異種データベースシステムの連携技術

に関する研究

池　田　哲　夫

異種データベースシステムの連携技術に関する研究

目次

図表

1. 序論

1.1. 研究の目的

データベース技術は，データベース技術出現以前にはデータ群がアプリケーションと括り付けで管理されていたのを，データ群をアプリケーションとは独立に統合的に蓄積・管理することにより以下のような利点をもたらすことを狙いに1960年代頃から研究が始まった技術である．

- データの冗長性を排除できる．組織内のデータが一元的にデータベースに格納されることにより，物理的な格納量を減らすことができる．
- データの一貫性の保証が容易になる．同一実体に対応するデータは一つだけであり，更新はそのデータにだけ行えばよく，一貫性の保証が容易になる．
- データの共有性を向上できる．データに対するアクセスインタフェースが共通化されるので，複数のアプリケーションによるデータの共有が容易になる．

特に1970年に，簡易な利用者インタフェースおよび明確な理論的基礎を特徴とする関係データモデルが出現したことにより，関係データモデルに基づくデータベース技術の研究および商用データベース管理システムの開発が活発に行われ，1980年代から1990年代にかけて関係データモデルに基づくデータベースシステムが急速に普及し，現在もその普及の勢いは衰えていない．

データベース技術は統合化の考え方に基づき発展してきたが，データベースシステムの普及と共に，既存の複数のデータベースシステムを連携して利用したいという要求が発生し，そのような要求に対処するための研究・開発も行われてきた．

ここで，既存の，互いに異種性を有する（即ち，データモデル，質問言語，データ項目の値の表現形式などが同一とは限らない）データベースシステムを連携して利用する形態の代表的なものに，連邦データベースシステム，企業アプリケーション統合（EAI：Enterprise Application Integration），データウェアハウス(DWH:Data Warehouse)がある（図1.1）．

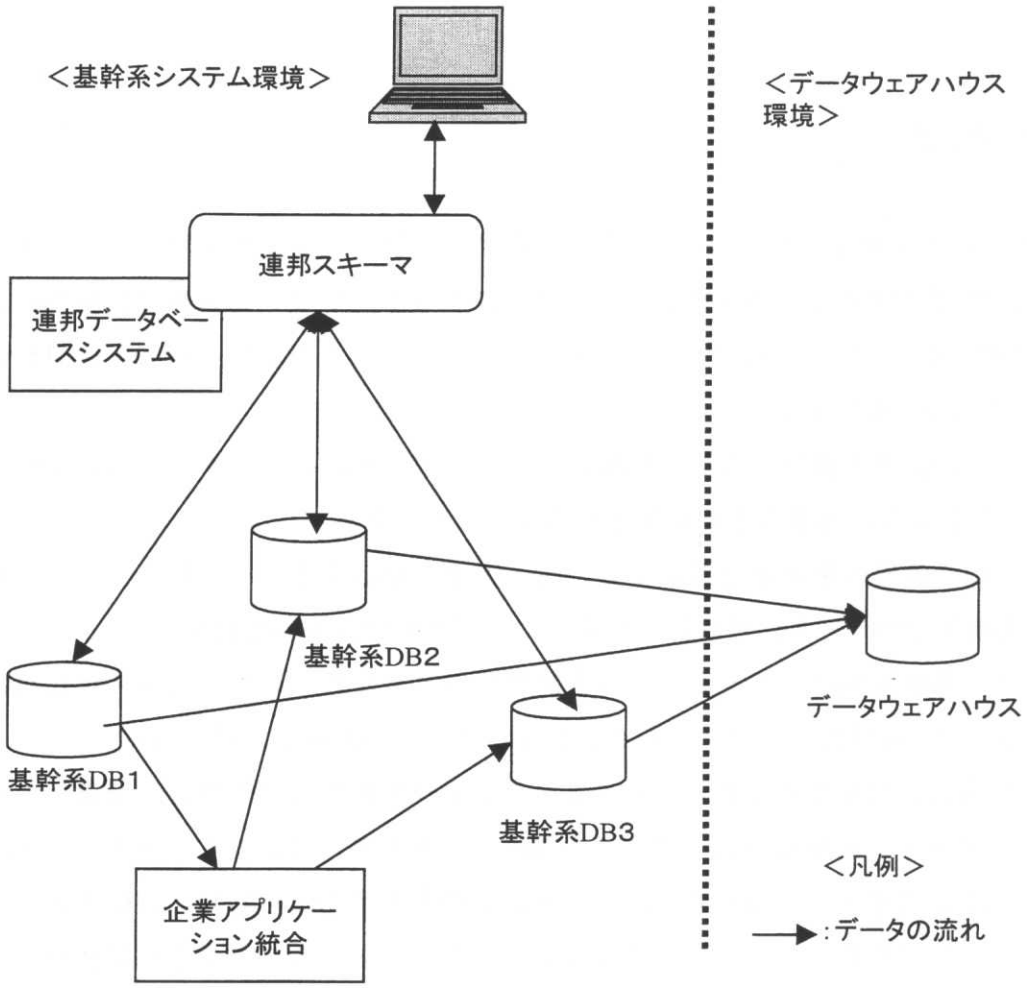


図 1.1　異種データベースシステムの連携の形態

連邦データベースシステムとは，複数のデータベースシステムのスキーマを収集・分析し，一つのスキーマ(連邦スキーマと呼ぶ)に論理的に統合し，利用者は連邦スキーマがあたかも通常のスキーマであるかのようにアクセスすることを可能にするシステムを言う．連邦データベースシステムの基本技術には，スキーマ（連邦スキーマ）の構築技術や，利用者からの質問を要素データベースシステム向けの質問に分解し，要素データベースシステムから返却された質問結果を統合して利用者に返却する技術（質問処理技術と呼ぶ）や，複数のトランザクションを矛盾無く並列走行させる技術（トランザクション同期制御技術と呼ぶ）などがある．

EAIとは，データベースシステムのデータとアプリケーションを統合する技術，また

はそれらの技術を用いて構築されたシステムを言う．EAIは，統合の対象となるものの種別に応じて４つのタイプに大別される．データレベルのEAIと，アプリケーションインタフェースレベルのEAIと，メソッド（共有および再利用可能なロジックをメソッドと呼ぶ）レベルのEAIと，ユーザインタフェースレベルの技術のEAIとである．それらのうち，異種データベースシステム連携に最も関連の深いものはデータレベルのEAIである．本論文ではデータレベルのEAIをデータ交換システムと呼ぶ．連邦データベースシステムとEAIとの違いは，連邦データベースシステムでは連邦スキーマを介した仮想的な連携を実現するのに対し，EAIではデータベースシステム間で直接的にデータを交換して連携を実現する点である．

データウェアハウスとは，意志決定支援用の大量データを格納したデータベースシステムを言う．データウェアハウスの基本技術には，ソースデータベースシステムのスキーマを統合してデータウェアハウス向けのスキーマを構築するスキーマ構築技術や，ソースデータベースシステムからデータを抽出・変換・格納するデータロード技術や，データウェアハウスのデータを様々な視点から分析する多次元分析技術や，データウェアハウスから新たな知識を発見するデータマイニング技術などがある．連邦データベースシステムやEAIとの違いは，連邦データベースシステムやEAIは，顧客管理や在庫管理などの組織の基幹業務を効率良く行うためのデータベースシステム（基幹業務系データベースシステムあるいは基幹系データベースシステムと呼ぶ）の連携を実現するものであるのに対し，データウェアハウスは基幹系データベースシステムからデータを抽出して構築されるものである点である．

近年，組織，特に企業はグローバルな競争に巻き込まれており，経営の効率化や経営戦略の変革を従来以上にスピーディに行うことを迫られている．このような状況のもとでは，連邦データベースシステムを用いて複数のデータベースシステムを連携して顧客管理，在庫管理などの基幹業務を効率化することや，EAI を用いて複数のデータベースシステムを連携して，データ入力作業の軽減，二重投入の回避，データ品質の向上，企業内を流れるデータの速度の向上などを可能とすることや，既存の基幹系データベースシステムからデータを収集してデータウェアハウスを構築して的確な経営上の意思決定を可能にすることなどをスピーディに実現できることが強く要求されるようになっている．

既存のデータベースシステムを連携することは一般に難度が高く，多くの稼動を要する作業であると認識されている．難度が高いものとなっている要因は大別して二つある．データベースシステムの自律性(autonomy)と，データベースシステム間に存在

する異種性(heterogeneity)とである．自律性とは，個々のデータベースシステムが，複数データベースを連携したシステムとは独立に自システムの設計や実行方法を決定できることを言う．自律性は，データモデル，質問言語などを独自に決定できる自律性である設計自律性と，通信をいつどのように行うかを決定できる自律性である通信自律性と，質問・コマンドの実行方法，実行順序を決定できる自律性である実行自律性とに大別される．異種性とは，データベースシステム間に存在する様々な差異を言う．代表的なものに，データモデルの違い，質問言語の違い，データ異種性などがある．データ異種性は，データの名称の違い，データの構造の違い（例：一つのデータベースシステムでは１データ項目で表現されるものが他のデータベースシステムでは複数データ項目で表現される），値の表現形式の違い（例：日付のyyyy/mm/ddとyy/mm/dd）などからなる．自律性を維持しつつ異種性を解消して，既存のデータベースシステムを効率的に連携する技術が求められている．

筆者はこのような状況を踏まえて，異種データベースシステム連携の基本技術の研究を研究課題として設定した.具体的には,ニーズの面からの重要性も高まっており，技術的にも重要と認識されている以下を研究課題として設定した.

・ 連邦データベースシステムにおけるスキーマの構築方式
・ 企業アプリケーション統合におけるデータ交換システムの構築方式
・ データウェアハウスにおけるオンラインデータローダの高速化方式

本研究の目的は，上記の課題を解決し，異種データベースシステム連携技術の確立に寄与することである．

本章では,以下,先ず異種データベースシステム連携技術に対するニーズを説明し，次いで従来の異種データベースシステム連携技術に関する研究の流れを説明し，次いで本研究における研究課題を説明し，最後にそれらの研究課題に対するアプローチを示す．

1.2. 異種データベースシステム連携技術のニーズの動向

組織，特に企業はグローバルな競争に巻き込まれており，従来以上に経営の効率化や経営戦略のスピーディな変革を実現することが求められている．

このような状況においては，以下に示すように，既存のデータベースシステムを連携するシステムに対するニーズが高まっており，従って既存のデータベースシステムを簡易に連携する技術に対するニーズも高まっているものと考える．

・ 顧客管理，事務処理，在庫管理などの基幹業務を効率化するために，複数の既存データベースシステムを要素データベースシステムとする連邦データベースシステムを構築する．
・ データ入力作業の軽減，二重投入の回避，データ品質の向上，企業内を流れるデータの速度の向上などを可能とするため，複数の既存データベースシステム及び既存アプリケーションを統合する企業アプリケーション統合（EAI：Enterprise Application Integration）を構築する．
・ 的確な経営上の意思決定を可能にするために，既存の基幹系データベースシステムからデータを収集してデータウェアハウス(DWH:Data Warehouse)を構築する．

既存のデータベースシステムを連携するシステムの構築に対するニーズの高まりを裏付けるデータとして，以下のようなデータが挙げられる．

・ イントラネットを導入する組織の増加が挙げられる［データベース白書00］．イントラネットとは，組織内の複数部門のデータベースシステムを接続するネットワークを言う．企業・団体におけるイントラネットの導入の割合が97年の調査では41.9%だったのが，99年の調査では63.3%に増加している．
・ EAIツールの売上額が99年の13億円から，2000年には74億円に増加すると見込まれている[日経システムプロバイダ00]．
・ データウェアハウスを導入する組織の増加が挙げられる［データベース白書00］．企業・団体におけるデータウェアハウスの導入の割合が97の年調査では15.0%だったのが，99年の調査では22.5%に増加している．

1.3. 従来の研究の動向

1.3.1. 異種データベースシステム連携の基本要件

異種データベースシステムを連携する際に配慮の必須な基本要件は次の二つである．データベースシステム間の様々な異種性を解消することと，データベースシステムの自律性を維持することとである．以下，それぞれについて説明する．

(1)異種性(heterogeneity)

個々のデータベースシステムは一般に相互に独立に設計・構築されたものであるため，データベースシステム間には様々な異種性が存在する[Sheth90]．

異種性には，データベースシステムに特に依存せず分散システム一般に見られる異種性と，データベースシステム固有の異種性とがある．前者には，ハードウェアの違い，オペレーティングシステムの違い，通信方法の違いなどがある．後者には，名称の違い，データ構造（例：一つのデータベースシステムではテーブルで表現されるものが他のデータベースシステムではデータ項目で表現される）の違い，値の表現形式（例：日付の yyyy/mm/dd と yy/mm/dd）の違いなどからなるデータ異種性と，データモデルの異種性と，質問言語の異種性，トランザクション同期制御方式の異種性などがある．異種データベースシステム連携技術の研究においては，主にデータベースシステム固有の異種性の解消法が研究されてきた．以下，データベースシステム固有の異種性について説明する．

データモデルには，関係データモデル(relational data model)[Codd70]に加えて，関係データモデル出現以前の代表的なデータモデルである，ネットワークデータモデル (network data model)[DDL73] や階層データモデル (hierarchical data model)[McGee77]や，関係データモデルよりも意味の表現に適しており，連邦データベースシステムのスキーマ構築における共通データモデルとして良く用いられる実体関連モデル(entity-relationship data model)[Chen76]や，オブジェクト指向モデル (object oriented model)[Cattell94]などがある．

質問言語は，データモデル毎に様々な質問言語が提案されている．代表的なものを挙げると，関係データモデルに対する SQL[Darwen97]，ネットワークデータモデルに対する NDL[JIS87]，オブジェクト指向モデルに対する OQL[Cattell94]などがある．

トランザクション同期制御方式には，資源にロックをかけることにより無矛盾な同期制御を保証する２相ロック方式(two phase locking method)[Eswaren76]，トランザクションに時刻印を付与し資源へのアクセス時に時刻印の大小で処理続行可能なトランザクションを決定することにより無矛盾な同期制御を保証する基本時刻印方式 (basic timestamp ordering method)[Bernstein80]，基本時刻印方式の拡張方式の一つであり読み込み時の性能向上を実現する多版時刻印方式(multiversion timestamp ordering method)[Reed83]，実行途中は他のトランザクションとの矛盾はないものとして処理を進め実行完了時に矛盾がなかったかを確認する楽観的方式(optimistic method)[Kung81],読みこみデータの集合と書きこみデータの集合が予め判明している場合にその情報を利用して効率的な同期制御を実現する先読みスケジューラ方式 (cautious scheduler method)[Katoh85]などがある．

表 1.1 データ異種性の分類

項番	異種性の分類 大分類	中分類	小分類	説明	Kimの分類との対応
1	スキーマ的異種性 注1)	1レコード対1レコード	名称の差異	同名異義あるいは異名同義.	I.A.1.a Table name conflicts
2			構成の差異	属性の多寡.	I.A.1.b Table structure
3		多レコード対多レコード		DBシステム間でレコードの対応が1対多あるいは多対多.	I.A.2 Many-to-many table conflicts
4		1データ項目対1データ項目	名称の差異	同名異義あるいは異名同義.	I.B.1.a. Attribute name conflicts
5			データ型の差異	例:CHAR対INTEGER	I.B.1.c.1 Data Type conflicts
6		複数データ項目対複数データ項目		DBシステム間でデータ項目の対応が1対多あるいは多対多.	I.B.2 Many-to-many attribute conflicts
7		レコード対データ項目		片方のDBシステムでのレコードが他方のDBシステムのデータ項目に対応.	I.C Table-versus-attribute
8	データ的異種性 注2)	誤データ	誤入力データ	(片方或いは両方のDBシステムのデータに)正しくない値が入力されたこと.	II.A.1 Incorrect - entry data
9			陳腐化データ	異なるDBシステムのデータ間において更新の同期がとれていないこと.	II.A.2 Obsolete data
10		値の表現方法の差異	異なる表現	例:会社名での正式名称対略称	II.B.1 Different expressions
11			異なる単位	例:長さでのm対cm	II.B.2 Different units
12			異なる精度	値域(domain)の基数が異なること. 例:重さでの整数表示対{重, 中, 軽}の3段階表示	II.B.3 Different precisions

注1)スキーマ定義文(SQLのスキーマ定義文)の記述の差異として表現できる異種性.

注2)スキーマ定義文では差異を記述できない異種性.

データ異種性の分類としては, Kim ら[Kim93]が関係データモデルにおけるデータ異種性の分類を提案している (表 1.1). Kim の分類は, 先ず, SQL のスキーマ定義文の記述の差異として表現できる異種性であるスキーマ的異種性と, スキーマ定義文では差異を記述できない異種性であるデータ的異種性とに, データ異種性を大分類する. 次いで, 各分類を詳細化している.

Sheth ら[Sheth92]は, オブジェクト指向データモデルにおける意味的類似性 (semantic proximity) の分類を提案し, さらに, 意味的類似性の分類と, 構造的な異種性(スキーマで記述できる異種性)の分類との関連を整理している.Garcia-Solaco

ら[Garcia-Solaco96]は，Kimの研究やShethらの研究を踏まえて，オブジェクト指向データモデルにおけるデータ異種性の分類を提案している．

(2)自律性(autonomy)

個々のデータベースシステムは，一般に異種データベースシステムを連携するシステムとは独立に設計・構築されたものであるため，異種データベースシステムを連携するシステムとは独立した制御下にある．従って，異種データベースシステムを連携するシステムの構築・運用においては，個々のデータベースシステムの自律性を維持する必要がある．

自律性は，設計自律性，通信自律性，実行自律性の三つに大別される[Sheth90] [Bukhres96b] [Bouguettaya98].

