ファシリテータを利用した商用データベースの統合

吉野 利明[†]　寺本 良明[†]　川村 旭[†]　益岡 竜介[†]　吉田 武俊[†]

Commercial database integration using by a facilitator

Toshiaki YOSHINO[†], Yoshiaki TERAMORO[†] Akira KAWAMURA[†]
Ryusuke MASUOKA[†] and Taketoshi YOSHIDA[†]

あらまし　仲介エージェント技術の商用オンラインデータベースサービスへの適用および問題点とその考察について述べる。ユーザエージェント、仲介エージェント、データベースエージェントの 3 種のエージェントを利用し、エージェント間通信言語には KQML/KIF をベースとした ACL を採用している。仲介エージェントでは、広報情報に基づくデータベースの選択と、オントロジを利用した検索項目、条件名の変更などを行なっている。4 社から提供される 7 つの企業情報関連の商用データベースに適用したプロトタイプを完成した。負荷が高かったり、大量データを扱うデータベースシステムへの適用でも、データの有無チェック機能、分割回答機能を設けることにより、実用的レスポンスを確保している。現在、評価を進めながら運用システムを作成している。

キーワード　ソフトウェアエージェント、ファシリテータ、ACL、KQML/KIF、オントロジ

1. まえがき

インターネットをはじめとするネットワーク上に様々なデータベースが分散データベースとして構築されている。このネットワーク上での情報の共有・統合を効果的・効率的に実現できるような新しいデータベース技術が求められている。

この解決案の１つとして、ユーザと情報提供者の間にたって、情報の探し方や情報の所在に関する知識を持って仲介するエージェントすなわちファシリテータが考案、研究されている[1]。この種のファシリテータを中心としたエージェントの研究開発は、仮想カタログプロジェクト[2]をはじめとする電子商取引、既存資産の統合と再利用を目的とした分野[3]で応用研究が行なわれている。

これらのシステムの特徴はエージェント間通信言語 ACL(Agent Communication Language)を介してエージェントが作業を協調して実施する点にある。ACL は、一階述語論理をベースにした、異種プログラム間での知識の交換を行なう形式言語 KIF(Knowledge Interchange Format)[4]、内容部を KIF で記述するエージェント間知識操作言語 KQML (Knowledge Query and Manipulation Language) [5,6]、データベースで使用される用語を定義した辞書であるオントロジからなる。

本稿では、ファシリテータエージェントおよびそのメカニズムの商用オンラインデータベースへの適用について述べる。本システムの開発の目的は、1)エージェントベースのデータベース統合の実用性を検証すること、2) 検索に関する専門知識を持たないユーザの商用データベース検索をエージェントにより支援すること、3) 商用データベースの連携・統合のニーズに応えること、である。このために、本システムでは、7 種の商用データベースと、WWWデータベースを検索対象とした。 2 章では、商用データベースの現状と問題点について述べる。3 章では、本システムの概要、4 章では、エージェントシステムを商用データベースに適用する時の問題点と解決案を示す。5､6 章では、性能、課題を論じる。これらの問題点は、商用データベースにとどまらず、種々の既存データベースやアプリケーションを統合するシステムを開発するときに、頻繁に遭遇する問題である。

２．商用データベースの有用性と問題点

現在、日本では、3000 以上の商用データベースが利用可能である。それらは、新聞、人物、図書などの一般的情報から科学技術文献、特許などの自然科学技術、企業情報を中心とするビジネス情報など多様な分野にまたがる。

WWW などのオープン型データベースと比較すると、商用データベースの長所は

１）データの信頼性が高い、

２）特定の分野に関しては、時系列情報も保有し網羅的である、

[†] 株式会社富士通研究所、福岡市
Fujitsu Laboratories Ltd. 2-2-1 Momochi-hama, Sawara-ku,　Fukuoka-shi, 814 Japan

