

この資料は日本エム・テクノロジー学会員専用です。
この資料を学会員以外がコピーしたり、学会員以外に配布することを禁じます。

Copy right : M Technology Association - Japan

日本エム・テクノロジー学会事務局
〒259-1193 神奈川県伊勢原市望星台
東海大学医学部・基礎医学系
大櫛陽一

Tel: 0463-93-1121 ext. 2140

Fax: 0463-96-4301

Email: youichi@keyaki.cc.u-tokai.ac.jp

第17回日本MUMPS S 学術大会

特別講演(3) 「オブジェクト指向プログラミング、データベース」

- ◇オブジェクト・データベースシステムの最近の研究・開発動向
田中 克巳 神戸大学工学部計測工学科
- ◇次世代データベース SIM
足立 哲夫 日本ユニシス株式会社システム技術本部

資料集

1991年2月1～3日

大 阪

オブジェクト指向データベースシステムの最近の研究・開発動向

Recent Trends on Research and Development of Object-Oriented Database Systems

吉川正俊 (京都産業大学)† 下條真司 (大阪大学)‡ 田中克己 (神戸大学)‡†
† Masatoshi Yoshikawa (Kyoto Sangyo University)
‡ Shinji Shimojo (Osaka University)
‡† Katsumi Tanaka (Kobe University)

Abstract

In this paper, we will report on the recent trends on the research and development of Object-Oriented Database Systems. Especially, we will report on (1) ECOOP/OOPSLA'90 Conference held in Ottawa, Canada, (2) OODBTG workshop held in Ottawa, Canada, (3) OODB reserch activities at MIT and Brown University, and (4) product development at several OODBMS vendors (Servio, Object Design, Ontologic, Itasca, and Versant) in United States.

1 OOPSLA/ECOOP'90 国際会議

今年で4年目を迎えるこのOOPSLA国際会議は、今年はヨーロッパの同分野の国際会議ECOOPと合同の形で10月にカナダのオタワ市で開催され、2000人以上もの参加者を集めた。Technical Sessionでは200件もの応募のうちから29件を選ぶという超難関であり、そのため質の高いものとなった。Technical sessionの量の少なさをカバーするため、多くのワークショップやチュートリアル、BOFが開かれた。ここでは主にデータベース関連の話題について述べる。

1.1 Tutorial

全部で19のチュートリアルが行なわれ、かなり専門的なものから、初心者を対象としたものまで様々な内容にわたるがデータベース関連のものは以下のとおりである。

1.1.1 オブジェクトデータ管理 (R.G.G. Cattell)

話者は、Sunにおけるデータベースシステムのエンジニアリングマネージャであり、Simplifyの開発などに携わっていた。要点を以下にまとめる [1]。

- オブジェクト指向データ管理システム (ODMS) は、先進的なデータモデル (オブジェクト指向、関数型、拡張関係) と、DBアーキテクチャ (拡張DBシステム、DBプログラミングシステム、DBシステムジェネレータ、オブジェクトマネージャ) の組合せによって分類することができる。商用のODMS製品は、オブジェクト指向データモデル + モノリシック (GemStone, Orion) あるいは、オブジェクト指向データモデル + DBプログラミング言語システム (Ontos, Object Store, Versant, Objectivity, O2, ...) アプローチに基づいている。拡張関係、関数型、DBジェネレータ、オブジェクトマネージャは、より研究寄りのサイドから提案されている。
- ODMSの大きな利点は、言語の統合、より強力なデータモデル、そして性能である。
- 商用のDBMSは、工業応用または従来のビジネス応用を目指しており、その両方を目指しているのではない。我々のすべての要求を満足させることのできるDBMSは存在しない。DBMSを選択する場合は、応用からの要求 (質問、言語の統合、データ型、並行性、分散、従来からの応用...) に慎重に優先順位を付けることが本質である。

- ODMS を構築する場合、関係データベースシステムから出発したときの困難な問題としては、言語の統合、性能、新しいデータ型などがあり、オブジェクト指向言語から出発したときの困難な問題としては、ビュー、利用者インタフェース、標準化、OLTP の性能などがある。

- 研究課題としては以下のものを挙げるができる。

- データベース機能のより良い支援のための OO プログラミング言語の拡張。
- リスト、集合、埋め込みプログラムなどを支援する新しい、より強力な質問言語。
- ビュー、カプセル化、手続き的および宣言的部分を組み合わせた、データ独立性に対する解決法。
- ビジネスおよび工業応用双方に対して妥当な性能を得るデータベースシステム。

- 質問言語については、未だ標準が存在せず、言語独立な SQL の拡張というアプローチと、質問言語機能を追加したプログラミング言語という 2 つのアプローチがある。

- 話者が行ったエンジニアリング DB ベンチマーク [2] の結果が興味を集めていた。このベンチマークの特徴は、モデル独立な点で、DBMS のアーキテクチャに焦点を絞ったものになっていることである。Sybase RDBMS と種々の ODBMS に対して、lookup, traversal, insert という比較的単純なベンチマークを行っており、ODBMS の圧倒的な効率の良さが示されている。話者は、ODBMS が効率上優れている理由として、ODBMS がキャッシュ、OID による参照、swizzling (OID を物理的なポインタに書き換える操作) をサポートしていることを挙げている。

1.1.2 Moving to an OODBMS (Mary E. S. Loomis)

話者は、Versant Object Technology 社の技術サポート担当副社長である。ODBMS の構成要素を DBMS、Object Model、Goodies ととらえ、ODBMS ベンダーという立場もあり、ODBMS が既存の RDBMS に比べてどのような点ですぐれており、RDBMS と ODBMS の平和的共存の技術 (種々の SQL ゲートウェイ) や、RDBMS から ODBMS への段階的移行の重要性を強調していた [3]。

ODBMS と関係データベースの選択基準としては、以下のいずれか一つでも該当するものがあれば、ODBMS を使うべきであるとしている。

- C++, C あるいは Smalltalk を使用しており、共有、永続性などを必要とする場合。
- 意味論を表現するために 6 個以上の表を必要とするような応用。
- 非レコードデータ構造を必要とする応用。
- 値に基づかない探索を必要とする応用。
- 5000 行を越えるコードがあり、その保守性を必要とする場合。
- 関係データベースによる性能が 10 倍以上遅すぎる場合。
- 応用がグループ作業の環境で実行される場合。

1.2 Technical Session

1.2.1 論文発表

データベース関連の論文は、CMU の B.S.Lerner らのデータベース再編成 [4] に関するものと、Alberta 大学の D.D.Straube らの OODB の質問の型の一貫性に関するもの [5] の 2 件だけであった。後者は、質問処理において生成される中間結果がヘテロニアスなオブジェクトの集合になることに着目し、この各要素に対して後に適用されるメソッドが用意されているかどうかを型検査するものであり、型集合 (conformance) 間の順序関係をもとにして型推論の規則を提案している。

1.2.2 パネル: オブジェクトデータベース管理の諸問題

本パネルには代表的な OODB ベンダー (Ontologic: T.Andrews, Versant: M.Loomis, ObjectDesign: D.Weinreb, Servio: J.Stein (司会)) と、HP の B.Ken という顔ぶれが揃ったこともあって盛況であった。少し難しいセッションになると客が減ってしまうことを考えると、OODB への関心の高さがうかがえる。パネルは以下の 3 つの課題についてそれぞれの意見が述べられた。

- 効率
応用・キャッシュ・オブジェクト管理部を一体化したクライアントプロセス、storage 管理部・ページキャッシュ・DB を一体化したサーバ

プロセスにもとづく OODB のアーキテクチャの重要性の指摘 (M.Loomis)。

パフォーマンスは 10% の inspiration と 90% の perspiration から得られるとし、ODBMS の設計・実現・解析・チューンアップのサイクルを高能率で行なえることの重要性の指摘 (T.Andrews)。

4GL など書かれる商業的応用と C や C++ など書かれる工業的応用は区別すべきである (D.Weinreb)。

- 問い合わせ言語について

SQL の拡張 vs. プログラミング言語の拡張という 2 つの方法があるが、前者はインピーダンスミスマッチがあるが SQL ユーザからの移行が容易で標準化が比較的やりやすいという利点がある。後者は、インピーダンスミスマッチはないが、言語毎の拡張が必要で、また、標準化が難しいという欠点がある (J.Stein)。OSQL (Object SQL) がとりあえずの急場しのぎであろう。

また、標準化を行なうためには、Object に対する標準的な概念が必要である。特に、オブジェクトを構文的にとらえることは簡単だが、意味的に取り込むことはかなり難しい (D.Weinreb)。

