限界、合理性の限界、働きかけの限界がある。このような環境下、当該ビジネス目的を達成する方法として、個々の機能の convergence（集約）、divergence（拡散）を繰り返しながら有機的・動的に要素を組み直しながら目的を達成するカプリングによる問題解決は有効な手段となる。

Problem Solving by Coupling and its Applications Makoto Yoshida, Noriyuki Iwane

Department of Intelligent Systems, Hiroshima City University
3-4-1. Ozuka-higashi, Asaminami-ku, Hiroshima, 731-3194

There exist several methods for problem solving. This paper describes the new approach, coupling method, which resolves the problem by repeating convergence and divergence of subjects. The special features of this method are that it resolves the global problems that can not be solved by local problem solving techniques. In the e-business world, some enterprises repeats M&A for their living, this is the applied technique of our coupling method.

1. はじめに

塩沢は、人間能力の3つの限界を定義している[1]。

- ・視野の限界：行動の選択・決定にあたって、考慮される状況は部分的なものではない。
- ・合理性の限界：比較や推論や考察の範囲は狭い。
- ・働きかけの限界：作用対象は、つねに限定されており、働きかけは空間的・時間的に局所的である。

そして、複雑さの3つの様相を同時に上げている。

- ・対象の複雑さ：システム自体が大規模で、全体としてどうなっているか分析困難である。

- ・主体にとっての複雑さ：システムの構成単位が外界の状況を判断して行動する時に問題になる複雑さで、判断主体の情報処理能力との関係で決まってくる。
- ・認識における複雑さ：論理的思考の限界ないし学問の限界という点から見た対象分析の困難さであり、われわれの理解能力との関連で問題になる複雑さである。

多様化の時代、e-ビジネスの進展に伴うバリューチェーンの解体と再統合によるデコンストラクションの起きつつある現代はまさに上記様相を挺していると言える[11]。このような問題を克服すべく、各種理論と方式（カオス理論、自己組織化理論、

シナジェティックス理論、オートポイエーシス理論、等）が考案されている[12]。著者らは、これらの問題を解決する方法として、個々の構成単位が convergence（集約）、divergence（拡散）を繰り返しながら有機的に構成単位を組み直しながら、一貫性を保ちながら目的を達成する方法を提案する（図1-1参照）。図1-1は、本方式（カプリング）により視野の限界が解かれることを、デカプリングにより働きかけの限界が解かれることを示している。

本論文では、カプリングモデル及びカプリング制御方式を紹介するとともに、当該方式の適用可能分野を示しその有効性を示す。

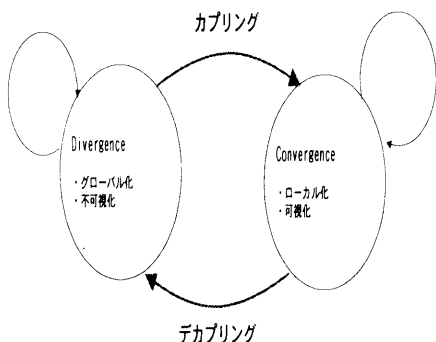


図1-1 カプリング/デカプリング

2. カプリング

2.1. カプリング・デカプリング概念

主体間のカプリング・デカプリングの概念図を図2-1に示す。本論文でのカプリングとは convergence を意味し、デカプリングとは divergence を意味する。図2-1における S1,S2,S3,S4 は各主体を示している。図2-1は、主体 S1,S2 がカプリングにより S1/S2 に集約(convergence)され、S3,S4 が S3/S4 に集約され、S1/S2、S3/S4 が S1/S2/S3/S4 に集約されている状態を示している。デカプリングはその逆操作を意味する。

