

静岡大学 博士論文

集中・分散・移動環境における  
データベースのアクセスと管理  
に関する研究

平成9年7月

大学院電子科学研究科  
電子応用工学専攻

和田雄次

## 論文要旨

データベース(DB)技術の研究は主にソフトウェアの世界で進歩してきた。しかしながら、ハードウェアやOSなどのプラットフォームに代表される計算環境の発展を見ずしてはその歴史は語れない。その歴史を見てみると、約20年程前には大型汎用計算機とTSS端末の組み合わせによる集中計算環境が全盛を極めた。そして、その後UNIXワークステーションが登場し、クライアント/サーバ(C/S)構成がオープンシステムのプラットフォームとなり、分散計算環境が実用化された。しかし、それも束の間10年位前に小型高性能化を実現したパソコンが登場した。更に、最近ではMS-Windowsが業界標準としての地位を不動のものとし、今後も更なる発展が予想される。この結果、携帯端末の普及とインターネット等の通信インフラの整備により移動計算環境が実用化水準に達してきた。こうした計算環境の発展の過程で、筆者は各々の計算環境において要求されるDB技術を整理し、応用ソフトウェア、利用者インタフェース(UIF)、基本ソフトウェア、そしてツールの面からDB技術を研究してきた。以下に、各々の研究課題とその成果を記述していく。

集中計算環境において利用者が要求するDB技術としては数多くあるが、筆者はDB処理の高速化技術やDB設計の自動化技術に焦点を絞って研究を実施した。

DB処理の高速化技術は従来から多種多様な試みがなされてきたが、筆者は医療情報DBとして著名なMUMPS DB(木構造)に注目し、ソフトウェアで高速化を実現する技術としてDBのバッファ管理方式の研究を試みた。この結果、従来のLRU法に代わる新たな方式として木構造の各層に優先順位を考慮したBPL法を提案し、DBアクセス時のバッファ不在頻度を最小化することに成功した。

一方、DB設計の自動化技術も従来から研究が盛んに行われてきており、設計方法論の研究、また概念設計、論理設計、物理設計それぞれの設計段階での設計支援ツールの開発、さらには設計後のDB性能最適化なども研究されてきた。そこで、筆者は論理設計、物理設計、スキーマ性能評価の各設計フェースを統合化し、かつ対話的なUIFを提供した画面指向型DB設計・評価ツールDMSDBDAを研究試作した。この結果、DBを実際に開発する前の性能やコストの見積もり、構築済みDBの再設計や性能改善が可能となった。

分散計算環境において利用者から要求されたDB技術としては集中計算環境と共通して

いるものが多いが、筆者はC/S環境における使い易いDBのUIF、企業の経営者が行う意思決定のような高度な業務への適用などを研究の対象とした。

従来のDBのUIFは、SQL言語に代表される問合せコマンドをTSS端末キーボードから入力していた。そこで筆者は新たに視覚的な概念によるDBのUIFを提案し、アイコン、マルチウインドウなどをベースにしたオブジェクト先行型UIF MELQUERYをUNIXワークステーション上に研究試作した。続いて、このMELQUERYをUNIXワークステーションとDBサーバを連結させたC/S環境に拡張したMELQUERY/Rを研究開発した。この結果、分散計算環境でも集中計算環境でも操作が統一化された位置透過なエンドユーザコンピューティングの概念を実現した。

一方、意思決定業務への適用はDBの技術が登場して以来の永遠の課題であるが、その研究は意思決定そのものではなくその支援技術から始まり、現在はAI技術を利用した意思決定そのものの研究も行われている。筆者は先ず意思決定支援のためのツールDSSの研究開発を手掛けた。DSSは問題向きDSSと汎用的なDSSの二系統に大別されるが、筆者は問題向きDSSを包含した汎用型DSSの研究を目指し、EDPとOAの融合化、垂直分散型、統合化利用者視野を特長とするDSS-GOALを研究開発し、DBの意思決定業務への適用可能性を向上させた。その後、意思決定支援技術の新たな研究課題として、データマイニング技術に注目し、C/S環境におけるデータマイニングサーバの研究試作を現在継続している。

最近の携帯端末の低価格化、高性能化、高機能化によりその普及が急速に進んできたため、そのデータ通信機能を用いた移動計算環境でのDBアクセス要求も同時に高まってきた。しかし、移動計算環境でのDB技術に関する研究はまだ緒についたばかりである。例えば携帯端末利用者が移動しながらDBサーバをアクセスする際、どの無線通信セルの中にいるのか、あるいはどのセルからどのセルへ移動したのかを意識することなく効率良くDBアクセスすることが困難となる問題があった。そこで筆者は移動計算環境において携帯端末利用者が複数セル間を移動しながらDBサーバをアクセスする時、移動先セルでDBアクセス時のページ不在発生頻度を最小化するためのDBキャッシュ管理方式を提案し、計算機シミュレーションによりその有効性を示した。この管理方式の応用分野としては、先述したデータマイニングサーバや携帯型防災情報収集システムを考えている。

# 目 次

<b>第 1 章</b>	<b>序論</b>	<b>1</b>
1.1	研究の背景および目的	1
1.2	従来研究の概観	3
1.3	研究の成果と特徴	6
1.4	論文の構成	9
<b>第 2 章</b>	<b>関連研究</b>	<b>12</b>
2.1	緒言	12
2.2	データベースの利用者インタフェース	12
2.3	データベースを用いた意思決定支援技術	14
2.4	データベースの設計自動化支援・性能評価方式	15
2.5	移動体計算環境におけるデータベース管理方式	17
2.6	結言	19
<b>第 3 章</b>	<b>分散計算環境におけるデータベース の利用技術</b>	<b>21</b>
3.1	緒言	21
3.2	使い易いデータベース利用者インタフェース技術	24
3.3	OA業務および意思決定業務への応用	49
3.4	結言	73

<b>第4章</b>	<b>集中計算環境におけるデータベース システムの性能評価</b>	<b>75</b>
4.1	緒言	75
4.2	データベースの設計自動化支援・性能予測方式	76
4.3	データベース管理システムのバッファ管理方式	107
4.4	結言	134
<b>第5章</b>	<b>移動体計算環境におけるデータベース システムの性能評価</b>	<b>136</b>
5.1	緒言	136
5.2	システムの概念とモデル化	137
5.3	データベース・キャッシュ管理方式	143
5.4	計算機シミュレーションによる性能評価	150
5.5	性能評価結果の考察	156
5.6	結言	160
<b>第6章</b>	<b>結論</b>	<b>161</b>
<b>謝辞</b>		<b>166</b>
<b>参考文献</b>		<b>167</b>
<b>研究成果一覧</b>		<b>178</b>
<b>付録</b>		<b>180</b>

# 第1章 序論

## 1.1 研究の背景および目的

データベース技術の研究は主にソフトウェアの世界で進歩してきた。しかしながら、ハードウェアやOS<sup>注1</sup>などのプラットフォームに代表される計算環境の発展を見ずしてはその歴史は語れない。その歴史を見てみると、約20年程前には大型汎用計算機とTSS<sup>注2</sup> 端末の組み合わせによる集中計算環境が全盛を極めた。そして、その後UNIXワークステーションが登場し、クライアント/サーバ・アーキテクチャがオープンシステムのプラットフォームとなり、分散計算環境が実用化された。しかし、それも束の間10年位前に小型高性能化を実現したパソコンが登場した。更に、最近ではMS-Windows95<sup>注3</sup> が業界標準としての地位を不動のものとし、今後も低価格化、高機能化、高性能化のための更なる発展が予想される。この結果、携帯端末の普及とインターネット/イントラネットなどの通信インフラの整備により移動計算環境が実用化水準に達してきた。

一方、こうした計算環境の発展を基盤にして、データベース利用者の要求に応えるために必要なデータベース技術を各々の計算環境毎に整理してみると、基本的な要求は、日常業務へのデータベース適用の簡便性、データベースへのアクセス操作の簡潔性、アクセス応答の高速性、データベースの構築と保守の容易性と信頼性などである。こうした要求はいつの時代にも議論されてきたある意味では普遍的なものではあるが、上記の計算環境の進歩に沿ってこうした要求に応えようすると、実現提供すべきシーズは各計算環境毎に相違点がある。例えば、集中計算環境ではデータベースの構築と回復、信頼性、データベース集中アクセス処理の軽減、分散計算環境ではデータベースの位置透過性、配置最適化、通信量、分散更新と整合性などであり、移動体計算環境ではデータベース利用者の無線通信セル間の移動、データベース自体の移動、通信セル単位でのレプリカ管理、無線通信回線の信頼性、携帯端末の電源問題などである。従って、こうした各々の計算環境に固有なデータベース技術課題を解決することが研究の対象となる。そこで、筆者は、各々の計算環境におけるデータベース要求を基にして、各々の計算環境に最適なデータベース応用ソフトウェア（例えば、OA<sup>注4</sup> 業務や意思決定業務など）、データベース利用者インタフェース、データベース基本ソフトウェア（例えば、データベース管理システム）、そしてデータベース設計評価ツールを追求し、実現するために、研究試作や検証実験を通じて、集中計算環境、分散計算環境、そして移動体計算環境のためのデータベース技術、とりわけデータベースのアクセスと管理の問題を解決することを本論文の目的とする。

以下、集中計算環境、分散計算環境、そして移動体計算環境毎のデータベースのアクセ

---

<sup>注1</sup> Operating System

<sup>注2</sup> Time Sharing System

<sup>注3</sup> 米国Microsoft Corp.の商品名

<sup>注4</sup> Office Automation

スと管理に関する研究の背景と目的を記述する。

**集中環境** 集中計算環境において利用者（ここでは、データベース応用システム設計者を含めた広義のデータベース利用者を意味する。）が要求するデータベース技術としては数多くあるが、筆者はデータベース処理の高速化技術やデータベース設計の自動化技術に焦点を絞って研究を実施することを目的とする。

データベース処理の高速化技術は従来から多種多様な試みがなされてきたが、筆者は医療情報データベースとして著名なMUMPS<sup>25</sup> データベース（木構造）に注目し、ハードウェアをいったん固定化し、ソフトウェアの世界の中で高速化を実現する技術としてデータベースのバッファ管理方式の研究を試みる。具体的には、MUMPSデータベースの各層のデータブロックに優先順位を考慮することにより、データベースをアクセスした時のバッファ不在頻度を最小化するような、従来のLRU法に代わる新たなバッファ再配置アルゴリズムを提案し、その有効性を検証することが目的である。

一方、データベース設計の自動化技術も従来から研究が盛んに行われてきており、設計方法論の研究、また概念設計、論理設計、物理設計それぞれの設計段階での設計支援ツールの開発、さらには設計後のデータベース性能最適化なども研究されてきた。そこで、筆者はデータベースを実際に構築する前の応答性能やアクセスコストの見積もり、構築済みデータベースの再設計や性能改善を可能とするような、論理設計、物理設計、スキーマ性能評価の各設計フェーズを統合化し、かつ対話的な利用者インタフェースを提供した画面指向型データベース設計・評価ツールを研究試作することを目的とする。その主要な要素技術は、データベースの論理構造、物理構造を数式化し、その応答性能やアクセスコストをその数式上で予測できる評価モデルであるので、このモデルの構築が本研究の大きな目的の一つである。

**分散環境** 分散計算環境において利用者から要求されたデータベース技術としては集中計算環境と共通しているものが多いが、筆者はこの中で特にクライアント/サーバ環境における使い易いデータベース利用者インタフェースの実現、そして企業の経営者などが行う意思決定のような高度な業務への適用性の向上などを研究の目的とする。

従来のデータベース利用者インタフェースは、SQL<sup>26</sup> 言語に代表される問合せコマンドをTSS端末キーボードから入力していた。そこで筆者は新たに視覚的な概念によるデータベース利用者インタフェースを提案し、アイコン、マルチウィンドウなどをベースにしたオブジェクト先行型利用者インタフェースをUNIXワークステーション上に研究試作する。続いて、この利用者インタフェースをUNIXワークステーションとデータベースサーバを連結させたクライアント/サーバ環境に拡張した技術を研究開発する。この結果、分散計算環境でも集中計算環境でも操作が統一化され、かつデータベース位置透過なエンドユーザコンピューティングの概念を実現することを本研究の目的とする。

一方、意思決定業務への適用はデータベースの技術が登場して以来の永遠の課題である

<sup>25</sup> Massachusetts General Hospital Utility Multi-Programming System

<sup>26</sup> Structured Query Language

が、その研究は意思決定そのものではなくその支援技術から始まり、現在はAI技術を利用した意思決定そのものの研究も行われている。筆者は先ず意思決定支援のためのツールである意思決定支援システムDSS<sup>註7</sup>の研究開発を手掛ける。DSSは特定の問題解決に特化した問題向きDSSと汎用的な問題解決機能から成る汎用型DSSの二系統に大別されるが、筆者は問題向きDSSを包含した汎用型DSSの研究試作を目指し、EDP<sup>註8</sup>とOAの融合化、垂直分散化、操作の統合化などの技術開発を目的とする。この結果、データベースの意思決定業務への適用性の向上を計る。その後、意思決定支援技術の新たな研究課題として、最近脚光を浴び始めたデータマイニング技術に注目し、クライアント/サーバ環境におけるデータマイニングサーバの研究試作を現在実施している。

**移動環境** 最近の携帯端末の低価格化、高性能化、高機能化によりその普及が急速に進んできたため、そのデータ通信機能を用いた移動計算環境でのデータベースアクセス要求も同時に高まってきた。しかし、移動計算環境でのデータベース技術に関する研究はまだ緒についたばかりである。移動体計算環境において利用者から求められるデータベース技術もやはり、集中計算環境や分散計算環境と共通するものも数多くあるが、先述したように移動体計算環境独自の技術課題もある。例えば、携帯端末利用者が移動しながらデータベースサーバをアクセスする際、どの無線通信セルの中にいるのか、あるいはどの無線通信セルからどの無線通信セルへ移動したのかを意識することなく効率良くデータベースアクセスすることが困難となる問題がある。また、こうした移動アクセス中に携帯端末が電源切れしてしまい、データベースアクセスが途中で断続してしまい、回復が不可能となる問題もある。そこで筆者はこうした課題の中から移動計算環境において携帯端末利用者が複数の無線通信セル間を移動しながらデータベースサーバをアクセスする時、移動先の無線通信セルでデータベースをアクセスする時のページ不在発生頻度を最小化するためのデータベースキャッシュ管理方式を新たに提案し、計算機シミュレーションによりその有効性を検証することを本研究の目的とする。

以上まとめると、筆者は計算環境が集中計算環境から分散計算環境そして移動体計算環境へと進展化していく過程で、先ず各々の計算環境においてデータベース利用者の要求を明らかにし、次にその要求を満たす最適なデータベースのアクセスと管理の技術を研究開発し、最終的には最近新しい計算環境として注目され、その利用が徐々に実用的になってきた移動体計算環境を取り上げ、この計算環境に求められるデータベースのアクセスと管理の方式を提案し、その有効性を検証することを本研究の目的としている。

## 1.2 従来研究の概観

本研究課題である「集中・分散・移動環境におけるデータベースのアクセスと管理に関する研究」の中で、筆者が具体的に研究を行ってきた課題である「データベースの利用者

<sup>註7</sup> Decision Support System

<sup>註8</sup> Electronic Data Processing



インタフェース」、「データベースを用いた意思決定支援技術」、「データベースの設計自動化支援・性能評価方式」、「移動体計算環境におけるデータベース管理方式」に焦点を絞り、従来研究動向を概観する。次に、その動向の中に於ける本研究の各々の課題の位置づけを明確にする。

**データベースの利用者インタフェース** データベースの利用者インタフェースを改善するための研究開発は以前から盛んに行われている。その代表的な技術は、視覚的インタフェース、例示型インタフェース、自然言語、知的インタフェース、第四世代言語などである。

視覚的インタフェース技術の特長は、操作の対象がアイコンやウィンドウという形で利用者の目に見えるようになっていて、視覚的な対象をポインティングデバイスで直接操作できることなどである。

例示型インタフェース技術の特長は、データベース検索の例題を示しながら端末フルスクリーンモードでデータベース操作することである。

自然言語技術の特長は、自然言語が人間やシステムに対して柔軟であること、すなわち利用者の持つ要求のあいまい性を吸収できることである。具体的には、自然言語によるデータベース問い合わせ文をSQL文に変換する技術が研究されている。

知的インタフェース技術の特長は、知識処理機能の応用にあり、知的ガイダンス機能や知的情報検索機能などが主に研究されている。

第四世代言語の特長は、ソフトウェアの生産性向上を意図したプログラムレスツールであり、データベース機能、スプレッドシート機能、グラフィックス機能などが統合化されている点を特長をしている。現在、数多くの製品が市場に出回っている。

以上まとめると、自然言語や知的インタフェースはまだ研究段階の技術であり、例示型インタフェースと第四世代言語はかなり成熟した技術であるが、視覚的インタフェースは研究段階から実用化レベルへと推移している段階の技術であると考えられる。ビットマップディスプレイに代表されるコンピュータ入出力技術の最近の進歩・普及を考えると、先述した利用者インタフェースの中で視覚的インタフェースが今後ますます重要な技術となっていくと思われるので、本研究でもデータベースの視覚的利用者インタフェース技術の研究開発を実施する。

**データベースを用いた意思決定支援技術** 企業や組織における経営者や管理者に代表される意思決定者のためのデータベースシステムを利用した意思決定支援技術の代表例として、「意思決定支援システムDSS」と「データマイニング技術」がある。

まず、DSSは企業や組織の管理者や経営者が短期的な生産計画、販売計画、利益計画などの戦術的意思決定、あるいは新製品開発、設備投資などの戦略的意思決定を行なう際に彼らの業務遂行を支援するための計算機プログラムツールである。例えば、金融機関では支店設置計画モデルシミュレーション、流通業ではスーパーの坪当りの売り上げ高寄与率分析、製造業ではプラント建設に伴う投資採算分析などに適用されている。コンピュ

ータ市場に流通しているDSS製品ソフトウェアはその提供機能から分類すると、問題向きDSSと汎用型DSSの2系統が存在する。問題向きDSSは、例えば企画部門や財務部門などの特定部門の担当者を対象としたシステムであり、シミュレーション機能やモデル記述機能を主体としている。従って、市場分析、経営分析計画などの業務には向いている。

一方、汎用型DSSは企業内利用者の要求をすべて満たすことを意図したシステムであり、データベース検索、作表、ビジネスグラフ作成、報告書作成、モデル記述、シミュレーション、統計分析などの広範囲なデータ処理機能を提供している。従って、問題向きDSSは汎用型DSSに完全に包含できると考えられるので、本研究でもこの汎用型DSSの研究開発を推進する。

次に、近年データベースから情報を発掘する技術であるデータマイニングが注目されている。中でも相関ルールと呼ばれるデータ間の相関関係を発見しようとする分野では、その処理方式等について、さまざまな提案がなされている。相関ルール生成の高速化については、並列化による研究が知られている。また、データベースのデータに関する用語辞書をデータマイニングの際利用する研究も行われているが、その用語辞書の中の階層上で適切な水準の相関ルールを自動的に選択する方式についても研究例が報告されている。本研究では、この相関ルールを中心に、適用業務のノウハウ情報や階層型辞書情報を機能追加したデータマイニング技術を研究する。

データベースの設計自動化支援・性能評価方式 データベースの性能を左右する要素として、データベースの設計・構築、そしてデータベースのバッファ管理方式などが考えられる。

まず、データベースの自動設計技術は、近年盛んに研究されている重要な技術課題の一つである。データベース設計は、情報構造の設計、論理構造の設計、物理構造の設計の3クラスに分類できる。現在、各設計の方法論に関する研究成果が数多く報告されている。一方、設計評価ツールも数多く報告されているが、その基本的な方式は数式モデル、シミュレーション、この両者の混合の3種類に大別できる。本研究では情報設計は取り上げず、論理設計と物理設計を研究の対象としており、その研究目的はCODASYL<sup>註9</sup> 準拠データベースのスキーマの設計と性能評価のためのツールを開発することである。

次に、医療データベースとして利用が盛んになってきたMUMPSデータベースは、グローバルファイルとも呼ばれ、スパース状の多次元配列構造をしており、そのアクセス効率はシステム全体の性能を大きく左右する。データベースのアクセス効率はデータベース用バッファの容量や置換方式、およびデータベースの物理構造などに大きく影響される。

本研究は上述の要因の置換方式のみに着眼したものである。MUMPSデータベース用のバッファ置換方式は、MUMPSデータベース固有の構造やアクセス方式の特徴を活かしたものが既に提案されているが、その提案を実現するには不明な点が幾つかありその有効性に関する記述もない。そこで本研究はその提案を基にし、バッファ単位に優先順位を

<sup>註9</sup> Conference on Data Systems Languages

定め、その優先順位に従ったバッファ置換アルゴリズムを研究する。

移動体計算環境におけるデータベース管理方式 近年携帯電話や携帯型コンピュータの低価格化や高性能化により、その普及が急速に進んできた。こうした移動体計算の進展に伴い、そのコンピューティング環境からのデータベースアクセス要求も高まってきているが、その研究は緒についたばかりである。例えば、既存データベースシステムに対するノマディックコンピューティングの影響（具体的には、バッテリー電源、無線通信などの問題）について議論している。この他、クライアントの電源断が高頻度で発生する無線コンピューティング環境におけるデータベースキャッシュ無効化手法の研究、モバイルコンピュータのための静的および動的なデータ割当アルゴリズムのコストの研究、移動体計算という新パラダイムに対してデータベース技術から挑戦する際の問題点（例えば、無線データ放送、通信切断管理、電源節約型データアクセス）の提示、データを多数の利用者に無線放送するときのそのデータの編成方法とアクセス方法に関する研究などが発表されている。更に、移動通信環境における質問処理の問題、移動体計算環境においてデータ要求を統合利用できる移動体ビューの提案、特定のセルを指定したときのビューの維持方法やビュー機構の実装方法なども研究されている。また、モバイルデータベースシステムに適用する通信コスト削減プロトコルについても研究されている。

こうした移動体計算環境におけるデータベース技術の研究動向を概観してみると、まだまだ研究されていない新しい課題が存在することに気が付く。たとえば、インターネットに代表される広域の計算機通信網に分散して接続された多数のデータベースシステム、すなわち広域分散データベースシステムを利用するとき、ある固定されたサイトのみを利用者が滞在してそこから他のサイトにあるデータベースをリモートアクセスできる環境を実現する基盤技術と、その計算機通信網の中を利用者が自由に移動して、いつでもどこからでも欲しいデータを適切なデータベースからアクセスできる環境を提供する基盤技術とは、自ずとその性質が異なり、新たな技術のブレークスルーに基づくインフラの提供が必要になると考えられる。

そこで、本研究ではこうした新たな基盤技術の提供を目指して次のような移動体計算環境に適応したデータベースシステムの研究を実施する。

先ず、本研究では最近議論されている無線モバイルコンピューティングシステムと類似のシステムを仮定する。しかし、既に発表されているデータベースキャッシュ管理アルゴリズムとは異なる新しいアルゴリズムを提案し、かつ評価していく。すなわち、従来研究では、無線携帯型コンピュータが無線モバイルコンピューティングシステムと接続したり、あるいは切断したりするとき、その無線携帯型コンピュータ上でのデータベースキャッシュの問題を扱っているのに対し、本研究では無線携帯型コンピュータの利用者がある無線通信セルから別のセルへと移動するとき、最寄りの無線通信セルに配置されているデータベース上でのキャッシュの問題を扱う。

### 1.3 研究の成果と特徴

本節では、先ず本研究の実施方法と有効性評価方法を記述し、次に研究成果とその特徴

を述べていく。

本論文では、計算環境が集中計算環境、分散計算環境、そして移動体計算環境へと進展していく過程に注目し、各々の環境におけるデータベース利用者要求を満足させるためのデータベース要素技術を提案し、その提案技術をソフトウェアとして研究試作あるいは、計算機上でシミュレーションすることにより、その提案技術の有効性や効果を検証する。

計算環境の進歩の歴史を見ると、集中・分散・移動体の順であるが、それぞれ重要な計算環境であり、現在も未来も共存共栄していく技術である。そこで、本論文では最も分かり易いデータベースの利用技術の研究から入り、データベースの性能評価の研究へと議論を進める形式でまとめるので、まずは分散環境におけるデータベースの利用技術、次に集中計算環境におけるデータベースシステムの性能評価、そして移動体計算環境におけるデータベースシステムの性能評価の順番で各々の成果を記述する。

分散環境におけるデータベースの利用技術に関する研究は、データベース利用者インタフェース技術、意思決定支援技術、データマイニング技術の3件の研究課題ともそれぞれの提案技術や概念をソフトウェアとして研究試作して、試使用を通じてその有効性や効果进行评估する。

また、集中計算環境におけるデータベースシステムの性能評価に関する研究は2件あるが、第一のデータベースの設計自動化支援・性能予測技術の研究もその設計・評価に関する数式モデルをソフトウェアとして研究試作して、試使用を通じてその有効性や効果进行评估する。一方、第二のバッファ管理方式の研究は、計算機上にそのアルゴリズムをモデル化し、統計的な手法によるシミュレーションを行い、今回提案の新方式と従来方式を対比させながらその有効性を定量的に評価する。

そして、移動体計算環境におけるデータベースシステムの性能評価に関する研究は、データベース・キャッシュ管理方式のアルゴリズムを計算機上でモデル化し、統計的な手法によるシミュレーションを行い、今回提案した複数種類の方式のそれぞれについてその有効性を定量的に比較評価する。

続いて、研究成果とその特徴をまとめる。

分散環境におけるデータベースの利用技術 分散計算環境におけるデータベースの利用技術に関する3件の研究課題に対する成果を総括すると、先ずクライアント/サーバ環境におけるデータベース位置透過性を実現し、かつ視覚的な概念を実現したデータベース・ユーザフレンドリインタフェース技術を研究試し、エンドユーザコンピューティングを実現した。

具体的には、先ずUNIX系エンジニアリング・ワークステーション向けにデータベース・ユーザフレンドリ・インタフェースMELQUERYを研究試作した。このMELQUERYはオブジェクト先行型の視覚的利用者インタフェースであることが主要な特徴であり、リレーショナル・データベースのエンドユーザにとって使い易いインタフェースで提供できた。

次に、ホストコンピュータの共用リレーショナル・データベースに対して、クライアント/サーバ方式により、UNIX系エンジニアリング・ワークステーションから視覚的かつ対話形式の利用者インタフェースによって検索が行える遠隔データベースリクエストMELQUERY/Rも研究試作した。

更に、垂直分散化と操作の統一性を特長とするDSS-GOALを研究試作し、データベースの意思決定業務への適用性を向上させた。このDSS-GOALはEDPとOAの融合化をはかり、統合化利用者視野を提供することを特長とする処理の垂直分散型DSSであり、利用者である意思決定者にパーソナルテーブルという統合化利用者視野を提供し、かつ従来のEDPやOA業務に対して同一の利用者インタフェースを提供することにより、その親和性と操作性を向上させた。更に、ホスト計算機と端末の間で共通ファイルの導入による処理と資源の垂直分散化を実現することにより、システム導入性、例えば異機種との接続性を向上させた。

そして、データベース技術とAI<sup>注10</sup> 技術と統合化した新たな意思決定支援技術として最近注目されているデータマイニング技術を研究し、従来のデータベースアクセス技術やDSSでは提供できないデータ間規則性の発見が可能となった。その規則性を用いることによりより高度な意思決定支援技術が実現された。

具体的には、大規模なデータベースからデータ間に潜在する有効な相関ルールを発見するために業務ノウハウとオントロジ（概念辞書）を利用したデータマイニングの手法を先ず提案した。

次に、このデータマイニング手法の中心的な技術である属性指向の相関ルール生成機能についても提案した。そして、これをデータマイニングサーバKnodi asとして実装して、ルール発見の柔軟性と高速化が実現できることを示し、この方法論の有効性を議論した。

集中計算環境におけるデータベースシステムの性能評価 集中計算環境におけるデータベースシステムの性能評価に関する2件の研究課題に対する成果を総括すると、先ず論理設計、物理設計、スキーマ性能評価の各設計フェーズを統合化したデータベース設計自動化・性能予測ツールDMSDBDAを試作し、データベースのアクセス費用と性能をデータベース構築の事前に予測と改善を可能とした。データベース管理者が新規のデータベースを開発する際、本ツールを用いるとデータベースを実際に開発する前にデータベースの性能や費用を見積ることができる。一方、既存のデータベースに本ツールを適用するならば、そのデータベースの性能改善の手助けにもなる。DMSDBDAはFORTRAN言語で記述されており、そのプログラム数は約6,000行、その容量は約55K語である。本ツールの対象とするデータベースの規模は、データ項目タイプが最大500件、グループタイプが最大50件、セットタイプが最大50件、エリアが最大5件である。

---

<sup>注10</sup> Artificial Intelligence

次に、木構造をしたMUMPSデータベースの各層に優先順位を考慮したバッファ管理方式を提案し、データベースアクセス時のバッファ不在頻度を最小化することを計算機シミュレーションにより検証した。実際には、MUMPSデータベースに適したバッファ置換アルゴリズムを、計算機シミュレーションにより、バッファ面数ごとに調査した。MUMPSデータベース用バッファ置換アルゴリズムの性能評価に関する報告は本研究のほかにはまだ見られないが、データベースの設計上、その評価結果は重要な情報である。

更に、今回の評価結果の具体的な利用成果としては、BPL\*法が医療情報システム<<MELCOM MUMPS>>に採用されて、商品化されたことが挙げられる。なお、本研究のシミュレーションプログラムはFORTRAN言語で記述し、そのステップ数は約1,500行、サイズは約34K語である。

移動体計算環境におけるデータベースシステムの性能評価 移動体計算環境におけるデータベースシステムの性能評価に関する研究課題に対する成果を総括すると、先ず無線携帯型コンピュータ利用者が複数セル間を移動しながらホストデータベースをアクセスするときセル間でシームレスなデータベースアクセスを保証するための時間制約付きデータベースシステムを提案した。次に、その携帯型コンピュータ利用者が複数セル間を移動しながらホストデータベースをアクセスするとき、移動先のセルでデータベースアクセスフォルトを発生する頻度を最小化するためのデータベースキャッシュ管理方式（すなわち、全セルキャッシュ法、隣接セルキャッシュ法、1セル置きキャッシュ法、2セル置きキャッシュ法、3セル置きキャッシュ法）も提案した。そして、各々のキャッシュ管理法の下でのホストデータベースアクセス経過時間の平均値やセルデータベースの平均キャッシュヒット率をシミュレーションにより定量的に測定して比較評価し、本研究にて提案したデータベースツキャッシュ管理方式が、移動体計算環境におけるデータベースアクセスに有効であることを示した。なお、本研究で採用したシミュレーション方法は、乱数発生によるモンテカルロ法を用いている。また、このシミュレーションプログラムはC言語で記述し、総ステップ数は約10K行である。

## 1.4 論文の構成

本論文は全6章から構成されている。

第1章では本研究の背景および目的、従来の研究の概観、本研究の特徴等を記述した。

第2章では本研究課題である「集中・分散・移動環境におけるデータベースのアクセスと管理に関する研究」の中で、筆者が具体的に研究を行ってきた課題である「データベースの利用者インタフェース」、「データベースを用いた意思決定支援技術」、「データベースの設計自動化支援・性能評価方式」、「移動体計算環境におけるデータベース管理方式」に焦点を絞り、従来研究動向を調査する。次に、その動向の中に於ける本研究の各々の課題の位置づけを明確にする。

続いて、本研究課題である「集中・分散・移動環境におけるデータベースのアクセスと管理に関する研究」の中の「データベースのアクセスに関する研究」は第3章に、一方この中の「データベースの管理に関する研究」は第4章と第5章に記述する。

第3章では、先ずデータベースの利用者にとって使い勝手が良いデータベース利用者インタフェース技術とは何かを追求する。そして、そこから得られた利用者インタフェース要求に対して、スタンドアロン・ワークステーションの集中計算環境に向けた視覚的な概念に基づくオブジェクト先行型ユーザフレンドリインタフェース技術と、その技術をクライアント/サーバ・アーキテクチャの分散計算環境向けに拡張し、その分散計算環境におけるデータベース位置透過性を実現したデータベース・ユーザフレンドリインタフェース技術を研究試作した結果を記述する。そして、分散計算環境においてもエンドユーザコンピューティングの道具として有効であることを実証する。

続いて、データベースの利用を推進させるために必要なその応用技術は何かを追求する。具体的には、OA業務と意思決定支援業務に注目し、こうした業務においてデータベースを有効活用するための技術として、DSSとデータマイニング技術を取り上げる。そして、垂直分散化と操作の統一性を特長とする対話形式のDSSを研究試作し、OAの世界でデータベースの意思決定業務への適用性を向上させた結果を記述する。次に、このDSSでは得ることが困難であるデータベースに内在するデータ間規則性を抽出できるデータマイニング技術に、データベース応用業務に関する専門ノウハウ情報などを統合化したデータマイニングサーバを現在研究試作しているが、その成果をまとめる。

第4章では、データベース処理の永遠の課題である高速化方式を追求する。このデータベースの高速化技術は、従来のデータベースマシンに代表されるハードウェアによる方式、データベース自体の構築方法の工夫やデータベース管理システムの中のデータベースアクセス方式の改良などに代表されるソフトウェアによる方式、そしてデータベース以外の技術要素、例えば並列処理技術の利用、クライアント/サーバ・アーキテクチャの採用、通信回線速度の高速化などによる方式が研究開発されてきた。

筆者は、この中でハードウェア環境やシステム環境が事前に決められているという前提の上でのデータベース処理高速化技術を研究開発するために、上記の2番目であるソフトウェアによる高速化方式に着目し、次の2件の課題を研究した。

第一は、データベース構築のための設計自動化とその性能予測方式すなわちスキーマの設計自動化方式、第二はデータベース管理システムの中でデータベースアクセス性能を大きく左右するバッファ管理方式である。

そこで、先ず集中計算環境において、データベースの論理設計、物理設計、スキーマ性能評価の各フェーズを統合化した対話形式のデータベース設計自動化支援・性能予測ツールを研究試作した。そして、データベースを構築しようとする際、このツールを用いてそのアクセス費用と性能を実際にデータベースを構築しなくとも事前に予測し、またデー

データベースをいったん構築した後でもそのデータベースを再設計した後のアクセス費用と性能を見積もりを可能とした結果を記述する。一方、このツールを構成する主要な要素技術はデータベースの論理構造、物理構造を数式化し、その応答性能やアクセスコストをその数式上で予測できる評価モデルであるが、本研究では従来の評価モデルである Gerriksenモデルを拡張した数式モデルを作成し、ツールとして実現したので、そのモデルの詳細な数式を付録にまとめる。

次に、同じく集中計算環境において、医療情報データベースとして広く世の中に普及している木構造データベースMUMPSを取り上げて、その木構造の各層の個々のデータブロックについてバッファ置換の優先順位を考慮した新バッファ管理方式を提案し、従来のバッファ管理方式に比べて、データベースをアクセスした時のバッファ不在頻度を最小化することを計算機シミュレーションにより検証した結果をまとめる。

第5章が本論文の中心的な研究成果であるが、ここでも前章に引き続きデータベース処理の永遠の課題である高速化方式を追求するが、ここでは特に集中計算環境、分散計算環境、そして移動計算環境へと計算環境が進展してきたが、この中でデータベースアクセスする場所を選ばない移動体計算環境に焦点を絞り、この移動体計算環境において最近必要性が高まってきたデータベースアクセスを高速化するための方式を研究する。具体的には、1台の共通データベースサーバと複数の無線通信セル（この無線通信セルにはそのセル専用のローカルなデータベースが配置。）から構成される移動体計算環境を仮定する。

そして、その環境にいる携帯端末利用者が、その環境の中の無線通信セル間を移動しながら共通データベースサーバをアクセスする時、そのデータベースサーバと各無線通信セルごとに配置したセルデータベースの間で行うデータベース・キャッシュ管理方式を新たに提案する。次に、データベースアクセス時のページ不在発生頻度やそのアクセス経過時間を計算機シミュレーションを行うことにより、今回提案したデータベース・キャッシュ管理方式が有効であるか否かを比較評価した結果を記述する。

第6章では、第3章から第5章に記載する研究成果の各々についてまとめると同時に、具体的な応用分野への展開、研究試作したソフトウェアの利用状況、そして今後の研究課題をまとめる。この中で、特に移動体計算環境でのデータベース・キャッシュ管理方式は、移動体計算環境の共通基盤技術として今後かなり大きな応用可能性があると期待できる。例えば、地震や火災などに事前に備えるための防災情報システム、またそうした地震や火災などが発生したあとの危機管理情報システムへの展開が考えられる。こうしたシステムでは、災害現場にいる個人個人と災害対策本部との間での有線によるインターネット/イントラネット経由でのデータベースアクセスはもちろん、その有線回線が断線した場合の代替通信回線である無線通信（携帯電話から衛星通信回線までを含めた。）による情報の双方向通信、すなわちデータベースアクセスが必須な技術となるからである。



## 第2章 関連研究

### 2.1 緒言

本研究の主題である「集中・分散・移動環境におけるデータベースのアクセスと管理に関する研究」を構成する「データベースの利用者インタフェース」、「データベースを用いた意思決定支援技術」、「データベースの設計自動化支援・性能評価方式」、「移動体計算環境におけるデータベース管理方式」の各々について、過去20年間における国内外の関連研究動向を調査し、本研究課題の位置づけを整理していく。

それぞれ、2.2節、2.3節、2.4節、および2.5節において記述する。

### 2.2 データベースの利用者インタフェース

最近、データベースの利用者インタフェースを改善するための研究開発が盛んに行われている[Koji88]。

その代表的な技術は、

- 視覚的インタフェース
- 例示型インタフェース
- 自然言語
- 知的インタフェース
- 第四代言語

などである[Maki87]。

#### 視覚的インタフェース

この技術の特長は、操作の対象がアイコンやウィンドウという形で利用者の目に見えるようになっていること、視覚的な対象をポインティングデバイスで直接操作できることなどである[Davi86, Fras86, Gold85, Heil85, Kato87-1, Kato87-2, Wats88]。

#### 例示型インタフェース

この技術の特長は、データベース検索の例題を示しながら端末フルスクリーンモードでデータベース操作することである[Chan80, Chan81, Whan87, Zl0075, Zl0077]。

#### 自然言語

この技術の特長は、自然言語が人間やシステムに対して柔軟であること、すなわち利用者の持つ要求のあいまい性を吸収できることである[Maki87]。具体的には、自然言語

によるデータベース問い合わせ文をSQL文に変換する技術が研究されている[Naka86]。

### 知的インタフェース

この技術の特長は、知識処理機能の応用にあり、知的ガイダンス機能や知的情報検索機能などが主に研究されている[Maki87]。

### 第四代言語

この特長は、ソフトウェアの生産性向上を意図したプログラムレスツールであり、データベース機能、スプレッドシート機能、グラフィックス機能などが統合化されている点を特長をしている。現在、数多くの製品が市場に出回っている[Mart86]。

以上まとめると、自然言語や知的インタフェースはまだ研究段階の技術であり、例示型インタフェースと第四代言語はかなり成熟した技術であるが、視覚的インタフェースは研究段階から実用化レベルへと推移している段階の技術であると考えられる。

ビットマップディスプレイに代表されるコンピュータ入出力技術の最近の進歩・普及を考えると、本節で述べた利用者インタフェースの中で視覚的インタフェースが今後ますます重要な技術となっていくと思われるので、本研究でもデータベースの視覚的利用者インタフェース技術の研究開発を進めてきた。

ところで、ネットワーク環境による分散環境のシステムにおける利用者インタフェースの要件に、

- 操作環境の統一性
- 処理に必要とされる操作の簡略化がある。

### 操作環境の統一性

ワークステーションで作業を行う場合、そのワークステーション内のローカル処理も、ネットワークを経由したりリモート処理も、同形式の環境による利用者インタフェースで行えることが望ましい。すなわち、処理を実行するコンピュータごとに異なる操作環境を提供するのではなく、処理するコンピュータの相違やネットワークの存在を利用者に意識させないような操作環境を提供するということである。

### 操作の簡略化

ワークステーションでは、マルチウィンドウなどのグラフィック機能やマウス操作により、利用者インタフェースの操作性向上や簡略化が行われてきている。

以上のことから、ネットワーク環境における情報の一元管理用コンピュータとしてホストコンピュータを位置づける場合、UNIX系エンジニアリング・ワークステーションから

ホストコンピュータのデータベースサービスを利用するとき、UNIX系エンジニアリング・ワークステーションの機能を生かした、ローカルサービスとの違和感がないような操作環境が必要となる。

一方、本研究ではUNIX系エンジニアリング・ワークステーション上にデータベース操作を視覚的かつ対話形式に行える利用者インタフェースとして、MELQUERYを研究試作している[Aka87, Aka86, Kana88, Kana87, Naka89, Tagu86, Wada89, Wada88, Wada87]。

そこで、UNIX系エンジニアリング・ワークステーションのデータベースの利用者インタフェースMELQUERYをもとに、クライアント/サーバ方式により、ホストコンピュータのデータベースのアクセスを行う遠隔データベースリクエストMELQUERY/Rへの拡張開発も行った[Mori91]。

## 2.3 データベースを用いた意思決定支援技術

企業や組織における経営者や管理者に代表される意思決定者のためのデータベースシステムを利用した意思決定支援技術の代表例として、「意思決定支援システムDSS」と「データマイニング技術」がある。

各々、2.3.1節、および2.3.2節にて記述していく。

### 2.3.1 DSS

DSSは企業や組織の管理者や経営者が短期的な生産計画、販売計画、利益計画などの戦術的意思決定、あるいは新製品開発、設備投資などの戦略的意思決定を行なう際に彼らの業務遂行を支援するための計算機プログラムツールである[Hiro83, Wada84]。例えば、金融機関では支店設置計画モデルシミュレーション、流通業ではスーパーの坪当りの売り上げ高寄与率分析、製造業ではプラント建設に伴う投資採算分析などに適用されている。

コンピュータ市場に流通しているDSS製品ソフトウェアはその提供機能から分類すると、問題向きDSSと汎用型DSSの2系統が存在する[IBM83, NIKKEI82]。問題向きDSSは、例えば企画部門や財務部門などの特定部門の担当者を対象としたシステムであり、シミュレーション機能やモデル記述機能を主体としている。従って、市場分析、経営分析計画などの業務には向いている。一方、汎用型DSSは企業内利用者の要求をすべて満たすことを意図したシステムであり、データベース検索、作表、ビジネスグラフ作成、報告書作成、モデル記述、シミュレーション、統計分析などの広範囲なデータ処理機能を提供している。従って、問題向きDSSは汎用型DSSに完全に包含できると考えられる。

### 2.3.2 データマイニング技術

近年、データベースから情報を発掘する技術であるデータマイニングが注目されている。中でも相関ルールと呼ばれるデータ間の相関関係を発見しようとする分野では、その処理方式等について、さまざまな提案がなされている。

相関ルール生成の高速化については、並列化による研究が知られている[Agw94, Shin96]。

また、データベースのデータに関する用語辞書をデータマイニングの際利用する研究も行われているが、その用語辞書の中の階層上で適切な水準の相関ルールを自動的に選択する方式についても研究例が報告されている[Fuku96, Han95, Shrk95]。

## 2.4 データベースの設計自動化支援・性能評価方式

データベースの性能を左右する要素として、データベースの設計・構築、そしてデータベースのバッファ管理方式などが考えられる。

それぞれ、2.4.1節、そして2.4.2節にて記述していく。

### 2.4.1 データベースの自動設計技術

データベースの自動設計技術は、近年盛んに研究されている重要な技術課題の一つである。データベース設計は、

- 情報構造の設計
- 論理構造の設計
- 物理構造の設計

の3クラスに分類できる。

現在、各設計の方法論に関する研究成果が数多く報告されている。例えば、文献[Beri77, Bube76, Schi79, Smit77-1, Smit77-2]が情報構造の設計について、また文献[Chen76, Chen77-1, Chen77-2, Gerr75, Nava78, Hous79]が論理構造の設計について、また、文献[Gerr77, Schk78]が物理構造の設計について各々研究している。

一方、設計ツールも数多く報告されている。CODASYL形データベース向けのツールとしては、文献[Gamb77]のDBDDSS、文献[Hult77]のARTS/DB、文献[Bere77]のツール、文献[Teor78]のDBDE、文献[Yone79]のPEALなどが報告されている。また、その基本的な方式は

- 数式モデル
- シミュレーション
- この両者の混合

の3種類に大別できる。

本研究は情報設計は取り上げず、論理設計と物理設計を研究の対象としており、その研究目的はCODASYL準拠データベース[Tayl 76]向けの設計評価ツール（特に、CODASYLスキーマの設計と性能評価のためのツール）を開発することである。

本論文ではCODASYL仕様に準拠した商用データベース管理システムの一つであるMELCOM-EDMSデータベース管理システム[Mel co76]を取り上げ、このEDMS専用開発した設計評価ツール（DMSDBDAと呼ぶ）について述べていく[Wada81, Wada80]。

DMSDBDAはデータベース管理者向けの対話形式の設計支援ツールであり、CODASYL形データベース設計のための数式モデルとして文献[Gerr 77]が発表したモデル（Gerritsenモデルと呼ぶ）をEDMS向けに修正した数式モデルを基盤にしている。先述のDBDDSSもGerritsenモデルを基盤にしているが、DMSDBDAはDBDDSSと比べると、以下の特徴を有している。

### (1) 設計フェーズ

DBDDSSは物理設計と性能評価とスキーマ生成の機能を提供しているが、DMSDBDAはこの3機能に加えて論理設計の機能も提供している。

DMSDBDAの論理設計は、データ項目タイプの定義から始まり、グループタイプの定義、セットタイプの定義、エリアの定義へと進む。

## 物理設計

DBDDSSの物理設計は、

- ロケーションモード、
- セットオーダ、
- エリア容量などの定義機能を提供しているが、

DMSDBDAはこの定義機能のほかに、

- EDMSが提供しているINDEXED<sup>注11</sup> ロケーションモード、
- グループオカレンスの記憶範囲（ページ範囲と呼ぶ）の定義機能も提供している。

### (2) 利用者インタフェース

DBDDSSはタイプライタ形式の端末装置からデータベース設計のコマンドをタイプインする方式をとっているが、DMSDBDAはCRTキャラクタディスプレイ装置に表示されるメニュー画面を選択しながら設計を進めていく方式をとっている。

### (3) 事前情報

DBDDSSを用いて設計や評価を行うとき、データベース管理者はデータ項目名、レコード名やセット名などを覚えていなければならないが、DMSDBDAでは一度定義したデータ項目名、グループ名、セット名などの情報はツール側が保持しており、且つ必要なとき

---

注11 索引付け指定

には画面に表示するので、データベース管理者はその情報を覚えていなくともよい。

#### 2.4.2 データベース管理システムのバッファ管理方式

医療データベースとして利用が盛んになってきたMUMPSは、1967年に米国マサチューセッツ総合病院のコンピュータサイエンス研究所にて開発されたものである。

MUMPS言語と呼ばれる高水準インタプリタ言語が提供されており、更に多数のオンライン端末装置からMUMPS言語を用いて、データベースを容易に共用することができる [Oogu76, Degu79]。

MUMPSデータベースはグローバルファイルとも呼ばれ、スパース状の多次元配列構造をしており、そのアクセス効率はシステム全体の性能を大きく左右する。データベースのアクセス効率はデータベース用バッファの容量や置換方式、およびデータベースの物理構造などに大きく影響される [Fern78, Lan77, Sher76, Wass75-1]。

本研究は上述の要因の置換方式のみに着眼したものである。MUMPS用置換方式は、文献[Wass75-2]からMUMPSデータベース固有の構造やアクセス方式の特徴を活かしたものが提案されているが、その提案を実現するには不明な点が幾つかありその有効性に関する記述もない。

そこで本研究は文献[Wass75-2]の提案を基にし、バッファ単位に優先順位を定め、その優先順位に従ったバッファ置換アルゴリズムを2種類提案する。更に従来から広く普及しているLRU<sup>注12</sup>法も取り上げ [Bowi77, Fern78]、以上の3方式のいずれがMUMPSデータベースに最も有効であるのかを調べる。

そのためにデータベースとアクセス要求およびバッファ管理を計算機上にてシミュレーションし、各々のバッファ置換アルゴリズムにより管理されているバッファを用いてデータベースをアクセスした際のバッファ不在頻度を調査する [Wada79-1, Wada79-2]。この調査結果はMUMPSデータベース用バッファ置換アルゴリズムの方式を選択する際の支援情報として利用できる。

### 2.5 移動体計算環境におけるデータベース管理方式

近年、携帯電話や携帯型コンピュータの低価格化や高性能化により、その普及が急速に進んできた。こうした移動体計算の進展に伴い、そのコンピューティング環境からのデータベースアクセス要求も高まってきているが、その研究は緒についたばかりである [Nishi97, Tsuk96]。

例えば、文献[Alon93]は既存データベースシステムに対するノマディックコンピューティングの影響（具体的には、バッテリー電源、無線通信などの問題）について議論している。

文献[Barb94]はクライアントの電源断が高頻度で発生する無線コンピューティング環境におけるデータベースキャッシュ無効化手法を提案し、評価している。

---

<sup>注12</sup> Least Recently Used

文献[Huan94-1, Huan94-2]はモバイルコンピュータのための静的および動的なデータ割当アルゴリズムを提案し、そのコストを評価分析している。

文献[Imie92, Imie93, Imie94-1]は移動体計算という新パラダイムに対してデータベース技術から挑戦する際の問題点（例えば、無線データ放送、通信切断管理、電源節約型データアクセス）を提示している。文献[Imie94-2]はデータを多数の利用者に無線放送するときのそのデータの編成方法とアクセス方法を提案評価している。

文献[Kami95]は移動通信環境における質問処理の問題を扱っている。

文献[Naka95, Shir96]は移動体計算環境においてデータ要求を統合利用できる移動体ビューを提案し、更に特定のセルを指定したときのビューの維持方法やビュー機構の実装方法を議論している。

文献[Lai95]はモバイルデータベースシステムに適用する通信コスト削減プロトコルについて議論している。

こうした移動体計算環境におけるデータベース技術の研究動向を概観してみると、まだまだ研究されていない新しい課題が存在することに気が付く。たとえば、インターネットに代表される広域の計算機通信網に分散して接続された多数のデータベースシステム、すなわち広域分散データベースシステムを利用するとき、ある固定されたサイトのみを利用者が滞在してそこから他のサイトにあるデータベースをリモートアクセスできる環境を実現する基盤技術と、その計算機通信網の中を利用者が自由に移動して、いつでもどこからでも欲しいデータを適切なデータベースからアクセスできる環境を提供する基盤技術とは、自ずとその性質が異なり、新たな技術のブレイクスルーに基づくインフラの提供が必要になると考えられる。

そこで、本研究ではこうした新たな基盤技術の提供を目指して次のような移動体計算環境に適応したデータベースシステムの研究を開始した。

まず、本研究では文献[Barb94, Imie93]にて議論されている無線モバイルコンピューティングシステムと類似のシステムを仮定する。しかし、文献[Barb94]しているデータベースキャッシュ管理アルゴリズムとは異なる新しいアルゴリズムを提案し、かつ評価していく。すなわち、文献[Barb94]では、無線携帯型コンピュータが無線モバイルコンピューティングシステムと接続したり、あるいは切断したりするとき、その無線携帯型コンピュータ上でのデータベースキャッシュの問題を扱っている。一方、本研究では無線携帯型コンピュータの利用者がある無線通信セル（以下、単にセルと呼ぶ。）から別のセルへと移動するとき、最寄りのセル内の無線基地（以下、モバイルサポートサーバと呼ぶ。）におけるデータベースキャッシュの問題を扱っていく。

具体的には、まず無線携帯型コンピュータを用いてホスト計算機に接続されている大容量ホストデータベースをアクセスする移動体計算環境において、その携帯型コンピュータの利用者が複数のセル間を移動しながら、最寄りのモバイルサポートサーバ（例えば、ワークステーションやPCサーバ）と交信し、そこからは有線を経由し、ホストデータベースをアクセスするためのシステムとして、新たに時間制約付きデータベースシステムを提案する[Wada96-1]。

次に、その携帯型コンピュータ利用者が複数セル間を移動しながらホストデータベース

をアクセスするとき、移動先のセルでデータベースアクセスフォルトが発生するので、その発生頻度を出来る限り最小化することが要求される。この要求に応えるために、本研究ではその携帯型コンピュータ利用者がセルを次々に移動していくことに注目し、最新利用データをその利用者の移動に先回りして、ホストデータベースと移動先のセルデータベース間でキャッシュする新たなデータベースキャッシュ管理方式を数種類提案する[Wada96-2]。

最後に、各々のキャッシュ管理方式の下でのホストデータベースアクセス経過時間の平均値やセルデータベースの平均キャッシュヒット率をシミュレーションにより定量的に比較評価する[Wada96-3]。

## 2.6 結言

「データベースの利用者インタフェース」の研究は1980年代から研究が活発化し、使い易さを求めることは永遠の課題であり、そのため現在でもなおその研究は行われている。しかしながら、本論文の研究課題以外で、この課題でまだ解決されずに、残っている主要な問題としては、先ず

- ・最近爆発的に普及したインターネットやイントラネット上のデータベースに代表される広域分散データベースをアクセス機能、

- ・このアクセス機能とWorld Wide Webブラウザなどのインターネット関連ソフトウェアとの連携技術、

などが考えられる。次に、データベースに限定されない共通的な利用者インタフェース技術については、

- ・利用者インタフェース仕様の動的な流れを文書化する技術

- ・利用者インタフェースの使い勝手を定量的に評価する技術

などが未解決な問題である。

続いて、「データベースを用いた意思決定支援技術」について見てみると、「意思決定支援システムDSS」は1980年代前半、また「データマイニング技術」は米国にて1990年代半ばからその研究が始まっている。特に、「データマイニング技術」は現在、最もホットな研究課題の一つとなっている。これは、データベースを意思決定業務を支援するための道具として実用化していくことが利用者からは強く求められていることに他ならない。この課題でまだ解決されずに残っている問題としては、

- ・意思決定そのもの技術（支援ではない）

である。しかしながら、この問題は解決するには大変困難であるので、そこへ向かうための段階的な問題として、例えばデータマイニングの技術分野では

- ・データベースからの規則性抽出の自動化技術

- ・その抽出の高速化技術

- ・その規則の品質向上技術

- ・その規則の検証技術



などが考えられ、更に

- ・データマイニング技術とDSSの融合化技術
- もその次の段階の問題として考えられる。

そして、「データベースの設計自動化支援・性能評価方式」について見てみると、1970年代から研究が始まったこの課題も「使い易い利用者インタフェース」の研究と同様に、永遠の課題であり、データベース処理の高速化研究ということで絶えることなく、地道に継続している。しかしながら、本論文の研究課題以外で、まだ解決されずに、残っている主な問題は、

- ・分散および移動体計算環境におけるデータベース設計の自動化と性能評価の技術
  - ・集中計算環境におけるテラバイト級の超大容量データベース（データウェアハウス）の構築技術と高速化技術
- などである。

最後に、「移動体計算環境におけるデータベース管理方式」はここ数年前に世の中で研究が活発化しており、「データマイニング技術」と同様今後の最もホットな研究課題の中心となっていくと思われる。しかしながら、本論文の研究課題以外で、まだ解決されずに、残っている問題は数多くあるが、例えば