(a)設計自律性(design autonomy)

個々のデータベースシステムが，データモデル，質問言語，データ（テーブルやデータ項目）の名前付け，管理制約（整合性制約や，運用時間帯の制約や，更新ポリシーなど）などの設計項目について，何を用いるかを自主的に決定できることを言う．

(1)で説明した異種性は,主にデータベースシステムの設計の自律性から生じている.

(b)通信自律性(communication autonomy)

個々のデータベースシステムのデータベース管理システムが，他のデータベースシステム或いは異種データベースシステムを連携するシステムと何時どのように通信するかを自主的に決定できることを言う．

(c)実行自律性(execution autonomy)

個々のデータベースシステムのデータベース管理システムが，外部操作（異種データベースシステムを連携するシステムから発せられた操作）及びローカルな操作の実行方法，実行順序を自主的に決定できることを言う．

1.3.2. 連邦データベースシステムの研究動向

先ず，連邦データベースシステムの概要を説明し，次いで，連邦データベースシステムの基本技術のうち,スキーマ構築,質問処理の２つについて研究動向を説明する．

1.3.2.1. 連邦データベースシステムの概要

複数のデータベースシステムのスキーマを収集・分析し，一つのスキーマ(連邦スキーマと呼ぶ)に論理的に統合し,利用者は連邦スキーマがあたかも通常のスキーマであるかのようにアクセスすることを可能にするシステムを，一般に連邦データベースシステムと言う．利用者は物理的に分散されたデータの所在や各要素データベースシステム間に存在する様々な異種性を意識する必要がない．連邦データベースシステムの管理機能が，利用者からの質問を要素データベースシステム向けの質問に分解し，要素データベースシステムから返却された質問結果を統合して利用者に返却する機能（質問処理機能と呼ぶ）や，複数のトランザクションを矛盾無く並列走行させる機能（トランザクション同期制御機能と呼ぶ）などの機能を提供する．

ここで，本論文における連邦データベースシステムのモデルを説明する．モデルを設定する狙いは，論文を理解するための共通的な基盤を提供することにある．

モデルはスキーマと機能とからなる．それぞれについて説明する．

(1)スキーマ

スキーマとは，データベースシステム中のデータの構造，関連，各種の整合性制約などを記述したものである．関係データベースシステムの場合，スキーマは，表の定義やデータ項目の定義や整合性制約などからなる．

本論文での連邦データベースシステムのスキーマ構成としては,一般に良く知られ，連邦データベースシステムの適切なスキーマ構成と認識されているSheth ら[Sheth90]の５階層スキーマを採用する．

Sheth らのスキーマ構成の特徴は，集中型のスキーマ階層として良く知られているANSI/X3/SPARC[Tsichritzis78]の３階層スキーマを基に，それらを拡張して連邦データベースシステム向けのスキーマ階層を定義していることである.ANSI/X3/SPARCの３階層スキーマは，データベースシステムが全体としてモデル化している対象世界の論理的なデータ構造を記述したものである概念スキーマ(conceptual schema)と，記憶装置上に格納されたデータの物理的な構造を記述する内部スキーマ(internal schema)と,利用者/アプリケーションに対してデータベースシステムの一部を切り出したものである外部スキーマ(external schema)とからなる．

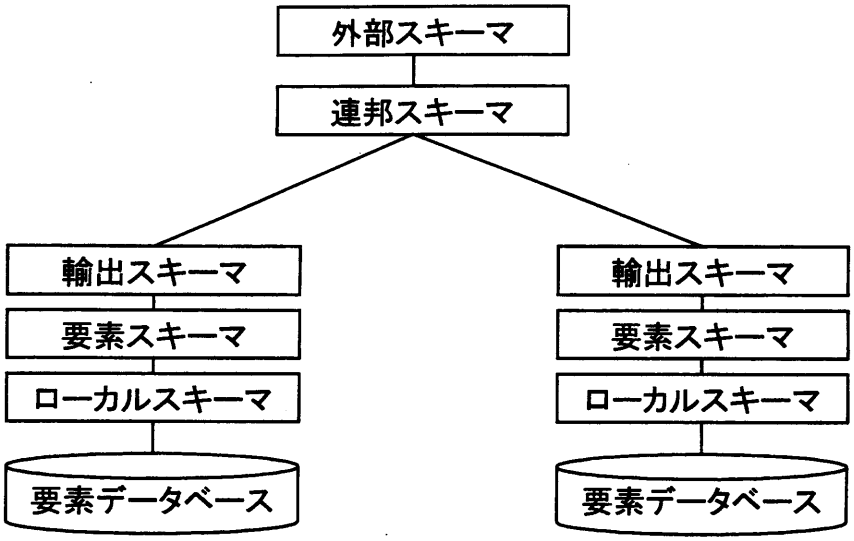


図 1.2 連邦データベースシステムのスキーマ構成

5階層スキーマを説明する（図1.2）.

- ローカルスキーマ(local schema)
  要素データベースシステムの概念スキーマである.
- 要素スキーマ(component schema)
  ローカルスキーマを共通データモデルで表現されたスキーマに変換したスキーマである．なお，共通データモデルとは，連邦スキーマの作成を行う際の作業用のデータモデルを言う．共通データモデルとしては，意味を表す能力の高いデータモデルが良く用いられる．例えば，実体関連モデルなどが良く用いられる.
- 輸出スキーマ(export schema)
  要素データベースシステムの管理者は，連邦データベースシステムを介して要素データベースシステムの全てをアクセス可とするわけではない．輸出スキーマは，連邦データベースシステムのために用いられる要素スキーマの一部である.
- 連邦スキーマ(federated schema)
  複数の輸出スキーマを統合したスキーマである.
- 外部スキーマ(external schema)
  外部スキーマは，利用者/アプリケーションに対するスキーマである.

(2)連邦データベースシステムの主要機能

連邦データベースシステムの有すべき主要機能に関して，標準的なものは存在しない．ここでは，研究の目的（異種データベースシステム連携技術の確立）を鑑み，従来研究[Sheth90] [Bouguettaya98] [Ozsu98]での主要機能のうち，連携の実現に関連の深い機能を，主要機能として挙げることとする．

(a)質問処理機能(query processing)

連邦データベースシステムに対して発せられた質問を複数の要素データベースシステム向けの質問に分割し，また，要素データベースシステムから受け取った質問結果を統合して利用者に返却する機能．質問分解の際に質問の実行性能が最高になるような最適化を行う機能も含む．

関連する研究動向の説明を1.3.2.3節で行う．

(b)トランザクションの同期制御機能(concurrency control)

複数の更新を含みうるトランザクションを，無矛盾性を保ちつつ，並列走行させる機能．

データウェアハウスのオンラインデータローダは本技術を用いて実現されることと，本論文の個別研究課題の一つとしてオンラインデータローダの高速化方式を設定したこととから，データウェアハウスの研究動向の説明の節（1.3.4節）において，関連する研究動向の説明を行う．

(c)スキーマ構築機能(schema integration)

輸出スキーマを収集・分析し連邦スキーマの作成を行うのを支援する機能．

関連する研究動向の説明を1.3.2.2節で行う．

次節以降では，連邦データベースシステムの主要な技術的課題の研究動向を説明する．

1.3.2.2. 連邦データベースシステムのスキーマ構築に関する研究動向

連邦データベースシステムのスキーマ構築は技術的難度の高い作業であり，連邦データベースシステムの主要技術的課題の一つと認識されている．

本論文では，先ずスキーマ構築のプロセスを分類し，次いでその分類に沿って研究動向を説明する．

(1)連邦データベースシステムのスキーマ構築プロセスの分類

Ramら[Ram98]の整理を基に，スキーマ構築の典型的なプロセスを整理する．

(a)スキーマ変換(schema translation)：以下のタスクからなる．

・ ローカルスキーマから共通データモデルのスキーマ（要素スキーマ）への変換．

・ 連邦データベースシステムに対してアクセスを許す部分(輸出スキーマ)の定義．

・ 輸出スキーマの収集．

(b)実体・データ項目間の関連特定(interschema relationship identification)：以下のタスクからなる．

・ 実体間，データ項目間，実体とデータ項目間の関連の特定．

(c)統合スキーマ作成(integrated schema generation)：以下のタスクからなる．

・ 特定された関連を基にしての，統合スキーマの作成．

・ 輸出スキーマと連邦スキーマの間のマッピングの整理

(2)構築プロセス対応の研究動向

(a)スキーマ変換

スキーマ変換における技術的難度が高いタスクは，ローカルスキーマを共通データモデルのスキーマ（要素スキーマ）に変換するタスクである．

ネットワークデータモデルで表現されたスキーマを関係データモデルで表現されたスキーマに変換する方法[Zaniolo79]や,実体関連モデルで表現されたスキーマを関係データモデルで表現されたスキーマに変換する方法[Teorey86]や，実体関連モデルで表現されたスキーマをネットワークデータモデル或いは関係データモデルで表現されたスキーマに変換する方法[Elmasri85]や,関係データモデルで表現されたスキーマをオブジェクト指向モデルで表現されたスキーマに変換する方法[Urban91]など,データモデル間において，スキーマを変換するための方法が提案されている．従来の代表的なデータモデル（ネットワークデータモデル，階層データモデル，実体関連モデル，関係データモデル，オブジェクト指向モデル）の間のスキーマ変換方法はほぼ確立したものと考える．

輸出スキーマの定義は，要素データベースシステムの管理者が連邦データベースシステムに対してアクセス可とする部分を自システムの設計方針に基づいて定義する作業であり，技術的難度が高いタスクではない．また，輸出スキーマの収集は，各要素データベースシステムの輸出スキーマを，スキーマ間の関連特定プロセス以降のプロセスを実施するサイトに収集する作業であり，技術的難度が高いタスクではない．

まとめると，スキーマ変換に関しては，方式の研究がほぼ確立しており，後続の構築プロセスであるスキーマ間の関連特定に比べて容易に作業可能となっていると考える．

(b) スキーマ間の関連特定

スキーマ間の関連特定において特定の対象となる関連の種類は，同義性(equivalence)およびその細分類である．

Navatheら[Navathe86]は実体間の同義の細分類として，等価(equal)，包含(contain)，オーバラップ(overlap)，互いに素(disjoint)，の４種類の関連を提案している．Larsonら[Larson89]はNavatheらの研究を拡張し，データ項目間の同義の細分類として同様な分類を提案している．Spaccapietraら[Spaccapietra94]は，Navatheらの研究とLarsonらの研究とが実体とデータ項目とが同義の場合（表1.1における「レコード対データ項目」異種性に対応）を扱っていないのに対し，実体とデータ項目が同義の場合も扱う拡張を行っている．

スキーマ間の関連特定において問題となるのは，関連を有するものの組み合わせを発見する作業の稼動が膨大になりうることである．二つの輸出スキーマ間の関連を特定する場合を考える．第１の輸出スキーマの実体（テーブル）数をn1，データ項目数をm1，第２の輸出スキーマの実体（テーブル）数をn2，データ項目数をm2とすると，関連有無を判断しなければならない実体の組み合わせの数，データ項目の組み合わせの数は，組み合わせの数を減らすための工夫を特に施さない場合，それぞれO(n1*n2)，O(m1*m2)になる．作業を効率化するための工夫が要求される．

完全自動化により効率化を図る方法が考えられるが，スキーマ間の関連特定を自動化することは一般に不可能であると認識されている[Sheth90]．なぜならば，スキーマ間の関連特定を自動化可能にするためには，事前に全実体，全データ項目の意味を与えておく必要があるが，それは現実的には不可能だからである．

従って，スキーマ間の関連特定を効率化するためには，関連を特定すべき実体の組み合わせあるいはデータ項目の組み合わせを絞り込む工夫が必要である．

本項では，以下，現実世界の汎用的な知識を用いることにより，組み合わせの数を絞り込むことをねらった研究を説明する．

Yuら[Yu91]は概念階層(concept hierarchies)を用いたデータ項目間の関連特定の支援方法を提案している．概念階層は特定の業務に依存しない汎用的な概念の階層(上位がより抽象的な概念を表し，下位がより具体的な概念を表す)として構築される．デ

ータ項目は，この概念階層を用いて以下のように特徴付けられる．この特徴づけは，概念階層の利用者が手作業で行うことを前提にしている．

・ システム中の概念数をｎとすると，ｎ次元ベクトルで特徴付けられる．
・ ベクトルの各次元の値は対応する概念がデータ項目にあてはまるならば１，あてはまらないならば０とする．

データ項目の近似度は，対応するベクトル間の近似度で計算される．近似度計算は自動化可能であり，例えば近似度が一定の閾値以下のものを，関連を有するデータ項目の候補として提示できる．

Brightらの研究[Bright91]は，名称の類似度に基づく実体・データ項目間の関連特定の支援方法を提案している．既存の汎用のシソーラスを基に用語のネットワークを構成する．ネットワークのリンクには用語間の関連の種類(類義語，上位語，下位語)に依存して重みが付けられている．名称間の類似度は，ネットワーク上での重み付きの距離として表される．距離が小さいほど類似度が大きいことを表す．類似度計算は自動化可能である．Brightらの類似実体・データ項目の検出処理においては，距離が閾値以下の用語が一つ見つかると処理が終了するが,距離が閾値以下の用語の集合を，関連を有する実体・データ項目の候補として提示できるように機能拡張することは容易であると考える．

Colletら [Collet91]はCarnotと呼ばれるスキーマ統合ツールを提案している.ツールの中心となるのは，CyCと呼ばれる知識ベースである．約50000の実体と関連とが格納されている.CyCは規範となる大局的スキーマとして機能する.スキーマ統合の最初のフェーズにおいて，要素データベースシステムのスキーマがCyCに対応付けされる.要素データベースシステムの方が豊富な実体或いはデータ項目を有している場合は，CyCを拡張する.実体・データ項目の対応付けの基本機構は，名称或いは名称の同義語との文字列一致である．名称或いは名称の同義語と文字列一致した実体・データ項目が関連を有するデータ項目の候補として提示される．利用者はそれらの候補の中から関連付けを行う他のデータ項目を手作業で決定する．

関連を有するものの組み合わせの発見支援方法に関して研究動向をまとめると，概念階層，用語ネットワーク，知識ベースなどの汎用的な知識を用いることにより，関連を特定すべき実体の組み合わせあるいはデータ項目の組み合わせを自動的に絞り込むことが可能になる．しかしながら以下の問題点があると考える．

・ (Yuらの研究に固有の問題) データ項目毎のｎ次元ベクトルの作成は，基本的には手作業で作成することを前提としており，概念数が多い実際の現場におい

ては大きな作業量が必要になると考える．半自動化の方式を提案しているが，方式の説明が詳細度に欠けており，その方式によってどの程度稼動が削減されるのか不明である．

- （Brightらの研究に固有の問題）複合語の名称を扱えない．
- （Colletらの研究に固有の問題）名称が複合語の場合，名称を形態素解析せずに比較しており，関連を有するデータ項目が候補として提示される確率が低いと思われる．

また，汎用的な知識を用いることをしない関連特定の研究においては，汎用的な知識を用いる場合と同程度の効率で関連を特定すべき実体の組み合わせあるいはデータ項目の組み合わせを絞り込むことは困難であるという問題があると考える．

ここで，Zhaoら[Zhao95]，Reckら[Reck96] [Reck97]は，検索インタフェースを簡易化することを主たる狙いとして，連邦スキーマにおいて普遍関係(universal relation)[Ullman89]を採用した連邦データベースシステムの研究を行っている．普遍関係とは，データベース中の全データが必ずその中に含まれる単一の(仮想的な)関係を言う．普遍関係を用いることにより，関係データベースシステムでの質問を簡単化すること，即ちデータ項目と関係の組を意識すること無くデータ項目だけを意識すれば質問可能とすることが可能になる．Zhaoらの研究とReckらの研究はいずれも方式提案に留まっており，彼らの提案を実装した場合の連邦スキーマ構築稼動は明らかではないが，普遍関係の導入により，実体間の関連特定作業，関連を特定された実体を基にしての統合スキーマの作成作業などが不要になり，スキーマ構築全体の稼動を削減することが期待できる．

まとめると，スキーマ間の関連特定に関しては，特定すべき関連の種類についての研究はほぼ収束している．しかしながら，関連を有するもの（実体・データ項目）の組み合わせを発見する方法については，効率的な方式が確立しているとは言い難い．また，実証はされていないものの，普遍関係の採用は，スキーマ間の関連特定を含むスキーマ構築作業全体の効率化に有効であると期待できる．

(c)統合スキーマ作成

先ず，特定された関連を基にして統合スキーマを作成するタスクに関する研究の動向を説明する．

Navatheら[Navathe86]は実体間に定義された関連の種類（等価，包含，オーバラップ，互いに素）を基に実体を統合する規則を提案している．Larsonら[Larson89]は

Navatheらの研究を拡張し,データ項目間に定義された関連の種類を基にデータ項目を統合する規則を提案している．Spaccapietraら[Spaccapietra94]は，実体とデータ項目が同義の場合に，それらを統合する規則を提案している．

統合スキーマ作成を半自動化するツールも報告されている．Sheth[Sheth88]らはNavatheらの統合規則を用いるインタラクティブな統合スキーマ作成ツールを報告し，Hayneら[Hayne90]は,Navatheらの統合規則を拡張した規則を用いるインタラクティブな統合スキーマ作成ツールを報告している．

Navatheら[Navathe86]，Larsonら，Spaccapietraらの研究においては，１つの関連種類に対応する規則は１つだけである．これに対して，彼らの提案する規則が常に利用者にとって最適の規則であるとは限らず，複数の規則を考慮する方が良いことを，Navatheら[Navathe96]やGellerら[Geller92]らが述べている．

Navatheら[Navathe86]，Larsonら，Spaccapietraらの研究においては，例えば，雇用者を表す実体「雇用者１」と「雇用者２」がオーバラップの関係にあり，「雇用者１」は「雇用者２」に無いデータ項目「血液型」を持ち，「雇用者２」は「雇用者１」に無いデータ項目「身長」を持つ場合，「雇用者１」と「雇用者２」に対して汎化（generalization）の関係にある新たな実体を作成する規則を提案している．これに対して，Navatheら[Navathe96]は，「雇用者１」と「雇用者２」とを削除し，「雇用者１」と「雇用者２」の共通データ項目とNULL可のデータ項目「血液型」と「身長」とを持つ実体を新たに作成する規則の方が利用者にとって使いやすい場合があると述べている．

Gellerら[Geller92]は，意味的には類似していなくても構造的に類似しているスキーマ（の部分）を統合することにより，利用者に有益な場合があると述べている．例えば，大学事務処理用の連邦データベースシステムにおいて，学生募集とその応募者に関するスキーマの部分と，求人情報とその応募者に関するスキーマの部分とは，意味的には全く異なるが，構造的には似ている．これらのスキーマの部分を統合することにより，スキーマの仕様の記述量の削減できる効果と，利用者の認知的負荷を軽減できる（一方のスキーマ部分の意味を理解できれば他方のスキーマ部分の意味を容易に理解できる）効果とが期待できると述べている．

次いで，特定された関連を基にして統合スキーマを作成するタスク以外のタスクに関する研究の動向を説明する．

輸出スキーマと統合スキーマ（連邦スキーマ）との間のマッピングの整理に関しては，前述のNavatheら，Larsonら，Spaccapietraらの研究において，統合規則の適用毎

に統合規則の適用対象となった輸出スキーマの実体・データ項目と統合スキーマの実体・データ項目との対応関係をスキーマに登録することにより容易に整理できることが述べられている.