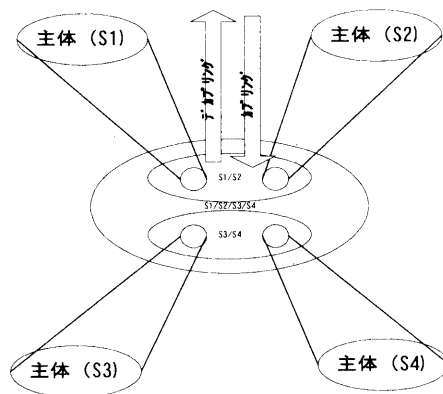


図2-1 概念図

2.2. カプリングモデル

カプリング環境を E を4タプル $E=\{S,O,J,M\}$ としてモデル化する。S は主体、O はオブジェクト、J は主体の状態、M は主体間のメッセージである。主体間の共通要素としてのオブジェクトが存在し、オブジェクトを媒介として主体間にデッドロック（死の抱擁）が発生し、主体間での問題解決が非常に困難な時、又は非常に時間を要する時に、カプリングが有効に働く。カプリング（一体化）は、主体群と当該コンテキストの関係（オブジェクト関係）によりモデル化される。

2.2.1. コンフリクト

オブジェクト O1,O2 に対して主体 S1、S2 が共に関与し、オブジェクト獲得に伴うコンフリクト（衝突）が生じている状態を図2-2 (a) は示している。当該コンフリクト状態を主体間の巡回有向グラフ（wait-for graph）で表わした図が2-2 (b) である。S1 は S2 が O2 をリリースするのを待っており、S2 は S1 が O1 をリリースするのを待っている状況であり、無限 wait 状態（デッドロック状態）を示している（図2-2 (c) 参照）。巡回有向グラフがループを構成する時、無限 wait 状態となる。

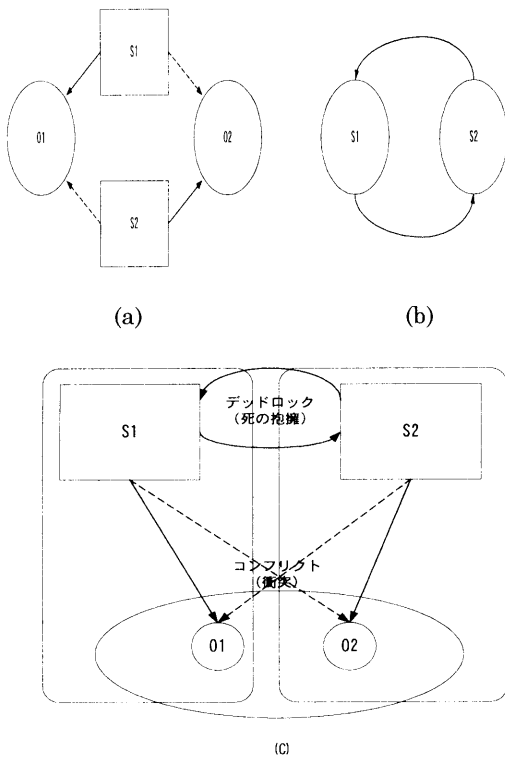


図 2-2 コンフリクト状態図

2. 2. 2. 局所性問題

本論文では、環境を内部環境と外部環境に分離し、ある主体 (S_i) が対象とするオブジェクトに対して制御権 (所有権) を持つ場合、当該主体と当該オブジェクトを内部環境にあると言い、それ以外をオブジェクトは外部環境にあると言う。図 2-2(C) は、 S_1 は O_1 を制御できるが、 O_2 は制御できない、また S_2 は O_2 を制御できるが、 O_1 は制御できない状態を示しており、 S_1-O_1 を S_1 の内部環境、 S_2-O_2 を S_2 の内部環境、 S_2, O_2 を S_1 の外部環境、 S_1, O_1 を S_2 の外部環境と呼ぶ。また、内部環境を当該主体の局所環境と呼ぶ。

各主体の局所環境が分離している場合、図 2-3 に示すようにオブジェクト間のコンフリクトが発生し、主体間が無限 wait 状態になる可能性が生ずる。ここでの問題は、

各主体 S_1, S_2 はオブジェクト O_1, O_2 については認識しているが、局所環境でのみの制御可能性が保証され、グローバル環境での制御ができないことにある。

