

分散したデータベースと知識ベースの統合モデル

Qiming Chen

カリフォルニア大学

ロスアンゼルス分校

上林 弥彦

京都大学工学部

本稿では各種のデータベースと知識ベースを統合するための新しいモデルの提案を行なう。種々のデータは階層構造の集合で表現され、それぞれは複合オブジェクト等を表現する。システムの効率向上のための知識もデータとは異なる分類が行なわれそれ自体も階層構造で整理できる。これらの2つの階層構造を結び付けるため、オブジェクト指向的な階層を構成するコンテキストモジュールを用意する。複数のコンテキストモジュールは単一の階層だけでなく複数の階層を実現する。コンテキストモジュールとデータや知識はリンクで結ばれておりこのリンクは動的に変化し得る。この構造はオブジェクト指向において、クラス階層とクラス内データの対応を動的にしたこと、処理コストの高い参照関連を減らし得ること、クラスとメソッドとの対応も動的にしたこと、複数の汎化階層を扱えること等の一般化を行なったものとみなすこともでき、オブジェクトのクラス間移動やビューといったオブジェクト指向で扱うのが困難な問題も扱い易くなる特色がある。

A New Model to Unify Distributed Data and Knowledge Base Systems

Qiming Chen

University of California

Yahiko Kambayashi

Kyoto University

In this paper we will develop a new model called a hyper-coupling data-knowledge coordination model, suitable for integrating databases (DBs) and knowledge-bases (KBs) under distributed environment. Instead of permitting arbitrary links among data and knowledge, the model permits three kinds of hierarchies; token object hierarchy for data integration, knowledge module hierarchy for subject knowledge integration, and context module hierarchy for data/knowledge integration. Data can be shared by multiple subjects with minimal duplication and migration; communicating context modules mediate multiple DBs and KBs. A hyper coupling data-knowledge coordination model can therefore be viewed as a web linking context modules with the underlying DBs and KBs dynamically. Through such dynamic linking, data and knowledge can be properly tailored and coupled to fit different cooperating contexts. Moreover, the cooperations of data and knowledge are specified abstractly and handled separately from the independent data and knowledge management.

1. まえがき

データベースおよび知識ベースの統合システムは、独立かつ不均一なデータベースや知識ベースから、利用に応じたデータを繋ぎ合わせて使うことになる。この場合に巨大な統合システムを作るのは処理効率や維持管理の面から適切ではなく、個々の独立性を持たせたまま動的に統合してゆくのが望ましい。本稿では、この問題に対処するためデータ・知識の超結合モデルを提案し、その応用について検討する。データと知識の協調を実現するためには次のような方式が与えられる。

- 1) 1つのデータ・知識ベースシステムとしてまとめる。
- 2) 1つの知識ベースと複数の（分散）データベースシステムによって構成する。
- 3) 各問題に対してデータベースと知識ベースを統合したシステムを用意し、これらが協調して動くように全体のシステムを構成する。
- 4) 知識は主題別に分類し、データはクラス分けし、それらの間を動的に統合する。

上記3)はこのようなシステムを作るための一般的な方法として知られており、分散知識・データベースによる協調処理のモデルとなっている（図1（a））。この方法では、データと知識の間の対応が固定しており次のような問題がある。

- a. データをまとめる方法は種々あるが、1つのデータのまとめ方だけが適用されるため、特定の利用に対してのみ効率が良くなる。
- b. 突然必要となる利用形態で、あらかじめ予測できないものに対処しなければならない。例えば、地震の起こったときの処理等で、すべての可能性を考えて設計するのは不可能である。
- c. 対応が固定しているため、データのクラス間移動やデータの重複、参照関連の複雑化といった問題が生じる。

