

## HI-AMSアーキテクチャ上の異種メディアの統合

木内 光, 矢野 聡司, 小西 修  
高知大学理学部数理情報科学科  
780-8520 高知県高知市曙町 2-5-1  
{hikiuchi, satoyano, konishi}@is.kochi-u.ac.jp

### 概要

近年の急速なインターネットの発展に伴い、ネットワーク上に異種情報源が分散して存在するという新たな環境が実現された。このようなオープンでダイナミックな情報空間を構成するネットワーク環境において、利用者が目的とする情報を適切に探し出すことは非常に困難である。こういった問題を解決するための一つの手段として、我々はエージェントモデルを用いた分散メディア環境における情報の統合を実現する、アクティブ・メディア・システム HI-AMS の研究を行っており、そのプロトタイプシステムを開発した。このシステムは KQML と CORBA の環境を融合した新しいシステムである。HI-AMS と情報源は位置透過的なアクセス、双方向通信といったことが可能である。そしてその応用例として、異種メディアの統合や、遠隔教育システムにも実際に適用して、その有効性を確認した。

キーワード：メディア・システム、協調エージェント、ラッパー、KQML-CORBA、ECA ルール、義務論理

## Integration of the different heterogeneous media on the HI-AMS architecture

Hikaru KIUCHI, Satoshi Yano, Osamu KONISHI  
Dept. of Information Science, Faculty of Science, Kochi University  
2-5-1 Akebono-cho Kochi 780-8520 Japan  
{hikiuchi, satoyano, konishi}@is.kochi-u.ac.jp

### Abstract

The new environment of distributed information sources suggests that we reconsider a lot of concepts in a field of database technology. In the network environment which constitutes such open and dynamic information space, it is very difficult for a user to start looking for target information pertinently. We were studying active media system HI-AMS which realizes integration of information in the distributed media environment using the agent model as one means for solving such problem, and developed the prototype system. This system is a new system which united environment of KQML and CORBA. HI-AMS and the information source have position penetration-access and possible both-directions communication. And as the example of application, it actually applied also to integration of different-species media, and the remote educational system, and the validity was checked.

Key words: media system, cooperative agents, wrapper, KQML-CORBA, ECA rules, deontic logic

## 1 はじめに

昨今の目覚ましいインターネットの発展に伴い、ネットワーク上に独立して構築された異種情報源が分散して存在するという新たな環境が実現している。この新たな環境において、様々な異種分散情報源への位置透過的アクセスと統一的な利用というものが求められるようになった。しかし、これらの異種分散情報源は半構造データなども含んでおり、データ定義が不明瞭であり、また異なるプラットフォーム上で異なるデータ形式で存在している。この状況下で利用者が目的の情報を適切に探し出すことは極めて難しい。こういった問題を解決する一つの方法として、我々はアクティブ・メディア・システム HI-AMS を研究し、そのプロトタイプシステムを開発した。HI-AMS は、ダイナミックかつオープンな環境に対応、スケラビリティ、エージェント、そして能動データベースといった考えに基づいて開発されている。そして、エージェント知識操作言語として広く知られている KQML と分散オブジェクト技術として異言語・異機種間でのインターオペラビリティを提供する CORBA の環境を融合することによって、HI-AMS はより高度なブローカー（質問に対して最適な情報源を紹介する）能力を有する。また、CORBA の提供するネーミングサービスによって HI-AMS と情報源は位置透過的アクセスを可能にし、遠隔教育システムにおけるモバイル参加といったようなことへも柔軟に対処することが可能である。本稿では、まず HI-AMS のアーキテクチャ、関連研究について簡単に述べた後、HI-AMS 内のエージェントの協調促進を行う方法について具体的に述べる。そして異種性を解決するためのラッパーアーキテクチャについて言及し、KQML-CORBA 環境に触れた後、HI-AMS を実際に適用した応用例である、(1) 異種メディアの統合、(2) 遠隔教育システムへの適用、について説明する。

