

分散協調処理による異種問題解決系の統合利用

小西 修†

松浦 春選‡

†高知大学 理学部 情報科学科

‡高知大学 理学研究科 情報科学専攻

情報検索システムや各種データベースシステムなどの異種問題解決器を統合するための分散協調処理モデルを提唱し、それが ECA 機構を持つアクティブデータベース上で実現されることを示す。この統合は、利用者と各種応用システム間の仲介者システムによって実現される。仲介者システムは、各問題解決の知識源が黑板上の知識と情報を共有しながら問題解決をはかるという分散協調処理モデルに基づく。ここで、黑板はオブジェクト指向データベース上で考えられ、知識源はエージェントであり、エージェント間の分散協調を制御するスケジューラと問題解決のプランは ECA ルールによって実現される。このアクティブデータベース上の分散協調処理によって、統合利用の経験学習知識の保存管理、エージェント間の並列制御、エージェントの追加による問題解決能力の向上などの特徴が保証される。

Cooperative Distributed Processing Approach for Integrating Different Heterogeneous Problem Solvers

Osamu Konishi

Haruyori Matsuura

Department. of Information Science, Kochi University

This paper proposes a new approach for integrating different heterogeneous problem solvers such as information retrieval systems or different databases. We introduce the cooperative distributed processing model represented on an active DBMS with an ECA rule mechanism. The integration using a mediator model is realized by the blackboard system and multi-agent system. The agents must cooperate via information on the blackboard to achieve their goals. The blackboard system is implemented on top of an active object-oriented DBMS. Thus the scheduler and goal plans for the agents can be realized by an ECA rule mechanism. The proposed model can facilitate significant functions such as saving and reuse of use histories, concurrency control of agents and addition of autonomous agents.

1 はじめに

図書館や美術館では最近、書籍や美術品のデータベース（以下DB）化が進みまた、それをインターネット上で公開している所もちらほらでてきた。このことから、これからは企業などを対象としたDB以外に個人的な趣味や興味でアクセスしたくなるようなものが増えると考えられる。

現在のDBはプロ向きで、独自のアクセス方法があり、効率良く検索するのににも経験が必要になる。しかも、上記のような個人的な興味でのアクセスは数多くのDBにまたがる場合がほとんどだといえる。星の数ほど経験をしないと自分の欲しい情報が取って来れないのでは、誰も利用しない。

これに限らず現在のコンピュータは使用する際に多くの経験を必要とする。そこで我々は経験を必要としないフロントエンドシステムを提案する。これに必要なものはコンピュータ自身が持つ経験だと考える。これによって情報検索システムや各種データベースシステムなどの異種問題解決器を統合し、利用者と各種応用システムとの間の仲介者によってそれぞれの問題解決器を利用者が直接操作する必要をなくする。仲介者は利用者の要求を受けると過去の学習則に従って問題解決器向きの命令に変換、未経験のものは利用者に質問を繰り返し行ない意味を限定した後これを行なう。こうすることで利用者はコンピュータに関する知識を一切必要とせず、各種問題解決器を統合利用できる。この仲介者による統合は、各問題解決の知識源が黑板上の知識と情報を共有しながら問題解決をはかるという分散協調処理モデルに基づく。ここで、黑板はオブジェクト指向データベース上で考えられ、知識源はエージェントであり、エージェント間の分散協調を制御するスケジューラと問題解決のプランはECAルールによって実現される。このアクティブデータベース上の分散協調処理によって、統合利用の経験学習知識の保存管理、エージェント間の並列制御、エージェントの追加による問題解決能力の向上などの特長が保証される。

2 関連研究

異種問題解決系の統合という課題では二つのアプローチがある。一つは問題解決器をエージェント化し、それらの共通知識記述言語によってシステム側の統合をはかろうとするものである [7][8][9]。もう一つはユーザとアプリケーションの間の仲介者専門家をおくことによって、ユーザが異種の問題解決器を利用できるようにするものである。これは従来、情報検索の研究分野で仲介者モデルとして提唱されているもので、その機能の位置づけをエキスパートシステムとしたため、まだ、モデルの提唱ないしはプロトタイプの段階となっている [4][5][6]。我々の研究ではユーザ個人の専門情報の適時の収集利用という環境、また、ユーザ自身のエージェント化も課題として考えるという立場から、後者の仲介者モデルのアプローチをとる。