３)キーワードやシソーラスを利用して、論理演算、部分一致などきめ細かい検索が可能である、

４)ディレクトリサービスがあり、定期的に更新されている、

である。

上記のような特徴から、関連する情報を複数のデータベースから横断検索して、それらの情報を時系列データなどと組み合わせ整理することを目的とする場合、WWW 情報よりも適している。すなわち、企業の戦略を決定するために情報を収集する場合には、商用データベースのほうが優れているといえる。

欠点は

１)検索料金が高い、

２)コマンドが不統一、

３)検索が遅い、

である。さらに、WWW 情報よりも優れてはいるが、商用データベースの使い辛い点として

４)ディレクトリの提供はあるが、不慣れなユーザにとっては、その情報は十分であるとはいえない、

５)統一的なシソーラスがない、キーワードの不足、が指摘されている[7]。ここで述べた長所のうち、１と２はコンテンツに対する評価である。その他の長所と欠点は、データベースシステムまたは運用上の問題である。上述の問題に対処するためエージェント技術を導入する。

本システムでの検索対象は、利用頻度の統計[7]で国内製１位、海外製 2 位である企業情報・財務情報を中心にビジネス分野に関するものとした。本システムを利用することにより、エンドユーザは、指定した企業の情報に関して、商用データベースの中にどのような種類の情報があるのか、どの情報提供者がデータベースを提供しているかを意識せずに、どこかのデータベースで情報が提供されていれば、企業概要、財務情報、役員の個人情報、新聞記事などの情報が獲得可能となる。

データベースを統合する現在の２つの方式とエージェントを利用した統合を比較する。１つ目は企業間ＥＣをサポートするものである。例えば、富士通の WitWeb (Worldwide Inter-company Trade Web)である。これは、基本的には、データベースを物理的に 1 箇所に集め統合するという方式である。今後、データベースの地域的、規模的な拡大はこれまでにない速さで展開するであろうから、データを集中させるだけでは、対応が困難になることが予想される。

２つ目の方式は、データウェアハウス[8]を構築することである。意志決定支援や生産性向上のために、社内データに名前規約など一貫性を持たせて統合する方法である。分散データウェアハウス[9]という、一部のデータをローカルに持たせて統合し、処理を進めるという構築技法も提案されている。しかし、これは、データの後日の利用方法を検討して、初期データに加工を施すなどして、データを収集するため、初期目標になかったデータの獲得はできない。また、データが企業向けである場合が多く、個人での利用には利便性は少ない。加えて、上述の２方式とも、データ統合のために、データの移植などが必要であり、データベース数が増大し続けている現状を考慮すると、拡張性の点で十分であるとはいえない。

エージェントを用いた本システムは、拡張性を含む、ここで議論した問題点を克服することができる。次節では、システムの構成について述べる。

３．システム構成と動作概要

3.1 システムの概要

システム構成と動作の概要を述べる。試作したシステムでは、4 社から提供される７つの商用データベースとWWWデータベースをエージェントを使って連携する。全体構成を図１に示す。本システムでは３種類のエージェントを使用する。1)データベースプロバイダーまたはデータベースに割り当てられ、KQML メッセージから検索コマンドを生成、実行して、検索結果を KQML で返すデータベースエージェントと、2)ユーザに対応し、GUI と KQML メッセージ変換を行うユーザエージェント、3)情報交換の仲介をするファシリテータエージェントである。ファシリテータは、知識ベースとオントロジを持つ。前者には、ディレクトリ情報と変換知識がある。オントロジには、各データベースの企業を識別するために、企業 ID[1]、企業名、住所、業種が入っている。

[1] 1 社のデータベースから得たものであり、全データベースで共通に使えるものではない。オントロジのテストも兼ねてこの方式を採用した。現在共通企業ＩＤの作成中である。