## 2 HI-AMS 概観

我々の提唱するアクティブ・メディア・システム HI-AMS のアーキテクチャを図 1 に示す。

アクティブ・メディア・システム HI-AMS は、異種分散情報源を統合し、利用者の要求に対して最適な結果を返す自律的なシステムである。このメ

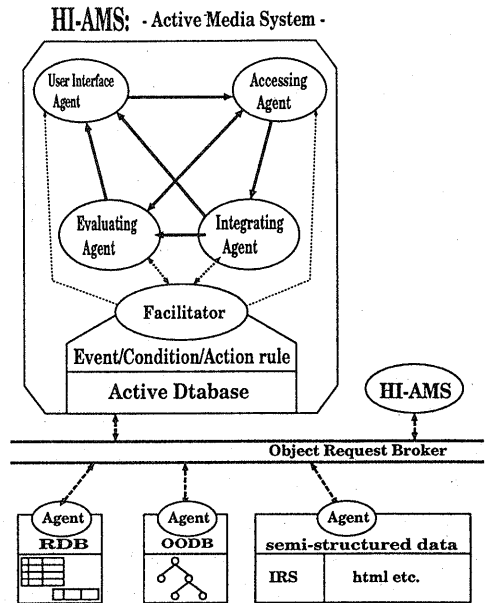


図 1: HI-AMS アーキテクチャ

ディア・システムの基本的な目標は、データベースやセマンティクス、あるいは種々のデータモデルやアクセス言語間の変換という既存の能力を越えて、情報の統合とアクセスを WWW 上のデータのダイナミックなアセンブリの領域に移動させることである。このように、HI-AMS が問題解決にあたる世界は、既存のスキーマによるデータベースや画像やテキストなどの半構造データも含んだ異種分散情報源であり、それらはオープンでダイナミックな情報空間を構成する。こういった環境に対応するため、HI-AMS にはアプリケーションの情報源からの独立や、最適な結果を返すといった知的処理を行う柔軟なアーキテクチャが必要となる。そこで、HI-AMS はアクティブ・データベースとそのフロントエンドに存在するファシリテータを核とし、アクティブ・データベースの ECA ルール機構を用いてエージェント群のコミュニケーションをファシリテータが協調促進する。HI-AMS 内のエージェント群はユーザ・インタフェース・エージェント（質問・検索法）とアクセス・エージェント（アクセス・集約）、及び情報統合・評価・エージェント（情報解析・学習・探索戦略）から構成される。各エージェントは、利用者からの要求に見合った情報を、ネットワーク

上に分散している異種情報源にアクセスし、収集・統合・評価するために、協調かつ自律して作業を進めていく。これによって、柔軟かで、かつ信頼性のあるメディア・システムが実現される。関連研究に、エージェントベースのメディエータの研究として、"InfoSleuth"[1]があるが、次の点で HI-AMS は異なっている。

- InfoSleuth は、情報源を Web 環境としているが、HI-AMS は、さらに既存のスキーマデータや ORB 環境にも対応している。
- エージェント群の協調促進を、エージェントの状態と行動推移を ECA ルールで表現した能動データベース上で実現している。
- データ評価機能を積極的に位置づけ、自己組織化マップによる自動クラスタリング機能を有している。

### 3 エージェント間の協調促進

#### 3.1 エージェント間の協調

HI-AMS を構成する各エージェントは、利用者からの質問に答えるために、互いに協調しながら作業を進める。このとき、各エージェントの調停を行い、協調を促進させる役割を担うファシリテータは、各エージェントの状態を管理しなければならない。そのため、各エージェントは処理依頼を受けたり、処理を実行・終了したりと、何らかの状態変化が起こる度に、ファシリテータに自分の状態を告知する。これにより、各エージェントはファシリテータを介して、協調相手の状態を知ることが可能となり、より豊かなエージェント間の協調が行える。具体的にエージェント群の振る舞いの流れを説明する。

- (1) 情報源エージェントはその固有の名前を CORBA ネーミングサービスに登録する。
- (2) ユーザ・インタフェース・エージェントからの質問が、アクセス・エージェントに送られる。
- (3) アクセス・エージェントは、その質問をファシリテータに送る。