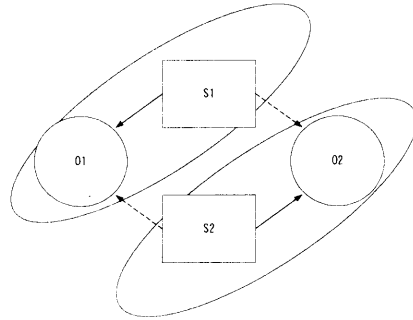


図 2-3 主体間の無限 wait 状態

[カプリング定理] カプリングによる無限 wait 解消定理：

「主体間に無限 wait 状態 (ループ) が発生する場合、次の方法により当該状態を解消することができる。主体間の無限 wait 状態は、主体間の巡回有向グラフによって決定することができる。巡回有向グラフがループを形成する場合、主体間は無限 wait 状態となる。その場合、当該グラフを有限状態機械 (Finite State Automaton) とみなし、当該グラフと等価 (Equivalent) な有向グラフを考える。もし、当該等価グラフにおいて全ての主体がそれぞれの内部環境に対応づけられる状態に閉じたサイクルのみで構成できれば、当該状態を解消することができる。(カプリングによるデッドロック解消と呼ぶ)」

図 2-4 にカプリングによる無限 wait 解消の例を示す。また、当該環境を図 2-5 に示す。図 2-4 における L_i, S_i は主体を意味し、主体 $\langle L_1, S_1 \rangle$ 、 $\langle L_2, S_2 \rangle$ はそれぞれ同一内部環境にあり、主体間にループが構成されている (無限 wait 状態が発生している) ことを示している (図 2-4 (a))。図

2-4 (b)は(a)と等価な巡回有向グラフを示しており、主体 S1, S2 がカプリングを起こし、主体 S1 / S2 となっている状態を示している。図 2-4 (a)において主体 S1 は内部環境に閉じない主体間リンク (S1->L2)、また主体 S2 は主体間リンク (S2->L1) を持っており、かつ主体間ループが構成されているため、無限 wait 状態となっている。一方、当該グラフと等価な図 2-5(b)においては、主体 S1 と S2 がカプリングを起こし、S1=S2 と同等な状態となっており、(a)の (S1->L2)、(S2->L1) リンクは、それぞれ (S1=S2->L2) (S2=S1->L1) となり、内部環境に隠蔽された構成へと変換されている。よって、上記定理により当該状況を解消することが可能となる。

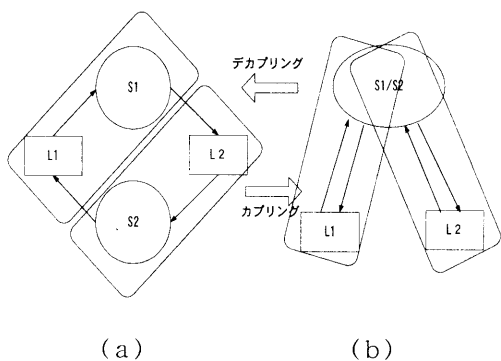


図 2-4 デッドロック解消事例

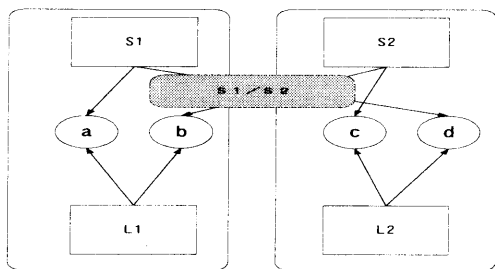


図 2-5 コンフリクト環境

2. 3. カプリング制御

本節では、カプリングによるデッドロック解消制御メカニズムについて記述する。図 2-6 にカプリング制御モデル、図 2-7 にカプリング制御機構を示す。カプリングが行われると、対象オブジェクトに対する操作は全て、内部環境に隠蔽され、当該ローカル主体に集約され、当該主体のアクションとして実行される (図 2-6 参照)。当該モデルは、主体とオブジェクトとコントロールボックスから構成され、コントロールボックスは、チケットボックスとキャッシュボックスから構成されている (図 2-7 参照)。コントロールボックス内では、オブジェクト間のコンフリクト (衝突) 問題、主体間のコンテキスト問題が処理される。各主体は、外部環境にあるオブジェクトをアクセスする場合は、全て当該モデルに基づき処理される。

図 2-7 では、主体 B, C は外部環境オブジェクトをアクセスしており、当該アクセス要求は、主体 A のキュー Qa に入ってくる。主体 A のコントロールボックスエージェントは、要求を Qa 又は Qb から取出し、チケットボックス制御及びキャッシュボックス制御を実行する。処理要求内容によって当該要求は、Qb にキューインされたり、または Qc を通して処理される。