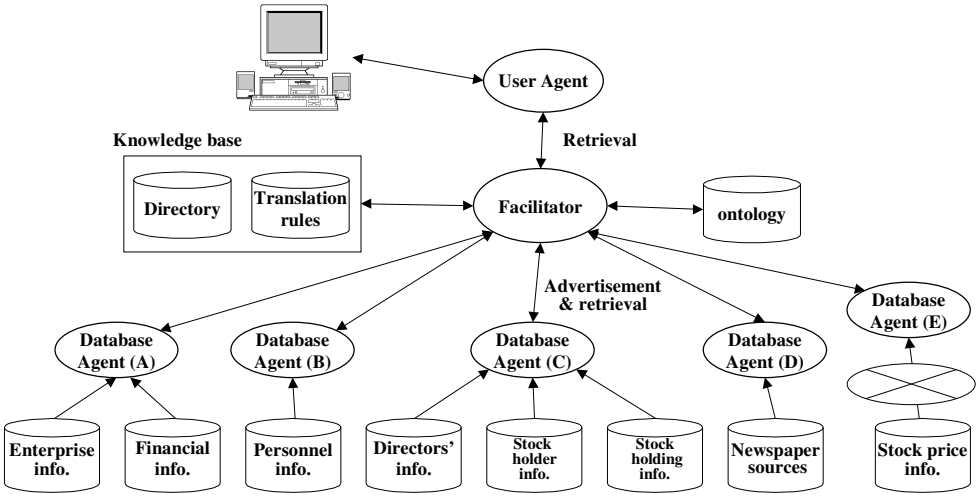


図 1 システム構成図
Fig. 1 System configuration

動作の概要を以下に述べる。データベースエージェントからファシテータに KQML performative advertise を使って広報する。このメッセージには各データベースエージェントが提供する情報の内容とアクセス方法が記述されている。ファシリテータは知識ベースにこれを格納する。

ユーザは企業名を入力する。検索結果は企業名、住所、業種、企業 ID の組である。この検索は図 1 のオントロジで行われる。この後の検索は、この企業 ID が有効なデータベースでは ID を利用して、利用できないデータベースでは、企業名をキーとして検索を行う。ユーザが企業名を選択し、企業を確認する画面例を図 2 に示す。企業の確認後、再度、ファシリテータへ検索メッセージを送る。検索要求の ACL メッセージはデータベースエージェントの広報情報にあわせて変形される。この変換は後述する Translation 機能を用いてファシリテータで行う。この時、検索要求を送信すべきデータベースエージェントを 3.3 で述べる方法にて決定する。データの獲得は通常 2 往復のメッセージの送受信で実現している[2]。それらは、１）該当データの有無を問い合わせる、２）１の結果をユーザに提示して、コンテンツの提示要求のあったものに対して、その獲得の実行メッセージを送信する、である。

データベースエージェントでは、商用データベースアクセスのための対話型コマンドを生成して、検索を行っている。検索結果は、順次ファシリテータ経由でユーザエージェントに送られ、ユーザに提示される。

図 2企業選 択画 面(左)と確認画 面
Fig. 2 Company name selection window(left) and confirmation window(right)

3.2 ファシリテータの構成

ファシリテータの構成と処理の概要を述べる。ファシリテータは企業間 EC 用として仮想カタログプロジェクト[2]で開発したものを利用している。モジュール構成を図 3 に示す。図 3 中の各モジュールの機能を以下に説明する。

KQML Message Receiver/Sender: April[11]通信ライブラリを介して、他のエージェントとの間で KQML

[2] 無駄な検索とデータ転送を行わないためである。

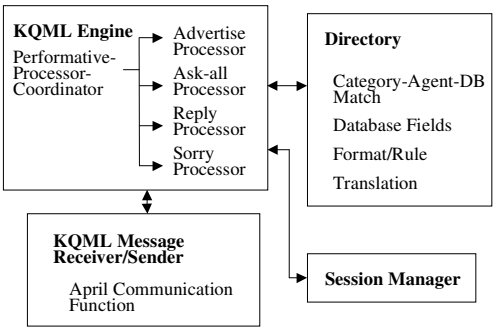


図 3ファシリテータの構成
Fig. 3 Configuration of the facilitator

メッセージの送受信を行なう。

KQML Engine: Performative-Processor Coordinator が、KQML 受信メッセージを取り出し、メッセージをパース後、該当する Advertise Processor などの各 Processor に引き渡す。各 Processor はメッセージを処理する。

