

TM-0680

知識処理応用のための
データベース利用環境について

宮地泰造, 辻 秀一(代表),
金枝上教史

February, 1989

©1989, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191-5
Telex ICOT J32964

Institute for New Generation Computer Technology

知識処理応用のためのデータベース利用環境について

An Environment for Knowledge Processing Based on a Database Management System

宮地 泰造* 辻 秀一* 金枝上 敦史**

Miyachi Taizo Tsuji Hidekazu Kanaegami Atsushi

* 三菱電機情報電子研究所, ** ICOT

* Mitsubishi Electric Co. ISED Lab. ** ICOT

1. はじめに

知識処理システム(KIPS)の開発が活発に行われている。そのようなシステムの中において、データベース、知識ベースが重要な役割を演じている。我々は知識ベースの研究とICOT周辺の知識処理システムへの知識ベース、データベースの提供のためにKappa(Knowledge Applications Oriented Advanced DBMS/KBMS)の研究開発を行っている[Yo88a]。Kappaは、ユーザインタフェース層、知識ベース層、データベース層から構成されている(図1.1参照)。

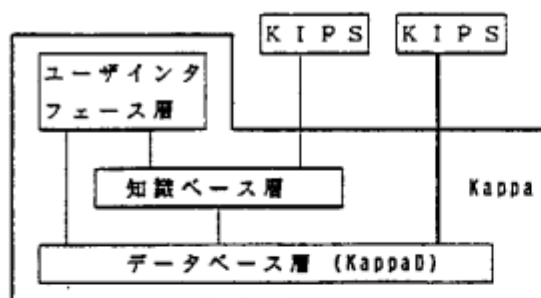


図 1.1 Kappa の構造

ユーザインタフェース層はKappaとユーザとのインタフェースを実現する。知識ベース層は演繹データベースとDOOD(Deductive & Object Oriented Database)の実現を目指している。データベース層(以下、「KappaD」と略す)は、知識ベース層およびユーザにデータベース管理システムの機能を提供する。

現在KappaDはすでに開発されており、いくつ

かの応用システムとKappaDとが接続されて、応用システムからKappaDが有効に利用されはじめている。KappaDはオブジェクト指向の概念を取り込んだ論理型プログラミング言語:ESP[Ch84]で記述されており、ワークステーション:PSI[YU84]上で稼動している。

本稿では、KappaDについて、知識処理の応用分野の要求に合わせて拡張したデータベース機能の特長とその有効性について報告する。まず、第二節では応用分野とその反映について述べ、第三節は項による知識表現、第四節は応用分野への拡張性とプログラミング言語の操作性の改善[Ba88]、第五節は意味ネットワークと多重継承、最後に第六節でさらなる改良のための議論について述べる。

2. 応用分野の要求とその反映

2.1 応用分野の要求

KappaDではKIPSの応用分野の多様でかつ変化する要求に応えることが一つの出発点であった。ICOTおよびその周辺の知識処理システムの応用分野としては、自然言語処理、定理証明処理、エキスパートシステムなどがある。より詳細には、電子化辞書[E088]、知的対話[KI88]、機械器訳[E088]、線形代数の証明支援[Sa88]、ハイブリッド型エキスパートシェル[HK86]、エキスパートシステム(レンズ設計システム)[HK88]がある。