OSQL と SQL に関しては、RDB と OODB の共存を考え、SQL データや SQL プログラムを OODB で処理できるようにする必要がある (M.Loomis)。

SQL は重要。declarative (宣言的) な SQL に imperative な機能 (すなわちメソッド) を呼べるようにするアプローチが有効 (T.Andrews)。

OQL は DDL/DML 両方の機能を持ち、単なる集合的検索以上のものをサポートすべき。キーポイントは、相互運用可能性であり、参照モデルの共通化が必要 (B.Kent)。

- OODB 研究と今後

今後の研究の方向として OO 質問代数 (メソッド実行や最適化や view 含む)、より進んだ並行制御 (グループウェアや性能向上のためのもの、長時間トランザクション)、スキーマ進化と版管理、安全性、完全性制約・推論機構のとりこみ、オブジェクト指向設計の DB への導入、などが重要項目としてあげられた。

1.2.3 OMG (Object Management Group) の BOF

OMG は主としてオブジェクト指向ソフトウェアに関するベンダーを中心とした国際的なコンソシアムであり、オブジェクト管理に関して、用語統一・参照モデルの構築、応用プログラムインタフェースの作成、分散オブジェクト管理、オブジェクト指向データベースへのインタフェースの作成、その他の共通的なサービスの仕様決定などを目的として 1989 年にスタートした。主な議論の対象は、オブジェクト参照モデル、データ管理システム、グローバルなメッセージパッシング、質問言語であったが、ただ、内容は技術的な議論は少なく、むしろ既存の技術のどれを選択するかに終始しているような印象を得た。

2 OODBTG ワークショップ

1990 年 10 月 23 日に、カナダ、オタワ市のシャトーローリエ (Chateau Laurier) ホテルにおいて本ワークショップが開催された [6]。OODBTG (オブジェクト指向データベース・タスク・グループ) は 1989 年 1 月に設立された団体で、ANSI/SPARC の X3 標準化委員会 (情報処理システムに関する標準化委員会) に対するアドバイザリーグループの一つである DBSSG (Database Systems Study Group) の要請により、オブジェクトデータベースシステムの標準化の候補となる項目を決定するための調査活動を行なっている。OODBTG の最終技術レポートは 1991 年に DBSSG に提出される予定で、この内容には X3 標準化委員会がオブジェクト指向データベースシステムの標準化活動をどのように行なうかに関する勧告が含まれる予定である。

本ワークショップは、1990 年 5 月にニュージャージー州のアトランティック市において開催されたワークショップに続く 2 回目のもので、この 2 回のワークショップの内容を考慮して最終技術レポートが作成される。さらに、OODBTG は、現在、オブジェクトデータベース管理システムに関する参照モデル及びサーベイを作成中である。

以下では、本ワークショップの概要を述べる。

2.1 オブジェクトシステムの標準化の諸側面 (R.Osborn, Concurrent Computer Corp.(英国))

OODB の標準化に関しては、従来の伝統的なデー

データベースの立場からではなく、トータルな情報システム(オブジェクトシステム)の立場から行なうべきであることを強調し、この立場から、標準化されるべき種々の項目を挙げている。特に、オブジェクト記述については、外部インタフェースと内部インタフェースの区別の重要性及び各々の内容について述べている。

ここで、「外部インタフェース」は、オブジェクトの識別性(identity)と型(type)を「内部インタフェース」は、クラス、メソッド、インスタンス変数を記述するものである。「型」は関数のインタフェースであり、オブジェクトの実現はその型とは独立で、同じ型でも相異なるオブジェクトの実現があっても構わないことにしている。

2.2 OODB の応用プログラムインタフェースに関する Strawman 参照モデル

(E.Perez, TI社)

応用プログラムとOODBMSとの間のインタフェースを次の3つのレベルに分類し、これに基づく参照モデルを提案している。

- (1) システム・インタフェース(システムやトランザクションの開始/終了など)
- (2) オブジェクト・インタフェース(オブジェクトの割り付け/解放, 参照変数への代入/比較, 永続性の指定, オブジェクトの更新やフェッチ)
- (3) 付加的な諸機能(拡張型トランザクション(入れ子型など)、オブジェクトのグループ化、メモリへの常駐性の指定、前バージョンの取り出し、オブジェクトのロック、オブジェクトの命名など)

これに基づき、GemStone, Iris, ObjectStore, VERSANT, ODE, Ontos, ORION, Postgres, 及びZeitgeist(TI社で作成)の機能比較を行なっている。尚、本調査を行なった同社は、1990年8月にDARPA/ISTOと3年間の契約を締結し、“Modular and OPEN OODBMS”を開発することになっている。

2.3 OODBMS の DDL の標準化へのアプローチ (A.E. Wade, Objectivity 社)

OODBMSの標準化にあたっては、最小限、DDLとDMLの標準化が必要であり、ここでは前者のDDLの標準化に関する事項について述べている。OODBMSオブジェクトレベルのgranularity、導出グラフ管理、のDDLを構築するには、「専用の定義言語の開発(STEP/Expressなど)」と「既存のプログラミング言語の枠内での拡張」の2つのアプローチがある

が、ここでは、後者のアプローチをC++をベースに行なった場合に必要となる機能として特にポータブルなプリミティブ(long,shortといったマシン依存のデータ型ではなくint32等を用意する)、アソシエーション(リンク/関連のことで、対応関係やその方向の類別及びバージョンに関する振舞いの記述)、可変サイズデータ(パラメタライズド・クラス)、aggregateなどのクラスのサポートが重要としている。

2.4 OODB におけるトランザクション処理のための Strawman 参照モデル (C.War TI社)

トランザクション処理方式として、従来のserializable executionに加えて、non-serializable execution(従来のread/write conflictに関する直列性を問題にしないやり方)が重要であるとし、長期間(long-duration)トランザクションを含め、トランザクション処理に関する基本機能を27項目挙げ、10のOODBMS(Avalon C++, GemStone, Objectivity/DB, ObjectStore, Ontos, ORION, O₂, Postgres, VERSANT, Zeitgeist)に対して対照比較を行なっている。

2.5 ODBMS におけるトランザクションとバージョンング (M.Loomis, Versant Object Technology 社)

主に、Versant ODBMSにおけるトランザクション処理及びこれに関連するバージョンング方式について述べている。トランザクションモデルはshort及びlongの2種のトランザクションをサポートする。longトランザクションはいくつかのセッション(session)の系列であり、各セッションはいくつかのshortトランザクションの系列からなる。longトランザクションに対する操作として、“long lock”があり、これが“short lock”と異なる点は、persistentでオブジェクトレベルのgranularityをサポートしている点などである。ユーザは、各shortトランザクション中で個人(private)DBへのcheckoutや、共有(public)DBへのcheckinなどを行なう。バージョン管理は、プリミティブの同定、オブ参照メカニズム、状態通知、分散DB上でのサポートなどが必要になるとし、これに基づき、これらのための操作群を同定している。トランザクション/

バージョン管理機能の実現方法については、言語の拡張による方法及びクラスライブラリによる実現という2つのアプローチが考えられるが、Versantでは後者のアプローチを採用している。

2.6 CAD Framework Initiative (CFI) の概要 (R.Spickelmier, Objectivity 社)

CADの種々のツールの共存、および相互運用可能性(interoperability)を可能とするような、Design Automationのフレームワークの構築を目標として1988年2月に設立された組織がCFIである。現在モトローラ社のA.Graham氏をpresidentとし、日米欧約20社(WSメーカ、CADメーカなど)が参加している。

アーキテクチャ、設計方法論、設計データ管理、設計表現、ツール間通信、ユーザインタフェース、システム環境に関する技術委員会があり、1990年6月に開催されたDA Conferenceで、CFIのユーザ・ゴール・目的に関する文書を発表している。

特に、CFI Storage managerに関するグループは、ユーザ(AT&T、モトローラ、TI、UCB)と主なODBMSベンダから構成されており、3つの文書を作成済または作成中である。これは、(1)ゴールと要件に関する文書(配布中)、(2)機能仕様書(作業用ドラフト)、(3)インタフェース仕様書(1991年のDA Conferenceで発表予定)である。基本概念は、構造的なオブジェクトであるdata entity、リンク、composite object、コンテナ、language binding(C,C++)、オブジェクト指向モデリングである。