この問題に対応する1つの方法は、自由度の高いビューを固定した構造から作ることである。しかし、構造が固定しているためビューの処理効率が高いとは言えない。

データベースや知識ベースでは、データや知識の寿命を永くするために、利用状況に柔軟に対応できることが重要である。4)の1つの実現法は図1（b）に示すようにデータ間、知識間およびデータ・知識間に自由にリンクを作れるようなモデルである。しかしこのようなモデルは自由度が高すぎて、スキーマとしてあまり有効とは言えない。データ間の結合はデータ階層の形で扱い、知識間の結合は知識階層の形で扱う。データは同時に利用されるデータどうしがまとめられる傾向にあるのに対し、知識は同じ主題のものがまとめられる傾向であるため、この2つの階層は一般に一致しない。データと知識間の対応を完全に自由にせずここに1つの階層を設けることにする。これをコンテキストモジュールの階層と呼ぶ。コンテキストモジュールは単なる容器で対応するデータおよび知識はデータベースおよび知識ベースから供給される。この対応はリンクを中心に実現される。全体的なモデルを図2に示す。

コンテキストモジュールの階層では次のような意味でオブジェクト指向データモデルの一般化に対応する。

- 1) コンテキストモジュールに対応するデータはクラス内のオブジェクトに対応する。
- 2) コンテキストモジュールに対応する知識の一部はメソッドや意味制約等に対応する。
- 3) 1つのコンテキストモジュール内のデータの型は必ずしも全て同じでなくてもよい。コンテキストモジュールの階層は知識の継承等のために用いられる。
- 4) コンテキストモジュール内には実データや知識がないため、それぞれがオブジェクト指向ビューに対応している。このため、複数の階層を定義してもデータや知識の重複といった問題はない。

データと知識は完全に分かれるわけではない。例えば、知識と例外データによってデータベースを構成している場合や、特定のデータ集合固有の意味制約は他のデータベースで利用されるわけではないので、これらはデータベース側に蓄えられる。逆に特定の主題の知識と結び付いたデータは知識ベースとして蓄えられることになる。

例1：例えば衛星からの画像は目的によってか

なり異なる扱いがなされる。地図を作るためならベクトル集合として扱うことができる。3次元の地図であるDTMでは3次元ビューを扱う必要がある。このように目的によって画像は他のオブジェクトと異なる結合のしかたを必要とする。また異なるデータベースでは利用できる画像処理パッケージの違いによって、異なるデータ構造が利用される。地理情報システムと資源探査システムでは同じデータに対しても適応される処理プログラムが異なる。処理プログラムも知識とみると、データと知識の動的な結合が必要な例である。一般に利用法によって、そこで使われる処理プログラムや知識表現と適合するようにデータが変換されることになるが、このようなデータ構造を応用ごとに設定するのは大変である。また同じ名前の処理（例えば「人工的着色」）が利用法によって、異なる意味や実現法となっているということも起こり得る。

このような検討から、コンテキストモジュールの階層をデータと知識の統合のために入れた、データ・知識超結合協調モデル(hyper coupling data-knowledge coordination model)を新たに提案する。コンテキストモジュール階層導入によって次のようなことが可能となる。

1) 多重移入機構

コンテキストモジュールは独立に存在するデータベースや知識ベースの一部ずつを1つにまとめて応用向きのモジュール階層を構成できる。

2) データと知識の統合

知識およびデータの表現は統一されていない。したがってコンテキストモジュールでは、知識表現の統一とデータ表現の統一とを行なう。データベースから種々の形で表現されたデータ、スキーマ情報、データに対する意味制約が入ってくる。これらを1つの形に変換する変換モジュールや知識表現をその応用向きに統一するためのモジュールが必要である。

3) 協調処理の定義

知識とデータが協調できるための、高水準の規則やプロトコルを定義する能力が必要である。

4) オブジェクト間の通信

1つの利用目的のために作られたコンテキストモジュール階層と別の目的のために作られたコンテキストモジュール階層とが互いに通信して協調的に処理することも必要である。1つのコンテキストモジュール階層を1つのデータ・知識ベースシステムとしてみなすと、図1(b)に示すようなシステム構造となり、それらが協調して問題解決にあたる形となる。