この中でとくに自然言語処理と定理証明とを中心に応用分野からの要求を整理した。それぞ

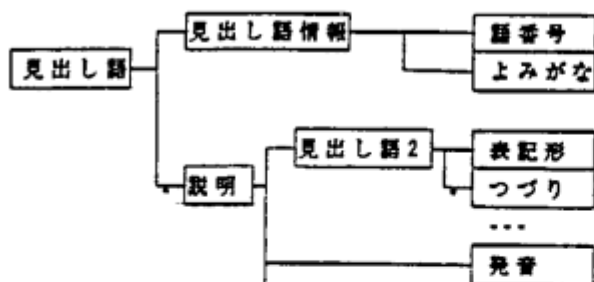


図 2.1 国語辞書のデータ構造例

れの応用分野のデータと応用分野からの要求はつぎのようであった。

(1) 自然言語処理

自然言語処理システムはいろいろの辞書を使用する。4つの基本辞書と各種専門辞書そして概念辞書の作成が行われており、その大きさは、基本辞書が各20万件、専門辞書は10万件、概念辞書は90万件になる見込みであった。図 2.1は基本辞書の1つである国語辞書のデータ構造の例である。この辞書は見出し語の単位に情報がまとめられており、全体は葉に値を持つ木構造であり、図中の*印のついた属性はその下に繰り返しが存在することを現している。例えばSpeech Informationは複数の部分木を持ち、Variant Formに複数の値があることを示している。また概念辞書は概念をノードとし、各概念間をis-a, has-aなどの関係で結んだネットワーク構造の集まりである。電子化辞書開発のプロジェクト[ED88]では、これらの辞書を開発している。そして、これらの辞書の利用方法として、オリジナルの辞書を個別応用で利用するために、辞書のスキーマの変更が必要とされている。また、辞書の共同利用も必要とされている。

われわれは自然言語処理を分析した結果、電子化辞書の格納に着目した。電子化辞書の要求の中でデータベースに対する要求として、『国語辞書に見られるような入れ子構造や繰り返し構造の素直な格納・操作』、『概念辞書の概念間の関係の素直な格納・操作』、『大量のデータの蓄積(数十万件のデータ)』、『高速な検索(実用システム・レベル)』、『スキーマの柔軟性(属性の追加・削除など)』、『マルチユーザのための資源管理』があった。さらに継続した調査の結果発見された要求に、『意味ネットワークの複数の観点からの利用』

[TN87]、『意味ネットワークにおける多重継承が起こす不整合の回避』[To87],[TN87]がある。

(2) 定理証明処理

定理証明のためには、多くの公理、推論規則、定義、定理そして証明が必要とされ、これらのデータは、1つの理論、そして証明によって得られた推論規則という2種類に分けられる。この理論は、構造を持たないテキスト形式と、内容が木構造に展開された項の形式との2つの形式に表現される。推論規則は項にして表現される。

われわれは定理証明処理を分析した結果、テキスト・データと項の格納・検索に着目し、その実現を行った。

以上の2つの応用からの要求の他に、知識処理全体に関係して、多種類のデータを多機に処理できるための新たな機能の追加要求が出るのが予測された。

さらに現在、データベースの容易な利用のために『プログラミング言語とデータベース言語のミスマッチ』[Ba88]が話題となっている。

2.2 要求の反映

知識処理の応用分野からの要求に応えるために、KappaDで追加したデータベース・モデルおよび機能を説明する。

・非正規関係

国語辞書に見られるような入れ子構造や繰り返し構造の素直な格納・操作を実現するために、KappaDでは非正規関係(Nested relation)をデータモデルとして実現した。KappaDでは、非正規関係の表現指向アプローチ[Hi87]をとっており、正規関係の効率良い格納方法として非正規関係を用いている。たとえば辞書は非正規関係の構造をしており、KappaDではそれを効率良く蓄積でき操作できる。

・意味ネットワーク

概念辞書の概念間の関係の素直な格納・操作を実現するために、KappaDではデータモデルの一つとして意味ネットワークを実現した。一般に、意味ネットワークでは多重継承が起こす不

整合の問題がある。KappaDの意味ネットワークでは複数の観点からの利用を可能にし、その結果多重継承が起こす不整合を回避できるようにした。この応用方法については第五節参照。

KappaDでは、一般の意味ネットワークをネットワークと呼び、KappaDの意味ネットワークはネットワークの集合である。ネットワークは、ノードとリンクから構成され、ノード間はリンクで接続される。ノードおよびリンクは属性と属性値を持っている。また、異なるネットワーク内に同一ノードが存在できる（図 2.2参照）。

意味ネットワーク操作には、ノードとリンクの検索、読み、追加、削除、属性値の更新があり、さらにノードには現在位置の設定がある。操作の対象はノード、リンク、ネットワークである。操作は1つのネットワーク内で行われるが、ネットワーク間の移動もできる。

・項の蓄積・検索

項の蓄積および、単一化やパタンマッチによ

る項の検索機能を実現した。項は、等価条件ではなく単一化やパタンマッチにより検索できる。この検索機能は観言語であるESPの論理型言語の特長である単一化を実現することにより、ESP利用者はからの使い易さを確保する目的と定理証明支援からの要求を反映させた結果である。また、利用者はこの項の蓄積・検索機能を利用することにより手続きとデータのカプセルやESPのオブジェクトを蓄積・検索できるようになった。詳細は第三節を参照。