- ・移動体計算環境におけるデータベースシステムアーキテクチャの標準化
  - ・移動体計算環境におけるデータベースアクセス言語（含、利用者インタフェース）
  - ・データベース自体が移動する環境におけるデータベースアクセス管理方式
  - ・そのデータベースの障害回復方式
- などが考えられる。

## 第3章 分散計算環境におけるデータベースの利用技術

### 3.1 緒言

分散計算環境におけるデータベースの利用技術の研究課題として、先ず「使い易いユーザフレンドリ・インタフェース技術」、次にデータベース技術のOA業務および意思決定業務への代表的な応用事例である「意思決定支援システムDSS」と「データマイニング技術」を取り上げて、研究開発した成果を以下に記載する。

**使い易いユーザフレンドリ・インタフェース技術** エンドユーザに対してデータベース管理システムを提供する際、データベース管理システムの機能として必要なものの一つに使いやすさがある。

既存の商用データベース管理システムでエンドユーザが使用するものでは、キーボードからコマンドを打ち込むコマンド型の利用者インタフェースを持つものがほとんどである。しかしながら、キーボード操作が多いので、入力ミスを引き起こしやすく、データベースの名称やデータ属性更にコマンドの構文を覚える煩雑さもある。

一方、エンドユーザに対して提供されるデータベース管理システムとしては、そのデータモデルが利用者にとってわかりやすく、かつ理論体系が整っているという点からリレーショナルモデル<sup>注13</sup>が世の中の主流となっており、データベース言語としてはJISにてSQLが規格化された。

このような状況のもとで、本研究においてもUNIX系エンジニアリング・ワークステーションの持つマウス、アイコン、マルチウィンドウの機能を用いて、視覚的かつ対話的にデータベースをアクセスするユーザフレンドリ・インタフェースMELQUERYを研究試作した。

ところで、パソコン、ワークステーションの処理能力の向上やネットワーク環境の整備に伴い、コンピュータの利用環境はこれまでのホストコンピュータと端末から構成される集中処理システムから、ワークステーションなどを中心とした様々なコンピュータがネットワークで相互接続されている分散処理システムへと移行してきている。

このような分散環境のシステムにおけるホストコンピュータの役割の一つに、共用データの一元管理機構を提供するデータベース処理がある。システム全体の共通のデータのサーバとして、ワークステーションからネットワーク経由のアクセスによって利用される。ホストコンピュータの端末からデータベースを利用するときの利用者インタフェースは、キャラクタベースのコマンド型インタフェースがほとんどである。しかし、エンド利用者がワークステーションからサーバとしてホストコンピュータのデータベース処理を利用する場合、リモート端末としてTSSエミュレータ等を利用した端末と同様の利用者イ

<sup>注13</sup> リレーショナルデータベースのデータ構造を表す数学的なデータモデルである。なお、リレーショナルデータベースとは2次元表データの集合であり、その表に対する演算（選択、射影、結合など）が数学的に定義されている。

ンタフェースを用いるのではなく、ワークステーションの利用環境に即した利用者インタフェースが必要とされる。

このような状況のもとで、UNIX系エンジニアリング・ワークステーションをクライアント、ホストコンピュータをサーバとしたクライアント/サーバ方式により、ホストコンピュータのリレーショナル・データベース管理システムRDMSIIに対して、UNIX系エンジニアリング・ワークステーションの機能を生かした視覚的でかつ対話形式により、データベース検索を行なえる遠隔データベース・リクエスタMELQUERY/Rへの拡張開発も行った。

**意思決定支援システムDSS** DSSは、組織の管理者が業務上意思決定を行なうときに、その業務遂行を支援するための計算機プログラムツールである。DSSに求められる一般的な機能はデータベース操作、モデル操作、統計予測分析、報告書作成、ビジネスグラフ作成などがある。

本研究にて研究開発したDSS-GOALは、EDPとOAの融合化、垂直分散型、統合化利用者視野の3点を特長としている。第1の特長は、オペレーショナルデータベースを用いるEDP業務の延長上にDSS-GOALを位置付け、更にDSS-GOALの運用形態をOA化することが可能となることである。第2の特長は、ホスト計算機と端末の間でのDSSの機能と資源の分散化である。ホスト側で、モデル作成、シミュレーション、統計予測、データ操作を実行し、一方端末側でビジネスグラフの作成、作表を実行する。従って、ホスト計算機と端末の間でデータベースやグラフのデータをファイル転送することにより、負荷の集中防止がはかれる。第3の特長は、DSSの各々の機能に対する利用者視野を統合化することにより、操作性の共通化と向上がはかれることである。

**データマイニング技術** 近年の計算機性能の向上、記憶装置の大容量、低価格化により、企業では膨大なデータを蓄え、処理することが可能となってきた。また、エンドユーザコンピューティング環境の浸透により、エンドユーザによる情報系業務に対するニーズが高まっている。このような、大量のデータをエンドユーザが意思決定支援などの情報系業務に活用しようとする手法としてOLAP<sup>注14</sup>やデータマイニング(Data Mining)がある。

データマイニングとは、データベースに蓄えられた大量のデータから企業活動において価値のあるルールを発掘(Mining)しようという技術である。データマイニング技術の実用化のキーワードとして、ルール発掘に要する時間の問題と、結果として獲得されるルールの価値に関する問題がある。前者は、言うまでもなく大量のデータ処理に費やされる時間の問題であり、後者は、重要度の低いルールを排斥し、いかにして有効なルールのみを獲得するか、という問題である。

本研究では有効なルールを獲得するためのアプローチとして、業務ノウハウやオントロジを利用したデータマイニング手法を検討している。ここで言う業務ノウハウとは、既に有効性が確認されているルールや専門家の知識を蓄積したものであり、対象とするルー

<sup>注14</sup> On-Line Analytical Processing

ルは相関ルールである。 相関ルールとは事象Aと事象Bがあったとき、事象Aが起きたとき同時に事象Bが起こる確率がY%であるとき、「事象AのときY%の確率で事象Bが同時に起こる。」というルールを相関ルールという。

本論文では、この業務ノウハウやオントロジを活用して、データマイニングサーバ Knodiasを研究試作したので、その結果を述べる。

## 3.2 使い易いデータベース利用者インタフェース技術

以下、3.2.1節ではデータベースの利用者インタフェースに求められる機能、3.2.2節では本研究にて研究試作したデータベースのユーザフレンドリインタフェースMELQUERY、3.2.3節ではこのMELQUERYを拡張した遠隔データベース・リクエストMELQUERY/R、そして3.2.4節と3.2.5節ではその操作例について述べていく。

### 3.2.1 データベースの利用者インタフェースに求められる機能

データベースを用いる業務は、

- 定型処理業務
- 非定型処理業務

に分類できる。前者はデータベースに対するバッチ处理的な利用形態が中心であるが、一方後者は対話处理的な利用形態が多い。

したがって、前者はCOBOLに代表される手続き型言語で作成した適用業務プログラムが適しているが、後者はワークステーションやパソコンなどを用いた対話形式の操作が適している。

一方、データベースアクセスの処理形態は

- 集中処理型
- 分散処理型

に分類できる。前者は大型ホスト計算機を用いる集中型データベースシステムであり、後者はワークステーションと大型ホスト計算機の組合せ、またはワークステーション同士の分散データベースシステムとして実現される。

この中で、今後の主流と予想されるワークステーションを基にした分散データベースシステムを用いて、非定型業務を処理するときに求められる使い易いデータベース・利用者インタフェースとしては、

- 操作する対象が視覚的に認識できる、
  - 操作の対象は機能に先行して選択できる、
  - 操作方法は対象の内容や存在位置のいかにかわらず、機能ごとに統一化している、
  - データベース操作機能が孤立しないで、データ加工や報告書作成の機能と統合化している、
- などが考えられる。

### 3.2.2 MELQUERY

この節は本研究にて研究試作したデータベースのユーザフレンドリ・インタフェースMELQUERYについて記述する。

3.2.2節に述べた要求を満足するために、3.2.1節に述べた技術動向の中で技術的完成度が高く、かつ実用化レベルに近付いているという点から視覚的インタフェースを選択し、MELQUERYを研究試作した。具体的にはMELQUERYはUNIX系エンジニアリング・ワークステーション上に搭載し、このワークステーションの提供するデスクトップ環境デスク

トップマネージャーの下で動作する。

デスクトップマネージャーの下で動作する応用サービスには、文書処理、電子ファイル、電子メール、ジオメトリック図形、グラフ処理などがあり、MELQUERYもこの応用サービスの一つ、すなわちデータベースサービスとして位置付けられる。

### (1) 特長

MELQUERYは次のような特長を有している。

#### ●第一の特長：

- ・オブジェクト先行型の利用者インタフェース、
- ・リレーショナル・データベースのデータ操作とパーソナルテーブルのデータ加工を行うための統一化された視覚的な利用者インタフェース。

データベースのデータ操作やパーソナルテーブルのデータ加工の際、表示されている操作対象の中から特定の操作対象を先にマウスでポインティング選択し、次に機能を選択した後実行へと進む。従って、利用者はデータベースやパーソナルテーブルの名称、内容、その中のデータ項目名、データ型、表示属性などを記憶してキーボード入力しなくともよい利点がある。

#### ●第二の特長：

- ・データベース操作内容の標準SQL言語<sup>[JIS87]</sup>への変換、
- ・リレーショナル・データベースのアクセス。

従って、MMI<sup>注15</sup> は下位のデータベース管理システムからのモジュール独立性が高いので、標準SQL言語を提供する任意のデータベース管理システムとの接続が容易となる利点がある。

### (2) 構成

図3.1はMELQUERYのソフトウェア構成であり、各構成要素の機能は次のとおりである。

#### MMI

MMIは、エンジニアリング・ワークステーションのマウス、メニュー、マルチウィンドウ、アイコンを介して利用者からの入力を受けて、操作対象の内容を表示したり、あるいは実行結果をウィンドウ上に表示したりする。

#### SQLi/f

SQLi/f<sup>注16</sup> は、利用者の入力した操作内容をSQL文に変換し、次のRDBASEi/f<sup>注17</sup> にそのSQL文の処理を依頼し、その実行結果をMMIへ戻す。

---

<sup>注15</sup> Man Machine Interface

<sup>注16</sup> SQL Interface

<sup>注17</sup> RDBASE Interface

## RDBASEi/f

RDBASEi/fは、SQLi/fから受け取ったSQL文を、次のRDBASE固有のデータ操作言語に変換し、RDBASEにそのデータ操作の実行を依頼し、その実行結果をSQLi/fへ戻す。

## リレーショナル・データベースシステムRDBASE

RDBASEは、エンジニアリング・ワークステーション上に提供するデータベース管理システムである。

## PTMS<sup>注18</sup>

パーソナルテーブルは、MELQUERYで用いる個人用スプレッドシートのことである。PTMSでは、MMIからの入力内容を受けて、パーソナルテーブルのデータ加工（例えば、表計算）を行う。

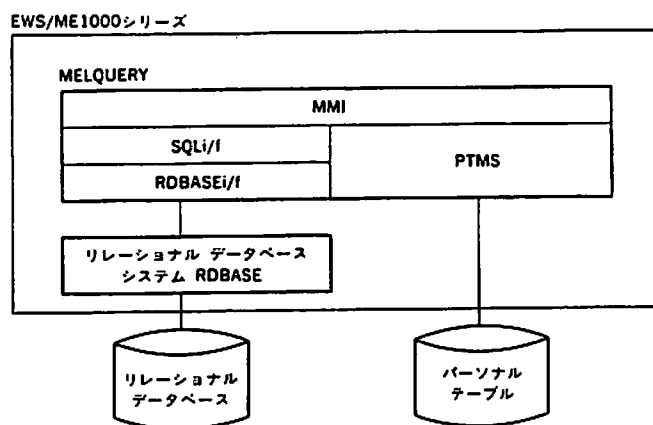


図3.1 MELQUERYのソフトウェア構成

### (3) 機能

MELQUERYは、UNIX系エンジニアリング・ワークステーションのローカル・データベースに対して、データの検索、挿入、更新、索引付け、回復などの機能を提供している。そして、その基本的な考え方は、操作対象であるオブジェクトと操作機能であるコマンドの組合せから成り立っている。

## オブジェクト体系

<sup>注18</sup> Personal Table Management System

MELQUERYの操作対象、すなわちオブジェクトは、

- リレーショナルデータベースのリレーション<sup>注19</sup> 全体集合を意味するDBBOXオブジェクト
  - パーソナルテーブルの全体集合を意味するPTBOXオブジェクト
  - 個々のリレーションを意味するDBオブジェクト
  - 個々のパーソナルテーブルを意味するPTオブジェクト
- がある。 これらのオブジェクトは図3.2に示すようにアイコンとして表示される。

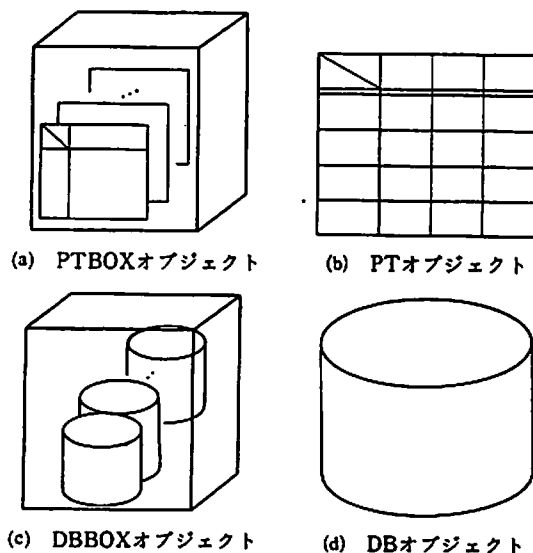


図3.2 MELQUERYオブジェクトのアイコン表示

### DBオブジェクト

DBオブジェクトのリレーションは、RDBASEのリレーションと同一の概念であり、このリレーションのアトリビュートに許されるデータ型は、日本語・文字型・整数型・実数型の4種類である。

### PTオブジェクト

PTオブジェクトのパーソナルテーブルは、DBオブジェクトに対するリレーショナル演算の結果（導出表）として二次記憶上に格納され、簡単な表計算が可能である。PTオブジェクトのパーソナルテーブルは、図3.3に示すような二次元表形式のデータ構造をしており、項目はデータ型に相当する物理属性、表示形式である表示属性などが定義される。

<sup>注19</sup> リレーショナルデータベースを構成する2次元形式の表データであり、選択、射影、結合などのリレーショナル演算が定義されている。



	C1	C2	C3
	商品コード	商品名	在庫量
L1	1004	D	0
L2	2011	K	4
L3	1002	B	2
L4	1001	A	5
L5	1003	C	3
L6	0001	I	10

(a) データ

	C1	C2	C3
項目名	商品コード	商品名	在庫量
物理属性	S.5c	1.3c	S1
表示属性	RX	RX	RX
意味属性			
価値属性	RJ	RJ	RJ

(b) 属性

図3.3 MELQUERYのパーソナルテーブル

DBオブジェクトのリレーションは、アトリビュートやタプルの順序には意味がないが、PTオブジェクトのパーソナルテーブルは表計算の結果（例えば、合計値や平均値）を格納する項目や行を特定化する必要性が高いので、項目や行には識別番号を定義してその順序に意味を持たせている。

例えば、PTオブジェクトのパーソナルテーブルの第一番目の行はL1、第二番目の行はL2、一方第一番目の項目はC1、第二番目の項目はC2のように識別番号を割り付ける。

すべてのオブジェクトは、リレーションの名称とアトリビュート名、パーソナルテーブルの名称と項目名などがマウスを用いたウィンドウ間転送機能により、キーボードを介さずに指定できるので、誤入力の減少が大きな利点である。

### コマンド体系

MELQUERYのコマンドは、

- 一般コマンド
- 基本コマンド

に分類できる。表3.1にコマンド体系を示す。

表3.1 MELQUERYのデータ操作コマンド

コマンド 種類	コマンド名	コマンド機能	オブジェクト			
			DBBOX	PTBOX	D B	P T
基本コマンド	LIST	BOX内に存在するPTとリレーションの一覧表示	○	○	×	×
	DISPLAY	PTとリレーションの表示	×	×	○	○
	DROP	PTとリレーションの削除	×	×	○	○
	STORE	PTとリレーションのBOXへの格納	×	×	○	○
一般コマンド	SELECT	リレーションの検索	×	×	○	×
	PROJECT	リレーションの射影	×	×	○	×
	JOIN	リレーションの結合	×	×	○	×
	SORT	リレーションの分類	×	×	○	×
	CREATE	リレーションの生成	×	×	○	×
	INSERT	リレーションの挿入	×	×	○	×
	UPDATE	リレーションの更新	×	×	○	×
	CREATE INDEX	索引の生成	×	×	○	×
	DROP INDEX	索引の削除	×	×	○	×
CALCU	PTの簡易表計算	×	×	×	○	
その他	PRINT	リレーションの印刷	×	×	○	×
	ACCESS	リレーションのアクセス権限定	×	×	○	×
	RECOVER	リレーションの回復	×	×	○	×

注 ○：オブジェクトとして指定可能  
 ×：オブジェクトとして指定不可能

## 基本コマンド

基本コマンドはリレーション名やパーソナルテーブル名の一覧表示や内容表示などの基本機能を提供する。

具体的なコマンド名は、一覧表示LIST、表示DISPLAY、削除DROP、格納STOREである。一覧表示コマンドは、DBBOXオブジェクトとPTBOXオブジェクトが操作対象であり、各々データベースのリレーション名とパーソナルテーブルの名称などを一覧表示する。表示コマンドは、DBオブジェクト又はPTオブジェクトのデータと属性を表示する。削除コマンドは、DBオブジェクト又はPTオブジェクトを削除する。格納コマンドは、DBオブジェクトとPTオブジェクトを各々DBBOXとPTBOXへ収納する。

## 一般コマンド

一般コマンドは、データベースに対するリレーショナル演算機能とパーソナルテーブルに対する簡単な表計算機能を提供する。この一般コマンドは、MELQUERYの中心的な機能である。具体的な一般コマンド名は、DBオブジェクトに対するリレーショナル演算として選択SELECT、射影PROJECT、結合JOIN、生成CREATE、更新UPDATE、挿入INSERT、分類SORT、及び索引の生成CREATE INDEXと削除DROP INDEXなどの機能を提供する。一方、PTオブジェクトに対しては、簡易表計算CALCUの機能を提供する。

基本コマンドと一般コマンドは、既に選択したオブジェクトの種類や個数に対応したメニューやフォームからマウスを用いて選択する。

## その他

DBオブジェクトのリレーションに対するアクセス権限定義ACCESS、回復RECOVER、印刷PRINTの機能を提供する。印刷はDBオブジェクトのウィンドウ上の出力ボタンをクリックして実行する。

尚、基本コマンドと一般コマンドは同時実行可能であるが、一般コマンド同士は同時実行不可能である。

### (4) オブジェクトとコマンドの対応付け

オブジェクトとコマンドの間の対応付けを整理すると、図3.4に示すようになる。すなわち、

- DBオブジェクトの空リレーションを用いて新しいリレーションをCREATEすると、DBオブジェクトのリレーションとなる。
- DBBOXオブジェクトをLISTすると、存在するリレーション名がすべて一覧表示されるので、その中からリレーション名を指定するとDBオブジェクトが生成される。
- このDBオブジェクトに対するリレーション演算による導出結果はPTオブジェクトに変換して、PTBOXオブジェクトにSTOREできる。
- また、このPTオブジェクトはCALCUすることも可能であり、その結果もPTBOXオブジェクトへSTOREできる。
- 一方、PTBOXオブジェクトをLISTすると、PTBOXオブジェクト内にのみ存在するパーソナルテーブル名がすべて一覧表示されるので、その中からパーソナルテーブル名を指定するとPTオブジェクトが生成される。
- このPTオブジェクトもCALCUすることが可能である。

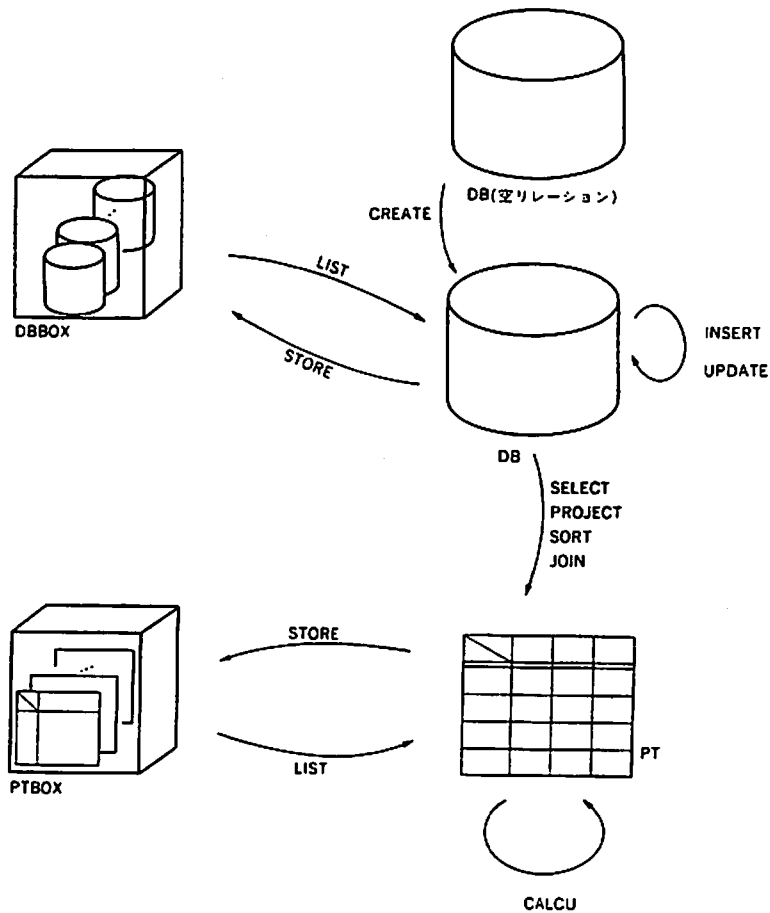


図3.4 MELQUERYにおけるオブジェクトとアイコンの関係

### 3.2.3 MELQUERY/R

図3.5にMELQUERY/Rのシステム構成を示す。

MELQUERY/Rは、ホストコンピュータのデータベースに対してアクセスを行うデータベースサービスであり、UNIX系エンジニアリング・ワークステーションの提供するデスクトップ環境、デスクトップマネージャの下で動作する。デスクトップ環境の応用サービスとしては、文書処理、電子ファイル、電子メールなどがあり、MELQUERY/Rもこの応用サービスの一つに位置づけられる。

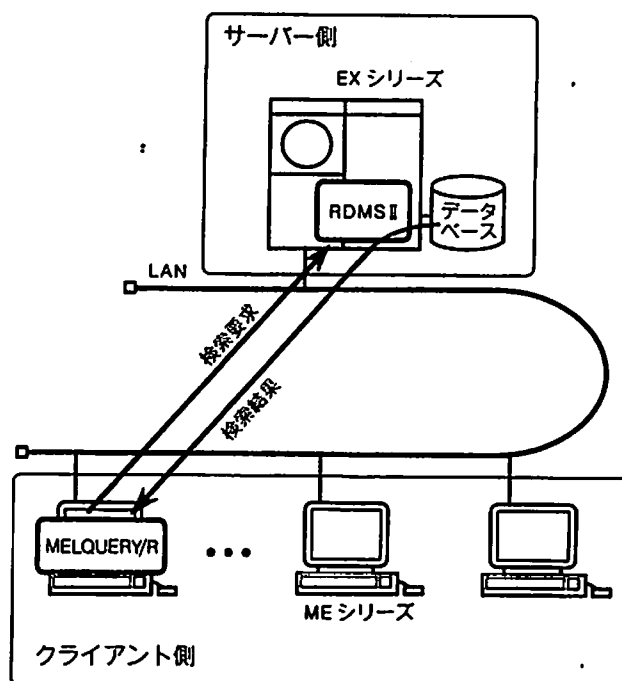


図3.5 MELQUERY/Rのシステム構成

### (1) 特長

MELQUERY/Rは次のような特長を持っている。

- UNIX系エンジニアリング・ワークステーションの機能を生かした、マルチウィンドウ、アイコン、メニュー、フォームによる画面表示やマウス操作により、視覚的な対話形式でホストコンピュータのデータベースの検索が実行できる。 コマンド操作は、表示中の指定可能な操作対象の中から目的のアイコンを指示し、それを契機にシステムが提示するメニューやフォームにこたえていくことによる対話形式で進んでいく。 そのため、利用者はデータベースの詳細な情報やコマンドの構文規則を記憶しておく必要はなく、気軽に操作を行うことができる。
- MELQUERY/Rとデータベースとのインターフェースには、文献[JIS87]で規格化されているリレーショナル・データベースの操作言語SQLで行われる。 すなわち、利用者から入力された操作内容は、いったんSQL文に変換されてからデータベースへアクセスするため、接続されるデータベース管理システムとの独立性が高い。
- UNIX系エンジニアリング・ワークステーションとホストコンピュータとの間の通信プロトコルとして、実質的な業界標準であるTCP/IP<sup>注20</sup>と、ネットワークアーキテクチャ規約に基づくMNA-P<sup>注21</sup>をサポートしている。

<sup>注20</sup> Transmission Control Protocol/Internet Protocol

<sup>注21</sup> Multi-shared Network Architecture-Protocol

●UNIX系エンジニアリング・ワークステーション上のアプリケーションからホストコンピュータのデータベースを利用するためのアプリケーションインタフェースを提供している。アプリケーションの記述言語としては、C言語とFORTRAN言語の二つを用意している。これにより、プログラム中でデータベースの検索を行うアプリケーションが容易に作成できる。

(2) 構成

MELQUERY/Rのアクセス対象のデータベースは、ホストコンピュータのリレーショナル・データベース管理システムRDMSIIである。このRDMSIIをOAサーバ・リクエストインタフェース(OAIF-R,OAIF-S)のサービスプログラムの一つであるリレーショナル・データベースサーバRQL/DSから利用する。

MELQUERY/Rの動作は次の通りである。

- マウス操作などによる利用者からの入力はSQL文に変換され、一般化文書交換規約GIAのコマンドに組み込まれてOAリクエストインタフェース(OAIF-R)に送られる。
- OAIF-Rは、ASN.1<sup>注22</sup>で定義されている転送構文形式でOAサーバインタフェース(PAIF-S)にコマンドの送信を行う。
- OAIF-Sは、受信したコマンドをRQL/DSに渡し、RQL/DSはRDMSIIに処理を依頼する。
- 実行結果は、コマンドとは逆の順序でMELQUERY/Rに戻り、ウィンドウ上に表示される。

MELQUERY/Rのソフトウェア構成を図3.6に示す。MELQUERY/Rの各モジュールの機能は次の通りである。

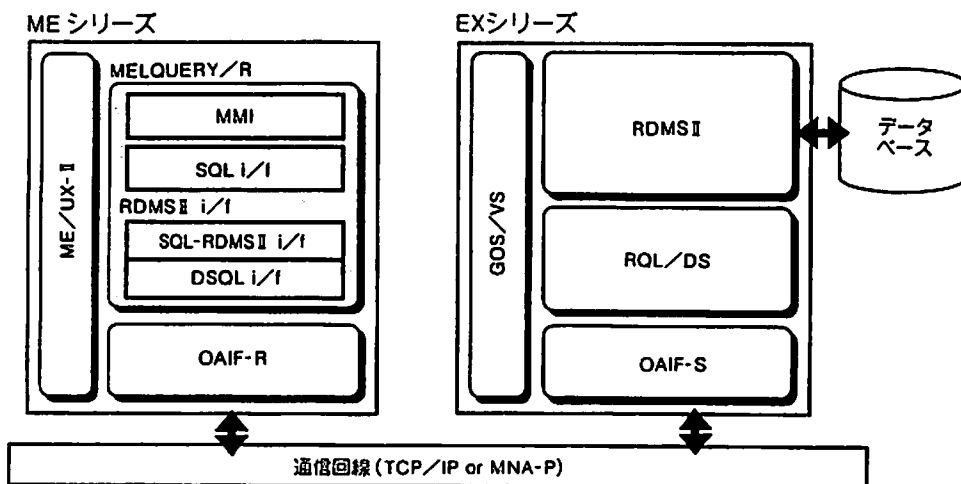


図3.6 MELQUERY/Rのソフトウェア構成

注22 Abstract Syntax Notation 1

## MMI

利用者からの入力を受け付け、その情報を次のSQLi/fに渡す。また、実行結果をSQLi/fから受け取り、ウィンドウ上に表示する。

## SQLi/f

利用者の入力情報からSQL文を生成し、次のRDMSIi/f<sup>注23</sup> にそのSQL文の処理を依頼する。また、RDMSIi/fから実行結果を受け取り、それをMMIへ返す。

## RDMSIi/f

OAIF-R、OAIF-SによるROS<sup>注24</sup> のメッセージ交換サービスを用いてサーバとの通信を行う。RDMSIi/fは、次の二つのモジュールから構成されている。

### ・SQL-RDMSIi/f

SQL i/fから受け取ったSQL文をもとに、DSQLi/fを使って、サーバ と通信を行い、実行結果をSQLi/fへ返す。

### ・DSQL i/f

SQL-RDMSIi/fで呼び出された関数をGIAに基づくコマンドに変換し、OAIF-Rに送る。そして実行結果を受け取り、SQL-RDMSIi/fへ返す。このモジュールはライブラリ形式になっており、ここで応用プログラム インタフェースを提供している。

## (3) 機能

ホストコンピュータのデータベースに対してMELQUERY/Rから行える機能は、データの検索処理である。

一般にデータベースの機能にはデータの構造や属性を定義するデータ定義機能、データの検索・更新を行うデータ操作機能、データ整合性の維持や障害時の回復などを行うデータ制御機能があるが、MELQUERY/Rではデータ検索のみをサポートしている。これはMELQUERY/Rを共通情報を一元管理するようなネットワークシステムの基幹データベースをアクセスするための、エンドユーザ向け利用者インタフェースとして位置づけたためである。

データベースの管理・運用は、エンドユーザの役割ではないため、データ定義や制御はデータベース管理者が直接データベース管理システムのコマンドを利用して行い、MELQUERY/Rの利用者には提供しない。

また、データの更新処理については、MELQUERY/Rのような非定型業務を目的とした場合、更新処理の起動から実行までの時間が長くなると予想され、並行処理制御上の問題

---

<sup>注23</sup> RDMS II Interface

<sup>注24</sup> Remote Operation Service

が大きいことと、エンドユーザによる誤更新を防止する手段が現在ないこと等から、MELQUERY/Rの機能から除外した。

検索対象のデータは、幾つかのリレーションごとにデータベースとして定義され、データベース単位で運用される。

MELQUERY/Rが利用者に提供するリレーションの定義情報は、リレーション名、所有者名、属性情報（アトリビュート名、物理属性、サイズ）である。

MELQUERY/Rのデータ操作コマンドは、表3.2に示す通りである。

### 一覧表示

ホストコンピュータのデータベースの中から利用者がアクセス権を保有するリレーションの名称と所有者名の一覧表示を行う。

### 射影・選択・分類

利用者の設定した射影条件、選択条件、分類条件をもとに、リレーションからデータの検索を行う。 コマンドを起動すると、まず操作対象のリレーションの属性情報又はデータの内容の表示が行われ、それをもとに条件式の入力を行うことができる。

### 結合

リレーションの結合を伴うデータの検索を行う。

### 表示（データ・属性）

リレーションの全データ又はリレーションの属性情報を表示する。

### ファイル変換

表示中のデータをUNIX系エンジニアリング・ワークステーション上にファイルとして保存する。これにより、他のアプリケーションにデータを取り込むことが可能である。

### 格納

表示中のRDBアイコンをウィンドウから消去する。

## (4) 操作体系

MELQUERY/Rにおいて利用者が行う操作は、

●キーボードからの入力操作

●マウスによる操作

の二つである。 マウスは二つのボタンのマウスを使用する。

マウス操作には、

●クリック（左ボタン）

●ダブルクリック（左ボタン）



- プレス&リリース（右ボタン）  
の3種類がある。

表3.2 MELQUERY/Rのデータ操作コマンド

コマンド名	コマンド機能
一覧表示	RDBBOX 内リレーション名の一覧表示
射影	リレーションの射影(project)
選択	リレーションの選択(select)
分類	リレーションの分類(sort)
結合	リレーションの結合(join)
表示(データ・属性)	リレーションのデータ/属性の表示
ファイル変換	ME シリーズ上でのデータの保存
格納	RDB アイコンの画面から消去・保存

操作対象としては、

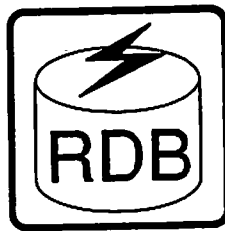
- アイコン
  - ウィンドウ
  - フォーム
  - メニュー
- がある。

図3.7はMELQUERY/Rで使用するアイコンである。

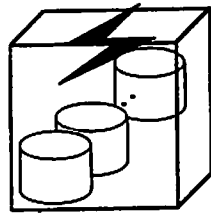
- MELQ処理アイコンはMELQUERY/R起動用アイコンであり、デスクトップ環境の他のサービスアイコンと同格で常時表示されている。
- RDBBOXアイコンとRDBアイコンは、MELQUERY/Rが動作している間、MELQアイコンウィンドウ上に表示される。
  - ・RDBBOXアイコンはデータベースを表し、MELQUERY/R動作中常に表示されている。
  - ・RDBアイコンは個々のリレーションを表し、リレーションの一覧表示のとき名称をクリックすると、その名称のRDBRアイコンが表示される。

マウス操作は、

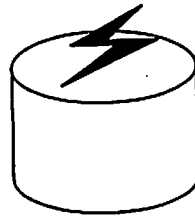
- アイコン
  - メニュー
  - ウィンドウ、フォーム上のボタン
- を対象に行う。キーボード入力は、検索条件式の指定などに用いる。



(a) MELQ 処理アイコン



(b) RDBBOX アイコン



(c) RDB アイコン

図3.7 MELQUERY/Rのアイコン

表3.3 MELQUERY/R操作体系

操作対象	操作	操作内容
MELQ 処理アイコン	ダブルクリック	MELQUERY/R 起動
MELQ アイコンウインドウ		
ボタン	クリック	MELQUERY/R 終了
RDBBOX アイコン	クリック	一覧表示
RDB アイコン	クリック	メニュー表示
	プレス&リリース	表示・格納
メニュー	クリック	射影・選択・分類, 結合
表示ウインドウ(データ・属性)		
ボタン	クリック	スクロール, ファイル変換, ウインドウ消去
フォーム		
ボタン	クリック	コマンド実行, 終了, 入力取消, 条件式指定
入力域	キーボード入力	条件式指定
	ウインドウ間転送	条件式指定

また、検索条件式の指定などにおいても、画面に表示されている項目名やデータ、ボタンで用意されている演算子などは、マウスのクリックによるウインドウ間転送で入力することができる。

表3.3にMELQUERY/Rの操作体系を示す。

### (5) 応用プログラムインタフェース

MELQUERY/Rでは、ホストコンピュータのデータベースを応用プログラムから検索するためのインタフェースとして、11種類の関数を用意している。

利用者はC言語又はFORTRAN言語で記述したプログラムからこれらの関数を呼び出すことにより、ホストコンピュータのデータベースを検索することができる。

提供する関数の一覧を表3.4に示す。

表3.4 MELQUERY/Rの応用プログラムインタフェースの関数一覧

C	FORTRAN	機 能
control Session ()	D_CSSN	データベースとの接続/切断
dynamic Prepare ()	D_PP RP	SQL コマンドの登録
dynamic Declare Cursor ()	D_DCUR	カーソルの定義
dynamic Open ()	D_OPEN	SQL コマンドの実行
dynamic Describe ()	D_DCRB	検索結果属性の取り出し
dynamic Fetch ()	D_FTCH	検索結果データの取り出し
dynamic Close ()	D_CLOS	検索の後処理

この応用プログラムインタフェースでは応用プログラムの実行時において、SQL文を動的に生成して実行を行うことができる。

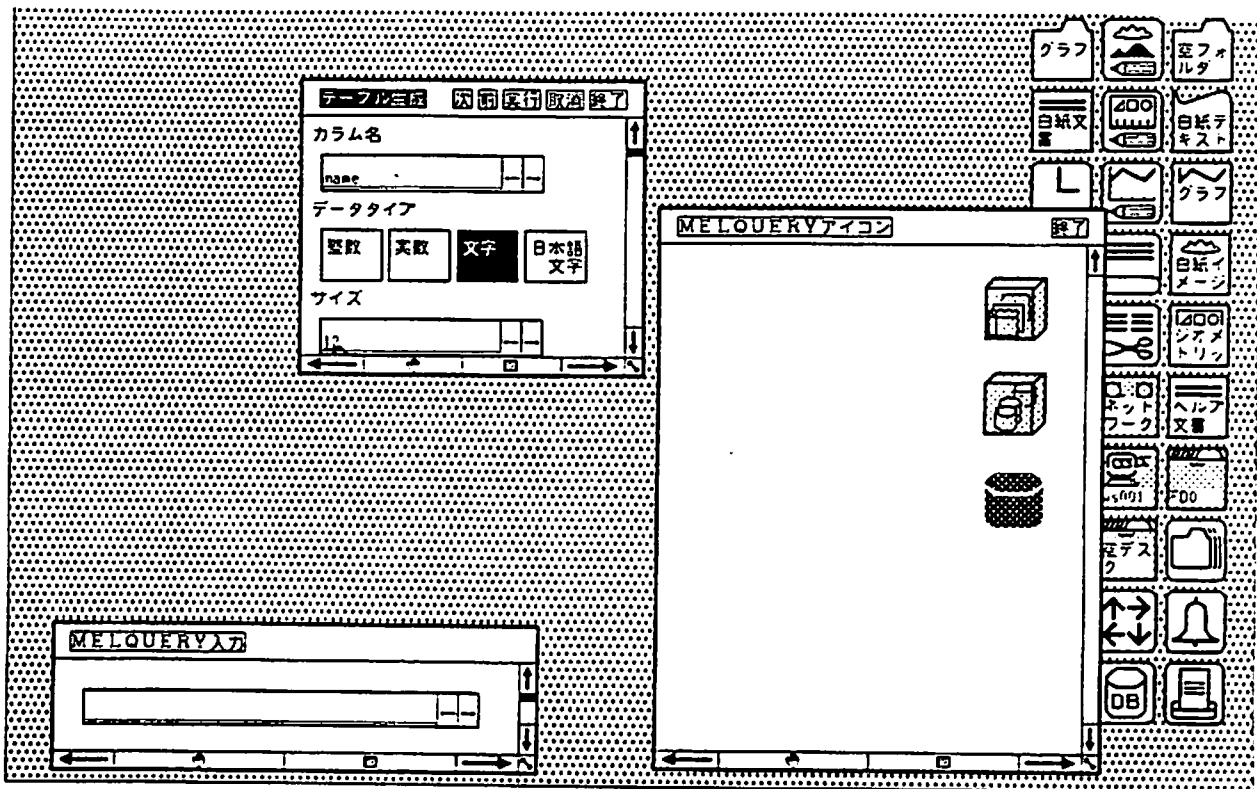
#### 3.2.4 MELQUERYの操作例

この節はデスクトップ環境からデータベースサービスMELQUERYを呼び出して、リレーショナル・データベースをアクセスする操作の例を示す。

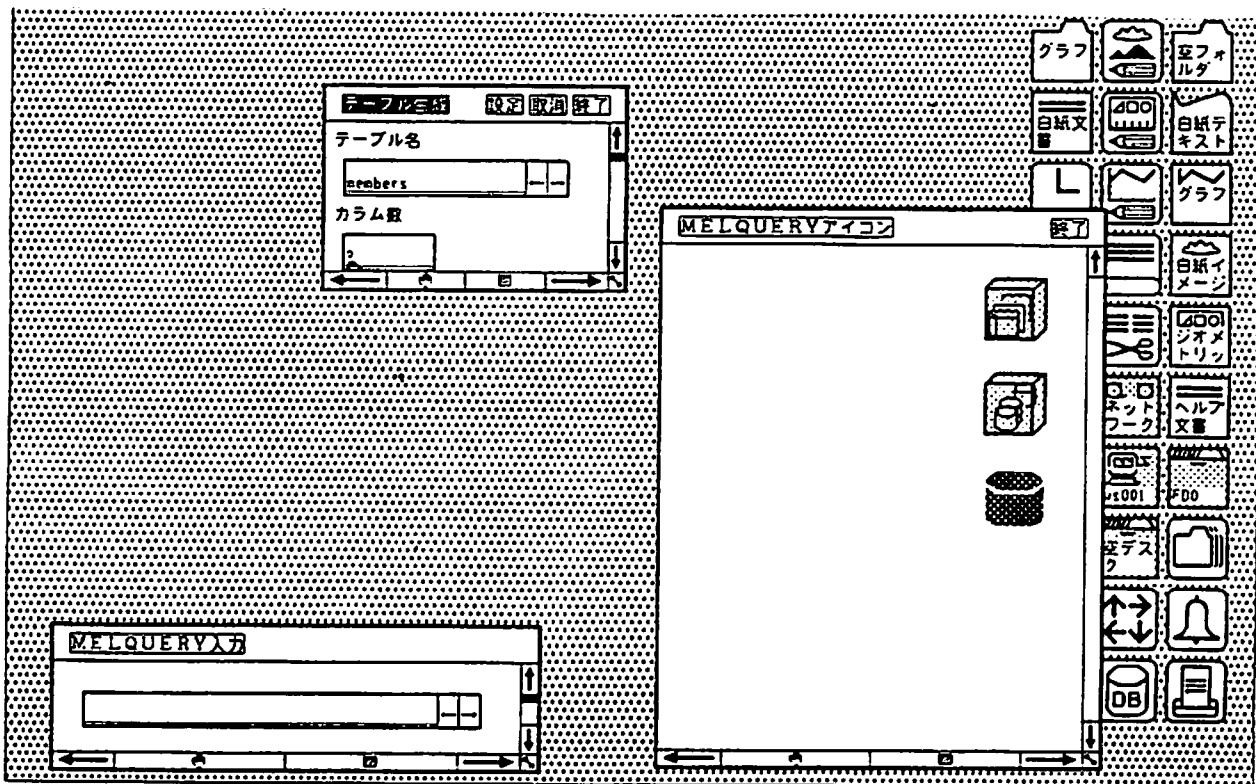
##### (1) リレーシヨンの生成

アトリビュートがname,address,balanceから成るリレーシヨンmembersを新規に生成する例を図3.8に示す。

この例ではDBオブジェクトのアイコン（空リレーシヨン）をマウスでクリックしてオブジェクトを指定した後、リレーシヨンの名称（テーブル名）、アトリビュートの個数（カラム数）、各アトリビュートの名称（カラム名）、データ型（データタイプ）、データ長（サイズ）を入力している。



(b) カラム名の定義



(a) テーブル名の定義

図3.8 MELQUERYのリレーション生成画面

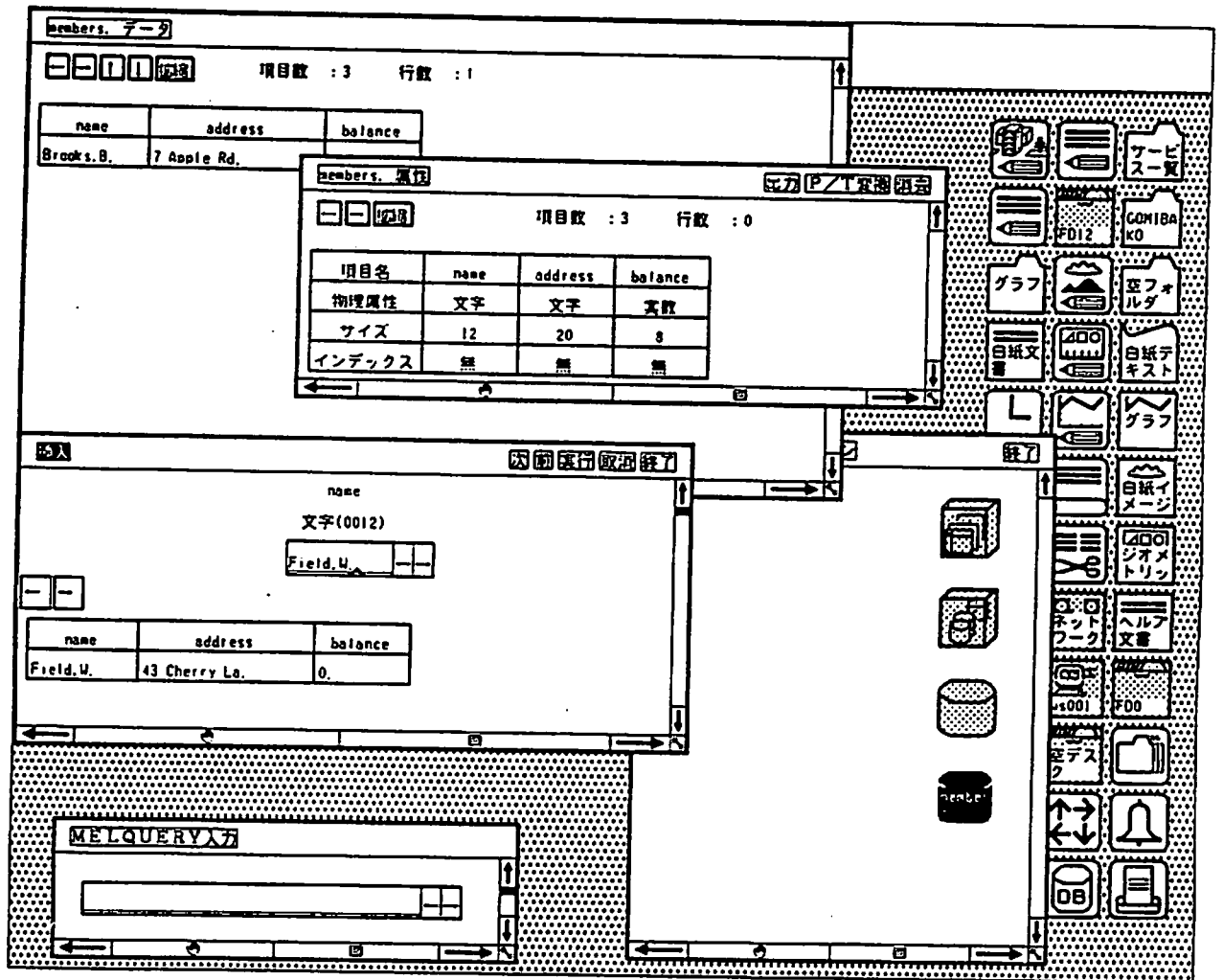


図3.9 MELQUERYのリレーション挿入画面

### (2) リレーションの挿入

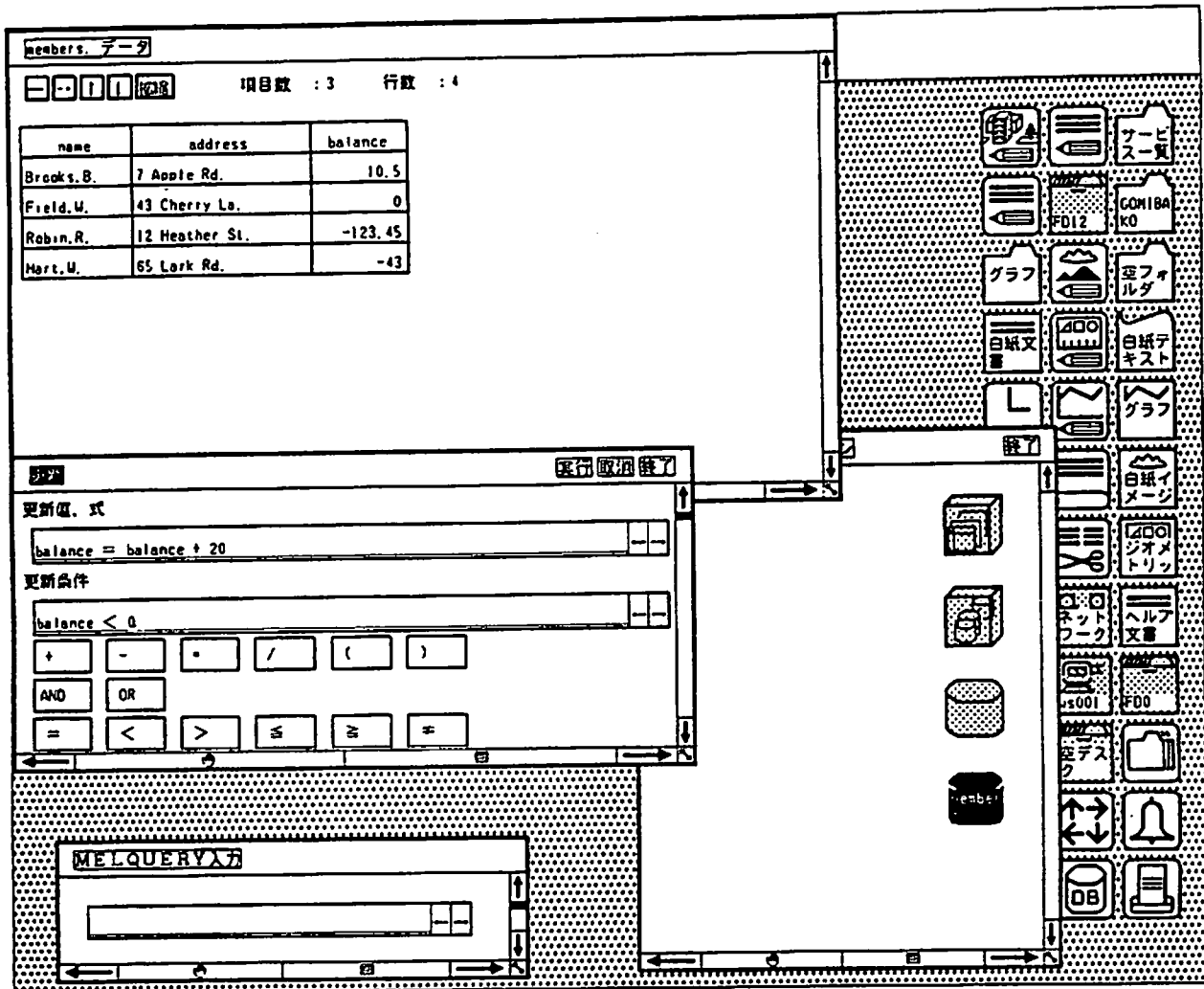
リレーションmembersにアトリビュートnameがField, W.,アトリビュートaddressが43Cherry La.,アトリビュートbalanceが0.0であるタプルを挿入する例を図3.9に示す。

この例ではDBオブジェクトのアイコン(members)をマウスでクリックしてオブジェクト指定した後、タプル値を入力している。

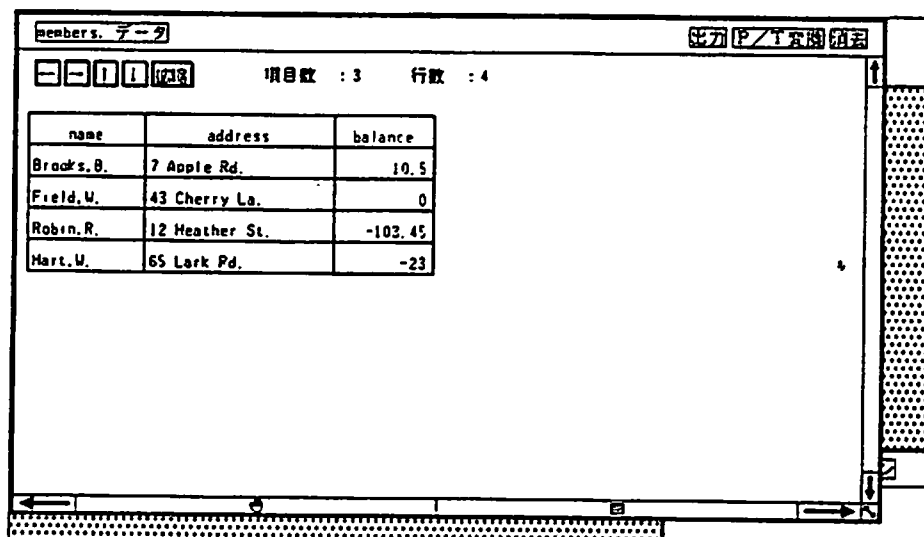
### (3) リレーションの更新

リレーションmembersのデータの中でアトリビュートbalanceの値が負であるデータを選び、そのアトリビュートbalanceの値に20を加算して更新する例を図3.10に示す。

この例ではDBオブジェクトのアイコン(members)をマウスでクリックしてオブジェクト指定した後、更新条件式として  $balance < 0$ 、また更新式として  $balance = balance + 20$  をマウスによるウィンドウ間転送機能とキーボードにより入力している。



(a) 更新条件式の指定



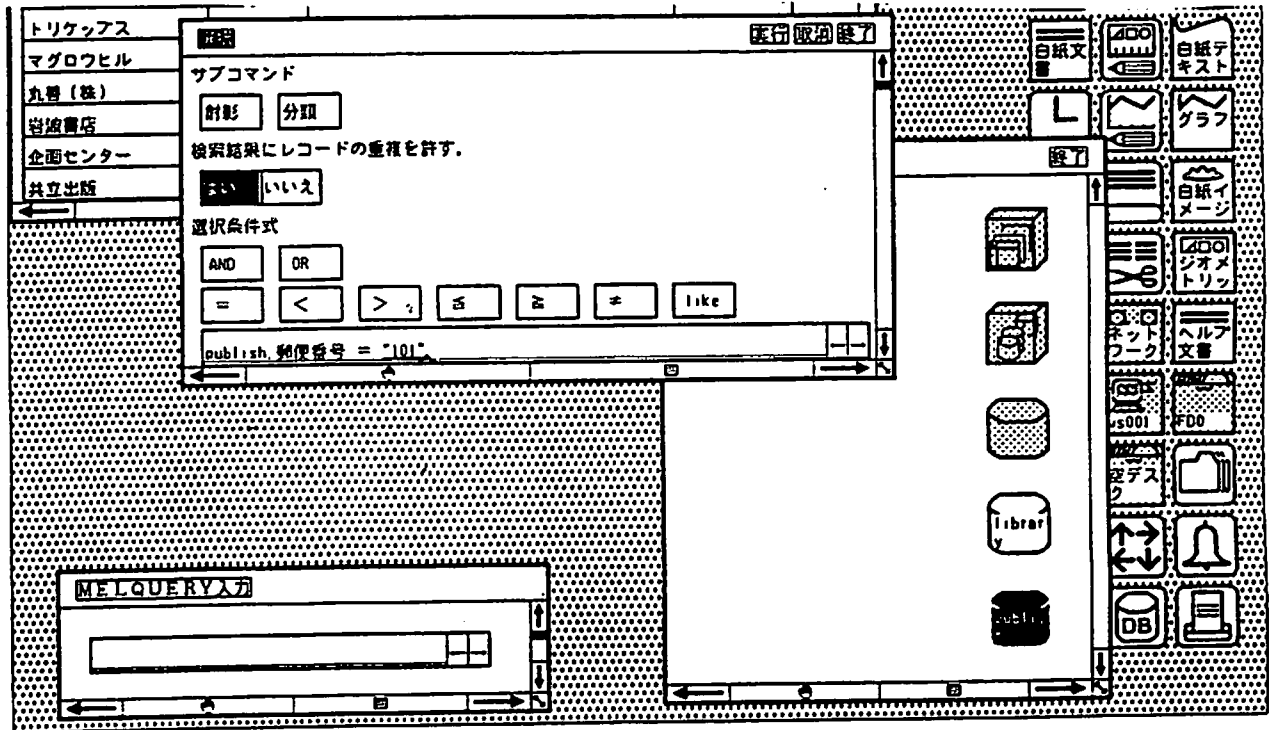
(b) 更新結果の表示

図3.10 MELQUERYのリレーション更新画面

#### (4) リレーションの検索

リレーションpublishから郵便番号が101番であるデータを検索する例を図3.11に示す。

この例ではDBオブジェクトのアイコン(publish)をマウスでクリックしてオブジェクト指定した後、選択条件式として *publish.郵便番号="101"* をマウスによるウィンドウ間転送機能とキーボードにより入力している。



