マルチメディア通信と分散処理　57－14
（1992. 9. 25）

概念学習機能を持つ分散協調エージェントシステム

中沢 実　　服部 進実
金沢工業大学工学部情報工学科
〒921 金沢南局区内扇が丘 7-1

あらまし　近年、LANなどのネットワークを用いて広域化されたコンピュータ資源を統合的に処理するネットワークコンピューテングに関しては、クライアント・サーバモデルを基本に研究開発が行われているが、一方、複数の自律エージェントの相互作用を扱うマルチエージェントシステムの研究も次世代コンピューテーションモデルとして重要となりつつある。本論文では、分散協調エージェントシステムのプラットフォームについて筆者の考え方を述べる。すなわち、コンセプトの考え方を導入し、それに明示的にルールを結合させることで、推論を効率化し、いくつかの学習アルゴリズムを行いやすくしている。同時にこの学習機能を拡張して、複数のエージェント間の通信により制約関係を学習するシステムを提案する。

和文キーワード　概念学習　分散協調エージェントシステム　コンセプトとルールの結合
自己組織化メカニズム　マルチエージェント間の制約学習

Distributed Cooperative Agent System with Concepts Learning Function

Minoru Nakazawa　　Shimmi Hattori
Department of Information Science
Kanazawa Institute of Technology
7-1,Ougigaoka,Kanazawa-south area ,Ishikawa, 921, Japan

Abstract

The research and development of network computing with client-server model has been currently accelerated in distributed computer resources on wide area network like LANs. On the other hand, the multi-agent system with correlation mechanism among autonomous agents is recognized to be important as next generation computation model. In this paper, we propose new self-organized mechanism with constraint learning among multi-agents, introducing the learning algolism with explicit concept-rule connections in the process of communication among autonomous agents.

英文 key words　Concepts Learning　Destributed Cooperative Agent System　Concept-Rule connections
Self-organized mechanism　Constraint learning with Multi-agents

1 まえがき

近年の計算機環境において、分散処理やダウンサイジング指向により、ワークステーションやパソコンが、LANやISDNなどのネットワークを用いて多様なコンピュータ資源を統合的に処理するネットワークコンピューティングの開発が盛んに行われている。

しかし、多くのソフトウェアが、サーバ・クライアント方式をとっており、そのプロセス処理をサーバに依存している場合が多い。この場合に、大規模ネットワークへの拡張対応、より複雑な処理を行なうシステムへの要求には、管理的にも、機能的にも、限界があると思われる。そこで、最近エージェント指向システムが注目されている。これまでの多くのエージェント指向の研究は、自律した各エージェントが他のエージェントとお互いに協調しつつ処理を進めるためのものであるが、そのための現実的な手法に関しては未だ技術的に未成熟の段階にあるのが実状である。

また、今日のようにコンピュータネットワークが発達した時代の分散知識ベースは、ネットワーク中で各専門分野を含む知識ベースが協調処理する知識ベースの集合体である。このような分散知識ベースを作成するためは、既存の知識ベースをネットワーク上で相互結合するアプローチを行い、協調を実現するための高次の機能を実現することが課題である。

本論文では、このような上記の課題を解くために、知識ベースの自律的な分散管理、専門分野の異なった知識ベース間の効率的使用方法、各知識ベースの内容の競合の解消方法など、あらかじめ予想できない知識が必要とされるため、分散された知識ベース間でのルール強度の調節法や、概念を用いた学習法を提案している。実装例として交換ノードのB-ISDN (Broadband-ISDN)化におけるマルチメディアを統合した高速パケット通信を実現するためのパケット分配処理部を取り上げる。すなわち、ハードウェアで多数のパケットを同時に並列処理を行わせる高速パケット通信路を、パケット種類ごとの処理の単位でエージェント化し、効率的にパケットを処理する方式として、トラヒック処理状況により、一種のプロセスマイグレーションを自律的に学習制御する例を検討しているがこれについては紙面の都合上、別の機会に発表する予定である。

以下、2章においては、分散協調問題解決の現在の動向より何が不足していているかを知識処理の観点から述べ、3章では、本システムにおける学習方式と、エージェント間における制約に基づく学習方式を提案し、エージェント間における学習の重要性について述べる。