2.7 その他の発表の概要

- Primitives for Schema Updates in an Object Oriented Database System (R.Zicari)
OODBのスキーマ更新のためのプリミティブな操作の提案
- The Need for a DML: Why a library interface isn't enough.(J.Orenstein)
Associative retrieval機能を実現するために、いかなる言語にも埋め込めるDBLを作るとを主張。Embedded DMLアプローチの重要性を指摘。
- Foundations for Object-Oriented Query Processing (K.Davis)

クラスをオブジェクトの集合と見なした上で、これらを実行できるクラス代数を提案。

- Inheritance and Generalization in Intelligent SQL (S.Khoshafian)
既存のRDB言語SQLをオブジェクト指向風に拡張し、汎化/特化や多重継承などが扱えるSQLを提案。SQL3とは異なる。
- Object Databases as Generalizations of Relational Databases (Cetin Ozbutun)
RDBを順次拡張し、ODBMSに進化させる方法を提案。外来キーにもとづくオブジェクト参照機能、集合値の導入、関数定義機能、オブジェクト名としてのホスト変数、再帰質問など。
- Ada SQL Interface (T.J.Wheeler)
AdaプログラムとSQLデータベースとの間のインタフェースを抽象的インタフェース、Adaビュー、SQLビューを用いて実現する方法について述べている。

3 大学におけるOODB研究

3.1 Brown University

S.ZdonikのグループではOODBMS Encoreなどの研究をおこなっている。Encoreはバイトストリームに対するデータベースであるObserverとその上の型システムとしての役割を果たすEncoreの二つからなる。

Observerはマルチユーザーに対する並行制御(ロックに基づくトランザクション)や、アプリケーションとの間のデータのキャッシング、記憶管理などを行うサーバクライアント型のデータベースカーネルである。汎用のデータベースであり、Encoreだけを対象としたものではない。ちなみにIntermediaはobserverに基づいている。ロックは協調的な作業のためには、直列可能性の条件が制限が強すぎるので、柔軟なロックメカニズムを採用している。優れているのはそのプリフェッチメカニズムであり、クラスタの中のオブジェクトをアクセスするとクラスタ全体がクライアント上に展開される。現在、ニューラルネットを用いたクラスタリングの研究も行なっている。

Encoreは最終的な目標はOODBPLであるが、現在はその半分しか達成できていない。C言語の上にObject指向の要素を加えており、プリプロセッサを通して処理される。C++に永続性を与えるアプローチをとらなかったのは、現在のC++は振る

舞いの記述に重点があり、データベース応用では振る舞いのモデルよりも、データのモデルの方がより重要であると考えている。この意味で、Encoreは一般のOOPと比べるとより厳密な型の概念が特徴である。継承には、コードの再利用と構造の継承という2つの意味がある。また、型とクラスは別物であり、クラスはインスタンスの集合を指す。型階層は、reflectiveな構造になっており、type自身もオブジェクトである。join, viewを型システムに組み込むことが今後の作業。EncoreのGUI(X + Interview)を通じて型定義やメソッド定義をインタラクティブに行うことができる。クラスは動的に組み込むことができ、メソッドの組み込みには別の教授が作ったダイナミックローダを使っている。

現在の研究テーマとして、他に、CSCWを意識したCooperative transaction、メソッドを含むintegrity constraint、navigational accessの高速化などがある。

3.2 MIT Intelligent Engineering Systems Lab.

MITのD.Srinam助教授らのグループは、OODBを核としたCSCW環境DICE(Distributed and Integrated Computer-aided Engineering)の研究を行っており[7]、その一環として、GemStoneを用いた黑板モデル型のシステムMagpieBridgeを作っている。建築の意匠設計者や構造技術者がグループで設計作業を行なうためのもので、設計物の制約チェック、版管理、ネットワーク上での対話などが可能となっている。C++ベースのOODBMSを用いなかったのは、スキーマ設計に関してこれらのものはrigidすぎるという点を強調している。

4 OODB ベンダーの動向

4.1 Servio Corporation

現在、同社のGemStone(R2.0)には、Database, Tools, Language Interfaces, DataBridges, Portsの五つの製品群がある。Databaseは、データベースの核に当たる部分。この部分はほとんど言語独立であり、多言語対応にしようとしている(C++を含む)。Toolsとして、OODBのアプリケーション開発者のためのツールを揃えようとしている。とくに、C++のための完全な開発環境を揃えようとしている。schema designer, form toolkitなどはこの例であり、すべて、X window + Motifの環境

で使えるようになっている。また、今後、デバッガなども揃える予定であるという。また、アプリケーションインターフェースとして、smalltalk用のフォームベースツールFACETを用意している。Portに関しては、現在すでに、SUN以外の計算機、Sony News, Dec Station, Machintosh (Client Only)でGemStoneが動いている。しかし現在は、サーバーとクライアントは同種アーキテクチャの計算機でないといけな。Language Interfaceとしては、Small80 (R4.0)からGemStoneが利用でき、さらに、APIを通じて他言語もサポートしたいとのことである。C++は、R2.1(来年春の予定)の多重継承を待ってGemStoneにも組み込む予定。Data Bridge関連は、既存の関係データベース(Sybase)や知識ベース(NexpertObject)との互換性を保つため、データベースに質問を送り、その結果の関係をその場で、クラスとして取り込む仕掛をつくらせている。これには、RPCメカニズムが使われている。

GemStone 2.0は、バッファ・キャッシュ管理により以前より2.5倍以上の高速化、ノンストップ化(garbage collectionはメモリー上とディスク上の両方が並行して走るようになり、システムを止める必要がなくなった。)、C++インターフェースのサポート、ネストしたオブジェクトのアクセスの高速化、楽観的・悲観的並行制御の両方のサポートなどが特徴である。次の2.1版では、従来のものの3倍の高速化、複数のUnixファイルによるデータベースの実現、分散DB用の2相コミットのサポートなどが予定されている。さらに、schema designerでのメソッドの設計やグラフィカルな問い合わせ機能を付加する予定。

4.2 Object Design社のObjectStore

1988年にT.Atwoodにより設立された当社は、現在、Sun3/SPARCをプラットフォームとする、C++をベースとするOODBMS ObjectStoreを開発販売している。ObjectStoreの特徴は、(1)型に独立な永続性付与、(2)高性能なページフォールト・アーキテクチャ、(3)パラメタライズドタイプのサポート(AT&T C++ 3.0に使用される予定)、(4)最適化機能を持つ質問言語、(5)C++用の種々の開発ツール(ハイパーテキスト的なスキーマデザイナー・ブラウザなど)である。特に、低コストでオブジェクト指向環境への移行が図れることを強調している。これについては、例えば、Unix上のPDSでドロー系の描画ソフトIdraw(Interview, C++)や、ViewLogic社のECADソフトViewLogic Schematic Ed-

itor (Cで33000行)が極めて容易に ObjectStore 上に移植できたことを例に挙げている。プログラムの書き直し箇所も極めて少なく(後者では約100行のみとのこと)、これによって、オブジェクトへの永続性の提供の他、マルチユーザ環境、版管理機能、質問機能などが利用できるようになる。

応用分野として、従来の OLTP に対して DMAP (Design, Modelling, Analysis & Prototyping) という応用形態を特に重要視している。これは、OLTP に比べ、分散指向で、少ないトランザクション数、大きな検索データサイズ、複雑なデータ構造、長期間トランザクション、データの高い再利用性、各データに対してさほど高くないアクセス頻度、などが要求される分野である。CAD/CAM/CIM などの技術分野への応用や OA 応用の他に、大規模スプレッドシートなどの MIS 応用が予想以上の伸びを示しているとのことである。過去には、CAD ベンダーは、ハード・OS・利用者インタフェース・DB をすべてカスタムメイドで作っていたが、1990年の時点では、DB 部分のみがかりうじてカスタムメイドで残っており、これらが汎用の ODBMS で置き換えられることを狙っている。

4.3 Ontologic 社の Ontos

C++ 対応の ODBMS Ontos は、この分野での先発組と言うこともあって、現在ユーザ数が 300 と多く、スキーマデザイナー (Ontos Schema Designer 1.0) やハイパーテキスト風フォームジェネレータ Studio などのツールの開発が盛んに行なわれている。Schema Designer は、(1) ユーザ制御によるクラス階層の一部の可視化/不可視化、(2) ハイパーテキスト風の C++ 型定義情報のブラウジングや実行、(3) インスタンスの内容の表示・交信、(4) クラスの動的な追加などが行なえる。Studio は、オブジェクトをフォーム形式で表示するためのツールで、例えば、オブジェクトを値に持つ属性は自動的にその値表示のボタンになる。また、Query By Form で、質問をフォームに付けることもできる。