まとめると，統合スキーマ作成に関しては，スキーマ間の関連特定プロセスで特定された関連の種類に応じた統合規則が提案されている．より適切な統合規則を提案する研究などが引き続き行われているものの，統合スキーマ作成方式はほぼ確立しており，前述のプロセスであるスキーマ間の関連特定に比べると，容易に作業可能になっていると考える.

1.3.2.3. 連邦データベースシステムの質問処理に関する研究動向

質問処理機能とは，連邦データベースシステムに対して発せられた質問を受理し，複数の要素データベースシステム向けの質問に分割し，要素データベースシステムに質問を発行し，要素データベースシステムから受け取った質問結果を統合して利用者に返却する機能を言う.

主要な技術的課題は，連邦データベースシステムの質問言語で記述された質問を要素データベースシステムの質問言語で記述された質問に変換する質問言語変換(query language translation)の方式と，実行時性能が最高になるように質問実行方式を決定する方法である最適化(query optimization)の方式とである.

両者は技術的難度が高くかつ実行時性能に大きな影響を及ぼす技術であると認識されており，連邦データベースシステムの主要な技術的課題であると考えられている.

(1)質問言語変換方式の研究動向

Rusinkiewiczら[Rusinkiewicz85]は，汎用の記号処理技法を用いて，関係質問言語（関係データモデルに基づく質問言語）の族の相互変換を行う方式を提案している.
Howellsら[Howells87]は,関係質問言語の族の変換を自動化する方式を提案している.
PROLOGで構築された変換システムが，一つの質問言語の仕様を入力として，他の質問言語の仕様を出力する.

Markowitzら[Markowitz93]は，関係データベースシステムをオブジェクト指向質問言語（オブジェクト指向データモデルに基づく質問言語）で質問する方式を提案している.その研究の中でオブジェクト指向質問言語からSQLへの変換方式を提案している.
予め関係データモデルで記述されたスキーマとオブジェクト指向データモデルで記述

されたスキーマとそれらの間のマッピング情報とを利用して変換する方式を提案している.

Czejdoら[Czejdo92]は，逆にオブジェクト指向質問言語を関係質問言語に変換する方式を提案している.Markowitzらの研究と同様にスキーマ間のマッピング情報を用いて質問言語間の変換を実現する方式を提案している.

Sybase社のDirectConnect/OmniConnect[Sybase00a]やIBM社のDataJoiner[IBM00a]などの商用の連邦データベースシステムにおいては，関係質問言語から，他の関係質問言語や階層型質問言語（階層型データモデルに基づく質問言語）などへの変換を，その実現方法の詳細は不明なものの，実現している.

まとめると，現時点での，研究および商用製品における代表的な質問言語の間の変換方法はほぼ確立しているものと考える.

(2)最適化方式の研究動向

実行時性能が最高になる質問実行方式を決定する方式，すなわち最適化方式は，集中型関係データベースシステムの研究においてSelingerら[Selinger79]が提案したコスト評価方式を用いる最適化方式が，その後の分散データベースシステムや連邦データベースシステムの最適化方式の研究においても主流である.

Lohmanらの研究[Lohman85] を基に，分散データベースシステムにおけるコスト評価方式の原理を説明する.

- 同一の質問の実行方式の候補を列挙する．例えば，関係データモデルの結合演算の実行方式には，二つのテーブルが同一要素データベースシステム内にある場合の方式として，二つのテーブルをソートしてから突合せるマージジョイン法(merge join method)と[Selinger79]と片側のテーブルのレコードにマッチする他方のテーブルのレコードをその都度フェッチするネステドループ法(nested loop method)[Selinger79]とがあり，二つのテーブルが異なる要素データベースシステム内にある場合の方式として，片側のテーブルの存在するサイトに他方のテーブルをまとめて送ってから結合する方式や片側のテーブルのレコードにマッチする他方のテーブルのレコードをその都度フェッチする方式などがある．質問が複数の演算から構成される場合は，単一の演算の実行方式のバリエーションと各演算の実行順序のバリエーションの組み合わせで様々な実行方式候補がありうる.
- 全ての実行方式候補について，スキーマが保持している統計情報（レコード数，データ項目の値の最大値，最小値など）を利用し，実行コスト(execution cost)

を計算する．コスト計算後に，コスト最小の候補を選択し，実行する．

コストの計算式としては例えば以下のような式を用いる．

総コスト＝

T_{CPU}*命令数+$T_{I/O}$*I/O数+T_{MSG}*メッセージ数+T_{TR}*転送バイト数

ここで，T_{CPU}は１命令あたりの平均CPU時間，$T_{I/O}$は１I/Oあたりの平均I/O時間，T_{MSG}は１メッセージの送受信に要する固定オーバヘッドの時間，T_{TR}は１バイトあたりの平均データ転送時間を表す．

連邦データベースシステムにおける最適化の研究としては，要素データベースシステムの自律性を考慮した研究，すなわち要素データベースシステムの性能関連情報をグローバルには完全には知ることができないということを考慮した研究が見られる．具体的には，要素データベースシステムの性能（コスト計算式）をどのように精確に推定するかに焦点をあてた研究が主に見られる．

Duら[Du92]は，コスト計算式が判明している要素データベースシステムと，コスト計算式が不明な要素データベースシステムとが混在する環境において後者のコスト計算式を実測値に基づき導くことにより全体的な最適化を行う方式を提案している．Adaliら[Adali96]は，最適化時点以前に要素データベースシステムにアクセスした際の性能情報をキャッシュしておき，そのキャッシュ情報を基に最適化を行う方式を提案している．

まとめると，連邦データベースシステムにおける最適化の研究は，コスト評価方式を用いる研究，特に要素データベースシステムの性能（コスト計算式）をどのように精確に推定するかに焦点をあてた研究が主に行われている．

1.3.3. 企業アプリケーション統合の研究動向

先ず，企業アプリケーション統合の概要を説明し，次いで企業アプリケーション統合における基本技術のうち，データ変換システムの研究動向について説明する．

1.3.3.1. 企業アプリケーション統合の概要

企業アプリケーション統合（EAI：Enterprise Application Integration）は，1990年代後半に生まれた新しい概念である．本節では，他の文献で参照されることの多いLinthicumの著書など[Linthicum99][タスクIT新書編集部00]を基に，EAIの概要を

説明する.

EAI とは，データベースのデータとアプリケーションを統合利用するための技術の体系，あるいはそれらの技術を用いて構築されたシステムを言う.

従来から，データ入力作業の軽減，二重入力の回避，データの品質の向上を主な狙いとする，データ統合の技術およびアプリケーション統合の技術は存在した．それらが，近年EAIの概念の下に包括されるようになったのは，以下の2つの要因による.

- 全社的な視点を持たずに，局所的な最適化を目的にデータ統合あるいはアプリケーション統合を繰り返していると，長い年月の後には，システム間の連携がスパゲティ状態になり，却って保守を難しくしてしまうことが認識されたこと.
- ERP (Enterprise Resource Planning) パッケージ（統合基幹業務パッケージとも呼ばれる),例えば,SAP社のR/3[SAP00]やPeopleSoft社のPeopleSoft[PeopleSoft 00]などの登場とその急速な普及により，経理，生産管理，販売管理，人事管理などの基幹業務を，コンピュータ・システムを使って密接に関係付けながら実行することが容易になった．ERPパッケージをも統合の対象とすることより，全社的な視点を持ってデータ統合やアプリケーション統合を実現することが非現実的ではなくなったこと.

EAI は，4つのタイプからなると言われる[Linthicum99].それぞれのタイプについて説明する（図1.3).

- データレベル
- アプリケーションインタフェースレベル
- メソッドレベル
- ユーザインタフェースレベル

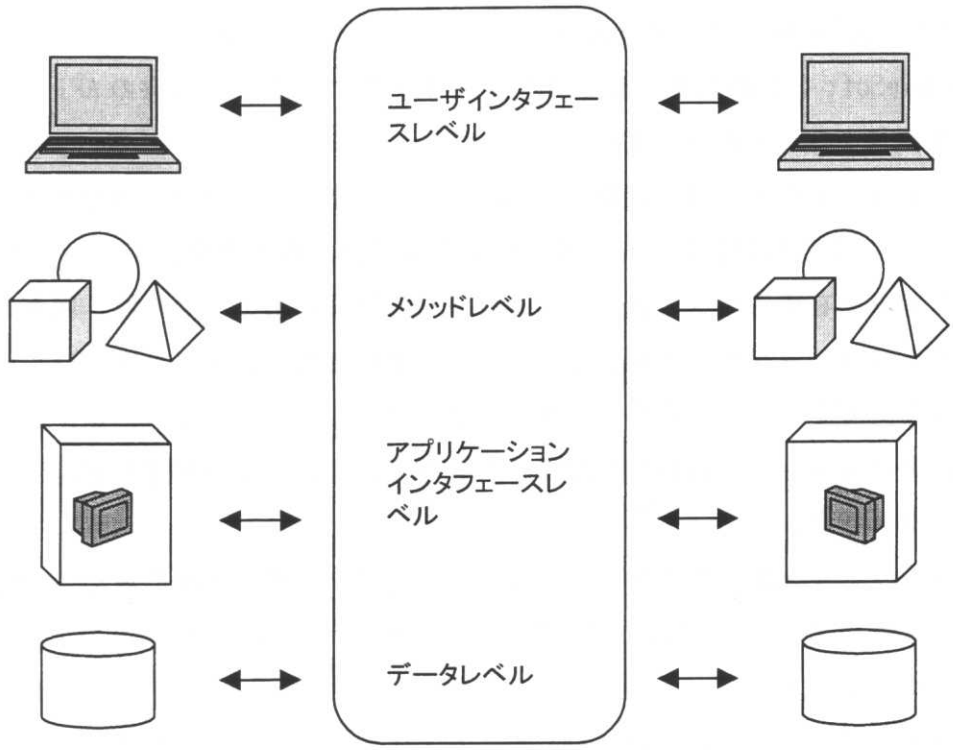


図 1.3　企業アプリケーション統合のタイプ

データレベルのEAIは，データベースシステム間でデータを交換する処理及び技術を言う．一つ以上の交換元のデータベースシステムからデータを抽出・変換し，一つ以上の交換先のデータベースシステムに格納することからなる．

[Linthicum99]においては，データベース間で直接データを交換し連携を図る形態に加えて，連邦データベースシステムを用いてデータベースの連携を図る形態も，データレベルのEAIに含めている．本論文においては，連邦データベースシステムに関しては別の節（1.3.2 節）で説明を行っていることから，データベース間で直接データを交換し連携を図る形態にのみ注目することとする．データレベルのEAIのうち，データベース間で直接データを交換し連携を図る形態をデータ交換システムと本論文では呼ぶこととする．

アプリケーションインタフェースレベルのEAIは，パッケージアプリケーションや自組織で開発したアプリケーションを，アプリケーションインタフェース（API: Application Programming Interface）を介して結合し，それらのアプリケーションが

ビジネスロジックとデータを共有できるようにする技術である.

ERP(Enterprise Resource Planning)パッケージ,例えばSAP社のR/3やPeopleSoft社のPeopleSoftなどの急速な普及に伴い,それらのパッケージ固有のAPIに接続可能な機能を提供することが特に重視されている.

メソッドレベルのEAIは，組織内において，ビジネスロジックの共有を可能にする技術を言う．例えば,顧客レコードを更新する処理は，顧客管理システムだけでなく，経理システムや在庫管理システムからも行いたい場合がある．この時，顧客レコードを更新するメソッドが共有可能になっていれば便利であり，メソッドレベルのEAIはそのような共有を可能にするものである．ここで言うメソッドとは，オブジェクト指向技術におけるメソッドの概念を拡張したもので，共有及び再利用が可能になっているロジックを指す.

アプリケーション間でのメソッドの共有を可能にする機構は既にいくつか存在する.例えば，分散オブジェクト技術，アプリケーションサーバ，トランザクション処理モニタなどがある.

ユーザインタフェースレベルのEAIは，アプリケーションの結合において，ユーザインタフェースをも統一的なインタフェースの使用を可能にする技術である.例えば,メインフレームのアプリケーションとクライアント・サーバ型のアプリケーションとでは，通常データのアクセス方法や処理の方法が異なる．このような処理を統一的な環境で実現しようというのが，ユーザインタフェースレベルのEAIである.

異種データベースシステム連携に最も関連の深い技術は，４つのタイプのうち，データレベルのEAI（データ交換システム）である．そこで，本節では，以下データ交換システムの研究動向について説明する.

1.3.3.2. データ交換システムの研究動向

データ交換システムの研究動向を説明する.

異種データベースシステム連携の基本要件である異種性解消，自律性維持に着目して研究動向を説明する.