2 分散協調問題解決

分散協調問題解決[2, 3]は、インテリジェントな自律エージェントがグループの相互作用を用いて、協調し問題を解決する人工知能問題のサブフィールドである。分散協調問題解決は人工知能技術と分散処理技術の両方に関与し、従来の分散処理技術の一般的なデータ処理技術の領域とは多少異なる。なぜなら、分散協調問題解決は、データとそのデータに関する手続きが係わるため、データに対する協調ではなく、データ＋手続きの「プロセス(object)」に関する協調を含むからである、従来の分散処理技術は、異なる集合でほとんど完成したタスクを実行するために、固定化されたサーバシステムにプロセスを実行させたり、クライアントシステムに対して決まったサービスのみを実行されるものであった。

これに対して、分散協調問題解決は、図1に示すごとく大きく次の3つの要素の相互関連からなる自律エージェントの階層的プロトコルの処理過程と考える。[4]

自律性：

各エージェントが自身の価値基準、つまりコスト関数を持ち、各々が自分自身の価値基準に従って取るべき行動を選択すること。

分散協調性：

各エージェントの自律性により生じるエージェント相互間の利害の対立を認識し、各エー

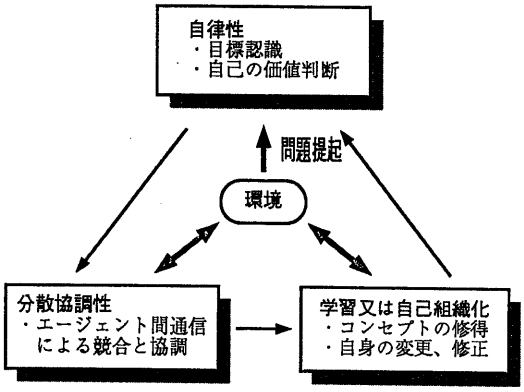

図 1: 自律エージェントの考え方

ジェント間の状況に応じた解決処置を採る機能。

学習又は自己組織化：

過去の知識の利用状況や現在の状況に応じて、他のエージェントとの関係を含めて、最適な実行ができるように自分自身を変更更新し、新たな状態を生成し、コンセプトを構成しうること

現在の所は、**自律性**や**分散協調性**に関しては、研究が盛んに行われているが、**学習又は自己組織化**に関する研究はまだほとんど行われていないと思われる。複数の自律的なエージェントにおいて問題を解く場合、あらかじめどの様な結果になるかを予測するのは一般的に不可能でましてあるエージェントが故障したときの対処方法までを予測することは不可能である。これは、エージェント間での大域的なタイマーが存在しないことや、各エージェント間の制約が問題に対して特殊なものになりさまざまなアプリケーションに対して汎用的には使えないものになりがちとなる。そこで、問題に応じた各エージェント間の制約を自動生成する機能が望まれる。そこで、本論文では、**学習又は自己組織化**が最も、必須の要素と把える立場を取る。

以上のことから、どのようにして分散協調問題を解いていくかを、アルゴリズム的に入力しておくのではなく、状況に応じた問題解決を行うために、問題解決手法を知識ベースで表現し、図2の様なシステムにおいて各エージェント (計算主体) の問題における位置関係やそれぞれのエージェントが所有しているコンセプトまでの学習を行なう。言い換えれば分散協調型帰納推論システムを提案する。

3 分散協調型帰納推論システム

3.1 概念を用いた学習の特徴

アプリケーションに依存する情報をもとに、エージェント間の制約ルールの生成を行ったり、既存のルールを変更したりする学習は、帰納学習の一種になる。帰納学習には、いくつかの方式があるが、我々は、以下に述べる理由により、PI(Process of Induction) の考え方を基本方式とした。[5, 6]

効率的で利用環境に即した学習：

PI では、ルールの生成・更新・変更を、推論過程および概念の有用さに依存して行なう。これにより、処理範囲が絞られ効率が上がると共に、利用環境に即した学習を行なうことができる。

例外的な入力の取り扱い：

実際の分散協調環境でのシステムの利用状況では、各々の自律エージェントでは、必ずしも自身のエージェントのルール通りにシステムを実行するとは限らない。何らかの原因で例外的な実行を行なう場合がある。そこで学習機能は、多くの例の中に含まれているそれらの例外に左右されることなく、ルールの学習が行なえる必要がある。PI は、学習に必要な例の個数、および許容できる反例の個数をルールごとに決定するという考え方を取り、例外が存在しても適切な学習を可能とする工夫がされている。