5) コンテキストモジュールの階層

コンテキストモジュールは利用目的に応じた階層を構成する。知識の継承を可能とする汎化階層のほかに種々の階層を実現できる。1つのコンテキストモジュール集合の中に複数の階層が存在し得る。

複数の階層が実現されると次のような利点がある。

A. 汎化階層や部品階層のような代表的な階層を複数個利用することにより、メソッド定義やデータ間の意味制約を単純化できる。

B. セキュリティ階層のようにデータのもつ意味制約とは異なる階層も定義できる。特に利用者グループによって異なるセキュリティ階層を実現できるようなモデルとしても有用である。

C. オブジェクト指向データベースにおけるビューの問題は、1つのコンテキストモジュール階層を定義することによって実現できる。従来の方法では、システムがもつ階層が主でビュー階層は従となっており、主階層のデータを消去すると従階層も消えてしまうことになる。本モデルでは全てが対等であるため、必要なくなった階層は独立に自由に消去することができる。この性質は、スキーマ変換をしなくても利用形態の変化に追従できる利点がある。

オブジェクト指向データベースにおけるオブジェクトの参照関連は本モデルでは対応する原データへのリンク等によって実現される。この方式のオブジェクト指向モデルに対する利点は以下のとおりである。

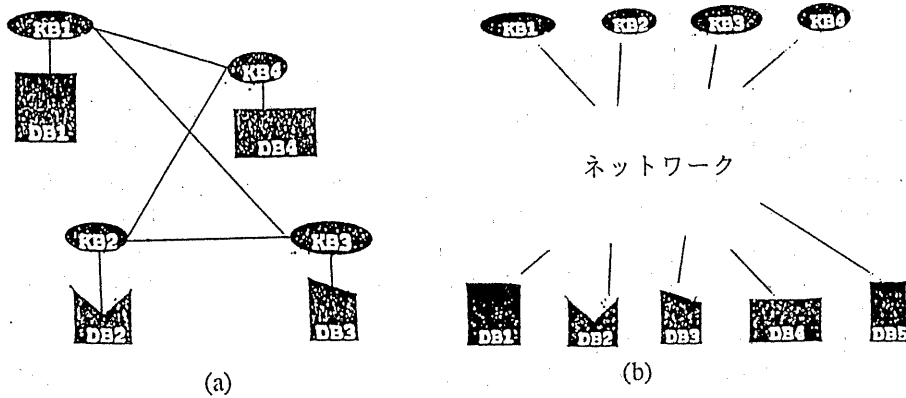


図1 データと知識ベースの協調モデル

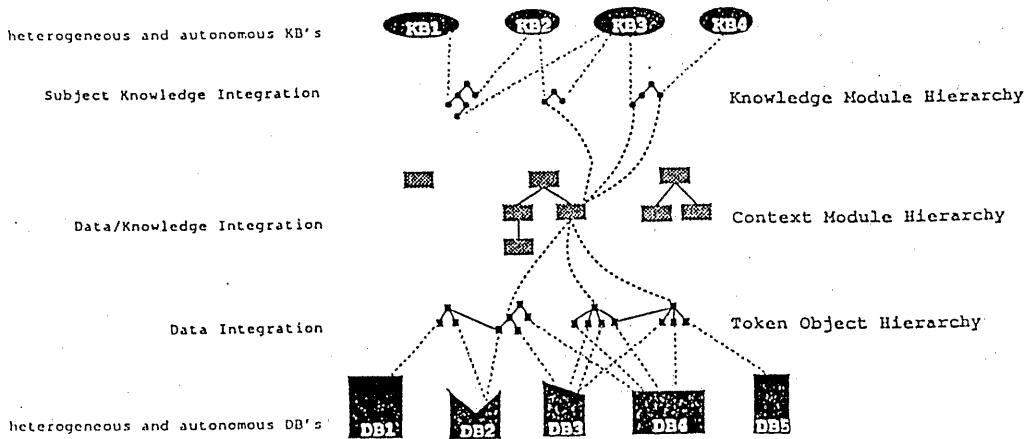


図2 データ知識超結合モデル

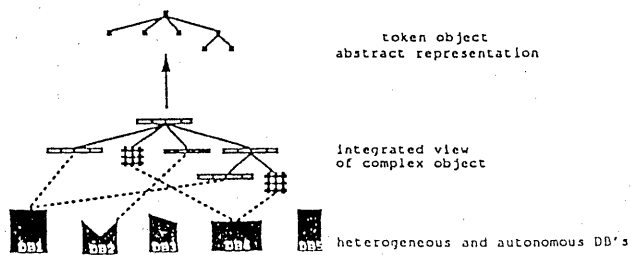


図3 トークンオブジェクト階層

- A. 同一オブジェクトが複数のクラスに属していても、データをコピーすることなく扱うことができる。
- B. オブジェクトの更新に伴う参照関連の更新は困難な問題であった。オブジェクトの値、参照された値などを区別して扱う必要があり、特に更新連鎖の問題は困難な問題といえる。本モデルでは原データの変更で扱うことができる。
- C. オブジェクト識別子もデータの側で記憶しておけば、同じデータが複数のコンテキストモジュールからリンクされていても同一オブジェクトを表現していることが判定できる。オブジェクトのクラス間移動によってもオブジェクト識別子は変化しないため、オブジェクト指向モデルの1つの困難な問題を解決できる。