(a) 選択条件式の指定

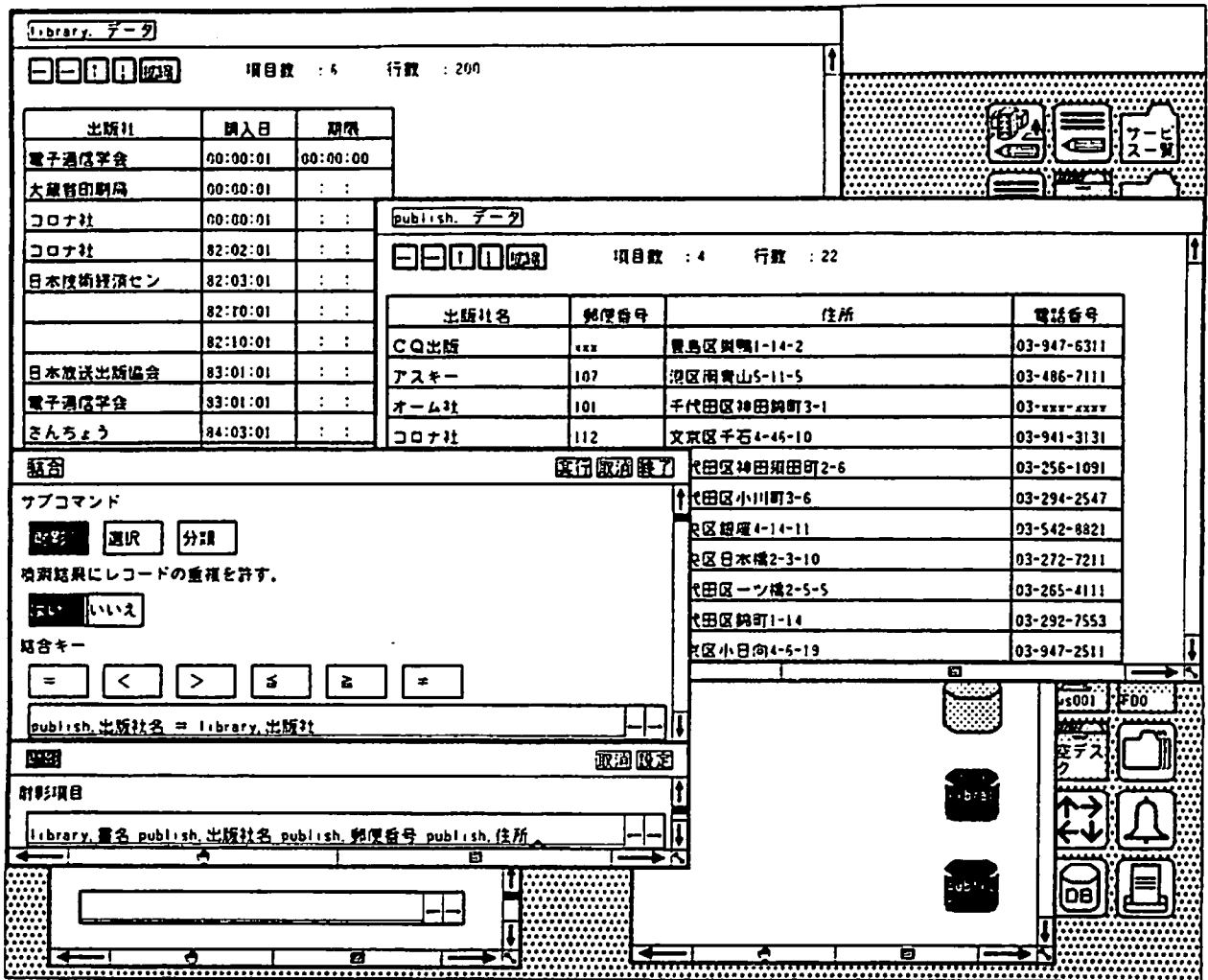
検索結果: データ

項目数 : 4 行数 : 8

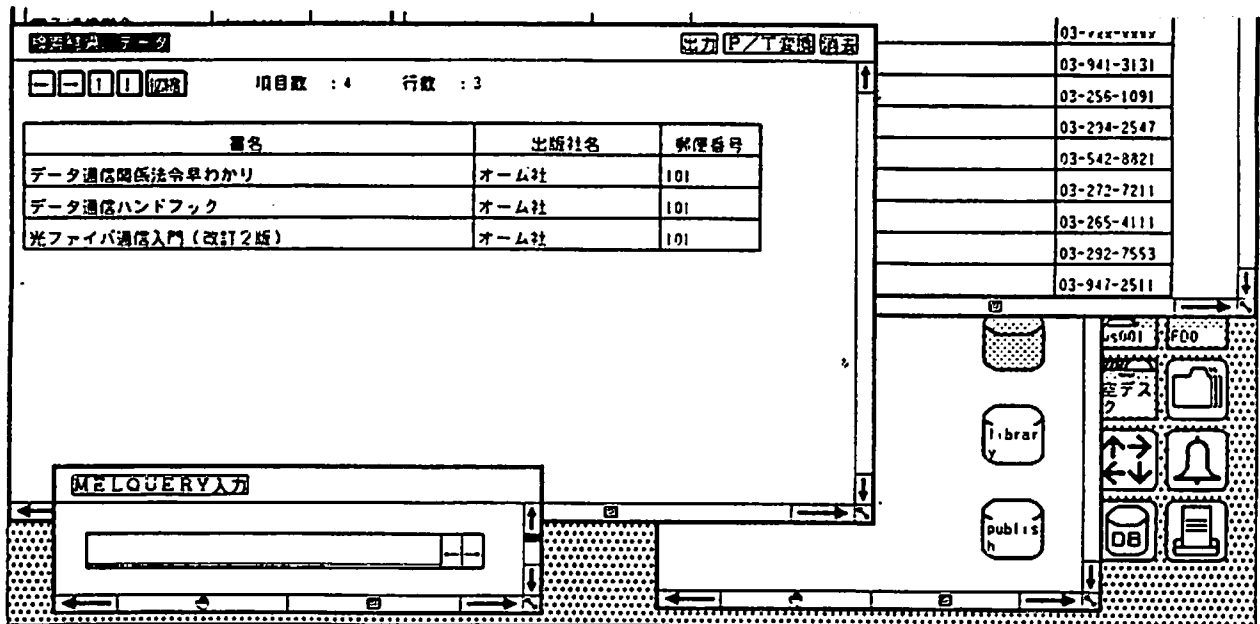
出版社名	郵便番号	住所	電話番号
岩波書店	101	千代田区一ツ橋2-5-5	03-265-4111
啓学出版	101	千代田区神田神保町1-46	03-233-3731
オーム社	101	千代田区神田錦町3-1	03-xxx-xxxx
サイエンス社	101	千代田区神田須田町2-6	03-256-1091
電気通信学会	101	千代田区富士見町2-8	03-331-7348
産経	101	千代田区神田佐久間町1-11	03-253-0111
企画センター	101	千代田区錦町1-14	03-292-7553
トリケッパス	101	千代田区小川町3-6	03-294-2547

(b) 検索結果の表示

図3.11 MELQUERYのリレーション検索画面



(a) 結合キーと射影項目の指定



(b) 結合・射影の結果の表示

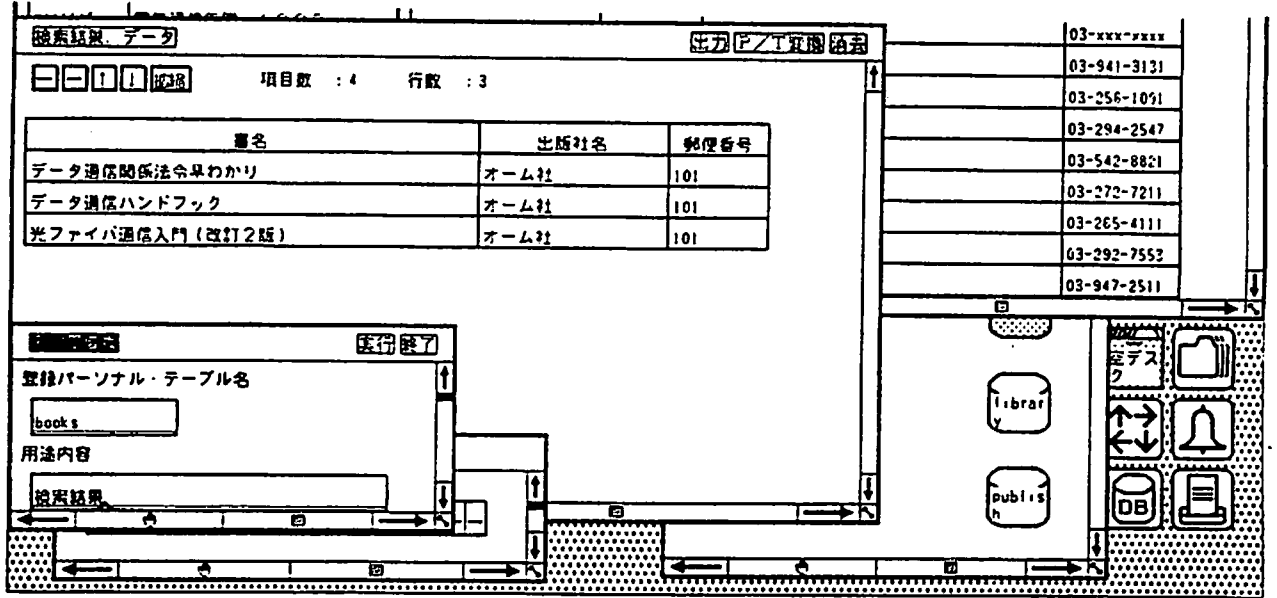
図3.12 MELQUERYのリレーション結合・射影画面



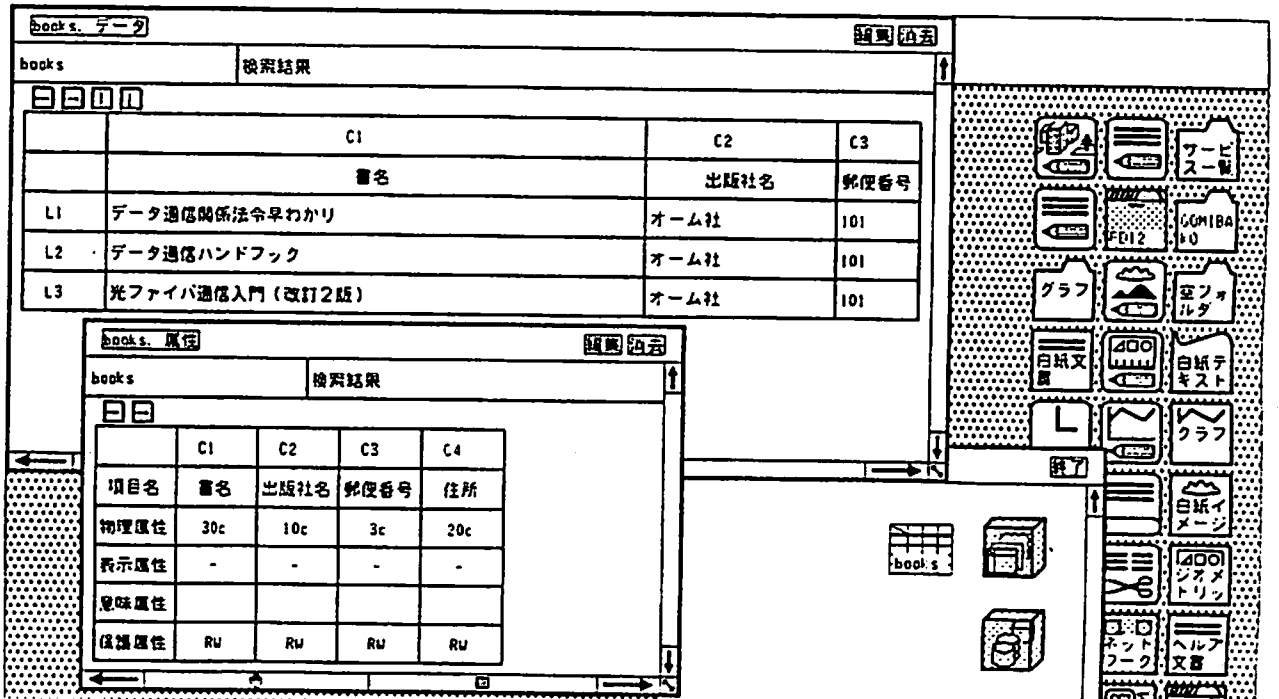
(5) リレーションの結合と射影

二つのリレーションlibraryとpublishを共通する出版社の名称で結合する例を図3.12に示す。

この例ではDBオブジェクトのアイコン(libraryとpublish)をマウスでクリックしてオブジェクト指定した後、結合キーとして、publish.出版社名=library.出版社、また射影するアトリビュート名としてlibrary.書名、publish.出版社名、publish.郵便番号、publish.住所をマウスによるウィンドウ間転送機能とキーボードにより入力している。



(a) パーソナルテーブル名の指定



(b) パーソナルテーブルの生成

図3.13 MELQUERYのパーソナルテーブル変換画面

### (6) パーソナルテーブルへの変換

図3.12に示した結合結果のリレーションをパーソナルテーブル(books)として保存する例を図3.13に示す。

### 3.2.5 MELQUERY/Rの操作例

MELQUERY/Rを用いたホストコンピュータのデータベース検索の例として、結合コマンドの操作を示す。

#### (1) 起動

UNIX系エンジニアリング・ワークステーションのデスクトップ環境からMELQUERY/R起動後の初期画面を図3.14に示す。

MELQUERY/Rは起動すると、まず無条件にデータベースの一覧表示を実行する。この一覧表示の中から操作対象のリレーションの名称をマウスでクリックすると、リレーションをRDBアイコンとして表示する。

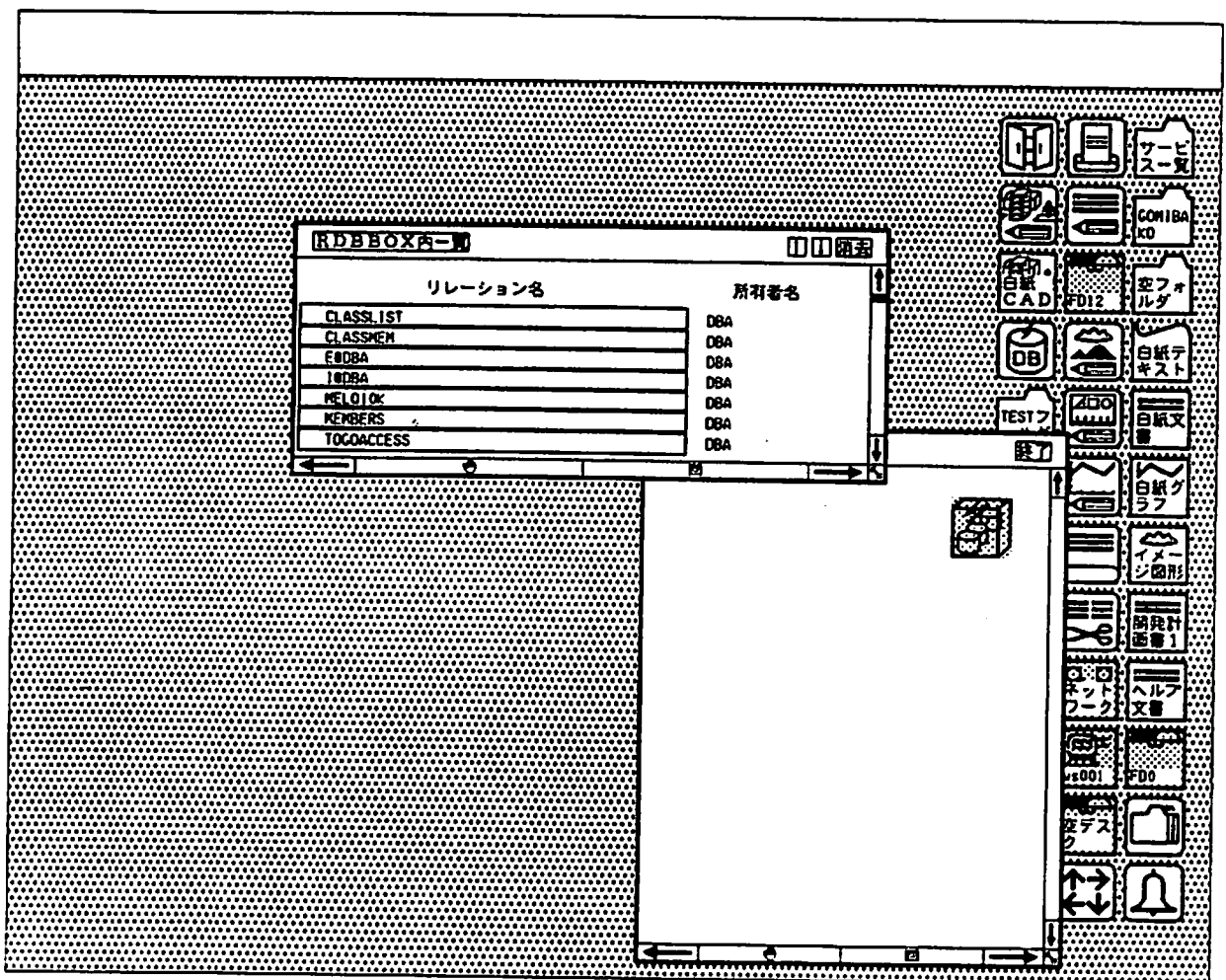


図3.14 MELQUERY/Rの起動画面

## (2) コマンド指定

RDBアイコンに対するコマンドの指定画面を図3.15に示す。

操作対象のリレーションのRDBアイコンをマウスでクリックすると、アイコンの表示が反転してコマンドのメニューが表示される。

結合の場合は、複数のRDBアイコンをそれぞれクリックするとメニューが現われる。この例では、二つのリレーションmemclassとclassmemをクリックしたときの結合コマンドのメニューである。

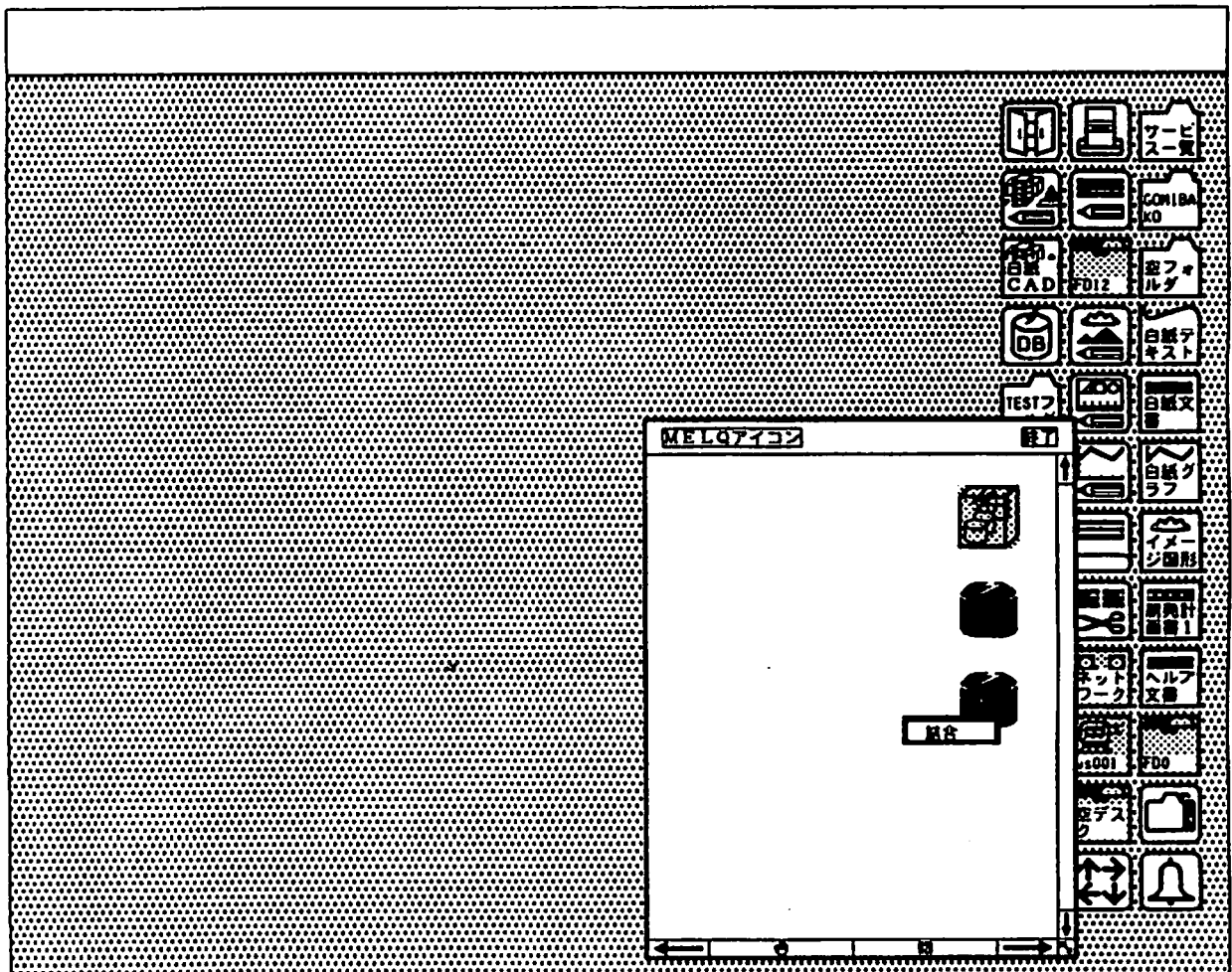


図3.15 MELQUERY/Rのコマンド指定画面

### (3) 検索条件式指定

結合条件式を指定する画面を図3.16に示す。

コマンドの起動により、まず対象のリレーションmemclassとclassmemの属性情報が表示され、次に条件入力用のフォームが表示される。このフォームに結合条件式 `memclass.code = classmem.code` を指定している。

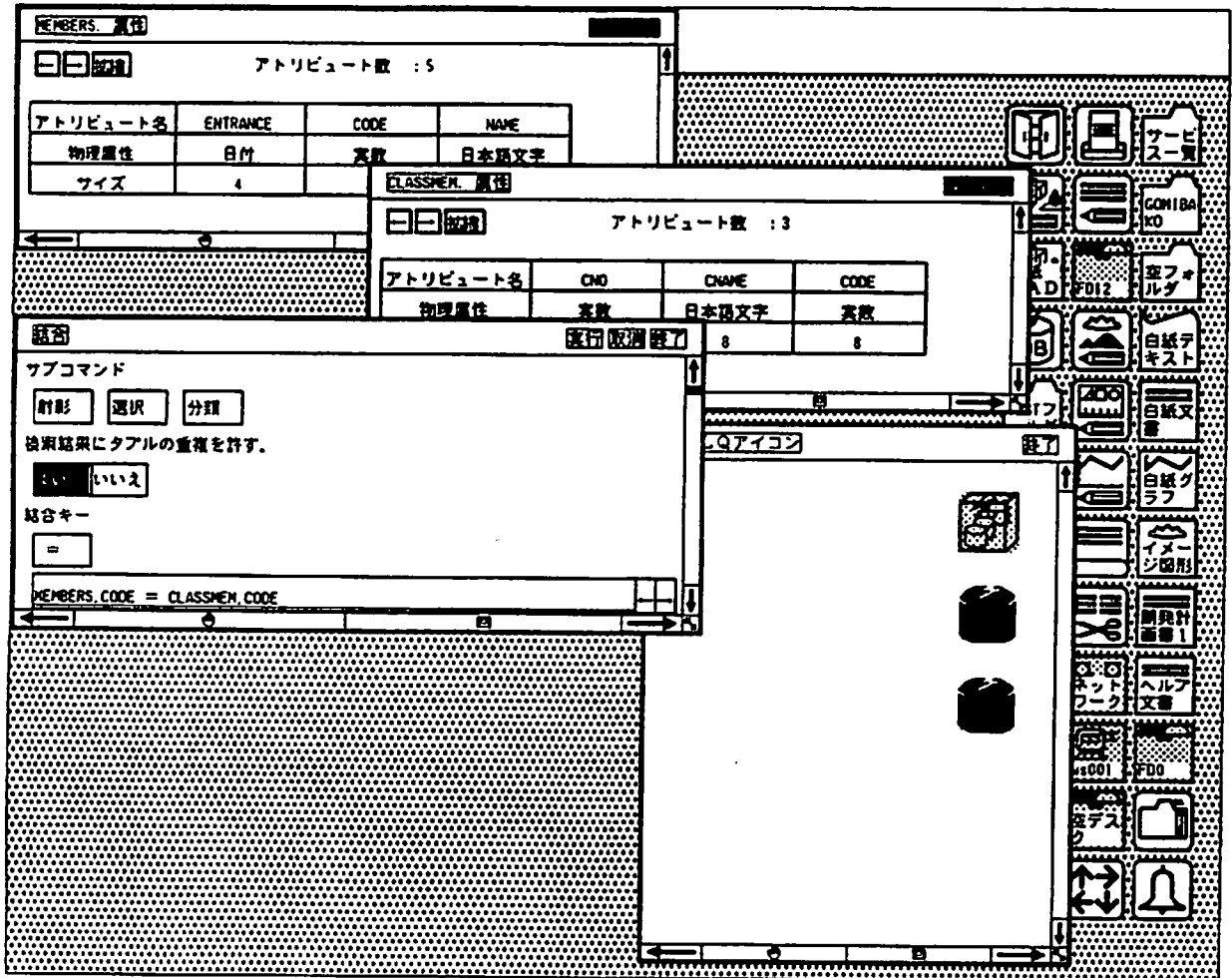


図3.16 MELQUERY/Rの条件式指定画面

(4) 検索結果表示

検索結果のリレーションの表示画面を図3.17に示す。 検索結果のリレーションの表示は新規にウィンドウが生成され表示される。

**MEMBERS. 属性** (属性)

属性数 : 5

属性名	ENTRANCE	CODE	NAME
物理属性	日付	英数	日本語文字
サイズ	4		

**CLASSM. 属性** (属性)

属性数 : 3

属性名	CNO	CNAME	CODE
物理属性	英数	日本語文字	英数

**検索結果. データ** (ファミリー)

属性数 : 8      タプル数 : 11

ENTRANCE	CODE	NAME	AGE	ADDRESS	CNO
1980.10.28	2754	吉田	30	中野	96
1980.10.28	2754	吉田	30	中野	44
1983.12.10	2866	山下	46	世田谷	44
1984.08.16	4126	林	42	新宿	51
1981.02.28	5264	鈴木	23	中野	28
1981.02.28	5264	鈴木	23	中野	7
1980.03.22	6428	酒井	32	新宿	51
1982.10.02	8458	佐藤	24	世田谷	44
1982.10.02	8458	佐藤	24	世田谷	7
1982.04.12	8588	吉村	28	新宿	65
1982.04.12	8588	吉村	28	新宿	7

図3.17 MELQUERY/Rの検索結果表示画面

### 3.3 OA業務および意思決定業務への応用

#### 3.3.1 統合化利用者視野を提供する垂直分散型DSS-GOAL

##### 3.3.1.1 DSSの要件

DSSが経営者や管理者などを支援し、十分な効果を上げるために必要とされる要件には、

- 使い易い利用者インタフェース
  - システム導入の容易さ（異機種間相互接続性）
  - 機能の有用性と豊富さ
- が考えられる。

DSSの利用者は、一般的には経営者や管理者といった計算機操作には不慣れな人が多いので、その利用者インタフェースとして要求される要素としては第1に初心者でも熟練者でも操作し易いような対話形式のコマンド体系あるいはメニュー画面選択を提供すること、第2にDSSの操作途中にエラーが発生した場合や操作方法が途中で不明になったとき利用者をHELPする対話型支援機能を提供すること、第3に使用頻度の高い処理やコマンドなどは登録省略化そして履歴管理できる機能を提供することが考えられる。

更には、日本語の文章などをそのまま入力できるような自然言語インタフェースやエラー発生時にその復旧法のノウハウを知識ベースから獲得できるような知的HELP機能なども必要になっていくと考えられる。

DSSが企業などの利用者環境に導入される場合、利用する計算機システムや外部環境（例えば、外部データバンク）に対して柔軟に対応できるようにDSSは作成されていなければならない。従って、利用する周辺端末装置や基本ソフトウェア（例えば、ライブラリ）標準的かつ汎用性を有している必要がある。

DSSの利用者にとって有用な機能が豊富に提供されていることが望ましい。例えば、データベースからデータ抽出とデータ操作、業務モデルの作成とモデルシミュレーション、統計分析、時系列予測、多変量解析、ビジネスグラフ作成などの機能が必要とされている。

こうした機能は市場に流通している大多数のDSSに共通した機能であるが、DSSが標準的な機能だけでは不十分であるような場合は利用者自身が作成した新しい機能や画面を追加できるような環境もまた必要とされる。また、システム構築上の見地から見ると、1台の大型ホスト計算機にDSSのすべての機能と資源を集中させる方式とホスト計算機と端末（例えば、パソコン）に機能と資源を分散させる方式の2通りが考えられるが、こうした機能と資源の分散あるいは集中がシステム全体の整合性を乱すことなく実現されなければならない。更に、DSSに必要とされるデータベース管理システムの機能を考察すると、意思決定に使用されるデータは、時間属性を有していることが多い。すなわち、時系列データである。例えば、各年度毎の売上げ実績データや日々の外国為替レートなどがある。従って、DSSの求めるデータベース管理システムは、時系列属性を記述できるデータモデルが望まれる。例えば、各時刻毎のデータを格納する2次元リレーションが時間軸に沿って複数個存在するような3次元のリレーション空間を表現できるデータベー

ス管理システムアーキテクチャが必要である。

本研究では、上記の問題点に対する考察結果をもとに研究開発してきた「統合化利用者視野を提供する垂直分散型DSS-GOAL」について記述する[Wada85-1, Wada85-2]。

### 3.3.1.2 システム構成と機能

#### (1) 特長

DSS-GOALの主要な機能は、

- EDPとOAの融合化
- 垂直分散型
- 統合化利用者視野

の3点である。

#### EDPとOAの融合化

事務処理用計算機は、定型あるいは非定型のEDP業務に用いるオペレーショナルデータベースを保有していることが多い。例えば、社員の人事データや給与データ、部課の実務実績報告データなどがオペレーショナルデータベースに蓄積されることが多い。意思決定を実際に行なう管理者は、このオペレーショナルデータベースとのアクセスインターフェースが必要である。今回、研究開発したDSS-GOALは、既存のオペレーショナルデータベース（例えば、ネットワーク型データベースEDMS、VSAMファイル）を用いるEDP業務との親和性（すなわち、既存データベースへの問合せインタフェースの共通化）を高めている点が特長の一つである。更に、DSS-GOALのEDP業務の処理結果がビジネスグラフあるいは表の形で提供されるというOA指向も特長の一つである。すなわち、EDPとOAを融合化したDSSである。

#### 垂直分散型

意思決定する経営者や管理者は、その業務の性格上、機密情報を取り扱うことが多いと考えられる。従って、意思決定に関する情報は意思決定の過程では、意思決定者自身の情報として秘密にしておきたいことが多いと考えられる。この要求はDSSのシステム構築の観点から見ると、大型ホスト計算機とそのTSS端末から成る集中型システムよりもパソコンやワークステーション等のスタンドアロン型の装置が適していると考えられる。すなわち、意思決定者は、意思決定に関する情報を個人用ファイルとして蓄積保管し、かつタイムリーに参照することが好都合となる。しかしながら、意思決定をするために必要な基礎データは、やはり大規模なデータベースから抽出してくる必要がある。また、逆に意思決定がなされた後の情報は、企業内の共有データベースに蓄積し、企業内の関係者間で共通化する必要もある。

こうした相反する要求を内包する意思決定環境に適応するために、本研究にて研究開発したDSS-GOALは大型ホスト計算機と端末（例えば、ワークステーションやパソコン）とから成るシステムを考えて、ホスト計算機側に大規模データベースを配置し、かつ端末

側には個人用データベース（パーソナルテーブルと呼ぶ）を配置することにより、資源を分散化し、同じに大規模データベースとパーソナルテーブル間でのファイル転送を可能としている。

一方、DSSの処理という観点からは、DSS-GOALは端末側ではその得意とするビジネスグラフの作成、作表などの処理を提供し、大型ホスト計算機側ではモデルシミュレーション、統計分析、多変量解析、時系列予測のようなCPU負荷の大きな処理を実行することにより機能の分散化を行なっている。すなわち、機能と資源が大型ホスト計算機と端末間で垂直分散したDSSである。

### 統合化利用者視野

DSS-GOALは、その利用者である意思決定者に対して、パーソナルテーブルという利用者視野を提供している。パーソナルテーブルは2次元表形式のデータ構造を有しており、次の機能を提供している。

- 利用者が会話形式でそのテーブル枠の形式を自由に作成変更できる。
- パーソナルテーブルの項目は、その値、データ長、データ型、意味属性などを会話形式で自由に定義変更できる。
- 検索、射影、結合などの関係演算、項目間や行データ間の算術演算、併合、分離などのテーブル操作が会話形式で実行できる。
- パーソナルテーブルの内容は報告書やビジネスグラフにして出力することができる。また、モデルプログラムからも入出力できる。

DSS-GOALは、このパーソナルテーブルをすべての機能に対して使用するため、DSS-GOALで取り扱うデータはすべて統合化された利用者視野で把握できるという点が大きな特長である。

## (2) 構成

### ハードウェア構成

DSS-GOALは、当社の汎用計算機MELCOM DPシリーズ、COSMOシリーズ、メガミニシリーズ上で動作し、その端末は日本語ワークステーションM4378、パソコMulti 16、オフィスターミナルM5000（但し、Multi 16とM5000はエミュレータM4374Eを用いる。）を使用する。 図3.18にそのシステム構成図を示す。

### ソフトウェア構成

DSS-GOALのソフトウェアはホスト側と端末側に分散配置される。

ホスト側ソフトウェアは図3.19に示す如く、オペレーティングシステムVOS、UTS/VSのもとで動作する。 データベース操作は、ネットワーク型データベースEDMSやVSAMファイルをアクセスし、データ抽出することによりパーソナルテーブルが生成できる。 テーブル操作は関係演算処理、集合演算処理、初等統計演算処理のデータ加工とテーブルデータのファイル転送機能を提供している。 モデルベース操作は、モデルプログラムの作成編集、モデル式の登録参照、モデルプログラムのシミュレーション実行を提



供している。統計分析と時系列予測は、編集処理、多変量分析、傾向曲線当てはめ、季節調整法、投資効果分析を提供している。 端末側ソフトウェアは例えばパソコンのとき、図3.20に示す如くオペレーティングシステム日本語CP/M86のもとで、日本語ワークステーションエミュレータM4374E、ビジネスグラフ作成パッケージMulti BG/M、作表パッケージMultiplanが動作する。

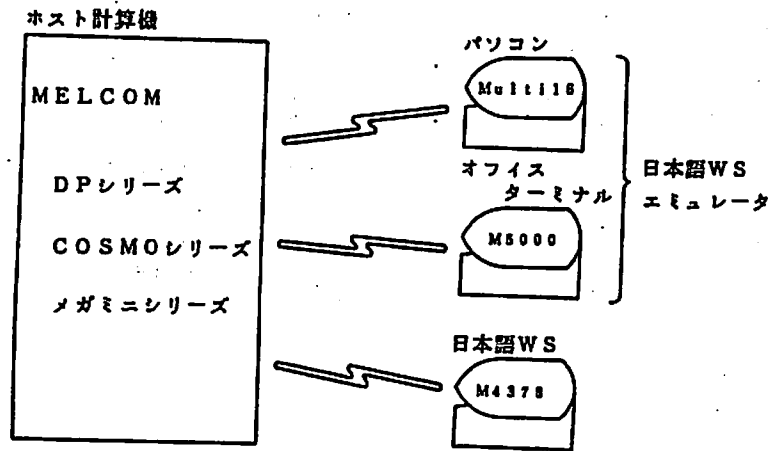


図3.18 DSS-GOALのシステム構成図

### (3) 機能

DSS-GOALの機能について、先述した3件の特長を中心にして記述していく。 DSS-GOALの基本利用者インタフェースはコマンド形式であり、一部の機能はメニュー選択のインタフェースが混在している。

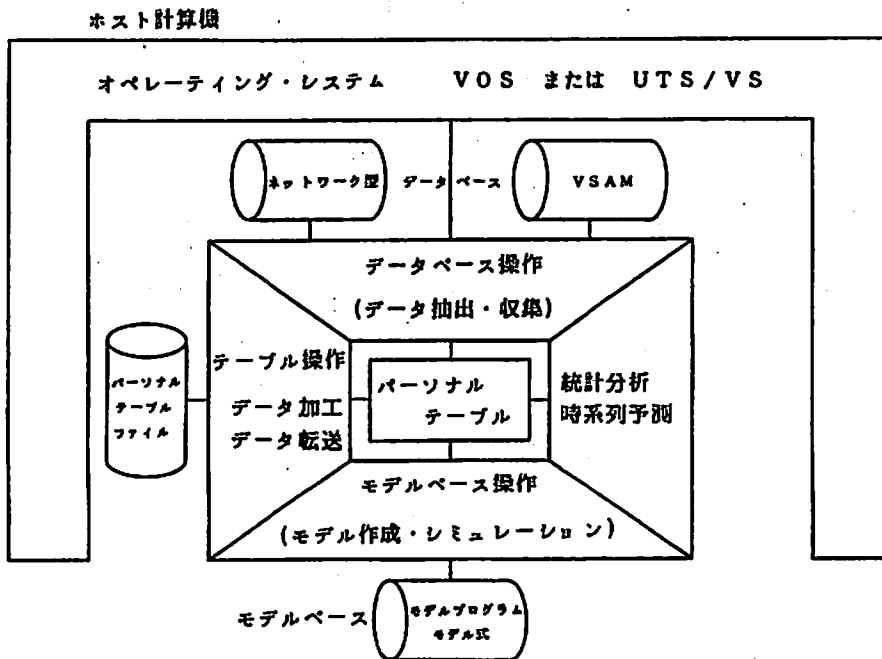


図3.19 ホスト側ソフトウェアの構成

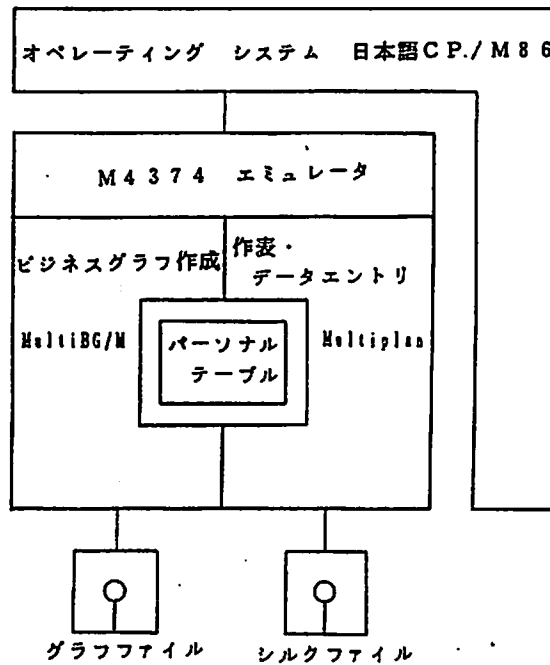


図3.20 端末側ソフトウェアの構成 (パソコンの場合)

### EDPとOAの融合化

EDP業務を遂行するために、DSS-GOALはネットワーク型データベースEDMSからデータ抽出するためのユーザコマンドCPEDMとVSAMファイルからデータ抽出するためのユーザコマンドCPVSPを提供している。CPEDMコマンドはEDMSデータベースのエンドユーザ向け問合せ言語DPと同一の利用者インタフェースを提供しているので、EDMSのオペレーショナルデータベースを使用しているEDP業務担当者は何ら新しい操作法を収得することなく、DSS-GOALからEDP業務が遂行できる利点を有している。CPVSPコマンドもVSAMファイルのエンドユーザ向け問合せ言語VIPと同一の利用者インタフェースを提供しているので、同様の利点を有している。

CPEDMコマンドは、アクセスするデータベース名（これは、サブスキーマ名と同一）、エリア名、グループ名、データ項目名をパラメータとして指定するが、アクセスの最小単位はデータ項目名である。データ項目名がデータベース内で一意でないときは、グループ名やエリア名で修飾する。CPEDMコマンドで検索した結果は、データ項目から構成される平面ファイルの構造をしたパーソナルテーブルに変換される。このとき、ネットワーク型データベースへのアクセスは、アクセス開始グループとアクセス経路を陽に指定することにより実行されるので、任意のネットワーク構造のデータベースがアクセスできる。

OA化をはかるために、DSS-GOALはデータベースからのデータ抽出結果であるパーソナルテーブルをデータ加工して、ビジネスグラフ化するためのユーザコマンドGRAPH、パーソナルテーブルを帳票の形で報告書化するためのユーザコマンドREPORを提供している。

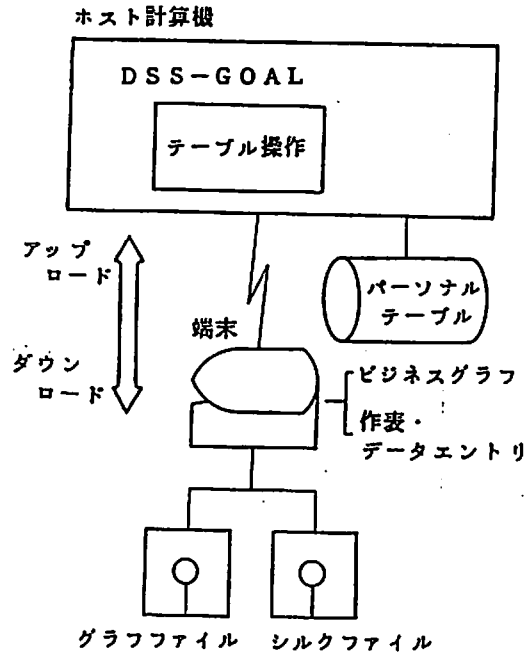


図3.21 垂直分散化

### 垂直分散型

処理の分散化のために、DSS-GOALは図3.21に示すように、ホスト計算機と端末の間でパーソナルテーブルをファイル転送するためのユーザコマンドを提供している。

ホスト側で作成編集したパーソナルテーブルはユーザコマンドTRNGRAを用いて、端末側のグラフファイルへとファイル転送して、端末側のグラフ作成パッケージMulti BG/Mでグラフ化する。

パーソナルテーブルへのデータエントリは、ホスト計算機側で実行する方法の他に、端末の作表パッケージ日本語Multiplanが利用できる。パソコン側で日本語Multiplanによりデータエントリしたパーソナルテーブルは、ユーザコマンドSLKTRNコマンドにより端末からホスト計算機へとアップロードする。

逆に、ホスト側で作成編集したパーソナルテーブルはユーザコマンドTRNSLKによりホスト計算機から端末へとダウンロードする。SLKTRNコマンドとTRNSLKコマンドは共にシルクファイルという共通フォーマットのファイルを用いてホスト計算機と端末の間でファイル転送する。従って、ホスト計算機のパーソナルテーブルは、日本語Multiplanにより端末のパーソナルテーブルとしてスタンドアロンモードでデータ加工できる利点がある。

### 統合化利用者視野

DSS-GOALは意思決定を支援するすべての機能に対して、統合化した利用者視野であるパーソナルテーブルをその入出力対象として意思決定者に提供している。すなわち、データベースからの抽出結果、データ操作の処理対象と処理結果、統計分析、多変量解析、時系列予測の被分析データと分析結果がパーソナルテーブルという利用者視野で提供

される。

パーソナルテーブルに対する基本的なデータ処理コマンドは、名称一覧表PTLIS、内容表示PTDIS、属性表示PTATR、削除PTDEL、新規作成PTCRT、編集PTEDT、併合UNION、分離PTSEP、内容選択SELEC、四則演算CALCU、射影PROJE、結合JOIN、分類SORT、合計PTSUM、統計TOSUM、割合RATIO、平均AVERAが提供されている。

パーソナルテーブルに対する応用的なデータ処理コマンドは、統計分析として相関分析CORRE、順位相関分析ORDCO、初等統計ELSTA、標準化行列STAND、多変量解析として判別分析DSCRI、主成分分析PRINC、因子分析FACTO、クラスター分析CLUST、時系列予測として傾向曲線当てはめFITLN、指数平滑法EXPSM、季節調整法EPAADが提供されている。

## その他

DSS-GOALは上記の機能に加えて、モデル作成とシミュレーション機能を提供している。モデル作成編集コマンドはモデルプログラムの作成MPCRT、編集MPEDT、内容表示MPDIS、削除MPDEL、モデルプログラム名称一覧表示MPLIS、モデル式の作成MECRT、編集MEEDT、削除MEDEL、モデル式名一覧表示MELISが提供されている。モデルシミュレーションコマンドは単純シミュレーションSIMUL、代替案分析WHTIF、目標探索GOALS、感度分析SENSIが提供されている。

### 3.3.1.3 使用例

DSS-GOALを用いたテーブル操作、モデル記述、統計分析の使用例を次に記述する。

#### (1) テーブル操作

パーソナルテーブル名の一覧表示は、PTLISコマンドの実行により図3.22に示すように、テーブル名、テーブルの用途と内容、テーブルの作成変更日時（年月日と時刻）が表示される。このとき、一覧表示が一画面分を超えたならば、ファンクションキーF3の押下により次画面が表示される。逆はF4の押下により前画面が表示される。

パーソナルテーブルの編集は、PTEDTコマンドの実行により図3.23に示すような編集機能のメニュー画面が当該パーソナルテーブルと共に表示される。編集機能は、項目または行の追加と除去、項目間または行間の入れ換え、項目単位の内容置換、項目の属性変更、特定項目値の変更、項目または行の複写、テーブルの用途／内容の書き換え、編集後のテーブル内容の解除と登録を提供している。テーブルの内容表示が一画面分を超えたときは、ファンクションキーF1、F2、F3、F4の押下により画面が各々、右、左、下、上の方向にスクロール表示する。

\*\*登録パーソナル・テーブル一覧

テーブル名	用途/内容	作成日時
IYUBAI	社員別販売実績	85/05/02 11:14
JINJI	人平表	85/04/26 11:25
SAPORO	札幌支店 (売上実績)	85/04/25 18:16
SENDAI	仙台支店 (売上実績)	84/12/20 12:00
TOKYOU	東京本店 (売上実績)	85/04/14 20:07

ファンクションキーを選んで下さい。

F3=次画面 F4=前画面 F8=操作戻り F10=終了

図3.22 パーソナルテーブル名の一覧表示

\*\*テーブル編集

	C 1 部	C 2 課	C 3 名前	C 4 性別	C 5 給料	
L 1	総務	庶務	岸島 亨	男	276000	神奈川県鎌倉
L 2	総務	庶務	田村 哲也	男	234000	東京都品川
L 3	総務	庶務	西田 由紀子	女	184000	神奈川県横浜
L 4	総務	庶務	松田 梨果	女	159000	神奈川県鎌倉
L 5	総務	人平	高野 克子	女	291000	東京都世田
L 6	総務	人平	山崎 利恵子	女	174000	東京都新宿
L 7	総務	人平	長島 幸子	女	149200	神奈川県横浜
L 8	総務	経理	岡村 武修	男	238100	神奈川県鎌倉
L 9	総務	経理	渡辺 明子	女	148000	東京都練馬
L 10	総務	経理	高井 明子	女	138900	神奈川県藤沢
L 11	営業	営業 1課	坂本 猛	男	315400	東京都港区
L 12	営業	営業 1課	山谷 進一	男	257000	神奈川県鎌倉
L 13	営業	営業 1課	田辺 茂子	女	194500	東京都調布
L 14	営業	営業 1課	池田 順子	女	167000	東京都杉並
L 15	営業	営業 2課	川崎 浩	男	304100	神奈川県鎌倉

テーブル編集 1. 追加 2. 除三 3. 入れ換え 4. 置換 5. 属性変更  
6. 値変更 7. 複写 8. 用途/内容 9. 解除 10. 登録  
選択項目を入力して下さい。

F1=右側 F2=左側 F3=次画面 F4=前画面 F5=指定画面 F8=操作戻り F10=終了

図3.23 パーソナルテーブルの編集

パーソナルテーブルの属性表示は、PTATRコマンドの実行により図3.24に示す如く、項目ごとにその項目名、物理属性、表示属性、表示属性、保護属性が表示される。物理属性はデータ型とデータ長を表現しており、例えば、C12は最大12バイトの文字列、I10は最大10桁の整数を意味する。表示属性は、テーブル上での表示形式を規定しており、無指定時はC型ならば左詰め表示、I型ならば右詰め表示を意味する。保護属性は項目の内容更新を許可するか（書換可）、あるいは禁止するか（書換不可）を規定する。

\*\*\*テーブル属性表示

	項目 1	項目 2	項目 3	項目 4	項目 5	項目 6	項目 7	項目 8	項目 9
項目部									
物理属性	C12	C12	C16	C6	I10	C50	14	C20	C10
表示属性									
保護属性	書換可	書換可	書換可	書換可	書換可	書換可	書換可	書換可	書換可

ファンクションキーを選んで下さい。

---

F1=右折 F2=左折 F8=操作支援 F10=終了

図3.24 パーソナルテーブルの属性表示

(2) モデル記述

MPEDTコマンドを用いて作成したモデルプログラムの内容はMPDISコマンドの実行により図3.25に示すように表示される。モデルプログラムは時系列周期定義、変数属性定義、モデル式、モデル式登録名から構成される。時系列周期とはシミュレーションするときのデータの時間間隔であり、例えば暦年、年度、半年、四半期などがある。変数属性とはデータ（変数）の物理属性とスケール因子であり、例えば%、千円、百万円、十億円のスケール表現として各々、P、T、M、Bが使用できる。変数名は日本語も使用できる。モデル式は変数、定数、演算子の組合せである。図3.26は多変量解析の主成分分析の結果から作成したモデル式の一例である。変数は時系列周期単位での時間遅れが定義できる。図3.25の例では、固定費<-1>が一周期分の時間遅れを表現している。

また、パーソナルテーブルからのデータ入力は、変数の代入文の右辺にそのテーブル名と項目名を記述する。図3.25の例では、

生産台数=SEISAN.C2

がパーソナルテーブルSEISANの第2項目からのデータ入力を表現している。定数は整数値、固定小数値、浮動小数値を記述できる。演算子は加減乗除とべき乗を記述できる。

● ● モデル・プログラム内容表示

モデル・プログラム内容	
売上高	生産台数 * 販売単価
変動費	売上高 * 変動費率
付加価値	売上高 - 変動費
固定費	固定費<-1> * ( 1 + 固定費伸び率 )
利益	付加価値 - 固定費
損益分岐点比率	( 固定費 / 付加価値 ) * 100
変動費率	0.6
生産台数	SEISAN.C2
販売単価	HANBAI.C2

ファンクションキーを選んで下さい。

F3=次画面 F4=前画面 F8=操作支援 F10=終了

図3.25 モデルプログラムの表示

● ● モデル式の作成

主成分1 = 国語 \* 0.4730 + 社会 \* 0.4734 + 数学 \* 0.4670 + 理科 \* 0.3846 + 英語 \* 0.0794 + 一般常識 \* 0.4242

式を更新・登録しますか？ (Y/N)

Y  
F8=操作支援 F10=終了

図3.26 モデル式

### (3) 統計分析

図3.27は学生ごとの国語、社会、数学、理科、英語、一般常識の試験の点数を表現したパーソナルテーブルである。このパーソナルテーブルに対する多変量解析の中の相関分析は、CORREコマンドにより図3.28に示すような各試験科目間の相関係数行列が得られる。図3.27のパーソナルテーブルに対する多変量解析の中の主成分分析は、PRINCコマンドにより図3.29に示すような主成分の重み式が得られる。図3.28や図3.29の分析結果は、パーソナルテーブルとして登録できるので、テーブル操作機能でアクセスできる。図3.30はある物質の成分と調質条件と強度の相互関係を定量的に表現したパーソナルテーブルである。このパーソナルテーブルの各項目データ間の傾向はFITLNコマンドにより分析できる。

● ● 相関分析 パーソナル・テーブル名=SEISEKI2

	C 1	C 2	C 3	C 4	C 5	C 6	C 7
	名前	国語	社会	数学	理科	英語	一般常識
L 1	堀田 政利	91	84	79	74	97	87
L 2	高瀬 茂	87	90	83	80	88	82
L 3	鈴木 恵津子	83	75	96	81	98	91
L 4	福永 康人	77	92	96	88	98	85
L 5	宮本 幸	91	77	80	78	70	94
L 6	接井 典子	81	83	77	80	67	100
L 7	岡沢 幸一	86	81	98	90	97	85
L 8	磯貝 千果子	79	88	92	83	95	88
L 9	川上 啓子	83	76	61	67	57	86
L10	佐藤 悦子	70	75	81	67	91	78
L11	市川 義明	87	89	90	85	100	91
L12	平石 和昭	79	29	33	55	79	84
L13	田中 典子	63	62	88	51	96	70
L14	白井 理恵子	52	56	50	60	85	77
L15	谷口 公一郎	81	91	73	63	95	88

計算対象項目を入力して下さい。

A

F1=右横 F2=左横 F3=次画面 F4=前画面 F5=指定画面 F8=操作支援 F10=終了

図3.27 多変量解析 (相関分析)



・相関係数行列

	C 1	C 2	C 3	C 4	C 5	C 6	
	相関係数行列	国語	社会	数学	理科	英語	—
L 1	国語	1.0000	0.5870	0.5955	0.5198	-0.0779	
L 2	社会	0.5870	1.0000	0.7964	0.4813	0.1881	
L 3	数学	0.5956	0.7964	1.0000	0.3131	0.3128	
L 4	理科	0.5198	0.4813	0.3131	1.0000	0.1546	
L 5	英語	-0.0779	0.1881	0.3128	0.1546	1.0000	
L 6	一般常識	0.6640	0.4305	0.5054	0.4865	-0.1997	

登録しますか？ (Y/N)

---

F1-右横 F2-左横 F3-次画面 F4-前画面 F5-指定画面 F8-操作支援 F10-終了

図3.28 多変量解析 (相関係数行列)