(4) ファシリテータは、ネーミングサービスに登録されている情報源の中からマッチングをとり、最適の情報源エージェントにアクセス・エージェントに知らせる。

(5) アクセス・エージェントは、その情報源エージェントに質問を発し、結果を受け取る。

ここに、ファシリテータは、各エージェントの行動（生成、起動、休止、消滅など）やエージェント間のメッセージ転送を把握して、それらの活動の促進を図る。

#### 3.2 義務論理

エージェント間の通信は、要求とその要求の採否と見ることができる。我々は各エージェントの状態に着目し、ここに、許可、禁止、義務の判断を入れることによってファシリテータは各エージェントの行動を管理し、協調促進が行える。この3つの状態をモデル化したものが、様相論理の一つである義務論理 [3] であり、それぞれ「しても良い（許可）」、「しなければならない（義務）」、「してはいけない（禁止）」ということの意味する。この義務論理を適用すると、エージェントの初期状態は「許可」であり、処理依頼のメッセージを受け取ると「義務」状態に移行し、処理の主体も移される。処理が終了すると、依頼主に処理結果等のメッセージを返して「許可」状態に移行し、初期状態に戻る。もし、エージェントが処理依頼のメッセージを受け、「義務」状態に移行したが依頼された処理に失敗した場合や、一貫性制約を破る可能性が存在する場合は「禁止」状態にして、それ以後の処理依頼の受付や行動を行わないようにする。問題解決後、ファシリテータが「禁止」状態から「許可」状態、即ち初期状態に戻す。これらは、エージェントの要求の間の関係を明示する。例えば、ある要求の履行は、他の履行を義務づける。あるいは、他の履行を許可または、禁止する。このような要求間の義務論理関係の表現は、要求に対する理由の階層を表現し、要求間の優先度の基本的な表現を与える。これは、HI-AMS の目的とそれを実現する機能において、ある要求を含むか除外するかに対する理由を表現することになる。

### 3.3 メッセージトランザクション

HI-AMS のシステム上のエージェント群の振る舞いは、次の2点で制約される。

- (1) エージェントは、自己の外へのアクションについては、必ずファシリテータに問い合わせねばならない。ファシリテータは、それに必ず答えなければならない。また、ファシリテータは問い合わせという事実がないのに、応答することは禁止される。
- (2) エージェントは、ファシリテータから紹介された他のエージェントへのみ、要求をすることが許可される。

各エージェント間、またはエージェントとファシリテータのメッセージのやりとりは、相手のあるメッセージトランザクションと捉えることが出来る。例えば、ユーザ・インタフェース・エージェントが送った処理依頼メッセージを受け取ったアクセス・エージェントは、ファシリテータに依頼された処理を行うために必要な情報源のドメイン情報などを質問する。

### 3.4 協調促進とECA表現

各エージェントの協調を促進するファシリテータは、データベースのフロントエンドに存在し、各エージェントの状態を更新することによって管理する。この時、データベースには、エージェントの状態を記述する以下のようなテーブルが必要である。

`agent [name,type,state]`

`name` はエージェントの名前、`type` はエージェントの種類、`state` はエージェントの状態を表現している。例えば、ユーザ・インタフェース・エージェントがアクセス・エージェントへメッセージを送りたいという通知をファシリテータが受け取ると、`type` がアクセス・エージェントであり、`state` が「許可」であるエージェントを検索し、その名前をユーザ・インタフェース・エージェントに教える。

また、エージェントの動向を逐一記述するための、いわゆるログ・テーブルも必要である。

`agent_log [id,name,operation,  
sender,receiver,ope_state]`

このテーブルはエージェント `name` において、処理 `operation` をエージェント `sender` から依頼された、またはエージェント `receiver` へ依頼して、その処理の状態が `ope_state` であることを表現している。`id` はユーザが要求を発してからその解答を得るまでの一連のタスクを識別するための番号である。以下に HI-AMS を実際に利用した場合のログ・テーブルの例を示す。

id	name	operation	sender	receiver	ope_state
1	UIA	askQuery	user	UIA	finish
1	UIA	ask	UIA	AA	finish
1	AA	ask	UIA	AA	finish
1	AA	getData	AA	SA	finish
1	AA	askSource	AA	F	finish
1	AA	tellResult	AA	IA	finish
1	IA	tellResult	AA	IA	finish
1	IA	tellIntinfo	IA	EA	finish
1	EA	tellIntinfo	IA	EA	finish
1	EA	tellResult	EA	UIA	finish
1	UIA	tellResult	EA	UIA	finish
1	UIA	tellResult	UIA	user	commit