この仲介者モデルでは、各知識源が黑板上の知識を共有する分散協調処理が考えられる [5] が、我々の研究では、この知識源をエージェント化すると共に黑板上の知識も学習情報として再利用できるようにするため、永続的データとして管理が可能なように、この黑板モデルをデータベース管理システム上で実現することを考えた。これはデータベース管理システムに能動的な機能を持たせたアクティブデータベース [1][2][3] であるが、本研究のように、分散協調処理のためのアクティブデータベース応用はこれからの興味深い課題である。

3 基本モデル

3.1 異種問題解決器

ユーザが気象情報を手に入れたいと考えたとき、必要なデータは気温、気圧、衛星画像などとなり、アクセス対象となるDBは複数になる。しかも、対象が数値データ、画像データなど一つのプログラムによってこれらすべての範囲をカバーするには問題が多い。そこで機能を限定した問題解決器を複数用意してお

き、統合利用することで対象範囲の広さをカバーすることができる。しかし全く分野の違う機能を統合するため、問題解決器間のやり取りに問題が発生しやすい。

3.2 仲介者モデルと分散協調

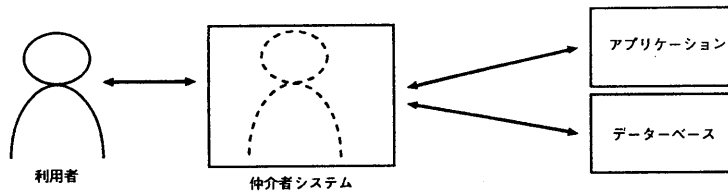


図1 仲介者モデルの概念

異種問題解決器はいくつか提案されているが、我々はその統合にあたって、図1のように従来からある色々な解決器を仲介者モデルを用いて最小限の変更で利用することを考えた。近年、Tcl と呼ばれるインタプリタ言語が注目を浴びている。これは色々なツールに共通のコマンドを提供しユーザは Tcl の文法を修得するだけで、ツールを使いこなすことができる、というものであるが、ユーザの要求を自動的にこれに変換できれば、解決器の変更点は Tcl への対応だけで済む。これには仲介者モデルを利用する。ユーザにはあくまでシステム上の仲介者を相手に色々な要求を自然言語に近い形で出してもらいこれを仲介者が解析して Tcl へと変換する。また仲介者に学習機能を持たせることで、日常繰り返される要求は自動化できる。

仲介者は辞書を参考にしながら変換を行なうことになるが、この辞書にオブジェクト指向を導入する。ユーザからの要求に上位にあるクラスの単語が存在した場合一段下位のクラスの単語を複数提示しユーザに選択してもらおう。こうすることで自然言語にありがちな曖昧な（複数の意味にとれる）単語をなくし Tcl に変換しやすい形にできる。

3.3 黑板モデルとマルチエージェント

人間的な曖昧な要求に応えるためにシステムには広範囲にわたる知識や方法が必要となる。それにはシステムを得意分野を限ったものの集合で構成し、それを協調させることで運用する方法が多く、利点を持つ。現在数多くあるエキスパートシステムもこの手法を採用するものが多い。そこでここでは曖昧な要求からツールの共通な言語までの変換の過程に黑板モデルとマルチエージェントを用いる。幾つかの利点として、黑板によりエージェントの緻密な制御がいらぬ。エージェントは他のエージェントに関する知識を必要としない。エージェント間の会話に一定のフォーマットが必要でない。エージェントの追加にシステム全体の知識を必要としない。があり、欠点には確実に結論を導きだす保証がない。黑板モデルに必要なある種の評価関数を設定しにくい。などが考えられる。確実に結論、つまりコマンドが出る保証が無いのは、ユーザにとって問題である。しかも、黑板モデルに必要なある種の評価関数を設定しにくいいため無駄も多くなる。そのためエージェントの流れを利点を損なわない程度に管理してやる必要がある。ここではエージェントマネージャを用意しシステムの管理を行なうことでこの欠点を克服できると考える。

3.4 分散協調処理のためのアクティブデータベース

エージェントの協調作業を行なうため、エージェント同士の通信に黑板上の共有データを使う。またそれは、問題解決の戦略のための知識データでもある。ここで、この協調作業に用いる共有データ、及び協調作業によって生成された戦略データをデータベースシステムに蓄え、データを高度に管理、共有することを考える。黑板モデルで使われる知識データは階層構造を有しており、オブジェクト表現される。また、オブジェクトモデルの持つカプセル化やメソッドとの統合は、協調作業支援に適していると考えられるた