異種性解消の観点から対処の必要な課題は，データ異種性の解消である．データ変換システムは，連邦データベースシステムとは異なり，データベースシステム間で直接データを交換することにより連携を実現する.このため,(連邦データベースシステ

ムでは解消が必要となる）質問言語の異種性やトランザクション同期制御方式の異種性などを解消することは，データ交換システムの課題とはならない．

Hammer ら[Hammer91]は，EXTRACTと呼ぶデータ交換システムの開発・実行環境を提案している．

データ交換の定義は大きく二つのステップに分かれる．交換元/交換先のデータベースシステムの登録，登録されたデータベースシステム間での交換方法の定義の２ステップである．データベースシステムの登録は，データベースシステムからの/への入出力プログラムのテンプレートの作成などプログラムを作成する(EXTRACTにおいては文法の作成と呼んでいる）ことなどにより行う．交換方法の定義はGUI上での簡単な操作（レコードやデータ項目の選択や，データ項目値変換関数の選択)で可能としている．

データ異種性解消の観点からは，データ項目の対応付けの仕組みや，データ項目値変換関数のライブラリが提供されていることから，解決可能なデータ異種性の範囲は明確ではないものの，Kimの分類におけるスキーマ的異種性の大部分とデータ的異種性のうちの値の表現方法の差異の大部分とが解消可能と思われる．また，誤データ異種性（入力誤りや入力遅延に起因して発生するデータの不整合）のうち入力遅延に起因する不整合に関しては，データ交換を行うこと自体で解消されるものと考える．

IBM社のDataPropagator[Bontempo98] [IBM00b]は，DB2を始めとする関係データベースシステムや，階層型データベースシステムIMSや，ファイルシステムVSAMから，DB2を始めとする関係データベースシステムへのデータ交換を行うツールである．ユーザフレンドリなインタフェースを用いて，交換元データベースシステムの指定，交換先データベースシステムの指定，データ交換の頻度・契機の指定，変換処理の指定を行う．変換処理には，表を用いてコード化された項目値を変換する処理や，ストアドプロシジャを定義しての変換処理などが含まれる．

データ異種性解消の観点からは，コード化された項目値を変換する処理や，ストアドプロシジャを定義しての変換処理が可能なことから，解決可能なデータ異種性の範囲は明確ではないものの，Kimの分類におけるスキーマ的異種性の大部分とデータ的異種性のうちの値の表現方法の差異の大部分とが解消可能と思われる．誤データ異種性のうち入力遅延に起因する不整合に関しては，データ交換を行うこと自体で解消されるものと考える．

データウェアハウス構築用のデータ交換システム（データローダと呼ぶ．1.3.4節参照）においては，交換元の入力誤りを検出し修正すること（すなわち，誤データ異種性のうち入力誤りに起因する不整合を解消すること）の重要性が強く認識され

[Burch97],交換元の入力誤りを検出し修正する機能の研究･開発が多くなされている．データウェアハウス構築時用の入力誤りの検出および修正機能を，データクリーニング(data cleaning)あるいはデータクレンジング(data cleansing)と呼ぶ．

Hernandezら [Hernandez98]は,知識ベースシステムを用いたデータクリーニングの方式を提案している．レコードのキーの等価性に関する規則を予め知識ベースに登録する．レコードをキー値のアルファベット順にソートした後，近傍のレコードの等価性を，登録されたキーの等価性規則に基づき判断する．等価なレコードの見落としの確率を低くするために，複数のキー候補に関して等価性規則を登録し，ソートおよび判断を複数パス実行させるようにしている．

Hinrichsら[Hinrichs99]は，ガンの疫学調査記録(epidemiological cancer registry)のためのデータクリーニングシステムを提案している.疫学調査記録の構築には，複数の情報源（病院，検査機関など）からのレコードを，既存レコードとの間の不整合を除去してから登録することが必要であり，知識ベースシステムを用いたデータクリーニングの方式を提案している．どのような値が両立し得るかなどを記述した規則を予め登録しておき,新たなレコードが入力された時に,既存の患者レコード,腫瘍レコードと照合して不整合の有無をチェックする．

Vality社のIntegrity[Vality00] [Bontempo98]は，データクリーニング用のツールであり，知識ベース，変換テーブル，パターンマッチング，統計解析などの多様な手段を利用して，利用者がデータクリーニングを行うことを可能にしている．

まとめるとデータ異種性に関しては,Kimの分類でのデータ異種性についてほぼ網羅的に解消の方式が提案されている．特に，入力誤りに起因する不整合の解消に関しては，データクリーニング関連の研究において，様々な方式が提案されている．

自律性維持の観点からの対処の必要な主要な課題は,交換元/交換先データベースシステムにおける既存機能や性能への影響の極小化である．

IBM社のDataPropagatorやSybase社のReplication Server[Sybase00b]は，データベースの複製データを作成し交換する機能を提供するシステムである．それらのシステムにおいては，データベース処理のログ（障害回復用のデータ．更新操作やトランザクション完了の履歴を含む.）を利用してデータ交換用データ（複製データ）を作成する機能を設けることにより，交換元システムの既存機能および性能への影響の極小化を図っている．

Zhugeら[Zhuge95][Zhuge96]は,交換元のデータベースの変更データを交換し交換先

のデータウェアハウスでビューを実体化する方式を提案している.Zhugeらの方式において，交換元のデータベースシステムに要求される条件は，変更データに時刻印を付与することと，データウェアハウスからの質問に回答する能力を有することだけである.

ここで，上記研究とは逆のケース，即ち，交換先のデータベースシステムの既存機能や性能への影響を極小化しつつデータ交換を行う必要があるケースが考えられる.典型的なケースとして，交換先のデータベースシステムがレガシーシステム[Brodie95]であり，そのため既存機能の更改が困難であり，データ交換システムの構築時に,交換先のデータ入力方法を改造無しに用いることを要求されるケースがある.既存データ入力方法には，下記に挙げる制約を有するものがしばしば見られる.

・ 複数関連データの一括入力

特定の関連を有する，一般には可変個の関連データをまとめて入力しなければならないという制約.

例えば，参照関係の基数が1対多（多は可変とする）という意味制約を有するレコードA，Bを考える．交換元の1トランザクションで1つのAと複数（可変個）のBを作成可能であり，交換先ではそれらの関連データをまとめて入力しなければならないという制約が考えられる.

この課題に対する解決法を与えた研究は見当たらない．既存の研究におけるデータ定義能力は関係データベースシステムのデータ定義機能（繰返しを含まない，フラットなデータ構造の定義機能）を大きく超えるものでは無く，また，既存の研究におけるデータ変換能力は，関係データベースシステムの関係演算機能（選択，結合など）と，データ型変換機能や対応表変換機能等のデータ項目レベル変換機能とを合わせた機能を大きく超えるものでは無い．従って，既存の研究においては，この要求条件に応えるために必要な，可変回繰返し構造体を含むデータ構造の定義，可変回繰返し構造体を含むデータ構造の作成，可変回繰返し構造体間のデータ変換，などの機能は有していない.

この課題に対する解決法としては，個別にプログラミングすることによって構築されたデータ交換システムが見られるだけである．それらのデータ交換システムにおいては，プログラムロジックによって交換先のデータベースシステムでの入力方法への影響の極小化を実現している．それらのデータ交換システムは汎用言語でプログラミングされていることから，構築の稼動・期間が大であるという問題がある.

まとめると，自律性維持に関しては，交換元データベースシステムへの影響の極小

化に関して様々な解決の方式が提案・実現されている．しかしながら，交換先データベースシステムへの影響を極小化しつつデータ交換を可能とすることに関しては，解決法を与える研究が見当たらず，また，個別プログラミングによる解決策には構築稼動大という問題がある．該課題の解決法およびその解決法を備えたデータ交換システムの簡易な構築方式が望まれる．

1.3.4. データウェアハウスの研究動向

先ず，データウェアハウスの概要を説明し，次いでデータウェアハウスの基本技術のうち，データロード技術について研究動向を説明する．

1.3.4.1. データウェアハウスの概要

データウェアハウスは1990年代に生まれた概念である．本節では，データウェアハウス(Data Warehouse)の概要について，データウェアハウスの提唱者であるW.H.Inmonの著書など [Inmon96][Inmon99] [Jarke00]を基に説明する．

データウェアハウスとは，意志決定支援業務向けのデータベースシステムである．1990年代に入り，意志決定支援業務向けのデータベースシステムは従来のデータベースシステムと異なる特性を有しており，その実現のためには従来のデータベース技術を発展させて新たな技術が必要であるとの認識が生まれた．その認識の基にデータウェアハウスの概念が提唱されるようになった．

従来のデータベースシステム，即ち顧客管理や在庫管理などの組織の基幹業務を効率良く行うためのデータベースシステム（基幹業務系データベースシステムあるいは基幹系データベースシステムと呼ぶ）と比較した場合のデータウェアハウスの特性は４つあると一般に言われている．それぞれの特性について簡単に説明する．

(1)サブジェクト（主題）指向

データウェアハウスは，従来の基幹系システムのアプリケーションから独立して，企業における主要な主題（例：顧客，製品）に沿って，データモデル化を行う．

(2)統合

基幹系データベースシステムのデータをデータウェアハウスにロードする際に，データが統合される．統合は，アプリケーションによって異なる，データ項目の名前の付け方，データ体系（データ型と精度），コード体系などを統一的なものにする．

(3)時系列

基幹系データベースシステムのデータは一般にデータの現在値のみを示すが，データウェアハウスのデータは時系列である．すなわち，データウェアハウスはある時点の値を示すスナップショットデータを時間軸に沿って保持する．

(4)不変性

データウェアハウスに対する操作は検索と新たなスナップショットデータの追加だけであり，データウェアハウスに蓄積されたスナップショットデータの内容が変更されることはない．

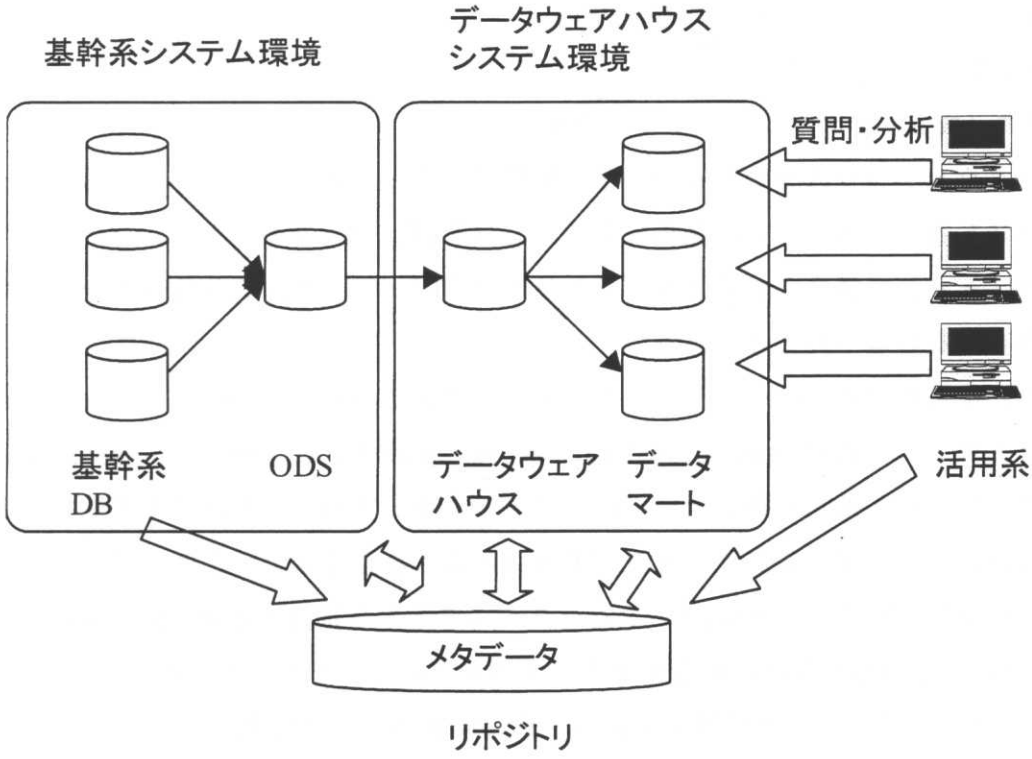


図 1.4　データウェアハウスの構成

データウェアハウスの構成を次に説明する（図1.4）．

データウェアハウスは，基幹系データベースシステムからデータを抽出・変換・統合し，物理的に別のデータベースシステムとして構成する．データウェアハウスは，全社レベルのデータウェアハウスであるグローバルデータウェアハウス（セントラル

データウェアハウス，エンタープライズデータウェアハウス，コーポレートデータウェアハウス，または単にデータウェアハウスとも呼ばれる）と，部門レベルのデータウェアハウスであるローカルデータウェアハウス（データマート（Datamart）とも呼ばれる）の2階層からなる．

データウェアハウスとデータマートの特徴を以下に記す．

(1)データウェアハウスの特徴

・ 全社レベルのデータ共用

・ 大規模データウェアハウス

・ 多数のサブジェクトから構成

(2)データマートの特徴

・ 部門/エンドユーザレベルの個別共用

・ 小規模データウェアハウス

・ 少数（数個）のサブジェクトから構成

データウェアハウスとデータマートの関係は，データモデルでいうところの「データベース」と「ビュー」との関係になる．しばしば，データマートは，マテリアライズドビューとして構成される．マテリアライズドビューとは，具体化された，実体として存在するビューのことを言う．

また，基幹系データベースシステムとデータウェアハウスとの間にオペレーショナルデータストア（ODS：Operational Data Store）を設けることがある[Inmon99]．ODSは，サブジェクト指向と統合という点ではデータウェアハウスと共通の特性を有するが，現在値のみを有する点とデータが更新されるうる点ではデータウェアハウスとは特性が異なる．ODSは，基幹系データベースシステムから抽出・変換・統合されたレコードを有する．ODSを設ける典型的なケースは以下のようなケースである．

・ データウェアハウスの構築・更新は一般に時間のかかる作業である．遅延の少ない意志決定支援を可能にするためにODSを設ける．

メタデータは，データウェアハウスのデータの定義情報であり，リポジトリと呼ばれるデータベースに格納される．データソース（基幹系データベースシステム）の定義情報，ODSのスキーマ情報，データウェアハウスのスキーマ情報，データマートのスキーマ情報，ODS・データウェアハウス間のマッピング情報，データウェアハウス・データマート間のマッピング情報などからなる．

活用系は，データウェアハウスを意志決定支援に利用するために，メタデータを参照して，データの検索や分析，レポート作成を行う計算機環境である．

なお，図1.4は典型的な構成を示すものであり，必ずしも３つの構成要素（データウェアハウス，データマート，ODS）全てが揃っていなければならないことを示すものではない．３つの構成要素のうち，１つないし2つだけが存在する構成もありうる．

次にデータウェアハウスの構築について説明する．

データウェアハウスの構築は大きく２つの処理に分けられる．スキーマ統合と，データロードである．

スキーマ統合では，情報のソースとなる基幹系データベースシステムのスキーマを分析・統合して，データウェアハウスのスキーマ（または，ODSやデータマートのスキーマ）を構築する．このスキーマを統合スキーマと本論文では呼ぶこととする．なお，スキーマ統合の技術は，連邦データベースシステムのスキーマ構築の技術と本質的に変わるものではないことが認識されている[Jarke00]．

データロードでは，基幹系データベースシステムからデータウェアハウスに実際のデータを抽出・格納する．データロードに並行して，データ体系（データ型と精度）やコード体系を統一するためのデータ変換を行ったり，誤りデータを検出・訂正するデータクリーニングを行ったりする．

次にデータウェアハウスの活用技術,すなわち意志決定支援技術について説明する．

意志決定の支援技術は，大きく検証型と発見型に分けられる．それぞれについて簡単に説明する．

(1)検証型

検証型は，事前に仮説を立てて，データウェアハウスからデータを引き出し，その仮説の検証を行う意志決定の手法である．仮説とは，ビジネスでの意志決定を行う上で意志決定の根拠となる知識や規則を言う．例えば,「ある製品の売上には，～なる地域特性がある」,「ある販売チャネルにおけるある製品の売上の年度毎の変化には，～なる特性がある」などが仮説の例である．検証型の手法としては，通常の質問機能と多次元分析とがある．多次元分析とは，複数の属性項目（次元）を次々に切り替えてデータを分析する手法を言う．

(2)発見型

発見型は，事前に仮説を立てずに，データウェアハウスから未だ知られていない新たな仮説，すなわち新たな知識を発見する意志決定支援の手法である．発見型の技法としては，データマイニング(data mining)がある．マイニングとは採掘するという

意味であり、データウェアハウスから新たな有用な知識を発見する手法を言う．

異種データベースシステム連携の観点から最も基本的な技術は，スキーマ統合とデータロードである．スキーマ統合に関しては，連邦データベースシステムのスキーマ構築技術と等価であると認識されていることと，連邦データベースシステムのスキーマ構築技術の研究動向は1.3.2節で説明済みであることとから,本節での研究動向の説明は省略する．本節では，これ以降データロード技術の研究動向について説明する．

1.3.4.2. データロードの研究動向

データロード技術の研究動向を説明する．

ソースデータベースシステム（基幹系データベースシステム）からデータウェアハウスへデータロードを行うプログラムをデータローダまたは単にローダと呼ぶ．

データローダは，その運用形態からオフラインデータローダとオンラインデータローダとに分類される．オフラインデータローダは，基幹系データベースシステムにおける通常のトランザクション処理とは非同期に，運用時間帯を別にするなどして，データのロードを行う．オンラインデータローダは，トランザクション技術を用いて，基幹系データベースシステムにおける通常のトランザクションと同期してデータのロードを行う．

データウェアハウスでは，通常大量のデータの検索を必要とすることから，データロード処理と通常のトランザクションとを共存させることは性能的にしばしば問題となりうる．従って，オフラインデータローダが使われることが一般には多い．オンラインデータローダが使用されるのは,ODSを用いて鮮度の高い情報を基にした意志決定支援を行いたい場合などである．それぞれについて，研究動向を説明する．

1.3.4.3. オフラインデータローダの研究動向

オフラインデータローダに関する研究動向を説明する

以下を考慮すると，オフラインデータローダは，機能的にはEAIにおけるデータ交換システムとほぼ等価であると言えると考える．

・ オフラインデータローダもEAIにおけるデータ交換システムも，１つ以上のデータベースシステムからデータを抽出・変換し，１つ以上の格納先（データウェアハ

ウスあるいはデータベースシステム）にそれらのデータを格納する技術である．

・ データウェアハウスは，格納時のインタフェースという観点からは通常のデータベースシステムと違いは無い．

データ交換システムに関する研究動向は，既に1.3.3節で説明済みであることから，オフラインデータローダ技術の説明は省略する．

1.3.4.4. オンラインデータローダに関する研究動向

オンラインデータローダに関する研究動向を説明する．

オンラインデータローダは，トランザクション技術を用いて実現される．トランザクション技術は，データウェアハウス出現以前から，データベース研究の一環として研究されてきた．

そこで，本節では，先ずデータベース研究におけるトランザクション技術の研究動向について説明し，次いでデータウェアハウス向けのトランザクション技術の動向について説明する．

(1)データベース研究におけるトランザクション研究の動向

複数のトランザクションを矛盾が発生しないように並列走行させる方式であるトランザクション同期制御方式は，技術的難度が高くかつ実行時性能に大きな影響を及ぼす技術であると認識されており，データベースシステムの主要な技術的課題であると考えられている．