ルール形式と整合したコンセプト表現：

アプリケーションに関する知識は、IF-THEN ルール形式の表現を用いている。PI は、これ

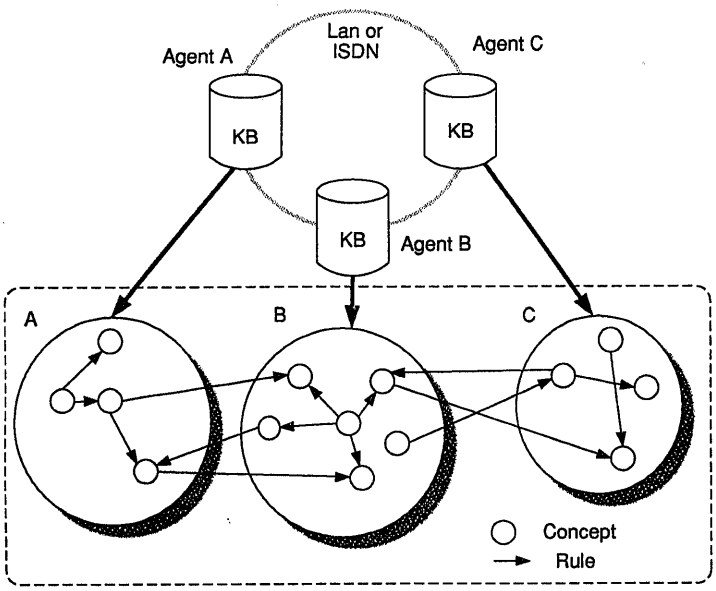

図 2: 分散協調型問題解決システムの概念

に Concept というデータ構造を用いているが、基本的にはルール表現と同様の表現形式の知識を対象としているので、本システムとの整合性がよい。また、もちろん既存の一般知識を手作業で修正する形で学習を進めることもできる。

この PI の考え方を、分散協調環境でも取り扱えるようにエージェント単位の PI のアルゴリズムを複数のエージェント相互間に拡張し、各エージェント相互間がスケーラブルに接続可能なようにシステムを構成した。

3.2 分散協調環境のためのプラットフォーム

図 3に、分散協調環境で各エージェントが通信を行いながら概念生成や概念間のルール生成を行うための汎用的機能すなわち分散協調プラットフォームを示す。図 3が示すように、大きく 3 つの部分と、その間のインターフェースを取るための共有のメモリー群からなる。

学習モジュール：

学習モジュールは、基本的に CommonLisp と C 言語で記述されている。内容はアプリケーションのモデルとなるルールベース部があり、その中に推論エンジンとなる推論部、Concept と Rule で表現された宣言的知識を組み込んである知識ベース部 (KB) がある。それと、ルールベース部の宣言的知識を帰納学習を行なう表 1 の様ないくつかの学習アルゴリズムが収めてある学習機能部、ルールベース部や学習機能部全体を監視する役割を持つ制御部がある。

通信モジュール：

この内容は図 3に示しているように、他エージェントとのコンセプトの伝達、問題、環境情報の入出力などの外部とのインターフェースや、エージェント内部の情報とのリンクなどを行ない、他のエージェントに自エージェントの内部の情報を示したり、他のエージェントへメッセージを送るためのモジュールである。またこのモジュールは、ソフト的に他のエージェ