応用としては、CIM 関係では、General Dynamics 社が社内利用中の Integrated BOM System がある。部品アセンブリ情報の DB で、2人で8ヶ月の開発期間を要したという (Studio を利用)。効率が RDB より 100 倍以上高速という。また、地図情報システムとしては、オーストラリア国土地理院 (CSIRO) が開発したものがある。地図の種々の要素がすべてオブジェクトとして扱われており、Ontos 導入からプロトタイプ開発終了までに5ヶ月し

か要しなかったという。OEM としては、OS/2 上で動く Ontos を用いたビデオ情報システムを Fluent 社が開発販売を行なっている。

4.4 Itasca 社の Itasca

MCC の ORION プロジェクトの成果を製品化する目的で CDC が中心になって 1989 年に設立され、現在、分散型 ODMS Itasca を販売している。MCC の OODS グループは現在も継続中で、主に OODB のためのツール開発や共有型の heterogenous DB の研究を行なっている (リーダーは Bellcore の Phil Cannata 氏)。Itasca 社からは Itasca システムと共に、研究員も派遣している。ORION に比べて現在の Itasca は性能にして 10 倍以上の向上が図られたという。

Itasca の特徴は、(1) 組型オブジェクトに対する *composite object* のサポート、(2) スキーマ変更を含む全ての DB 操作はトランザクションとして扱う、(3) 個人/共有 DB に対するチェックイン/チェックアウト機能、(4) 版管理 (版の生成はクラス単位で宣言)、(5) メソッド変更時にも応用プログラムの再コンパイルや再リンクが不要 (メソッドも DB 中に格納)、(6) クラス階層木全体に対する B+ 木索引、などである。

応用分野としては、技術情報システム、マルチメディア、CASE、地理情報、AI などが重要としている。Concept modeller 関係で Wisdom 社が Itasca を利用予定とのことである。

4.5 Versant Object Technology 社

現在同社は、C++ および C 応用のプラットフォームとしての ODBMS Versant を開発販売している。今後の予定として、(1) Smalltalk-80 のサポート、(2) O-SQL 質問言語のサポート、(3) RDB (Oracle, DBC, DB2 など) との gateway およびこれらにまたがる分散トランザクション実現のための Versant Star の開発 (1991 年出荷予定)、などを予定している。(1) については、MIS 関係の顧客からの需要が増加していることによるという。(3) については、同社が、他 DBMS との共存を重要視していることによる。Versant の開発では、既存の RDBMS の拡張や、OOPL の下に storage manager を開発するというアプローチを採用していない。これは、前者は十分な効率が得られないこと、後者では、マルチユーザ機能などの追加が困難という理由による。

CIM 関係の応用事例として米国海軍の CIM DB である RAMP (Rapid Aquisition of Maintenance

Parts) プログラムがある。CAD のデータ交換標準案 PDES/STEP モデルに基づく機械部品アセンブリおよび配線基板アセンブリのデータベースであり、前者は Oracle, 後者は Versant を用いている。Oracle に対して 100 倍以上の性能が得られているという。(ECAD の配線図オブジェクト (2000 個の部品オブジェクト) の読みだしで 100 倍早いデモがある。Oracle では 5 つの表の join に相当。) PDES/STEP モデルは、concurrent engineering をサポートするプロダクト/プロセスのモデルで、CIM のための効率化・統合化を図ることを目的としている。このモデル自身、ODBMS モデルとの一致性が極めて高いとのことである。

5 あとがき

以上述べてきたことから、OODBMS は現在、第一期の成熟期に急速に向かいつつあると考えられる。実際、各ベンダーは、市場の要求と現在までのオブジェクト指向技術の粋を集めて、完成度の高い商品を出そうとしており、OODBMS 本体の上に乗せるスキーマ設計ツールや 4GL 的なものが充実しつつある。また、ユーザによる応用開発も着実に進みつつある。このため、ベンダーとしてはすでに標準化に気持ちが向かっているとの印象を持った。また、MIS 応用には Smalltalk-80 がよいとの声を 2,3 のベンダーから聞かれることも印象的である。おそらく、これらの製品が落ちつこうとする方向は以下のようなものになるだろう。(1) persistent object storage としての ODBMS、(2) 問い合わせには SQL を用いる、(3) fancy なクラスブラウザ、メソッドの編集機能などのサポート、(4) ベースとなる言語は C++ が主流、(5) heterogeneous な環境でのサーバークライアント構成、(6) 他の DBMS との共存や緩やかな移行を強く意識、(7) schema evolution、意味レベルの version mechanism、ビュー、定型的な関連 (relationship) 以外の制約はあまり扱わない、(8) 並行制御に関しては悲観的方法および長期間トランザクションに基づく楽観的方法をとる。

参考文献

- [1] Cattell, R.G.G., *Object Data management*, Tutorial material, ECOOP/OOPSLA'90, October 1990.
- [2] Cattell, R.G.G. and Skeen, J., *Engineering Database Benchmark*, Tech. report, Sun Microsystems, April 1990.
- [3] Loomis, M.E.S., *Moving to an OODBMS*, Tutorial material, ECOOP/OOPSLA'90, October 1990.
- [4] Lerner, B.S. and A.N. Habermann, *Beyond Schema Evolution to Database Reorganization*, Proc. of ECOOP/OOPSLA'90, pp.67-76, October 1990.
- [5] Straube, D.D. and Ozsu, M.T., *Type Consistency of Queries in an Object-Oriented Database System*, Proc. of ECOOP/OOPSLA'90, pp.224-233, October 1990.
- [6] SPARC/DBSSG/OODBTG, *Proceedings of OODBTG Workshop*, October 1990.
- [7] Srinam, D. et al., *An Object-Oriented Framework for Collaborative Engineering Design*, To be published in Computer-Aided Cooperative Product Development, April 1990.

次世代データベース SIM

(Semantic Information Manager)

平成三年二月二日
日本ユニシス株式会社
システム技術本部
足立 哲夫

講演の構成

- データモデルの発展
- S I Mモデル

データモデル

- 知識を定式化する「思考道具」

- 3つの構成要素
 - － 対 象

 - － 操 作

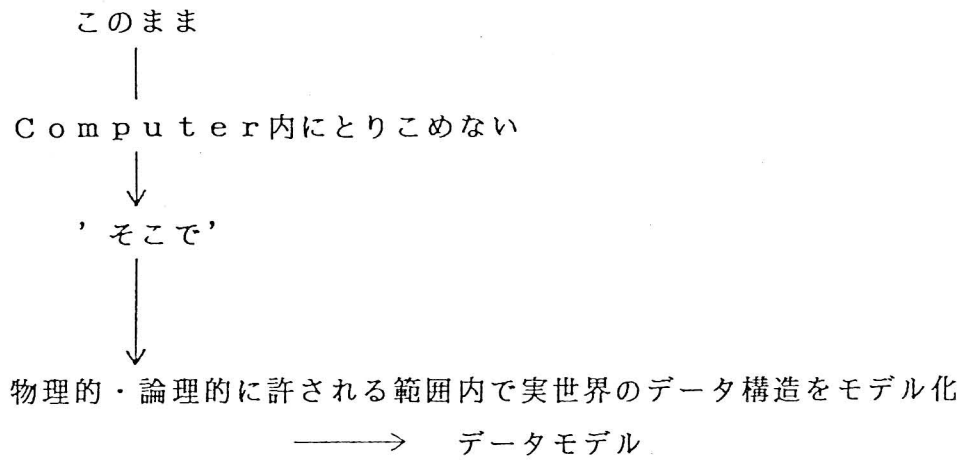
 - － 制 約

- ビジネス環境の事実をデータベースの対象に変換する方法

データベース化したい対象世界

例えば、企業の人事情報、営業活動とかである。

R e a l W o r l d



「利用者から見て、どのようなdataが、どのような関係を持ってデータベース上に存在しているかを示す」

伝統的なデータモデル

- ファイル・システム
 - － 高いパフォーマンス
 - － 整合性制約は許されない
 - － 柔軟性の欠如

- 階層型、または、ネットワーク型
 - － 高いパフォーマンス
 - － 限定された整合性制約
 - － 限定された柔軟性

リレーショナル・データモデル

■ 特 長

- － 高い柔軟性
- － 単純な関係は容易
- － パフォーマンスを向上させるために多くのリソースが必要

過去のデータベースの問題

- 意味制約条件を表現する能力の欠如
 - － 複合値問題
 - － 独立したデータ間の関係づけ
 - － 完全性の保証

例 複合値問題

複数の値を持つ項目をどうやって表現するか

W [著者, 書名, 出版社, 発行年月日, 分野]