本モデルでは次の3つの階層（それぞれが複数個存在する）が独立に保持されて

- [1] データベース階層
- [2] 知識モジュール階層
- [3] コンテキストモジュール階層

さらに [1] と [3]、[2] と [3] の階層は時間的に変化可能な動的リンクによって結合している。

このようなモデルは人間が種々の新しい問題を解くときによく似ているといえる。

例2：新しい学校を作る場合、計画するために

1. 問題領域を明らかにし、どのようなデータや知識が必要かについて考える。
2. データを種々の場合から集める。例えば、地図、近隣状況、交通等。
3. これらのデータを扱える形に変換し学校の場所の条件、決定のための手続き、解の許容性に関する規則等の知識を付加する。

このモデルの大きな特色をまとめると次の3つになる。

[1] 独立性：データや知識を独立に変更できる。

[2] 動的変更：データと知識を必要に応じて連動させ得る。

[3] 協調処理：複数の独立なオブジェクト指向データベース（データは共有してよい）が協調的に処理を実行するのと等価なものが実現できる。

2. データベースの階層

分散システムにおいては、データは種々の場所で複雑な構造で蓄えられている。これらのデータはトークンオブジェクトモデル [23] に基づいて、複数個の階層にまとめられる。複合オブジェクトは、文章、ラスタイメージ、音等の種々のデータを基礎として構成される。概念的に関連のある異なるデータベースの統合のために、複合オブジェクトの型を構成オブジェクトの型を用いて記述し、複合オブジェクトの構成を構成オブジェクトの識別子を用いて記述する。

トークンオブジェクトは複合オブジェクトの構成階層の木または部分木の識別子である。1つのトークンオブジェクトは1つの複合オブジェクトを表現しオブジェクト構成階層でその下にある全ての複合オブジェクトを代表する。この表現は抽象的で実際のデータ値とは独立している。この階層の一番下のトークンを葉トークンと呼ぶ。葉トークンは実際のデータへの索引となっている。例えば地理情報システムでは、ある場所の情報、その場所の歴史、地図などが別々のデータベースに記憶されており、それらは異なる葉トークンからリンクされている。単純値を葉トークンに対応させることも可能であるが、ここではレコードや画像のように論理的にまとまったデータの集まりを葉トークンに対応させるものとする。図3にこの構成を示している。