並列走行させるトランザクション同士の競合や，各種障害の発生に起因して，データベースの矛盾が発生することのないように，（トランザクション同期制御機能をその一部として含む）トランザクション処理においては，ACID特性と呼ばれる以下の性質が成り立つことを保証する必要がある[Haerder83]．

・ 原子性(Atomicity)：データベースに対する操作が全て有効であるか，或いは全て無効であるかのいずれかであること．

・ 無矛盾性(Consistency)：無矛盾な状態のデータベースにおいてトランザクションを実行した場合，トランザクション実行後のデータベースの状態も無矛盾であること．

・ 分離性(Isolation)：トランザクションがあたかも他のトランザクションからの干渉がなかったかのように実行されること．

・ 耐障害性(Durability)：コミットされたトランザクションの効果が，データベース中で，障害などに起因して消滅することのないこと．

同期制御方式は，ACID特性のうちの主に分離性を保証するための機構である．同期制御が正しいことの基準（正当性基準と呼ぶ）としては，通常，直列可能性（serializability. 複数のトランザクションの並列走行の効果が，複数のトランザクションをある全順序に従って走行させた場合の効果に等しくなること)が用いられる．

以下，複数の要素データベースシステムに処理がまたがるトランザクションをグローバルトランザクション(global transaction)と呼び，グローバルトランザクションを一定の基準（例：一要素データベースシステムに閉じた部分）に従って分割したものをサブトランザクション(subtransaction)と呼ぶ．また，処理が1要素データベースシステムに閉じるトランザクションをローカルトランザクション(local transaction)と呼ぶ．

集中型データベースシステム或いは分散データベースシステム（複数のデータベースが共存するシステムのうち，それらのデータベースを管理するデータベース管理システムが全て同質のシステムを分散データベースシステムと呼ぶ）の同期制御方式としては異種性の説明でも述べたように，資源にロックをかけることにより正当な同期制御を保証する2相ロック方式(two phase locking method)[Eswaren76]，トランザクションに時刻印を付与し資源へのアクセス時に時刻印の大小で処理続行可能なトランザクションを決定することにより正当な同期制御を保証する基本時刻印方式(basic timestamp ordering method)[Bernstein80]，基本時刻印方式の拡張方式の一つであり読み込み時の性能向上を実現する多版時刻印方式(multiversion timestamp ordering method)[Reed83]，実行途中は他のトランザクションとの矛盾はないものとして処理を進め実行完了時に矛盾がなかったかを確認する楽観的方式(optimistic method)[Kung81],読みこみデータの集合と書きこみデータの集合が予め判明している場合にその情報を利用して効率的な同期制御を実現する先読みスケジューラ方式(cautious scheduler method)[Katoh85]などがある．

連邦データベースシステム向けのトランザクション同期制御方式の研究は，集中型データベースシステム或いは分散データベースシステムの同期制御方式の研究をベースにして行われた．それらの研究は3種類に大別される．一つ目の分類の研究は，集中型データベースシステム或いは分散データベースシステムでの同期制御方式と同様に直列可能性を正当性基準とし，その正当性基準を満たす同期制御方式を研究するものである．二つめの分類は，直列可能性を緩和した正当性基準を設定し，そのもとで

の同期制御方式を研究するものである．三つめの分類は，二つめの分類までが汎用的な環境での適用をねらうものであったのに対し，適用環境或いは適用アプリケーションを限定し，通常のトランザクションモデルとは異なるトランザクションモデルを導入することなどにより，高性能化を狙うものである．二つめ，三つめの分類の研究が行われている理由，特に三つめの分類の研究が行われている理由は，直列可能性を正当性基準として採用する条件のもとでは，トランザクション同期制御方式の高性能化には限界があると認識されていること[Sheth90]にある．

一つめの分類の研究動向を説明する．

Georgakopoulos らの研究[Georgakopoulos91]では，チケット(ticket)と呼ばれるデータを用いて強制的にグローバルトランザクション間の競合を発生させる．グローバルトランザクションは必ずチケット値を読み，加算し，要素データベースシステムに値を書きこまなければならない．こうすることによって，グローバルトランザクションとローカルトランザクション間の直列可能性は，ローカルデータベース管理システムの同期制御機能で制御可能となる．グローバルな制御機構は，全要素データベースシステムにまたがるグローバルトランザクションの直列可能性を確認しさえすればよい．全要素データベースシステムにまたがるグローバルトランザクションの直列可能性は，グローバルトランザクションのサブトランザクションから返却されるチケットの値を用いることにより判定可能である．Breitbart らの研究[Breitbart91]は，ローカルデータベース管理システムの同期制御方式が rigorous なる性質を満たせば，一般には直列可能性を保証しないグローバルな同期制御方式を用いても直列可能性の保証が可能になりうることを示している．同期制御方式が rigorous であるとは，あるトランザクションが読みこんだデータはそのトランザクションがコミット或いはアボートされるまでは他のトランザクションによる書きこみを禁止する方式であることを言う．

二つめの分類の研究動向を説明する．

Ramamritham ら[Ramamritham96]は，cooperative 直列可能性を提案している．cooperative 直列可能性とは，通常の直列可能性が個々のトランザクション同士の競合関係を基に決定されるのに対して，ある特定の集合（cooperative トランザクション集合と呼ぶ）と他の個々のトランザクションとの間の競合関係を基に決定される直列可能性である．Mehrotra ら[Mehrotra92]は m-直列可能性を提案している．m-直列可能性は，redo（やり直し）トランザクションを用いて２フェーズコミットプロトコルを疑似する環境での使用を目的に提案された正当性基準である．サブトランザクションがアボートされたにもかかわらず，全体の決定がコミットだった場合は，redo トラ

ンザクションが実行される．各要素データベースシステムにおいて，全ての他のトランザクションが集合｛サブトランザクション，そのサブトランザクションのredoトランザクション｝に対して直列化可能な場合，データベースの無矛盾性が保証される．m-直列可能性は，cooperative直列可能性の一形態とみなすことができる．

三つめの分類の研究動向を説明する．

CAD (Computer Aided Design)・CAM (Computer Aided Manufacturing) や，ソフトウェア設計など長時間トランザクションが必要となる環境向けの研究が多く見られる．

Garcia-Molinaら[Garcia-Molina87]は，Sagaと呼ばれるトランザクションモデルを提案している.ACID特性のうち分離性を犠牲にして性能向上を図るアプローチである．トランザクションをサブトランザクションに分割して実行する．サブトランザクションは完了時点で資源を解放する．サブトランザクションが失敗した場合は，そのサブトランザクションをやり直す（前進回復）か，サブトランザクション全体を取り消す（後退回復）かのいずれかができる．後退回復のために補償トランザクション（compensating transaction．サブトランザクションの効果を取り消すトランザクション）が必要となる．サブトランザクションの完了時点で，他のトランザクションがサブトランザクションの書きこんだデータを読み込むことができるという意味において分離性が保証されない．

Puら[Pu88]は,Splitトランザクションと呼ばれるトランザクションモデルを提案している．ACID特性のうち原子性を犠牲にして性能向上を図るアプローチである．トランザクションの実行途中に規則を動的に評価してトランザクションを分割し，分割されたトランザクションを別々に実行できることが特徴である．トランザクション全体のコミット可否に関わらず，サブトランザクションをコミットすることを許すという意味において，原子性が保証されない．

Elmagarmidら[Elmagarmid90]は，Flexトランザクションと呼ばれるトランザクションモデルを提案している．ACID特性のうち原子性を犠牲にして性能向上を図るアプローチである．トランザクションに対して「機能的に等価な」（functionally equivalent）サブトランザクション列を定義可能にしている．一部のサブトランザクションが失敗しても，そのサブトランザクションが失敗した場合用のサブトランザクションが「機能的に等価な」サブトランザクション列に含まれていて後者のサブトランザクションが成功すれば，トランザクション全体は成功する．耐障害性の高いモデルである．トランザクションに含まれる各サブトランザクションが必ず実行されると

は限らないという意味において，原子性が保証されない．

まとめると，連邦データベースシステム向けののトランザクション同期制御方式の研究は大別して以下の三つが行われている．①直列可能性を正当性基準とし，その正当性基準を満たす同期制御方式を研究するものと，②直列可能性を緩和した正当性基準を設定し，そのもとでの高性能な同期制御方式を研究するものと，③適用環境或いは適用アプリケーションを限定し，通常のトランザクションモデルとは異なるトランザクションモデルを導入することなどにより，高性能な同期制御方式を研究するものとの三つである．

(2)オンラインデータローダ向けのトランザクション研究の動向

ODSのオンラインデータロード用トランザクションのアクセス特性は,多量のデータの検索と少量のデータの更新とを行うことである．これ以降，ODS用のオンラインデータロード用トランザクションと同様なアクセス特性を有するトランザクションを大量検索型グローバルトランザクションと呼ぶこととする．

ここで，少量のデータの検索・更新を行うトランザクション（グローバルとローカルの両方を含む）と，大量検索型グローバルトランザクションとが共存する環境を考える．前者のトランザクションを「通常のトランザクション」と呼ぶこととする．両者が共存する環境において，両者を高性能に実行させることは簡単ではないことが知られている[Pirahesh 90]．例えば，単純に２相ロック方式を適用すると，大量検索型グローバルトランザクションの大量READが，通常のトランザクションを多数かつ長時間ウェイトさせるという問題を生じる．

両者が共存する環境向けに従来提案されているトランザクション同期制御方式としては,２相ロック方式を基本にしつつ,大量検索型グローバルトランザクションのREADにのみ特別な規則を適用する方式が知られている．代表的な２つの方式を説明する．

(a)ダーティリード

大量検索型グローバルトランザクションのREADにおいて，READ対象ページを他トランザクションが既に排他ロック（WRITEロック）していても，その排他ロックを無視してREADする．ACID特性のうち，分離性を犠牲にして，性能向上を図るアプローチである．

(b)予備系のREAD

耐障害性の向上を狙いとして，現用のデータベースとは別に予備系のデータベース（即ち，現用データベースのバックアップ）を設けている場合がある．この時，大量

検索型グローバルトランザクションのREADにおいて，現用データベースの代わりに，予備系のデータベースをREADする方式が知られている．

上記２つの方式は両方とも問題がある．(a)には，未コミットの更新途中のページをREADすることから，正当性（直列可能性）が保証されないという問題がある．(b)には，現用データベースに比べると，古いデータ（例えば１日前のデータ）を読むことになるという問題がある．

以上より，両者のトランザクションとが共存する環境において，正当性を保証しつつ，大量検索型グローバルトランザクションが鮮度の高いデータをREADするトランザクション同期制御方式が望まれる．さらに，通常のトランザクションにはミッションクリティカルなものが一般に多数存在することを考慮すると，通常のトランザクションの高性能な実行を可能にするトランザクション同期制御方式が望まれる．

1.3.5. 異種情報源連携技術の研究

前節までは，情報源をデータベースシステムに限定した場合の研究の動向を説明した．WWW（World Wide Web）の急速な普及に伴い，WWW上の情報源（HTML，プレーンテキスト，Postscriptなど）もデータベースシステムと同様に情報源として扱いたいという要求が急速に高まっており,WWW上の情報源をも連携して使用可能とするシステム（異種情報源連携システムと呼ぶ）の研究が近年盛んに行われている．

異種情報源連携システムの研究動向を説明する．異種情報源連携システムは，ウェブ言語システム(web language system)，情報ブローカシステム(information broker system)に大別される．以下，順に説明する．

1.3.5.1. ウェブ言語システムの研究動向

ウェブ言語システムとは,Yahooなどのウェブ検索エンジンで実現しているような内容ベースの検索と，ウェブページをノードとみなしハイパーリンクをアークとみなしたネットワークでの構造ベースの検索の両立を図るシステムを言う．

Mendelzonら[Mendelzon96]は，WebSQLと呼ばれるウェブ言語を提案している．

WebSQLは，WWWを二つの仮想的なテーブル，DocumentとAnchorでモデル化する．Documentの各タプルはWWW上のオブジェクト（HTML,postscrpt，画像，音声など）を表し，Anchorの各タプルはWWW内のオブジェクト間のハイパーリンクを表す．SQLライクな質問言語を提供する．FROM節でMENTIONS述語を用いることにより内容ベースの検索

を指定する．FROM節でハイパーリンクによる接続関係に関する条件を指定することにより，構造ベースの検索を指定する．質問処理機能は，内容ベースの質問に関しては，ウェブ検索エンジンを利用して処理を行う．構造ベースの質問に関しては，独自に開発したグラフの類似構造発見アルゴリズムを用いて処理を行う．

Konopskiら[Konopski95]は，WebSQLと良く似たW3QLと呼ばれるウェブ言語を提案している．FROM節に条件式（format=Latex, author=Einsteinなど）を記述することにより，内容ベースの検索を記述する．FROM節にグラフ構造をテキスト記述することにより，そのグラフ構造にマッチするWWW上の部分グラフの検索を記述する．質問処理機能は，内容ベースの質問に関しては，ウェブ検索エンジンを利用して処理を行う．構造ベースの質問の処理機能の詳細は不明である．

Lakshmananら [Lakshmanan96]はWebLogと呼ばれるウェブ言語を提案している．Weblogの構文とセマンティクスは述語論理に基づいている．質問処理機能の詳細は不明である．

まとめると，ウェブ言語システムを用いることにより，WWWの中において，内容ベースあるいは構造ベースの質問が可能になっている．これらの言語システムにおいては，利用者に同義の情報をまとめて統一的な形式で見せることを可能にする統合スキーマは存在しない．利用者が個々の情報源の所在，情報源が有している情報の種類などを知ったうえで，質問を発行する必要がある．質問発行者の負荷が大きいアプローチであると考える．

1.3.5.2. 情報ブローカシステムの研究動向

データベースなどの構造化情報や，HTMLのような半構造化情報や，テキストファイルのような非構造化情報などをそのまま利用者に見せるのではなく，抽象化して統一的な形式で見せ，利用者からは高水準な言語で統一的なアクセスを可能にするシステムを情報ブローカシステムという．情報ブローカシステムは，質問の解釈機能や，メディエーション(mediation)機能と呼ばれる，発せられた質問を，質問や情報源に関する知識を用いて適切な情報源あるいは別の情報ブローカシステムに振り向ける機能や，ラッパ(wrapper)機能と呼ばれる，質問に回答可能とするために，情報源に付加する機能などを有する．

Garcia-Molinaら [Chawathe94] [Garcia-Molina97]のTSIMIISは，構造化情報源（データベースなど）と半構造化情報源（HTMLページなど）とを統合するためのシステムである．特徴は，共通的なオブジェクト情報交換モデルOEM(Object Exchange Model)

と質問言語LOREL(Light weight Object REpository Language)とを提案していることと，ラッパとメディエータ（mediator．メディエーション機能を実現するシステム構成要素）を自動生成するためのツールを提案していることである．OEMは多様な種類のデータの記述を可能とする.LORELは,メディエータの提供する統合スキーマに対して,SQLライクな構文での簡易なアクセスを可能にする.ラッパとメディエータの自動生成ツールは，宣言的なルール定義を基にラッパとメディエータを自動生成する，データ異種性解消の観点からは，データ異種性解消のための特別な手段は提供していない．

Levyら [Levy96]のInformation manifold(IM)は，WWW上の異種情報源（関係データベース，書誌情報，ネームサーバなど）の集合に対して統一的なアクセスを提供するシステムである．特徴は，情報源の定義方法と質問実行プランの生成機能である．情報源は，情報源に含まれるオブジェクトの内容と質問回答能力を記述することにより定義する．内容は，仮想的な関係あるいはクラスに対する質問の形式で記述する．仮想的な関係やクラスの集合を世界ビュー（World View）と呼び，利用者は世界ビューを対象に質問を発行する．質問実行プラン生成機能は，質問の構文と情報源の定義とを基に，実行可能な質問実行プランを効率的に生成する．データ異種性解消の観点からは，オブジェクトの内容定義において世界ビューの同一の関係あるいはクラスを参照することによりオブジェクトの同義性を表現可能としていることが特徴である．なお，同義のオブジェクトの発見方法に関しては発見支援機能を有する旨の記述があるものの詳細は不明である．

Tomasicら[Tomasic96]のDISCOは，異種情報源（データベースシステム，ファイル，ウェブ上の情報源）に対して統一的なアクセスを提供するシステムである．DISCOは，ODMG93及びOQL[Cattel94]の拡張を,共通的なデータモデル及び質問言語として用いている．特徴は，①質問処理実行時に，質問プラン作成時は利用可能と想定していた情報源がネットワーク断などの理由で利用不可になった場合に代替情報源を探すことと，代替情報源が見つからなかった場合は部分的に欠落した質問結果を返却することと，②中間言語を設けて，情報源の質問回答能力を中間言語の文法を記述することで定義可能とすることにより，ラッパの作成を容易化することである．データ異種性解消の観点からは，オブジェクトが同一のクラスに属すると定義することによりオブジェクトの同義性を表現可能としていることが特徴である．なお，同義のオブジェクトの発見方法に関しては支援機能は提供していない．

Arensら[Arens93]のSIMSは，異種情報源（データベースシステム，知識ベースシステムなど)を含むネットワーク上での情報検索の方式を提案する．オントロジ技術(人