傾向曲線としては、多重回帰、ステップワイズ回帰、多項式回帰、指数曲数、べき乗曲線、成長曲線（ゴンベルツとロジスティック）が提供されている。例えば、多重回帰による傾向曲線当てはめは図3.31に示すような多重回帰式が得られ、モデル式として登録するとモデルプログラムから参照できる。

・主成分並み式

パーソナル・テーブル名=SEISEKI2

	C 1	C 2	C 3	C 4	C 5	C 6	C 7
	並み行列	国語	社会	数学	理科	英語	一般常識
L 1	主成分 1	0.4730	0.4734	0.4670	0.3846	0.0794	0.42
L 2	主成分 2	-0.2308	0.2002	0.2702	-0.0212	0.8214	-0.39
L 3	主成分 3	-0.0326	-0.2501	-0.4556	0.8195	0.2372	0.02

登録しますか？ (Y/N)

---

F1-右横 F2-左横 F3-次画面 F4-前画面 F5-指定画面 F7-メニュー F8-操作支援 F10-終了

図3.29 多変量解析 (主成分分析)

• • 傾向曲線当てはめ

	C 1	C 2	C 3	C 4	C 5
	成分 A	成分 B	調整条件 A	調整条件 B	強度
L 1	0.580	1.620	1.681	2.0828	84.1000
L 2	0.640	1.650	1.643	1.8261	73.4000
L 3	1.080	1.620	1.857	2.4969	91.2000
L 4	0.960	1.380	2.522	2.4914	124.0000
L 5	0.840	1.590	1.919	2.0212	114.0000
L 6	0.700	1.950	1.881	1.9191	91.4000
L 7	1.040	1.260	1.447	1.9868	108.1000
L 8	0.530	1.450	2.000	1.8129	102.9000
L 9	0.650	1.370	1.959	1.8921	83.6000
L10	0.750	1.750	1.881	2.3324	67.6000
L11	0.300	1.770	1.919	2.4942	80.8000
L12	0.980	2.000	1.919	1.8062	81.7000
L13	0.640	1.570	2.400	1.8976	114.6000
L14	0.700	1.460	2.079	1.9956	106.8000
L15	0.620	1.540	2.161	2.2810	97.7000

- 傾向曲線の選択 1. 多重回帰 2. ステップワイズ 3. 多項式 4. 指数  
5. ベキ 6. ゴンベルツ 7. ロジスティック

番号を選択して下さい。(1XX)

1

F1=右戻 F2=左戻 F3=次画面 F4=前画面 F5=指定画面 F10=終了

図3.30 傾向曲線当てはめ

• 多重回帰曲線

強度 = 51.65987+ 31.72321\*調整条件A+ 34.32206\*成分A -28.61770\*成分B

登録しますか? (Y/N)

F7=戻り F8=操作支援 F10=終了

図3.31 多重回帰式

### 3.3.2 属性指向データマイニングサーバKnodias

#### 3.3.2.1 研究の動機

##### (1) 理想のデータマイニングとは因果関係まで求めるもの

現在のデータマイニングでは、表面的に成立する規則性を機械的に抽出し、その事実が示唆する因果関係は、ユーザが推定して付すこととなる。いわば、規則性を結論とする物語を創造するのであり、この物語があってはじめて規則性に説得力が加わる。

理想上のデータマイニングとは、こうした因果関係の説明まで出力するものであると考える。一方、データベースに格納されているデータ単体では、この物語（あるいは因果関係）を生成するのに必要な情報を含んではいない。背景知識と呼ぶべき付帯情報を外部から与える必要がある。その最も原始的な形が、データとして格納されている項目の内容を属性値の列挙の形で表現したものである。これが、各領域で共有されている概念の構造を反映したものであるとき、これをオントロジ<sup>[Nishi95]</sup>と呼ぶことができると考える。

##### (2) 属性を用いて発見の品質を高めたい

相関ルール生成の代表的アルゴリズムApriori<sup>[Agrw94]</sup>では、データ中の項目名だけを見てマイニングするのが基本だが、結果を説明するのはむしろその属性値間の共通性であったり、そこに記された機能概念間に成立する依存関係の場合もあると考えられる。

また、マイニングの結果得られた知識から不要なものを捨てる際、自明、あるいは偶然の一致などの判断をするが、そこでは、物語性がない、もしくは予測可能なので陳腐、といった評価がなされていると考えることができる。こうした理由から、マイニング結果を説明するのに属性情報を利用すること、さらには属性情報を含めてマイニングすること、とを検討する必要があると考えた。

##### (3) 個人の水準の背景知識も統合したい

さて、一方で、オントロジの属性項目が共通なものについては、属性値を表形式で整理しうるので、データ本体が関係表で表現されているならそれとjoinすることで、属性を含んだ形のマイニングへの単純な拡張が可能である。しかしここで重要なのは、データ本体は一般には既存で共有されているものであり、オントロジはユーザが必要に応じて作成する点である。属性情報を利用したマイニングを行うためには、論理的空間的に遠隔での、共有データと局所的属性情報を統合利用する機構が必要である。

さて、こうした性質を有する属性指向のマイニングを実現する方法を、現在研究開発中のデータマイニングサーバKnodias<sup>[Ishi96]</sup>を前提に考察した。Knodiasはデータマイニングを目的とする一連のシステムの総称である。属性指向マイニングの議論をする便宜上、まずKnodiasの全体のコンセプトを次節で説明し、その後、本題の属性指向のマイニングについて議論することにする。

※ 本論文で用いる用語と概念の定義など

本論文で『属性』とは、データが指し示すオブジェクトを表現する属性であり、例えば『バター』というデータでは、製造者や、製品カテゴリ、などがあげられる。また、オントロジとデータ辞書の違いは、オントロジが各領域で共有される意味辞書であるのに対し、データ辞書は普通は意味管理まで含まず、またデータベースごとに定義されるものという点である。データ辞書の内容に加え、意味情報を補う形でオントロジを用いる。

今回オントロジを持ち出した背景には、CALISで提唱されているように、データを共有化する必要が社会的に認識されてきていることにある。データ共有のためには意味的な水準でも合意事項が必要で、その実現方法としてオントロジを用いることがある、という事情がある。分散された異種データベースを統合利用するニーズはデータマイニング以外にも共通に存在するのであり、そのソリューションのために、オントロジを共通に用いよう、というのが適用の動機である。

### 3.3.2.2 業務ノウハウ活用方式

#### (1) 相関ルール発掘の課題

相関ルールを効率よく求めるアルゴリズムとしてAprioriアルゴリズムがある[Agrw 94]。このAprioriアルゴリズムでデータの相関ルールを発掘すると、得られるルールは大量で、冗長なルールや意味のないルールを多数含んでいることが多い。さらに、この意味のないルールをふるい分けて得られるルールのうち有効なルールの大半は既知のルールであったりする。つまり、未知の有効なルールが、得られるルールの中にさらに埋もれている状況が発生するという問題がある。これは、ルールを発掘しようとするデータに対して、何も付加情報を与えないで、一律に処理を施すために起こると考えられる。

#### (2) データマイニングサーバKnodiasの概略

先に述べたようにデータマイニングの結果得られるルールは既知のものであることが多い。この既知であるルールというのは、人間がその業務において経験的に得た知識、専門家などにより確立されている知識、OLAPやデータマイニングの結果得られた知識などが考えられる。本研究ではこれらの知識を業務ノウハウとして蓄積し、これを参照してデータマイニングを行う方式を検討している[Ishi96]。

図3.32にその構成の概略を示す。図3.32において、業務ノウハウとは{シャツ、ネクタイ}や{雨、傘}のように既に獲得された相関のある項目を集合の形に表現したものである。また、オントロジとは、「靴 = シューズ」のような同義語や、「衣類 > シャツ」のような概念語を格納する辞書である。

#### (3) 業務ノウハウを用いたルール発見

次に、業務ノウハウとオントロジを用いたルール発見の手法について説明する。

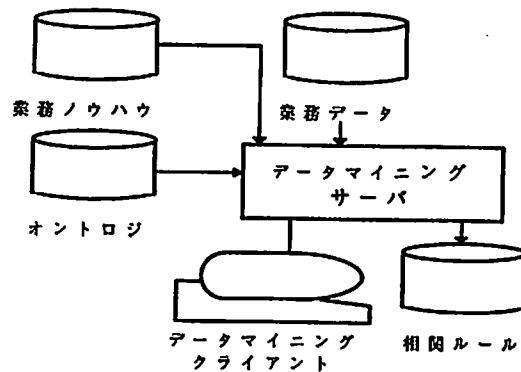


図3.32 Knodiasの構成の概略

### マスクとしての業務ノウハウの利用

データマイニングの結果得られるルールは既知のルールを多数含み、その結果、得られるルールが大量になり有効なルールが分かりづらくなってしまうことが多い。そこで、業務ノウハウが既知のルールであることを利用し、獲得したルールから既に業務ノウハウに蓄積されているルール以外のものをルールとして採用することにより、有効なルールを発見しやすくすることができる。

また、逆に業務ノウハウにあるルールが発見されない場合も考えられる。この場合、発見されないルールが専門家などにより確立された普遍のルールであれば、その既知のルールが成り立っていないという問題提起にも利用できると思われる。

### トレンド分析のための業務ノウハウの利用

一般に、流通業や小売業などにおいては社会的環境の変化などによるデータの傾向の変化がある。この傾向の変化（トレンド）を捕らえることは大変意義のあることである。

業務ノウハウに蓄積されたルールを過去のルールとしてとらえれば、それを時系列に扱いルールの変化からトレンドを分析することが可能となる。

### 候補属性抽出のための業務ノウハウの利用

相関ルールを生成する際、データベースのすべての属性について一律に相関を求めるのではなく、相関のありそうな属性に“あたり”をつけて相関を求めることにより有効なルールを発見することを考える。そこで、業務ノウハウに存在するルールの項目に着目し、その項目とルール発掘するデータのスキーマの属性とを比較する。ここで、業務ノウハウに一致する属性に着目してその属性のデータ間について相関分析することにより有効なルールの発見を目指す。

ここで、業務ノウハウとスキーマの属性を比較する際、必ずしも業務ノウハウの項目名とスキーマの属性が一致するとは限らない。そのため、オントロジを参照して、同義語

として定義されているものについては同一であるとみなすことにより、業務ノウハウを活用することが可能となる。

### 関連ルール抽出へのオントロジの利用

関連ルールを求める場合、個々のデータでは関連がないように見えるが、それらを包含する上位概念で関連を調べると有効な関連ルールが得られる場合が考えられる。そこで、関連ルールを発掘する際、データをオントロジで定義してある上位の概念に置き換えて、その置き換えたデータから関連ルールを発掘することにより有効なルールを発見することが期待される。

#### 3.3.2.3 属性情報利用方式

データマイニングにおいて、対象となるデータベースに格納される項目に関する属性情報を用いることで、

- 前処理、後処理の手段が豊富になる。
- 関連ルール生成を高速化できる可能性がある。

といったメリットがある。このアプローチを、ここでは、属性指向のマイニングと呼ぶことにする。

本節では、属性指向のマイニングの効果について検討した結果を議論する[Wadax96]。

#### (1) データマイニングサーバKnodiasの特長

##### データベースサーバに置かれる

近年、計算機資源は分散化の道をたどって来た一方で、企業の基幹データベースを始め、重要なデータは依然として集中的に管理される場合が多い状況といえる。データウェアハウスも基本的にはこの形態のデータベースを中心に置いており、この集中された大規模データを扱うデータマイニングシステムを、そうした共通のデータベースサーバ上に置くのは自然といえる。Knodiasはこうした中央集中的データマイニングを指向したサーバとして研究を進めているシステムである。図3.33に示すように、大福帳データベースと多次元データベースとを主な構成要素とするデータウェアハウス環境を土台として、その上にデータマイニング機能を構築する、というのが基本思想である。

##### プラグインのプラットフォーム

また、次々に提案される新しいマイニングアルゴリズムを迅速に取り込んでゆくために、基本部をアルゴリズム独立の形で作成し、各アルゴリズムをプラグインシステムの形で実装できる、いわばAIにおける「黒板」に近い、オープンな性質のシステムをめざしている。

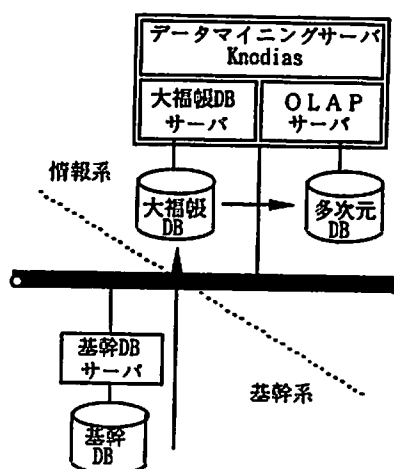


図3.33 Knodiasの基本構成

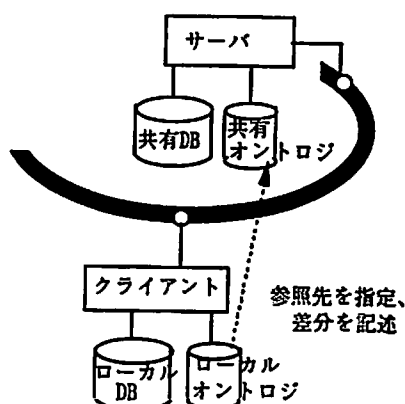


図3.34 分散するオントロジの統合利用

### イントラネット対応

一方、共有資源に対して多様な計算機環境からユーザがアクセスする利用方法を考えた場合、ユーザの使用環境の相違から利用方法に著しい制約を受けないようにするために、いわゆるイントラネット対応とし、もはや大多数のユーザが利用するウェブブラウザからアクセスする形式とし、クライアントをJava<sup>注25</sup>で記述することによってポータビリティを確保する、という方式を採用することは、むしろ自然といえる。

### オントロジを分散管理

データマートは、データウェアハウスのなかで必要な一部をユーザ環境に降ろしてきたものといえるが、個々のユーザがローカルに管理するデータがそれとは別に存在する場合

<sup>注25</sup> Sun Microsystems社の商標

があり、マイニングの際にはこのローカルデータ群と共通資源のデータとを統合して実行する場面もあると考えられる。本来整合性の保証されないデータベースを統合利用するには、意味的に等価でしかし表現の異なるデータを、同一のものとして明示的に対応づけ、同じ表現ながら全く異なるデータを、異なるものとして扱うようにやはり明示的に、調停（メディエート）を行う必要がある。

Knodiasでは、図3.34に示すように、共有資源のデータはもとより、各分散資源に対して、オントロジを付すことで、この調停を容易化することを狙う一方、本論文では以下に述べるように、属性指向のマイニングを実現することを目指している。

### システムスケーラビリティ

応用ドメインによって、データの量が異なるから、データウェアハウスがスケーラブルである必要があるのと同様に、データマイニングも各応用で必要十分な性能を有するように、小規模データ用パソコンソフトから、大規模のものではプラント時系列データ用分析システムまで、連続に拡張できるスケーラブルな性質が望まれる。

Knodiasでは、データと処理の規模に応じた多様なハードウェア構成に対応し、必要な性能を確保できる、システムとしてのスケーラビリティをめざしている。スタンドアロンのパソコン用小規模パッケージから、大規模なサーバシステムまで、機能的に一貫性を有するソリューションの体系を実現したいと考えている。

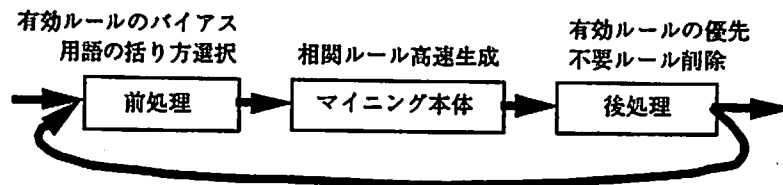


図3.35 マイニングの各段階での属性情報の利用

#### (2) 属性指向のマイニング方式とその効果

ここでは、データマイニングサーバKnodiasのコンセプトを前提に、属性指向のマイニングの実現方法と効果について述べる。属性情報は、ここではオントロジとして記述されていることを前提とし、特に、属性情報を用いた相関ルール生成に課題を絞って検討を試みる。

オントロジは、対象となる専門分野で共有される概念に関する辞書である。自分の扱おうとする領域の用語を整理して、データエントリでの整合性を維持するだけでなく、外部のデータベースとの概念体系の差異によるスキーマ等が合致しない状況において、対応関係を管理もしくは生成し、自分の領域に同化して扱うことを可能とする。用語間の関係、



すなわち同義語や、階層の上下関係など、用語辞書としても記述される内容をも包括する。実装の際、オントロジはフレームのような形式で記述されるのが通例である。マイニング処理本体だけでなく、その前処理と後処理の重要性はよく指摘されるが、ここでも説明の便宜上、図3.35に示すように、マイニング全体をこうした3段階に分けて、属性指向のマイニングの効果について検討を加える。

### 前処理における属性利用の効果

単一のドメインであっても、用語階層は必ずしも一つに定まるものではない。例えば、POS<sup>注26</sup>で記録される製品のデータは最も詳細のレベルのIDと考えられるが、製品名の階層として、その製造会社を上位と考えて括った方が都合がいい場合もあり、一方、製品分類でのより広い括りを上位と考えた方がいい場合もある。理想的には、あらゆる括りに対して相関ルールを生成してみて、有効と思われるものを選択していったほうがいいのだが、これは計算量が膨大となることが容易に推察できる。その妥協点として、前処理の段階でユーザが適切な括を指定し、必要に応じて詳細度を下げて相関ルールを生成する、という方法がとらうる。

今回の方式では、大福帳DBが作成される段階で、オントロジとデータ項目が比較され、出現回数が加算されて頻度情報としてオントロジに対応づけて格納される。新規のトランザクションではこの頻度情報が保守される一方、未知語に対してはその出現を記録し、後でオペレータが既存のオントロジに関係づけ、属性情報を記述することで、オントロジを成長可能のものとすることができる。

相関ルール生成の際には、最小頻度としてsupport値なるものを指定することがある。事前に頻度情報が集計されている場合、使用されている用語とその階層をインデントーション付のリストで表示し、個々の出現頻度値を付し、support値を超えているものについては色を変えるなどして、どの用語が相関ルールにかかるかを明示的に示すことができる。さらに、個々の用語でsupport値に満たない場合でも、その上の階層で括ればsupport値を超えることがあるので、システムではデフォルトとして、上位の方向に探して、そのしきい値を超える最初の用語を採用し、相関ルール生成に用いることができる。ユーザは適宜、所望の水準の用語を採用すべく、使用する用語としてより上位のものを選択することができる。

このようにして前処理で使用する用語を指定して相関ルールを生成し、その結果を見てまたこの水準に戻って指定を変えて実行、といった繰り返しでここで想定する利用方法である。

さて、特定のルールに対してバイアスをかける場合、最も簡単な方法は、このルールの右辺と左辺に登場する語に着目して相関ルールを生成する方法である。この時、正確に一致する語だけを選んだのでは、ただひとつの相関ルールの統計量を求めていることになり、それほど効果はない。そこで、オントロジを用い、同一もしくは類似の属性項目値を有する用語を関係語とみなし、その関係語で出現頻度がしきい値を超える語のみに対し、相関ルールを生成する。

---

<sup>注26</sup> Point of Sales

このように、前処理の段階で属性情報を用いることにより、相関ルール生成の方法を制御することが可能で、これによってより柔軟な知識発見が可能となる。

### 後処理における属性利用の効果

相関ルールの生成の際には、不要なルールが生成される一方、価値のあるルールが表示の優先順位の下の方になって、ユーザの目から埋もれる可能性がある。これを避けるために、相関ルール生成の後処理として、特定のルールについてはユーザの目に触れる前に削除し、逆に重要と判断される種類のルールでは、表示の優先順位を上げる処理を行う必要がある。

後者、すなわち特定ルールの優先順位を上げるには、生成された相関ルールに対し、バイアスとなるルールに含まれる語を有するルールを選択し、表示優先度を上げることで実現できる。バイアスとなるルールが外部から導入したものの場合、オントロジを参照して、語の置換を行ってから対応するルールを探す、という手順を踏むことで、ルールを同化して利用することができる。

一方、不要なルールが多量に生成されるのはデータマイニングでは大きな問題と認識されているが、ドメインの知識があって初めてルールが自明とか不要と判断できるのであり、フィルタを自動的にかける万能の方法が存在するわけではない。

そこで、今回検討した方式では、ユーザが削除指定したルールを記憶しておいて、次回以降表示しないという、簡単な方法を検討した。理想的には、属性情報を用いて類似ルールの削除を自動的に行いたい、用いる語の水準を変えてマイニングする利用方法を考えたとき、水準を上げる方向でも下げる方向でも自動的に削除すべきかが明らかではない。ユーザに毎回聞く、という方法もあるが、むしろ正確に一致するもののみ自動削除、という簡単な方法のほうが、マイニングの挙動が予測しやすい、見通しのよいシステムとなると考えられる。

### マイニング時の高速化の可能性

大規模データに対する相関ルール生成では、ディスクをスキャンする回数を減少することで、高速化できる可能性がある。例えばAprioriの例では、Agrawalが指摘する通り、アイテムセットの生成を2回分まとめて実行してメモリに保持できれば、サポートを数え上げる回数を減らせる。枝刈りをせずに処理を進めるために、候補増加によるメモリ爆発が問題となるが、この量を処理可能な範囲に維持できる問題であれば、ポトルネットワークとなりがちなディスクアクセスを数分の一という程度まで高速化は可能となる。

オントロジに頻度情報を付す今回検討した方法では、まず最初の数え上げ段階はない。使用する語の総量を頻度情報をもとに絞り込み、2項組を生成する。この時の候補の数は、使用する語の数をNとすれば $N(N-1)/2$ であり、この段階で3項組を生成すれば $N(N-1)(N-2)/6$ である。保持すべき候補の数はこの和であり、これがメモリ制約を満たすものか、Nから計算して検査すればよい。

今回は、あらかじめユーザが候補数上限数を与える方式を検討した。4項組以降も同様。ただし、3項組以降から候補を生成するとき、枝刈りがありうるので、事前に計算

できる候補数はその最大値である。

こうした方法で実際に高速化できるのは、特殊な場合と考えられる。すなわち、ユーザが与えるsupport制約を満たす項目数Nが少なく、データのスキャンがボトルネックとなるほどデータベースが大きいドメインを対象とする場合である。

#### 3.3.2.4 Knodiasの研究試作

本研究では、これまでに、効果的な相関ルール抽出処理のためのひとつのアプローチとして、業務ノウハウやオントロジを適用したマイニング手法を検討し、その構想について議論してきた[Ishi96, Wadax96]。

現在研究中のデータマイニングサーバKnodiasは、この業務ノウハウを背景知識として相関ルール抽出に活用し、さらにオントロジに記述された属性情報（用語の概念的階層）を用いることによる属性指向の柔軟なデータマイニングを目指している。

今回、このKnodiasの構想の一部を機能として実装する試作を行った[Yasu97]ので、本論文では、このKnodiasの特長と実現方式について議論する。

##### (1) コンセプト

Knodiasの基本的なコンセプトである「業務ノウハウの活用」「オントロジの利用」の2つについて説明する。

#### 業務ノウハウ（既得知識）の活用

業務ノウハウは、既に有効性が確認されているルールや、各分野の専門家の有する知識、以前に行ったデータマイニングの結果等を、ルールの形式で蓄積したものである。Knodiasは、これをデータマイニングに利用する。

一般にデータマイニングで機械的に規則性を掘り起こした場合、その結果得られるルールの数は膨大であり、かつその多くはユーザが求めているものではないと考えられる。即ち、偶然に成り立っている意味のない相関であったり、公然またはユーザにとって既知のルールが、その多くを占める結果になる。従ってユーザは、この大量に抽出された相関ルールの中から、そのユーザにとって価値あるルールを見つけ出す「宝探し」を行うことになるが、この過程において、ルール取捨の判断の基準となるのは、ユーザの持つ背景知識であると考えられる。

Knodiasは、主にこのユーザの背景知識を補うものとして業務ノウハウを活用する。さらにこの業務ノウハウは、前述のルールの取捨といった後処理だけでなく、相関をとる項目の組合せの絞込み等にも活用可能であると考えられる。

#### オントロジの利用

項目間の相関を求める場合、個々の項目そのもの同士の相関を求めるのではなく、ある括りによって上位概念を形成し、この上位概念間の相関に着目した方がよい場合があると

考えられる。例えば、POSデータとして記録されるデータは具体的な商品そのもののIDであるが、これをそのまま「〇〇製サラダ油500ml」という最も詳細なレベルで扱うよりも、「油」「サラダ油」といった、上位概念で括って扱った方がよい場合もある。

そこで、Knodiasは、マイニング対象データ中のデータ項目に関する属性情報を定義したオントロジを想定し、必要に応じてデータ項目を上位概念語へ置き換えて扱うことを可能にする。また、オントロジには、各データ項目の、データ中の出現頻度情報を持たせておくことにより、相関をとる項目の組み合わせの絞り込みといった処理に利用する。

## (2) 実現方式

### システム構成

図3.36は今回研究試作したKnodiasのシステム構成を表した図である。Knodiasは、イントラネット接続されたデータベースサーバ上に基本部（本体）を置き、ユーザはクライアントからWWW<sup>注27</sup>ブラウザを介してアクセスする方式をとった。これにより、ユーザが利用環境の違いによって制約を受けないようにしている。

図3.37は、サーバ上のデータマイニング基本部の構成の概略図である。

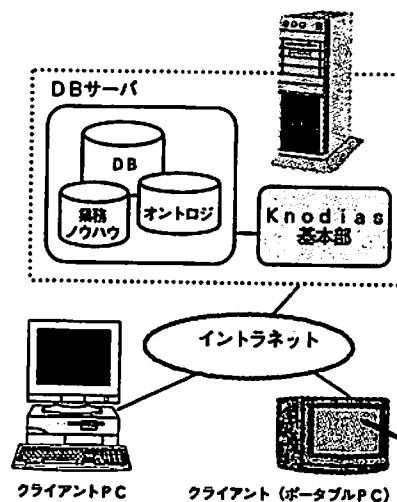


図3.36 Knodiasのシステム構成

この基本部は、マイニングエンジン制御部、前処理部、後処理部、DBアクセス機能群、GUI<sup>注28</sup>制御部からなる。実際にマイニングを行うエンジン（アルゴリズム）は独立

<sup>注27</sup> World Wide Web

<sup>注28</sup> Graphical User Interface

に作成し、マイニングエンジン制御部にプラグインのイメージで組み込むような形で用いられる。前処理部では、オントロジを用い、ユーザの指定により相関を求める階層水準の決定等を行う。後処理部は、業務ノウハウを用い、結果としてユーザに提示するルールの取捨を行う。

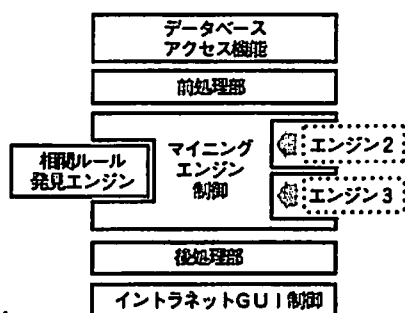


図3.37 Knodias基本部の構成

### (3) Knodiasによるデータマイニングの特長

今回研究試作したKnodiasで相関ルール抽出を行う場合の特長について述べる。

#### オントロジの利用による効果

ユーザは、まずマイニングしようとするデータに含まれるデータ項目について、それぞれ、オントロジに定義されたどの用語階層で扱うかを指定することができる。この時、オントロジに格納されている各データ項目の頻度情報が、マイニング時に用いる足切り基準として用いられるsupport値なる最小出現頻度を下回るものは、表示によってそれがわかるので、ユーザはこれを参考にして適切な用語階層を決定することが可能である。この前処理により、上位概念への置き換えを指定されたデータ項目は、マイニングエンジン制御部にてその上位概念に置き換えられて、エンジン部で扱われる。

#### 業務ノウハウの利用による効果

業務ノウハウに含まれる個々のルールには、そのルールの扱いに関する情報が付与されている。その情報とは、例えば、これは優先的に扱うべきルールであるとか、不要なルールである、といった類のものである。

Knodiasでは、この情報を用い、データマイニング後の後処理として、結果として表示するルールの整理を行うことができる。「優先」のルールは前の方に表示する、「不要」のルールは表示しないようにすることで、ユーザの「宝探し」を支援する。

また、データマイニングの結果得られたルールに対し、結果表示の画面上で行った操作（削除など）の履歴は、ユーザノウハウとして、業務ノウハウと同じ形式で自動的に登録され、以降、業務ノウハウと併せて使用することが可能である。

### 3.4 結言

分散計算環境におけるデータベースの利用技術に関する3件の研究課題に対する成果を総括すると、先ずクライアント/サーバ環境における位置透過性を実現した視覚的なデータベース・ユーザフレンドリインタフェース技術を研究し、エンドユーザコンピューティングを実現した。更に垂直分散化と操作の統一性を特長とするDSSを試作し、データベースの意思決定業務への適用性を向上させた。そして、データベース技術とAI技術と統合化した新たな意思決定支援技術として最近注目されているデータマイニング技術を研究し、従来のデータベースアクセス技術やDSSでは提供できないデータ間規則性の発見が可能となった。その規則性を用いることによりより高度な意思決定支援技術が実現された。

以下に、それぞれの研究課題に対する成果と今後の研究課題をまとめる。

先ず、UNIX系エンジニアリング・ワークステーション向けに研究試作したデータベース・ユーザフレンドリ・インタフェースMELQUERYについて記述した。MELQUERYはオブジェクト先行型視覚的利用者インタフェースである点が特長であり、リレーショナル・データベースに対するリレーショナル演算をエンドユーザにとって使い易いインタフェースで提供している。

次に、ホストコンピュータのリレーショナル・データベースに対して、クライアント/サーバ方式により、UNIX系エンジニアリング・ワークステーションから視覚的かつ対話形式の利用者インタフェースによって検索が行える遠隔データベースリクエストMELQUERY/Rについても記述した。

今後の課題としては、システム全体ではホストコンピュータ以外のデータベースサーバとの水平分散による分散データベースシステム、そしてインターネットやイントラネットへ対応するための他ソフトウェアとの連携（例、World Wide Web ブラウザとの連携やデータベース絞り込み検索などの機能の拡張）が挙げられる。また、データマイニングサーバなどへの融合を図った分散計算環境から移動体計算環境への拡大も考えられる。

次に、本論文では、EDPとOAの融合化をはかり、統合化利用者視野を提供することを特長とする処理の垂直分散型DSS-GOALについて記述した。DSS-GOALは利用者である意思決定者にパーソナルテーブルという統合化利用者視野を提供し、かつ従来のEDPやOA業務に対して同一の利用者インタフェースを提供することにより、その親和性と操作性を向上させた。更に、ホスト計算機と端末の間で共通ファイルの導入による処理と資源の垂直分散化を実現することにより、システム導入性、例えば異機種との接続性を向上させた。

今後の研究課題には、DSSの主要な機能であるモデルベース管理機能のシステムアーキテクチャ作り（例えば、モデルベース記述/操作、モデル辞書）、DSSの資源や処理の水平分散化技術などが考えられる。

続いて、大規模なデータベースからデータ間に潜在する有効な相関ルールを発見するために業務ノウハウとオントロジを利用したデータマイニングの手法を先ず提案した。

次に、このデータマイニング手法の中心的な技術である属性指向の相関ルール生成機能について述べた。そして、これをデータマイニングサーバKnodiasとして実装して、ルール発見の柔軟性と条件付き高速化が実現できることを示し、この方法論の有効性を議論した。

最後に、この研究試作したKnodiasの実現方式を整理した。

今後の研究課題は、Knodiasにおける相関ルール生成の更なる高速化、およびその生成された相関ルールの高品質化が先ず考えられる。また、同時にこのKnodiasの属性指向マイニング手法を多種多様な実業務のデータベースに適用した評価である。そして、DSSとデータマイニング機能を融合させた意思決定機能そのもの（言い換えれば、支援ではない機能）が未来の大きな研究課題である。

## 第4章 集中計算環境におけるデータベースシステムの性能評価

### 4.1 緒言

集中計算環境におけるデータベースシステムの性能評価に関する研究課題として、先ず「データベースの設計自動化支援・性能予測方式」、次に「データベース管理システムのバッファ管理方式」を取り上げて、研究開発した成果を以下に記載する。

**データベースの設計自動化支援・性能予測方式** データベースの自動設計技術は近年世界各国のデータベース研究者により盛んに研究されているテーマの一つである。

本章は、CODASYL 形データベースの設計と評価（特に、スキーマの設計とその性能評価）のために開発したツールDMSDBDAについて記述する。

DMSDBDAはMELCOM-COSMO700/900計算機上にて稼働されるEDMSデータベース管理システム（CODASYL仕様準拠）専用のツールである。DMSDBDAはデータベース管理者向けの対話形ツールであり、その機能は、

- 論理設計
- 物理設計
- 性能評価
- スキーマDDL<sup>注29</sup>生成

の4フェーズから成る。DMSDBDAはオンライン端末として接続されているCRT<sup>注30</sup>ディスプレイ装置を用いて実行され、更にメニュー検索機能を提供しているため、平易な利用者インタフェースが実現されている。データベースを新規に開発する際、DMSDBDAを使用することにより、データベースを実際に開発する前に、そのデータベースの性能や費用などを見積ることができる。

また、既に開発されているデータベースにDMSDBDAを適用するならば、そのデータベースの性能を改善するのに役立つ。

**データベース管理システムのバッファ管理方式** データベースとの実入出力回数を減らすための手段として、データベース用バッファの容量を増やすことや低いバッファ不在頻度が得られるバッファ置換アルゴリズムを利用することなどが考えられる。

データベースは現在、多種多様の構造を持つものが製品化され、あるいは研究中であるが、本研究は木構造のMUMPSデータベースを取り上げ、バッファ容量が限られているときいかなるバッファ置換アルゴリズムを採用したならば効率的なアクセスが実現されるかということ調査するために、データベース生成とアクセス要求およびバッファ置換を計

<sup>注29</sup> Data Definition Language

<sup>注30</sup> Cathode Ray Tube



算機上にてシミュレーションし、バッファ不在頻度を定量的に調べる。 図4.1にMUMPSシステムの概観を示す。

バッファ置換アルゴリズムは次の3方式を取り上げる。

- MUMPS開発委員会の提案を拡張したBPL<sup>注31</sup>法
- 更にそのBPL法を改善したBPL\*法
- 従来から広く普及しているLRU法

MUMPSデータベースを設計する際、本シミュレーションを行なうことにより、ある与えられたデータベースとバッファの容量に対して最も効率の良いバッファ置換アルゴリズムが選択できる。

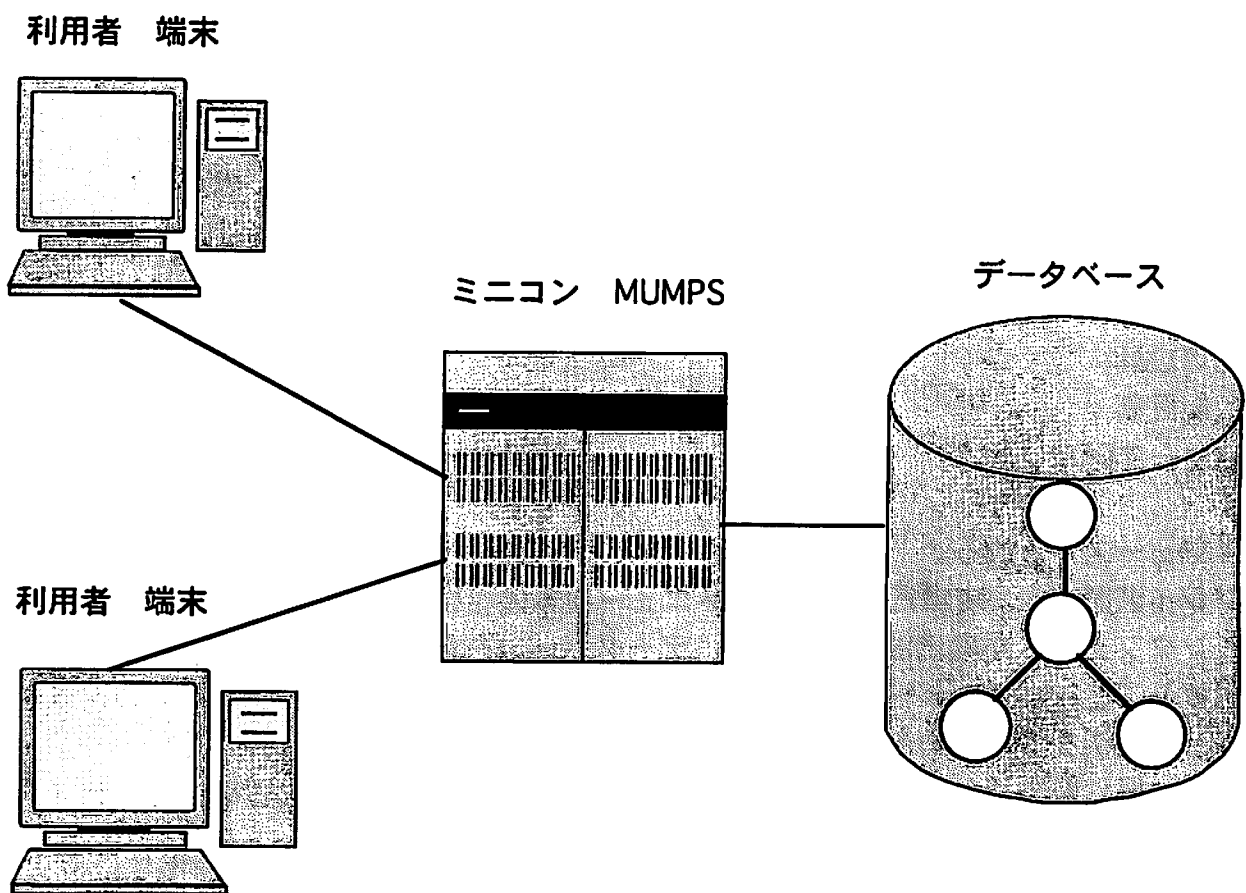


図4.1 MUMPSシステムの概観

## 4.2 データベースの設計自動化支援・性能予測方式

以下、4.2.1節ではDMSDBDAの機能、そして4.2.2節ではその使用例について述べていく。

<sup>注31</sup> Buffer Priority Level

#### 4.2.1 DMSDBDAの機能

DMSDBDAは図4.2に示されるごとく、

●データ仕様

●問合せ情報

●ハードウェア情報

●データベース管理システム情報

を入力し、次にその入力情報を基にして、

●論理設計

●物理設計

●性能評価

●スキーマ生成

の処理を行い、最後に

●データベース応用プログラムの実行時間

●データベース応用プログラムのアクセス費用 (i.e.主記憶の使用費用)

●データベースの記憶費用

を推定し、最終設計内容に合致したスキーマDDLファイルを生成する。

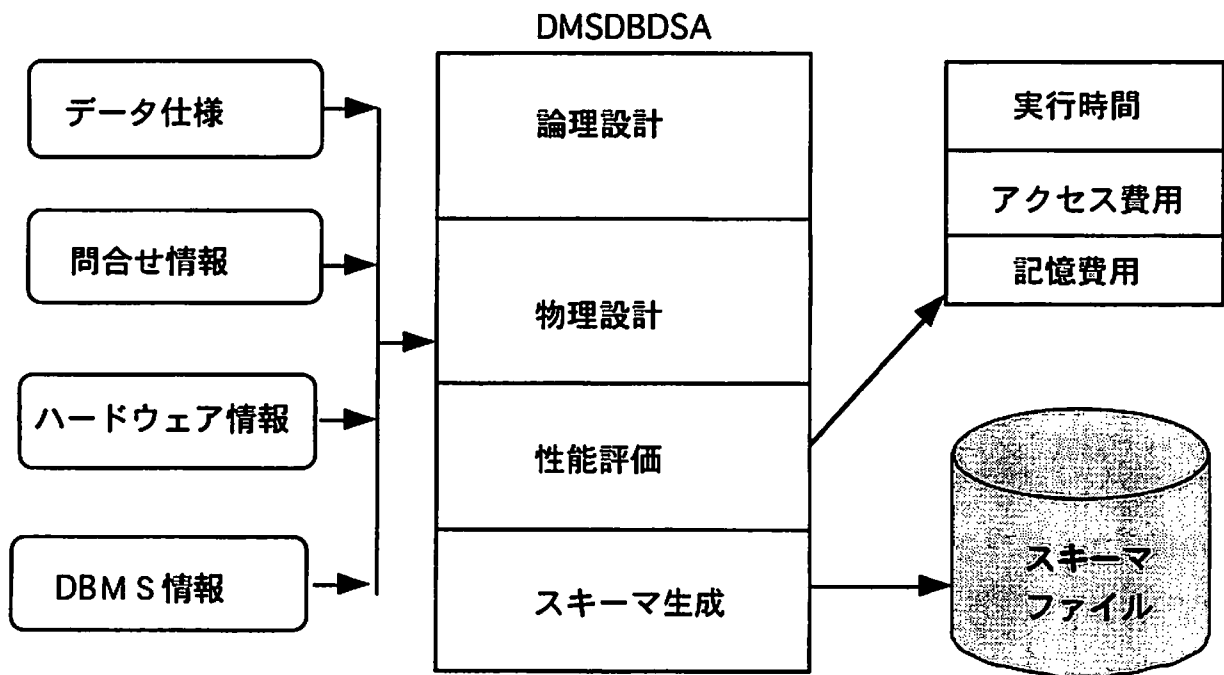


図4.2 機能の構成

#### 流れ

DMSDBDAは図4.3に示されるごとく、

●数件のデータ項目タイプを集合化（クラスタ化と呼ぶ）することから始まる。このデータ項目タイプの集合（クラスタと呼ぶ）は、データベース化する実体を表現する属性の

集合を意味し、データベース設計の基本単位として使用する。

●次に、DMSDBDAは論理設計→物理設計→性能評価と処理が進み、設計内容の性能が Gerritsenモデルの修正モデルを用いて推定される。

●データベース管理者は評価結果が満足できる水準に達したか否かを判定する。

・その評価結果が満足できないときは現在の物理構造を再設計する。つまり、物理設計と性能評価のフェーズが繰り返される。

・一方、その評価結果が満足できるときはスキーマDDLファイルの生成フェーズへと進み、DMSDBDAの実行は終了する。

●しかし、現在のデータベース論理構造を変更し、代替のデータベースも設計、評価したいときは、スキーマファイル生成フェーズへと進まずに、論理構造の再設計を行うことができる。

以降、DMSDBDAの論理設計、物理設計、性能評価、論理/物理再設計、そしてスキーマ生成を記述していく。

### (1) 論理設計

論理設計は、

- クラスタのグループ化
- グループ間のセット定義
- エリアの定義
- スキーマの定義
- データ形式とデータ長の定義

等の機能から構成されており、この定義はCRTディスプレイ装置と対話しながら行われる。

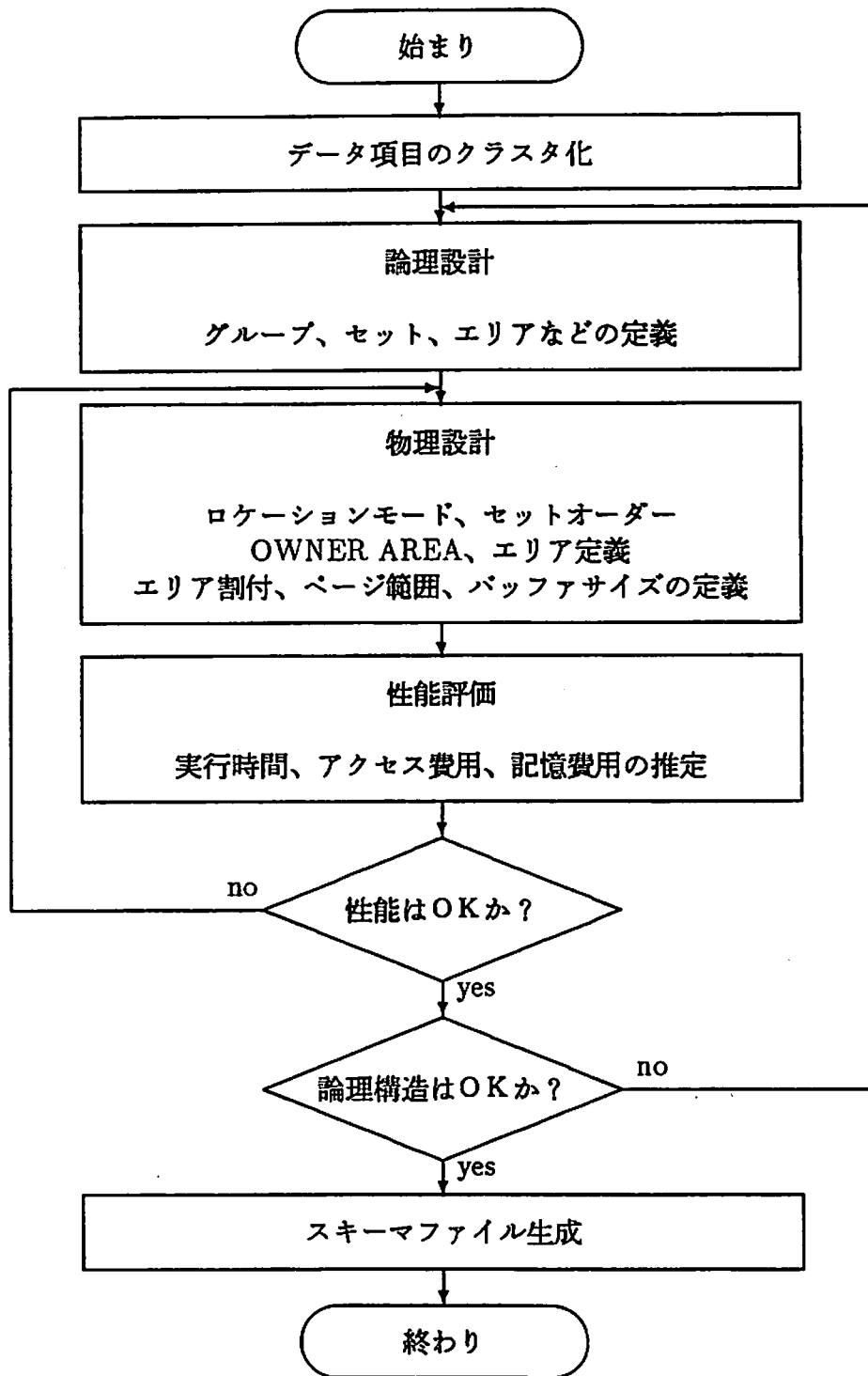


図4.3 機能の流れ図

## 手順

論理設計は図4.4に示される如く6ステップから成る。すなわち、

### ●データ項目のクラスタ化

クラスタは1個以上のデータ項目タイプから成る情報集合であり、論理設計の基本単位である。本ステップは、

- ・クラスタの新規作成
- ・既存クラスタの構成
- ・既存クラスタの表示

の3機能を有し、その機能の選択はCRTディスプレイ上に表示されたメニュー画面を通して実施される。

### ●クラスタのグループ化

グループは1個以上のクラスタから成る情報集合である。本ステップは、

- ・グループの新規作成
- ・既存グループの分割
- ・既存グループの統合

の3機能を有し、その選択はメニュー画面を通して実施される。

### ●セットの定義

セットは2つのグループを関係付けるための情報である。本ステップは

- ・セットの新規作成
- ・セットの追加
- ・既存セットの削除

の3機能を有し、その選択はメニュー画面を通して実施される。

- ・グループ間の対応関係が1対多の時は1つのセット、
- ・また対応関係が多対多の時は2つのセットと連結グループ

を指定する。本ステップが完了するとデータベースの論理構造がCRT画面に図形表示される。

●エリアの定義

本ステップは、  
・エリア名の定義  
・エリアサイズの初期化  
の2機能から成る。

●スキーマの定義

本ステップは、  
・スキーマ名の定義  
・パスワードの指定  
・サブスキーマ生成用プライバシーロックの指定  
・スキーマ変更用プライバシーロックの指定  
の4機能から成る。

●データ形式とデータ長の定義

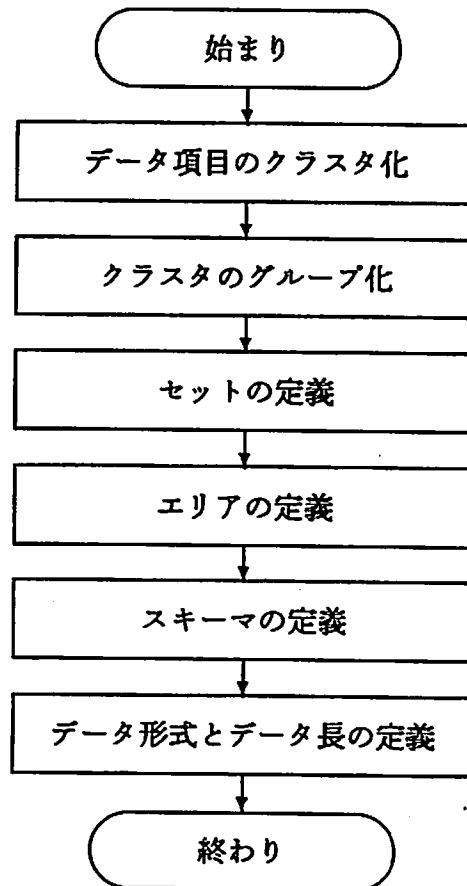


図4.4 論理設計の流れ図

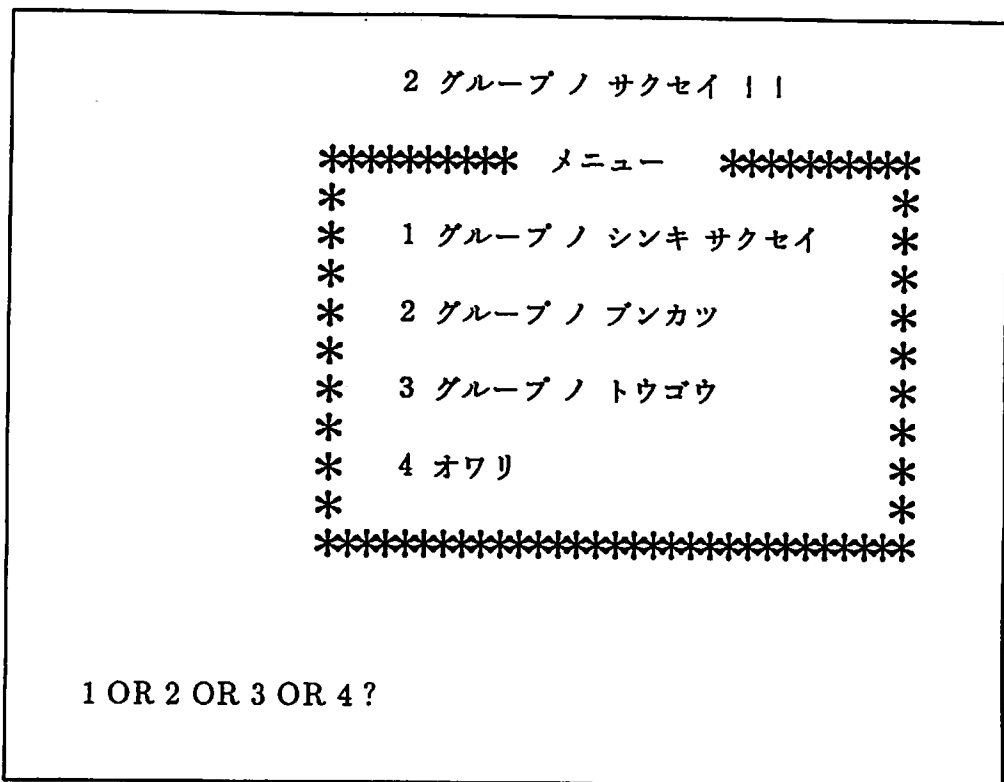


図4.5 グループ作成の画面

図4.5は、グループ化を行う際のCRT画面の一例であり、グループ化のために準備されている種々の処理機能がメニュー画面として表示されている。

本フェーズにおいて設計された論理構造は以下の判定条件（条件1から条件4）を満足しなければならない。従って、本論理設計フェーズはデータベースの構成要素（グループやセット等）の定義のほかに、設計内容の実現可能性を判定する機能をもつ。

### 判定条件

その判定条件を次に述べる。

[条件1] 一つのデータ項目タイプはただ一つのクラスタに含まれる。  
ここで、データ項目タイプの索引集合 $l$ とクラスタの索引集合 $C$ を各々、式(4.1)と式(4.2)のように定義する。但し、 $d$ はデータ項目タイプの総数、 $l$ はクラスタの総数とする。

$$l = \{1, 2, \dots, d\} \quad (4.1)$$

$$C = \{1, 2, \dots, l\} \quad (4.2)$$

そして、クラスタ拘束変数  $T_{t,c}$  ( $t \in I, c \in C$ ) を式(4.3)のように定義する。

$$T_{t,c} = \begin{cases} 1 & (\text{データ項目タイプ } t \text{ がクラスタ } c \text{ に含まれているとき}) \\ 0 & (\text{逆に含まれないとき}) \end{cases} \quad (4.3)$$

すると、条件1の必要十分条件は、式(4.4)のように記述される。

$$\sum_c T_{t,c} = 1, \quad t \in I \quad (4.4)$$

[条件2] 一つのクラスタはただ一つのグループタイプに含まれる。

ここで、グループタイプの索引集合  $R$  を式(4.5)のように定義する。但し、 $n$  はグループタイプの総数とする。

$$R = \{1, 2, \dots, n\} \quad (4.5)$$

そして、グループ拘束変数  $G_{c,i}$  ( $c \in C, i \in R$ ) を式(4.6)のように定義する。

$$G_{c,i} = \begin{cases} 1 & (\text{クラスタ } c \text{ がグループ } i \text{ に含まれているとき}) \\ 0 & (\text{逆に含まれないとき}) \end{cases} \quad (4.6)$$

すると、条件2の必要十分条件は、式(4.7)のように記述される。

$$\sum_i G_{c,i} = 1, \quad c \in C \quad (4.7)$$

[条件3] 一つのグループタイプはただ一つのエリアに属す。

ここで、エリアの索引集合  $A$  を式(4.8)のように定義する。

$$A = \{1, 2, \dots, n\} \quad (4.8)$$

そして、エリア拘束変数  $W_{i,a}$  ( $i \in R, a \in A$ ) を式(4.9)のように定義する。

$$W_{i,a} = \begin{cases} 1 & (\text{グループ } i \text{ がエリア } a \text{ に属すとき}) \\ 0 & (\text{逆に属さないとき}) \end{cases} \quad (4.9)$$



すると、条件3の必要十分条件は、式(4.10)のように記述される。

$$\sum_a W_{i,a} = 1, i \in R \quad (4.10)$$

[条件4] 再帰セットの定義は許さない。

上記の4件の条件の中で、条件3はGerritsenモデルにも規定されているが、残りの3件の条件は本研究にて新たに導入したものである。

### 操作上の利点

本ツールの論理設計は次のような操作上の利点が生まれた。

[操作1] クラスタのグループ化のとき、DMSDBDAはクラスタを2件ずつ画面に表示し、その2件のクラスタを一つのグループにまとめる(yes)か否(no)かをデータベース管理者に照会する。