— 著者が複数いる場合

著者フィールドの値だけ異なって残りが同じレコードを著者分だけ作る。

— 分野が複数に渡る場合

分野フィールドの値だけ異なって残りが同じレコードを分野分だけ作る

■ 冗長性

前提条件 「同一書名の本が同一出版社から同じ年にでない」を設定

W_x [著者, 書名, 出版社, 発行年月日]

W_y [書名, 出版社, 発行年月日, 分野]

と、分解すると冗長性は回避される（第四正規化分解）

もとの世界を $W = W_x + W_y$ と考えるのは不自然

著者 (A氏, B氏, C氏), 書名 (SIM), 出版社 (ユニシス),

発行年月日 (1988), 分野 (コンピュータサイエンス, ビジネス)

— 属性, () 内は属性値

複合値問題

複数の値を持つ項目をどうやって表現するか

書名	著者名	出版社	発行年月	分野
データベース入門	A氏、T氏	ユニシス社	1988年4月	コンピュータサイエンス、 ビジネス
InfoExec	A氏	ユニシス社	1988年9月	コンピュータサイエンス

重複した値を持つレコードを複数作る

書名	著者名	出版社	発行年月	分野
データベース入門	A氏	ユニシス社	1988年4月	コンピュータサイエンス
データベース入門	T氏	ユニシス社	1988年4月	コンピュータサイエンス
データベース入門	A氏	ユニシス社	1988年4月	ビジネス

⋮

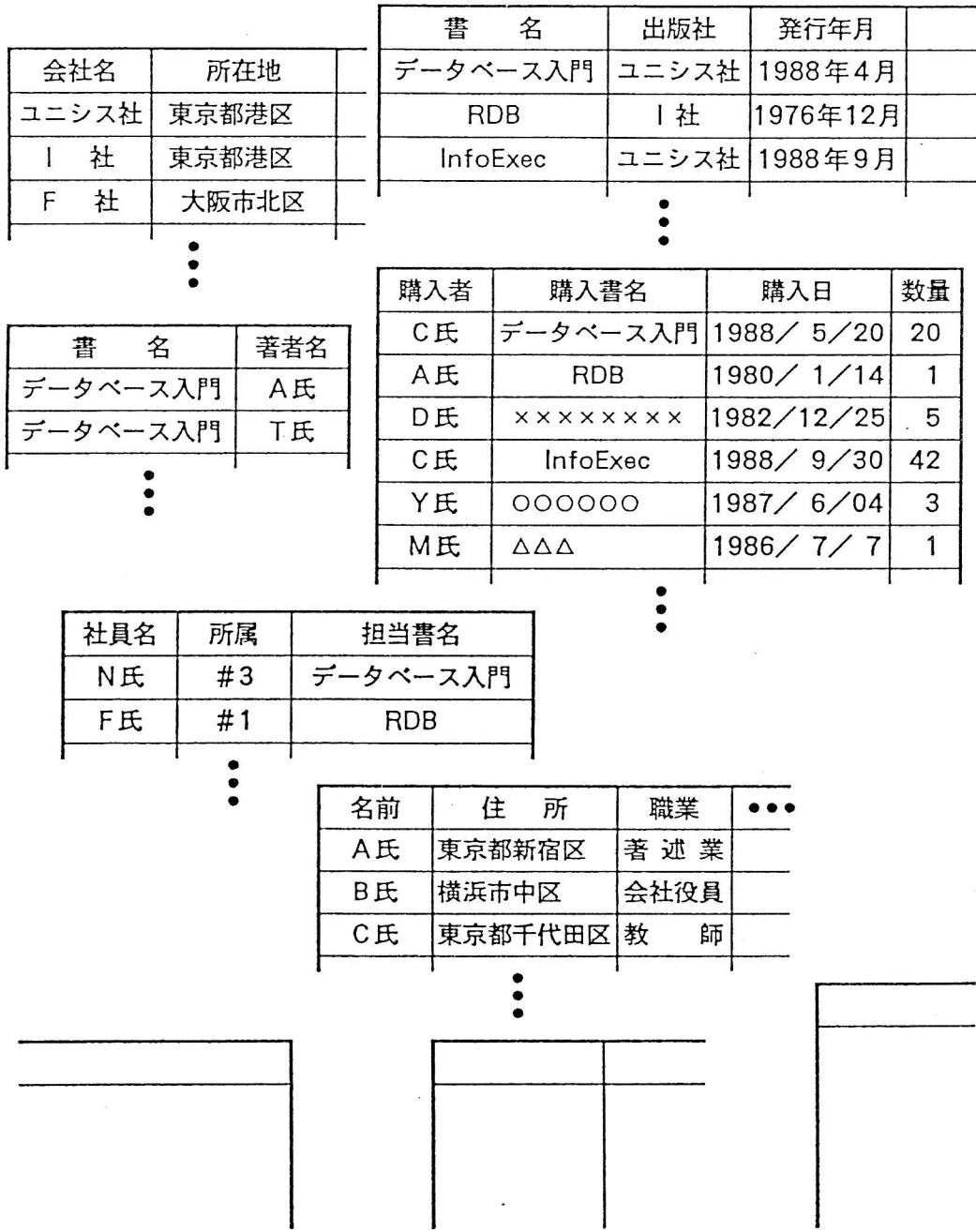
独立したいくつかの構造にわけると

書名	出版社	発行年月
データベース入門	ユニシス社	1988年4月
InfoExec	ユニシス社	1988年9月

書名	著者名
データベース入門	A氏
データベース入門	T氏
InfoExec	A氏

書名	分野
データベース入門	コンピュータサイエンス
データベース入門	ビジネス
InfoExec	コンピュータサイエンス

独立したデータ間の関係と完全性の保証

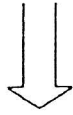


過去のデータベースの問題

「実世界の情報構造を正確に書き下す」

「実世界を表現する機構が使用者がデータを扱う時の

自然な思考過程と近ければ近いほど良い」



”回避するため”

専門的技術

適用業務プログラム作成時に考慮

人工的構造

■ 実世界のデータを関数従属関係でまとめ、正規化する過程で

- データ及びデータ間の意味
- データ及びデータ間の制約条件

が、構造的に表現する方法がない。

次世代データベース誕生の背景

■ 既存のデータベースにおける表現能力の欠如

正規化によって、複合オブジェクトは存在しない
関数従属による人工的なキー属性の設定
表間の関連は表現できない
関数従属関係以外の関係を表示できない

■ 概念データベースと論理データベースのギャップを埋める

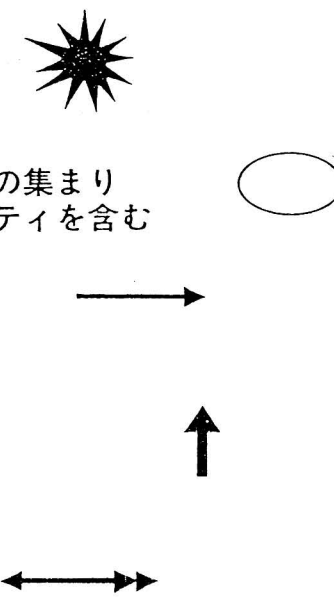
自然な表現能力を持ったデータベースの要求

セマンティックモデル誕生の背景

- 物理設計と論理設計の分離
 - － 3層スキーマにおける概念スキーマ
- 論理設計のための道具
- 既存データモデルに対する研究成果
- 実世界と抽象化し、より正確に表現する方法

セマンティック モデル

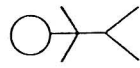
S I M モデル

- エンティティ
 - 実世界の興味の対象
 - レコードまたは行ではない
 - サロゲートを持つ
 - クラス
 - 共通の特徴を持つエンティティの集まり
 - クラスには表示可能なエンティティを含む
 - 属性
 - エンティティの特徴
 - 値としてエンティティも持てる
 - 単一値または複数值を持てる
 - サブクラス
 - スーパクラスの部分集合
 - スーパクラスの属性の継承
 - 複数のスーパクラスを持てる
 - 整合性
 - 汎化階層構造
 - 参照整合性制約
 - 属性の整合性制約
 - 汎用的な整合性制約
- 

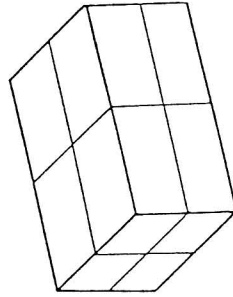
SIMモデル (セマンティック モデル)

◎エンティティ (Entity)