工的なシステムを構築する際の基本語彙・概念を整理・利用するための技術）を用いる．特徴は，ドメイン（対象領域）および情報源を定義するする方法と，ドメインに対する質問を基に効率的に質問実行プランを作成する方法とである．ドメインは，知識表現言語を用いて語彙の階層および語彙間の関連を記述することにより定義する．情報源は，情報源の種類と，質問最適化用の統計データと，情報源の内容とをドメインの語彙を用いて記述することにより定義する．質問実行プラン作成機能は，汎用的なプラン生成機能を用いて，情報源の選択と質問実行プランの生成・最適化を行う．データ異種性解消に関しては，情報源を定義する際に，ドメインの語彙を厳密に使用することにより自動的にデータ異種性を解消可能としている（同名同義，異名異義が保証される）．

Bayardoら [Bayardo97]のInfoSleuthは,動的に変化するネットワーク環境における情報検索のアプローチを提案する．InfoSleuthはエージェント技術，オントロジの技術を用いる．InfoSleuthのアーキテクチャは，異種の半自律的なエージェント（利用者，ブローカ，オントロジ，実行，資源など）のネットワークからなる．各エージェントは固有の機能を実現する．エージェントは，Knowledge Query and Manipulation Language(KQML)を用いて互いに通信する．利用者エージェントは，エンドユーザが利用者が質問を作成することと，質問結果を利用者のニーズに適した形式で表示することとを支援する．ブローカエージェントは，要求とサービスとのマッチング（特定のサービスを要求するエージェントと，その要求を満たすエージェントとを結びつけること）を行う．全てのエージェントは自分のサービスを自己申告するので，ブローカは容易に要求に対応する情報源を発見できる．オントロジエージェントは，オントロジ（概念のネットワーク）の管理を行う．オントロジは，ユーザが質問を作成するための語彙を提供する．タスク実行エージェントは，質問の実行を行う．資源エージェントは，特定の情報源（関係データベースやテキストドキュメントの集合）へのアクセスを行う．データ異種性解消に関しては，各エージェントが自分のサービスを自己申告する際に，オントロジの語彙を厳密に使用することにより自動的にデータ異種性を解消可能としている（同名同義，異名異義が保証される）．

まとめると，情報ブローカシステムは，ウェブ上の異種情報源連携システムに関して有望な解を提供する．ラッパ機能を提供することにより各情報源の自律性を維持しつつ，システムによって方法の差異はあるもののデータ異種性の解消を可能にしている．但し，現状の情報ブローカシステムが扱っている情報源の種類・数は，ウェブ上に実際に存在する情報源の種類・数に比べるとはるかに少ない．ラッパ作成技術，情

報源定義技術などの研究において，スケーラビリティを向上させるために一層の研究が必要と考える．

1.4. 本研究の研究課題

1.3節で説明した研究動向を踏まえ,異種データベースシステム連携の基本技術の研究を研究課題として設定した．具体的には，研究動向において重要な課題として認識されており，かつ，筆者の所属する組織においてニーズが顕在化したあるいはニーズが顕在化すると想定された以下を研究課題として設定した．

(1) 連邦データベースシステムにおけるスキーマ構築方式

(2) 企業アプリケーション統合におけるデータ交換システム構築方式

(3) データウェアハウスにおけるオンラインデータローダ高速化方式

本研究の目的は，上記の課題を解決し，異種データベースシステム連携技術の確立に寄与することである．

1.5. 本研究のアプローチ

研究課題毎にアプローチを説明する．

1.5.1. 連邦データベースシステムにおけるスキーマ構築方式

スキーマの構築方式の研究においては，研究動向で説明した以下の課題を解決する必要がある．

・ 構築簡易化：スキーマの構築を簡易化することが要求される．

ここで，本課題の研究における前提条件について述べる．要素データベースシステムのデータモデルは，関係データモデルに限定することとする．

これは以下の知見に基づく．

・ 現状の組織において構築されているデータベースシステムの殆ど（８割以上）は関係データモデルに基づくデータベース管理システム（関係データベース管理システム）によって構築されている[データベース白書00]．

・ オブジェクト指向データベース管理システムなど，関係データモデル以外のデータモデルに基づくデータベース管理システムも導入されつつある．しかしながら，

組織でのビジネス用途においては，暫くの間は関係データベースシステムが主流であり，他のデータモデルに基づくデータベースシステムは関係データベースシステムを補完する位置付けであると考える．従って，要素データベースシステムのデータモデルを関係データモデルに限定しても研究の有用性が損なわれることは少ないと考える．

課題に対しては，以下に示すように，実体レベルの構築作業とデータ項目レベルの構築作業のそれぞれに効率的な支援機能を備えることによって対処する．

・ スキーマ構築作業全般を通して実体レベルの作業の稼動の削減を図るため，連邦スキーマの関係（テーブル）として普遍関係[Ullman89]を導入することとする．普遍関係の導入により，実体間の関連特定作業，関連を特定された実体を基にしての統合スキーマの作成作業などが不要になり，スキーマ構築全体の稼動が大きく削減されることが期待できる．

・ 同義のデータ項目の組み合わせの発見作業の稼動の削減を図るため，用語辞書を用いた類似データ項目分類支援機能を設ける．名称或いは名称の類義語との一致によって，関連を特定するデータ項目の範囲を絞り込む．

1.5.2. 企業アプリケーション統合におけるデータ交換システム構築方式

データ交換システムの構築方式の研究においては，研究動向で説明した以下の課題を解決する必要がある．(1)(2)はデータ交換システムの有すべき機能に関する課題であり，(3)はデータ交換システムの構築方式に関する課題である．

(1) データ異種性解消

データ異種性を解消することが要求される．

(2) 自律性維持

交換先データベースシステムの入力方法に影響を与えることなくデータ交換を可能とすることが要求される．

(3) 構築簡易化

データ交換システムを簡易に構築可能とすることが要求される．

これらの課題に対して，①先ず，様々な形態のデータ交換に必要な操作機能のモデル（データ交換処理モデルと呼ぶ）を明確化し，②次いでそのモデルに沿ったデータ交換システム開発・実行環境を提案することにより対処する．

データ交換処理モデルは，形式的なデータと機能のモデルの考案と，それに基づく

操作機能の導出とを特徴とする．

データ交換システム開発・実行環境は，データ交換の仕様を簡易に記述できる高水準仕様記述言語を設けることと，基本的な操作機能と項目値変換関数との部品化と再利用を可能とすることとを特徴とする．

1.5.3. データウェアハウスにおけるオンラインデータローダ高速化方式

オンラインデータローダの研究においては，研究動向で説明した以下の課題を解決する必要がある．

- 大量検索型グローバルトランザクション（オンラインデータローダ用トランザクション）と通常のトランザクションとが共存する環境向けのトランザクション同期制御方式，特にミッションクリティカルなものが一般に多数存在する通常のトランザクションの高性能な実行を可能にするトランザクション同期制御方式が要求される．

併せて以下の課題を解決する必要がある．

- 異種性解消：ローカルなデータベース管理システム間の異種性を簡易に解消して，提案するトランザクション同期制御方式が実現できること．すなわち，ローカルなデータベース管理システムの機能への影響（改造）を少なくしつつ，提案するトランザクション同期制御方式が実現できること．

同期制御方式に関しては，グローバルトランザクションが検索専用の場合は，同期制御に起因する副作用が小さい（アボートは一切発生せず，他のトランザクションとの間で待ちが発生するのはREADの実行時に未完了のトランザクションによって作成された版を選択しそのトランザクションの完了を待つ場合のみ）という特長を持つ方式として多版時刻印方式[Reed83]があることに着目し，該方式と同期制御方式の主流である２相ロック方式とをベースに新たな方式を考案することで対処する．

異種性解消方式に関しては，以下の方針に基づき，新たな方式を考案することで対処する．

- 外付けの機能（グローバルな質問処理機能およびトランザクション同期制御機能）と，ローカルデータベース管理システムのオリジナルな機能（質問処理機能およびトランザクション同期制御機能）を改造した機能とが協調動作して，提案方式を実現することを狙う．
- ローカルデータベース管理システムのオリジナルな機能（質問処理機能およびト

ランザクション同期制御機能）への改造は極小化する．

1.6. 本論文の構成

2章で連邦データベースシステムにおけるスキーマ構築方式について述べ，3章で企業アプリケーション統合におけるデータ交換システム構築方式について述べ，4章でデータウェアハウスにおけるオンラインデータローダ高速化方式について述べ，5章で論文全体のまとめを述べる．

1.7. 用語集

実体(entity)：実世界の「もの」を実体と呼ぶ．同一の実体であっても，用いるデータモデルによって，実体の表現方法は異なる．例えば，関係データモデルでは実体は関係（テーブル）で表現され，オブジェクト指向モデルではオブジェクトで表現される．

データモデル(data model)：データモデルとは，データベース中のデータとそれに対する操作を規定する枠組みである．具体的には，データベースシステムがどのようなデータ構造を持ち，どのようなデータ操作を行え，満たすべき整合性制約は何かを示すものである．

データベース(database)：複数の応用目的で，共有を意図して，組織的かつ永続的に格納されたデータ群のことを言う．

データベース管理システム(database management system)：データベースのデータを管理し，利用に供するためのソフトウェアをデータベース管理システムと呼ぶ．データベース管理システムが提供する主な機能には，質問を実行するための質問処理機能，複数トランザクションの並列走行時の無矛盾性を保証するトランザクション同期制御機能などがある．

データベースシステム(database system)：データベース管理システムと，それが管理するデータベースとを一体としてとらえたものをデータベースシステムと呼ぶ．

スキーマ(schema)：データベースシステム中のデータの構造，関連，各種の整合性制約などを記述したものをスキーマと呼ぶ．

トランザクション(transaction)：データベースに対する，一連の検索・更新操作の集まりをトランザクションと呼ぶ．並列処理されるトランザクション同士の競合や，各種障害の発生に起因して，データベースの矛盾が発生することのないように，トランザクション処理においては，ACID特性と呼ばれる以下の性質が成り立つことを保証する必要がある．

- 原子性(Atomicity)：データベースに対する操作が全て有効であるか，或いは全て無効であるかのいずれかであること．
- 無矛盾性(Consistency)：無矛盾な状態のデータベースにおいてトランザクションを実行した場合，トランザクション実行後のデータベースの状態も無矛盾であること．
- 分離性(Isolation)：トランザクションがあたかも他のトランザクションからの干渉がなかったかのように実行されること．
- 耐障害性(Durability)：コミットされたトランザクションの効果が，データベース中で，障害などに起因して消滅することのないこと．

データベースに矛盾が発生しないように複数トランザクションを並列走行させる機構をトランザクションの同期制御(concurrency control) と呼ぶ．同期制御は，ACID特性のうち主に分離性を保証ための機構である．

企業アプリケーション統合 (EAI：Enterprise Application Integration)：データベースのデータとアプリケーションを統合利用するための技術の体系，あるいはそれらの技術を用いて構築されたシステムを言う．EAIの技術は，データレベルの技術，アプリケーションインタフェースレベルの技術，メソッド（共有および再利用可能なロジックをメソッドと呼ぶ）レベルの技術，ユーザインタフェースレベルの技術の四つに大別される．

統合基幹業務ソフト (enterprise resource planning package)：経理，生産管理，販売管理，人事管理などの基幹業務を個別に行うのでなく，コンピュータ・システムを使って密接に関係付けながら実行することを企業資源管理 (enterprise resource planning) と呼ぶ．企業資源管理の実現に必要な機能をあらかじめ備えたソフトウエ

ア群を統合基幹業務ソフトと呼ぶ．代表的なものに，独SAP社のR/3，米PeopleSoft社のPeopleSoftなどがある．

データウェアハウス(Data Warehouse)：主として経営での意思決定を支援することを目的としたデータの集合（倉庫）を言う．データが特定主題向けであること（サブジェクト指向と呼ぶ）と，統合されていることと，時系列データを保有することと，データが検索・追加のみで更新がないこととを特徴とする．

データマート(Datamart)：全社レベルのデータウェアハウスと部門レベルのデータウェアハウスとを区別して呼ぶ場合に後者をデータマートと呼ぶ．全社レベルのデータウェアハウスと比較してのデータマートの特徴は,部門/エンドユーザレベルの個別共用を実現することと，小規模データウェアハウスであることと，少数（数個）のサブジェクトから構成されることである．

オペレーショナルデータストア（ODS：Operational Data Store)：データウェアハウスの一形態．遅延の少ない意志決定支援を可能にすることなどを目的に，基幹系データベースシステムと同じ環境に構築されるデータウェアハウス．ODSは，サブジェクト指向と統合という点では通常のデータウェアハウスと共通の特性を有するが，現在値のみを有する点とデータが更新されるうる点では通常のデータウェアハウスとは特性が異なる．

レガシーシステム(legacy system)[Brodie95]：典型的には，数百万行のコード行数を有し，COBOLやFORTRANなどの旧式のプログラミング言語でプログラミングされており，１０年以上にわたり開発されており，通常ミッションクリティカルなシステムを言う．新たな要求に合わせて，更改を行うのが困難であることが特徴である．

オントロジ(ontology)：データベースシステムシステムを始めとする人工的なシステムを構築する際の，構築要素として用いられる基本概念や語彙の体系を言う．共有・再利用が可能で，人間と計算機の双方に理解可能なものであることが求められる．

2. 連邦データベースシステムにおけるスキーマ構築方式

2.1. まえがき

本章では，連邦データベースシステム(federated database system)のスキーマ構築(schema integration)の方式を論じる[池田 99].

組織，特に企業はグローバルな競争に巻き込まれており，経営の効率化や経営戦略の変革を従来以上にスピーディに行うことを迫られている．このような状況のもとでは,複数のデータベースシステムを連携して顧客管理，事務処理，在庫管理などを効率化することをスピーディに実現できることが強く要求されるようになっている．

既存の，互いに異種性を有する（即ち，データモデル，質問言語，データ項目の値の表現形式などが同一とは限らない）データベースシステムを連携して利用する形態の代表的なものの１つに連邦データベースシステムがある．連邦データベースシステムは，複数の要素データベースシステムのスキーマを基に，論理的に統合されたスキーマ(連邦スキーマとも呼ぶ)を構築し，利用者は連邦スキーマがあたかも通常のスキーマであるかのようにアクセスすることを可能にするシステムである．

１章で述べたように，従来の連邦データベースシステムの研究には，効率的なスキーマ構築方式が確立されていないという問題がある．

筆者はこの問題に応えるため，新たなスキーマ構築方式を提案する．また，該方式を実現する連邦データベース検索システム DBSENA を試作し,実際の連邦データベースシステムのスキーマ構築に適用し，提案方式の有効性を検証する．

提案するスキーマ構築方式の主たる特徴は，①普遍関係を連邦スキーマにおいて利用可能とし,②用語辞書を用いた類似データ項目の分類支援機能を設けることである．

2.2. スキーマ構築方式への要求条件

スキーマ構築方式への要求条件を明確化する．

要求条件は，複数の要素データベースシステムのスキーマを基にスキーマを構築するための基本的な要求条件（データ異種性解消条件）と，従来技術の問題点を解決するための条件(スキーマ構築簡易化条件)の二つに大別される．

2.2.1. データ異種性の解消

既存の要素データベースシステムはそれぞれ他の要素データベースシステムと異なる設計方針に基づき設計されるため，様々なデータ異種性が生じる．

データ異種性の分類については，連邦データベースシステム[Sheth90]において，スキーマ定義言語/質問言語に要求される記述能力を明らかにすること等を狙いとして，関係データモデルにおけるデータ異種性を分類したKimの分類（１章表1.1参照）が知られている[Kim93]．

連邦データベースシステムにおいては，利用者からは同一の実体あるいは同一のデータ項目が例え異なる要素データベースシステムに分散して存在しても表現方法の違いを意識すること無くアクセス可能なように，１章の表1.1に挙げたデータ異種性を解消する機能が要求される．

2.2.2. スキーマ構築の簡易化

スキーマの構築が簡易であることが望まれる．

2.3. スキーマ構築方式

要求条件を満たす，スキーマ構築の一方式を提案する．

先ず，スキーマ構築方式を説明し，次いで該方式に対応するスキーマと連邦データベース管理機能とについて説明する．

2.3.1. スキーマ構築方式

先ず，スキーマ構築の主要なプロセスについて説明する[Ram98]．

(1)スキーマ変換(schema translation)：以下のタスクからなる．

・ ローカルスキーマから共通データモデルのスキーマ（要素スキーマ）への変換．

・ 連邦データベースシステムに対してアクセスを許す部分(輸出スキーマ)の定義．

・ 輸出スキーマの収集．

(2)実体・データ項目間の関連特定(interschema relationship identification)：以下のタスクからなる．

・　実体間，データ項目間，実体とデータ項目間の関連の特定.

(3)統合スキーマ作成(integrated schema generation)：以下のタスクからなる.

・　特定された関連を基にしての，統合スキーマの作成.

・　輸出スキーマと連邦スキーマの間のマッピングの整理

以下，上記の構築プロセスに沿って，提案するスキーマ構築方式を説明する.

(1)スキーマ変換

スキーマ構築作業全般を通して実体レベルの作業の稼動の削減を図るため，連邦スキーマの関係（テーブル）として普遍関係(universal relation)[Ullman89]を採用することとする．普遍関係とは，データベース中の全データが必ずその中に含まれる単一の(仮想的な)関係を言う．普遍関係は，関係データベースシステムでの質問を簡単化すること，即ちデータ項目と関係の組を意識すること無くデータ項目だけを意識すれば質問可能とすることを主たる狙いとして導入・研究されてきた概念である.