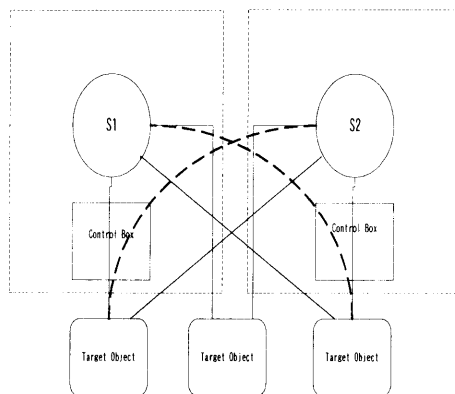


図 2-6 カプリング制御モデル

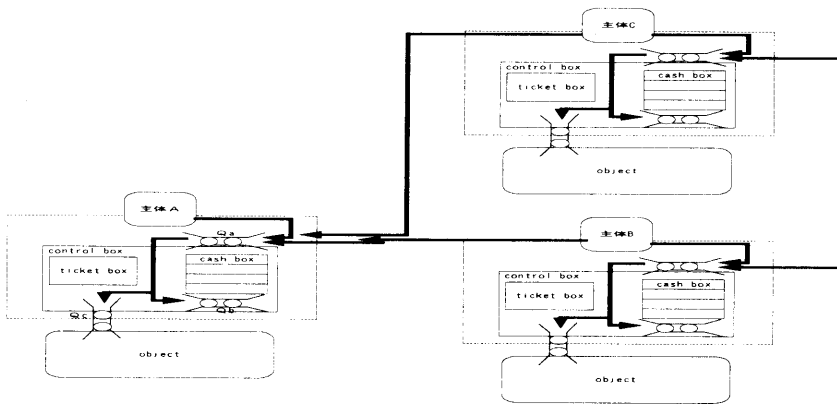


図 2-7 カプリング制御機構

＜チケットボックス制御＞チケットボックスには、Ticket-Box (req-count, req-list) が設定される。Req-count は、主体の動作中アクションカウント数（オブジェクトアクセス数）を示し、req-list はメンバーリスト（オブジェクトをアクセスする主体リスト）を示す。初期状態は req-count=0、req-list=null である。アクションが実行されると、count が+1され、終了すると-1される。また、他主体からの要求があると当該主体名が req-list に追加される。

＜キャッシュボックス制御＞キャッシュボックスはオブジェクトのコンフリクト制御を行う。キャッシュボックスには、Cash-Box(req-list,object-op)が設定され、メンバーリストと各メンバーによるオブジェクト操作が管理される。当該、主体においてデッドロック問題が発生すると、当該主体に纏わる全ての実行主体に対して、デッドロック発生通知が行われるとともに、各主体において当該後処理が実行される。この際、当該デッドロック通知を行ったメンバーに対する Ticket-box、Cash-box は初期化される。

2. 4. カプリング領域

基本的に、カプリングとはローカル化による主体の一体化を意味し、一体化することにより主体間のデッドロックを解消す

ることにある。

オブジェクト数を 'O'、主体数を 'S'、各主体のオブジェクトアクセス数を 'K' とすると、コンフリクトの確率 'C'、無限 wait となる確率 'M' は以下の数式で表わされる[2]。

$$\begin{aligned} C &\propto K^2 S / O \\ M &\propto K^4 S / O^2 \end{aligned}$$

上記式により、カプリングの適用領域としては、オブジェクトを頻繁に必要とする密集領域において有効であることがわかる。

3. カプリング事例

カプリング環境として主体 (S) とオブジェクト (O) から構成されるモデルを考える。主体、オブジェクトは共に一般化した対象を指すものとする。具体的には主体としては、企業、人、エージェント（コンピュータ内のアプリケーション）、オブジェクトとしては、商品、コンピュータリソース、等いろいろ考えられる。主体 (S) は各オブジェクト (O) を使用しながら目的を達成する。このような世界の状況は (S-O) 有向グラフとして表わすことが出来る。