トークンオブジェクト階層を用いると次のような利点がある。

- A. 分散マルチデータベースによる複合オブジェクトに対して統合的なビューを与える。
- B. データベースに対する抽象的な索引を構成する。複合オブジェクトの構成要素と、構成要素へのリンクとは独立に扱うことができる。
- C. 複合オブジェクト内のデータの異なる要素間の従属性を表現する。

例3：図4はデータベースに蓄えられている「学校の地図」と「学校の記述」が異なるデータを要素とする階層を構成している例を示している。これらの2つの要素は独立に更新できるのではなく、片方の更新が他方に影響することが多い。階層の記述の中には、データの意味制約も記述されており、複数のデータベースにわたる場合はプロトコルの形で表現される。文献[2]で示した地理データベースでは、出力は地図とその上の文字情報の合成で作られる。必要なデータを選別して出力できるような自由度の高いシステムを実現するためには地図の図形情報と文字情報は独立に蓄えられなければならない。

図5に示すように各トークンオブジェクトはトークンクラスの要素に対応する。1つのトークンクラスは複数のトークンオブジェクトを含むか、実データベースへのポインタを含む。

トークンオブジェクトはデータ辞書の代りではなく拡張したものといえる。1つのトークンオブジェクトは複合オブジェクトの基本構造を示す。応用による構造の違いに対処するためのオブジェクト再構成[7]はコンテキストモジュールで扱われる。

3. 知識モジュールの階層

分散システムでは知識は別々の局に蓄えられておりそれぞれは、異なる定義レベルや応用レベルを有している。知識には次のような種類がある。

- 1) データベースに対する意味制約：あるクラスのデータに対するスキーマやメソッドを定義する。
- 2) 対象分野に関する知識：対応するクラス内のオブジェクトに適用される知識。
- 3) 協調に関する知識：特定の応用・対象分野で、オブジェクトと対応する知識の協調を行なうための知識。

データベースに対する意味的制約はデータベース部分で扱われる。対象分野に関する知識は対象分野ごとに知識モジュールとしてまとめられる。知識モジュールは知識をカプセル化すると共に、対象分野に対するクラスにまたがる

知識、データベースにまたがるオブジェクト間に存在する知識（データベース側に蓄えられているのは応用に依存しないデータベースに対する意味制約）を表現する。対象分野に依存した知識は次のような形で表現される。

- A. 応用を表現したりオブジェクトを再構成するための規則。
- B. 各クラスのオブジェクトに対するメソッド。

データの定義域と値域

プロシージャ

関数に対する操作（構成、条件等）

関数として扱われる属性名（構成を隠すためのもの）

関数の形で表される属性値

- C. 制約、これには受動的なものと能動的なものがある。

知識の共有、再利用、合成、継承等を扱うために知識モジュールは階層を構成する。この階層は、一般によく知られている、汎化階層、部品階層、抽象化階層等がある。汎化階層上での知識モジュールは上位（Super）知識モジュールと呼ばれ、下の知識モジュールは下位（Sub）知識モジュールと呼ぶ。上位の知識モジュールに対して適用できる規則、メソッドは下位の知識モジュールにも適用できる。すなわち、下位知識モジュールは上位知識モジュールの一部の知識を受け継ぐと共に、それ独自の知識も持つことになる。

知識に対しても、複数の階層が定義でき、実際の知識とはリンクで結ばれる（図2参照）。このような構造を用いると、例えば人間を、性別、人種別、年齢別等、種々の分類を併立させることが可能となる。