・高機能コマンドの実現

多種類のデータに対する複雑な処理を容易にするためには“応用分野に対する機能拡張性”が必要である。KappaDでは、これを実現するために、必要なデータ操作を高機能コマンドとしてKappaDに登録できるようにした。これにより利用者定義のデータ操作を実現する環境の基盤を作った。これにより個別の応用分野特有の機能や複数の応用分野で共用される機能を実現することができる。詳細は 4.1節参照。

その他、プログラミング言語とデータベース言語のミスマッチへの対応では、オブジェクト指向プログラミング言語：ESPを観言語に採用したことによるメリットが考えられる。出てきている。詳細は 4.2節参照。

また、ICOT周辺の応用分野とそれらが使用するであろうKappaDのデータモデルおよびデータ型：項の関係は現在図 2.3のようになっている。

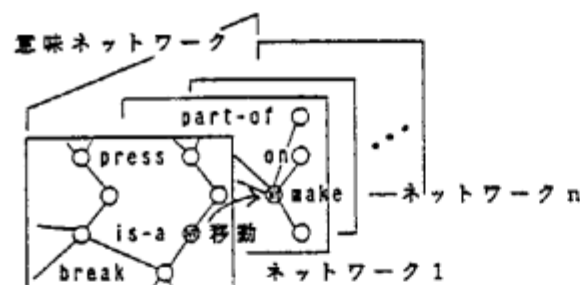


図 2.2 意味ネットワークの構造

応用分野

自然言語処理

定理証明処理 EXPERT-SYSTEM

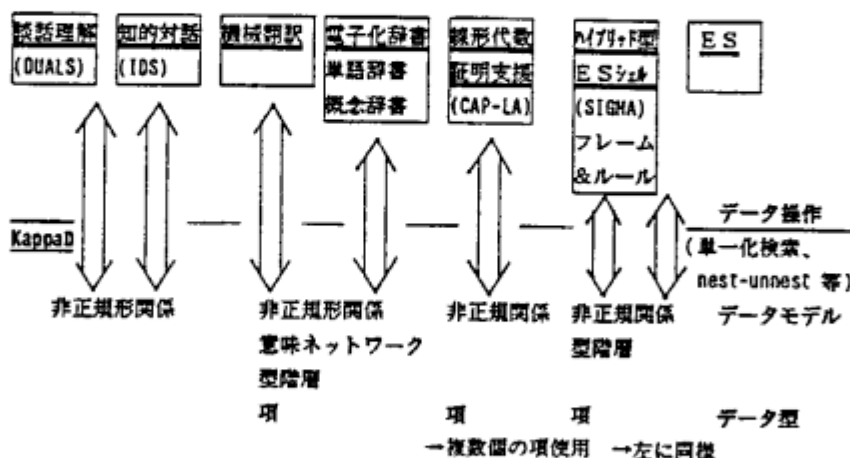


図2.3 応用分野と KappaD のデータモデルとの関係

3. 項 (TERM) による知識表現

項 (ターム) は、多くの応用分野において知識表現の内部形式として非常に有効である。項の定義と例はつぎのとおりである。

項の定義：

- (1) 変数または定数は項である。
- (2) f が n 引数 ($n \geq 1$) の関数記号でかつ $t_1, t_2, \dots, t_n$ が項であるならば、 $f(t_1, t_2, \dots, t_n)$ は項である。

〔例 3.1〕項： 好き (MissY, 太郎) : ~好き (太郎, MissY) (":~" が f に対応する)

KappaDでは属性値のデータ型として項を提供している。利用者は、木構造を項で表現だけでなく、定理、制約、メソッド、ルール記述を含むフレームが項で表現できる。KappaDは項を簡単に蓄積・検索できるので、これらのデータを簡単に扱える。

項の提供により利用者は、線形代数の証明支援における定理の格納およびハイブリッド型エキスパートシェルにおけるデータと手続きとのカプセル化が可能になる。

エキスパートシェルのフレームや証明支援の定理では、その内部がさらに独立した意味を持つ部分に分かれているために、1つのフレームや定理を複数の部分に展開し個々を操作しやすい形態にして蓄積している。

ここで応用例をいくつか上げる。

〔例 3.2〕定理用知識ベースの構造例

定理知識は、定理、ソート、ファンクションに関する3つの部分から構成されており、さらにその各々が名前、本体、前提条件、結論、テキストから構成されている。ここで、データ型が項である属性は名前以外の属性である (図 3.1参照)。