[利点1] 従って、データベース管理者はyesかnoのいずれかを指定するだけでよいので、クラスタの内容をすべて覚えていなくともよい。

[操作2] グループ間のセット定義のとき、DMSDBDAはグループ名の一覧表を画面に表示するので、データベース管理者はこの一覧表の中からオーナグループとメンバグループを選出し、セット名を定義する。

[利点2] 従って、データベース管理者はグループ名をすべて覚えていなくともよい。

[操作3] データ項目のデータ形式の定義のとき、DMSDBDAはデータ項目名を1件ずつ画面に表示し、データベース管理者にメニュー画面から適当なデータ形式を選択するように要求する。

[利点3] 従って、データベース管理者はデータ項目名のすべてを覚えている必要はなく、加えてEDMSの提供しているデータ形式の種類も知らなくてよい。

[操作4] グループのエリア定義のとき、DMSDBDAはグループ名の一覧表を画面に表示するので、データベース管理者はこの一覧表の中からグループを選出し、エリア名を定義する。

[利点4] 従って、データベース管理者はエリア名をすべて覚えていなくともよい。

この4件の利点の内、利点1、利点2、利点3が上記の条件1、条件2、条件4を新規に導入した結果であり、一方、利点4が従来条件3により既に存在しているものである。

## (2) 物理設計

物理設計は、図4.6に示す如く、7ステップから成る。すなわち、

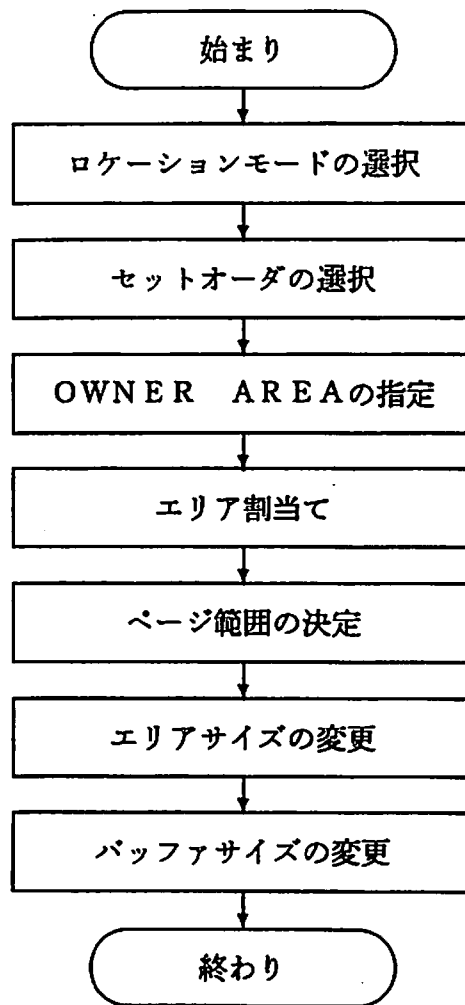


図4.6 物理設計の流れ図

- グループのロケーションモードの選択
- セットオーダの選択
- グループのOWNER AREA<sup>注32</sup>の指定
- エリア容量の指定
- グループのエリア割付
- グループの頁範囲の指定
- バッファ容量の決定

などから構成されており、この指定もCRTディスプレイ装置と対話しながら行われる。

ロケーションモードは、

- CALC
- VIAセット

<sup>注32</sup> CODASYL仕様のSYSTEMに相当

●INDEXED

のいずれかが選択できる。 CALCとVIAはGerritsenモデルでも定式化されているが、EDMSはINDEXEDロケーションモードも提供しているので、本研究ではこのINDEXEDも新たにモデル化した。

セットオーダは、

●NEXT

●NEXTとOWNER

●NEXTとPRIOR

●NEXTとOWNERとPRIOR

のいずれかが選択できる。 Gerritsenモデルはこの他にポインタ配列もモデル化しているが、EDMSはポインタ配列を提供していないので本研究のモデルから除外した。EDMSはグループのページ範囲の指定機能を提供しているので、本研究はこの指定機能を新たにモデル化した。 そして、残りの物理設計項目の内容はGerritsenモデルと同じである。

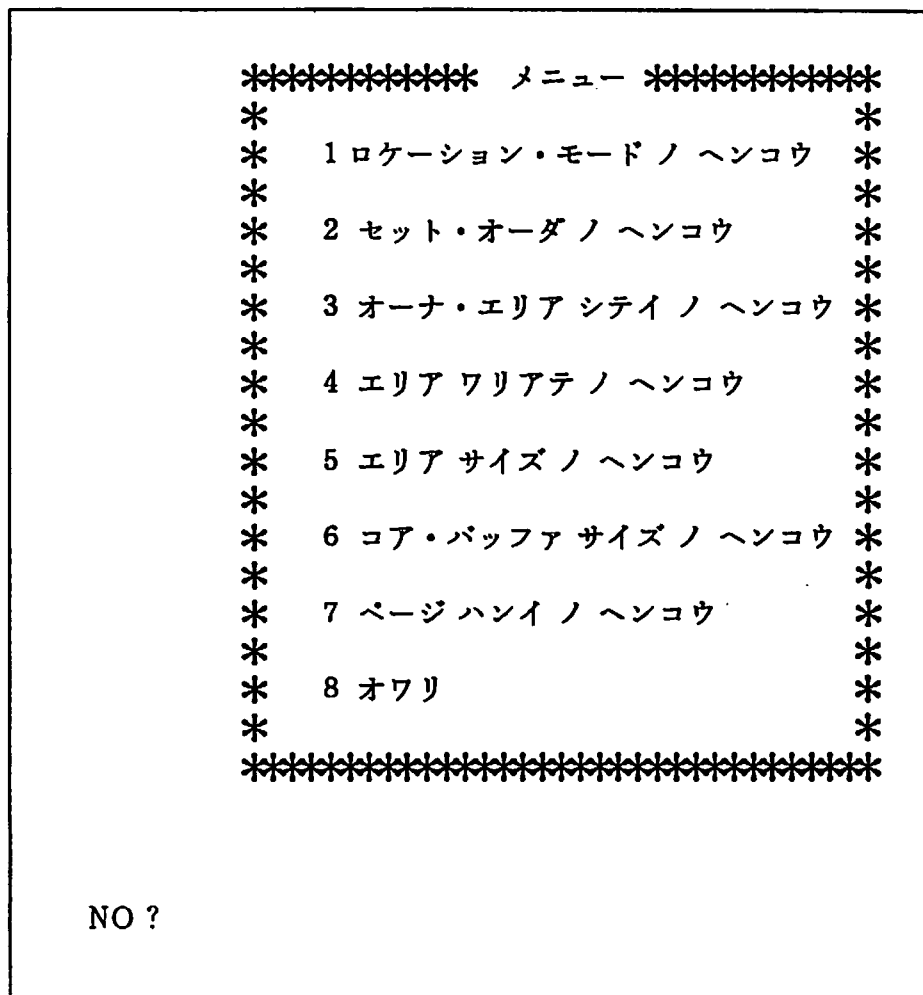


図4.7 物理設計の画面

図4.7は物理設計を行う際のCRT画面の一例であり、上記の設計項目がメニュー画面として表示されている。

本フェーズは先述した設計項目の初期化処理がある。

### 初期物理構造

論理構造が決定されると、初期物理構造が次のように決定される。

- CALCロケーションモード
- NEXTセットオーダ
- OWNER AREA指定なし
- データベース管理者指定のエリア容量
- データベース管理者指定のエリア割付
- データベース管理者指定のページ範囲
- 1ページのバッファ

この初期物理構造は次節の性能評価の結果を基にして修正することができる。更に、本フェーズはその物理構造の実現可能性を判定する機能をもつ。

### 判定条件

その判定条件を次に述べる。

[条件5] 一つのグループタイプはただ一つのロケーションモードをもつ。

ここで、ロケーションモードの索引集合  $L$  を式(4.11)のように定義する。但し、 $m$  はセットタイプの総数とする。

$$L = \{1, 2, \dots, m, m+1, m+2\} \quad (4.11)$$

また、 $j \in L$  のとき、

$$L = \begin{cases} \text{VIA} & (1 \leq j \leq m \text{ のとき}) \\ \text{CALC} & (j = m+1 \text{ のとき}) \\ \text{INDEXED} & (j = m+2 \text{ のとき}) \end{cases} \quad (4.12)$$

となる。そして、ロケーションモード拘束変数  $X_{i,j}$  ( $i \in R, j \in L$ ) を式(4.13)のように定義する。

$$X_{i,j} = \begin{cases} 1 & (\text{グループ } i \text{ がロケーションモード } j \text{ をもつとき}) \\ 0 & (\text{逆にもたないとき}) \end{cases} \quad (4.13)$$

すると、条件5の必要十分条件は、式(4.14)のように記述される。

$$\sum_j x_{i,j} = 1, \quad i \in R \quad (4.14)$$

[条件6] VIAセットロケーションモードは、そのセットのメンバのみに指定する。

[条件7] 一つのセットタイプはただ1種類のセットオーダをもつ。

ここで、セットタイプの索引集合  $S$  とセットオーダーの索引集合  $U$  を各々、式(4.15)と式(4.16)のように定義する。

$$S = \{1, 2, \dots, m\} \quad (4.15)$$

$$U = \{1, 2, 3, 4\} \quad (4.16)$$

また、 $u \in U$  のとき、

$$U = \begin{cases} \text{NEXT} & (u = 1 \text{ のとき}) \\ \text{NEXT} + \text{OWNER} & (u = 2 \text{ のとき}) \\ \text{NEXT} + \text{PRIOR} & (u = 3 \text{ のとき}) \\ \text{NEXT} + \text{OWNER} + \text{PRIOR} & (u = 4 \text{ のとき}) \end{cases} \quad (4.17)$$

となる。そして、セット拘束変数  $Y_{j,k}$  ( $j \in S, k \in U$ ) を式(4.18)のように定義する。

$$Y_{j,k} = \begin{cases} 1 & (\text{セット } j \text{ がセットオーダー } k \text{ をもつとき}) \\ 0 & (\text{逆にもたないとき}) \end{cases} \quad (4.18)$$

すると、条件7の必要十分条件は、式(4.19)のように記述される。

$$\sum_k Y_{j,k} = 1, \quad j \in S \quad (4.19)$$

[条件8] エリアの容量はそのエリアに属す全グループのサイズの合計よりも大きい。

[条件9] 2次記憶装置の容量は全エリアの容量の合計よりも大きい。

[条件10] 主記憶容量はバッファ容量とデータベース管理システム主記憶占有容量の和よりも大きい。

[条件11] VIAセットロケーションモードの場合、そのセットのオーナーとメンバは同じエリアに属す。

[条件12] INDEXEDロケーションモードをもつグループタイプは1エリア内に多くとも一つだけ存在する。

[条件13] グループのページ範囲はそのグループのオカレンスのサイズの合計よりも大きい。

[条件14] OWNER AREAの指定は、オーナーとしてのみ定義されているグループ (i.e.メンバーとしては定義されていないグループ) のみに許される。

上記の10件の条件の中で、条件5と条件7はGerritsenモデルの拡張であり、条件6および条件8から条件11はGerritsenモデルに既に規定されているが、条件12から条件14は本研究にて筆者が新たに導入したものである。

### 操作上の利点

本ツールの物理設計は次のような操作上の利点を有している。

[操作1] グループのロケーションモードの選択のとき、DMSDBDAはグループ名とそのグループに現時点において指定されているロケーションモードの内容を表示する。

又、その内容を変更したときはメニュー画面から適当なロケーションモードを選択するようにデータベース管理者に要求する。

[利点1] 従って、データベース管理者は各グループに指定されているロケーションモードの現在内容を覚えていなくともよく、且つEDMSが提供しているロケーションモードの種類も知らなくともよい。

[操作2] セットオーダの選択のとき、DMSDBDAはセット名とそのセットに現時点において指定されているセットオーダの内容を表示する。 又、その内容を変更したいときはメニュー画面から適当なセットオーダを選択するようにデータベース管理者に要求する。

[利点2] 従って、データベース管理者は各セットに指定されているセットオーダの現在内容を覚えていなくともよく、且つ、EDMSが提供しているセットオーダの種類も知らなくともよい。

[利点3] 他の物理設計項目についても上記と同様な利点がある。

この3件の利点の内、利点1と利点2はそれぞれ従来から拡張した条件5と条件7により一層強化されたものであり、一方利点3が上記の条件12から条件14を新規に導入した結果である。

### (3) 性能評価

性能評価は、

- 応用プログラムの実行時間の推定
- 応用プログラムのアクセス費用の推定
- データベースの記憶費用の推定

などから成る。 DMSDBDAはデータベースの現在の設計内容を評価する。 その評価は図4.8に示す如く、5ステップから成る。

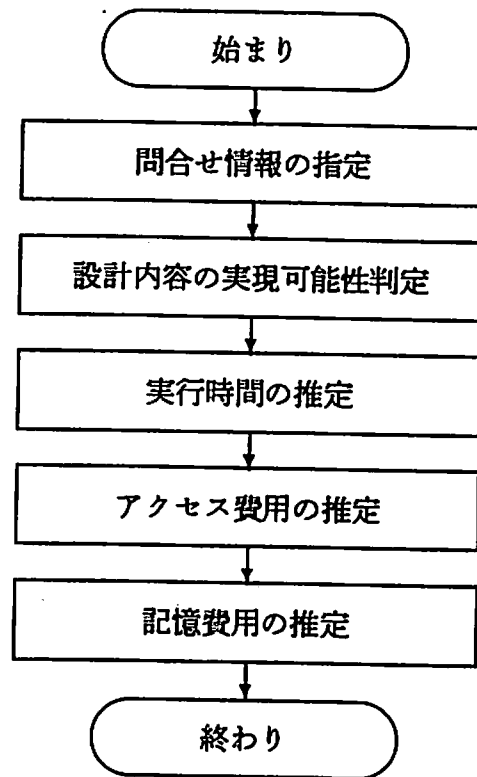


図4.8 性能評価の流れ図

DMSDBDAは性能評価を行うために、次の情報を使用する。

- 問合せ情報
  - ・アクセスパス
  - ・アクセス回数
  - ・希望実行時間
  - ・検索の論理条件
- ハードウェア情報
  - ・ページ長
  - ・主記憶容量
  - ・2次記憶容量
  - ・主記憶格納費用
  - ・2次記憶格納費用
  - ・2次記憶の平均アクセス時間
- データベース管理システム情報
  - ・ポインタ領域長
  - ・データベース管理システム占有記憶容量

・平均シーク回数 (CALCアクセス)

先ず、DMSDBDAは、データベースの性能を評価する前に、現在の設計内容が、実現可能であるか否かを判定する。前節で述べた条件5から条件14が、その判定の対象である。しかしながら、実現不可能であると判定された時は、データベース管理者は次節のようにデータベースの構造を再設計しなければならない。

次に、DMSDBDAは、上述の情報を用いて、与えられた問合せの実行時間を推定し、そして主記憶装置のアクセス費用とデータベース記憶費用も推定する。図4.9は現在のデータベース設計内容の性能評価の結果を表示したCRT画面の一例である。

性能はGerritsenモデルをEDMS向けに修正した数式モデル (なお、付録を参照されたい) から推定される。

### 応用プログラムの実行時間

応用プログラムの実行時間  $T$  は、

[実行時間1] セットオカレンスの論理的なアクセスの回数

[実行時間2] グループオカレンスの物理的サイズ  $Z$

[実行時間3] データベースのページの物理的なアクセスに要する平均時間  $PAT$  を基にして推定する。

データベースノヒョウカケツカ		
I レスポンス タイム I	I	900(フン) I
I-----I	I-----I	I-----I
I アクセス コスト I	I	167423(エン) I
I-----I	I-----I	I-----I
I ストレージ コスト I	I	10000(エン) I
I-----I	I-----I	I-----I
I トータル コスト I	I	177423(エン) I
I-----I	I-----I	I-----I

ゲンザイ ノ データベース ノ コウゾウ ヲ セーブ シマスカ ? (Y/N)

図4.9 性能評価の画面



## 論理アクセス回数

論理的なアクセスの概念は、Gerritsenモデルに定式化されているものを本研究はそのまま利用する。すなわち、

- オーナオカレンスの論理的キー付きアクセス
  - その論理的キーなしアクセス
  - メンバオカレンスの論理的セットアクセス
- の3種類がモデル化されている。

## 物理サイズ

Gerritsenモデルでは、 $Z$ は

- 当該グループに属す全データ項目のデータ長の合計
  - オーナポイント長
  - メンバポイント長
  - CALCポイント長
  - 特異セットポイント長<sup>注33</sup>
- を加算した値であるが、本研究はこの値に
- INDEXEDキー項目長
- を加算した値を $Z$ としている。

## 物理アクセス平均時間

$PAT$ は、ある一つのセットオカレンス内の全メンバオカレンスを検索するのに要する物理アクセスの平均時間を意味する。 $PAT$ は当該メンバグループのロケーションモードに依存するので、Gerritsenモデルは

- CALC
  - VIAセット
- の2ケースに分けて算出している。本研究はこの2種のロケーションモードの他に、
- INDEXEDロケーションモード
- も考慮にいれて $PAT$ を算出している。

従って、セット $j \in S$ における $PAT$  ( $PAT_j$ ) は、付録の式(4.2.28) のようになる。

$T$ は以上の情報を基にして算出するが、 $T$ はアクセスの形態に存在するので、Gerritsenモデルは、

- セット経由のメンバアクセス
- セット経由のオーナアクセス
- キーなしアクセス
- キー付きアクセス

の4ケースに分けて算出している。本研究は、このGerritsenモデルの算出方法に、

---

<sup>注33</sup> EDMSのOWNER AREAポイント長に相当

- INDEXEDロケーションモード
- セットオカレンスのPRIORポインタ<sup>注34</sup>

の機能を考慮した数式を補充することにより $T$ を算出している。

従って、問合せ $q \in Q$ の実行時間 $T_q$ は、付録の式(A.7)となり、この結果 $T$ は付録の式(A.8)のようになる。但し、 $Q$ は問合せ索引集合であり、式(4.20)のように定義する。また、 $v$ は応用プログラムの件数である。

$$Q = \{1, 2, \dots, v\} \quad (4.20)$$

### 応用プログラムのアクセス費用

応用プログラムのアクセス費用 $C_A$ は、応用プログラムの主記憶使用費用を意味する。

Gerritsenモデルは、 $C_A$ は主記憶の使用容量と使用時間（これは応用プログラムの実行時間 $T$ に相当する。）の積に、主記憶の使用単価を乗算することにより式(4.21)のごとく算出しており、

$$C_A = T(bB + Biase)C_1 \quad (4.21)$$

本研究もこの算出方法を利用している。但し、 $b$ はバッファページ数（前節で述べたごとく、 $b$ の初期値は1ページである。）、 $B$ はデータベースのページ長(Byte)、 $Biase$ はデータベース管理システムの占有記憶容量(Byte)、 $C_1$ は主記憶アクセスの単価（円）である。

### データベースの記憶費用

データベースの記憶費用 $C_S$ は、物理設計時に定義したエリア容量を2次記憶装置上に確保し、且つ維持するのに要する費用を意味する。

Gerritsenモデルでは、 $C_S$ は定義したエリア容量の合計と2次記憶装置の使用単価との積として式(4.22)のごとく算出しているが、

$$C_S = \sum_a p_a C_2 \quad (4.22)$$

本研究もこの算出方法を利用している。但し、 $p_a$ はエリア $a$ の容量（ページ数）、 $C_2$ は2次記憶装置1ページ当たりの格納費用（円）である。

### 操作上の利点

本ツールの性能評価は次のような操作上の利点を有している。

<sup>注34</sup> GerritsenモデルはNEXTとOWNERのみを考慮

[操作1] 応用プログラムの実行時間を推定するには、その応用プログラムのデータベースへのアクセスパスを指定する必要がある。

●DMSDBDAはまずアクセスの開始グループを指定するようにデータベース管理者に要求する。

●その開始グループが決まると、DMSDBDAはその開始グループと親子関係にあるグループと対応するセットの名前を表示する。

このとき、グループは長方形、またセットは矢印として図4.10のように表示される。

●DMSDBDAは次にアクセスするセットとグループを指定するようにデータベース管理者に要求する。

●同様の表示処理がアクセスの最終グループまで繰り返す。

[利点1] 従って、データベース管理者は設計されたデータベースの構造のすべてを覚えていなくともよい。

[利点2] DMSDBDAは数式モデルを用いて性能を推定するので、その推定結果が高速に表示される。すなわち、データベースの構造が初期設定または修正されるたびに、先に述べた  $T$ 、 $C_A$  および  $C_S$  の値を算出して、表示する。

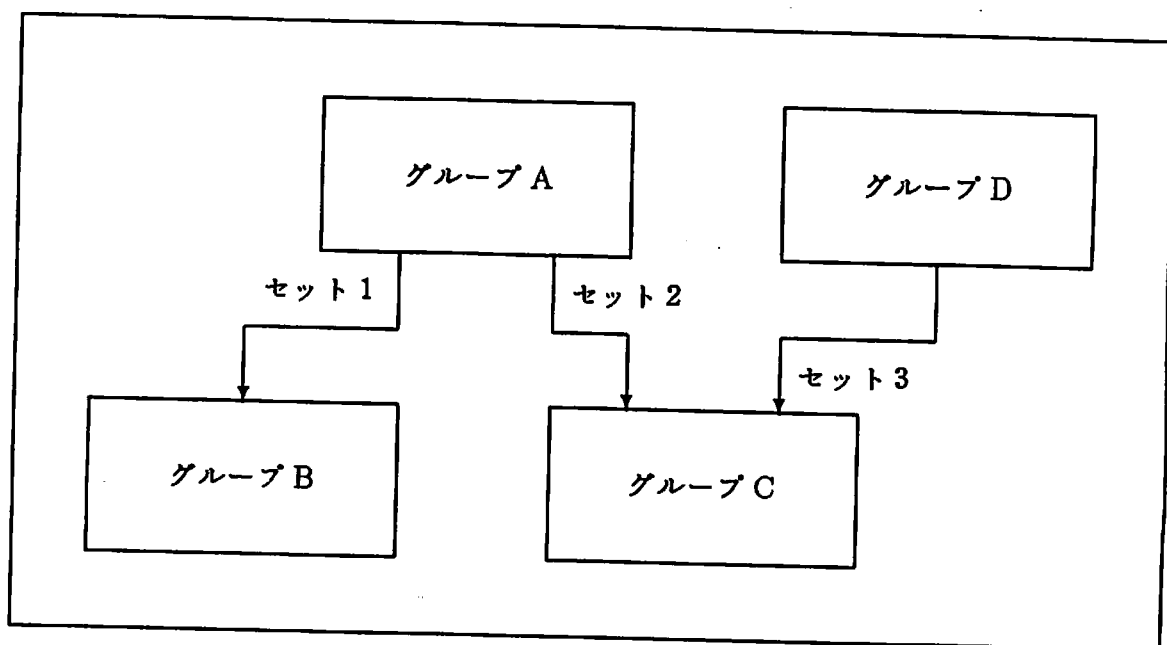


図4.10 アクセスパスの表示画面

#### (4) 論理／物理構造の再設計

データベース管理者は、データベースの現在の設計内容に対する性能評価が、ある希望する水準に達するまで、その論理構造または物理構造を修正することができる。

##### 物理構造の再設計

データベース管理者は、現在の性能を満足できない時または現在の設計内容が実現不可能である時は、その論理構造に影響を与えることなく、物理構造を再設計できる。その再物理設計項目は次に示す7件である。

- グループのロケーション・モード
- セットのオーダ(i.e.ポインタの種類)
- グループのOWNER AREA指定の有無
- グループのエリア割り当て
- グループのページ範囲
- エリアのサイズ
- バッファのサイズ

この再設計項目の選択は、CRT画面上に表示されるメニュー画面を通して実施される。

##### 論理構造の再設計

データベース管理者は、データベースの論理構造も変更することもできる。したがって、同じ問合せ要求に対して、種々の論理構造をもつデータベースを評価することができる。再論理設計項目は次に示す4件である。

- グループの分割
- グループの統合
- セットの追加
- セットの削除

論理構造が再設計されると、前節で述べた初期物理構造が指定される。したがって、以前のデータベースの物理構造はすべてクリアされる。さらに、問合せプログラムの情報(例えば、アクセスパスやアクセス回数など)やグループオカレンスの関連情報(例えば、オカレンス件数やエリア割り当て等)も、クリアされるので、再指定しなければならない。しかしながら、データ項目関連情報(例えば、名前、形式、ピクチャ及びデータ長)、クラスタ、エリア名、スキーマ名、ハードウェア情報及びデータベース管理システム情報は、最初の論理設計の時に指定された内容が維持される。

#### (5) スキーマ生成

DMSDBDAはデータベースの設計と評価が終了すると、その設計内容に合致したスキーマDDL文を自動生成し、2次記憶装置上にスキーマファイルを作成する。

図4.11は、自動生成されたスキーマDDL文を表示したCRT画面の一例である。

但し、スキーマDDLは次の制限のもとで生成される。

- CALCグループのCALCキー項目は重複不可

- メンバオカレンス挿入属性は自動
- セットオカレンスの選択基準はCURRENT[Melco76]とする。

```

1 SCHEMA    SCHM.
2 AREA      A1    CONTAINS  100  NUMBER  1.
3 GROUP     GA
4          WITHIN  A1  RANGE  1   100
5          LOCATION  CALC  A
6          DUPLICATE NOT
7          NUNBER   1.
8          A       TYPE  BINARY.

```

図4.11 スキーマDDL文の画面

#### 4.2.2 DMSDBDAの使用例

本章では、DMSDBDAを用いたデータベースの設計と評価の実例について述べる。先ず、基本的なネットワークデータベースの設計評価の例を次節にて、次に循環構造を有する複雑なネットワークデータベースの設計評価の例を次々節にて述べていく。

##### (1) 例題1：基本構造

#### サンプルデータベース

次のような自動車部品に関する情報をもつデータベースを考える[Hota78]。

- ある部品を製造するために必要な子部品
- ある子部品を使用して組み立てられる部品
- ある部品をある部品メーカーから購入するときの単位
- ある部品をある部品メーカーに発注したときの発注条件（納期、数量）

この自動車部品データベースに次のような11件のデータ項目を定義する：

- PARTCODE：部品番号
- PARTNAME：部品名
- STOCK：部品在庫量
- SUBPART：子部品番号
- AMOUNT：ある部品の製造に必要な各子部品の個数
- MAKERCOD：メーカー番号
- MAKERNAM：メーカー名
- QUALITY：部品メーカーの製造品質
- DATE：部品の納期
- QUANTITY：部品の納入数量
- COST：部品の単位

表4.1～表4.2にデータ項目の内容をまとめる。

表4.1は部品自身のデータであり、例えば、部品名Aの部品番号は1001、その在庫量は5個であることを意味する。

表4.1 部品情報テーブル（例題1）

PARTCODE	PARTNAME	STOCK
1001	A	5
1002	B	2
1003	C	3
1004	D	0
2011	K	4
0001	1	10
0002	2	8
0003	3	9
0010	10	7
0011	11	6
0012	12	10
0013	13	5

表4.2は、部品展開の情報であり、部品番号1001の部品は1002、1003及び0001の子部品から組み立てられ、その所要量は各々1個、2個及び3個であることを意味する。

表4.2 子部品情報テーブル (例題1)

PARTCODE	SUBPART	AMOUNT
1001	1002	1
1001	1003	2
1001	0001	4
1002	1004	1
1002	0010	2
1002	0011	3
1004	1003	1
1004	0013	4
2011	1004	2
2011	0010	3
2011	0012	6
1003	0002	1
1003	0003	4

表4.3は、部品の単価情報であり、部品0002は、メーカーPM01からは30円、またメーカーPM02からは20円の単位で発売されていることを意味する。

表4.3 単価情報テーブル (例題1)

PARTCODE	MAKERCOD	COST
0001	PM01	15
0002	PM01	30
0002	PM02	20
0003	PM01	45
0003	PM02	40
0003	PM03	30
0010	PM01	15
0011	PM03	12
0011	PM04	11
0012	PM01	18
0013	PM03	15
0013	PM04	13

表4.4は、部品の発注情報であり、部品0002はメーカーPM01とPM02の2社へ発注され、各々の納期は77120と77200であり、また各々の発注数量は10個と20個であることを意味する。

表4.4 発注情報テーブル (例題1)

PARTCODE	MAKERCOD	DATE	QUANTITY
0002	PM01	77120	10
0002	PM02	77200	20
0003	PM02	77060	15
0003	PM03	77100	50
0011	PM03	77100	10
0011	PM04	77150	40
0013	PM01	77200	50
0013	PM01	77250	100
0013	PM04	77100	30

表4.5は、部品メーカーの情報であり、例えばメーカー名ABCのメーカー番号はPM01であり、その製造品質はAクラスであることを意味する。

表4.5 購入先情報テーブル (例題1)

MAKERCOD	MAKERNAM	QUALITY
PM01	ABC	A
PM02	DEF	B
PM03	GHI	C
PM04	JKL	C

## データベース設計

DMSDBDAを用いて、上記の自動車部品データベースを設計する。

クラスタの定義 先づ、クラスタを次のように定義する：

- クラスタ1 : (PARTCODE,PARTNAME,STOCK)
- クラスタ2 : (SUBPART,AMOUNT)
- クラスタ3 : (MAKERCOD,MAKERNAM,QUALITY)
- クラスタ4 : (DATE,QUANTITY,COST)

である。

グループの定義 次にグループを4件、次のように定義する：

- グループGPART : クラスタ1



- グループGSUBPART : クラスタ 2
- グループGMAKER : クラスタ 3
- グループGCOST : クラスタ 4

セットの定義 次に、セットを 3 件、図4.12に示す如く定義する。

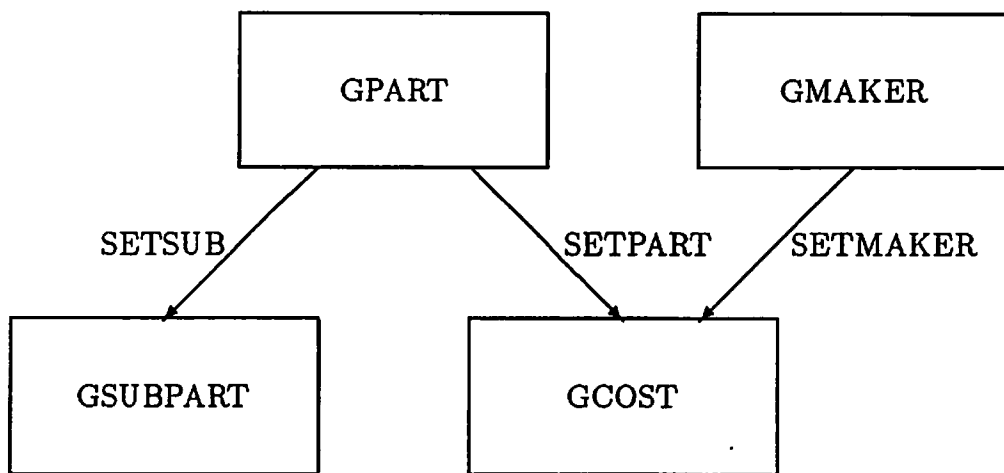


図4.12 自動車部品データベース (例題1)

エリアの定義 次に、200ページから成るエリアAREA1を定義し、すべてのグループがAREA1に含まれるとする。

各グループのページ範囲を次の如く定義する：

- GPART : 1～50ページ
- GSUBPART : 51～100ページ
- GMAKER : 101～150ページ
- GCOST : 151～200ページ

とする。

各グループのオカレンス件数は、

- GPART : 12件
- GSUBPART : 13件

- GMAKER : 4件
- GCOST : 12件

となる。

データ項目の形式、ピクチャア及びデータ長を表4.6に示す。

表4.6 データ項目の形式、ピクチャア、データ長 (例題1)

データ項目名	形式	ピクチャア	データ長 (Byte)
PARTCODE		9	4
PARTNAME	文字	X	2
STOCK	2進数		4
SUBPART		9	4
AMOUNT	2進数		4
MAKERCOD	文字	X	4
MAKERNAM	文字	X	3
QUALITY	文字	A	1
DATE		9	5
QUANTITY	2進数		4
COST	2進数		4

### データベース問合せ情報の指定

自動車部品データベースに対して、次のような問合せ要求を考える。

- 在庫量が5個以下である部品の名前を検索する。
- この検索された部品を組み立てるのに必要な子部品の名前と数量を求める。
- この検索で検索された子部品のメーカー、品質、納期、数量及び単位を検索する。

次に、この問合せ要求の

- 発生回数 : 1000回
- その希望応答時間 : 5分
- アクセス形式 : キー付き

とする。

この問い合わせ要求のアクセスパスは、GPART→GSUBPART→GPART→GCOST→

GMAKERとなる。このパスに従って、各グループのアクセス頻度は、表4.7に示す通りである。

表4.7 問合せのアクセスパスとアクセス比率（例題1）

順序	グループ名	セット名	アクセス比率(%)
1	GPART		50
2		SETSUB	
3	GSUBPART		100
4		SETSUB	
5	GPART		83
6		SETPART	
7	GCOST		100
8		SETMAKER	
9	GMAKER		100

### 性能パラメータの指定

また、性能パラメータは、

- データベースのページ長 $B$ は2KByte
- 主記憶と2次記憶の容量は、各々512KByteと200MByte
- 主記憶と2次記憶の使用単価 $C_1$ と $C_2$ は、各々0.01円と1,000円
- 2次記憶の平均順次アクセス時間と平均直接アクセス時間は各々、5ミリ秒と50ミリ秒
- EDMSのポインタサイズと主記憶占有容量 $Biase$ は各々、4Byteと1KByteと指定した。

### 性能評価の結果

DMSDBDAを用いて、自動車部品データベースを評価すると、次のようになった。

[物理構造1] 初期物理構造（表4.8の物理構造1）の応答時間 $T$ は67分、アクセス費用 $C_A$ は496,715円、記憶費用 $C_S$ は20万円と推定される。

[物理構造2] 次に、GSUBPARTグループはSETSUBセットのメンバであるので、そのロケーションモードをVIA SETSUBにし、同様の理由でGCOSTグループのロケーションモードをVIA SETPARTに変更すると（表4.8の物理構造2）、 $T$ は19分に減少し、 $C_A$ は146,718円に減少するが、 $C_S$ は変わらない。

[物理構造3] データベースの問合せは、GPARTグループからアクセスが開始されるので、そのロケーションモードをCALCからINDEXEDに変更してみると（表4.8の物理構造

3)、 $T$ は5分に減少し、 $C_A$ は39,736円に減少するが、 $C_S$ は変わらない。

[物理構造4] アクセスのエントリグループであるGPARTに対し、OWNER AREAを指定すると(表4.8の物理構造4)、 $T$ と $C_A$ と $C_S$ は変わらない。

[物理構造5] データベース問合せのアクセスは、SETSUBセット及びSETMAKERセットをメンバからオーナへと進むので、そのセットのオーダにOWNERポインタを追加すると(表4.8の物理構造5)、 $T$ は21分に増加し、 $C_A$ は158,946円に増加するが、 $C_S$ は変わらない。

[物理構造6] そこで、物理設計の内容を物理構造4へ戻し、次にバッファの影響を調べるために、その容量 $b$ を1ページから100ページへと増やすと(表4.8の物理構造6)、 $T$ は2分に減少し、 $C_A$ は349,439円に増加するが、 $C_S$ は変わらない。

表4.8 性能評価の結果(例題1)

物理構造	実行時間 $\bar{T}$ (分)	アクセス費用 $C_A$ (円)	記憶費用 $C_S$ (円)
1	67	496,715	200,000
2	19	146,718	200,000
3	5	39,736	200,000
4	5	39,736	200,000
5	21	158,945	200,000
6	2	349,439	200,000

従って、表4.8に示した6種の物理構造について、次の如く結論される：

- 問合せの応答時間をなるべく小さくしたいという評価基準のもとでは、物理構造6が推奨される。
- 費用をできる限り低くしたいという評価基準のもとでは、物理構造3または4が推奨される。

## (2) 例題2：循環構造

### グループ化

図4.13に示すごとく、まずデータ項目タイプを16件定義し、その項目から5件のクラスタ(1番から5番)を作成する。

次に、そのクラスタから5件のグループ(G1、G2、G3、G4、G5)を作成する。

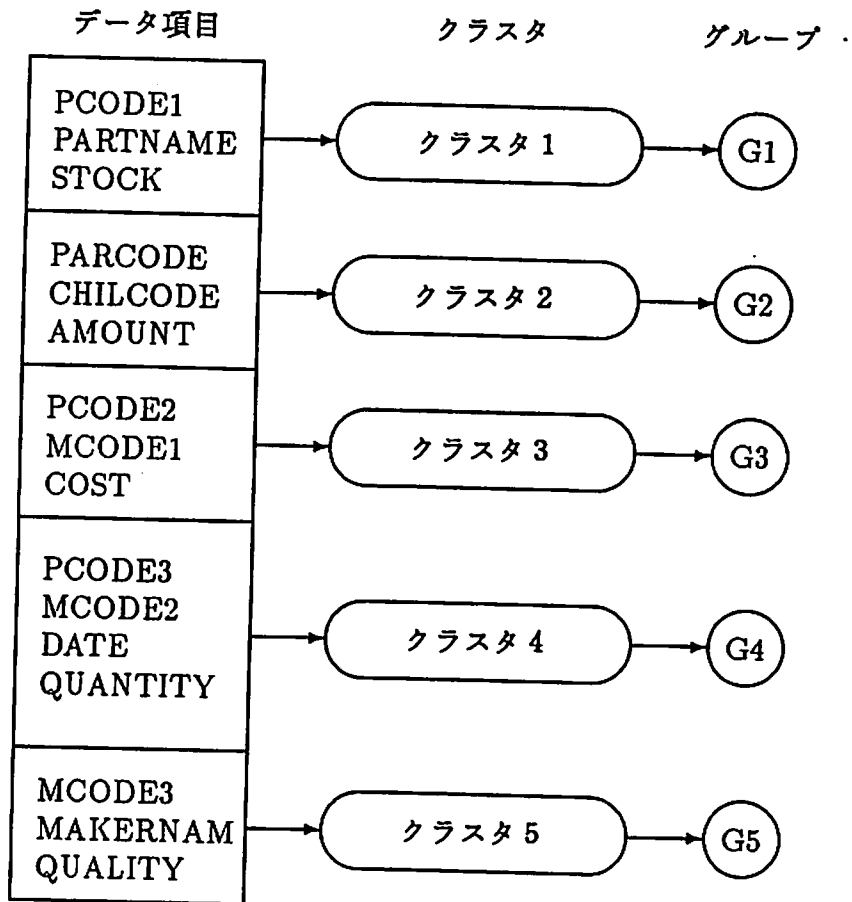


図4.13 データ項目のクラスタ化とクラスタのグループ化 (例題 2)

### セットの定義

次に、セットタイプを6件作成する。G1からG2へセットS1とS2、G1からG3へセットS3、G1からG4へセットS4、G5からG3へセットS5、G5からG4へセットS6を作成する。

以上の定義から図4.14に示される論理構造が設計された。

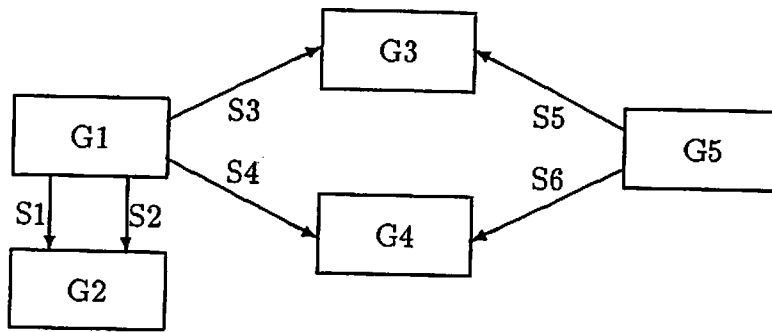


図4.14 論理構造の設計 (例題 2)

### エリアの定義

次に、200ページの容量をもつエリアAREA2を定義し、グループのページ範囲はすべて1~200ページとする。各グループのオカレンス件数は、G1が100件、G2が200件、G3が500件、G4が1,000件、G5が150件とする。データ項目のデータ長とデータ形式は、すべて10Byteの文字型とする。

### データベース問合せ情報の指定

データベースへの問合せとしては表4.9に示されるアクセスパスとアクセス比率をもつものを考える。そのアクセスはG1から開始され、S1を經由しG2へ、S2を經由しG1へ、その後G3からG5へと進み、G4で終了する。この問合せの実行回数を2,000回とする。

表4.9 問合せのアクセスパスとアクセス比率 (例題 2)

順序	グループ名	セット名	アクセス比率 (%)
1	G1		10
2		S1	
3	G2		40
4		S2	
5	G1		20
6		S3	
7	G3		60
8		S5	
9	G5		15
10		S6	
11	G4		45

### 性能パラメータの指定

また、性能パラメータは、前節と同一の値を指定した。

## 性能評価の結果

性能評価の結果は、次のようになった。

[物理構造1] 初期物理構造(表4.10)の性能評価結果は、実行時間 $T$ が60分、アクセス費用 $C_A$ が112,076円、記憶費用 $C_S$ が20万円と推定される。

[物理構造2] メンバグループG2、G3、G4のロケーションモードを各々、VIA S1、VIA S3、VIA S6に変更すると(表4.10の物理構造2)、 $T$ は33分に減少し、 $C_A$ は61,132円に減少するが、 $C_S$ は変わらない。

[物理構造3] 次に、アクセス開始グループG1のロケーションモードをINDEXEDに変更すると(表4.10の物理構造3)、 $T$ は3分に減少し、 $C_A$ は6,113円に減少するが、 $C_S$ は変わらない。

[物理構造4] 次に、G1にOWNER AREAを指定すると(表4.10の物理構造4)、 $T$ と $C_A$ と $C_S$ は変わらない。

[物理構造5] 次に、メンバからオーナーがアクセスされるセットS2とS5にOWNERポイントを追加すると(表4.10の物理構造5)、 $T$ は4分に増加し、 $C_A$ は9,169円に増加するが、 $C_S$ は変わらない。

[物理構造6] 次に、6件のセットのすべてに、PRIORポイントを追加すると(表4.10の物理構造6)、 $T$ と $C_A$ と $C_S$ は変わらない。

表4.10 性能評価の結果(例題2)

物理構造	実行時間 $T$ (分)	アクセス費用 $C_A$ (円)	記憶費用 $C_S$ (円)
1	60	112,076	200,000
2	33	61,132	200,000
3	3	6,113	200,000
4	3	6,113	200,000
5	4	9,169	200,000
6	4	9,169	200,000

### 4.3 データベース管理システムのバッファ管理方式

以下、4.3.1節ではMUMPSデータベースの構造とそのアクセス方式、4.3.2節ではMUMPSプログラムファイルとビットマップファイル、4.3.3節では3種類のバッファ置換アルゴリズム、4.3.4節ではそのバッファ置換アルゴリズムのシミュレーション評価方法、そして4.3.5節ではそのシミュレーション評価結果について述べていく。

#### 4.3.1 データベースの構造とアクセス方式

##### データベース構造

MUMPSデータベースは図4.15に示されるように、スパース状の多次元配列構造をしており、レベル付けされたレコードの階層からなる。

レコードはグローバル変数と呼ばれ、変数名と添字値がその一意な識別子として指定される。更に、主記憶上にのみ存在する変数と区別するために、グローバル変数名の先頭には記号が付記される。又、グローバル変数のレベル付けは添字を用いて実現され、添字の個数がレベル値になる。

例えば、 $\wedge G(1,3)$ は、

- 変数名：G
- 添字値：(1,3)
- レベル値：2

のグローバル変数を意味する。

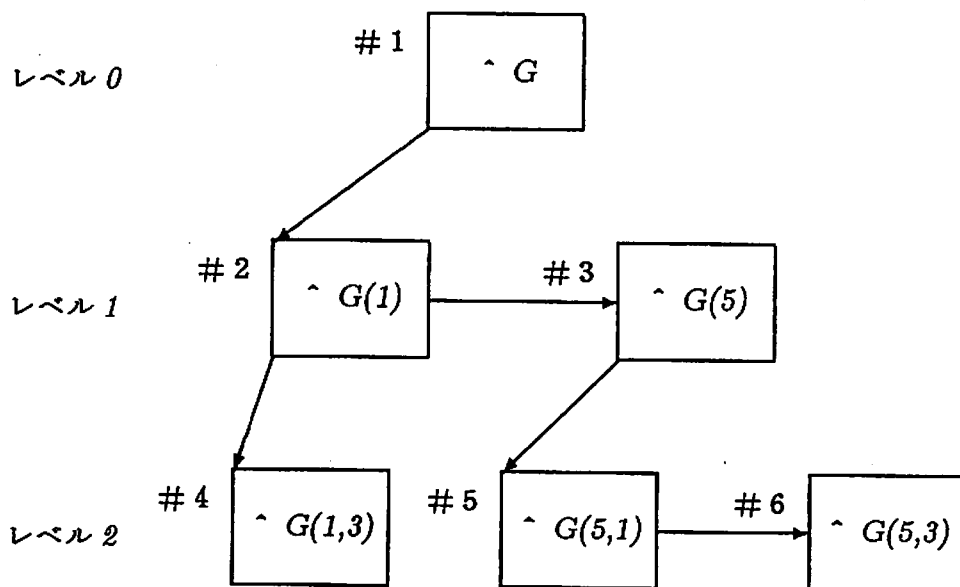


図4.15 データベースの構造



データベースは先に述べたごとく階層化されているが、最上位の変数はディレクトリ変数と呼ばれ、階層構造のルートとして機能する。例えば、図4.15の  $G$  がディレクトリ変数である。

グローバル変数レコードはブロッキングされて、ディスク上に格納される。レベル0のブロックがグローバルディレクトリブロック、その他のブロックがグローバルデータブロックと呼ばれる。図4.15に示されるように、レベル  $l$  にあるブロックはレベル  $l-1$  のブロック、レベル  $l$  の他のブロック（但し、存在するときに限る）、及び  $l+1$  のブロック（但し、存在するときに限る）とポインタを用いて連鎖される。

## アクセス方式

MUMPSデータベースのアクセス内容は、

- 変数の生成
- 変数の参照
- 変数の更新
- 変数の削除

であり、変数名と添字のみを指定するだけで実現される。

### (1) 参照、更新、削除

図4.15の変数  $G(5,3)$  をアクセスするときを例にしてアクセス方式を説明する。アクセスはまずディレクトリブロック（ブロック # 1）から開始され、ディレクトリ変数  $G$  が検索される。ブロック # 1 に  $G$  が格納されているので、レベル0のブロックの探索は完了し、アクセスはブロック # 1 から # 2 へと進む。アクセスする変数のレベル1の添字は5であるので、レベル1のブロックリストの中から  $G(5)$  を探索する。ブロック # 2 には  $G(5)$  が格納されていないので、ブロック # 2 から # 3 へと探索が進む。ブロック # 3 には  $G(5)$  が格納されているので、レベル1のブロックの探索は完了し、アクセスはブロック # 3 から # 5 へ進む。以後、同様な探索が続き、最終的にブロック # 6 がアクセスされる。

### (2) 生成

変数の参照、更新および削除は上述したアクセス方式に従うが、生成はこのほかに別の処理が加わる。

図4.15のデータベースに変数  $G(1,10)$  を新たに生成するときを説明する。アクセスはまず  $G(1,10)$  が存在するとして、上述した方式で実施され、ブロック # 1 から始まり、ブロック # 2、# 4 と進み、 $G(1,10)$  が未生成変数であることが分かる。

$G(1,10)$  はブロック # 4 に格納されることになるが、ブロック # 4 がオーバーフローするならば、新たに空きブロックが確保され、そのブロックに  $G(1,10)$  が格納され、同時にブロック # 4 とポインタチェーンされる。

### (3) フルシンタックス法とネイキッドレファレンス法

アクセス方式は先述した方法（フルシンタックス法と呼ぶ）のほかに、直前にアクセスされたブロックが属すチェーンの先頭ブロックからアクセスが開始される方式（ネイキッドレファレンス法と呼ぶ）がある。例えば、図4.15のデータベースにおいて変数  $G(5,1)$  がアクセスされた直後に、ネイキッドレファレンス法を利用して  $G(5,3)$  をアクセスすると（構文上、(3)と書く。）、探索はブロック#1からではなく、ブロック#5から開始され、高速なアクセスが実現される。

## 4.3.2 プログラムファイルとビットマップファイル

### プログラムファイル

MUMPS言語で記述されたプログラムはデータベースに登録することができる。プログラムファイルの構造は2レベルの階層からなるグローバルファイルと同じであり、レベル0及びレベル1のブロックは各々プログラムディレクトリブロック及びプログラムデータブロックと呼ばれる。

プログラムファイルのアクセス内容は、

- プログラムの登録
- プログラムのロード
- プログラムの削除

の3種類であり、プログラムの名前のみを指定することによりアクセスが実現される。

### ビットマップファイル

ビットマップはデータベース内のすべてのブロック利用状態を管理するための情報であり、それ自身もディスク装置上にファイルとして格納される。

ブロックの利用状態は、

- 現在使用中のブロック（使用ブロックと呼ぶ）
- 現在未使用のブロック（空きブロックと呼ぶ）
- 削除された変数の削除以前の下位ブロック（削除ブロックと呼ぶ）

の3種類である。

## 4.3.3 バッファ置換アルゴリズム

データベースをアクセスした後そのアクセス結果をバッファに取り込むとき、バッファプール内のどのバッファを置換するかを決定するアルゴリズムがバッファ置換アルゴリズムである。

本研究ではバッファ置換アルゴリズムとして、

- 従来から普及しているLRU法
- MUMPSデータベース固有の構造とアクセス方式の特徴をいかしたアルゴリズムとして提案されているBPL法
- 本研究においてBPL法を改良したBPL\*法

の3種類を取り扱う。

以下、各々のアルゴリズムの内容について述べる[Wada79-1, Wada79-2]。

## LRU法

LRU法とは最も長い間アクセスされなかったバッファを解放して、データベースの最新アクセス結果をそのバッファに取り込むアルゴリズムである。

### (1) 手続き

以下にLRU法の手続きをまとめる。

1. アクセス要求の発生したブロックがバッファ上に存在されているか否かを判定する。
  - (a) 保存されているときは処理4.へ分岐する。
  - (b) 保存されていないときは次へ進む。
2. 空きバッファがあるか否かを判定する。
  - (a) あるときは次の処理3.へ進む。
  - (b) ないときは処理7.へ分岐する。
3. アクセス要求の生じたブロックを当該空きバッファに読み込む。
4. 当該バッファを最新の使用バッファにする。
5. 他の使用バッファの最近の参照時点を一時点分だけ古くする。
6. 処理を終了する。
7. 使用バッファ群の中から最近の参照時点が最も古いバッファを抽出し、解放する。
8. 当該バッファの内容を、アクセス要求の生じたブロックに置き換える。
9. 処理4.へ分岐する。

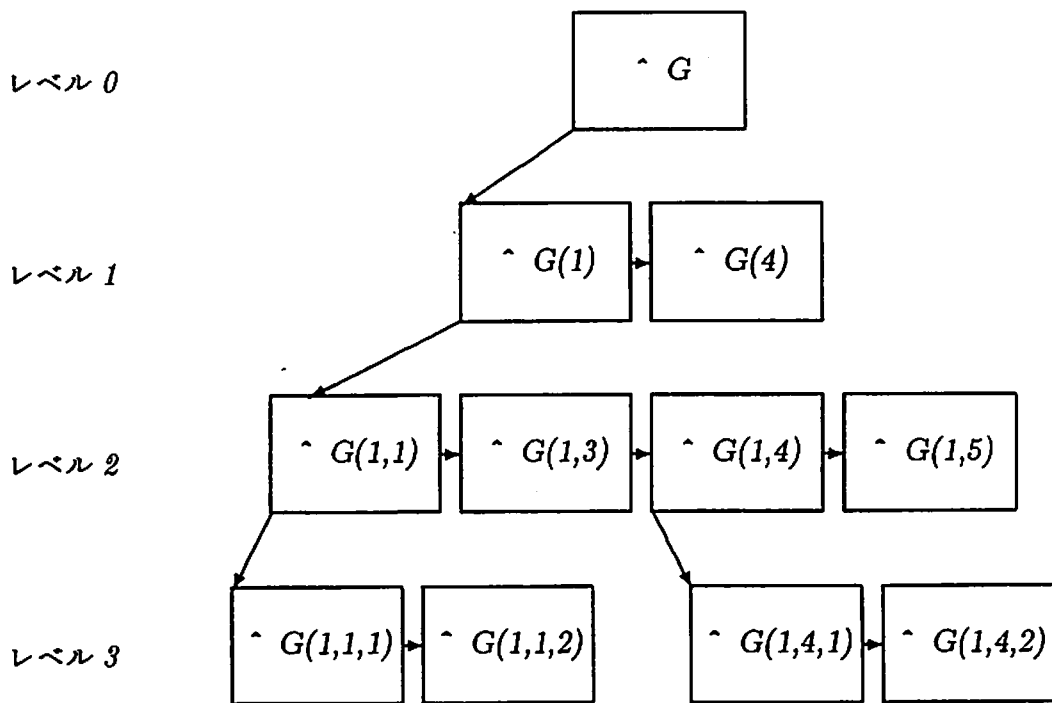


図4.16 グローバルファイルの例—その1

(2) 例題1 : フルシンタックス法

図4.16のグローバル・ファイル为例にしてLRU法を説明する。但し、バッファ面数は4とする。

グローバル変数 $\wedge G(1,4,1)$ をアクセスするとき

- 先づディレクトリ変数 $\wedge G$ がアクセスされ、
- 続いて $\wedge G(1)$ 、 $\wedge G(1,1)$ 、 $\wedge G(1,3)$ 、 $\wedge G(1,4)$ の順にアクセスされ、
- 最後にアクセスのターゲットである $\wedge G(1,4,1)$ がアクセスされる。

変数 $\wedge G(1,3)$ が読み込まれたときから、変数 $\wedge G(1,4,1)$ が読み込まれるまでのバッファの内容が変化していく様子が図4.17であり、バッファの肩の数字はバッファの内容の新しさを示し、数字が小さいほど内容が新しいことを意味する。

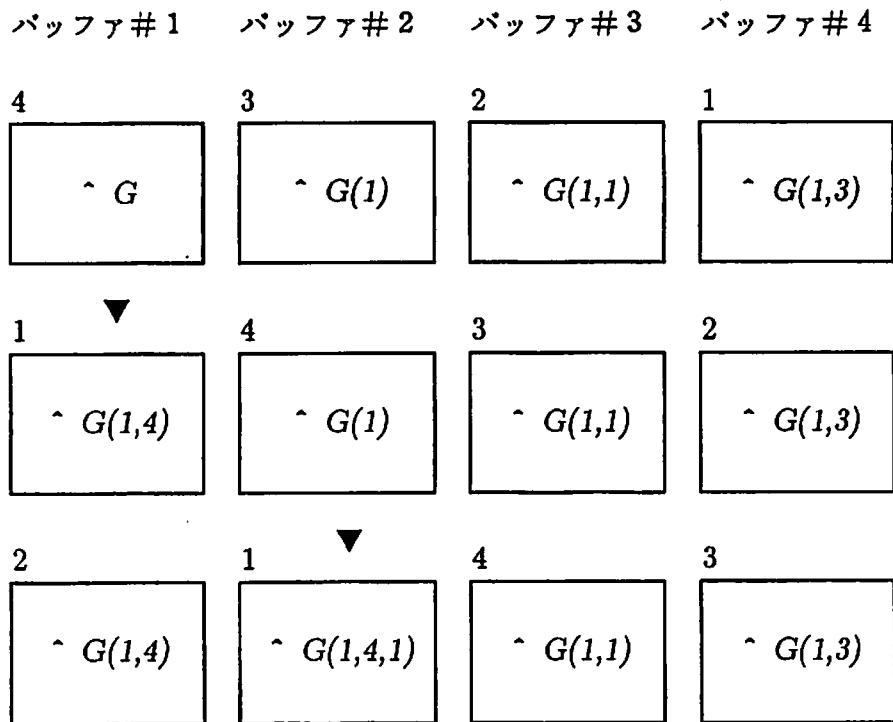


図4.17 LRU法によるバッファ置換の例（フルシンタックス法）

### (3) 考察1

LRU法のもとでMUMPSグローバル・ファイルアクセスした場合、アクセス頻度が高最も高いと考えられるディレクトリ変数が比較的早い時期にバッファ上から消失する性質がある。従って、次に別の変数、例えば  $G(1,1,1)$  をアクセスするときは、再びディレクトリブロックをディスクから読み込ませなければならない。