表 1: 各エージェントのログ

これらエージェントの状態を管理する二つのテーブルとアクティブ・データベースにおけるECAルール機構を用いることにより、義務論理における一貫性制約が表現出来る。例えば、次のような一連のエージェント間のやりとりについて考えてみる。アクセス・エージェントはファシリテータへ、「ユーザ・インタフェース・エージェントから質問を受けた」というメッセージを伝える。メッセージを受け取ったファシリテータはエージェントの状態を更新する。このとき、次のようなルールが定義できる。

```
E: UPDATE TO AccessAgent
WHERE current.agent = 'aa' AND
new.operation = 'ask' AND
new.from = 'uia'
C: if 'finish' != SELECT ope.state
FROM UserAgent
WHERE agent = 'uia' AND
from = 'user'
operation = 'ask'
A: UPDATE TO AccessAgent SET state = '禁止'
WHERE from = 'uia'
```

つまり、アクセス・エージェントがファシリテータへそのようなメッセージを送ることが出来るのは、それ以前にユーザ・インタフェース・エージェントがユーザから質問を受けているという前提があるという制約を表現している。この制約が破られた場合は、アクセス・エージェントの状態を義務論理における「禁止」状態にすることによって、その後の行動を禁止しなければならない。この「禁止」状態の時に行動を起こしてはいけない」という制約もECAルールで表現することができる。データベースのテーブルの更新操作はファシリテータが全て担っているため、本研究ではECAルールをファシリテータの一つの機能として組み込んだ。ECA機構を備えたファシリテータとデータベースが連携することによってアクティブ・データベースと同じ機構を実現している。

## 4 ラッパーアーキテクチャ

### 4.1 ラッパー

情報源から情報を収集するためには、その情報源固有の検索式(例えば、リレーショナルデータベースにおけるSQLのようなもの)を用いる必要がある。更に、このような検索式を用いるには、その情報源のスキーマやデータ構造を知らなければならない。従って、情報源の利用者はこのような検索式による検索方法を習得しなければならない。しかし、扱う情報源が増えるに従い、習得しなければならない検索方法も必然的に増えるため、利用者の負担は非常に大きくなる。そこで、利用者から与えられた一般的な問い合わせなどを個々の情報源が実行できる検索式へ変換するラッパーを用いることで、こういった問題を解決することができる。ラッパーによって、利用者は各情報源のスキーマやデータ構造を知る必要もなければ、各情報源の検索方法を習得する必要もない。利用者は一般的な問い合わせをラッパーに行うだけで、ラッパーが問い合わせの内容を解釈して、各情報源固有の検索方法によって質問を行う。利用者は容易に複数の情報源から結果を得ることができる。ここで、我々が研究開発をしているHI-AMSを利用者側と考えると、情報源への問い合わせは、正にラッパーへの問い合わせと考えることができ、異種情報源における検索式を熟

知する必要もなく、各々のラッパーに一般的な問い合わせを行うだけで良い。こういったラッパーを各情報源毎に用意することによって異種性を解決することができる。さらに、ラッパーとHI-AMSにおけるアクセス・エージェントとのやり取りは、エージェント同士のコミュニケーションとして捉えることもでき、本研究ではラッパー=エージェントとして考えている。