〔例 3.3〕フレーム用知識ベースの構造例

SIGMA[MX88]ではフレームを要素とする集合とその集合を要素とする集合を操作できる。そこで、フレームが手続き部、スロット部、集合関係部の3つの部分から構成されている。フレームの操作を容易にするために、各部を別個に項として蓄積している (図 3.2参照)。

定理知識

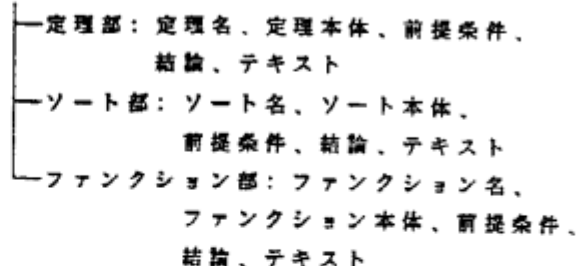


図 3.1 定理の構造例

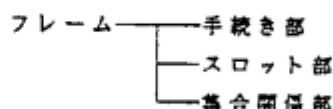


図 3.2 フレームの構造例

4. 応用分野への拡張可能性

KIPSの開発環境をより高度にして生産性を上げることが重要である。データベースの機能に基づく開発環境の拡充方法として、応用分野に対応する機能強化と使い易さの向上が考えられる。

4.1 高機能コマンドによる拡張可能性

個別の応用分野に必要とされるデータ操作や複数の応用分野で共用できるデータ操作機能を、データベース管理システムが管理・提供することにより、データベース管理システムの機能を応用分野別に強化し利用を容易にできる。

KappaDでは高機能コマンドとして、利用者定義のデータ操作をESPのプログラムとして容易に登録でき、実行することができる。これにより、データ操作の拡張を可能にした。

4.2 プログラミング言語の操作性の改善

プログラミング言語の操作性を妨げる要因として、親言語とデータベース操作言語との間に次の2つのミスマッチを考えることができる。

- (1) 操作記述レベルの相違によるミスマッチ
- (2) データ型の相違によるミスマッチ

KappaDでは、インプリメンテーション言語と親言語にESPを採用した。ESPはオブジェクト指向言語と論理型言語の特徴を備えており、これらの特徴によりプログラミング言語の操作性を上げることができる。本節ではその理由について分析し、さらに高機能コマンドによる操作性の改善方法を述べる。

最初に、(1)の操作レベルの相違によるミスマッチを考える。一般的に、オブジェクト指向型プログラミング言語では、オブジェクトの中にはオブジェクトに属するメソッドが複数個存在していて、複数の機能の個々に対して呼び出しの入り口がある。そこで、このオブジェクトの概念をDBMSのユーザインタフェースとして使うと、データベースの論理的概念であるテーブル、タプル、集合などをオブジェクトとすることにより、個々の操作をオブジェクトに埋め込むことができる。例えば、テーブルに対して検索、追加、削除の操作を埋め込む。これにより利用者は、論理的な概念に対して、操作の要求を出すことができる。

KappaDでは、親言語を対象指向型プログラミング言語にしたことにより、このユーザインタフェースが可能になる。

つぎに(2)のデータ型によるミスマッチについて検討する。応用分野で利用されるデータが豊富になるに従い、データベース操作言語およびプログラミング言語ではそれぞれに新しいデータ型の操作のためのデータ構造や操作機能を拡張していくことが行われてきた。その結果として、データベース操作言語とプログラミング言語間で、新しいデータの操作においてさらに新たなミスマッチが起きることが予想される。このミスマッチの問題は大きい問題として議論されはじめている。現在指摘されているミスマッチとしては、“tuple at a time”と“set at a time”の問題がある。ミスマッチは、たとえば複数のメディアを対象とする場合に、さらに大きくなると考えられる。これは是非解決すべきであると考えられる。

この解決方法の一つとして、両方の言語を改良して歩み寄りを図る方法がある。歩み寄りが難しい場合の解決策としては、データベースの操作をも含んだ新しいプログラミング言語を開