Directory: 図１の知識ベースに該当し、Advertise の情報を保管する。各データベースエージェント名とその担当カテゴリ情報を保管する Category-Agent Match 部, データベースのスキーマ情報を保管する Database Fields 部および Format/Rule 部, エージェントが利用するオントロジ名とオントロジ間の変換パタンまたは変換を実行する関数を格納する Translation 部からなる。

Session Manager: 回答の統合、回答の制限時間内での有無を管理するために、送信メッセージを保管する。

3.3 ACL の例と KQML Engine

ここでは、ACL の例をあげながら、その内容と上記の各 Processor での処理を説明する。

1)Advertise Processor

ファシリテータは 2 種のアドバタイズの形式を受理する。データベースエージェントで直接または用語変換のみで取り扱うことのできる Query 形式で広報が行われる Format-Based Advertisement とデータベースエージェントが扱うことのできるスキーマ形式で広報が行われる Rule-Based Advertisement[10]である。ここでは、後者の例をあげる。この Advertise メッセージはデータベースコンテンツのカテゴリ情報、問合せに利用できる項目名の情報、オントロジの指定、変換情報を持つ。役員情報データベースエージェントが生成する Advertise メッセージを例１にあげる。

例１) advertise メッセージ例

```
(advertise :content
 ((database  役員情報 DB)
  (=> (member ?x 役員情報 DB)(isa ?x 企業概要))
  (field-definition  役員情報 DB :企業名 'is-text "企業名")
  (field-definition 役員情報 DB :役員リスト 'is-text "抄録")
  (field-definition 役員情報 DB :個人名 'is-text "名前")
  (allows-relational-db-query 役員情報 DB)
  (default-ontology DB-agent@fujitsu.co.jp
                    A 社 ontology.kif)
  (sentence-translation database.fujitsu.kif
    '(企業名 ?x "日本電気")
     A 社 ontology.kif '(企業名 ?x "ＮＥＣ"))
  (sentence-translation database.fujitsu.kif
    '(役員情報 ?z ?directors)
     A 社 ontology.kif
    '(役員リスト ?z ?list-of-directors)))
  :ontology database.fujitsu.kif
  :sender DB-agent@fujitsu.co.jp
  … )
```

この Advertise メッセージは、①役員情報 DB は企業概要の範疇であることを含意記号=>と述語 member と isa で、②問合せで利用できる項目名は、企業名、役員リスト、個人名であることを関数 field-definition で示している。Advertise Processor は、Category-Agent-DB Match 部に①を、Database Fields 部に②を格納する。また、(allows-relational-db-query)の宣言によりデータベースエージェントでは、関係データベース的な扱いができることを宣言している。この情報は Format/Rule 部に登録する。:ontology はこのメッセージが利用するオントロジ名を示す。:default-ontology にはデータベースエージェントが利用するオントロジ名を記述し、Translation 部に格納する。オントロジの指定の方法には、メッセージ中の:ontology の値(発信側のみ)、Translation に格納された値、ファシリテータのデフォールト値の 3 種があり、実行時には、この順に優先的に利用される。

Advertise メッセージ中にオントロジ間の変換情報を記述する。変換情報はパタンを直接メッセージ内に記述する方法と、変換パタンを別途準備して、それを参照して変換を実現する関数を準備する 2 種の方法をサポートする。前者は特異例またはＡＣＬの変形に利用する。後者は企業ＩＤなど大量データの変換時に利用する。これらの変換情報は Translation 部に登録される。