め、オブジェクト指向データベースを採用する。ここで、エージェント同士だけでなく、エージェントとデータベースシステムの協調を考える時、システム自体がアクティブ性を持つことが要求される。データベースやエージェントがある条件を満足した時、つまりデータベースの内部の状況やエージェントによる外部の状況によって、データベースが自動的に次に対応する処理が用意される等の機能を実現できるアクティブデータベースを考える。具体的には、エージェントのスケジューラ（アジェンダとプラン機能を含む）の役割を果たす E（事象）C（条件）A（行動）機構が必要である。

4 設計概要

4.1 仲介者モデルのフレームワーク

4.1.1 各エージェントの関係

複数のエージェントが黑板を介して協調して処理する。図2に各エージェントと黑板の関係を示す。エージェントマネージャはそれぞれのエージェントを制御・管理する。

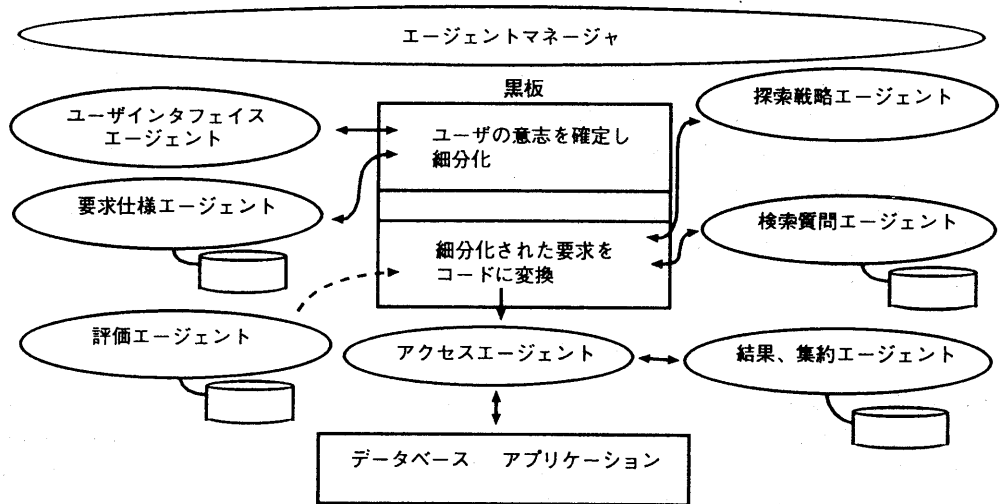


図2 各エージェントの関係

4.1.2 エージェントとその機能

エージェント7つと知識ベース A,B,C,D の4つを組み合わせで推論を行なう。

1. ユーザインタフェースエージェント

ユーザと他のエージェントの間にあって質問、返答の受渡しをおこなう。他のエージェントはユーザとのやりとりにこれを利用する。

2. 要求仕様エージェント

要求の概念を知識ベース A を使って専門用語で表現し、キーワードの組み合わせ（ブール表現 and/or/not）や出力要求度合いの確認（件数、項目など）で適切な結果が出力される工夫をする。

3. 探索戦略エージェント

要求仕様を受け、検索質問、アクセス、結果・集約エージェントの起動を制御し、結果・集約の評価により検索モデルを修正

4. 検索質問エージェント

知識ベース B に基づき利用要求から検索システムを選定し検索システムの質問形式に表現するなど実際のデータベースの検索式を作成

5. アクセスエージェント

知識ベース C を利用し実際の複数の検索システムへアクセスと結果を受理

6. 結果・集約エージェント

結果を集約して、適切なフォーマットで利用者に示す

7. 評価エージェント

利用者の評価を受けて結果の加工とその表示 利用者要求仕様の変更など経験学習モデルを知識ベース D へ記録

4.1.3 知識ベース

知識ベースは 4 つあり、それぞれ専門知識を保持する。

A. 専門領域知識ベース：概念マップ、シソーラス、利用者定義

B. 利用課題知識ベース：要求課題と対応データベースの関係、最新情報変化

C. 検索システム知識ベース：検索システムへのアクセス法、検索システムのコマンド体系、検索手法

D. 経験学習ベース：過去に行なわれた質問の返答など学習則に関するもの

4.2 エージェントマネージャとしての ECA 機構

4.2.1 黒板モデルの仕組み

黒板モデルは従来よりエキスパートシステムなどに利用されてきたが、知識源の if then をいくら厳密に記述しても黒板内の状態によっては、同時にいくつもの規則の条件部が満足され、その結果黒板の中の状態は行為部によっていろいろ書き換えられうる。したがって、一つの状態から出発した問題が多数の並列的な問題解決の中間的な状態になっているということが生じるため、効率よく問題が解けなくなる危険性がある。これを防ぐためにアジェンダとスケジューラが考えられた。アジェンダは各知識源の処理を一時的にブールスケジューラはプラン（目標）に従ってその中から最適な結果を選択、黒板に書き込む。しかしここでは、最適な結果を選択するための評価関数を設定しにくい。そのためそれに替わる機構として ECA(Event Condition Action) ルールを使ったフロー制御を行なう。