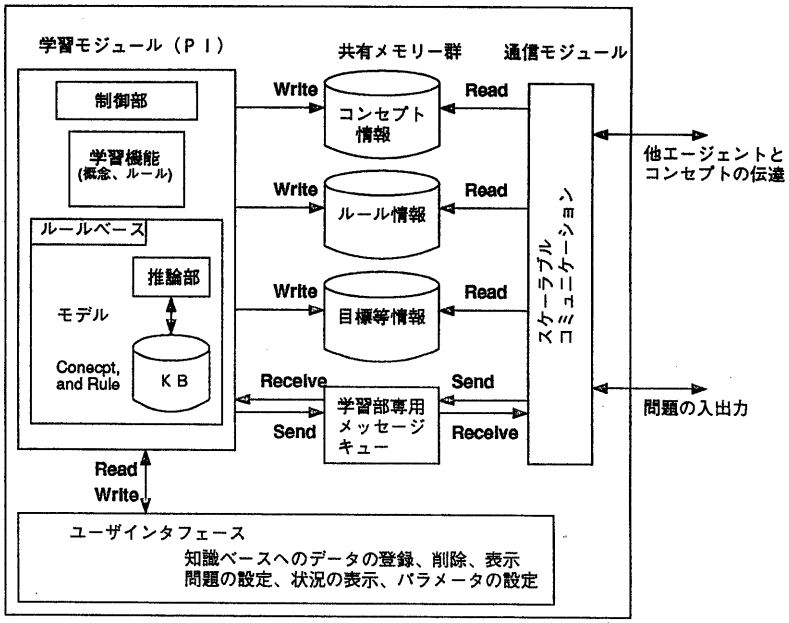

図 3: 分散協調型帰納推論プラットフォームの構成

学習の種類	概念、ルールでの手法
現在のシステムの淘汰	概念の選択方法 ルール強度の調節方法
新ルール、新概念、 概念結合の生成	条件単純化による一般化 事例に基づく一般化 既存ルールの特殊化 概念生成 概念結合

表 1: 学習メカニズムの手法

ントとのやり取りを行なっているときに、そのエージェントが切り離されたり、接続されたりすることに関係なく実行されるスケーラブルなシステムになっている。この通信モジュールは、C 言語で記述されている。

ユーザインタフェース：

ユーザインタフェースでは、オブジェクト指向言語 CLOS と X-Window インターフェースを持つ LispView で記述されている。図 3に示すように、知識ベースへの知識、データの登録、削除、表示、問題の設定、状況の設定、パラメータの設定などを行うことができる。

3.3 概念を用いた推論方法

本システムにおけるデータ構造は、さまざまな宣言的情報とルール群の両方を組織化するために、図 4に示すようコンセプトというデータ構造を用いており、コンセプトにはルールや上位コンセプトのポインタが接続される形になっている。ルールの処理サイクルは、図 5のようになっており、コンセプトレベルでまず使用するルールを限定し、一回の処理サイクルに複数のルールを同時に活性化させるようにしている。このときに複数の活性化ルールが相互排他的かどうかを調べるための相互排他的カテゴリーをチェックする。図 5の (6) で、他のエージェントと通信を行い、他のシステムの各種情報を使用しながら、自エージェントでの自律的な行動を行えるような機能をもつ。

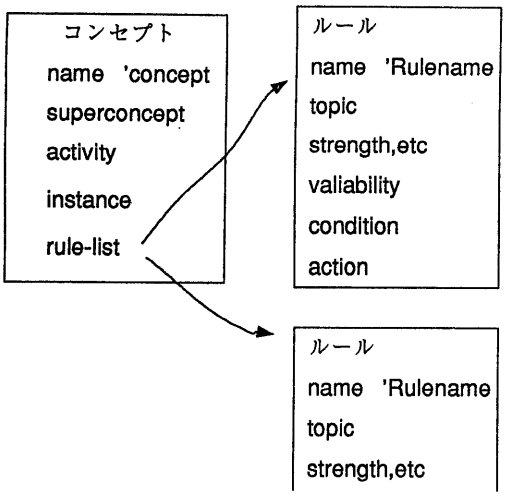

図 4: コンセプトとルールのデータ構造

3.4 概念による学習

本システムにおける概念の学習方法は、先の表1に示すように、大きく分けて、既存のルールの強度の変更と、新ルール、新概念の生成である。ルールの強度の変更はコネクショニスト・モデル的なものに相当するが、学習速度を向上させるためには、ルールの一般化といった帰納的なメカニズムも必要である。本論文では、表1の中の、ルール強度の調節法、事例に基づく一般化によるルールの生成について説明する。

ルール強度の調節法：

問題の状況によって重要なルールは異なると考えられる。ルールがどれくらい使用価値があるかを測定するには、強度によって測ることができる。強度とは、個々のルールに付属しているそのルールの過去の成功の度合いを示す値であり、そのルールの重要さを表すものである。強度などの値を導きだす数式は以下のようなパラメータを持っている。