変換パタンは関数 sentence-translation で定義する。例１では、" A 社 ontology"でのパタン (企業名 ?x

“NEC”)は、“ database.fujitsu”では、 (企業名 ?x “日本電気”)に置換すべきことを意味する。役員情報の変換も同様である。2 つ目の方法は、 2 つのデータベースで対応する内容情報をオントロジデータベースとして準備し、それらの間を変換する関数を function-translation *template translation message* の形式で定義する方法である。*message* で *template* にマッチした部分を関数 *translation* で翻訳する。

2) Ask-all Processor

次に、Ask-all Processor の処理例をあげる。この処理では、Category-Agent-DB Match 部に登録されているすべてのエージェントについて以下 1-3 の処理を繰り返す。

1) Advertise で登録した Category-Agent-DB Match 部の範疇情報と問い合わせ ACL の範疇の情報を比較する。

2) ユーザエージェントとデータベースエージェントのオントロジが異なっていれば、前述の変換情報を参照して、送信先のエージェントが理解できる形に変形する。

3) 2 で変換された質問形式が、 1 で一致したデータベースエージェントから発信された Advertise メッセージで指定された形式と語句からのみなっているか、すなわち、問合せすることのできる項目名とコマンド形式からなっているかを Database Fields 部と Format/Rule 部の情報と照合することによりチェックする。もし、そうなら、データベースエージェントに依頼メッセージを送信する。さらに、返信処理のために、Session Manager にこのメッセージを登録する。

3 の依頼メッセージがどこにも送信できなかった場合、 Sorry Processor にこの旨を伝える。 Sorry Processor は Sorry メッセージを送信元に返信する。Ask-all メッセージの例を示す。ユーザエージェントが発信する役員情報の獲得のメッセージを例 2 に、前述の advertise メッセージに従い、ファシリテータが変換した後のメッセージを例 3 に示す。役員一覧を獲得表示した画面例を図 4 に示す。

例 2) 役員情報の獲得要求 ACL

```
(ask-all
   :language   KIF
   :sender      User-Agent1
   :receiver   Facilitator
   :reply-with    'message0001
   :aspect         (?directors)
   :content   (and (isa    ?x   企業情報)
                   (企業名   ?x   ’日本電気)
                   (役員情報   ?x ?directors)))
```

例 3) 役員リストを求める ACL

```
(ask-all
   :language   KIF
   :sender      Facilitator
   :receiver   DB-agent@fujitsu.co.jp
   :reply-with    'message0002
   :aspect         (?list-of-directors)
   :content   (and (isa    ?x   企業情報)
                   (企業名 ?x    'ＮＥＣ)
                   (役員リスト  ?x ?list-of-directors)))
```

3) Reply Processor と Session Manager

データベースからの回答である Reply メッセージは Reply Processor に渡され処理される。それらは、相当するセッションの回答として、Session Manager に登録される。Session Manager では、関係するすべての reply メッセージが返ってきたか、決められた制限時間[3]を過ぎていないかをチェックする。上記を満足する場合、それらの結果をマージしてユーザエージェントに返す。制限時間までに、 1 つも Reply がない場合には、Sorry Processor に回答の処理依頼をする[4]。

4．商用ﾃﾞｰﾀﾍﾞｰｽ適用での考慮点と ACL の拡張

商用データベースにファシリテータを含むエージェントシステムを適用する時に生じる問題点と我々が取った解決策を述べる。前節で述べたオントロジと 2 種の広報形式は、それぞれ、a)全体を統一するキーとなる ID が存在しない、 b)同種のデータベースであっても検索手順が異なる、に対する解決方法の 1 つである。上記以外の顕著な問題点 2 点について議論