### (4) 例題2：ネイキッドレファレンス法

次にネイキッドレファレンス法の場合について図4.16を例にして説明する。但し、バッファは3面確保されているとする。

$\wedge G(1,1)$  がアクセスされた直後に、続いてネイキッドレファレンス法を利用して、新しい変数  $\wedge G(1,6)$  が生成されるまでのバッファの内容は図4.18の様に变化する。

●ネイキッドレファレンスが発生したときのグローバルファイルのカレントレベルは2であるので、アクセスの起点はディレクトリ変数ではなく、レベル2のヘッダブロック(この例では変数  $\wedge G(1,1)$  のブロック)である。

●変数  $G(1,5)$  が読み込まれたとき、変数  $G(1,6)$  が未生成であることがわかるので、ビットマップブロックが読み込まれて、未使用ブロックが確保されて変数  $G(1,6)$  が生成される。

(5) 考察2

この例に見られるようにLRU法のもとで、ネイキッドレファレンスを行なうとアクセスの起点となるカレント・レベルのヘッダブロックがバッファ上に保存されていない可能性がある。

以上の二例の考察からLRU法によるバッファ管理はMUMPSデータベースに適しているとは言えない。

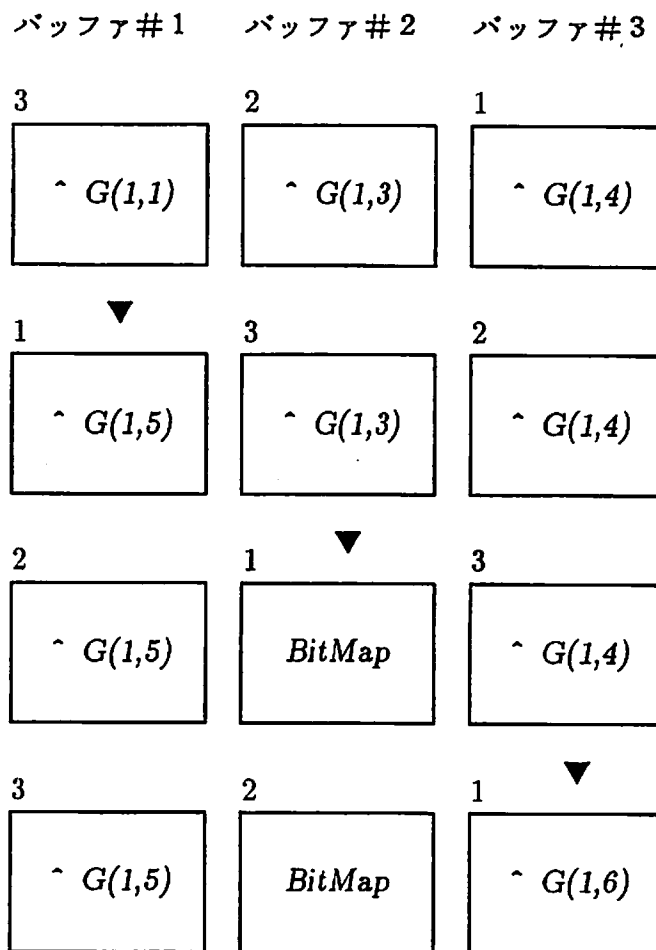


図4.18 LRU法によるバッファ置換の例（ネイキッドレファレンス法）

BPL法

BPL法とはバッファごとにその内容に応じた優先順位（以後、*bpl*と呼ぶ）を定義し、

*bpl*が最も低いバッファを解放して、データベースの最新アクセス結果をそのバッファに読み込むアルゴリズムであり、MUMPS開発委員会から提案されたバッファ置換アルゴリズムである[Wass75-2]。しかし、この提案にはグローバルディレクトリブロックとグローバルデータブックの*bpl*のみが明記されているに過ぎず、次の4点が不明である。

- [不明点1] プログラムディレクトリブロックの*bpl*
- [不明点2] プログラムデータブックの*bpl*
- [不明点3] ビットマップのブロックの*bpl*
- [不明点4] 同一*bpl*を持つバッファが2個以上存在したときの置換方式

上記の不明点1から3に対して本研究では表4.11に示す*bpl*を導入した。すなわち、各々の不明点に対して次のような解決策を考案した。

[解決策1] プログラムディレクトリブロックのアクセス頻度はグローバルディレクトリブロックと同程度に高いと推測できることから、その*bpl*はグローバルディレクトリブロックと同じに256とした。

[解決策2] プログラムデータブックは一度アクセスされたならば、同ブロックが近い将来再びアクセスされる可能性は他の種類のブロックに比べて非常に少ないと考えられるので、その*bpl*は最低の0とした。

[解決策3] ビットマップのブロックはグローバル変数の生成、更新および削除又はプログラムの登録や削除のときに同時に使用されるので、そのアクセス頻度はかなり高いと考えられるが、グローバル変数の参照やプログラムのロードのときには使用されないことから、その*bpl*はディレクトリブロックよりも低く、且つ、レベル1のグローバルデータブックよりも高い値、200とした。

[解決策4] 先述した不明点4の現象は同レベルのブロックが複数個アクセスされたときに発生するが、その解決法として本研究では前節で述べたLRU法を導入した。

表4.11 バッファの優先順位

バッファの内容	<i>bpl</i>
最新アクセスブロック	384
カレントアクセスレベルの先頭ブロック	320
グローバル/プログラムのディレクトリブロック	256
ビットマップブロック	200
レベル <i>l</i> のグローバルデータブック	$256/(l+1)$
プログラムデータブック	0

従って、BPL法とは文献[Wass75- 2]で提案されている**bpl**の基本的な思想に、本研究にて先述した事柄を肉付けしたバッファ置換アルゴリズムである。

### (1) 手続き

以下に、BPL法の手続きをまとめる。

1. アクセス要求が生じたブロックが、バッファ上に保存されているか否かを判定する。

(a) 保存されているときは処理 4.へ分岐する。

(b) 保存されていないときは次へ進む。

2. 空きバッファがあるか否かを判定する。

(a) あるときは次の処理 3.へ進む。

(b) ないときは処理 8.へ分岐する。

3. アクセス要求の生じたブロックを当該空きバッファに読み込む。

4. 当該バッファの**bpl**を384にする。

5. いままで**bpl**が384であったバッファの内容を調べる。

(a) カレントアクセスレベルの先頭ブロックであるときは、**bpl**を320にする。

(b) ディレトリブロックであるときは**bpl**を256にする。

(c) ビットマップブロックであるときは**bpl**を200にする。

(d) グローバルデータブロック (レベル*l*) であるときは**bpl**へ $256/(l+1)$ の結果をセットする。

(e) プログラムデータブロックであるときは**bpl**を0にする。

6. 処理 5.で設定した**bpl**を持つバッファが他に存在するか否かを判定する。

(a) 存在するときは当該バッファは最新内容のバッファにし、他の同**bpl**バッファは最近参照時点を一時点分古くする。

(b) 存在しないときは次へ進む。

7. 処理を終了する。

8. 使用バッファ群から最小**bpl**のバッファを抽出する。

9. 最小**bpl**のバッファが複数個存在するか否かを判定する。

(a) 存在するときはそのバッファ群から最近参照時点が最も古いバッファを抽出する。

(b) 存在しないときは次へ進む。

10. 抽出したバッファを解放し、アクセスの要求が生じたブロックに置き換える。

11. 処理 4.へ分岐する。



## (2) 例題3：フルシンタックス法

図4.16のグローバル・ファイルを例にしてBPL法を説明する。アクセスのターゲット変数は前節と同様に  $G(1,4,1)$  とし、フルシンタックス法を用いる。但し、バッファ面数は4とする。アクセスは、

- 先づディレクトリ変数  $\wedge$  から開始され、
- $\wedge G(1)$ 、 $\wedge G(1,1)$ 、 $\wedge G(1,3)$ 、 $\wedge G(1,4)$ の順にアクセスが続き、
- 最後に  $\wedge G(1,4,1)$  がアクセスされる。

図4.19は変数  $G(1,3)$ 、 $G(1,4)$ 、そして  $G(1,4,1)$  が読み込まれたときのバッファの内容の変化を示しており、バッファの肩の数字は  $bpl$  である。

1. 先づ始めにディレクトリ変数  $G$  がバッファ # 1 に、次に  $G(1)$  がバッファ # 2 に、 $\wedge G(1,1)$  が # 3 に、 $\wedge G(1,3)$  がバッファ # 4 に読み込まれる。

(a) このときバッファ # 1 の内容はディレクトリブロックであるので、その  $bpl$  は256となり、

(b) バッファ # 2 の内容はレベル1のブロックであるので、その  $bpl$  は128となり、

(c) バッファ # 3 の内容はレベル2のブロックであるので、 $bpl$  は85.3となり、

(d) バッファ # 4 の内容はレベル3のブロックであるので基本的な  $bpl$  は85.3であるが、現時点では最新アクセスブロックであるので、次に新しいブロックがアクセスされるまでは一時的に  $bpl$  は384となる。

この状態が図4.19の第1段階である。

2. 次に変数  $G(1,4)$  を読み込むとき、バッファがすべて使用中であるため、 $bpl$  がもっとも小さいバッファ # 3 が解放され、 $\wedge G(1,4)$  が読み込まれる。

(a) すると、バッファ # 3 の内容が最新アクセスブロックであるので、その  $bpl$  が384となり、

(b) 今まで最新アクセスブロックが格納されていたバッファ # 4 の  $bpl$  は384から85.3に変更される。なぜならば、バッファ # 4 の内容はレベル2のブロックであるからである。

この状態が図4.19の第2段階である。

3. 次に、 $G(1,4,1)$  が読み込まれるときも、同様の処理が行なわれ、図4.19の第3段階のような状態になる。

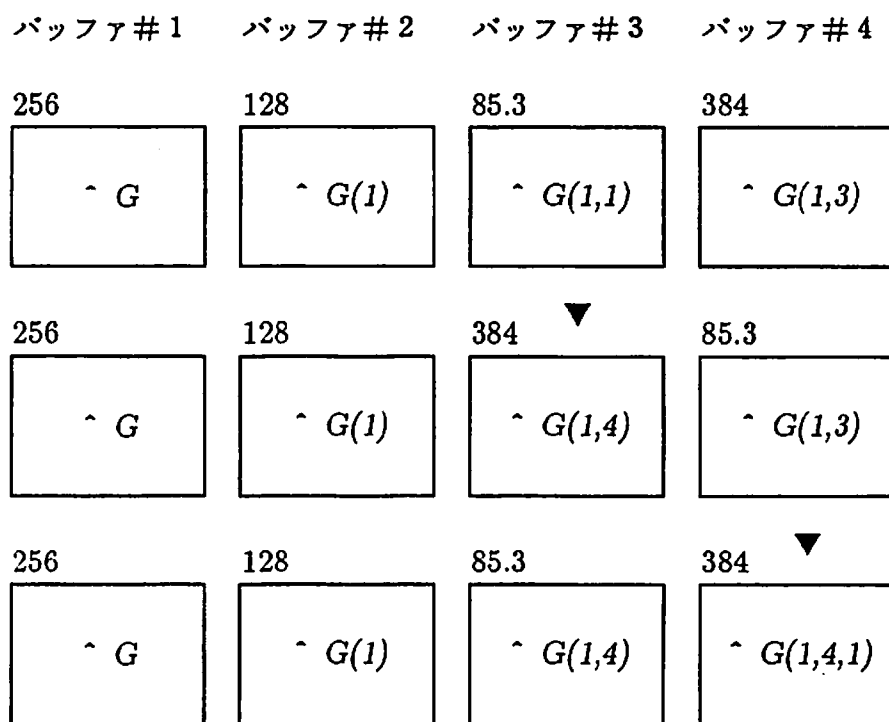


図4.19 BPL法によるバッファ置換の例（フルシンタックス法）

### (3) 考察3

この図4.19の例からも分かるように、BPL法のもとでMUMPSデータベースをアクセスした場合、ディレクトリ変数がLRU法よりも長時間バッファ上に保存されることが期待されるので、非常に好都合である。

例えば、次に変数  $G(1,1,1)$  をアクセスするとき、ディレクトリ変数  $G$  のブロックがバッファ上に保存されているので、ディスクからブロック読み込み操作が一回減少する。この性質はデータベースアクセスの応答時間が減少するため、非常に有利である。

### (4) 例題4：ネイキッドレファレンス法

次にネイキッドレファレンス法の場合について説明する。前節の例と同様に、図4.16のグローバルファイルをアクセス対象とし、変数  $G(1,1)$  が直前にアクセスされた時、続いてネイキッドレファレンス法を利用して、新たに変数  $G(1,6)$  を生成する場合について説明する。但し、バッファ面数は3とする。

1. 変数  $\wedge G(1,1)$  がアクセスされたときの状態が図4.20の第1段階であり、(a) バッファ#1には変数  $G(1,4)$ 、バッファ#2には  $G(1,5)$ 、バッファ#3に最新アク

セスの変数 $G(1,1)$ が読み込まれており、 $bpl$ は各々85.3、85.3、384である。バッファ#1と#2の $bpl$ の右隣りのカッコ内の数字は同一 $bpl$ を持つバッファの内容の新旧度を示しており、この例ではバッファ#2の内容がバッファ#1よりも新しいことを意味する。

(b)変数 $G(1,1)$ がアクセスされた直後にネイキッドレファレンス(6)が発生したので、カレントアクセスレベル、すなわちレベル2のヘッダブロックからアクセスが開始される。

(c)この時該当ヘッダブロックはバッファ上に保存されているので、ディスク入力が一回節約される。

2.次に変数 $G(1,3)$ を読み込むとき、

(a) $bpl$ が最小なバッファが2ヶ存在するので、その内容がもっとも古いバッファ、すなわちバッファ#1が解放されて、 $G(1,3)$ が読み込まれ、最新アクセスブロックとなり、

(b)一方今までの最新アクセスブロックが格納されていたバッファ#3はその $bpl$ が384から320に変更される。これは、バッファ#3の内容がカレントアクセスレベルのヘッダ・ブロックであるからである。

このときのバッファの状態が図4.20の第2段階である。

3.以後、変数 $G(1,4)$ 、 $G(1,5)$ 及びビットマップが読み込まれ、新しい変数 $G(1,6)$ が生成される。その様子が、図4.20の第3段階から以降である。

#### (5) 考察4

この例に見られるように、BPL法を用いると、ネイキッドレファレンスが発生したとき、アクセスの起点となるヘッダブロックが、LRU法よりも長い間バッファ上に保存されており、この性質は非常に有利である。

以上の考察からMUMPSデータベースをアクセスするとき、フルシンタックス法を用いるときもネイキッドレファレンス法を用いるときも共に、LRU法よりもBPL法の方が優れていることが分かる。

### BPL\*法

先述したように、BPL法は同じ $bpl$ を持つバッファ群に対してLRU法を用いたが、同レベルのブロックのチェーンが長くなると先頭ブロックがバッファ不在を起す可能性が強くなる。例えば、図4.21に示すようにグローバル変数の種類が増加するとディレクトリブロックのチェーンが長くなり、BPL法を用いるとそのチェーンの中でアクセス頻度が最も高い先頭のディレクトリブロックが最も先にバッファ不在になる。

こうしたBPL法の欠点を改善するために筆者は次の解決策を考案した。

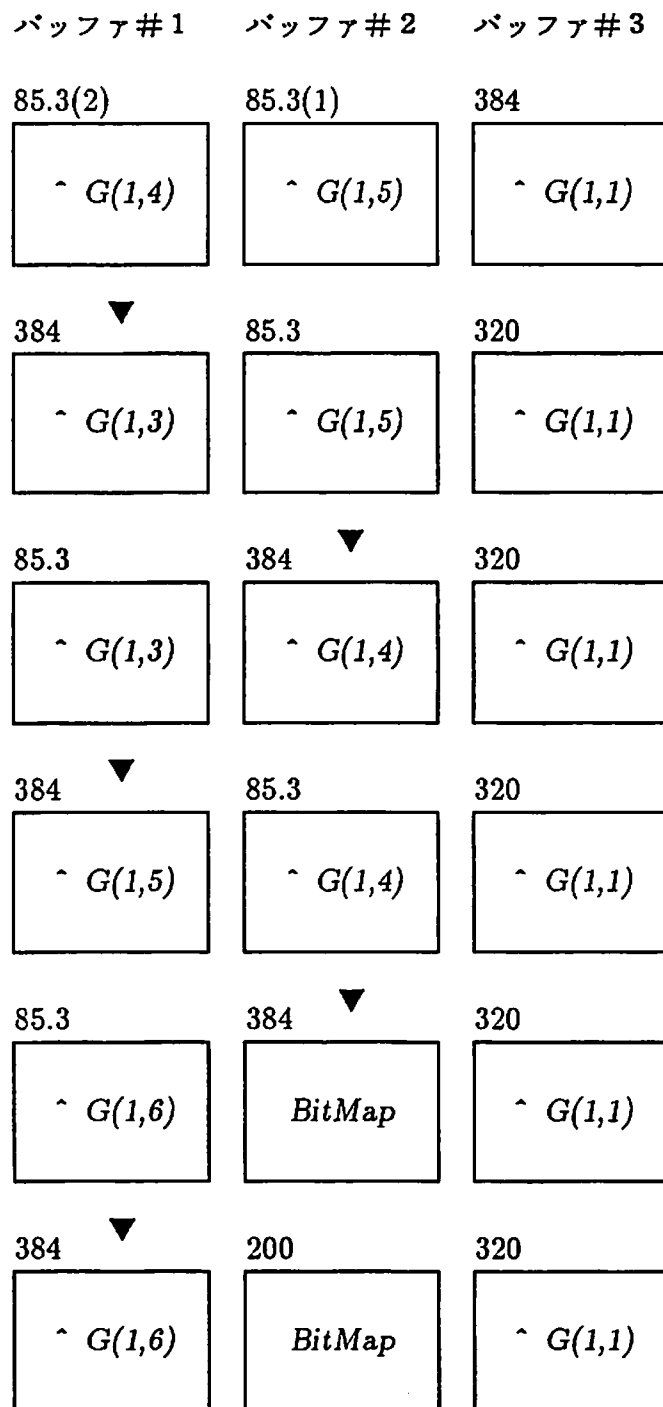


図4.20 BPL法によるバッファ置換の例（ネイキッドレファレンス法）

[解決策4\*] あるレベルのブロックチェーンの各ブロックに対してそのチェーンの各先頭ブロックからの相対的なロケーションであるブロックの深さ（以後、*Bdpt*と呼ぶ）という情報量を導入した。

例えば、図4.21のブロック#3の*Bdpt*は3である。

BPL\*法はBPL法を基にし、前節で指摘した不明点4に対しLRU法の代りに*Bdpt*という新しい情報量を導入したバッファ置換アルゴリズムである。

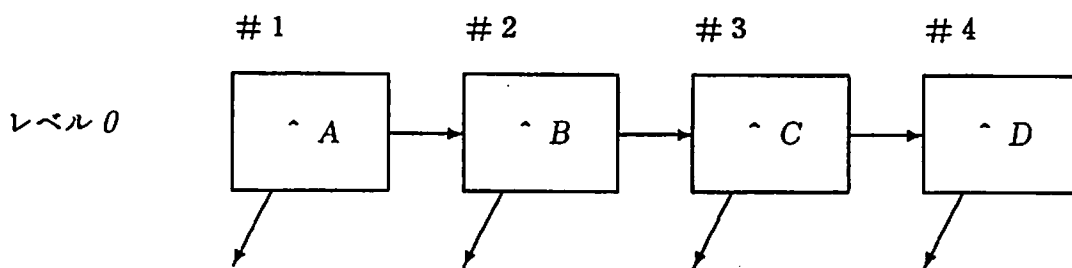


図4.21 グローバルダイレクトリブロックチェーン

(1) 手続き

以下に、BPL\*法の手続きをまとめる。

1. BPL法と同じ。
2. BPL法と同じ。
3. BPL法と同じ。
4. BPL法と同じ。
5. BPL法と同じ。
6. 当該バッファの*Bdpt*を設定する。
7. 処理を終了する。
8. 使用バッファ群から最小*bpl*のバッファを抽出する。
9. 最小*bpl*のバッファが複数個存在するか否かを判定する。
  - (a) 存在するときはそのバッファ群の中から最大*Bdpt*のバッファを抽出する。
  - (b) 存在しないときは次へ進む。
10. 抽出したバッファを解放し、アクセス要求が生じたブロックを読み込む。
11. 処理4.へ分岐する。

(2) 例題5 : BPL\*法とBPL法の比較

図4.22のグローバルファイル为例にしてBPL\*法を説明する。 但し、バッファ面数は3とする。

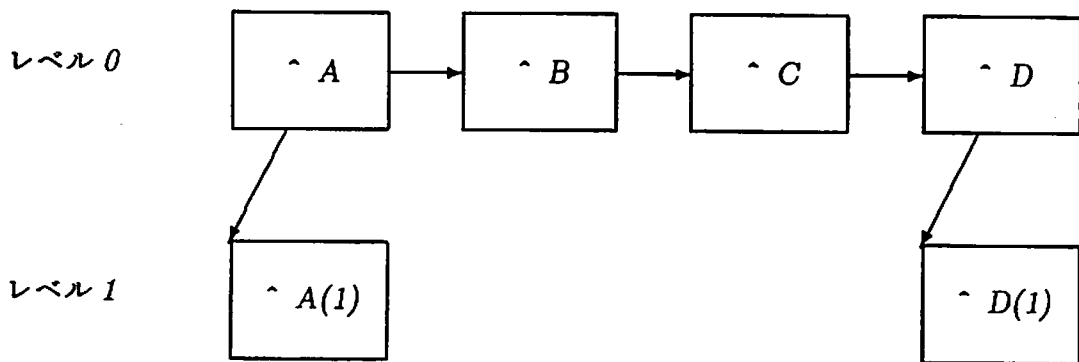


図4.22グローバルファイルの例—その2

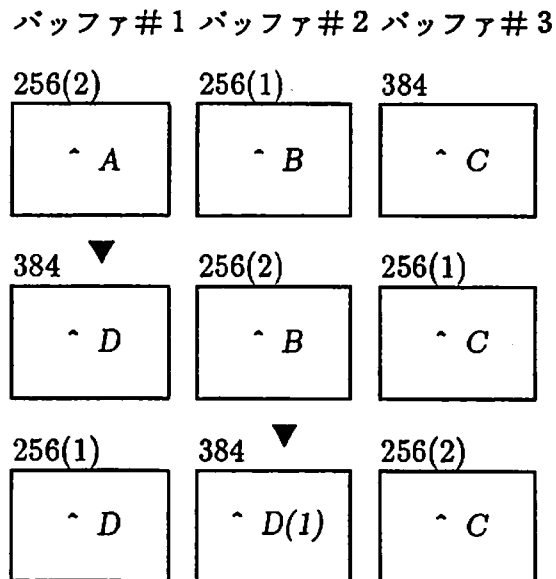


図4.23 BPL\*法とBPL法の比較—その1 .BPL法によるバッファ置換の例

1.今、変数  $D(1)$  をアクセスし、BPL法に従ってバッファを管理すると、バッファの内容は図4.23のように変化する。ディレクトリブロックチェーンの中で最もアクセス頻度が高いディレクトリヘッダブロック(この例では変数  $A$  のブロック)が早い時期にバッファ上から消失している。このため、次に別の変数、例えば  $A(1)$  をアクセスするときディレクトリ変数  $A$  のブロックを再びディスクから読み込まなければならない。

2.BPL\*法と前節のBPL法を比較するために、図4.23のアクセス例と同じもの、すなわち変数  $D(1)$  をアクセスしたときのBPL\*法によるバッファの内容の変化が図4.24である。

但し、バッファ面数も3とする。図4.24のバッファの肩の数字はbplであり、右隣りのカッコ内の数字はBdptである。

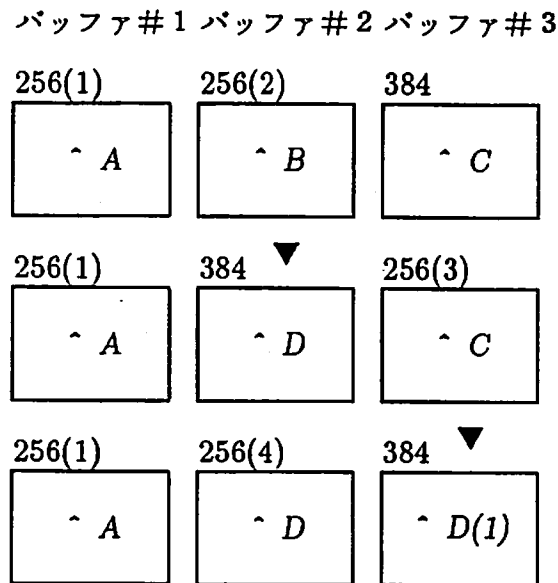


図4.24 BPL\*法とBPL法の比較—その2.BPL\*法によるバッファ置換の例

### (3) 考察5

図4.23と図4.24を比較すると、BPL\*法を用いると、BPL法を用いた時よりもディレクトリヘッダブロックがより長いバッファ上に保存されることが分かる。この性質はMUMPSデータベースには好都合である。

#### 4.3.4 シミュレーションの方法

データベースとアクセス要求およびバッファ置換を計算機上にてシミュレーションするために、本研究では乱数発生によるシミュレーション方法を用いた。データベースはま

ず初期負荷が与えられ、次にアクセス要求が発生し同時にバッファの置換が実施される。  
以下、初期負荷とアクセス要求のシミュレーション方法について述べる。

### データベースの初期負荷

初期負荷は計算機上で乱数を発生させることによりシミュレーション生成され、最小構成単位はブロックであり、ブロックはファイル内で一意な識別番号*Bid*を持つ。

#### (1) グローバルファイル生成

グローバルファイルをシミュレーション生成するに当り、まず、ブロックの総数*T*と最大レベル値*L*が外部から指定される。次に、ブロック*i*のレベル  $D_{Li}$ を決めるために次の仮定を設ける。

[仮定1] グローバルファイルはレベルが深くなる程、データ量が増加する。

そこで、指数分布する乱数（以後、指数乱数と呼ぶ）を*T*個発生させ、 $D_{Li}$ を式(4.23)のごとく決定する。

$$D_{Li} = L - \lceil x \rceil \quad (i=1,2,\dots,T) \quad (4.23)$$

但し、*x*は指数乱数であり、 $\lceil x \rceil$  は*x*に等しい整数あるいはより大きな整数の中の最小な整数である。

次にレベル*l*のブロックの総数*K<sub>l</sub>*を算出して、レベル*l*の*Bid*の組*B<sub>l</sub>*を式(4.24)のごとく定義する。

$$B_l \equiv \{ A+1, A+2, \dots, A+K_l \} \quad (4.24)$$

但し、

$$A = \begin{cases} 0 & (l=0 \text{ のとき}) \\ \sum_{i=0}^{l-1} K_i & (1 \leq l \leq L \text{ のとき}) \end{cases}$$

ブロック間のポインタチェイニングをシミュレーションするために、次の仮定を設ける。

[仮定2] レベル*l*のブロックからチェインされるレベル*l+1*のブロックの個数は前者のブ



ロックの各々について一様分布する。

逆に考えるとレベル $l$ のブロックは等確率 $S_{l||l+1}$  (次の式(4.25)にて与えられる。)でレベル $l+1$ のブロックから選出されることになる。

$$S_{l||l+1} = 1/K_l \quad (l=0,1,\dots,L-1) \quad (4.25)$$

そこで一様分布する乱数 (以後、一様乱数と呼ぶ。) を発生させ、レベル $l$ のブロックのBid ( $B_0$ とする) を式(4.26)のごとく決定する。

$$B_0 = B_{l,1} + n \quad (l=0,1,\dots,L-1) \quad (4.26)$$

但し、 $B_{l,1}$ は $B_l$ の第1要素であり、 $n$ は閉区間 $[0, K_l - 1]$ に一様分布する整数である。

## (2) プログラムファイル生成

プログラムファイルをシミュレーション生成するために、まず、プログラムの件数 $N$ と1プログラムのデータブロックの最大個数 $\alpha$ を外部から指定する。次に各プログラムが有するデータブロックの個数 $\beta$ を決定するために、次の仮定を設ける。

[仮定3] プログラムの容量は各々のプログラムについて一様分布する。

そこで、閉区間 $[1, \alpha]$ に一様分布する整数を発生させることにより、 $\beta$ を決定する。

プログラムディレクトリブロックの個数を $P_d$ 、プログラム $r$ の有するデータブロックの総数を $M_r$ とすると、そのデータブロックのBidの組 $C_r$ は式(4.27)のごとく定義される。

$$C_r \equiv \{F+1, F+2, \dots, F+M_r\} \quad (4.27)$$

但し、

$$F = \begin{cases} P_d & (r = 1 \text{ のとき}) \\ P_d + \sum_{i=1}^{r-1} M_i & (2 \leq r \leq N \text{ のとき}) \end{cases}$$

プログラム $r$ のブロック間のポインタチェイニングはディレクトリブロックから始まり

$C_{r,1}, C_{r,2}, \dots, C_{r,M_r}$ の順に行われる。但し、 $C_{r,i}$ は $C_r$ の第 $i$ 番目の要素とする。

### (3) ビットマップ生成

ビットマップのブロックの個数および1ビットマップが管理するブロック数は外部から指定する。

### グローバルファイルのアクセス

アクセスの単位はブロックであり、アクセス要求はレベルとBidが指定されることによりシミュレーションし、要求内容は

- 参照
- 生成
- 更新
- 削除

の4種類である。

アクセスレベル $l_a$ を決定するために次の仮定を設ける。

[仮定4] 変数のデータがアクセスされる頻度は、上位レベル、中間レベル、下位レベルの順に高くなる。

そこで指数乱数 $w$ を発生させ、 $l_a$ を式(4.28)のごとく決定する。

$$l_a = L_c - \lceil w \rceil \quad (4.28)$$

但し、 $L_c$ は生成済レベルの最大値とする。参照、更新および削除のときは一つの $l_a$ だけでよいが、生成のときは既存変数の下位に新変数が生成されることもあるので、 $l_a$ のほかに次のようなレベル $l_b$ が必要となる。すなわち、新たに生成する変数のレベルが $l_b$ であり、その変数に対してレベル $l_a$ までの変数は既に生成されているとする。

$l_b$ を決定するために次の仮定を設ける。

[仮定5] 既存変数の下位に新変数が生成されるとき、後者のレベルは前者のレベルに近いことが多い。

そこで、指数乱数 $y$ を発生させ、 $l_b$ を式(4.29)のごとく決定する。

$$l_b = l_a + |y| \quad (4.29)$$

ところで以上のレベルはフルシンタックス法を用いた場合であるが、ネイキッドレファレンス法を用いたアクセスもシミュレーションし、直前のアクセスレベルは参照、更新および削除のときは $l_a$ であり、生成のときは $l_b$ である。

アクセスするブロックのBid ( $B_a$ とする) を決定するために、次の仮定を設ける。

[仮定6] 同レベルにあるブロックへのアクセス要求は各ブロックに対して一様である。

すなわち、ブロック $B_a$ はレベル $l_a$ にあるブロック群から等確率 $H$  ( $H$ は式(4.30)にて与えられる) で選出される。

$$H = 1/K_{l_a} \quad (l_a=0,1,\dots,L_c) \quad (4.30)$$

但し、ブロック $B_a$ は、参照、更新および削除のときはアクセスのターゲットであるが、生成のときは追加生成が始まるブロックとなる。

そこで、一様乱数を発生させ、 $B_a$ を式(4.31)のごとく決定する。

$$B_a = B_{l_a,1} + m \quad (l_a=0,1,\dots,L_c-1) \quad (4.31)$$

但し、 $m$ は閉区間 $[0, K_{l_a}-1]$ に一様分布する整数である。

又、新たに生成されるブロックのBid ( $B_p$ とする) は次の式(4.32)のごとく決定する。

$$B_p = B_{L_c, K_{L_c} + 1} \quad (4.32)$$

なお、アクセス要求の内容は先述した4種であるが、その発生比率は外部から指定する。

### プログラムファイルのアクセス

アクセスの単位はファイルであり、アクセス要求はプログラムの名前、すなわち識別番号Pidを指定することによりシミュレーションし、要求内容は

●ロード

●登録

●削除

の3種類である。

ロードするプログラムのPid ( $P_L$ とする) を決定するために次の仮定を設ける。

[仮定7]プログラムのロード要求は各プログラムに対して一様である。

すなわち、登録済のプログラム件数を $N_C$ とすると、プログラム $P_L$ は $N_C$ 個のプログラムの中から等確率 $1/N_C$ で選出される。そこで閉区間 $[1, N_C]$ に一様分布する整数を発生させ、 $P_L$ を決定する。

登録は新プログラムの生成と既存プログラムの更新の二通りがシミュレーションされる。

まず、登録可能なプログラムの最大件数 $N_m$ が外部から指定され、次に登録するPid ( $P_C$ とする) を決定するために閉区間 $[1, N_m]$ に一様分布する整数を発生させる。

この結果、次の二通りに分かれる。

1.  $P_C \leq N_C$ のときは既存プログラムの更新であり、
2.  $P_C > N_C$ のときは新プログラムの生成である。

又、プログラム $P_C$ の容量は前節の仮定3の下で決定する。

削除するプログラムのPidはロードの場合と同様に、閉区間 $[1, N_C]$ に一様分布する整数を発生させることにより決定する。

なお、アクセス要求の内容は先述した3種であるが、その発生比率は外部から指定する。

#### 4.3.5 シミュレーションの結果

データベースの初期負荷は表4.12に示す値を用い、グローバルディレクトリブロック数 $G_d$ は可変とする。

アクセス要求は表4.13に示す比率で発生させた。

表4.12 データベースの初期負荷

内容	値
グローバルディレクトリブロック数	可変(1~4)
グローバルデータブロック総数	542
グローバルファイルのレベル	7
プログラムの件数	10
プログラムディレクトリブロック数	1
プログラムデータブロック総数	294
ビットマップブロック数	1
1ビットマップブロックの管理ブロック数	1,992

表4.13 アクセス要求の発生比率

アクセスの内容	比率
グローバルファイルのアクセス	0.9
参照	0.27
生成	0.27
更新	0.27
削除	0.09
フルシンタックスのアクセス	0.72
ネイキッドシンタックスのアクセス	0.18
プログラムファイルのアクセス	0.1
ロード	0.05
登録	0.025
削除	0.025

LRU法、BPL法およびBPL\*法により置換されるバッファを用いてMUMPSデータベースをアクセスしたときのバッファ不在頻度が、種々のバッファ面数（以後、 $B_s$ と呼ぶ）について調査された結果が図4.25から図4.28である。但し、 $G_0$ のみが1から4まで変化し、残りのパラメータは固定され、又、 $B_s$ は3から10まで変化させた（なお、図中のプロットは3種類であるが、一つの $B_s$ に対してプロットが3個表示されていない場合がある。\*が表示されていないときは+と同位置、+が表示されていないときは・と同位置である）。

図4.25から図4.28から次のことが考察される。

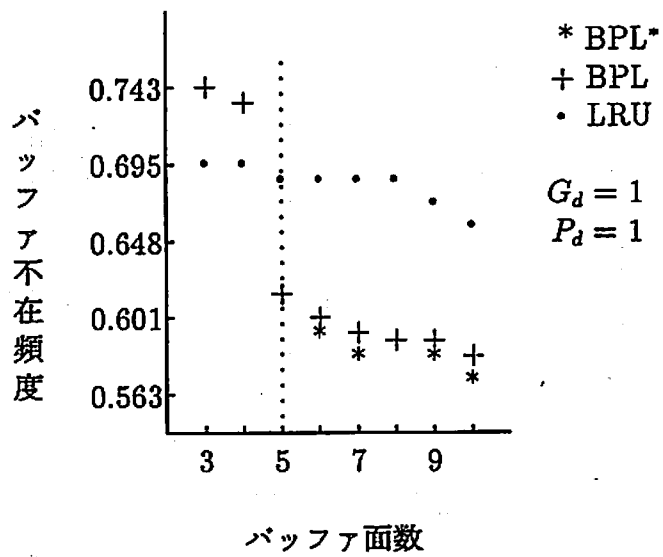


図4.25 バッファ不在頻度—その1.

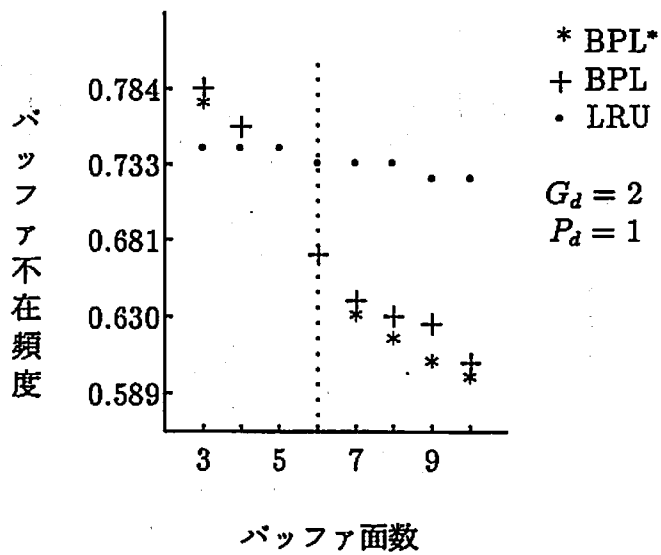


図4.26 バッファ不在頻度—その2.

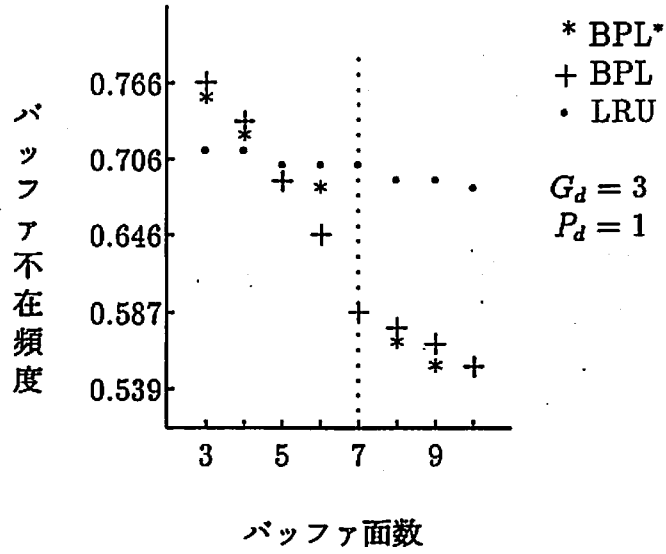


図4.27 バッファ不在頻度—その3.

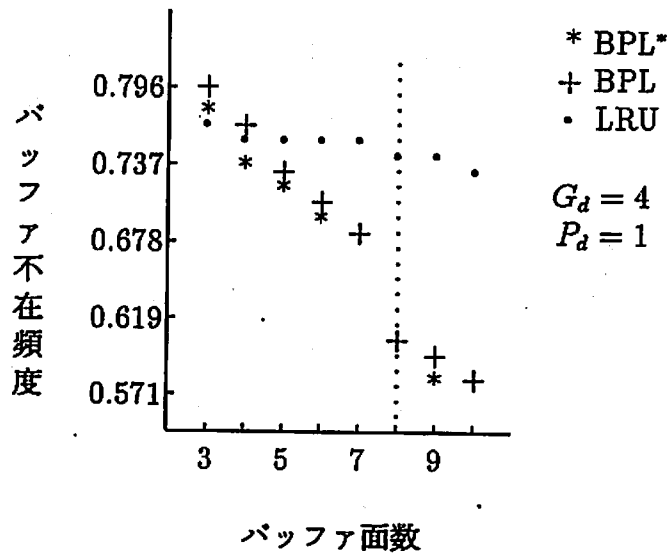


図4.28 バッファ不在頻度—その4.

1.  $B_S$ が5以上であるときはBPL法やBPL\*法がLRUよりも低いバッファ不在頻度を示すが、
2.  $B_S$ が4以下であるときはLRU法が最も低いバッファ不在頻度を示している。

これは、

- グローバルディレクトリ用に1面
- プログラムディレクトリ用に1面
- ビットマップ用に1面
- カレントレベルの先頭ブロック用に1面
- 最新アクセス用に1面

の計5面のバッファが確保され、*bpl*の効果が見れたからである。又、ビットマップブロック数を $B_m$ とすると、 $B_S$ が次の式(4.33)を満足するとき(図中の点線に対応する)。

$$B_S = G_D + P_D + B_m + 2 \quad (4.33)$$

BPL法やBPL\*法がLRU法に比して急激に減少し、*bpl*の効果が見れている。

次に $P_D = 2$ とし、 $G_D$ を1から3まで変化させたときの結果が図4.29から図4.31であり、先の結果と同様の事柄が考察される。

BPL法とBPL\*法を比較すると、後者のバッファ不在頻度がより低いことが分かる。ここに、*Bdpt*を導入した効果が見れている。

1. 5～6面以上のバッファが確保できるときはBPL\*法が最適であるが、
2. 逆のときはLRU法が最適である、

と結論される。



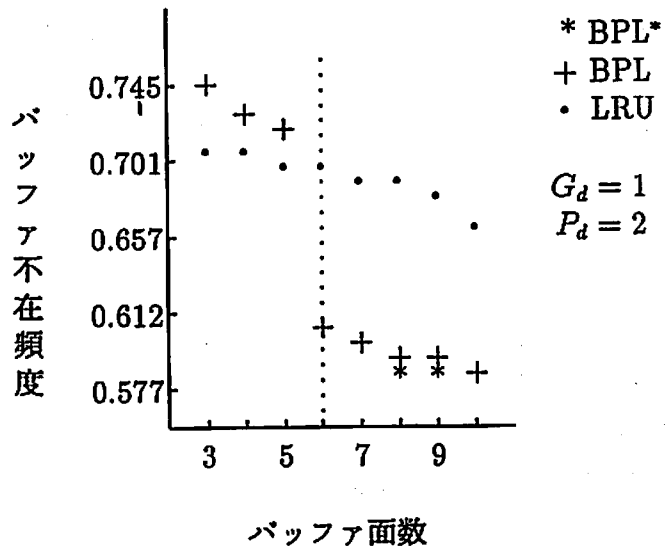


図4.29 バッファ不在頻度—その5.

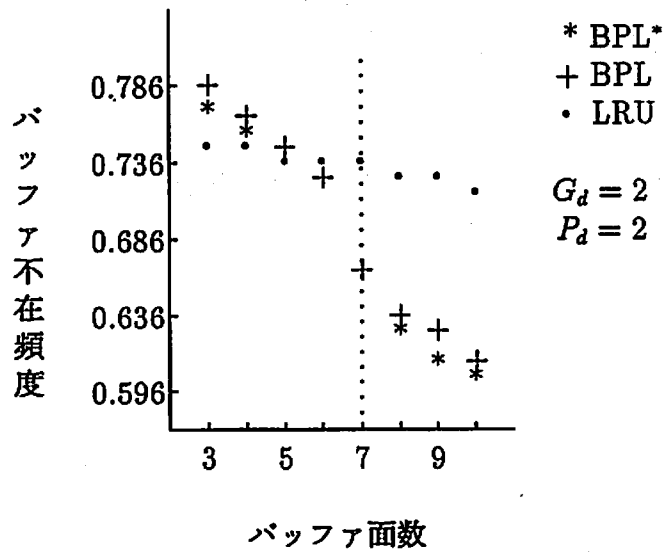


図4.30 バッファ不在頻度—その6.

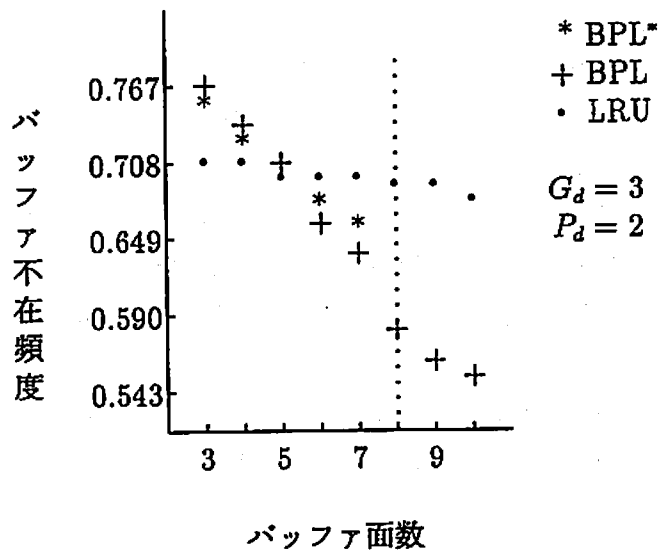


図4.31 バッファ不在頻度—その7.

## 4.4 結言

集中計算環境におけるデータベースシステムの性能評価に関する2件の研究課題に対する成果を総括すると、先ず論理設計、物理設計、スキーマ性能評価の各設計フェーズを統合化したデータベース設計自動化・性能予測ツールを試作し、データベースのアクセス費用と性能をデータベース構築の事前に予測と改善を可能とした。また木構造データベースの各層に優先順位を考慮したバッファ管理方式を提案し、データベースアクセス時のバッファ不在頻度を最小化することを計算機シミュレーションにより検証した。

以下に、それぞれの研究課題に対する成果と今後の研究課題をまとめる。

先ず、データベース自動設計の研究の一環として開発した設計評価支援ツールDMSDBDAについて記述した。本ツールを用いてデータベース設計を行う際、次の方針が標準と考えられる。

- メンバのロケーションモードはVIAセットに変更する。
- キー付きアクセスのときはアクセス開始グループのロケーションモードはINDEXEDに変更する。
- キーなしアクセスのときはアクセス開始グループにOWNER AREAを指定する。
- メンバからオーナーへのアクセスのとき、そのセットをOWNERポインタを追加する。
  - ・問合せの実行時間の最小化が第1の評価基準であるときはバッファ容量を増やす。
  - ・データベースの記憶費用の最小化が第1の評価基準のときはエリア容量をできる限り減らす。

又、データベース管理者が新規のデータベースを開発する際、本ツールを用いるとデータベースを実際に開発する前にデータベースの性能や費用を見積ることができる。

一方、既存のデータベースに本ツールを適用するならば、そのデータベースの性能改善の手助けにもなる。

DMSDBDAはFORTRAN言語で記述されており、そのプログラム数は約6,000行、その容量は約55K語である。本ツールの対象とするデータベースの規模は、データ項目タイプが最大500件、グループタイプが最大50件、セットタイプが最大50件、エリアが最大5件である。

今後の研究課題は、先ず集中計算環境において情報構造設計機能を追加し、更に論理および物理設計のためにモデルを拡張し（例えば、SORTEDセットオーダと重複CALCキー項目と可変長データ項目のモデル化と評価など）、そしてデータベースの保守費用の見積り方式の追加などがある。次に、分散計算環境および移動体計算環境におけるデータベース設計自動化支援・性能評価へと拡張することが更なる将来の研究課題であると考えられる。

次に、MUMPSデータベースに適したバッファ置換アルゴリズムを、計算機シミュレーションにより、バッファ面数ごとに調査した。MUMPSデータベース用バッファ置換ア

ルゴリズムの性能評価に関する報告は本研究のほかにはまだ見られないが、データベースの設計上、その評価結果は重要な情報である。

また、今回の評価結果の具体的な利用成果としては、BPL\*法が医療情報システム<<MELCOM MUMPS>>に採用されて、商品化されたことが挙げられる[Dequ79]。

本研究ではシミュレーション生成したデータベースの容量が約0.5MByteであり、且つ、データベース利用者はただひとりである場合を取り上げたが、データベースの容量が増加し大容量のデータベースになったときや多数の利用者がデータベースを同時にアクセスしたときのバッファ置換アルゴリズムの評価が今後の課題であると思われる。更に、実際のMUMPSデータベースシステムにおけるバッファ不在頻度を実測して、本研究のシミュレーション結果と照合することも今後の課題として残されている。

なお、本研究のシミュレーションプログラムはFORTRAN言語で記述し、そのステップ数は約1,500行、サイズは約34K語である。又、使用した計算機はMELCOM-COSMO700であり、本研究のシミュレーション結果(図4.25～図4.31)は図1枚当り30～40分のCPU時間で求められた。

今後の研究課題をまとめると、先ず木構造データベースアクセスモデルの精査、すなわち上述したような大容量データベースに対する性能評価と複数利用者の同時並行アクセスなどが考えられる。そして、次には移動体環境におけるデータベースバッファ管理方式への拡張、例えばモバイルデータベースのキャッシュ管理方式なども考えられる。

## 第5章 移動体計算環境におけるデータベースシステムの性能評価

### 5.1 緒言

近年携帯端末の低価格化や高性能化、高機能化により、その普及が急速に進んできた。

この結果、携帯端末のデータ通信機能を用いた移動計算環境の発展に伴い、この環境でのデータベースアクセス要求も同時に高まってきた。しかし、移動計算環境でのデータベース技術に関する研究はまだ緒についたばかりである。例えば携帯端末の利用者がある無線通信セルから別の通信セルへ移動しながら共用データベースをアクセスする際、どの通信セルの中にいるのか、あるいはどの通信セルからどの通信セルへ移動しているのかを意識することなく効率良くデータベースアクセスすることが困難である問題があった。

本研究では、先ず移動計算環境において携帯端末利用者が複数の通信セル間を移動しながら共用データベースをアクセスする時、移動先の通信セルでデータベースアクセス時のページ不在の発生する頻度を最小化するためのデータベースキャッシュ管理方式を数種類新たに提案する。最後に、計算機シミュレーションにより本研究にて提案したデータベースキャッシュ管理方式の有効性を検証していく。

## 5.2 システムの概念とモデル化

### 5.2.1 適用業務システム

現在具体的な適用業務システムの一つとして、第3章にて記述したデータマイニング・クライアント/サーバシステムを開発している。このシステムは広域分散データベースシステムをアクセスすることにより、大容量データベースに潜在する新しい知識や規則性を発見することを支援するものである。具体的には、このデータマイニング・クライアント・サーバシステムは、分散データベース上に存在する大量データ（例えば、テキストデータ、数値データ、画像データなど）を出来る限り高速にアクセスするので、そのデータベースシステムの振る舞いをシミュレーションすることにより、事前にデータベースアクセス性能を評価する必要がある。

また、別の適用業務システムとして、地震や火災などの非常緊急事態に対応するための防災情報システムも現在研究開発している。このシステムは、中央の防災本部の情報システム（ホストデータベース）、複数の現地対策本部の情報システム（現地単位のデータベース）、そして多数の現場作業員（無線携帯型コンピュータを所持。）から構成されており、防災本部と現地対策本部は有線あるいは無線の通信回線で接続、一方現地対策本部と現場作業員は無線通信回線で接続されている。従って、現場作業員は移動体計算の環境で現地対策本部のデータベース、さらには防災本部のデータベースをアクセスすることになる。

### 5.2.2 システムのモデル化

本研究が研究対象とする移動体計算環境におけるデータベースシステムは図5.1に示すようなモデルである。

全体のシステムは複数のセルの集合体であり、かつ各セル間には共通部分はないものとする。また、各セルには複数のモバイルユニット（例えば、無線携帯型コンピュータなど）と1台のモバイルサポートサーバ（例えば、ワークステーションやパソコン）が存在している。そして、各モバイルサポートサーバにはセルデータベースが構築されている。更に、このモバイルサポートサーバはシステムのある中心にあるホストコンピュータにネットワーク経由で接続されている。そして、そのホストコンピュータにはホストデータベースが構築されており、その部分集合がセルデータベースに格納されている。

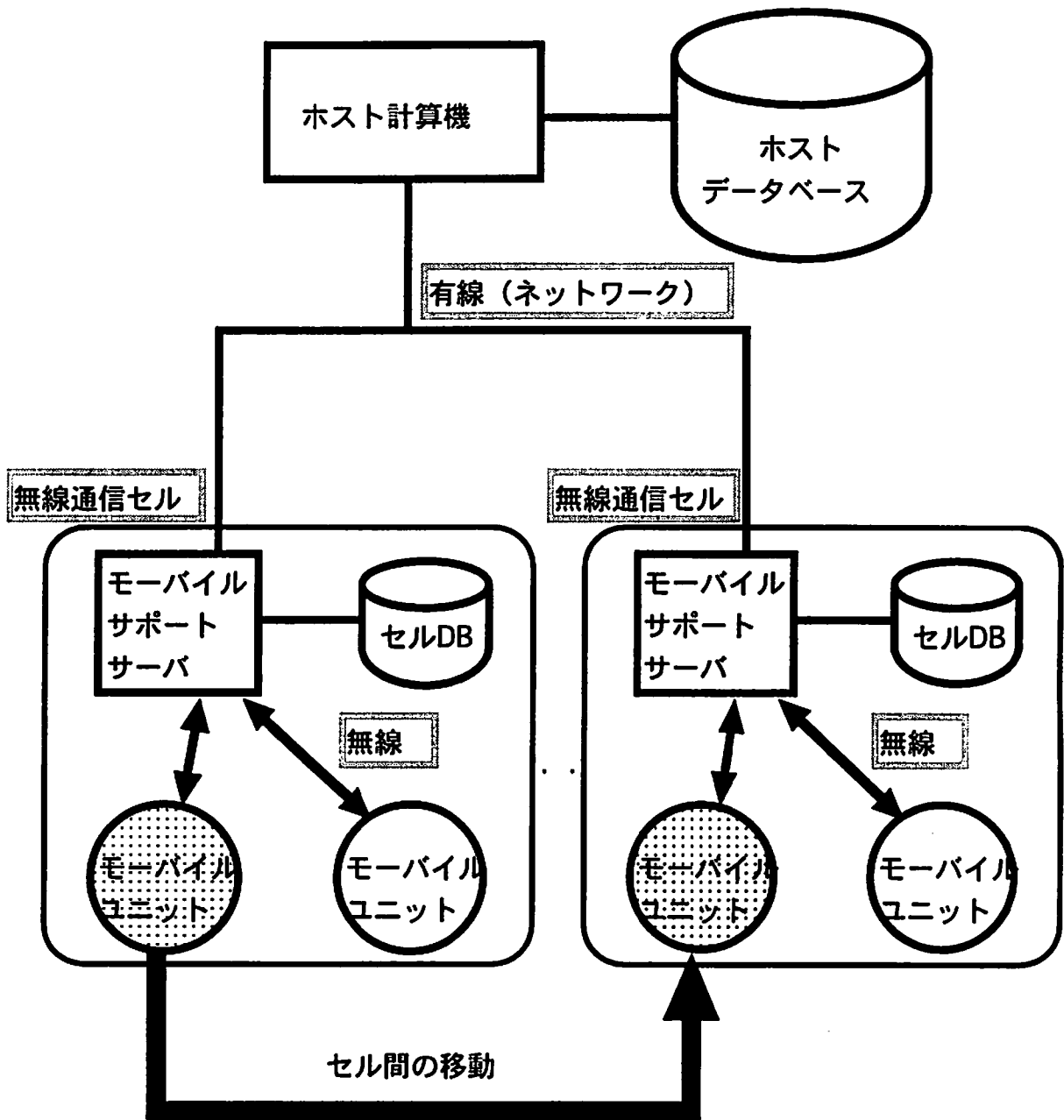


図5.1 システムモデル

従って、モバイルユニットの利用者が複数のセル間を移動しながら無線を用いてモバイルサポートサーバ経由でホストデータベースをアクセスすることができる。このとき、ホストデータベースと移動先の複数のセルデータベースの間でデータをダウンロードしたり、あるいはアップロードしたりするためのデータベースキャッシュが図5.2に示したように行なわれる。