普遍関係の採用に伴い，統合スキーマ作成プロセス(3)における実体レベルの統合作業が不要となり，従って本プロセスではその準備作業の一つであるローカルスキーマから共通データモデルのスキーマ（要素スキーマ）への変換が不要となる.

本プロセスでは輸出スキーマの定義および輸出スキーマの収集のみ必要となる.

輸出スキーマの定義は，研究の前提（1.5.1節）で述べたように要素データベースシステムを関係データベースシステムに限定したので，関係データベース管理システムのビュー機能などを用いることにより容易に定義可能である.

輸出スキーマの収集に関しては，自動収集機能を設けて効率化を図ることとする．要素データベースシステムを関係データベースシステムに限定したので，関係データベースシステムを対象に自動収集機能を設けることとする．ここで，同じ関係データベースシステムでも製品毎にスキーマ構造は少しずつ異なることより，代表的な商用関係データベース管理システム毎に個別にスキーマ自動収集機能を設けることとする(具体的な対応製品名は省略する)．このようにしたのは，ビジネスの場において使用される商用関係データベース管理システムの種類はほぼ収束しており，それらに対応しておけば，必要に応じて機能の追加は予定しているものの実際には新たな機能の作り込みが必要になる可能性は小さいと考えたためである.

また，スキーマ的異種性のうちの1データ項目対1データ項目の場合のデータ型の差異と，データ的異種性のうちの値の表現方法の差異とを解消するために，ドメイン概念を設け，各データ項目毎にドメイン値を設定することとする.

ドメインとは,筆者の属する研究組織で研究･開発したデータベース設計法[NTT96a]においてコードと呼んでいるものと同じ概念であり，具体的には，データ型と，精度と，null 可不可情報と，値範囲情報と，自然言語による説明とからなる．ドメインとして管理する情報の種類をこのようにしたのは，データベース設計法の使用経験，特に既存の複数データベースシステムを更改して新規データベースシステムを設計した経験から，データ型と値の表現方法の異同を判断するには実用上十分であると判断したためである．

(2)実体・データ項目間の関連の特定

普遍関係を採用することにより統合スキーマ作成プロセス(３)における実体レベルの統合作業が不要となることに伴い，本プロセスではその準備作業の一つである実体レベルの関連特定が不要となる．

本プロセスではデータ項目間の関連特定のみ必要となる．

筆者らの属する研究グループにおいて，データ標準化支援ツールの一部として開発した，類似データ項目の分類支援機能[関根 93]を用いてデータ項目間の関連特定の効率化を図ることとする．

類似データ項目の分類支援機能を簡単に説明する．

・ データ標準化支援ツールの一部である用語辞書を用いる．用語辞書の特徴は，①全ての用語を Durell の分類[Durell87]に基づき，値の種類を表す用語である区分語(class word. 例：番号，日付)と，データ項目が管理する対象を表す用語である主要語(prime word. 例：顧客，回線，工事)と，データ項目の内容を区別する語である修飾語(modifier word. 例：新規，重要，上位)との三つに分類してあることと，②全ての用語に対して類義語のグルーピング（個々の用語はただ一つのグループに属する）を定義してあること，である．

なお，用語辞書の構築方法の詳細に関して，付録１で説明する．

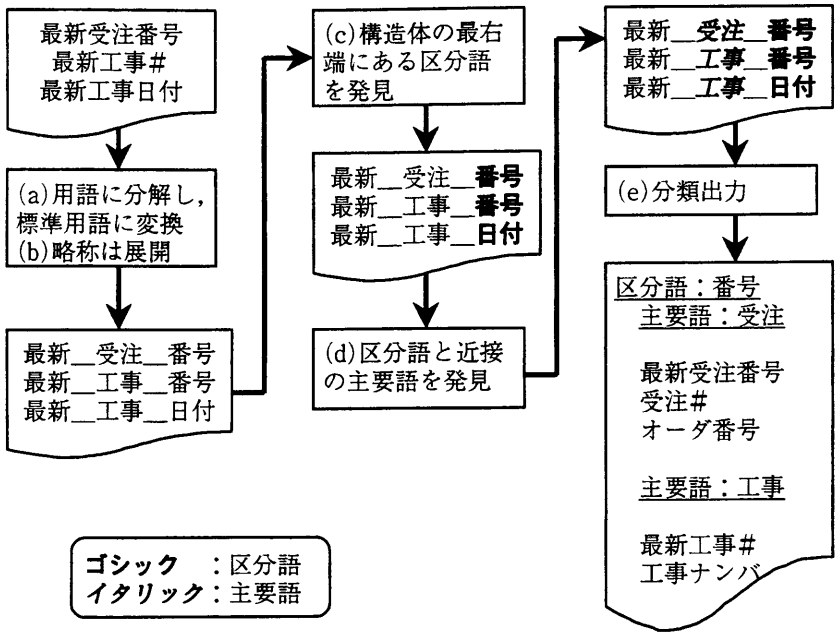


図 2.1　類似データ項目分類の手順

・ 分類支援機能実行時に自動的に，データ項目の名称を形態素解析し，分類対象のデータ項目名集合の中から，名称の区分語及び主要語が互いに類義語の関係にあるものを類似データ項目の候補として分類する．アルゴリズムの概要を示す(図2.1)．

(a)　データ項目名を用語に分解し，それぞれを用語辞書を用いて標準用語に分類する．

(b)　略称(＃など)がある場合，略称を標準用語に展開する．

(c)　データ項目名を構成する用語を末尾から順に左に探し，区分語になりうる標準用語を探す．

(d)　見つかった区分語から，さらに左方に主要語を探す．

(e)　区分語を大分類，主要語を小分類として，データ項目を分類する．

該支援機能で類似データ項目の分類が行われた後は，人間の判断により，分類されたデータ項目のグループの中からデータ項目間の関連（同義性）を特定する．

なお，同義性の分類としては，同義/異義の２分類を用いる．同義のさらなる細分類（等価，包含，オーバラップなど）を用いることはしない．

この支援機能を用いたのは，以下の知見による．

・ 1章で述べたように，汎用的な知識（用語辞書）を用いて類似データ項目の組合せの候補を自動的に絞り込むことによりデータ項目間の関連特定の効率化が期待できる．

・ Durell の分類を用いない類似データ項目分類支援の方式（互いに類義語である用語数が一定数以上のものを単純に類似データ項目候補とみなす方式）と比較して，本方式は高精度な方式（類似でないデータ項目が同一グループに分類される度合いの低い方式）であることが検証されている[関根 93]．

(3)統合スキーマの作成

普遍関係を採用することにより，実体レベルの統合作業，即ちテーブルの統合作業が不要となる．

本プロセスではデータ項目レベルの作業のみ必要となる．関連を有するデータ項目間のマッピングの整理は，特定された関連（同義性）の対応関係の基数に応じて作業を行う．作業例を以下に示す．

・ 1対1対応の同義のデータ項目に関しては，先ず対応する連邦スキーマのデータ項目を設け，次いで連邦スキーマのデータ項目と輸出スキーマのデータ項目とのマッピング情報をスキーマに登録し，最後に連邦スキーマのデータ項目と輸出スキーマのデータ項目との間の変換関数を決定し，必要ならば変換関数の作成を行う．

・ 複数対複数対応の同義のデータ項目に関しても同様の作業を行う．例えば，関連特定において，ある輸出スキーマのデータ項目「販売日付」が，他の輸出スキーマのデータ項目「販売年」「販売月」「販売日」と対応していることが特定できたとする．この時，例えば，先ず対応する連邦スキーマのデータ項目としては「販売日付」を設けることとし，次いで連邦スキーマのデータ項目と輸出スキーマのデータ項目とのマッピング情報をスキーマに登録し，最後に連邦スキーマのデータ項目と輸出スキーマのデータ項目との間の変換関数を決定し，必要ならば変換関数の作成を行う．

ここで，変換関数作成の稼動削減を図るため，予め標準的な変換関数を設けることとする．変換関数の品揃えとしては，データ交換システム開発・実行環境 DB-STREAM[池田 97]で設けた変換関数（DB-STREAM における項目レベル変換機能メソッドに相当）をそのまま設けることとする(表 2.1)．DB-STREAM では，既存の8つのデータベースシステムをサンプル調査し，データベースシステム間でのデータ変換機能を実現するため

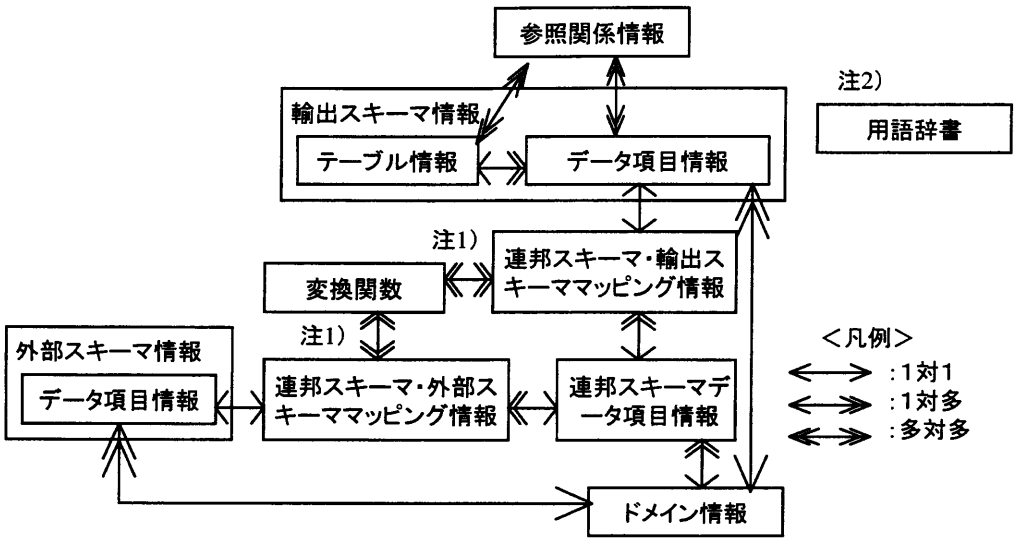
の機能としてデータ型変換関数や文字列操作関数など５８種類の変換関数を抽出し，それらの関数を関数ライブラリとして設けた．

表 2.1　変換関数(抜粋)

	データ項目値変換関数の分類		
項番	大分類	中分類	小分類
1	ﾃﾞｰﾀ型の変換	ﾃﾞｰﾀ型変換	INT->CHAR
			CHAR->INT
			CHAR->SHORTINT
			その他30種
2	ﾃﾞｰﾀ項目の統合・分解	文字列操作	文字列結合
			部分文字列
		算術演算	四則演算
			剰余
			最大,最小
3	ﾃﾞｰﾀの値の体系の変換	対応表変換	対応表変換
		文字列操作	文字列結合
			部分文字列
			左ｼﾌﾄ/右ｼﾌﾄ
			特定文字間文字列取出
			文字列比較
			その他11種 (文字探索/置換等)
		算術演算	四則演算
			剰余
			最大,最小

DB-STREAM で設けた関数群をそのまま設けることとしたのは以下の知見による．変換関数は，新たなドメインが現れる毎に必要になりうる(例：年月日の全く独自の表現方法)が， DB-STREAM の使用経験を通して，上記関数群を用意しておけば，追加が必要となる頻度はそれほど大きくないと期待できることが判明しているためである．

なお，普遍関係を用いる場合は，予め参照関係(キーデータ項目と外部キーデータ項目の関係)情報を登録しておく必要がある．これは，質問処理において，参照関係情報を基に結合条件を生成しなければならない場合があるからである(2.3.3 節で説明する)．ここで，参照関係情報の登録方式は，以下の方式とする．(2)で説明した類似データ項目の分類支援機能による分類が行われた後，人間の判断により分類されたデータ項目のグループの中から先ずテーブルのキーであるデータ項目を抽出する.次いで，グループの中から同義のデータ項目(すなわち外部キー)を抽出し，キーデータ項目と外部キーデータ項目の組を登録する．この方式では，人間の判断が必要となるが，類似データ項目の分類支援機能の利用により判断対象が絞り込まれるため，それほど大きな負荷なく判断可能と考える．



注1)マッピング情報と変換関数との間の関連が多対多となるのは，変換を変換関数の組合わせで実現することがあるため．
注2)用語辞書と他の管理情報との関連は略．

図 2.2 連邦データベースシステムのスキーマ

2.3.2. スキーマ

連邦データベースシステムのスキーマにおいては以下の情報を管理する(図 2.2)．

- 輸出スキーマ情報

  要素データベースシステムの輸出スキーマの情報．主な管理対象はテーブルとデータ項目である．テーブルに関しては名称，構成データ項目を管理し，データ項目に関しては名称，ドメイン情報を管理する．

- 参照関係情報

  要素データベースシステムにおける異なるテーブルのデータ項目間の参照関係を管理する．

- 連邦スキーマ情報

  連邦スキーマのスキーマ情報．連邦スキーマに含まれるデータ項目を管理する．各データ項目に関しては名称とドメイン情報を管理する．

- 外部スキーマ情報

  質問を発行する利用者向けのスキーマ情報．該当スキーマに含まれるデータ項目

を管理する．各データ項目に関しては名称とドメイン情報を管理する．

・ マッピング情報

輸出スキーマ，連邦スキーマ，外部スキーマのデータ項目間の対応関係を管理する．併せてそれらの間の変換用の変換関数名も管理する．

・ 変換関数

データ項目の値を変換する関数．

・ 用語辞書

類似データ項目の分類支援のために用いる用語辞書．

2.3.3. 連邦データベース管理機能

利用者が連邦スキーマを用いて連邦データベースシステムにアクセスするのを可能にするための，連邦データベース管理機能について説明する．

なお，本研究においては，連邦データベースシステムの更新機能は将来課題とし，検索関連機能のみを研究対象とした．

2.3.3.1. スキーマ管理機能

スキーマを管理する機能である．

通常のデータベース管理システムのスキーマ管理機能と同様な機能，即ちスキーマデータを格納効率良く格納する機能と，無矛盾性・耐障害性を保証する機能と，実行時処理機能とのインタフェース機能と，スキーマ入力・編集機能とを設ける．さらに既述の要素データベースシステムのスキーマ自動収集機能を設ける．

2.3.3.2. 実行時処理機能

実行時の処理を行う機能である．１章で説明した連邦データベースシステムの主要機能（1.3.3節）のうち，質問処理機能に相当する機能である．理解のし易さのため，機能をスキーマ間変換機能と（スキーマ間変換機能を除いた狭義の）質問処理機能とに大別して説明し，さらに質問処理機能の部分機能の一つである検索候補生成機能について詳しく説明する．

(1)質問の構文

質問の構文は，SQL[Darwen97]の構文を簡易化した以下の構文とする．普遍関係を採用したため，FROM節は不要となる．

項目名リスト WHERE {(項目 比較演算 項目or定数)を論理演算子で結んだもの}

なお,(3)の質問処理機能において質問を要素データベースシステム向けに分解したものは以下の構文になる.

SELECT 項目名リスト FROM テーブル名リスト

WHERE {(項目 比較演算 項目or定数)を論理演算子で結んだもの}

この構文は,ごく基本的なSQLの構文であり,一般的な商用関係データベースシステムにおいては,特に変換することなくそのまま処理可能な構文である.従って,連邦データベースシステムの質問言語と要素データベースシステムの質問言語との間の質問言語変換機能を,実行時処理機能の1つとして設けることはしなかった.

(2)スキーマ間変換機能

外部スキーマ,連邦スキーマ,輸出スキーマ間でデータ項目の名称と値の変換を行う機能である.