3. 1. サプライチェーンシステム

サプライチェーンとは、顧客-小売り-卸-製造業-部品・資材サプライヤー等の

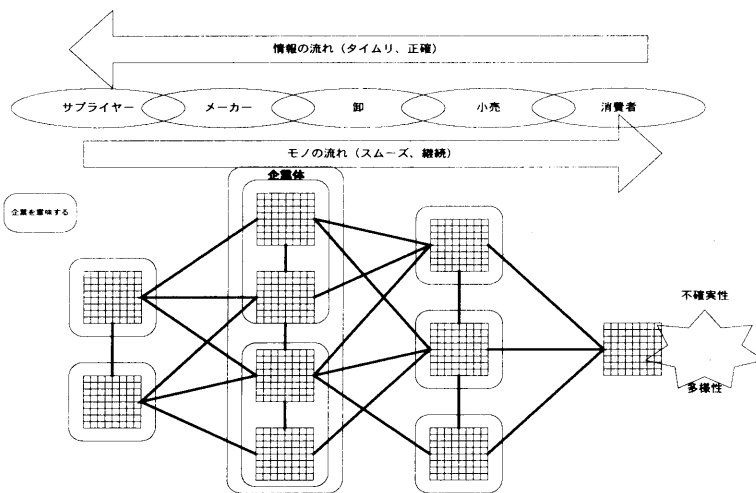


図3-1 SCM多重フレームモデル

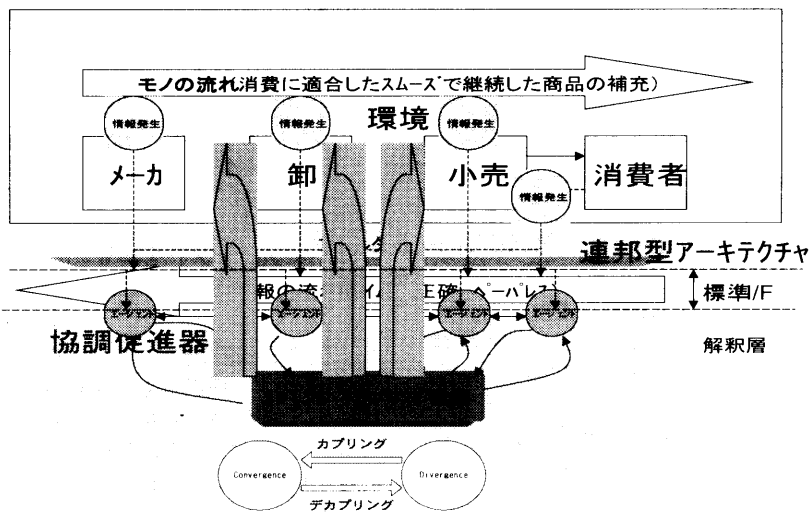


図3-2 SCM統合アーキテクチャ

供給活動の連鎖を示す[8]。そして、SCMの目的は、「不確実性の高い市場変化にサプライチェーン（供給連鎖活動）全体を機敏（アジル）に対応させ、ダイナミックに最適化を図ること」にある[9]。図3-1にSCMアーキタイプ多重フレームモデルを示す[6]。実際のSCMは、多重フレームモデルがタテにヨコに、独立にまたは重複して、複雑にグルーピングされ構成されてい

る。そして、そこにはオブジェクト（材料、商品、運輸、他）獲得に関するコンフリクトが発生し、企業間でのデッドロック（無駄な時間）が発生する可能性がある。1つのアプローチとしては、本論文で紹介したカプリングを多重化した多重カプリングによる方法が考えられる（図2-7に示すカプリングモデル方式）。当該方式を拡張し、統合的にカプリング/デカプリング制御を

行うSCM統合アーキテクチャを図3-2に示す。

3. 2. マルチデータベースシステムのグローバルデッドロックへの適用

異機種分散型データベースシステムであるマルチデータベースシステムにおいては、分散DBアクセスに加えて、既存のDBMSの自律性を保つことが重要となる[4,5,7]。本カプリング制御方式を採用することにより、自律性を保ちながら既存のシステムを変更することなく、グローバルな問題（グローバルデッドロック問題）に対応することが可能となる。