抽象化階層は汎化階層を一般化したものである。汎化階層では規則や関数は同じ表現をとれるが、抽象化階層では上のモジュールの規則ほど抽象的な表現が用いられる。

例えば、ケガなら外科医、カゼなら内科医へ行くという二つの規則を抽象化した規則は病気なら医者に行くという形になる[8、9]。

部品階層は、単純な知識の組合せで複雑な知

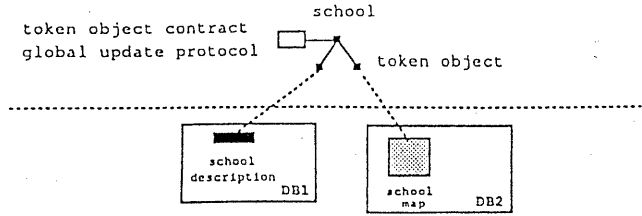


図4 例3

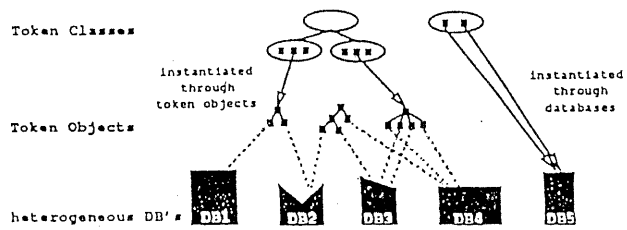


図5 トークンクラスとトークンオブジェクト

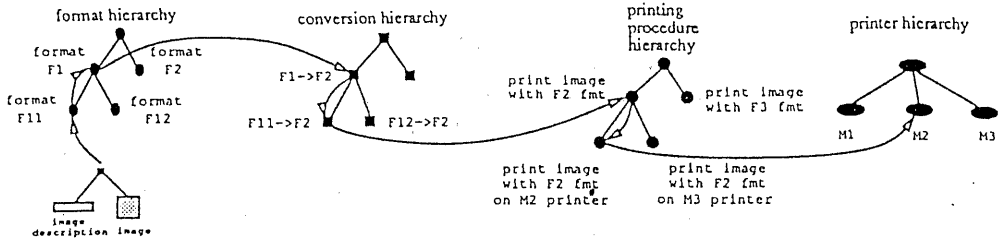


図6 ハイパーイメージプリンタ

識を表現するために用いられる。この階層で下の知識モジュールは部分モジュールと呼ばれる。例えば、C言語に関する知識モジュールの部分モジュールとしてシステムコールや算術演算に関するモジュールがある。モジュールの部分への分解は、各部品がそれ自体に意味があり、他の部品との関連が容易に記述できるときに限ってなされるべきである。例えば関数表現を許したオブジェクト指向規則表現言語L DL ++ [24] は知識モジュールやその階層を表現するのに適しているといえる。

4. コンテキストモジュール階層

コンテキストモジュールはデータと知識の統合のために必要なモデルである。

4.1 コンテキストモジュール

コンテキストモジュールはデータおよび知識への移入が主であるが、この移入の制御、データの変換、データと知識の協調のための知識も含んでいる。

[コンテキストモジュールの機能]

1. 複数のデータベースおよび知識ベースからデータおよび知識を取り込む。
2. 1つモジュール内のオブジェクトの型の違いを扱う。
3. データと知識の間のミスマッチを扱い、知識に適合したデータ形式に変換する。
4. 与えられた問題を扱うためにオブジェクトの再構成や仮想的な階層を構成する。
5. 目的に応じた協調知識により、データと知識の統合を行なう。
6. 他のコンテキストモジュールと交信するための公開されたインタフェースや特定の利用者向けのインタフェースを実現する。

[コンテキストモジュールの定義]

オブジェクトへのリンク
知識モジュールへのリンク
コンテキストモジュール内の知識
コンテキストモジュールに対する辞書
交信のためのインタフェース

4.2 データと知識の移入

移入の方法には次のようなものがある。

パスイン：データをコピーしてコンテキストモジュールで持つ。この場合知識モジュール内でデータが変化しても元データは元のままである。

チェックイン：データベースからデータがコンテキストモジュールにチェックインされると、チェックアウトするまで他のモジュールにチェックインすることができない。このことにより更新による矛盾が避けられる。

生成：実際のデータからビューのような機構で得るデータで、定義域やスキーマの不整合を扱うためのものである。新しいオブジェクトが生成されるが、ビュー更新と同じ問題が起り得る。