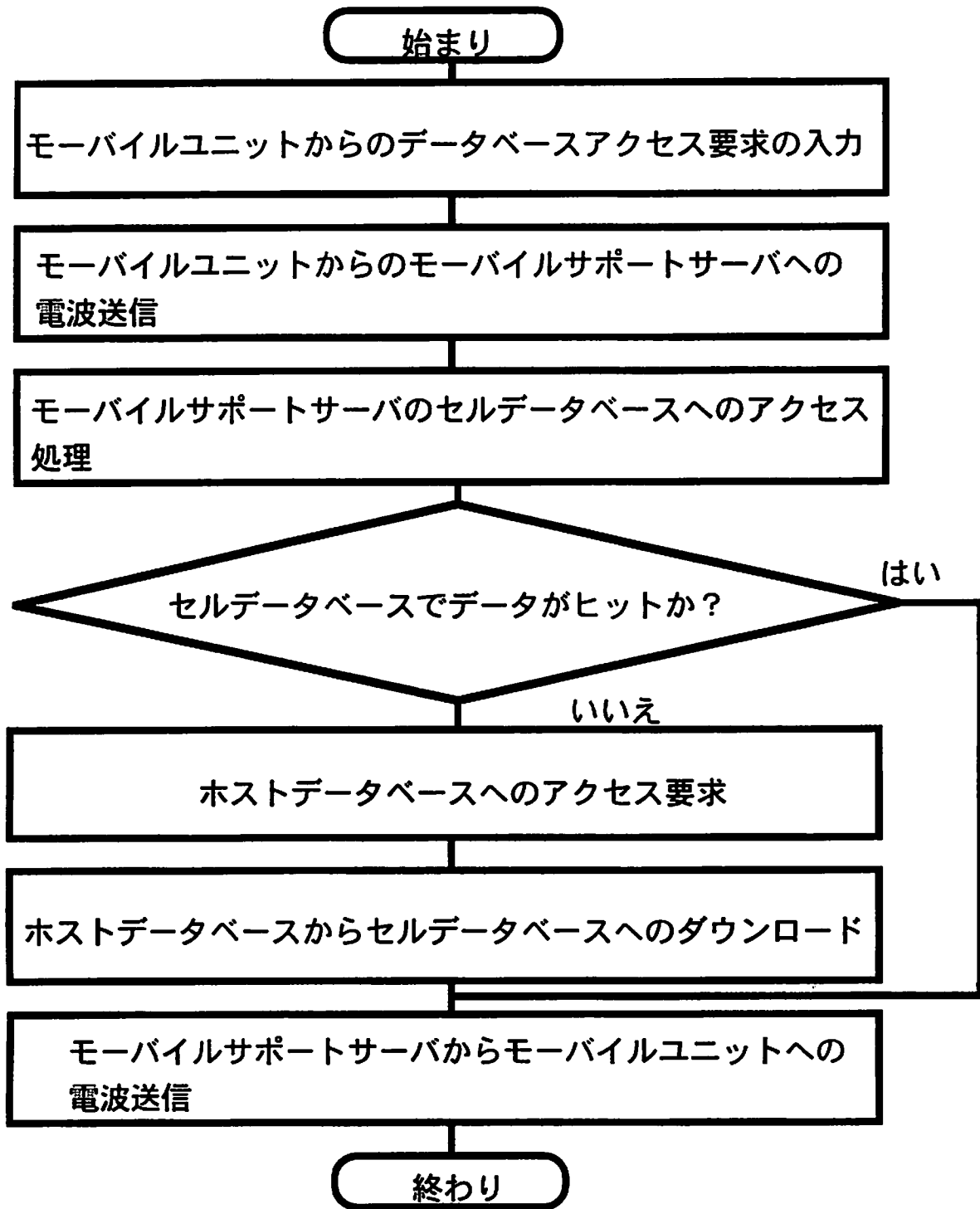


図5.2 データベースアクセスの流れ



### 5.2.3 問題

本研究では、以下の問題を抽出した。

- モーバイルユニットが当初データベースアクセス要求を発したセルとは異なる別のセルに移動した場合、モーバイルサポートサーバとの無線接続が切断される問題。
- モーバイルユニットがホスト計算機のデータベースアクセスサービスを連続して享受できない問題。
- モーバイルユニットは移動先セルにて、再度そのセル内のモーバイルサポートサーバに接続し直さねばならない問題。

そこで、こうした問題を解決するために、本研究では時間制約の概念をデータベースシステムに導入し、かつモーバイルユニットを考慮した新しいデータベースキャッシュ管理方式を提案する。

### 5.2.4 時間制約

本節では先ず時間制約付きデータベースシステムの概念を議論していく。

本研究で提案する時間制約付きデータベースシステムとは、図5.3と図5.4に示すようにホストデータベースアクセス要求を発したモーバイルユニットがモーバイルサポートサーバから見て自セル内に滞在しているか否かを判断するのに、そのモーバイルサポートサーバから当該モーバイルユニットへ無線送信したときの返信がある一定時間内にあったかどうかを調べる機能を有しているデータベースである。言い替えば、ある制約時間内に応答が無かった場合は、当該モーバイルユニットは別のセルに移動済みであると判断し、逆に応答が時間制約内にあったときは依然として自セル内に滞在していると判断する方式である。

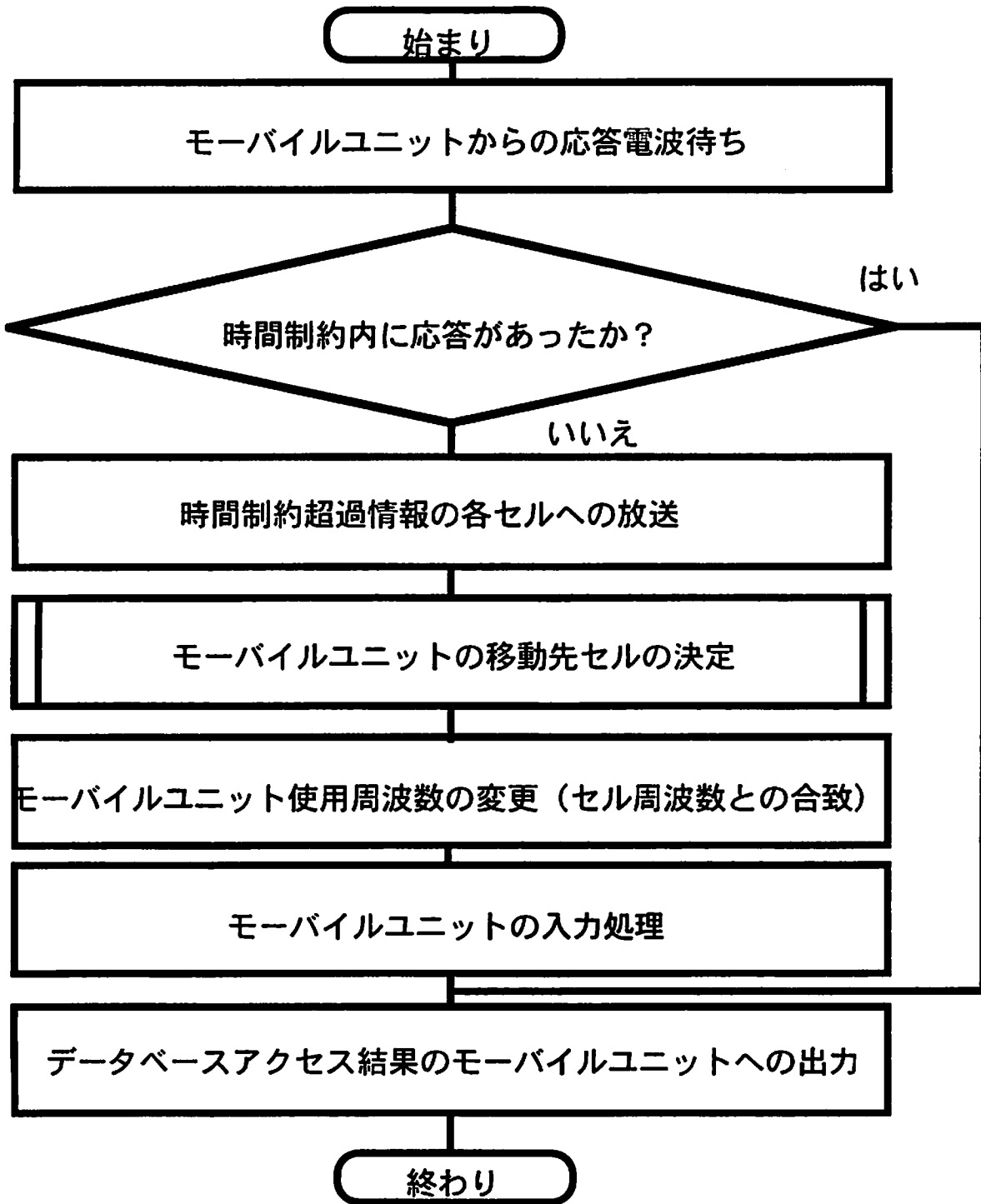


図5.3 出発セルにおける時間制約処理の流れ

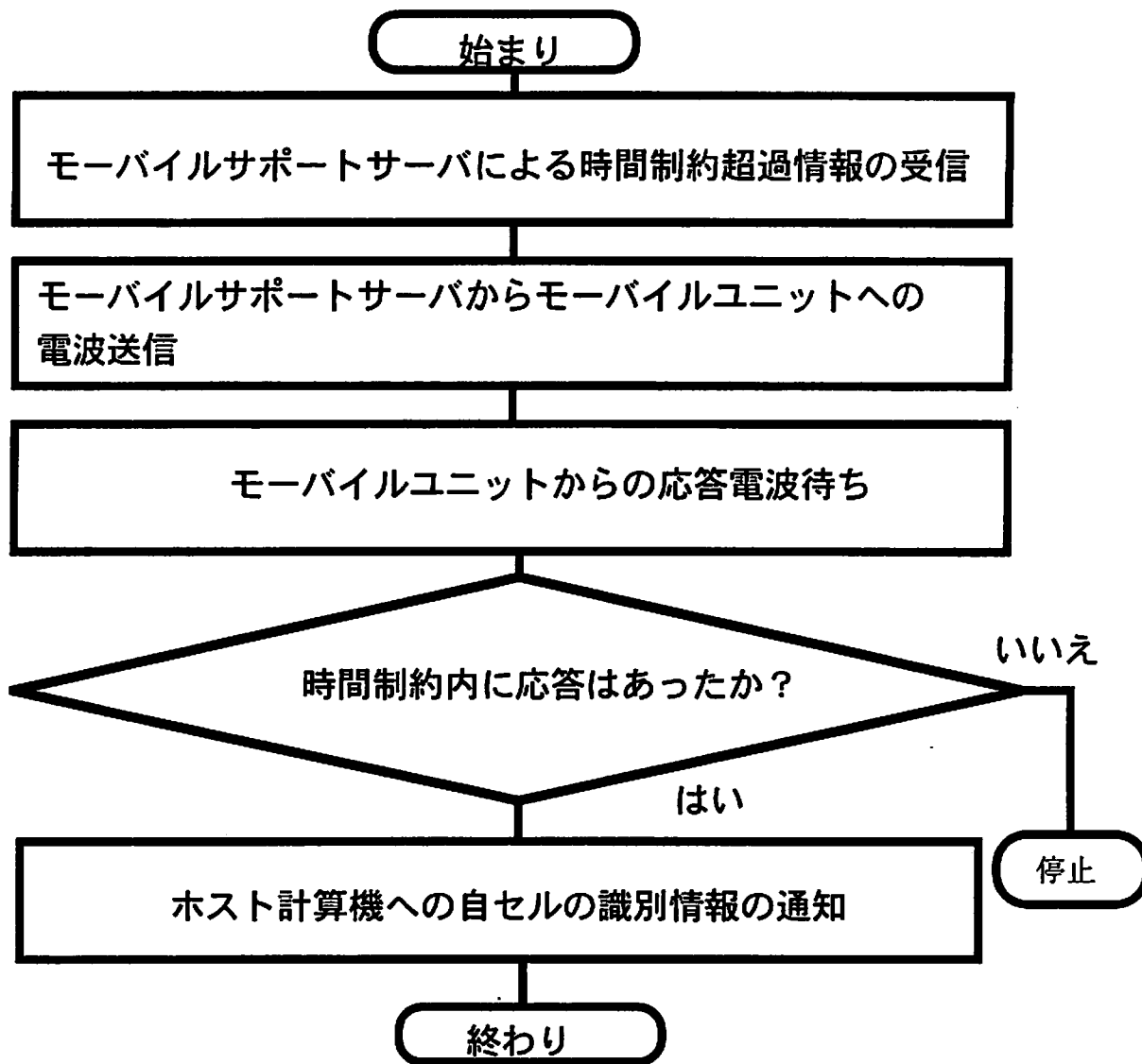


図5.4 目的地セルにおける時間制約処理の流れ

### 5.3 データベース・キャッシュ管理方式

本研究で提案するデータベースキャッシュ管理方式は、以下のような手続きである。

#### 手続き

- (1) モバイルユニットからのアクセス要求を受けた第1のセル内の第1のモバイルサポートサーバが、接続されている第1のセルデータベースにアクセス要求されたデータが存在するか否かを調べ、
- (2) データが存在しない場合、そのアクセス要求をホスト計算機へ通知し、
- (3) ホスト計算機はホストデータベースからのアクセス結果を第1のセルデータベース及び第1以外の第2のセル内の第2のモバイルサポートサーバに接続された第2のセルデータベースにも転送し、
- (4) 第1のモバイルサポートサーバは当該モバイルユニットが第1のセルに滞在しているか否かを判定し、
- (5) 当該モバイルユニットが第1のセル内に滞在していない場合、第1のモバイルサポートサーバは当該モバイルユニットが第1のセル内に滞在していないことを第2のモバイルサポートサーバに通知し、
- (6) 第2のモバイルサポートサーバは当該モバイルユニットが第2のセル内に滞在しているか否かを判定し、
- (7) その滞在を確認してから第2のセルデータベースに転送されたアクセス結果を当該モバイルユニットに転送する。

ここで、上記の第2のセルは、図5.5から図5.10に示した6種類の代替方式を有する。

●全セルキャッシュ法： 図5.5に示すように、ホスト計算機により管理されている全てのセルをキャッシュする。

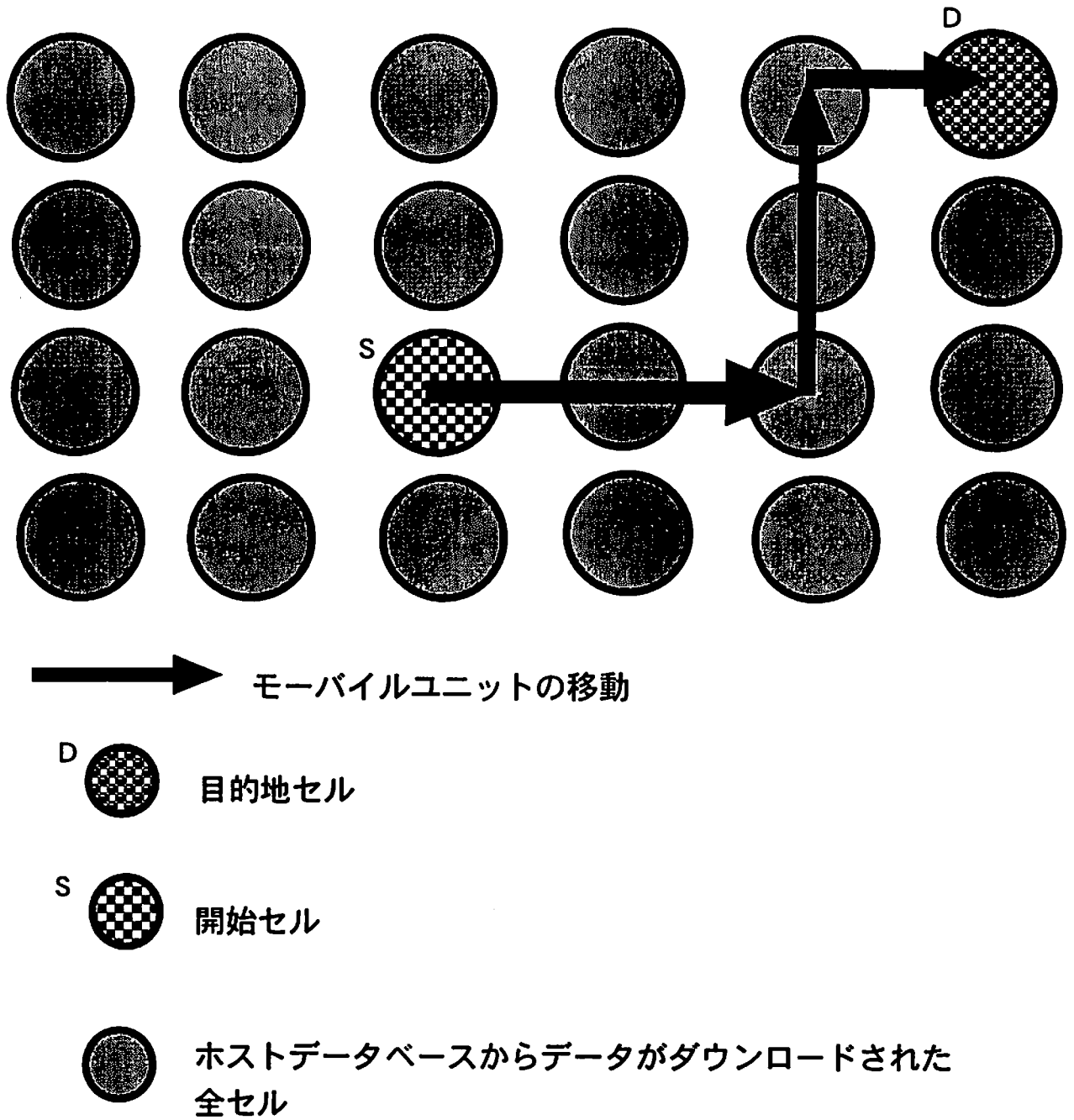


図5.5 全セルキャッシュ法

●隣接セルキャッシュ法： 図5.6に示すように、第1のセルに隣接しているセルをキャッシュする。

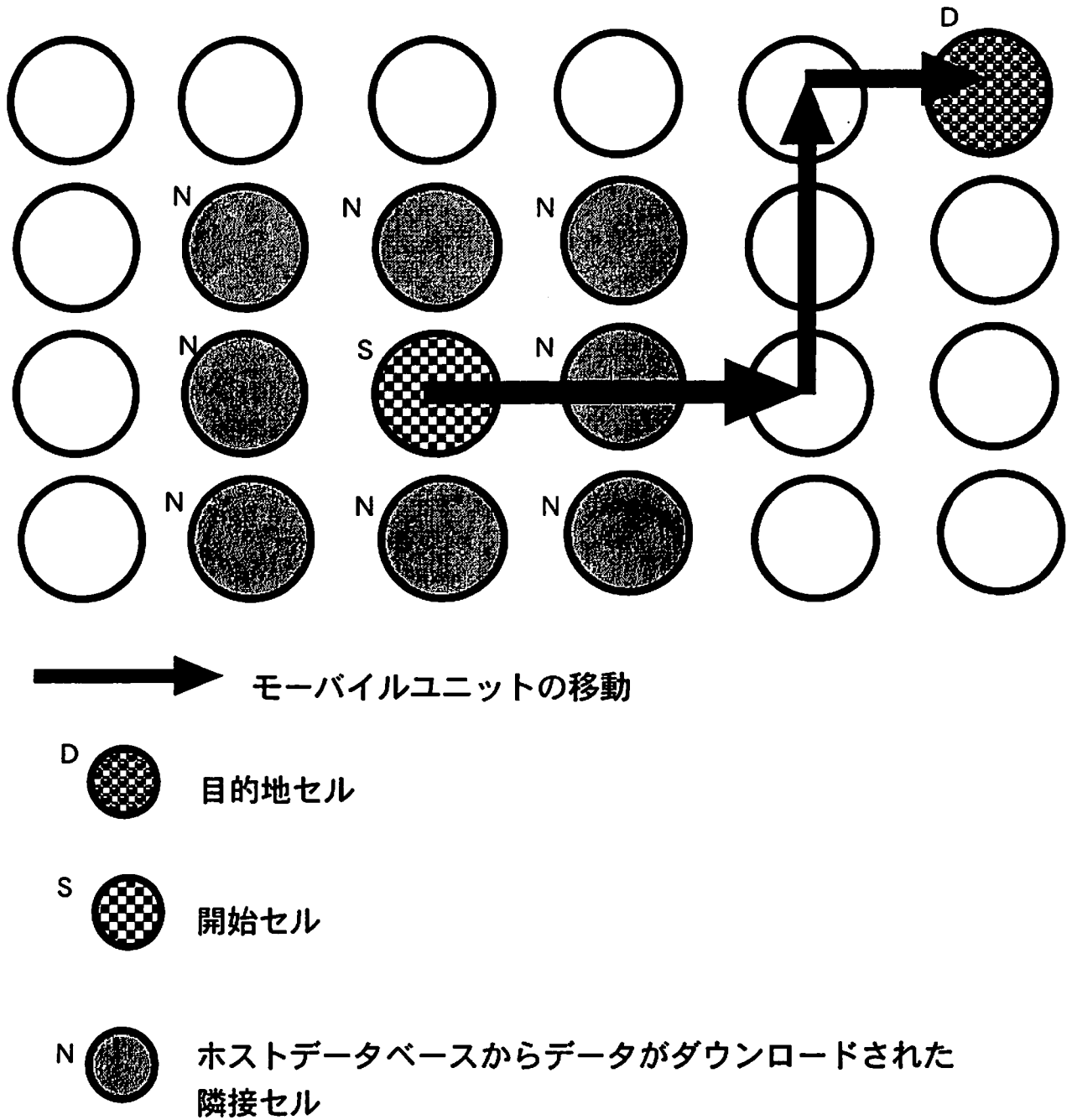


図5.6 隣接セルキャッシュ法

● 1セル置きキャッシュ法： 図5.7に示すように第1のセルから1個置きセルを繰り返してキャッシュする。

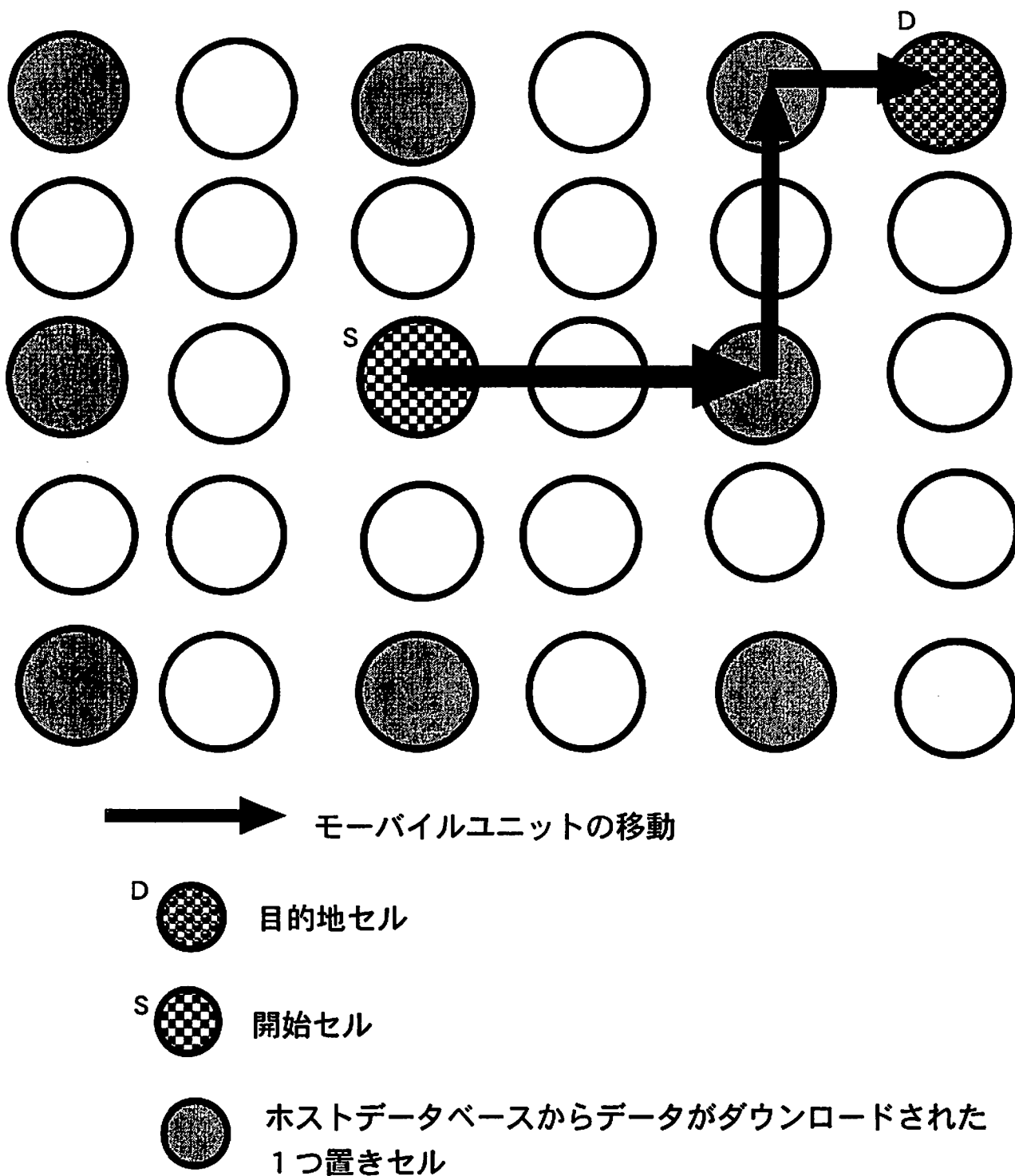


図5.7 1セル置きキャッシュ法

● 2セル置きキャッシュ法： 図5.8に示すように第1のセルから2個置きセルを繰り返しキャッシュする。

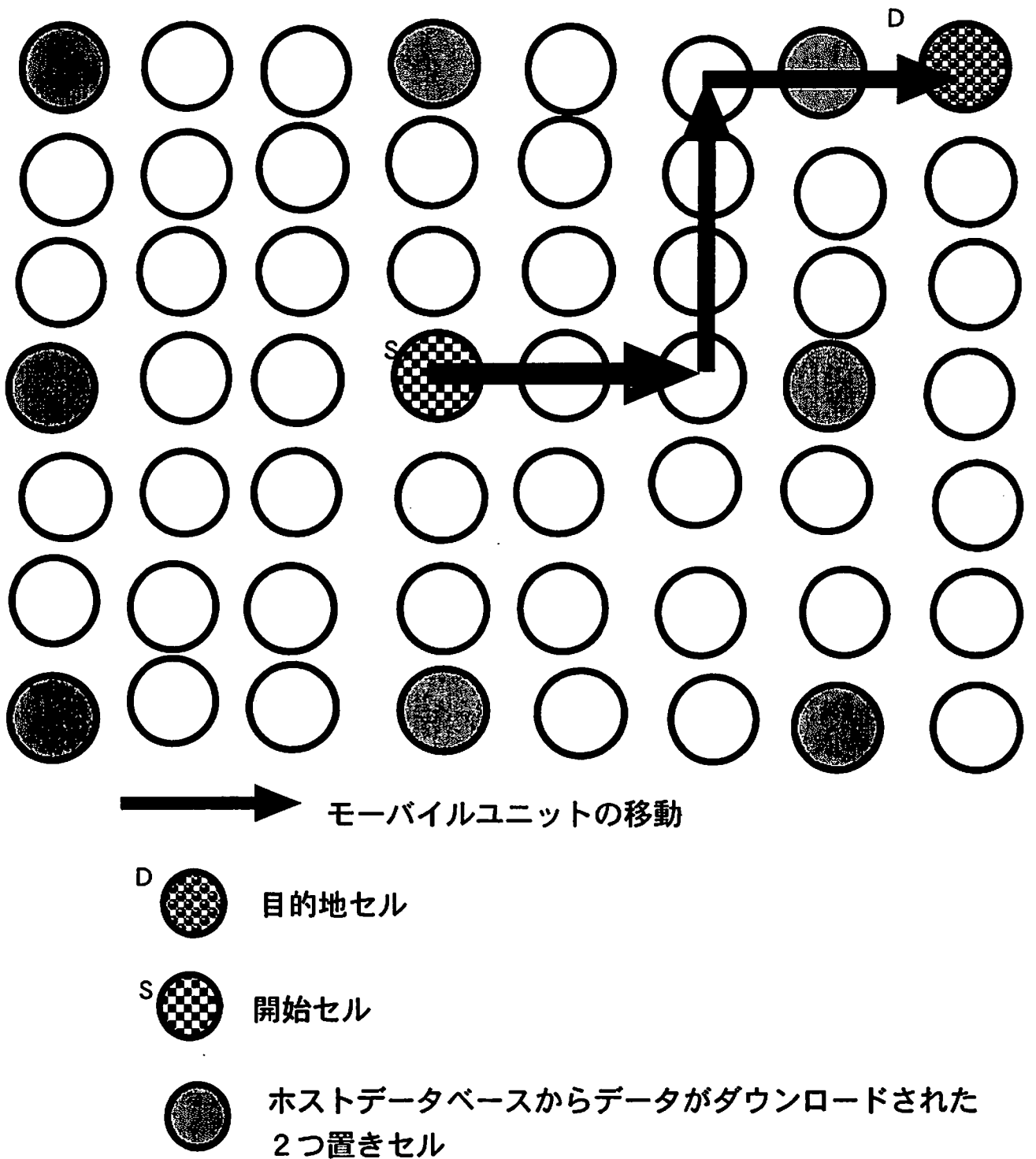


図5.8 2セル置きキャッシュ法



● 3セル置きキャッシュ法： 図5.9に示すように第1のセルから3個置きセルを繰り返しキャッシュする。

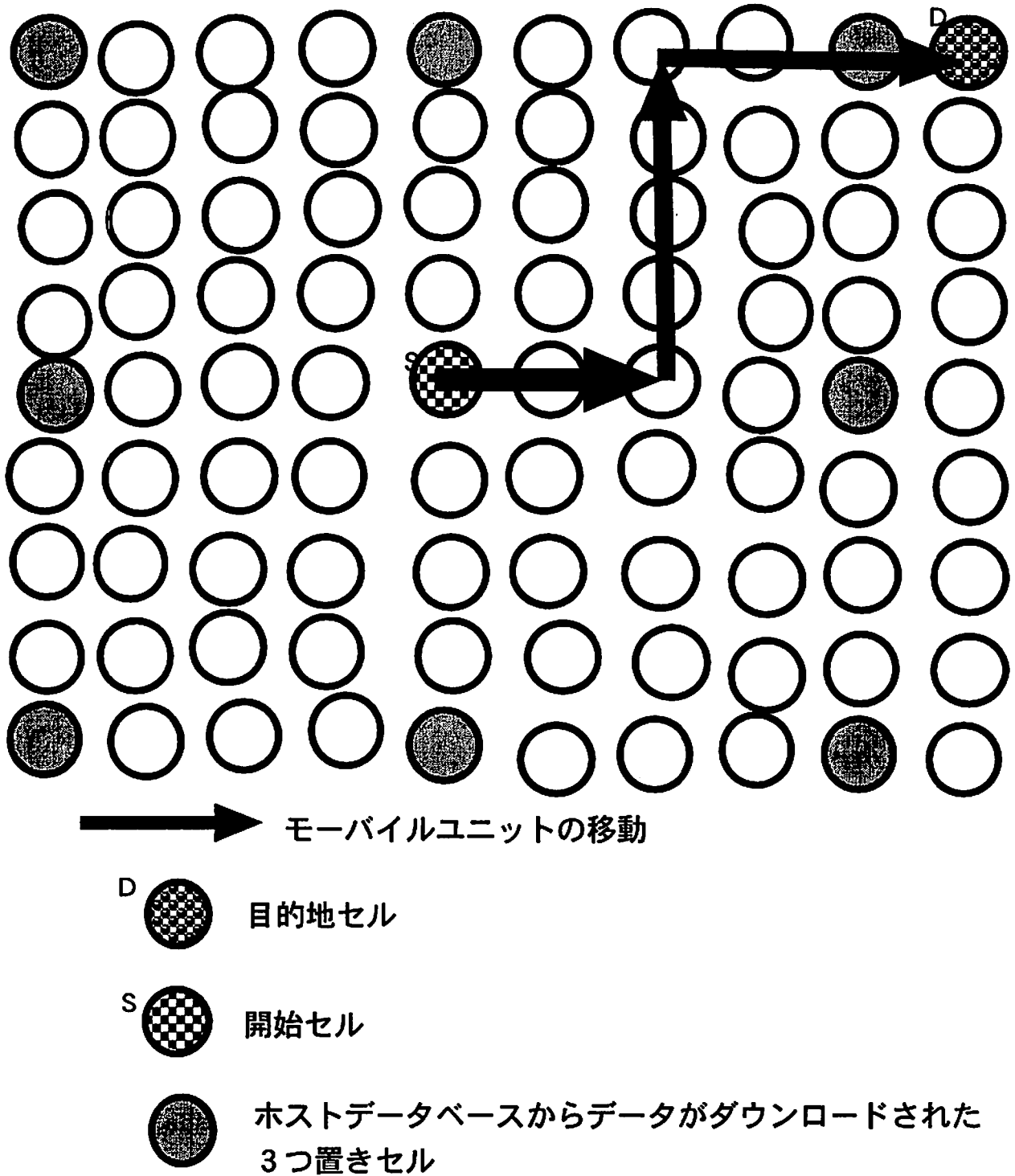


図5.9 3セル置きキャッシュ法

●自セルキャッシュ法： 図5.10に示すように第1のセルのみをキャッシュする。

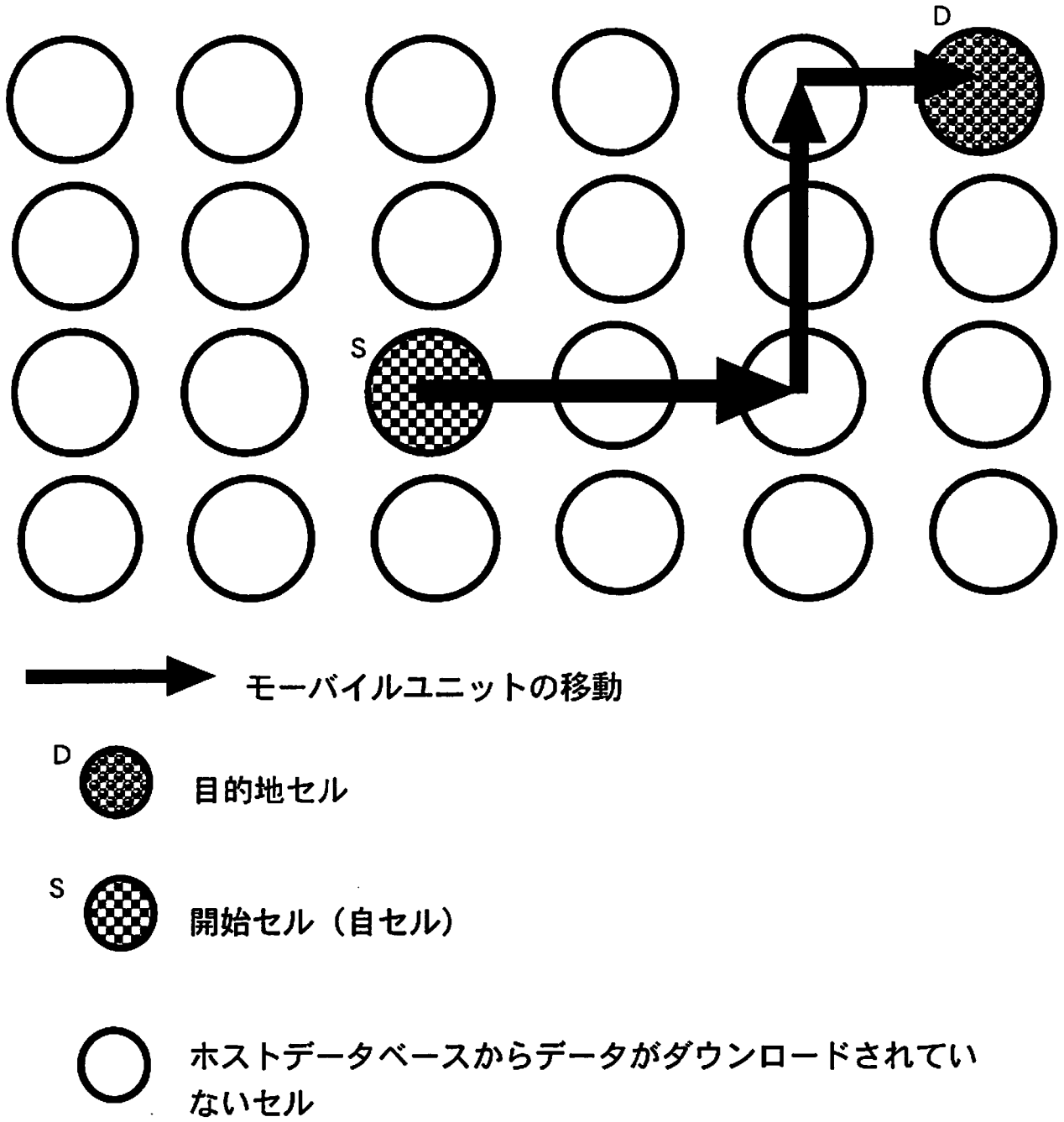


図5.10 自セルキャッシュ法

## 5.4 計算機シミュレーションによる性能評価

前節に記述したデータベースキャッシュ管理方式の性能を比較評価するために、モバイルユニットが複数セル間を移動しながら、無線によりモバイルサポートサーバ経由でホストデータベースをアクセスする状態を計算機シミュレーションした。

### 5.4.1 測定項目

このシミュレーションによる測定項目は、

- ホストデータベースアクセス経過時間の平均値
- セルデータベースの平均キャッシュヒット率

の2件である。

すなわち、前者のホストデータベースアクセス経過時間の平均値 $DB_t$ は、式(5.1)のごとく、ホストデータベースへのアクセス・シミュレーション1回当たりのアクセス経過時間 $DB_i$ の合計値 $\Sigma DB_i$  (ただし、 $i=1,2,\dots,loop$ )をそのシミュレーション回数 $loop$  (ただし、 $loop$ は次節を参照のこと。)で除した平均値である。

$$DB_t = \Sigma DB_i / loop \quad (i=1,2,\dots,loop) \quad (5.1)$$

ただし、 $DB_i$ は、

- (a)ホストデータベースアクセス入力要求のモバイルユニット処理時間、
  - (b)モバイルユニットからモバイルサポートサーバへの電波送信時間、
  - (c)モバイルサポートサーバとセルデータベースの処理時間、
  - (d)モバイルサポートサーバからホスト計算機までの有線通信時間、
  - (e)ホスト計算機とホストデータベースの処理時間、
  - (f)ホストデータベースからセルデータベースへの有線経由でのダウンロード時間 (含、ホストデータベースアクセス結果のデータ転送時間)、
  - (g)モバイルサポートサーバからモバイルユニットへの無線経由でのダウンロード時間 (含、ホストデータベースアクセス結果のデータ転送時間)、
  - (h)モバイルユニットの時間制約時間内の応答時間、
  - (i)モバイルユニット移動時間、
  - (j)ホストデータベースアクセス結果のモバイルユニット表示時間、
- などの和である。

一方、後者のセルデータベースの平均キャッシュヒット率 $CACHE_t$ は、式(5.2)のごとく、モバイルユニットがホストデータベースへアクセス要求を出した時、そのアクセス要求したホストデータベースの各ブロックと同一ブロックが当該モバイルユニットの存

在するセルデータベース  $CELLDB_j$  内に発見されたときの回数（すなわち、ヒット回数） $HIT_j$ と、 $CELLDB_j$ の各ブロックをアクセスした回数 $ACCESS_j$ を、当該モバイルユニットが移動した先のすべてのセルについて各々の合計値を算出して、除算した値である。

$$CACHE_t = \sum HIT_j / \sum ACCESS_j, \quad j \in M \quad (5.2)$$

ただし、 $M$  はモバイルユニットが移動するセルの集合である。

本研究で採用したシミュレーション方法は、乱数発生によるモンテカルロ法を用いている [Ishi 94]。また、このシミュレーションプログラムはC言語で記述し、総ステップ数は約10K行である。

#### 5.4.2 シミュレーション・モデル

図5.1.1に示したようなモデルにおいて、次のような統計的な性質を仮定する。

##### (1) 各計算機の処理時間

- ホスト計算機とホストデータベースの処理時間は、その和が平均 $T_h$ 秒の指数分布に従う乱数であると仮定する。
- モバイルサポートサーバとセルデータベースの処理時間は、その和が平均 $T_c$ 秒の指数分布に従う乱数であると仮定する。
- モバイルユニットの処理時間は、平均 $T_u$ 秒の指数分布に従う乱数であると仮定する。

##### (2) ホストデータベース格納

- ホストデータベースに格納するブロックの個数は、一様分布に従う乱数であると仮定する。

##### (3) ホストデータベースアクセスパターン

- ホストデータベースへのアクセスの種類は、一般的には、検索、挿入、更新、削除に分類できるが、本研究ではアクセス要求の発生頻度が最も高いと考えられる検索のみと仮定する。
- ホストデータベースへのアクセスの局所性は存在しない場合と存在する場合が考えられるが、本研究では前者の場合のみを扱う。

そして、アクセス目標のブロックが等確率 $1/B_h$ （但し $B_h$ は総ブロック数。）で決定されると仮定する。

##### (4) モバイルサポートサーバ動作

- あるセルにおいてモバイルサポートサーバが、そのセル内にモバイルユニットが滞在しているか否かを判定するためにそのモバイルユニットからの応答を待つ時間は、平均 $T_w$ 秒の指数分布に従う乱数であると仮定する。
- モバイルサポートサーバがモバイルユニットからの応答を待つ最大時間、すなわち時間制約時間は一定と仮定する。
- ホストデータベースとセルデータベースの間で行われるデータベースキャッシュにおけるブロック再配置法はLRU法と仮定する。

#### (5) モバイルユニット動作

- モバイルユニットが出発するセルは、等確率 $1/(N_x N_y)$  (但し、 $N_x$ は水平方向の通信セル総数、 $N_y$ は垂直方向の通信セル総数。) で決定されると仮定する。
- モバイルユニットが移動する先のセル (移動するセル個数の変化量) は、平均 $C_m$ 個の指数分布に従う乱数であると仮定する。
- モバイルユニットが移動していくセルのトポロジーは、一般的には不規則な2次元空間であるので、2次元のトポロジーを仮定する。
- セル間を移動するモバイルユニットは一般的には複数存在し、並行してホストデータベースをアクセスするが、本研究では上記のセルのトポロジーと同様にその第一ステップとしてシステム全体でただ1個であると仮定する。したがって、複数モバイルユニットのモデル化は次ステップの研究課題とする。
- 各々のセル内の距離はある平均移動距離の一様分布に従う乱数であると仮定する。
- モバイルユニットが各々のセルを移動していくときの速度は一定であると仮定する。

#### 5.4.2 入力パラメータ

また、次のようなパラメータを外部から指定する。

##### (1) システム構成

- ホストデータベース最大記憶可能容量 $hostdbsize$ (GByte)： ホストデータベースはギガバイト級の記憶容量を提供することが一般的であるので、本研究では表5.1に示すように、 $hostdbsize=1$  GByte を想定する。
- セルデータベース最大記憶可能容量 $celldbsize$ (MByte)： セルデータベースは、ホストデータベースの部分集合を保持することがその役割の一つであるので、メガバイト級の記憶容量と仮定し、本研究では表5.1に示すごとく、 $celldbsize=100$ Mbyte を想定する。従って、ホストデータベース対セルデータベースの記憶容量比は、約10対1である。
- セルの個数 $N$  (個)： セルの個数は、システム全体の規模を左右するものであるもので、一概には決められないが、本研究の実験的な趣旨からしてまずは小さな数から開始することに仮定し、表5.1に示すように  $N=N_x N_y=49$  を想定する。

## (2) 各計算機の平均処理時間

●ホスト計算機とホストデータベースの和 $T_h$  (秒) : ホスト計算機とホストデータベースの単位時間あたりに処理するトランザクション件数であると考えて、本研究では表5.1に示すように  $T_h=1$  (秒)、すなわち1秒間に1件のトランザクションを処理することを想定する。

●モバイルサポートサーバとセルデータベースの和 $T_c$  (秒) : モバイルサポートサーバとセルデータベースの単位時間あたりに処理するトランザクション件数であると考えて、本研究では表5.1に示すように  $T_c=10$  (秒)、すなわち10秒間に1件のトランザクションを処理することを想定する。したがって、ホスト計算機側は、モバイルサポートサーバ側の10倍の処理能力を有することを想定している。

●モバイルユニット $T_u$  (秒) : モバイルユニットが1回のデータベースアクセス結果をその画面に表示する平均時間であると考えて、本研究では表5.1に示すように  $T_u=5$  (秒) を想定する。

## (3) モバイルサポートサーバ

●セル内最大移動距離 $xdistance, ydistance$ (Km) : 1セルの範囲、言い換えればモバイルユニットがセル内を移動できる水平方向および垂直方向それぞれの直線距離の最大値であり、本研究では表5.1に示すように  $xdistance=100\text{Km}$ ,  $ydistance=60\text{Km}$  を想定する。

●時間制約時間 $timecnstrnt$  (秒) : 本研究では表5.1に示すように、 $timecnstrnt=20, 30, 60, 120, 180$  (秒) と変化させる。たとえば、 $timecnstrnt=180$ の場合、3分経過してモバイルユニットからの応答が無ければ、そのモバイルユニットはもう他のセルへ移動したと判断することになる。

●平均モバイルユニット応答待ち時間 $T_w$  (秒) : モバイルユニットがモバイルサポートサーバへ応答するために要する時間は平均60秒であると想定し、本研究では表5.1に示すように  $T_w=60$  (秒) を想定する。

## (4) モバイルユニット

●平均移動セル個数 $C_m$  (個) : このパラメータはシステムの物理的な特長を表すものではないが、本シミュレーションには必須なものである。本研究ではそれに合わせて表5.1に示すように、 $C_m=1, 3, 5$  (個) と変化させる。

●移動速度 $muspeed$ (Km/h) : モバイルユニットの実例としては自動車や電車などを前提としているので、本研究では表5.1に示すように、 $muspeed=80$ (Km/h) を想定する。すなわち、平均時速80Kmで移動する自動車あるいはオートバイなどからのデータベースアクセスモデルと言える。

#### (5) データ通信速度

●ホスト計算機とモバイルサポートサーバ間の有線通信速度 *comline*(Kbps) : 有線通信におけるデータ転送速度は現実のシステムでは数多く実現されているが、本研究では日常的な環境の一例として家庭におけるI SDNサービスを考慮して、表5.1に示すように、*comline*=64(Kbps) を想定する。

●モバイルサポートサーバとモバイルユニット間の無線通信速度 *wireless*(bps) : 無線通信におけるデータ転送速度も有線同様現実のシステムでは数多く実現されているが、本研究では日常的な環境の一例として携帯電話によるデータ通信を想定して、表5.1に示すように、*wireless*= 9600(bps) と設定する。

#### (6) データベースブロック

●ブロック長 *blocksize*(KByte) : 本研究では標準的なデータベース管理システムやオペレーティングシステムが提供しているブロック長を考慮して、表5.1に示すように、*blocksize*=4(KByte)を想定する。

●1ブロック当たりの平均処理時間(ホストデータベース) *hostmtime* (ミリ秒) : 本研究ではホストデータベースは標準的な大型大容量かつ高速なディスク上に構築すると考えて、表5.1に示すように、*hostmtime*=10(ミリ秒)を想定する。

●1ブロック当たりの平均処理時間(セルデータベース) *cellmtime* (ミリ秒) : 本研究ではセルデータベースは標準的な小型小容量かつ低速なディスク上に構築すると考えて、表5.1に示すように、*cellmtime*=50(ミリ秒)を想定する。

(7) データベースアクセス回数 *loop* (回) : 本研究ではデータベースキャッシュ管理方式の性能を平均値に代表される統計量により比較評価するため、データベースアクセス実行をかなりの回数シミュレーションする必要がある。そこで、表5.1に示すように、*loop*=500 (回) とする。

表5.1 シミュレーションの入力パラメータ

変数名	値
<i>hostdbsize</i> <i>celldbsize</i> $N_x$ $N_y$	1(GByte) 100(MByte) 7(個) 7(個)
$T_h$ $T_c$ $T_u$	1(秒) 10(秒) 5(秒)
<i>xdistance</i> <i>ydistance</i> <i>timecnstrnt</i> $T_w$	100(Km) 60(Km) 20~180(秒) 60(秒)
$C_m$ <i>muspeed</i>	1, 3, 5(個) 80(Km/h)
<i>comline</i> <i>wireless</i>	64(Kbps) 9600(bps)
<i>blocksize</i> <i>hostmtime</i> <i>cellmtime</i>	4(KByte) 10(ミリ秒) 50(ミリ秒)
<i>loop</i>	500(回)



## 5.5 性能評価結果の考察

本研究では、表5.1に示すような入力パラメータを指定し、その中で先述したキャッシュ管理方式の性能に顕著な影響を与えると定性的に予想される時間制約時間と平均移動セル個数を変化させてシミュレーションした。

### 5.5.1 時間制約時間

時間制約時間の変化に対する3種類のデータベースキャッシュ管理方式（自セルキャッシュ法、全セルキャッシュ法、隣接セルキャッシュ法）のホストデータベースアクセス経過時間の平均値 $DB_t$ を図5.11に示す。この図5.11からは、隣接セルキャッシュ法の $DB_t$ は全セルキャッシュ法よりも劣り、一方自セルキャッシュ法とほぼ同程度であることが判明した。また、時間成約時間が長くなればなるほど、 $DB_t$ が短くなっているが、これは一つのセル内に滞在している可能性が高くなった結果である。すなわち、他のセルへの移動が生じる可能性が低いので、その移動時間が短いためである。

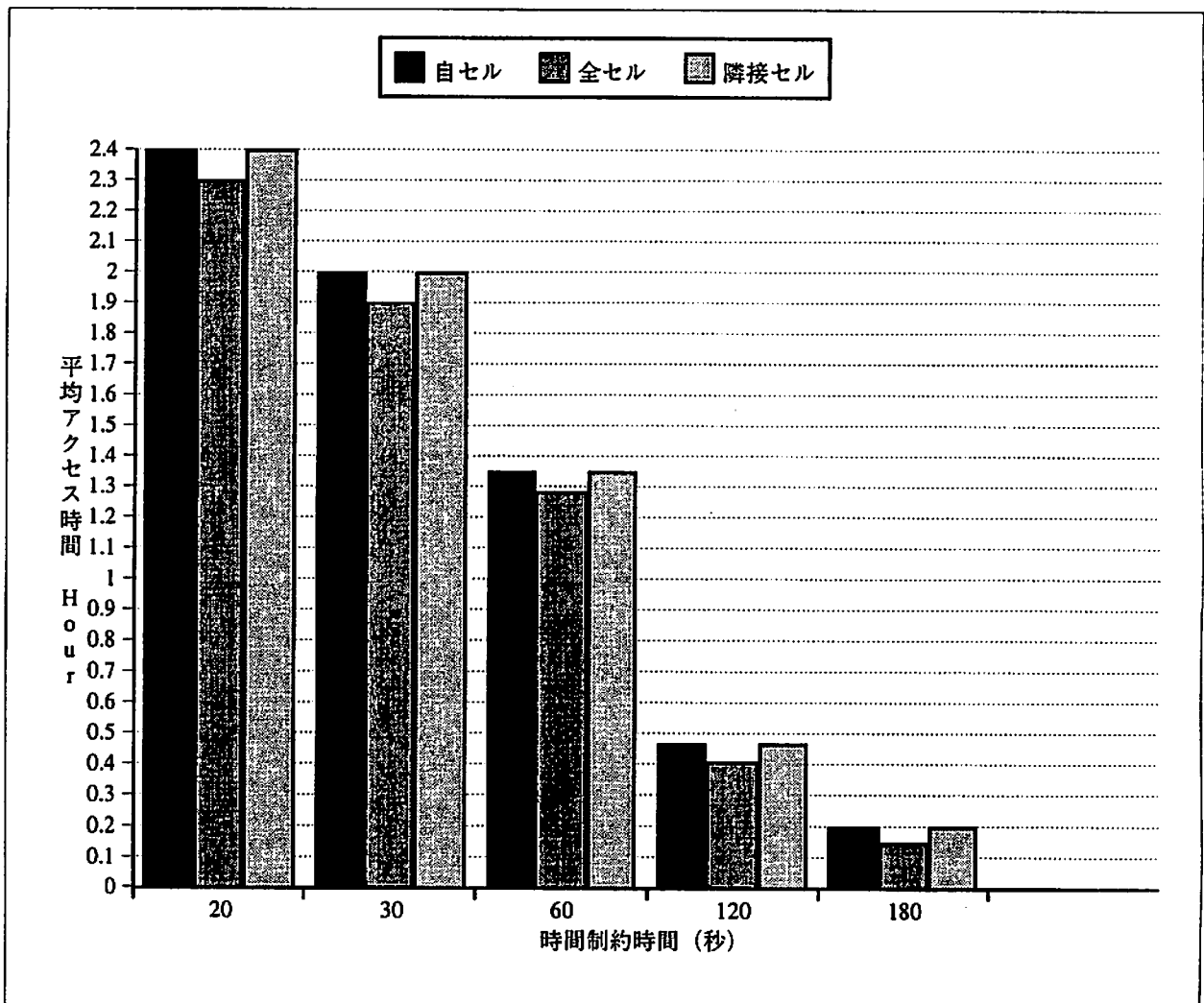


図5.11 ホストデータベースアクセス経過時間の平均値 対 時間制約時間

また、時間制約時間の変化に対する3種類のデータベースキャッシュ管理方式（自セルキャッシュ法、全セルキャッシュ法、隣接セルキャッシュ法）のセルデータベースの平均キャッシュヒット率 $CACHE_t$ を図5.12に示す。この図5.12からは、隣接セルキャッシュ法の $CACHE_t$ は自セルキャッシュ法よりも優れているが、全セルキャッシュ法よりも劣っていることが判った。