ルール C の支持度

$$V(C,t) := a\sum A(C,t)$$

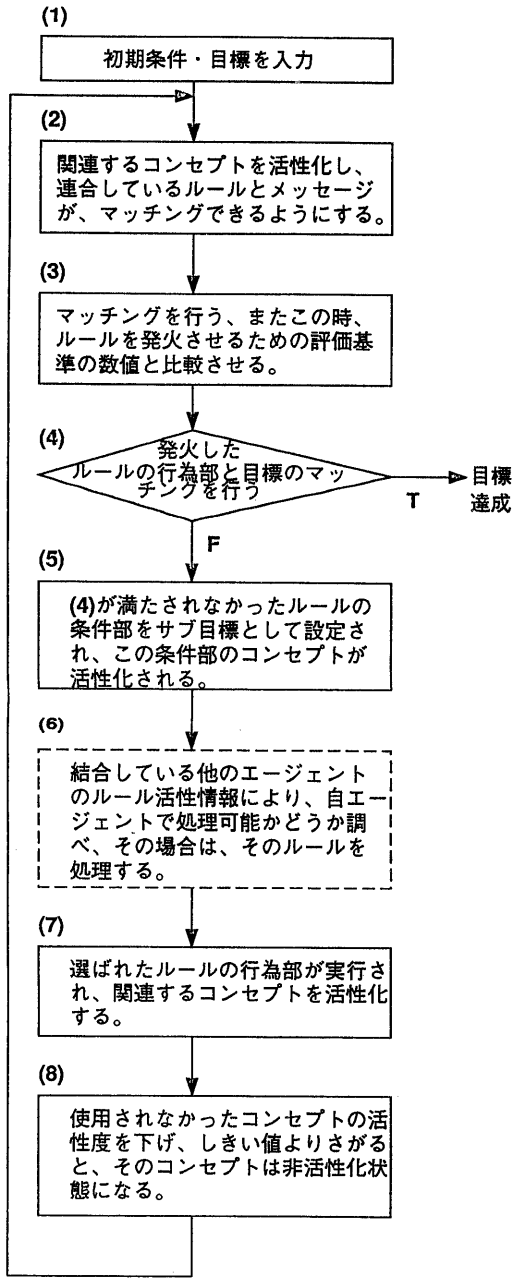

図 5: 処理サイクルのアルゴリズム

a:定数 (<< 1)
A(C,t): 時刻 t でルール C を付加しているコンセプトの活性度

ルール C の付け値

$$B(C,t) := bR(C)S(C,t)V(C,t)$$

b:定数 (<< 1)
R(C):特殊度、条件部の長さによる。
S(C,t):強度
V(C,t):C の支持度

もし、そのルールが目標に対して成功を導くような推論結果の一部であれば、それに相当するルールの強度が増し、目標に対して失敗を導くような推論結果の一部であれば相当するルールの強度は減らされる。図6は成功したときのルールの強度調節方法を示したものあり、逆に失敗したときには該フローの中で強度 S(C*,t+1) の計算を行う際、付け値 B(C,t) の符号を逆にすればよい。ルールは同時に発火するときに内容に競合がある場合、強度の高いルールを優先して用いることで、推論処理の効率化を図ることができる。

事例に基づく一般化：

本方式の一般化は、次の2つの制約を満たす場合に一般化が行われる。始めに、複数個の対象が、2つのコンセプトに共通の事例であること。2つめにその事例のコンセプトはそれぞれ高レベルの活性度を持ってなければならない。2つの条件を満たした場合、図7,8のアルゴリズムによってルールが生成される。この時に、分散的なルールを作成するために、結合されている他のエージェントのコンセプト情報ファイルから、自エージェントの高レベル活性度を持つコンセプトを見出し、2つ以上のコンセプトに共通な事例であるか調べる。その過程で、つながっているエージェント同士の中で一般的なルールが発見できれば、エージェント間の制約