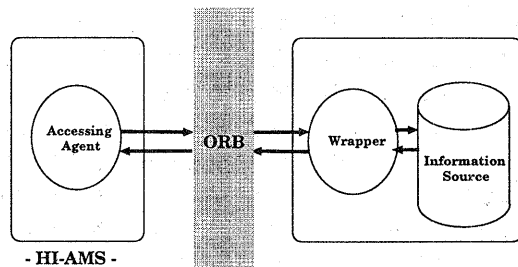


図 2: ラッパー

### 4.2 JavaIDL

Java IDLはJavaプラットフォーム上でCORBA環境を構築するためのもので、Java2より標準パッケージとして組み込まれている。CORBA(Common Object Request Broker Architecture) [4]とは、OMG(Object Management Group)と呼ばれる標準化団体が提唱するORB(Object Request Broker)の仕様のことで、ORBとは、ネットワーク間でオブジェクト通信を行う機構のことを指す。リモートオブジェクトの呼び出しを行うためには、クライアント、サーバ両サイドでORBを使用しなければならない。CORBAはIDL(Interface Definition Language)というインタフェース定義言語を用いることで、プログラミング言語に非依存な分散オブジェクト環境を提供する。CORBA以外の分散オブジェクト技術にはHORB(Hirano ORB)やRMI(Remote Method Invocation)があるが、これらはJava言語専用の分散オブジェクト環境である。しかし、CORBAはIDLによってJava環境だけではなく、異言語・異機種間でのインターオペラビリティを提供することができる。HI-AMSにおける情報源、すなわちラッパーをこのIDLによって定義された共通インタフェースに基づいて生成する

ことによって、前述したような異種情報源への一般的なアクセスというのが実現できる。HI-AMSは実装方法、つまりスキーマやデータ構造、ラッパーが情報源へどのように検索するかといった方法について知る必要はない。情報源もインタフェース定義を変更しない限り、HI-AMSとは独立して、情報源のスキーマやデータ構造の変更、それと連動したラッパーの実装部分の変更を行うことができる。なお、ここで言うスキーマやデータ構造の変更は、扱う情報の種類の変更(例えば文献情報を扱っていた情報源が、会社情報を扱う情報源に変更するといった)を含んでおらず、同種情報におけるスキーマやデータ構造の変更のことを指す。さらに、JavaIDLというJavaベースのラッパーを生成することによって、異なるプラットフォーム上でもスキーマやデータ構造が同じ情報源ならば同一のラッパーを利用することができ、より汎用性が高いものになる。HI-AMSで実際に利用している、マルチメディアデータを扱う情報源におけるラッパーの共通インタフェース定義を以下に示す。

```

module HIAMS{
  typedef sequence<octet> ByteArray;

  struct MultiMediaData{
    string      filename;
    sequence<ByteArray> buf;
  };
  interface SourceAgent{
    boolean init();
        //情報源の初期化
    boolean term();
        //情報源の終期化
    long select(in string column,
               in string keyword);
        //検索結果の数を返す
    MultiMediaData getData(in long select_no);
        //検索結果を返す
  };
};

```

### 4.3 位置透過性

CORBAには、オブジェクトに抽象的な名前を付与し、その名前によってオブジェクトリファレンスの登録/取得を行うためのCORBAネーミングサービスというオブジェクトの登録/取得機構がある。オブジェクトの管理や検索を行うために提供さ

れる最も基本的な機能で、CORBAネーミングサービスを利用することによって分散オブジェクトへの位置透過的アクセスが実現する。CORBAネーミングサービスを利用したServerオブジェクトの検索には、オブジェクトの名前とネーミングサービスが起動しているホスト名が必要である。クライアントは、ネーミングサービスの存在位置のみ知っていればよく、オブジェクトの配置を変更してもクライアントには一切影響が及ばない。HI-AMSがこのネーミングサービスを利用することによって、情報源の位置が変更されてもネーミングサービスに登録された情報源の名前のみでHI-AMSはその情報源へアクセスすることが可能である。特にダイナミックに刻々と変化するようなWWW環境においては、こういった位置透過的アクセスというものが非常に有効となる。

## 5 KQML-CORBA環境

KQML(Knowledge Query and Manipulation Language)[5]は、知識システム間で様々な知識を交換することを目的として設計された発話行為論に基づいたエージェント間知識操作言語である。複数のエージェント同士が協調して、問題解決を行うマルチエージェントシステムにおけるエージェント同士が通信を行うための言語として有名である。CORBAは前述のようなネーミング・サービスの登録機構によって分散環境における位置透過性を実現する上に、他の分散オブジェクト技術との大きな違いが異言語によるオブジェクト間の協調動作が可能となる分散オブジェクト環境を提供する。本研究では内部のエージェントの通信にはKQMLを用い、外部の情報源との通信にはORBインタフェースを介してメディア・システムと通信する。つまり、KQMLとORBを融合した環境を作っている。これによって、KQMLの通知機能とCORBAネーミングサービス機能を融合した、より高度なブローカー(質問に対して最適な情報源を紹介する)能力を有する。