参照：データをリンクにより参照することで、データのコピーによる問題を避けることができる。

コンテキストモジュールの役割は次のようになる。

スキーマや定義域の不整合の解消
スキーマの結合
型の変換
複合オブジェクトの生成
特徴抽出

5. 例

本節ではオフィスオートメーションに対して本稿のモデルを適用した例を示す。

ハイパーイメージプリンタは仮想的なプリンタであり、レーザプリンタ、インクジェットプリンタ、プロッタ等解像度やデータ構造の異なる複数のプリンタの複合である。このサーバをコンテキストモジュール上で実現する場合のデータと知識は次のようになる。

データ：イメージ、イメージ記述、印刷への要求
対象分野に関する知識：与えられたイメージデータの与えられたプリンタへのデータ変換
協調に関する知識

これらを用いて印刷のプロセスは次のようになる。

- 1) 入力書式の検査
- 2) 入力書式に適合した適切な印刷方式の選定
- 3) 完全に適合するものがなければ利用可能な印刷方式の中から選定
- 4) イメージの出力形式の変換

図6はこのハイパーイメージプリンタの機能を示したものである。協調に関する知識として次の4つの階層がある。

書式階層

書式変換手続き階層

イメージ印刷手続き階層

プリンタ階層

例えば図6において、データベースで用意した書式がF11であったとする。書式階層に伴うとF11は一般化されてF1となる。しかしながら、書式変換手続きとしてはF2に対するものしか用意されていないとする。まずF1からF2への変換手続きを見つけこれを特殊化してF11からF2への変換手続きとする。プリンタM2が利用可能な場合はF2からM2へのイメージ変換の手続きを用いてF2の書式を出力する。

これらの階層で必要とする種々の知識や手続きは、知識ベースに記憶されていてそれらのそれらの抽象的な名前呼び出すこととなる。階層をたどるときはこれらの手続きを必要としないので実際に必要なのはF11をF2にする手続きとF2をプリンタM2にする手続きでこれらは知識ベースから入力される。

さらに地理情報システムのように利用者が必要とする情報に関連した付加情報が多い方がわかりやすいときには、出力可能なプリンタの解像度に応じて付加情報の選択を行なう必要がある。これは2つのコンテキストモジュール(データベース質問とハイパーイメージプリンタ)間の協調通信によって実現できるが本稿では詳細は省略する。

6. むすび

本稿では分散したデータベースと知識ベースを統合する新しいモデルの提案を行ない、簡単な実例でその有効性を示した。

コンテキストモジュールの階層を1つにして汎化階層に限り、データをモジュール内にオブジェクトとして入れて、知識へのポインタをメソッドや意味制約とすると、コンテキストモジュールの部分はオブジェクト指向データモデルとなる。この意味でこのモデルはオブジェクト指向データモデルの一般化であり、コンテキストモジュールの部分でカプセル化、継承といったオブジェクト指向的特性を生かすことが可能となる。関連研究としては仮想クラスとビュー [1, 17] があるがコンテキストモジュールによるモデル化の方がはるかに一般的である。知識データバケット [18, 21] や意味的コンテキスト [3] の概念も、データと知識を特定の利用形態に対して適用するための方式であるが、データと知識の結合が動的でなく、データベースや知識ベースの独立維持についても問題がある。

コントラクト [12] はコンテキストモジュールをデータベースと統合したものとなっている。メディアエータ [22] は利用者と情報資源の間を結び付ける知的エージェントである。コンテキストモジュールは知識を自分で持たずに移入して利用していること、再利用のために階層構造になっている点が一般的である。

本モデルは非常に一般的であるために、効率面からの批判等があると思われるが、データと知識の独立維持と協調の共方を実現するにはこのような方法が最適であると考えている。