図2-5におけるS1, S2をグローバルトランザクション（複数サイトをアクセスするトランザクション）、L1, L2をローカルトランザクション（ローカルサイトのみアクセスするトランザクション）、a, bをサイト1におけるDBMSのデータ、c, dをサイト2におけるDBMSのデータとすると、図2-4（a）に示すグローバルデッドロックが発生する可能性がある。しかしながら、カプリング定理により図2-4（b）に示す巡回有向グラフが構成でき、既存のシステムを変更することなく、また新たなグローバル機能を追加することなく、当該問題を解決することができる（詳細については文献[5]を参照）。

図2-5（b）の解法の意味は、S2トランザクションとしての要求をS1トランザクションと見なすことであり、S1トランザクションとしての要求をS2トランザクションとして見ることである。これにより、サイトローカルなデッドロックが発生するため、各サイトローカルなデッドロック検出機構があれば、グローバルな検出機構がなくてもグローバルデッドロックを避けることが可能となる。

当該マルチデータベースアーキテクチャを図3-3に示す。

4. おわりに

本論文では、カプリングによるグローバル問題の解決方法及びカプリングによる制御方式について記述した。また、当該制御方式の適用事例としてSCM及びマルチデータベースシステムグローバルデッドロック回避方式について紹介した。

本方式は集約・拡散を繰り返す全てのビジネスモデルに適用される論理フレームワークを提供するものである。カプリングは視野の限界、働きかけの限界を克服する一手段である考える。しかしながら、現実の世界では、convergence, divergenceが有機的かつ動的に行われている。デカプリングのタイミングは主体の意味関係により異なってくる。また、カプリングの種類（tight, loose）にもいろいろある。各種方法論及びオーバーヘッド問題を次の課題としたい。

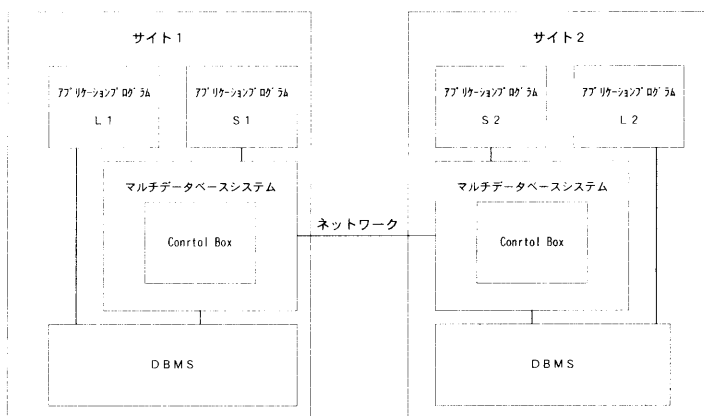


図3-3 マルチデータベースアーキテクチャ

参考文献

- [1] 塩沢由典、複雑系経済学入門、生産性出版、1997.9
- [2] P.A.Bernstein, E.Newcomer, Principle of Transaction Processing, Morgan Kaufmann Pub., 1997
- [3] M.L.スリカンス、M.M.アンプル、シンクロナス・マネジメント、ラッセル社、2001.2
- [4] Y.Matsushita, M.Yoshida, etc., A safe and Fast Concurrency Control Model for Query-Oriented Fully Redundant Distributed Databases, IEEE International Conference on Communications, ICC'83,June,1983
- [5] M.Yoshida, Global Deadlock Avoidance Mechnism in Multidabase Systems, JWCC-6,Fukuoka,July,1991
- [6] 吉田誠、坂本光範、流通業向け SCM ソリューション、沖電気研究開発 183 号、Vol.67 No.2,pp59-64、2000
- [7] Sangkeun Lee, etc., Revisiting Transaction Management in Multidatabase Systems, IEEE Distributed and Parallel Databases,9,2001
- [8] 藤野直明、サプライチェーン・マネジメントの本質と経営へのインパクト、DHB, Oct.-Nov. 1998, pp11-21
- [9] 園川隆夫、制約条件の理論が可能にするサプライチェーンの全体最適、DHB, Oct.-Nov. 1998, pp22-32
- [10] 西尾章治郎、他、岩波講座マルチメディア情報学7ー情報の共有と統合、岩波書店、1999.12
- [11] 唐澤豊、現代ロジスティックス概論、NTT 出版、2000
- [12] スチュアート・カウフマン、自己組織化と進化の論理、米沢富美子監訳、日本経済新聞社、1999