4.2.2 ECA 機構の役割

各エージェントは図 3 のように ECA ルールを基本とするエージェントマネージャに管理される。

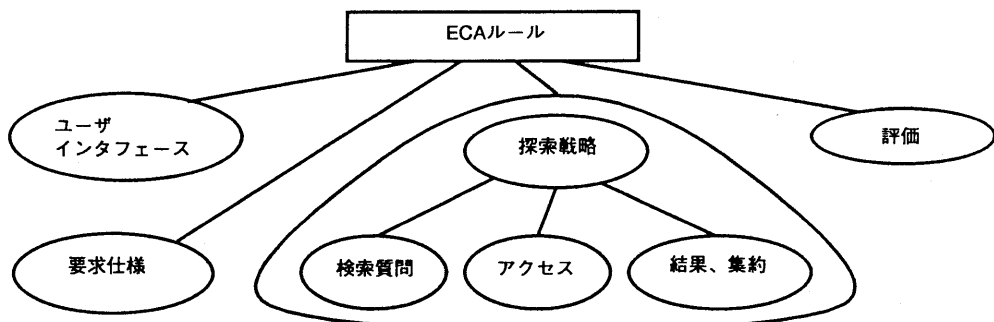


図3 ECA 機構と各エージェントの関係

図4に ECA ルールによって起動されるエージェントの流れを示す。この ECA ルールによって黑板モデルにおけるスケジューラ機能（プランとアジェンダを含む）を実現する。

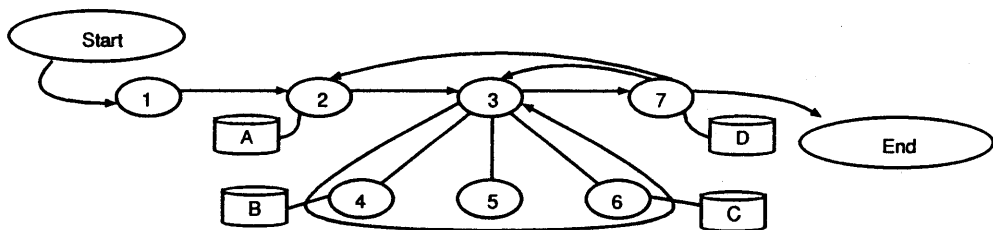


図4 エージェントの流れ

(注:図中の数字は前述のエージェント、英字は知識ベースを示す)

ここでの ECA ルールは次のような基本パターンをとる。

Event: データベースシステムのトランザクション管理、エージェントによるデータ操作
 Condition: 黑板上のオブジェクトの更新
 Action: エージェントの起動

ECA ルールの例：

E: 利用者がシステムをオープン	E: ユーザインタフェースエージェント一時待機
C: 黑板項目を開散	C: 再利用項目 On
A: ユーザインタフェースエージェントを起動	A: 評価エージェントの起動
E: ユーザインタフェースエージェント一時待機	E: 要求仕様エージェントのトランザクション終了
C: 新規要求 On	C: 要求仕様項目オブジェクト内容更新が終了 On
A: 要求仕様エージェントの起動	A: 探索戦略エージェントを起動
E: 探索戦略用の黑板項目を監視	E: 結果・集約ファイルのクローズ
C: 黑板項目の全ての項目更新の確認	C: 結果・集約ファイルの内部更新
A: 結果・集約ファイルのチェック	A: 評価エージェントを起動

5 システム構成

実装には C++ と Tcl/Tk を使用するが、知識はオブジェクト化されプログラム上に格納されていくが、プログラムの終了と同時にこれらが消滅しては困るので、オブジェクト指向 DB を使用し知識などを永続性のあるオブジェクトとして二次記憶に格納する。