さらに、KQMLは米国スタンフォード大学機械工学科で開発中のJava Agent Template(JAT)を利用し、CORBAにはJava2から標準パッケージと

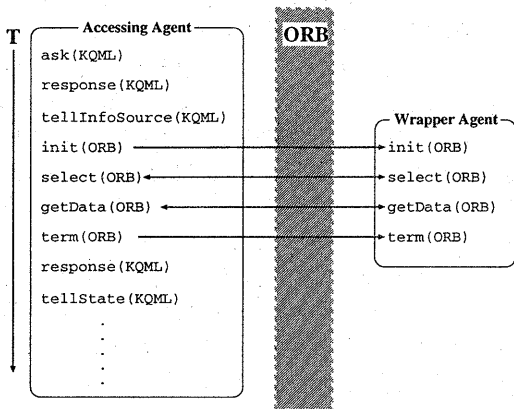


図 3: KQML-CORBA 環境におけるアクセスエージェントのメソッドの流れ

なった JavaIDL を用いているため、プラットフォームに非依存な分散オブジェクト環境を提供する。これらによって、我々が研究・開発している HI-AMS が対象とするような、オープンでダイナミックな分散環境における異種情報源の情報獲得・統合に有効であり、異種分散情報システム間のプラットフォーム・レベルでのデータの異種性を容易に解決することができる。

## 6 応用例

本研究では、HI-AMS の応用として (1) 異種メディアの情報統合と (2) 遠隔教育システムへの適用を行った。以下この 2 つ例について述べる。

### 6.1 (1) 異種メディアの情報統合

マルチメディアを対象にしているという観点から著作権等の問題も考え、今回はクラシック音楽にデータを絞った。例としては、利用者が「チャイコフスキーに関する情報が欲しい」といった非常に単純な要求を HI-AMS が受け取るとその要求を満たすような情報源から文字データ、画像データ、音楽データを獲得してそれらの情報を統合し、その結果を利用者に返すというものを考えた。利用者はブラウザを介して HI-AMS とやりとりするため、結果として表示されるデータは HTML である。異種分

散環境における統合例を示すために、文字データ、画像データ、音楽データはあらかじめそれぞれ別々の情報源へ格納しておいた。文字データは HTML フォーマットとして通常のディレクトリ階層に保存しておき、画像データは JPEG フォーマット、音楽データは mp3 フォーマットとして、フリーのリレーショナルデータベースとして広く普及している PostgreSQL にラージオブジェクトとして格納した。図 4 がその実験例である。

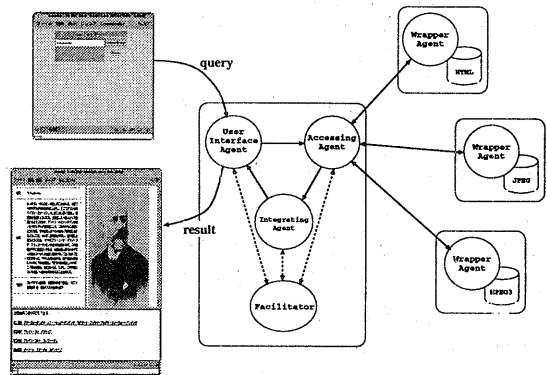


図 4: 実験例

### 6.2 (2) 遠隔教育システムへの適用

遠隔教育システムとは、様々なネットワーク技術を用いてオンラインで教育を受けることができるシステムである。通信教育とは異なり、双方向の通信を規定している。従来、衛星放送やテレビ会議システムなどが遠隔教育システムとして多く用いられてきた、しかし、今日のインターネット技術の発展や、高性能なパーソナルコンピュータの低価格化による一般普及といったことから、教師と学生とがリアルタイムに講義を受けられる環境が整いつつあり、遠隔教育システムのインターネット上への適用が盛んに研究されている。位置透過的アクセスを可能とした HI-AMS はこの遠隔教育システムへの適用も可能である。今回の適用では HI-AMS 側を教師、情報源側を生徒と置き換えて遠隔教育における宿題システムについて考えた。その一連の流れについて簡略に述べる。(図 5)