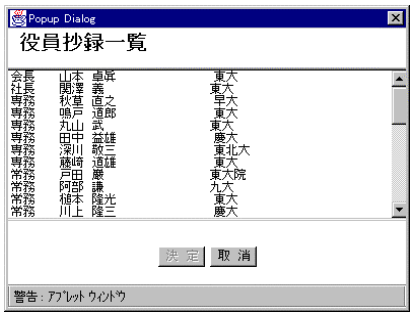


図 4役員一覧画面
Fig. 4 Director list window

[3] 1 分 30 秒に設定して評価中。

する。

１）回答データが大量である。

このケースは 2 種に分類される。1 つ目は回答すべきデータ数すなわち aspect の数が多い場合である。例えば、関連会社などの数百社のデータを同時に要求される場合である。このようなケースで全データを表示まで一度に処理しようとすると、回線速度や、データベース側での処理にかなりの時間[5]を要する場合がある。この場合、次の回答を要求する next performative を利用して 1 社ずつ回答を求めることも可能である。しかし、これでは、データベース側のセッションを長く保持したままになるか、または、セッションを 1 回ごとに確立することになる。前者は、セッションを一人が長く占有するため、多人数のアクセスを考えると不利である。後者は、セッションの確立を毎回行うため、レスポンスを悪くする原因となる。また、データベースの負荷を不用意に増大させることになる。このような状況を回避するため、aspect を数件単位でまとめて返すことができるよう next performative を拡張した next-list performative を定義した。なお、この performative では、データベースからの回答順を保証することが必要である[6]。

```
(next-list :number-of-items N   :order T
           :in-reply-to ... )
```

ここで、N は同時に返す aspect の数である。:order はデータベースからの回答順を保証する、ことを意味する。

2 つ目は aspect 一つの内容が大きい場合である。この場合、通常は商用データベース自身が、対応を取っていることがおおい。例えば、内容を解釈して適当に切る、または、画面サイズを考慮して行数で区切るである。論理的でないこの区切りを順次獲得する KQML を定義する。これは検索結果を連続して返す generator performative に :ascending-order keyword を追加する。データベースエージェントは、この keyword で指定された変数を内部状態として持つ作りとし、next performative に回答できるように拡張する。

```
(generator
  :aspect (?page)
  :ascending-order   (?counter)
  :content (and   (ask-all ...        ) ;;KQML が入る
                  (= (nth ?counter @page) ?page)))
```

２）回答が遅いデータベースが存在する。

データベースの負荷が大きい、または、そのためにアクセス数の制限などを行いデータベースの検索回答が遅くなる場合が起こる。このような状況で、検索システム全体のレスポンスの遅れを回避するため、ファシリテータでのデータの統合は最小限にする。統合しなくても、ユーザが解釈するのに支障がなければ、統合はしない。すなわち、即答する方を優先する。 言い替えれば、回答があったデータベースの結果を逐次ユーザエージェントに知らせることを原則とする。

さらにファシリテータを含む各エージェントに３．３節で述べた回答までの制限時間を設けている。制限時間以内に回答を受け取らない場合は、Sorry メッセージでの返答など、その対応処理を各エージェントが起動する。制限時間の設定とその対応を各エージェントに持たせることは、エージェントシステムに、簡単には停止しないという頑強性をもたらしている[7]。

５．開発言語と性能

このシステムの開発言語は、Java と Allegro Common Lisp を主に使っている。ファシリテータは Common Lisp で記述している。ユーザエージェントとデータベースエージェントの KQML メッセージと KIF 部の解釈／生成部は Java で記述している。エージェント間通信は Imperial 大と共同で開発した April[11]の通信ライブラリを Common Lisp, Java 版に拡張[12]したものを利用した。ユーザエージェントの一部を含むブラウザは Internet Explore 3.0 以降と Netscape Communicator 4.0 で動作している。ユーザエージェントの他の部分とファシリテータ、データベースエージェントは UNIX 上で動作する。Java アプレットで書いたクライアント上のユーザエージェントは、検索結果がWWWサーバ上のユーザエージェントに到着するのを監視している。