発する方法が考えられる。ICOTでは構造を隔て扱う論理型言語:CRLを検討している[Yo88b]。

以下にKappaDの現状を分析してみた。集合という概念はKappaには有るがESPには無いので、“tuple at a time”と“set at a time”のミスマッチは残っている。しかし親言語をオブジェクト指向言語として使用できるESPの環境では、集合をオブジェクト、個々の値をインスタンスと考えればよく、集合というオブジェクトから個々の値を貰ってこられればよい。集合演算については、該当する集合に対して集合演算の要求をメッセージ通信するか、集合演算を行えるオブジェクトに集合演算の要求をメッセージ通信すればよい。KappaDでは、集合演算について後者の方法が使えることと、KappaDを介して集合から値を得るようにできるので、結果として“tuple at a time”と“set at a time”のミスマッチは少なくなる。

また、集合演算と言う便利な機能があっても、集合の操作はLISTというデータ構造で行えれば良いと考えている利用者は、ESPの利用者を含めてかなり居ると考えられる。その人達には、集合をLISTの形式にして返せば、ミスマッチは削減されて利用者は使い易くなる。Kappaでは集合をLISTにするインタフェースを高機能コマンドとして利用者が登録することによりこれが実現できる。

5. 意味ネットワークと多重継承

多くの応用分野において、対象としている概念や概念間の関係を体系化して利用することは有効である[Le86]。概念間の関係は、意味ネットワークにより自由に表現・操作できる。

概念辞書は、体系化された概念および概念間の関係の代表格であり数十万という大量の概念を取り扱う。本節では、KappaDの意味ネットワークの有効な応用方法を概念辞書の場合について述べる。

概念辞書の知識構造には、「概念記述」と「概念体系」があり、とくに概念体系の表現に意味ネットワークが自然で自由度がある。

概念記述は、概念自体の意味や使われ方の記述である。例えば、『燕が翼で飛ぶ』という日本語に対して、概念記述はたとえば、

C1: 概念『燕という動物が速く動く』;

概念1:速く動く、

関係子:agent、

概念2:燕という動物、

C2: 概念『翼で飛ぶ』;

概念1:速く動く、

関係子:instrument、

概念2:動物が飛ぶなどの移動に使う部分の記述である。簡単な概念記述は正規関係でも表現できるが、より複雑な構造になったときには、非正規関係や項の構造が有効になると考えられる。

概念体系は、概念間の関係の記述である。例えば、“ジョイナーは人間である”の関係がある。ここで、概念は“ジョイナー”、“人間”であり、概念間の関係は“is-a”である。KappaDの意味ネットワークは、概念を“ノード”で概念間の関係を“リンク”で表現できる。分類階層は、関係の種類が1つのクラス間の関係を表現する。クラスは数学における広義の集合とし、分類とはクラスのことである。意味ネットワークとの関係は、クラスがノードに、関係がリンクに対応する。

さて、概念間の関係は利用される場面により異なる観点からの見え方が必要であることが指摘されている[TH87]。KappaDが提供する開発環境では、利用者がリンクの属性に属性値として観点を蓄積することにより、異なる観点や新たな観点を使用することが可能になる。

[例 5.1] シソーラスの異なる観点の追加例

観点:『用途』に基づいたシソーラスが有り、それに新たな物理的な観点の一つとして『形状』を追加することができる(図 5.2参照)。

さらに、シソーラスなどの概念関係では多重継承が利用されるが、一般に多重継承は不整合な属性値を同時に継承してくる可能性がある[To86]、[TH87]。これを処理できる知識表現が必要になる。以下に、不整合が起きる例とKappaDによる回避方法の例を示す。

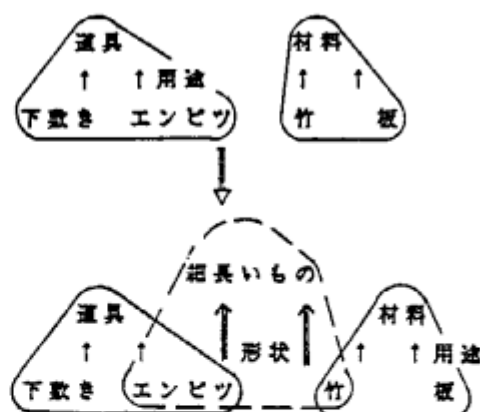
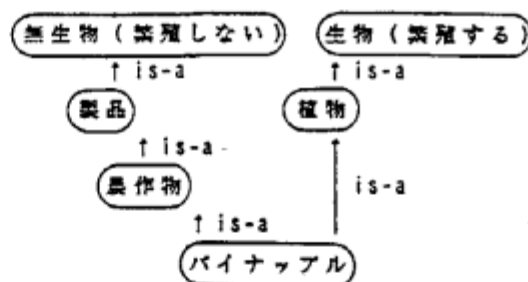