生徒は教師側における宿題掲示板を WWW で確認することによりその宿題を解き、それを HTML

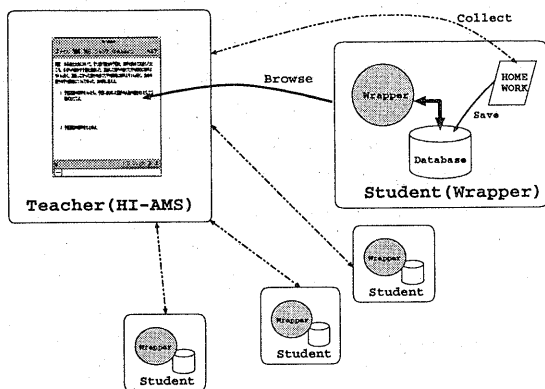


図 5: 宿題システム

あるいは通常のテキストフォーマットなどで保存する。保存先はデータベースでもコンピュータのHD上でもかまわない。ラッパーがデータスキーマ、フォーマットなどの異種性を解決してくれるため、生徒はそれぞれ独自のフォーマット、保存先を選択することができる。ラッパー(情報源)エージェントには生徒の学籍番号を固有の名前として持ち、生徒がラッパーエージェントを立ち上げるとCORBAネーミングサービスによってネームサーバにその固有の名前が登録されているため、教師側のHI-AMSはそのラッパーエージェントの名前さえ知っていれば、生徒がどのような位置からアクセスをしてもCORBAネーミングサービスの位置透過性によって、生徒側へアクセスすることができる。従って、教師は教師側の都合(宿題提出期限など)によって宿題を確実に回収することができる。

## 7 おわりに

本稿では、アクティブ・メディア・システムHI-AMS内のエージェント間における協調促進方法、そこへ適用した義務論理、ECAルール機構、そして異種情報源間に存在するデータの異種性を解決するためのラッパーアーキテクチャについて説明した。また、HI-AMSシステム内のエージェント間通信にはKQMLを用い、情報源とのやりとりにはCORBAを適用することでKQML-CORBAという、より高度なブローカー能力を持ち、位置透過的アクセスをも可能とした環境を実装することが

できた。このHI-AMSの適用例として、分散環境上に存在する異種メディアの統合と遠隔教育システムにおける宿題システムの実験を行い、その有効性を示した。しかし、今回の実装では、エージェント処理系は比較的単純であり、利用者からの複雑な要求に対しては答えることは出来ない。今後はさらにシステム内のエージェント処理系や、情報源とHI-AMSとのやりとりにおけるトランザクションについて考察し、より実用性の高いシステムを目指す。また、本システムの遠隔教育システムへの適用において、講義データベース、成績評価システムなどをも考慮に入れた拡張も検討している。

## 参考文献

- [1] Bayardo Jr. R.J. and et al., "InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments," SIGMOD'97, pp.195-206, May 1997.
- [2] O. Konishi: "Design and Implementation of active Mediation System -A Study on Agent Based Databases-, pp.149-152, In Advanced Database Ssystems for Integration of Media and User Environments'98, World Scientific, 1998.
- [3] H.Weigand , et.al., "Interoperable Transactions in Business Models - A Structured Approach", Advanced Information Systems Engineering, LNCS 1080, pp. 193-209, 1996.
- [4] Object Management Group, "Common Object Services Specification Volume I.", OMG Document No.94-1-1
- [5] T. Finin, J. Weber, G. Wiederhold, M. Genereth, R. Fritzson, D. McKay, J. McGuire, P. Pelavin, S. Shapiro, and C. Beck. Specification of the KQML agent-communication language. Technical Report EIT TR 92-04, Enterprise Integration Technologies, 1992. (Updated July 1993).
- [6] Pitoura, E. and Bhargava, B., "A Framework for Providing Consistent and Recoverable Agent-Based Access to Heterogeneous Mobile Databases", SIGMOD Records, Vol.24, No.3, pp.44-49, Sept.1995.
- [7] 中田 優作, "分散オブジェクト技術を用いた情報統合に関する研究", 高知大学大学院理学研究科情報科学専攻 1998 年度修士論文.
- [8] 小西 修, 木内 光: 分散協調処理におけるエージェントベース・トランザクション, 電子情報通信学会信学技報, Vol.98, No.486, DE98-36, pp.49-56, 1998 年 12 月