---

[4] 制限時間後の回答は無視される。

[5] 本システム運用時のレスポンス時間のユーザ要件は 30 秒である。

[6] 例えば、役員一覧を表示するとき、会長と社長の順が入れ替わってはいけない。

[7] 回答できない原因は、ネットワーク不調を含め様々であるが、待ち時間が予測できる、回答状況により別の作業を開始できるなど、ユーザには好評である。

性能[8]については、経過時間で通常 7 秒から 32 秒である。このうちデータベースエージェンのコマンドの実行および獲得データのデータベースシステムからデータベースエージェントまでの転送が 5 秒から 26 秒、その他の部分が 2 秒から 5 秒かかっている。ファシリテータの部分は、S7/300 上での測定で、往路が経過時間で 0.01 秒、復路がデータ量 22KB で 0.3 秒である。情報提供者では、9600bps の回線を利用しているところもあるが、提案のメカニズムはこれにも対応できていて実用に十分耐えうる性能である。

6. まとめと課題

商用オンラインデータベースへのファシリテータ技術の応用について述べた。このシステムを利用するとユーザはどのような種類のデータベースがどこにあるのかを知らなくてもテーマに関連した複数データベースの検索が可能となる。情報提供者は、データベースエージェントを作成することで自分の情報を宣伝しアクセス可能な状態にできるというメリットを持つ。

このシステムを採用することにより、2 節で述べた商用データベースの欠点のうち、２のコマンドの不統一、４のディレクトリの問題、５のシソーラスの問題は、広報メッセージを受理するファシリテータとオントロジの利用により解決できた。３の検索が遅いということに関しては、2 節で述べたデータの有無チェックを入れることによりデータベース側の負荷を最小とすることで解決を図った。１の料金の問題は、根本的には解決していないが、利用者増が安価な料金設定につながることを期待する。検討すべき課題として、1)ユーザエージェント側のクエリ ACL の作成規準、2)変換知識の獲得方法、3)回答データ量、回答時間が不明の場合の、データ量依存、回答時間依存の ACL の動的生成である。1)については、オブジェクト指向分析[13]で作成した分野モデルを規準とする方法が考えられる。2)については、広報情報から変換知識を作成する方法が考えられる。3)では、データベースエージェントが回答データ量、回答時間を推定し、ファシリテータが対応したＡＣＬを生成する方法が考えられる。

本方式では、ファシリテータが処理の要である。それだけ、ファシリテータに負荷を集中させてしまう。この負荷分散を検討する必要がある。その他の課題としては、システム構築支援の観点から、各エージェントや辞書を構築するための構築環境の提供があげられる。さらに、運用面からは、課金のためのログ情報の取得機能をユーザエージェントに持たせる必要がある。現在このシステムは、限定ユーザで利用可能であり、インタフェースなどの評価中である。この結果に基づき改良を加えて、商用システムとして利用する予定である。

謝辞：ACL メッセージの仕様を一緒に検討していただいた（株）富士通研究所ネットメディア研究センター丸山主任研究員をはじめとする SAGE グループ研究員の方々、データベースの検索および GUI について御意見、御協力をいただいた（株）富士通九州システムエンジニアリングの松嶋部長をはじめとする応用システム部の皆様に感謝します。

文献

[1]Michael R. Genesereth and Steven P. Ketchpel: Software Agents," Communications of the ACM, Vol.37, No.7, pp. 48-53, July 1994.

[2]丸山文宏, 益岡竜介, 菅坂玉美, 佐藤陽, 大竹聡, 渡辺光好: " SAGE(Smart Agent Environment) -仮想カタログ-," 第 54 回情報処理全大, 第３巻, pp.129-130, 1997.

[3]Donald F. Geddis, Michael R. Genesereth, Arthur M. Keller, and Narinder P. Singh, "Infomaster: A Virtual Information System, Intelligent Information Agents," Workshop at CIKM '95, December, 1995.