図 5.2 異なる観点の追加例

[例 5.2(a)] 多重継承における不整合

is-a の関係を使って概念間の関係を表現すると、継承の結果「パイナップル is-a 生物」から「パイナップルは繁殖する」、『パイナップル is-a 無生物』から「パイナップルは繁殖しない」が得られる(図5.3 参照)。これは不整合であり、同様な不整合は、シソーラスにおいても起きている[TH87]。



繁殖しない <=> 繁殖する

図 5.3 多重継承における不整合の例

KappaDの提供する開発環境では、利用者は、シソーラスを意味ネットワークで表現できるだけでなく、リンクの属性と属性値とをうまく利用することにより多重継承の不整合を回避することができる。すなわち、利用者はリンクの属性に属性値として観点を蓄積しておくことにより、知識を利用している場合に不整合が起きたとき、利用に於ける観点をリンクの属性値として示すことにより不整合を回避して必要な継承情報を得ることができる。

[例 5.2(b)] 多重継承における不整合の回避例

利用者は、is-aのリンクに観点：『保存』、『栽培』を属性値として蓄積しておくことにより、不整合を回避して、観点に合致した属性値を得ることができる（図 5.4参照）。

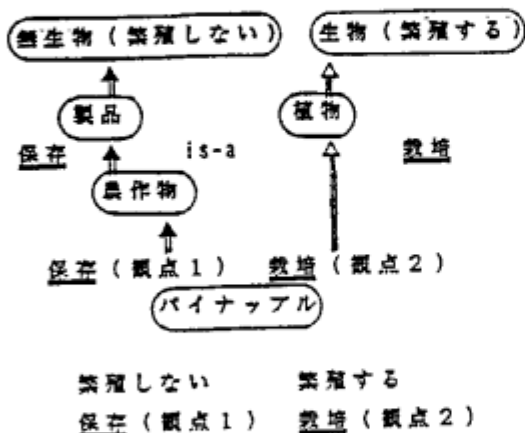


図 5.4 観点の蓄積による不整合の回避例

6. 議論

知識処理システムの多様な要求に応えるために、複数のデータモデルや拡張した機能について第2節で述べ、以下応用方法について述べた。KappaDでは、応用分野の要求に対して通常の関係データベース管理システムの機能のほかに、非正規関係や意味ネットワークや項の蓄積・検索機能、高機能コマンドを提供した。これにより複雑な構造のデータを素直に表現・蓄積・利用できる became になり、応用分野別のデータ処理機能を拡張できるようになった。また、利用者の上手な使い方により、意味ネットワークにおける複数の観点の格納やの不整合の回避もできるようになった。

しかし、必ずしも要求に対して十分な機能を実現できてない点もある。本章では、実現できた機能を順みながらこの改良について議論する。

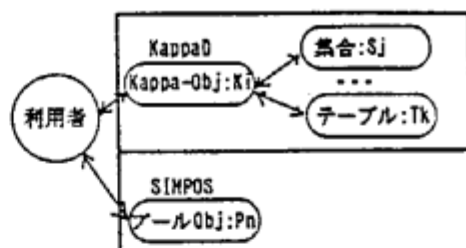
プログラミング言語のより良い操作性のためには、KappaDを介して集合から値を得られるだけでは不十分であり、集合やテーブルをオブジェクトとしてそれから直接に値が得られるようにすべきである（図 6.1 KappaD 部分参照）。

また、KappaDの簡単な利用方法として、SIMPOS[YU84]のプールオブジェクトの機能を考

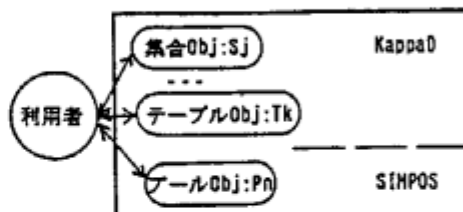
えている。このプールオブジェクトは、ハッシュテーブルなどの機能を持っており、プールオブジェクトに対する追加・検索・削除の機能が提供されている。この機能のインタフェースと同様なインタフェースでKappaDにデータを格納できることにより、ESPからのデータベースの利用がさらに容易になる（図 6.1参照）。