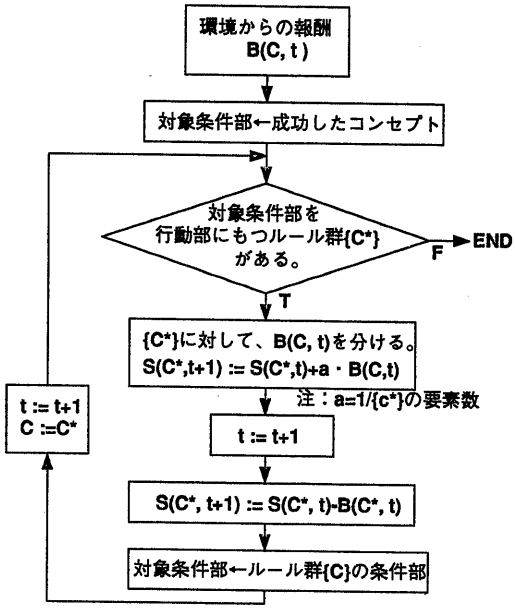

図 6: 成功したときのルールの強度の調節法

的なもしくは協調的なルールと見ることもできる。

4 むすび

本論文では、LAN 等で接続された複数のワークステーション上での、分散協調を行う自律エージェントシステムにおいて、各自律エージェント内に学習機能をもたせ、さらにその機能を拡張して、他のエージェントとの位置関係や、制約関係などを学習することで、分散協調問題解決のためのルールを生成するためのシステムについて提案した。

現在、本システムを用いた B-ISDN 上のマルチメディア情報を分散処理し、一種のプロセスマイグレーションを自律的に学習制御する交換ノードの実験システムの作成を行っており、この結果においては、今後の機会を見て追って発表する予定である。

参考文献

[1] 岩井 壮介他;“知識システム工学”, 計測自動制御学会,(1991)

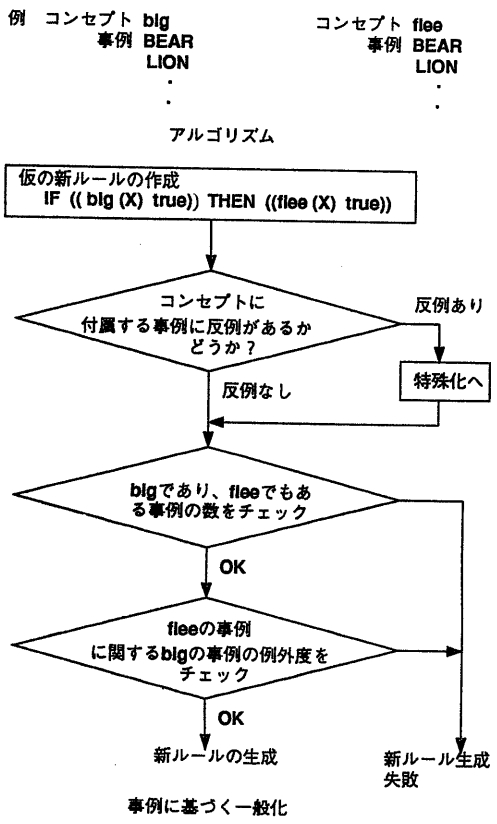

図 7: 事例に基づく一般化のアルゴリズム

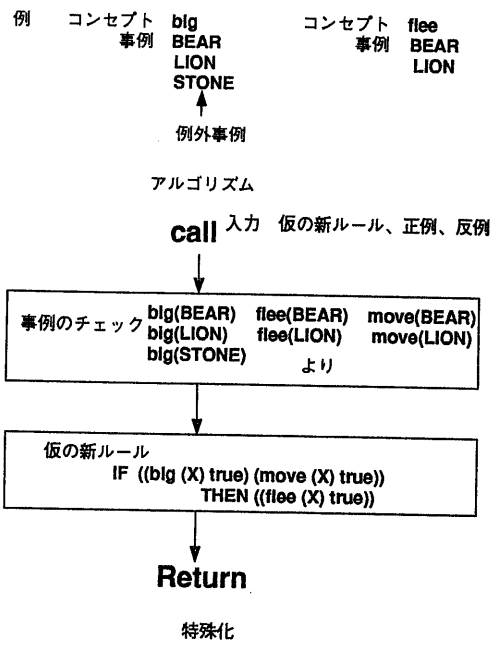

図 8: 特殊化のアルゴリズム

[2] 石田 亨:“協調問題解決”, 人工知能学会全国大会 (第 6 回), チュートリアル資料, pp.B1-B20,1992

[3] Keith S. Decker:“Distributed Problem Solving Technique: a servey. ” IEEE Transactions on Systems, Man, and Cybernetics, SMC-17(5):722-740, September / October 1988

[4] 中篠 将典:“1992 春号 日経インテリジェントシステム 別冊” 日経,1992

[5] J.H.Holland et al.:“Induction:Process of Inference, Learning, and Discovery”,the MIT press 1986

[6] 前田 潤他;“知的秘書サービス”AI89-62, 信学技法、1989