本研究についてコメントを頂いたカリフォルニア大学ロスアンゼルス分校のChu教授とZaniolo教授に探謝する次第である。

なお、本研究は文部省科学研究費国際協同研究の補助を受けている。

文献

- [1] Abiteboul, S., A. Bonner, "Objects and

- Views", ACM-SIGMOD'91, 1991.
- [2] Arikawa, M., Y. Kambayashi, "Dynamic Name Placement Functions for Interactive Map Systems", The Australian Computer Journal, 1991.
- [3] Banerjee J. et al. "Data Model Issues for Object-Oriented Applications", ACM Trans. on Office Information Systems, 1987.
- [4] Chen, Q., "A Rule-based Object/Task Modeling Approach", Proceeding of the SIGMOD International Conference on Management of Data, 1986.
- [5] Chen, Q. and G. Gardarin, "An Implementation Model for Reasoning with Complex Objects", Proceeding of the SIGMOD International Conference on Management of Data, 1988.
- [6] Chen, Q., "Extending the Object-Oriented Paradigm for Supporting Complex Objects", J. of Computer Sci. & Tech. Vol. 3, No. 2, 1988.
- [7] Chen, Q. and Y. Kambayashi, "Nested Relation Based Database Knowledge Representation", Proceeding of the SIGMOD International Conference on Management of Data, 1991.
- [8] Chen, Q., W. Chu and R. Lee, "Providing Cooperative Answers via Knowledge-based Type Abstraction and Refinement", in Methodologies for Intelligent Systems, 5, Z. W. Ras, M. Zemankova and M. L. Emrich eds, North-Holland, Elsevier Science Publishing Co., Inc, 1990.
- [9] Chu, W. and Q. Chen, "A Structured Approach for Cooperative Query Answering", to appear in IEEE Transactions on Knowledge and Data Engineering.
- [10] Conklin, J. "Hypertext: An Introduction and Survey", Computer, Sep. 1987.
- [11] Haan, B. et al, "IRIS Hypermedia", CACM, Vol. 35. No. 1, 1992.
- [12] Helm, R., I. Holland and D. Gangopadhyay, "Contracts: Specifying Behavioral Compositions in Object-Oriented Systems", OOPSLA'90, 1990.
- [13] Kent, W. "Solving Domain Mismatch and Schema Mismatch Problems with an Object-Oriented Database Programming Language", Proc. VLDB'91, 1991.
- [14] Kim, W., "A Model of Queries for Object-Oriented Databases", Proc. of VLDB'89, 1989.
- [15] Li, Q and D. Mcleod, "An Object-Oriented Approach to Federated Databases", Proc. of IEEE International Workshop on Interoperability of Multidatabase Systems, 1991.
- [16] Litwin, W. L. Mark and N. Roussopoulos, "Interoperability of Multiple Autonomous Databases", ACM Computing Survey, Vol. 22, No. 3, 1990.
- [17] Madsen, O and B. Moller-Pedersen, "Virtual Classes: A Powerful Mechanism in Object-Oriented Programming", OOPSLA'89, 1989.
- [18] Pedersen, C et al, "Data and Knowledge Bases as an Integral Parts of a Distributed Object Infrastructure", Proc. of IEEE International Workshop on Interoperability of Multidatabase Systems, 1991.
- [19] Raffi, A. "Multidatabase Management in Pegasus", Proc. of IEEE International Workshop on Interoperability of Multidatabase Systems, 1991.
- [20] Sheth, A. and J. Larson, "Federated Database Systems", ACM Computing Survey, Vol. 22, No. 3, 1990.
- [21] Weishar, D. and L. Kerschberg, "An Intelligent Heterogeneous Autonomous Database Architecture for Semantic Heterogeneity Support", Proc. of IEEE International Workshop on Interoperability of Multidatabase Systems, 1991.
- [22] Wiederhold, G. et al, "A Mediator Architecture for Abstract Data Access", Tech. Rep. No. STAN-CS-90-1303.
- [23] Woelk, D., W. Kim and W. Luther, "An Object-Oriented Approach for Multimedia Databases", Proceeding of the SIGMOD International Conference on Management of Data, 1986.
- [24] Zaniolo, C. "The Logic Data Language LDL++", Tech. Rep., 1992.