この他に、定理証明において、版管理の要求が出てきている。定理の証明には、証明方法の相違や証明を利用する人間の能力の相違により複数の証明の版ができると考えられる。証明の利用者が満足する版を見つけられるための版管理機能の実現が課題として考えられる。

また、近年知識処理システムにおいて、仮説推論の研究が盛んに行われている。有用な仮説を生成するためのデータベース機能の検討が必要と考えられる。たとえば、状況に応じて、ポイントを押さえた仮説の生成を実現するためには、状況とポイントとをインクリメントにかつ体系的に関連付けながら蓄積する機能、さらに仮説や仮説の生成方法を体系的に関連付けながら蓄積する機能などが課題として考えられる。仮説生成の手法の分析からの検討が必要と考えられる。



(a) 現在のアクセス形態



(b) 改良後のアクセス形態

図 6.1 オブジェクトへの直接アクセス

7. おわりに

知識処理システムに対して、データベース管理システムの機能を提供するために、ICOT周辺

の知識処理システムからの要求調査を行いそれに基づき KappaD を開発した。本稿では、その特長および有用性について報告した。KappaD では、主に電子化辞書と定理証明からの要求に応えるアプローチをとったが、実現したデータベース機能は他の知識処理にも十分有効である。知識処理システムからの要求はまだ多くあると考えられるが、要求に答えるために全ての機能を提供することは不可能である。今後は複数の応用分野で共用できる機能について引き続き調査分析し、開発を行っていきたい。

謝 辞

本研究の機会を与えて戴きご指導戴いた ICOT 第四研究室内田 俊一室長、横田 一正主任研究員、河村 元夫研究員、三菱電機 市川 照久部長、溝口 徹夫参事、平塚 尚志・ア・マ・ザ・に感謝致します。また、有益な助言を戴いた、JIPDEC 横塚 英氏、三菱電機の古武 洋氏、吉良 賢治氏、エキスパートシステム・グループのメンバー、および開発に協力して戴いた皆様に感謝します。

文 献

- [Ba88] Bancilhon, F., "Object-Oriented Database Systems," Proc., ACM PODS, pp. 152-162, 1988.
- [Ch84] Chikayama, T., "Unique Feature of ESP," Proc. FGCS'84, pp. 292-298, 1984.
- [ED88] Japan Electronic Dictionary Research Institute LTD., "Electronic Dictionary Project," 1988.
- [KI88] 近藤, 今村, "対話システム IDSにおける対話モデルと協調的応答生成", 情報知識工学と人工知能研究会, 88-AI-57, 1988.
- [Le86] Lenat, D., B., "CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks," The AI Magazine, 6, no. 4, pp. 65-85, 1986.
- [Hi87] 三浦, "非正規関係データベース理論の動向", アバウト・データベース・システム・シンポジウム 論文集, 12月, 1987年.
- [HK86] 三石, 吉良, 溝口, "知識ベース管理システム SIGMA の構築", 第32回情報全大, 1H-1, pp. 1179-1180, 1986.
- [HK88] 三石, 吉良, "知識ベース操作システム SIGMA の構築と評価実験", 第37回情報全大, 7H-8, pp. 1361-1362, 1988.
- [Sa88] Sakai, K., "Towards Mechanization of Mathematics - Proof Checker and Term Rewriting System," in Programming of Future Generation Computers, 1988.
- [TN87] 田中, 仁利, "上位/下位関係シソーラス ISAHP1 の作製 [I]", 情報自然言語研究会, 自然言語処理 64-4, pp. 25-44, 1987.
- [To87] Touretzky, D. S., Horty, J. F., and Thomason, R. H., "A Clash of Institutions: The Current State of Nonmonotonic Multiple Inheritance Systems," Proc., IJCAI, pp. 476-482, 1987.
- [Yo88a] Yokota, T., Kawamura, M., and Kanaegami A., "Overview of the Knowledge Base Management System (KAPPA)," Proc. FGCS '88, pp. 252-262, 1988.
- [Yo88b] Yokota, K., "Deductive Approach for Nested Relations," in Programming of Future Generation Computing II, eds. by K. Fuchi and L. Kou, North Holland, 1988.
- [YU84] Yokoi, T. and Uchida, S., "Sequential Inference Machine: SIM Its Programming and Operating System," ICOT Technical Report 87, 1984.