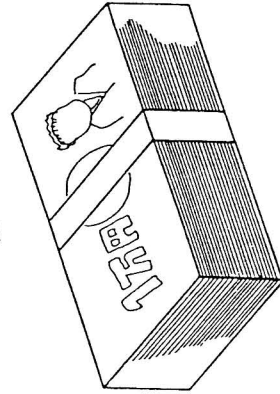
- 実世界をデータベース化する時に対象となるもの



人



物



金

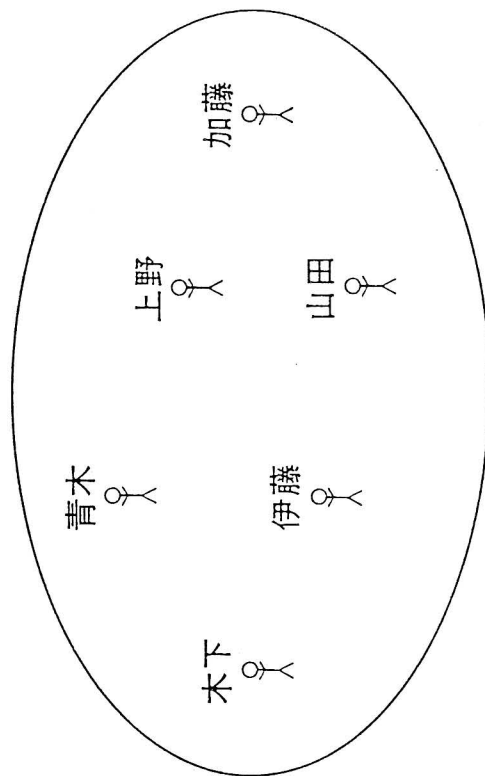
S I M モ デ ル

- クラス (Class)
- 同一タイプのエンティティの集合

(例) <クラス>

従業員：すべての従業員の集まり

[従業員]



S I M モ デ ル

● サブクラス (Subclass)

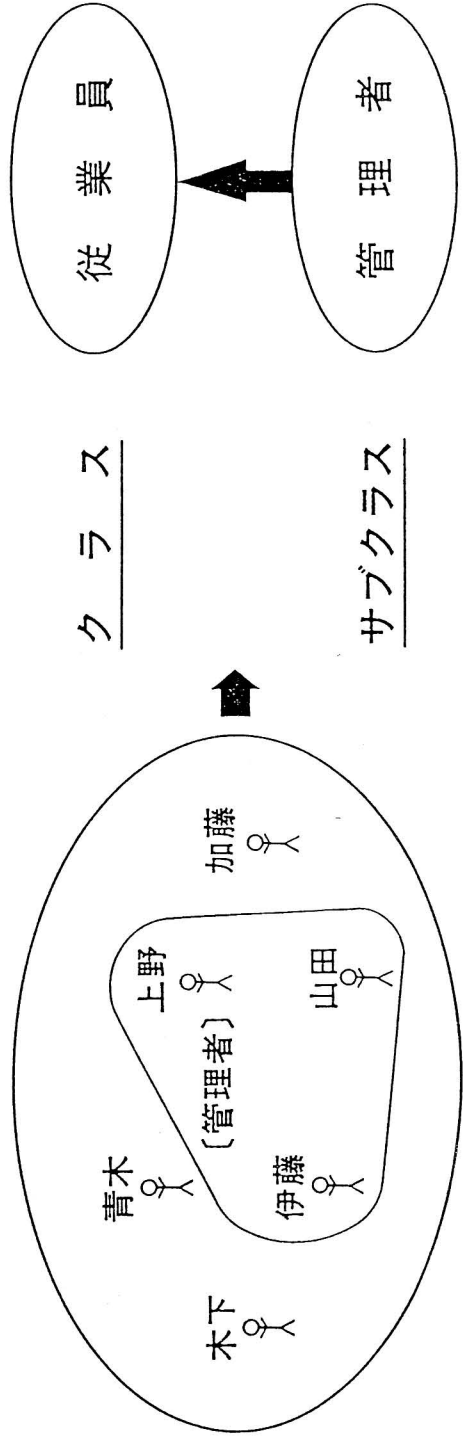
■ クラスのエンティティの部分集合

■ 汎化階層構造と呼ばれるデータ構造を表現する

(例) <サブクラス>

管 理 者 : 全従業員の内, 管理者である従業員のみの集合

[従業員]



SIMモデル

◎ 属性 (Attribute)

- エンティティの特徴を構成する

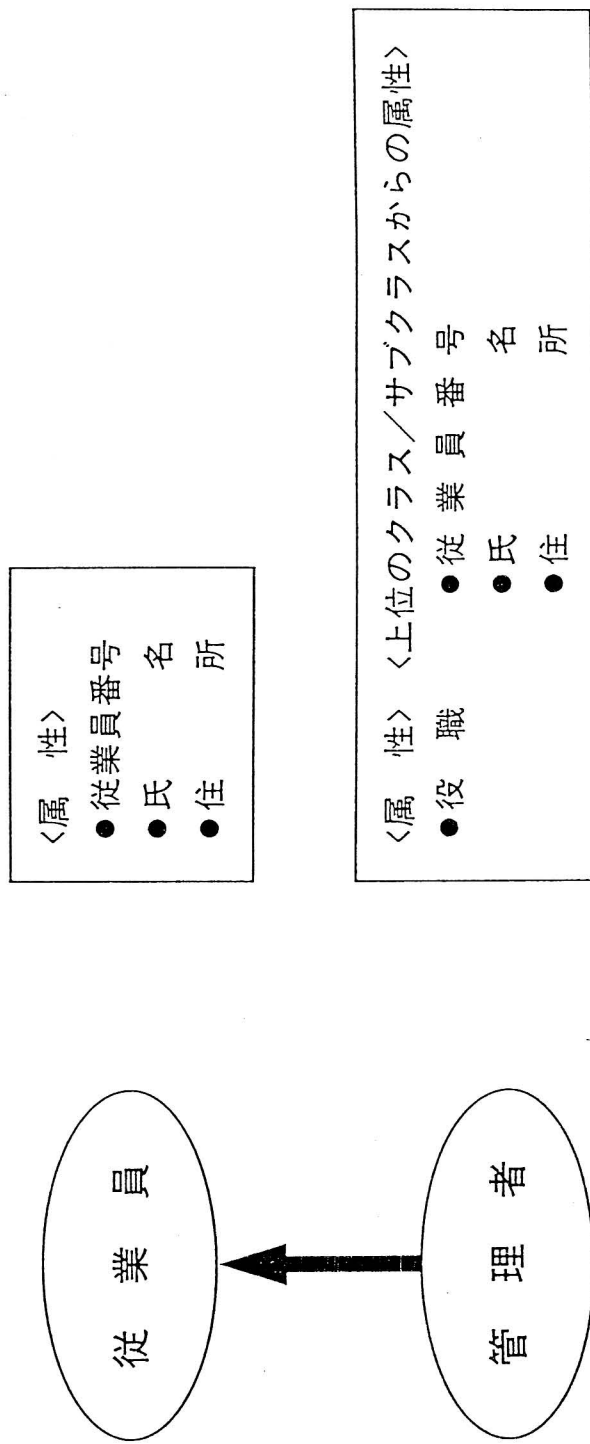
(例) 従業員の属性

従業員	従業員番号	氏名	住所
従業員	従業員番号	氏名	住所
従業員	従業員番号	氏名	住所

S I M モ デ ル

- 継承属性 (Inherited Attribute)
- サブクラスは上位のクラス又はサブクラスの属性を継承する

(例)