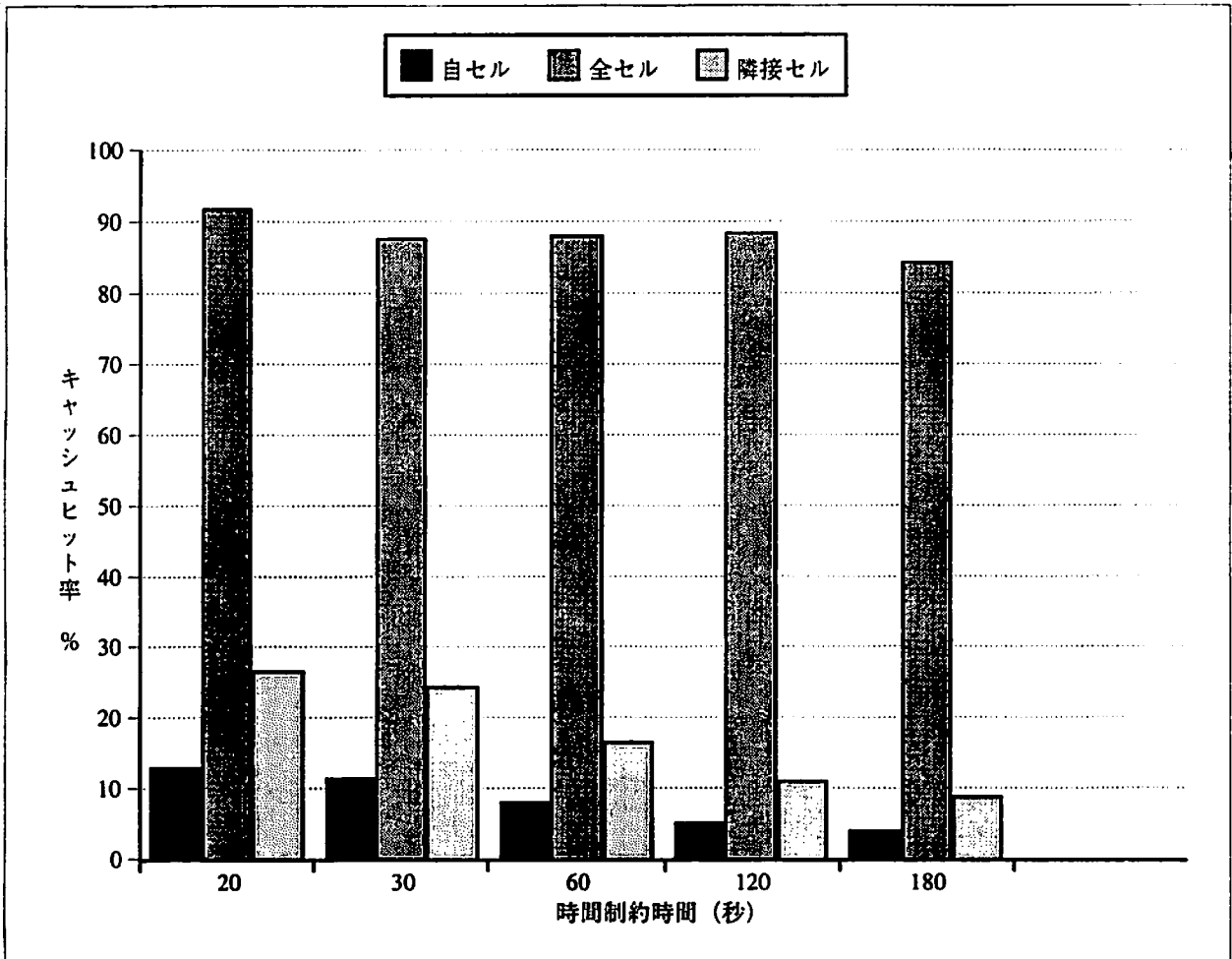


図5.12 セルデータベースの平均キャッシュヒット率 対 時間制約時間

図5.11と図5.12を対照してみると、隣接セル法の $CACHE_t$ は自セルキャッシュ法の $CACHE_t$ の約二倍であるにも関わらず、それぞれの $DB_t$ に大差が無い。これは、 $DB_t$ は式(5.1)に示すごとく、(i)モバイルユニット移動時間を組み込んでいるので、これが他の要素（例えば、式(5.1)の(e)ホスト計算機とホストデータベースの処理時間）よりもはるかに強く効いていると考えられる。

こうした点がシミュレーションモデル構築の難しいところであり、今後更なる研究が必要である。

### 5.5.2 平均移動セル個数

平均移動セル個数の変化に対する4種類のデータベースキャッシュ管理方式（隣接セルキャッシュ法、1セル置きキャッシュ法、2セル置きキャッシュ法、3セル置きキャッシュ法）のホストデータベースアクセス経過時間の平均値 $DB_t$ を図5.13に示す。この図5.13から、1セル置きキャッシュ法、2セル置きキャッシュ法、3セル置きキャッシュ法の $DB_t$ はお互いに大差は無いが、3者ともすべて、隣接セルキャッシュ法よりも優れていることが判明した。また、平均移動セル数が大きくなればなるほど、モバイルユニットの移動距離が増えるので、その結果移動時間が増えて $DB_t$ も増加している次第である。

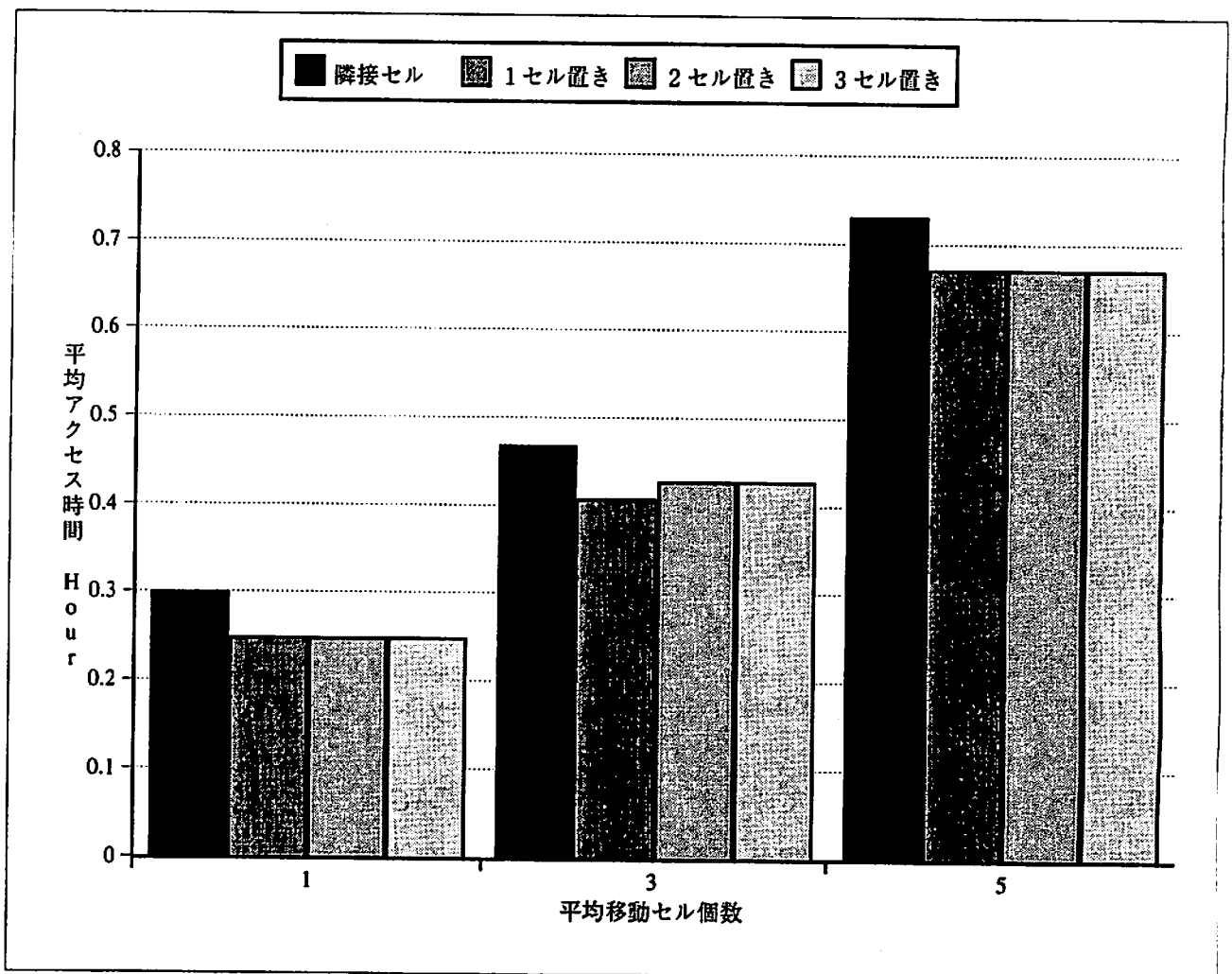


図5.13 ホストデータベースアクセス経過時間の平均値対平均移動セル個数

また、均移動セル個数の変化に対する4種類のデータベースキャッシュ管理方式（隣接セルキャッシュ法、1セル置きキャッシュ法、2セル置きキャッシュ法、3セル置きキャッシュ法）のセルデータベースの平均キャッシュヒット率 $CACHE_t$ を図5.14に示す。この図5.14からは、1セル置きキャッシュ法の $CACHE_t$ が上記の4者の管理方式の中で最良の値を示していることが判った。また、隣接セルキャッシュ法の $CACHE_t$ は残りの3者に比べてかなり劣ることが判明した。

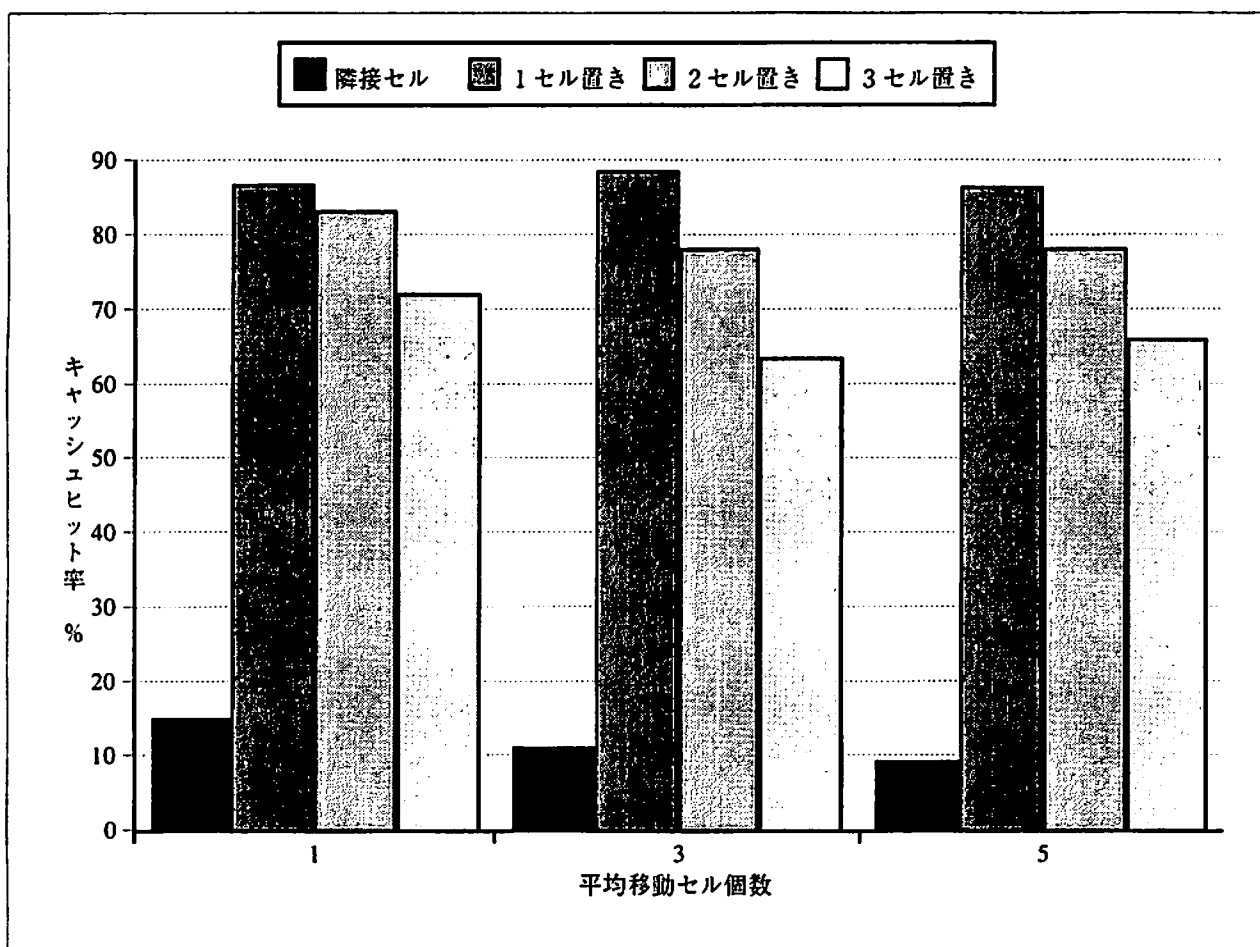


図5.14 セルデータベースの平均キャッシュヒット率 対平均移動セル個数

## 5.6 結言

本研究では移動体計算環境において、無線携帯型コンピュータ利用者が複数セル間を移動しながらホストデータベースをアクセスするときセル間でシームレスなデータベースアクセスを保証するための時間制約付きデータベースシステムを提案した。

次に、その携帯型コンピュータ利用者が複数セル間を移動しながらホストデータベースをアクセスするとき、移動先のセルでデータベースアクセスフォルトを発生する頻度を最小化するためのデータベースキャッシュ管理方式（すなわち、全セルキャッシュ法、隣接セルキャッシュ法、1セル置きキャッシュ法、2セル置きキャッシュ法、3セル置きキャッシュ法）も提案した。

最後に、各々のキャッシュ管理法の下でのホストデータベースアクセス経過時間の平均値やセルデータベースの平均キャッシュヒット率をシミュレーションにより定量的に測定して比較評価し、本研究にて提案したデータベースキャッシュ管理方式が、移動体計算環境におけるデータベースアクセスに有効であることを示した。

今後の研究課題としては、移動体計算環境におけるデータベースアクセスのモデルの精査の一言に尽きる。具体的には、まずはデータベースアクセスのための通信コストの測定評価、そして更には複数モバイルユニットの並行移動、無線通信回線のエラー時再実行処理、電源の切断と再接続後の処理、また無線通信セル相互間の共通部分あるいは間隙のモデル化などの問題が考えられる。

## 第6章 結論

本論文では、集中計算環境、分散計算環境、そして移動体計算環境におけるデータベースのアクセスと管理に関する研究をまとめた。計算環境の進歩は、集中・分散・移動体の順であるが、それぞれ重要な計算環境であり、現在も未来も共存共栄していく技術である。本論文ではデータベースの利用技術の研究から入り、データベースの性能評価の研究へと議論を進める形式でまとめており、分散環境におけるデータベースの利用技術、次に集中計算環境におけるデータベースシステムの性能評価、そして移動体計算環境におけるデータベースシステムの性能評価の順番で各々の成果について述べた。

初めに、分散計算環境におけるデータベースの利用技術に関する3件の研究課題に対する成果を総括すると、先ずクライアント/サーバ環境における位置透過性を実現した視覚的なデータベース・ユーザフレンドリインタフェース技術という解を見つけ、エンドユーザコンピューティングの問題を解決した。更に垂直分散化と操作の統一性を特長とするDSSを試作し、データベースの意思決定業務への適用性を向上させた。そして、データベース技術とAI技術と統合化した新たな意思決定支援技術として最近注目されているデータマイニング技術を研究し、従来のデータベースアクセス技術やDSSでは提供できないデータ間規則性の発見が可能となった。その規則性を用いることによりより高度な意思決定支援技術が実現された。

この中で、「データベースの利用技術」に関して、「使い易いデータベース利用者インタフェースMELQUERYとMELQUERY/R」の実装結果をまとめた。

また、「OA業務および意思決定業務への応用」に関して、「意思決定支援システムDSS-GOAL」を実装し、更に「データマイニングサーバKnodi as」の試作を行い、その結果をまとめた。

次に、集中計算環境におけるデータベースシステムの性能評価に関する2件の研究課題に対しては、先ず論理設計、物理設計、スキーマ性能評価の各設計フェーズを統合化したデータベース設計自動化・性能予測ツールを試作し、データベースのアクセス費用と性能をデータベース構築の事前に予測と改善を可能とした。また木構造データベースの各層に優先順位を考慮したバッファ管理方式を提案し、データベースアクセス時のバッファ不在頻度を最小化することを計算機シミュレーションにより検証した。

この中で、「集中計算環境におけるデータベースシステムの性能評価」に関して、「データベースの設計自動化支援・性能予測ツールDMSDBDA」を実装し、更に「MUMPSデータベース管理システムのバッファ管理方式BPL法、BPL\*法」の性能評価を行い、その結果をまとめた。前者のDMSDBDAは研究試作が完了する段階にまで到達し、実際のデータベース設計評価の問題を解決することができた。一方、BPL法やBPL\*法は、商用のMUMPSデータベース管理システムの中の構成要素と実装されて、医療情報データベースシステムなどの業務分野でその有効性が判った。

最後に、移動体計算環境において、無線携帯型コンピュータ利用者が複数セル間を移動しながらホストデータベースをアクセスするときセル間でシームレスなデータベースアクセスを保証するための時間制約付きデータベースシステムを提案した。次に、その携帯

型コンピュータ利用者が複数セル間を移動しながらホストデータベースをアクセスするとき、移動先のセルでデータベースアクセスフォルトを発生する頻度を最小化するためのデータベースキャッシュ管理方式（すなわち、全セルキャッシュ法、隣接セルキャッシュ法、1セル置きキャッシュ法、2セル置きキャッシュ法、3セル置きキャッシュ法）も提案した。

そして、各々のキャッシュ管理法の下でのホストデータベースアクセス経過時間の平均値やセルデータベースの平均キャッシュヒット率をシミュレーションにより定量的に測定して比較評価し、本研究にて提案したデータベースキャッシュ管理方式が、移動体計算環境におけるデータベースアクセスに有効であることを示した。

この「移動体計算環境におけるデータベースシステムの性能評価」に関する研究成果であるデータベース・キャッシュ管理方式は、移動体計算環境の共通基盤技術として今後かなり大きな応用可能性があるかと期待できる。例えば、地震や火災などに事前に備えるための防災情報システム、またそうした地震や火災などが発生したあとの危機管理情報システムへの展開が考えられる。こうしたシステムでは、災害現場にいる個人個人と災害対策本部との間での有線によるインターネット/イントラネット経由でのデータベースアクセスはもちろん、その有線回線が断線した場合の代替通信回線である無線通信（携帯電話から衛星通信回線までを含めた。）による情報の双方向通信、すなわちデータベースアクセスが必須な技術となると予想される。

以上の各々の研究成果間の関連性を整理してみると、「意思決定支援システムDSS-GOAL」の成果は、「使い易いデータベース利用者インタフェースMELQUERYとMELQUERY/R」の成果と共に、「データマイニングサーバKnodias」の研究へと活用された。一方、「データベースの設計自動化支援・性能予測ツールDMSDBDA」と「MUMPSデータベース管理システムのバッファ管理方式BPL法、BPL\*法」は、やはり「使い易いデータベース利用者インタフェースMELQUERYとMELQUERY/R」の成果と共に、「移動体計算環境におけるデータベースシステムの性能評価」に活かされた。そして、この二テーマの成果を基に「分散・移動体計算環境におけるデータベースシステムを用いた意思決定技術」の研究を更に進めていくことが重要であると考えている。

更に、データベース技術の全体像の観点から、本研究課題である「集中・分散・移動環境におけるデータベースのアクセスと管理に関する研究」の成果を概括してみる。データベース技術の全体像を、データベース要素技術の軸と計算環境の軸とから成る2次元マトリックス表に写像すると、先ずデータベース要素技術は、例えば情報処理学会の論文誌投稿用キーワード表によると、(1) データモデル、(2) データベース言語、(3) データベース設計、(4) 一貫性制約、(5) データベース管理システム、(6) 質問処理、(7) トランザクション管理、(8) データベース構成（編成とアクセス法）、(9) 信頼性・障害管理、(10) 並列・超並列データベース、(11) オンライントランザクション処理、(12) 分散マルチデータベース、(13) データベースOS、(14) 性能評価、(15) マルチメディアデータベース、(16) 演繹データベース、(17) オブジェクト指向データベース、(18) オフィスデータベース、(19) エンジニ

アリングデータベース、(20) 文書データベース、(21) 画像データベース、(22) 音声データベース、(23) 地理データベース、(24) データディクショナリ・リポジトリ、(25) データベース応用(システム、開発環境)、(26) データベースシステム技術(データ分析・設計・構築・運用)、(27) ハイパーテキスト、(28) ハイパーメディア、(29) データベースマシンなどに分類されている。一方、計算環境の軸は、今まで議論してきたごとく、(A) 集中計算環境、(B) 分散計算環境、(C) 移動体計算環境に分類できる。ここで、本研究の成果をこのマトリックス表に位置付けてみると、第3章の3.2節に記述した「使い易いデータベース利用者インタフェース MELQUERYとMELQUERY/R」の成果は、(2)と(6)の(A)と(B)に相当する。

また、同じく3.3節に記述した「意思決定支援システムDSS-GOAL」と「データマイニングサーバKnodias」の成果は、(26)の(A)と(B)に該当する。次に、第4章の4.2節に記述した「データベースの設計自動化支援・性能予測ツールDMSDBDA」は、(3)と(8)と(14)の(A)に当てはまり、同じく4.3節に記述した「MUMPSデータベース管理システムのバッファ管理方式BPL法、BPL\*法」の成果は、(5)と(8)の(A)に対応する。そして、第5章に記述した「移動体計算環境におけるデータベースシステムの性能評価」の成果は、(5)と(8)と(12)の(C)に位置付けられる。この結果を要約すると、上記のマトリックス表の87個の要素に対して、本論文にて議論してきたのはこの内14要素であることが判り、まだまだ今後研究すべき課題が山積みであると考えられる。

続いて、データベース技術の将来展望を議論していく。

データベース技術の研究開発の世界で最も影響力があったものはやはりリレーショナルモデルの提案である。このリレーショナルモデルの概念がE.F.Codd博士から発表されてから20数年が経過し、その後データベースの世界も標準化の思想が生まれ、10年程前のデータベース言語SQLのISOによる国際規格策定、そしてこのSQL規格後に急速に進展してきた業界標準という考え方が世の中に広く定着し、現在に至っている。この間、ポスト・リレーショナルデータベースとして、オブジェクト指向データベースに代表される新データモデルが数多く提案、研究され、そして実用化されてきた。また、ネットワーク技術の発達による分散処理技術の進歩に歩調を合わせて、研究が進んできた分散データベース技術、特にオンライントランザクション処理やクライアント/サーバ・アーキテクチャの技術と巧く連携したデータベース・サーバの普及も促進されてきた。次いで、マルチメディア情報処理技術の進歩と共に、マルチメディア・データベースシステムの応用研究開発も活発に行われている。

しかしながら、データベース技術の研究の基本は、やはりリレーショナルデータベースである。そして、今後の研究の方向は、このリレーショナルデータベースを拡張するアプローチ、言い換えればオブジェクト・リレーショナルアプローチであると考えられる。この最大の根拠は、リレーショナルデータベースの技術的な限界は明らかに存在するが、そうかと言ってその膨大な過去の資産を無視できないということである。こうした考え方の根拠を裏付ける別の具体的な出来事としては、1996年9月にインド国ボンベイ市にて



開催され第22回大規模データベース国際会議VLDB'96の中で、10年前のVLDB'86で発表された論文の中で、最も強いインパクトをデータベースの世界に与えた論文は、"Object and File Management in the EXODUS Extensible Database System"であるとして、著者であるIBM社Alamaden研究所のM. J. Carey博士らに「Best Impact Paper of VLDB'86」賞が授与された。因みに、この論文は、従来のリレーショナルデータベースを拡張して、オブジェクト等の概念を追加した拡張可能なデータベースシステムEXODUSについて報告したものであり、ここでのポイントは1986年頃はこうした拡張アプローチは異端児扱いされており、当時の主流はオブジェクト指向データベース管理システムを従来のリレーショナル・データベース管理システムとは独立した概念で実現するアプローチであったということである。例えば、Gemstone, Vbase, Orionなどがその典型的な例である。

一方、こうした将来展望と本論文との関係を整理すると、本論文はリレーショナルデータベースを研究の基盤として捉えて研究を行ない、それにオブジェクト指向の概念を一部ユーザインタフェースの分野に導入してきたので、本論文の研究もこの将来展望と同一線上に乗っていると言える。

次に、最新のデータベース技術の研究動向を整理してみると、次のように分類できる：空間アクセスメソッド、データウェアハウス、問い合わせ最適化、データマイニング、問い合わせ処理、問い合わせと情報検索、異種データベース、データベース応用、入出力最適化、オブジェクト指向問い合わせ処理、並列問い合わせ処理、データ完全性/安全性、OLAP(On-Line Analytical Processing)等。こうした研究動向の中で、本論文にまとめた研究課題のデータベースキャッシュ管理やバッファ管理は上記の分類の中の「入出力最適化」に近いものであり、他の課題（問い合わせ処理、異種データベースなど）にも深い関係がある。

最後に、先述したオブジェクト・リレーショナルアプローチにおけるリレーショナルデータベース管理システムの今後10年間の研究課題としては、（1）データベースサーバの機能と性能の強化、（2）クライアント統合、（3）並列化、（4）レガシーデータベース、（5）標準化の4項目が考えられる。各々の課題をもう少し、詳細化すると、（1）は産業上の用途に耐え得るような性能の達成、SQL言語の完全なオブジェクト化、拡張可能なアクセスメソッドなど、（2）はクライアント側のキャッシュとデータベース本体にまたがる問い合わせ処理など、（3）は抽象データ型処理の並列化、（4）はリレーショナルおよびリレーショナル以前のデータベースに格納されているデータの扱い、およびこのデータとオブジェクト・リレーショナルを合わせたような異種混合のデータソースにまたがる問い合わせ処理、（5）は外部のプログラミング言語で定義された抽象データ型に対する問い合わせ最適化情報の与え方、クライアント/サーバ間のインタフェースなどである。この課題は古くて新しい保守的なニュアンスがするが、データベースを研究していくためには避けて通れない非常に重要な課題であるので、筆者はこの中で特に（1）、（2）、（4）が重要であると考えており、本論文の研究もこうした課題の解決

のための第一歩となったと自負している。従って、こうした課題はデータベース技術の重要課題として、今後も多いに研究していくべきであると考えられる。

## 謝辞

静岡大学社会人博士課程における本研究に対してご指導ならびに有益なご意見を頂いた静岡大学情報学部教授 水野忠則博士、並びに静岡大学工学部教授 深林太計志博士に深く感謝致します。

更に、本論文に関し貴重なご助言を賜った静岡大学情報学部教授 中谷広正博士、および静岡大学情報学部助教授 伊東幸宏博士にも深く感謝します。

そして、日常の研究活動の中でご指導並びにご支援を頂いた静岡大学情報学部教授 渡辺尚博士、および静岡大学情報学部助教授 佐藤文明博士にも感謝します。

一方、静岡大学社会人博士課程への入学および研究活動の機会を与えて頂きました静岡大学情報学部教授 曾我正和博士（元三菱電機株式会社情報システム研究所所長）、三菱電機株式会社常務開発本部長 野間口有博士、財団法人道路交通情報通信システムセンター 角田正俊氏（元三菱電機株式会社情報技術総合研究所開発支援部次長）、三菱電機株式会社情報技術総合研究所ネットワークコンピュータ部部长 下間芳樹氏、同システム部部长 坂下善彦氏、ならびに三菱電機株式会社鎌倉製作所情報システム部部长 小島利明氏と同プロジェクト部部长兼CCVGグループマネージャ中島 務氏にも深く感謝します。

そして、三菱電機株式会社における本研究の遂行に当り、御指導を賜りました大阪大学基礎工学部教授 首藤勝博士（元三菱電機株式会社情報電子研究所副所長）、前静岡大学工学部教授 関本彰次博士（元三菱電機株式会社情報電子研究所主幹）、三菱電機株式会社情報システム製作所参事 溝口徹夫博士、ならびに三菱電機株式会社開発本部技師長 石田喬也氏に深く感謝致します。

最後に、本研究に御協力頂いた三菱電機株式会社情報システム製作所主事 金森卓郎氏、同 森川修一氏、元三菱電機株式会社情報システム研究所 中川路（旧姓、赤堀）のぞみ氏、ならびに三菱電機株式会社情報技術総合研究所チームリーダー 三石彰純氏、ならびに同主事 和田信義氏に感謝します。また、本研究に対して適切な議論をして頂いた静岡大学大学院理工学研究科博士前期課程2年 横田恵一君にも感謝します。

そして、本論文のワープロ化作業を支援してくれた三菱電機株式会社情報技術総合研究所開発支援部 内海美雪嬢、同情報通信システム開発センター 宮嶋美由紀嬢にも感謝します。

以上

## 参考文献

[Agrw94]

Agrawal, R., and Srikant, R. "Fast Algorithm for Mining Association Rules", Proc. VLDB'94 (1994).

[Aka86]

赤堀、金森、渡辺、和田、"ワークステーションDBMSにおけるユーザインタフェース" 第2回ヒューマン・インタフェース・シンポジウム, 2333, pp.429-434(1986-10).

[Aka87]

赤堀、金森、渡辺、和田、"分散DBMSのユーザインタフェース" 第3回ヒューマン・インタフェース・シンポジウム, 3222, pp.373-376(1987-10).

[Alon93]

Alonso, R. and Korth, H.F., "Database System Issues in Nomadic Computing." Proc. ACM SIGMOD 93, pp.388-392 (1993).

[Amano96]

天野、"第22回大規模データベース(VLDB'96)国際会議報告" ACM SIGMOD'96日本大会, pp.29-37 (1996-10).

[Barb94]

Barbara, D. and Imielinski, T., "Sleepers and Workaholics: Caching Strategies in Mobile Environments." Proc. of ACM SIGMOD 94, pp.1-12 (1994).

[Bere77]

Berelian, E. and Irani, K.B., "Evaluation and Optimization." Proc. 3rd VLDB, pp.545~555(1977).

[Beri77]

Berild, S. and Nachmens, S., "CS 4-A Tool for Database Design by Infological Simulation." Proc. 3rd VLDB, part II, pp.85~94(1977).

[Bowi77]

Bowie, J., "Methods of Implementation of The MUMPS Global Data Base." Laboratory of Computer Science at Massachusetts General Hospital(1977).

[Bube76]

Bubenko jr,J.A., Berild,S., Lindencrona-Ohlin,E. and Nachmens,S., *"From Information Requirements to DBTG-Data Structures."*

Proc. Conf. on Data: Abstraction, Definition and Structure, ACM SIGPLAN SIGMOD,Vol.8, No.2, pp.73~85(1976).

[Chan80]

Chang,N.S., and Fu,K.S., *"Query by Pictorial Example."*

IEEE Trans. Soft. Eng., Vol.SE-6, No.6,pp.519-524(1980).

[Chan81]

Chang,N.S., and Fu,K.S., *"Picture Query Languages for Pictorial Data-Base Systems."*

IEEE Computer Vol.14, pp.23-33 (Nov.1981).

[Chen76]

Chen,P.P., *"The Entity-Relationship Model: Toward A Unified View of Data."*

ACM TODS Vol.1, No.1, pp.9~36, March(1976).

[Chen77-1]

Chen,P.P. and Yao,S.B., *"Design and Performance Tools for Data Base Systems."*

Proc. 3rd VLDB, pp.3~15(1977).

[Chen77-2]

Chen,P.P., *"The Entity-Relationship Model-A Basis for The Enterprise View of Data."*

Proc. AFIPS NCC, Vol.46, pp.77~84(1977).

[Davi86]

Davison,J.W., and Zdonik,S.B., *"A Visual Interface for a Database with Version Management."*

ACM TODS, Vol.4, No.3, pp.226-256 (July 1986).

[Degu79]

出口、武田、峰崎、古家、和田, *"医療情報データベースシステム<<MELCOM MUMPS>>"*

三菱電機技報 Vol.53, No.2, pp.154~158 (1979).

[Fern78]

Fernandez, E. B., Lang, T. and Wood, C., "*Effect of Replacement Algorithms on A Paged Buffer Database System.*"  
IBM J. Res. Dev. Vol.23, No.2, pp.185~196 March(1978).

[Fras86]

Frasson, C., and Er-radi, M., "*Principles of an Icon-Based Command Language.*"  
Proc. of ACM SIGMOD'86, pp.144-152 (May 1986).

[Fuku96]

Fukuda, T., et al., "*Data Mining Using Two-Dimensional Optimized Association Rules : Scheme, Algorithms, and Visualization.,*"  
Proc. ACM SIGMOD'96(1996).

[Gamb77]

Gambino, T.J. and Gerritsen, R., "*A Data Base Design Decision Support System.*"  
Proc. 3rd VLDB, pp.534~544(1977).

[Gerr75]

Gerritsen, R., "*A Preliminary System for the Design of DBTG Data Structures.*"  
CACM Vol.18, No.10, pp.551~556 (1975).

[Gerr77]

Gerritsen, R., Gambino, T.J. and Germano, Jr., F., "*Cost Effective Database Design: An Integrated Model.*"  
Working paper 77-12-03, Dept. of Decision Sciences,  
The Wharton School, University of Pennsylvania(1977).

[Gold85]

Goldman, K.j., et al., "*ISIS:Information for a Semantic Information System.*"  
Proc. of ACM SIGMOD'85, pp.328-342 (May 1985).

[Han95]

Han, J., Fu, Y., "*Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases.,*"  
Proc. VLDB'95(1995).

[Heil85]

Heiler, S., and Rosenthal, A., "*G-WHIZ, a Visual Interface for the Functional*

*Model with Recursion."*

Proc. of the 11th VLDB, pp.209-218 (August 1985).

[Hiro83]

広内、小坂、"意思決定支援システムDSS構築の方法"

竹内書店新社(1983).

[Hota78]

穂鷹, "データベース要論"

共立出版社 (1978).

[Hous79]

Housel,B.C., Waddle,V. and Yao,S.B., "*The Functional Dependency Model for Logical Database Design.*"

Proc. Fifth VLDB (1979).

[Huan94-1]

Huang,Y., Sistla,P. and Wolfson,O., "*Data Replication for Mobile Computers.*"

Proc. of ACM SIGMOD 94, pp.13-24 (1994).

[Huan94-2]

Huang,Y. and Wolfson,O., "*Object Allocation in Distributed Databases and Mobile Computers.*"

IEEE Proc. of the 10th DE'94, pp.20-29 (1994).

[Hult77]

Hulten,C. and Soderlund,L., "*A Simulation Model for Performance Analysis of Large Shared Data Bases.*"

Proc. 3rd VLDB, pp.524~532(1977).

[IBM83]

IBM, "特集 意思決定支援システム"

IBM REVIEW 1983-91(1983).

[Imie92]

Imielinski, T. and Badrinath, B. R. , "*Querying in Highly Mobile Distributed Environments.*"

Proc. 18th VLDB, pp.41-52, (1992).

[Imie93]

Imielinski, T. and Badrinath, B. R., "Data Management for Mobile Computing."  
ACM SIGMOD RECORD, Vol.22, No.1, pp.34-39 (1993).

[Imie94-1]

Imielinski, T. and Badrinath, B. R., "Mobilewireless Computing."  
Commun. ACM, pp.18-28, (Oct. 1994).

[Imie94-2]

Imielinski, T., Viswanathan, S. and Badrinath, B. R., "Energy Efficient Indexing  
on Air."  
Proc. of ACM SIGMOD 94, pp.25-36 (1994).

[Ishi94]

石川 宏, "Cによるシミュレーションプログラミング"  
ソフトバンク株式会社, p.286 (1994).

[Ishi96]

石井他, "業務ノウハウを活用するデータマイニングサーバKnodias"  
第53回情報処理学会全国大会3R-5 (1996-9).

[JIS87]

JIS, "データベース言語SQL"  
JIS X 3005、日本規格協会(1987).

[Kami95]

上林、早瀬, "移動分散データベースにおける質問処理"  
電子情報通信学会 信学技報, DE95-28 (1995).

[Kana87]

金森、赤堀、渡辺、和田、溝口、'ワークステーションデータベース管理システム-WS・  
DBMS-',  
情報処理学会データベース・システム研究会, 57-6(1987-1).

[Kana88]

金森、中川路、和田、石田、'データベースのユーザ・フレンドリインタフェース  
MELQUERY',  
情報処理学会ADBS'88, pp.127-136(1988-12).



[Kato87-1]

加藤、外川、鈴木、織田、"視覚情報を用いたオブジェクト指向RDB検索法"  
電子情報通信学会論文誌 Vol.J70-D, No.11, pp.2236-2242 (1987-11).

[Kato87-2]

加藤他、"視覚情報を用いたオブジェクト指向によるRDB検索法"  
情報処理学会データベースシステム研究会DB-57-10 (1987-1).

[Koji88]

Kojima,I., and Tanuma,H., "*Workstation Database, Visual Interface, Distributed Processing, Object-Oriented, Local Area Networks.*"  
JAACE SYSTEMS AND CONTROL, Vol.32, No.5, pp.294-301 (1988).

[Lai95]

Lai, S. J., Zaslavsky, A., Martin, G. P. and Yeo, L. H., "*Cost Efficient Adaptive Protocol with Buffering for Advanced Mobile Database Applications.*"  
Proc. 4th DASFAA'95, pp.87-94, (April 1995).

[Lan77]

Lang, T., Wood, C. and Fernandez, E. B., "*Database Buffer Paging in Virtual Storage Systems.*"  
ACM TODS Vol.2, No.1, pp.339~351 March(1977).

[Maki87]

牧之内他、"マルチメディアデータベースの利用者インタフェース"  
情報処理、Vol.28, No.6, pp.694-704 (1987-6).

[Mart86]

Martin,J., and Leben,J., "*Fourth-Generation Language.*"  
Vol.2, Prentice-Hall, Englewood Cliffs, N.J., (1986).

[Melco76]

三菱電機株式会社、"MELCOM EDMS説明書"  
NM-SR00-54A(1976).

[Mori91]

森川、魚住、和田、大内、"遠隔データベース リクエストMELQUERY/R"  
三菱電機技報, Vol.65, No.7, pp.58-68 (1991).

[Naka86]

Nakagawa,M., and Kato,T., *"Methods of the Semantic Analysis in a Natural Language Interface for Relational Database."*

Trans. of Information Processing Society of Japan, Vol.27, No.11, pp.1069-1076 (Nov.1986).

[Naka89]

中川路、金森、和田、石田、'データベースのユーザフレンドリインタフェース-MELQUERY-における拡張機能"

情報処理学会データベース・システム研究会, 70-2(1989-3).

[Naka95]

仲秋、劉、塚本、西尾、"移動データベース環境におけるセル指定ビューの維持方法"

情報処理学会DSP研資料, pp.175-180 (1995).

[Nava78]

Navathe,S.B. and Schkolnick,M., *"View Representation in Logical Database Design."*

Proc. ACM SIGMOD, Austin, Texas, pp.144~156 May (1978).

[NIKKEI82]

日経、"製品調査： デシジョンサポートシステム DSSバックログ解消の強力なツールになるDSS"

日経コンピュータ、1982.12.27(1982).

[Nishi95]

西田、"ソフトウェアエージェント"

人工知能学会誌, Vol.10, No.5, pp.704-711 (1995).

[Nishi97]

西尾、塚本、"高度データベースプロジェクト 高度ネットワーク環境とデータベースシステム"

Computer Today, pp.9-14, No.79 (1997-5).

[Oogu76]

大櫛、"MUMPS—そのOS、プログラミング、ファイル"

情報処理学会データベース研究会 31-2 (1976-11).

[Schk78]

Schkolnick,M., *"A Survey of Physical Database Design Methodology and*

*Techniques."*

Proc. Fourth VLDB, pp.474~487(1978).

[Schi79]

Schiffner,G. and Scheuerman,P., *"Multiple Views and Abstraction with An Extended Entity-Relationship Model."*

Northwestern University Department of Electrical Engineering and Computer Science, No.79-03-DBM-02(1979).

[Sher76]

Sherman, S. W. and Brice, R. S., *"Performance of A Database Manager in A Virtual Memory System."*

ACM TODS Vol.1, No.4, pp.317~ 343 December(1976).

[Shin96]

新谷, 喜連川, *"データマイニングにおける相関関係抽出の並列処理方式の実装とその評価"*

情報処理学会 並列処理シンポジウムJSP'96, pp.97-104(1996).

[Shir96]

白井, 仲秋, 劉, 塚本, 西尾, *"移動型データベースのためのビュー機構の設計および実装"*

電子情報通信学会 信学技報, DE95-27 (1995).

[Shrk95]

Srikant, R., Agrawal,R., *"Mining Generalized Association Rules, "*

Proc. VLDB'95 (1995).

[Smit77-1]

Smith,J.M. and Smith,D.C.P., *"Database Abstractions: Aggregation."*

CACM Vol.20, No.6, pp.405~413(1977).

[Smit77-2]

Smith,J.M. and Smith,D.C.P., *"Database Abstractions: Aggregation and Generation."*

ACM TODS Vol.2, No.2, pp.105~133(1977).

[Tagu86]

田口、和田、*"ワークステーション用データベース"*

電気学会雑誌、Vol. 106, No. 12, pp.21-24(1986-12).

[Tayl76]

Taylor,R.W. and Frank,R.L., "CODASYL Data-Base Management Systems."  
ACM Comput. Surv., Vol.8, No.1, pp.67~103(1976).

[Teor78]

Teorey,T.J. and Oberlander,L.N., "Network Database Evaluation Using  
Analytical Modeling."  
Proc. NCC, pp.833~842(1978).

[Tsuk96]

塚本、"モバイルコンピューティング環境のためのデータベース"  
第3回データエンジニアリングフォーラム~新しいデータベース像を求めて~、  
pp.1~10(1996-4).

[Wada79-1]

和田、溝口., "MUMPSデータベースシステムのバッファ置換アルゴリズム"  
電子通信学会 論文誌 Vol.J62-D, No.11 pp.766~773 (1979-11).

[Wada79-2]

和田、溝口., "MUMPSデータベースのバッファ管理方式性能評価"  
電子通信学会 信学技報 EC78-70(1979-02).

[Wada80]

和田、溝口、中村., "データベースの設計と評価のためのツールの試作-MELCOM700デ  
ータベース(EDMS)向け-。  
情報処理学会DBMS研究会、17-4(1980).

[Wada81]

和田、溝口., "画面指向形のデータベース設計・評価ツールDMSDBDAの試作"  
電子通信学会 論文誌 Vol.J64-D, No.2 pp.140~147(1981-02).

[Wada84]

和田、溝口, "意思決定支援システム"  
オフィスオートメーション、Vol. 5、No. 5、pp. 153-160(1984).

[Wada85-1]

和田、金森、溝口、"統合化利用者視野を提供するDSS-GOAL"  
情報処理学会ADBS'85, pp.21-30 (1985-12).

[Wada85-2]

和田、立花、金森、中村、溝口、"<<MELCOM DPシリーズ>>向け意思決定支援システム DSS/1"

三菱電機技報, Vol.59, No.9, pp.51-55 (1985).

[Wada87]

和田、赤堀、金森、渡辺., "ワークステーションデータベース管理システムMELQUERY"  
電子情報通信学会データ工学研究会, DE-87-18 (1987).

[Wada88]

和田、中川路、金森., "データベースのユーザフレンドリインタフェースMELQUERY"  
三菱電機技報, Vol.62, No.10, pp.63-68 (1988).

[Wada89]

Wada,Y., Nakakawaji,N.,Kanamori,T. and Ishida,T., "A Table Driven User-Friendly Interface for Relational Databases."  
DASFAA'89, pp.150-157, April(1989).

[Wada96-1]

和田、水野、'移動計算環境におけるデータベースシステムのキャッシュ管理方式の性能評価',  
静岡大学大学院電子科学研究科 研究報告、No.17,pp.137-144(March 1996).

[Wada96-2]

Wada, Y. and Mizuno, T., "Database Cache Management Algorithms of A Timing Constrained Database System in Mobile Computing Environments."  
IEICE Trans., Vol.E79-A, No.7 (July 1996).

[Wada96-3]

Wada, Y. and Mizuno, T., "Performance Evaluation of Database Cache Management in Mobile Computing Environments."  
Proc. of the 9th ISCA, pp.678-683 (Sept. 1996).

[Wadax96]

和田(信)他、 "データマイニングサーバKnodiasにおける属性情報利用方式の検討"  
情報処理学会DBS研究会110-2(1996.10.24).

[Wass75-1]

Wasserman,A.I.,Sherertz,D.D.and Rogerson,C.L.,"*MUMPS Globals and Their Implementation.*"

MUMPS Development Committee 2/1,11 June(1975).

[Wass75-2]

Wasserman,A.I.,sherertz,D.D. and Zears,R.W.,"*Design of A Multiprogramming System for The MUMPS Language.*"

MUMPS Development Committee 2/2,11 June(1975).

[Wats88]

Watson,S.L., "*OS/2 Query Manager Overview and Prompted Interface.*"

IBM Systems Journal, Vol.27, No.2, pp.119-133 (1988).

[Whan87]

Whang,K.-Y.,et al., "*Office-by-Example: An Integrated Office System and Database Manager.*"

ACM TODS, Vol.5, No.4, pp.393-427 (Oct.1987).

[Yasu97]

安田他, "属性指向データマイニングサーバKnodiasの試作"

第54回情報処理学会全国大会(1997-3).

[Yone79]

米田、山口., "汎用DBMSの性能推定システム"

情報処理学会 論文誌 Vol.20, No.4 pp.314~321(1979-04).

[Zloof75]

Zloof,M.M., "*Query By Example.*"

Proc. NCC44(1975-5).

[Zloof77]

Zloof,M.M., "*Query-by-Example: A Data Base Language.*"

IBM Systems Journal, Vol.16, No.4, pp.324-343 (1977).

## 研究成果一覧

[Wada79-1]

和田、溝口., "MUMPSデータベースシステムのバッファ置換アルゴリズム"  
電子通信学会 論文誌 Vol.J62-D, No.11 pp.766~773 (1979-11).

[Wada79-2]

和田、溝口., "MUMPSデータベースのバッファ管理方式性能評価."  
電子通信学会 信学技報 EC78-70(1979-02).

[Wada80]

和田、溝口、中村., "データベースの設計と評価のためのツールの試作—MELCOM700データベース(EDMS)向け—"  
情報処理学会DBMS研究会、17-4(1980).

[Wada81]

和田、溝口., "画面指向形のデータベース設計・評価ツールDMSDBDAの試作"  
電子通信学会 論文誌 Vol.J64-D, No.2 pp.140~147(1981-02).

[Wada84]

和田、溝口, "意思決定支援システム"  
オフィスオートメーション、Vol. 5、No. 5、pp. 153-160(1984).

[Wada85-1]

和田、金森、溝口、"統合化利用者視野を提供するDSS-GOAL"  
情報処理学会ADBS'85, pp.21-30 (1985-12).

[Wada85-2]

和田、立花、金森、中村、溝口、"<<MELCOM DPシリーズ>>向け意思決定支援システムDSS/1"  
三菱電機技報, Vol.59, No.9, pp.51-55 (1985).

[Wada87]

和田、赤堀、金森、渡辺., "ワークステーションデータベース管理システムMELQUERY"  
電子情報通信学会データ工学研究会, DE-87-18 (1987).

[Wada88]

和田、中川路、金森., "データベースのユーザフレンドリインタフェースMELQUERY"  
三菱電機技報, Vol.62, No.10, pp.63-68 (1988).

[Wada89]

Wada, Y., Nakakawaji, N., Kanamori, T. and Ishida, T., "A Table Driven User-Friendly Interface for Relational Databases." DASFAA'89, pp.150-157, April(1989).

[Wada95-1]

和田、水野、'モバイルコンピューティング環境における時間制約付きデータベースシステムの性能評価"  
情報処理学会DPS研究会、71-4, pp.19-24(1995).

[Wada95-2]

和田、水野、'モバイルコンピューティング環境における時間制約付きデータベースキャッシュ管理方式"  
情報処理学会DPSワークショップ、pp.117-124(1995).

[Wada96-1]

和田、水野、'移動計算環境におけるデータベースシステムのキャッシュ管理方式の性能評価"  
静岡大学大学院電子科学研究科 研究報告、No.17, pp.137-144(March 1996).

[Wada96-2]

Wada, Y. and Mizuno, T., "Database Cache Management Algorithms of A Timing Constrained Database System in Mobile Computing Environments." IEICE Trans., Vol.E79-A, No.7 (July 1996).

[Wada96-3]

Wada, Y. and Mizuno, T., "Performance Evaluation of Database Cache Management in Mobile Computing Environments." Proc. of the 9th ISCA, pp.678-683 (Sept. 1996).



## 付録

(1) データベースの設計自動化支援・性能予測方式  
性能評価のために用いる数式モデルの概略を以下にまとめる。

[定義] 諸変数を以下のように定義する。

●変数 $z_i$ ,  $i \in R$ を式(A.1)のように定義する。

$$z_i = \begin{cases} 1 & (\text{グループ}i\text{がOWNER AREAを指定されているとき}) \\ 0 & (\text{逆に指定されていないとき}) \end{cases} \quad (\text{A.1})$$

●変数 $O_{ij}$ ,  $i \in R, j \in S$ を式(A.2)のように定義する。

$$O_{ij} = \begin{cases} 1 & (\text{グループ}i\text{がセット}j\text{のオーナーであるとき}) \\ 0 & (\text{逆にそうでないとき}) \end{cases} \quad (\text{A.2})$$

●変数 $M_{ij}$ ,  $i \in R, j \in S$ を式(A.3)のように定義する。

$$M_{ij} = \begin{cases} 1 & (\text{グループ}i\text{がセット}j\text{のメンバであるとき}) \\ 0 & (\text{逆にそうでないとき}) \end{cases} \quad (\text{A.3})$$

●変数 $Q_k$ ,  $k \in U$ を式(A.4)のように定義する。

$$Q_k = \begin{cases} 1 & (\text{セットオーダ}k\text{がOWNERポインタをもつとき}) \\ 0 & (\text{逆にもたないとき}) \end{cases} \quad (\text{A.4})$$

●変数 $P_k$ ,  $k \in U$ を式(A.5)のように定義する。

$$P_k = \begin{cases} 1 & (\text{セットオーダ}k\text{がPRIORポインタをもつとき}) \\ 0 & (\text{逆にもたないとき}) \end{cases} \quad (\text{A.5})$$

● $T_1$ : 2次記憶への平均順次アクセス時間 (秒)。

● $T_2$ : 2次記憶への平均直接アクセス時間 (秒)。

- $T'_2$ : バッファリング効果による2次記憶への実質的な平均直接アクセス時間(秒)。
- $f_2$ : CALCグループの検索に要する平均シーク回数。
- $KO_k, k \in U$ : セットオーダー  $k$  をもつセットのオーナーがもつポインタのサイズ(Byte)。
- $KM_k, k \in U$ : セットオーダー  $k$  をもつセットのメンバがもつポインタのサイズ(Byte)。
- $F_i, i \in R$ : グループ  $i$  のオカレンス件数。
- $E_j, j \in S$ : セット  $j$  の一つオーナーオカレンスに連結するメンバオカレンスの平均件数。
- $E(p_a), a \in A$ : エリア  $a$  の記憶利用効率。
- $z_i, i \in R$ : グループ  $i$  の一つオカレンスを記憶するのに要する領域の容量(Byte)。
- $h_i, i \in R$ : 1インデックスページ内のエントリの件数。
- $L_1$ : セットの最初のオカレンスを記憶するときのインデックスのレベル段数。
- $L_2$ : セットの最後のオカレンスを記憶するときのインデックスのレベル段数。
- $A_{i,j,q}, i \in R, j \in S, q \in Q$ : 問合せ  $q$  がセット  $j$  を経由し、グループ  $i$  を論理的セットアクセスする回数。
- $AA_{i,q}, i \in R, q \in Q$ : 問合せ  $q$  がグループ  $i$  を論理キーなしアクセスする回数。
- $AAA_{i,q}, i \in R, q \in Q$ : 問合せ  $q$  がグループ  $i$  を論理キー付きアクセスする回数。

[応用プログラムの実行時間] セット  $j \in S$  の一つのセットオカレンス内の全メンバオカレンスを探索するのに要する物理アクセスの平均所要時間  $PAT_j$  は、式(A.6)のようになる。

$$PAT_j = \sum_{i \in R} M_{i,j} \sum_{a \in A} W_{i,a} (X_{i,j} \{ \lceil Z_i E_j / \{ B (1 - 0.5 E(p_a)) \} \rceil - 1 \} T_1 + X_{i,m+1} E_j T'_2 + X_{i,m+2} E_j \{ (L_1 + L_2) / 2 + 1 \} T'_2)$$

(A.6)

従って、問合せ  $q \in Q$  の実行時間  $T_q$  は、式(A.7)のようになる。

$$T_q = \sum_{i \in R} \sum_{j \in S} A_{i,j,q} M_{i,j} \{ (1 - X_{i,m+2}) PAT_j + X_{i,m+2} E_j T'_2 \} + \sum_{i \in R} \sum_{j \in S} A_{i,j,q} C_{i,j} \sum_{k \in U} Y_{j,k} \{ C_k \{ X_{i,j} T_2 PAT_j / (PAT_j + T_1) + (1 - X_{i,j}) T'_2 \} + P_k (1 - C_k) \{ (1 - X_{i,m+2}) PAT_j / 4 + X_{i,m+2} \lceil E_j / 2 \rceil T'_2 / 2 \} \}$$

$$\begin{aligned}
& + (1 - P_k)(1 - Q_k) \{ (1 - X_{i,m+2}) PAT_j / 2 + X_{i,m+2} E_j T'_2 / 2 \} ] \\
& + \sum_{i \in R} AA_{i,q} \{ Z_i F_i T'_2 + (1 - Z_i) \sum_{a \in A} W_{i,a} p_a T_1 \} \\
& + \sum_{i \in R} AAA_{i,q} \{ (1 - X_{i,m+1})(1 - X_{i,m+2}) \{ Z_i F_i T'_2 / 2 \\
& + (1 - Z_i) \sum_{a \in A} W_{i,a} p_a T_1 / 2 \} \\
& + X_{i,m+1} \sum_{a \in A} \{ 1 + f_2(E(p_a)) \} T'_2 \\
& + X_{i,m+2} \{ \lceil \ln F_i / \ln h_i \rceil + 1 \} T'_2 ]
\end{aligned}$$

(A.7)

従って、応用プログラムの実行時間 $T$ は、式(A.8)のようになる。

$$T = \sum_{q \in Q} T_q \quad (A.8)$$

以上