5.1 OODB/C++

先に述べたように各エージェントの知識や黒板上の結果など保存しておきたいものは沢山ある。が、エージェントは起動したままで放置すると全く仕事をしないのに頻繁に黒板を検索し CPU パワーを削っていく。これを阻止するために用事の無いエージェントは kill するといいが、こうするとプログラム上で格納したオブジェクト（インスタンス）は消滅してしまう。学習などを行なうのにこれでは困るので、これを二次記憶上に展開する必要がある。このためオブジェクト指向 DB を利用する。OODB ではクラスに永続性を導入し永続性のあるクラスに属するものはすべて二次記憶に格納される。また、全く別のプログラムからも二次記憶上のクラスやオブジェクトを呼び出せる。ここでは黒板と各エージェントの知識ベースなどにこれを用いている。同時に使用する Tcl はそれのみでは大きなシステムの記述に向いていないのでユーザから内容が見える必要の全く無いコアの部分を C++ と OODB で記述する。

5.2 Tcl/Tk

Tcl/Tk は作成当時 UCB にいた John K. Outerhout によって作成されたインタプリタ型の言語である。Tcl は Tool Command Language の略、Tk は Tool Kit の略である。Tcl は似たものに Perl などがあるが、大きな相違は拡張性ともいう点にある。ユーザは自分の作ったツールに少し手を加え、Tcl のライブラリとリンクすることで、Tcl の命令にユーザの作った命令が増やせる。これにより現在数多くの拡張がなされているが、利用のために覚える必要のあるのは当然 Tcl のコマンド体系のみである。本システムでは Tcl を二通りに使用している。システムがツールを利用する際にこれを使うことで使用できるツールの数は一気に増える。またツール毎にコマンドを用意する必要も無くなるのでシステム側は Tcl への変換のみに留意すればよい。それと Tcl 本来の使い方として Tk とあわせてシステムのインタフェース部分をこれで記述する。またエージェントの大まかな記述は Tcl によって行なうことで、ユーザのエージェント本体のカスタマイズや新規作成を簡単にする。

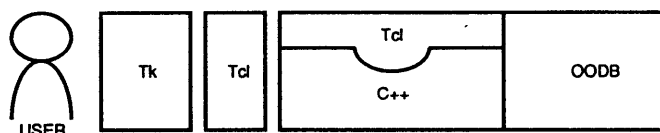


図 5 システムの構成

6 あとがき

情報検索システムや各種データベースシステムなどの異種問題解決器を統合するための分散協調処理モデルを提唱し、それが ECA 機構を持つアクティブデータベース上で実現されることを述べた。仲介者システムは、各問題解決の知識源が黒板上の知識と情報を共有しながら問題解決をはかるという分散協調処理モデルに基づいて、現在 C++ と OODB を利用してこのコアの部分の記述中であるが、これから問題になるのは、エージェントの知識や ECA ルールの作成である。プログラム部分よりもデータ部（知識）が重要になるため、これらに関するデバッガや新規作成のためのツールの充実が必要であろう。

参考文献

- [1] E. Anwar, L. Maugis, and S. Chakravarthy. "A new Perspective on Rule Support for Object-Oriented databases." SIGMOD RECORD 5-93, pp.99-108, 1993.
- [2] D. R. McCarthy and U. Dayal. "The Architecture of An Active Data Base Management System." Proc. of 1989 ACM-SIGMOD, pp.215-224, 1989.
- [3] U. Dayal. "Active Database management systems." Proc. 3rd Int. Conf. of Data & Knowledge Bases, pp.150-169. 1988.
- [4] W. B. Croft. "Approaches to intelligent retrieval." Information Processing and Management, 23, pp.249-254. 1987.
- [5] H. M. Brooks, P. J. Daniels, and N. J. Belkin. "Research on Information Interaction and Intelligent Information Provision Mechanisms." J. of Information Science, 12, pp.37-44, 1986.
- [6] P. Ingwersen. "Information Retrieval Interaction." Taylor Graham Publishing, 1993. "情報検索研究 認知的アプローチ" 日本語訳 藤原鎮男監訳, トッパン, 1995.
- [7] 横田一正, 相場 亮: "マルチエージェントによる異種問題解決系の構想" マルチエージェントと協調計算 III, pp.67-74, 近代科学社, 1994.
- [8] M. R. Cutkosky, et al. "PACT: An Experiment in Integrating Concurrent Engineering Systems." IEEE Computer, pp.28-37, Jan. 1993.
- [9] S. M. Huang, P. Smith, and J. Tait. "An Integrated Expert Database System." '94 Int. Conf on Object Oriented Information System, pp.196-209, Springer-Verlag, 1995.