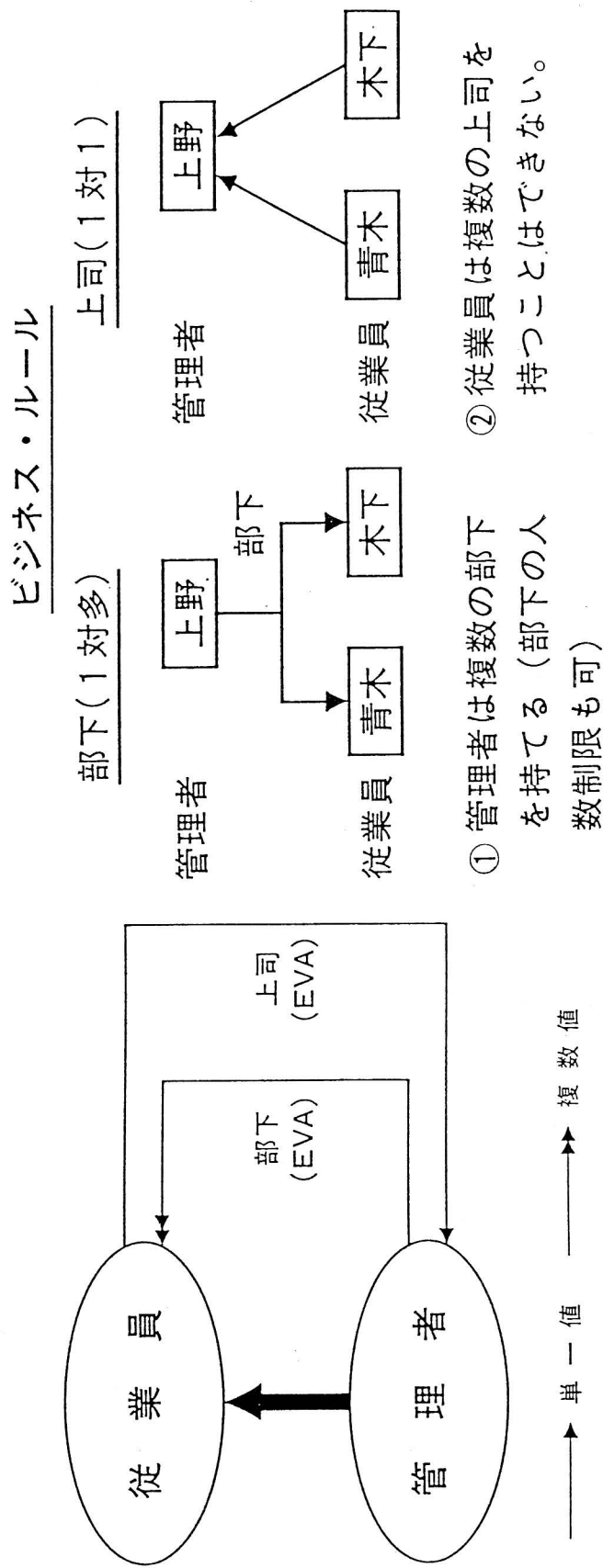


- 管理者の従業員番号, 氏名, 住所として利用できる
- 従業員でない人は管理者でないという事実が表現されている

S I M モ デ ル

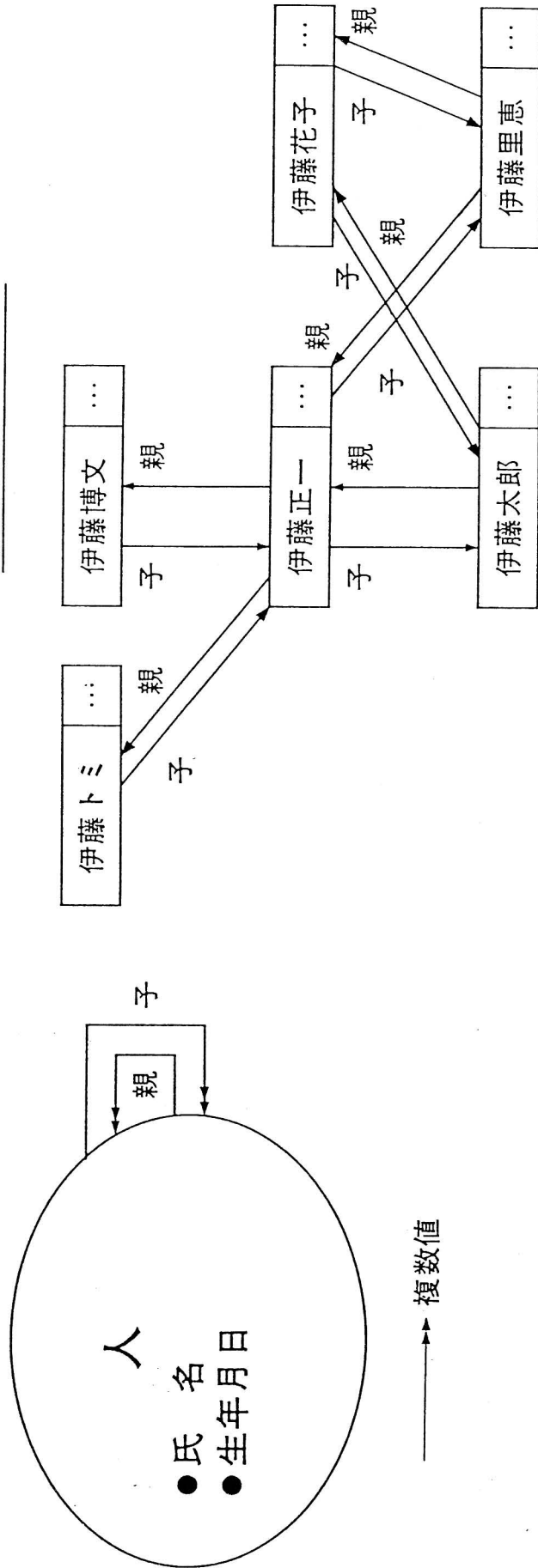
- データ値属性 (DVA) とエンティティ値属性 (EVA)
- データ値属性：従業員番号, 氏名, 住所, 役職などの表示することのできる (Data Valued Attribute)
- エンティティ値属性：他のクラスのエンティティや, 同一クラス内のエンティティ (Entity Valued Attribute) を指す属性。エンティティ間の関連を表現する

(例)



S I M モ デ ル

(例) エンティティ値属性 (多対多)

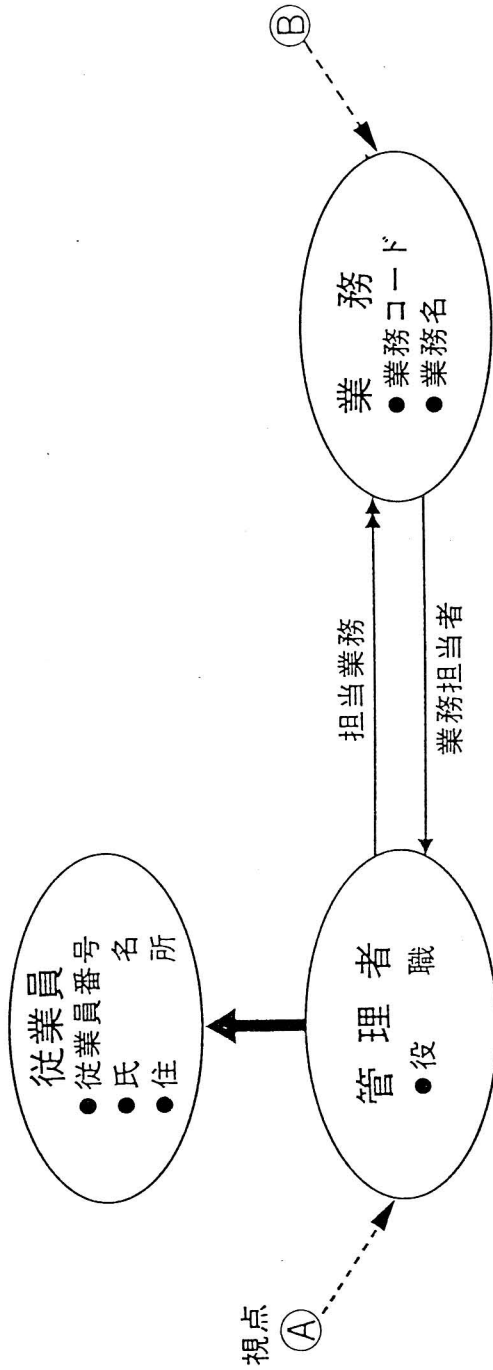


S I M モ デ ル

● 拡張属性 (Extended Attribute)

- エンティティ値属性に関連づけられているエンティティの属性は、拡張属性として参照できる

(例)



<視点クラス>

- ① 管理者：担当業務の業務名 ➡ <業務名> ○ F <担当業務>
- ② 業務：業務担当者の管理者の氏名 ➡ <氏名> ○ F <業務担当者>

S I M データモデル・データ操作

SIMデータモデル-DML

SIM Data Model-DML (1)