[4]Michael R. Genesereth and Richard E. Fikes: "Knowledge Interchange Format Version 3.0 Reference Manual," Logic Group Report Lgic-92-1, Stanford University, June, 1992.

[5]Tom Finin and Jay Weber: "Specification of the KQML Agent-Communication Language," June, 1993.

[6]Tom Finin and Jay Weber: "Specification of the KQML Agent-Communication Language," Feb., 1994.

[7]通産省機械情報産業局監修, データベース振興センター編："データベース白書 1997" , 1997.

[8]若井直樹："データベースの利用と期待される効果 ," DATABASE SYSTEM,Vol.1/No.1, pp.70-76, 12,1995.

[9]W.H. Inmon:"初めてのデータウェアハウス ,"藤本康秀 , 小畑喜一監訳 ,トムソン・パブリッシング, 1995.

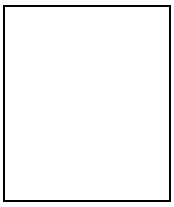
[8] 開発目標値として、エージェントシステムでのオーバヘッドを 4-5 秒としていたが、達成できている。

[10]Ryusuke Masuoka and Fumihiro Maruyama: “Ontology for Database Access,” AAAI Spring Symposium, 1997.

[11]牛嶋悟、高田裕志：“ April プログラミングガイド,” ソフトウェア科学会チュートリアル資料、ネットワーク指向プログラミング言語 第 6 章、pp.117-156,1996.

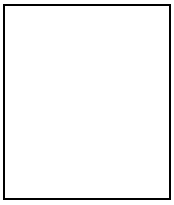
[12]寺本良明、吉野利明：“異種プログラミング言語間でのエージェント通信機能の実現,”情報処理学会第 55 回全国大会、Vol.2、pp.372-373、1997 .

[13]Sally Shlaer and Stephen J. Mellor: “オブジェクト指向システム分析” , 本位田真一 / 山口亮訳、近代科学社,1995.



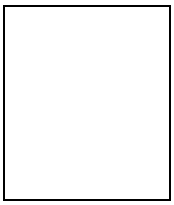
吉野 利明 （ 正員 ）

昭 55 九工大・工・情報卒。昭 57 九大大学院・総理工・情報システム学修士課程修了。同年 (株) 富士通研究所入社。自然言語処理、データベース、ソフトウェア工学, 人工知能関連の研究に従事。情報処理学会、IEEE-CS 各会員。



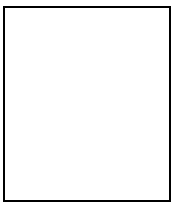
寺本 良明

昭 60 九大・工・電気卒。昭 62 九大大学院電気工学専攻修士課程修了。同年富士通(株)入社。平 4 (株) 富士通研究所移籍。現在エージェントの研究に従事。 情報処理学会会員。



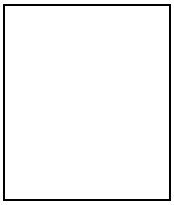
川村 旭

昭 60 名大・理・物理卒。昭 62 同大大学院工学研究科情報工学専攻修士課程修了。同年(株) 富士通研究所入社。 現在は、エージェント技術の研究開発に従事. 情報処理学会, 神経回路学会各会員。



益岡 竜介

昭 60 東大・理・数学科卒。昭 62 東大大学院・理・数学・修士修了。昭 63 (株) 富士通研究所入社。平 3 カーネギーメロン大学客員研究員。平 5 富士通研究所に戻り、ニューラル・ネットワーク、ソフトウェアエージェントなどの研究に従事。



吉田 武俊

平 3 東京農工大学大学院工学部数理情報工学研究科修了。同年富士通(株) 入社。基幹通信制御の研究開発を経て、現在、ソフトウェアエージェントの研究開発に従事。情報処理学会会員。