■ 直接属性

- クラスで宣言されている属性

(例) 人間の年齢
人間の配偶者

■ 継承属性

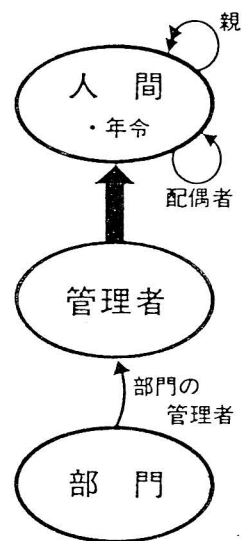
- スーパクラスから継承された属性

(例) 管理者の年齢
管理者の配偶者

■ 拡張属性

- 別の属性によって修飾される属性

(例) 人間の配偶者の年齢
部門の管理者の配偶者の親の年齢

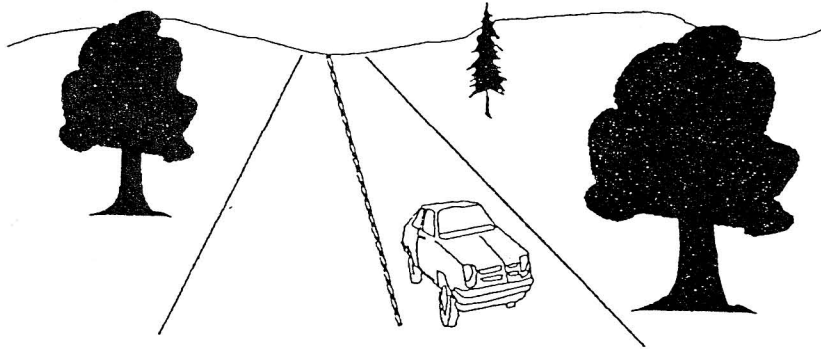


SIMデータモデル - DML

SIM Data Model - DML (2)

■ 視 点

- 問合せのための視点を選ぶ
- 与えられた視点から、データベースの他の部分は拡張属性として参照できる
- 経路の表現



SIMデータモデル - DML

SIM Data Model - DML (3)

- その他

- データ型(数値, 文字, 日付, 時間, その他)
- 集約関数(COUNT, MIN, MAX, AVG, その他)
- 修飾(SOME, ALL, NONE)
- 推移関数
- 空値の取り扱い



- 機密保護

- エンティティ制御 : アクセス
- エンティティ操作制御 : 許可権



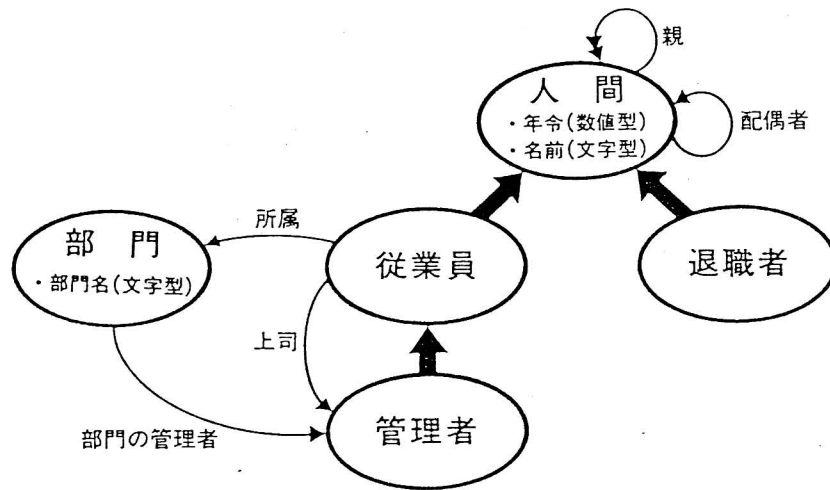
検 索 例

Examples

- RETRIEVE 名前 OF 学生
- FROM 人間
RETRIEVE 名前, 名前 OF 管理者 OF 人間 AS 従業員
WHERE 年齢 OF 人間 < 21
- INSERT 従業員
(名前 := "アラン ベル",
上司 := 管理者
WITH (名前 = "シドニィ スミス"))
- MODIFY 臨時管理者
(給与 := 給与 * 1.65)
WHERE 部門番号 OF 管理部門 = 1138
- DELETE 臨時管理者
WHERE 名前 = "ジョン ジェフリー"

簡単な検索例

Simple Query

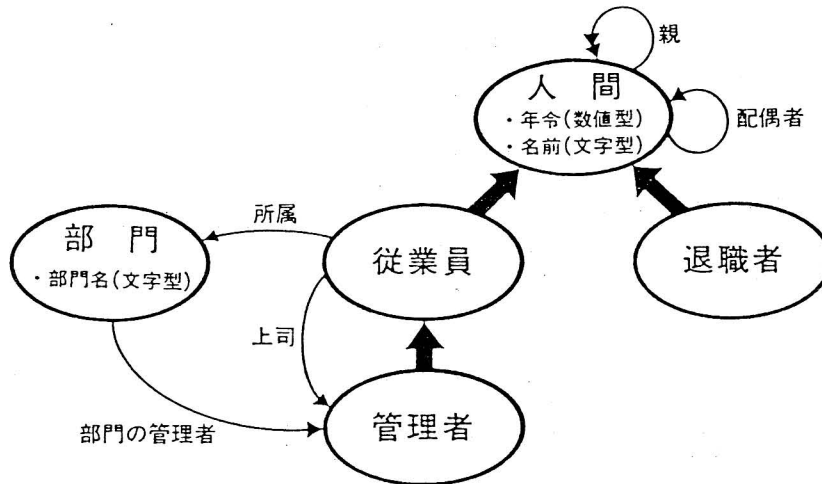


社内で一番若い管理者の名前を検索する

FROM 管理者 RETRIEVE 名前 WHERE 年齢 = MIN (年齢 OF 管理者)

複雑な検索例

Very Complex Query

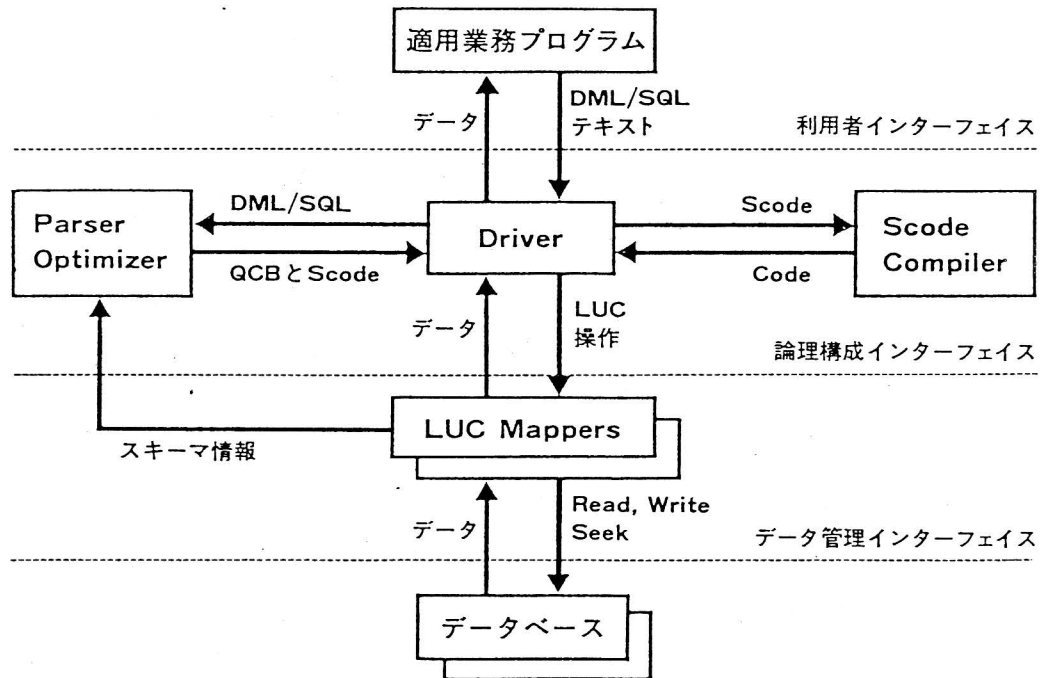


従業員の配偶者が退職者である従業員名と、その従業員が所属している部門名と、その部門の管理者の名前を部門に視点を置いて検索する。

```
FROM 部門 RETRIEVE 部門名, 名前 OF 部門の管理者,  
                        名前 OF INVERSE (所属)  
WHERE 人間 OF 配偶者 OF INVERSE (所属) ISA 退職者
```


SIM/SQLアーキテクチャ

SIM/SQL Architecture



第17回日本MUMPS学術大会
特別講演「オブジェクト指向プログラミング、データベース」
資料集
平成3年2月1日

編集 第17回日本MUMPS学術大会
大会長 河村徹郎

発行 日本MUMPSユーザーズグループ

日本MUG事務局
〒606
京都市左京区聖護院川原町54
京都大学医学部附属病院
医療情報部内