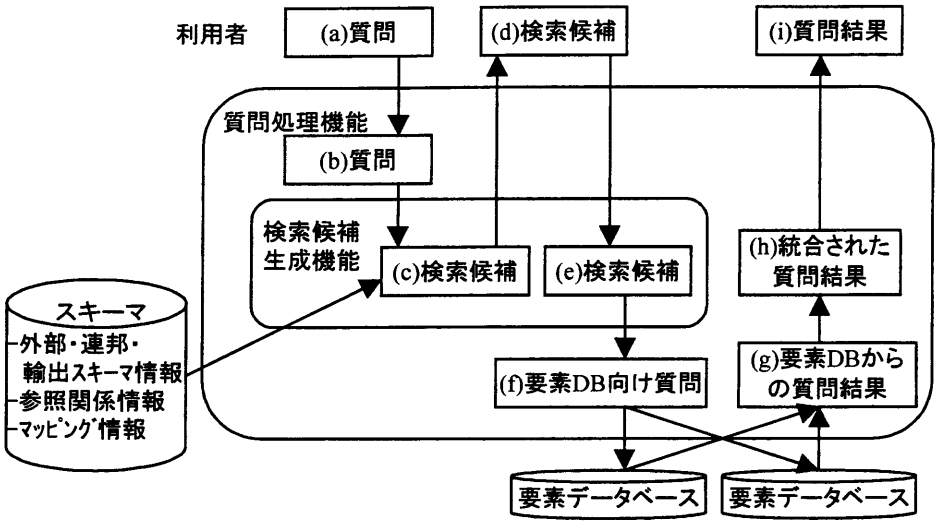


図 2.3 質問処理

(3)質問処理機能

利用者から発行された質問を処理する機能である．図を用いて機能を説明する(図2.3)．

- 利用者からの質問(図2.3(a))をスキーマ間変換した質問(図2.3(b))に対して，検索候補生成機能が検索候補(図2.3(c))を作成する．その際，(4)で説明する判断支援情報を検索候補に付加する．
- 提示された検索候補(図2.3(d))から，利用者に所望の検索候補を選択させる．
- 選択された検索候補(図2.3(e))を要素データベースシステムに対して発行する．検索候補が複数データベースシステムに跨る場合は，各データベースシステム向けの質問(図2.3(f))に分解してから発行する．
- 各要素データベースシステムからの質問結果(図2.3(g))を統合し(図2.3(h))，さらにスキーマ変換を施して利用者に返却する(図2.3(i))．

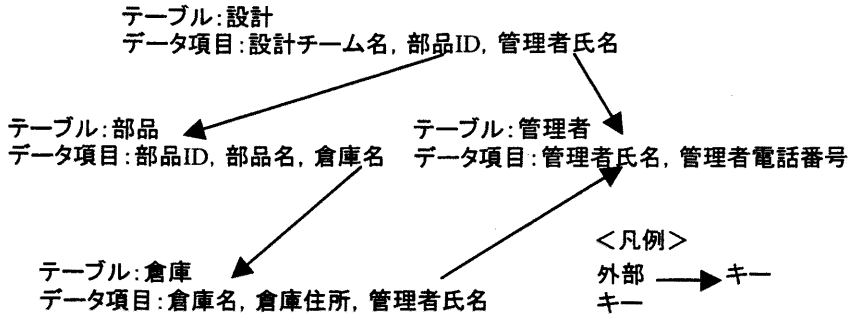


図 2.4 要素データベースシステムの例

(4)検索候補生成機能：曖昧性問題解決機能

検索候補生成機能は，質問処理機能の部分機能であり，普遍関係を採用した場合に問題となる曖昧性の問題(ambiguity problem)を解決するための機能である．

曖昧性とは，質問の構文中に現れるデータ項目が複数のテーブルに跨る場合に質問処理機能において結合条件(複数の結合条件の連接もありうる)の付加が必要となるが，その際結合条件の候補が複数生じうる事象を言う[Zhao95]．付加される結合条件は，テーブル間の参照関係の辿り方を意味するが，それが，質問を発行した利用者の意図と一致しているとは限らないことが問題となる．

曖昧性を例で説明する．図2.4に示すように，設計，部品，倉庫，管理者の４つのテーブルからなる要素データベースシステムを考える．４つのテーブルの間には図中の矢印で示す参照関係が存在すると仮定する．この時，以下の質問を利用者が発行することを考える．

管理者氏名，管理者電話番号

WHERE 部品ID＝” abc123”

２通りの結合条件の組み合わせが考えられ,それぞれの場合の質問の意味は異なる．

(a)部品と倉庫とを倉庫名で結合し，併せて倉庫と管理者とを管理者氏名で結合する．

これは部品を管理する倉庫の管理者情報の質問を意味する．

(b)部品と設計とを部品IDで結合し，併せて設計と管理者とを管理者氏名で結合する．

これは部品を設計する部門の管理者情報の質問を意味する．

曖昧性の問題に関しては以下の方法で対処することとする．

検索候補生成機能において検索候補を提示する際に，質問発行者の判断支援用に以

下の情報（判断支援情報と呼ぶ）をスキーマから取得し付加する．

・ 質問文に陽に現れるデータ項目に関してその属するテーブル名とデータベース名

・ 結合条件を付加する場合，結合条件及び各結合キーデータ項目の属するテーブル名とデータベース名

質問発行者が，判断支援情報を用いて，提示された複数の候補の中から所望の候補の選択を行う．

曖昧性の説明の(a)を例に用いて，具体的な判断支援情報の提示方法を以下に示す．データ項目の属するテーブル名は，＜検索項目＞と＜検索条件＞とにおいてデータ項目名を修飾する名称として示される(データベース名は簡単化のため略してある)．結合条件及び各結合キーデータ項目の属するテーブル名は，＜関連情報＞として示される．

＜検索項目＞管理者．管理者氏名，管理者．管理者電話番号

＜検索条件＞部品．部品ID＝”abc123”

＜関連情報＞部品．倉庫名＝倉庫．倉庫名 and

倉庫．管理者氏名＝管理者．管理者氏名

本方式では，質問発行者による候補選択作業が必要となるが，判断支援情報を明示することにより，大きな負荷なく候補の選択が可能と考える．

2.4. 連邦データベース検索システム DBSENA

前節で述べたスキーマ構築方式を実現する一システムとして，連邦データベース検索システム DBSENA を試作した．

2.4.1. アーキテクチャ

DBSENA は，大きくスキーマと連邦データベース管理機能からなる．DBSENA のアーキテクチャを図 2.5 に示す．

2.4.2. スキーマ

スキーマは 2.3.2 節で説明したスキーマを実現するものである．実現容易性を考慮して市販のデータベース管理システムを用いて作成した．

2.4.3. 連邦データベース管理機能

連邦データベース管理機能は 2.3.3 節で説明した連邦データベース管理機能を実現するものである．

利用者(アプリケーションプログラム)が実際に連邦データベースシステムにアクセスする際に用いるインタフェースである，DBSENA のアプリケーションプログラムインタフェース(API)について特徴を説明する．

API としては，コールレベルインタフェース[Darwen97]を採用した．これは，他の有力なインタフェース方式である埋め込み質問言語インタフェースと異なり，コールレベルインタフェースではプリプロセスが不要となることにより，アプリケーションの構築が単純化され，デバグ作業が容易化されるためである．具体的には C 関数のライブラリとして API を提供した．また，API 仕様の策定にあたっては，将来 JAVA などのオブジェクト指向言語に向けて API を拡張する際の拡張容易性を考慮してサン・マイクロシステムズ社が開発した JDBC(Java Database Connectivity)[Java97]を参考にした．

API は，操作対象のデータ種別によって，質問の操作，検索候補の操作，質問結果

の操作などに大別される．主な API を表 2.2 に示す．

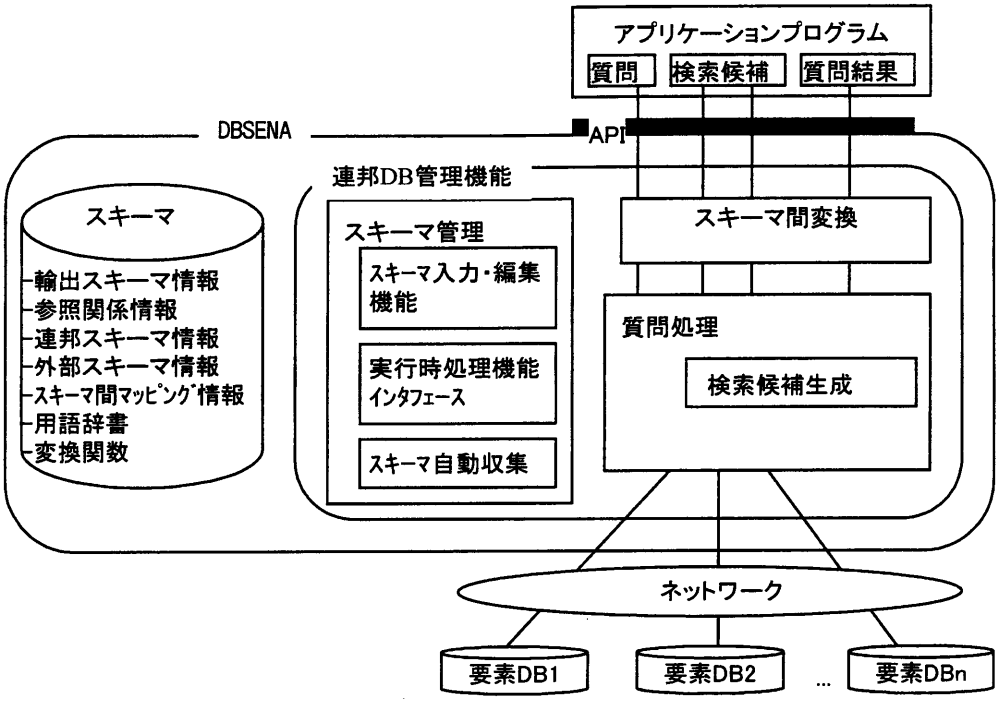


図 2.5 DBSENA のアーキテクチャ

表 2.2 DBSENA-API(抜粋)

分類	API名（C関数名）	説明
質問の操作	DBsena_Query_createQuerySet	質問を生成し，DBSENAに発行する．
検索候補の操作	DBsena_QuerySet_getQuery	判断支援情報を付加された検索候補を取得する．
質問結果の操作	DBsena_ResultSet_next	質問結果内で参照ポインタを次のレコードに設定する．
	DBsena_ResultSet_getCharArray	カレント参照レコードの指定された位置の文字列データを取得する．

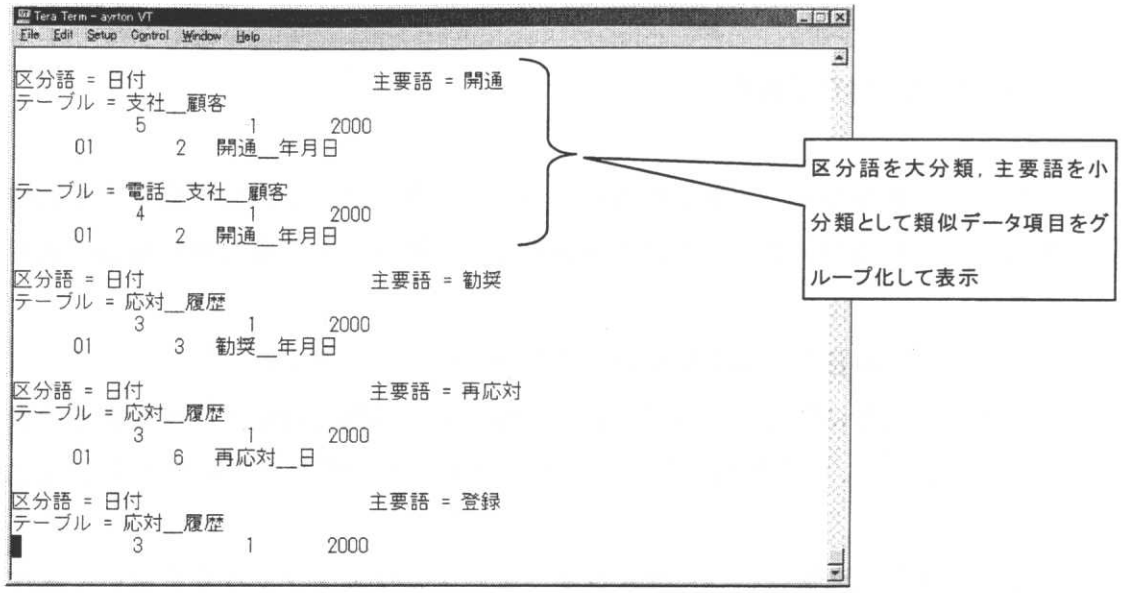
2.4.4. 定義・実行画面例

スキーマ構築方式の主要な特徴の一つである類似データ項目の分類支援機能の出力画面例と，連邦データベース管理機能の主要な機能である質問処理機能の実行画面例を示す．

類似データ項目の分類支援機能の出力例を図 2.6 に示す．

質問処理機能に関して，利用者による検索の指定例と対応する検索候補の表示例とを図 2.7 に示す．検索候補の選択例と検索結果の表示例とをそれぞれ図 2.8, 2.9 に示す．

なお，実行画面例は，ユーザインタフェースにウェブブラウザを用いて試作したデモンストレーション用検索機能の画面での例である．



注）外部ツール（データ標準化支援ツール Dbprompt/NAME）を利用．

図 2.6 類似データ項目の分類支援機能出力画面例

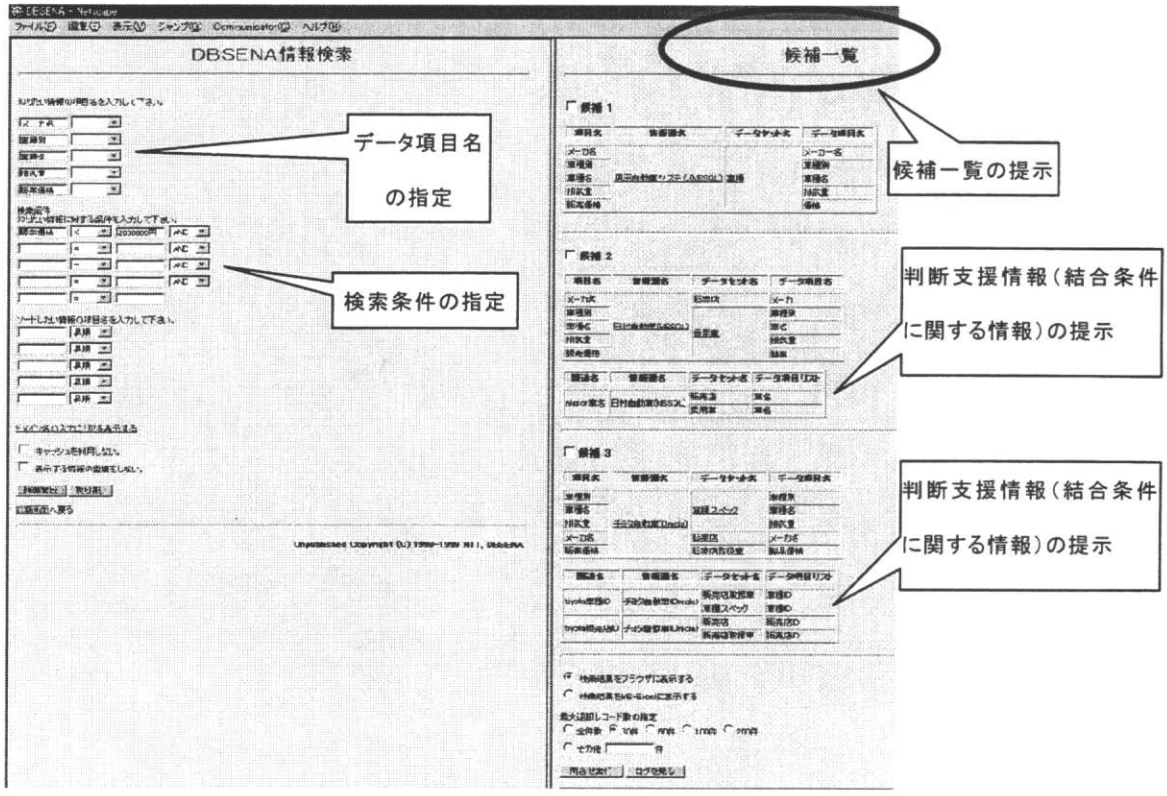


図 2.7 検索の指定例および検索候補の表示例

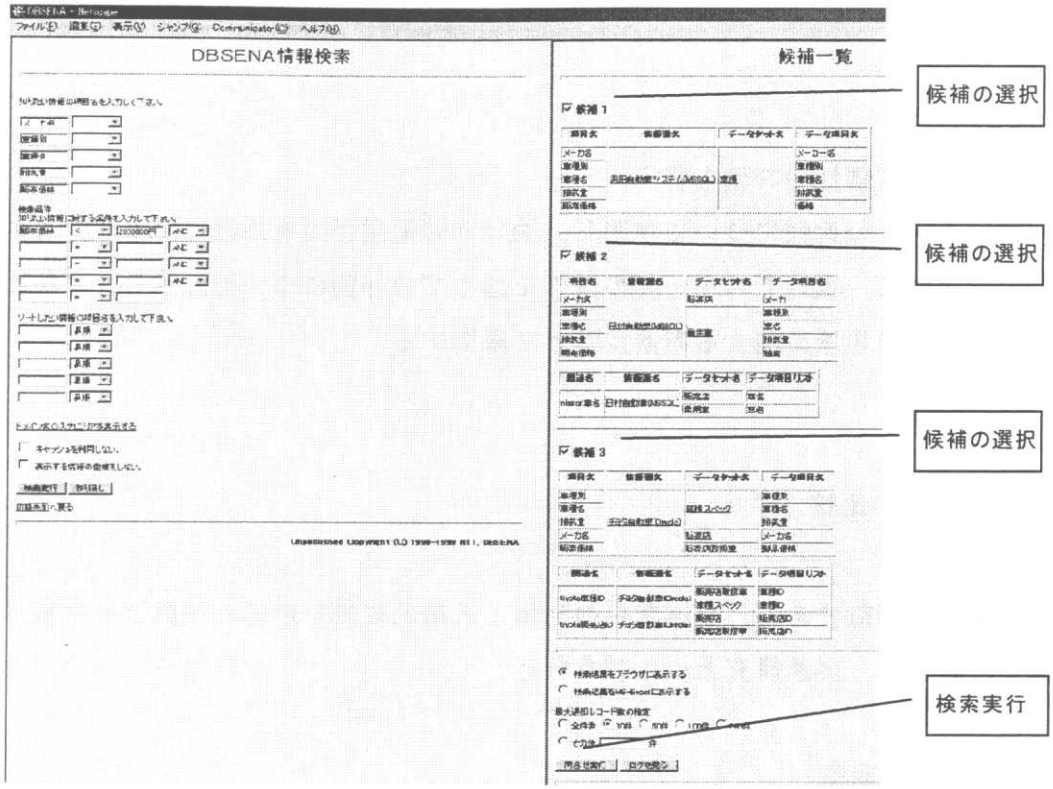


図 2.8 検索候補の指定例

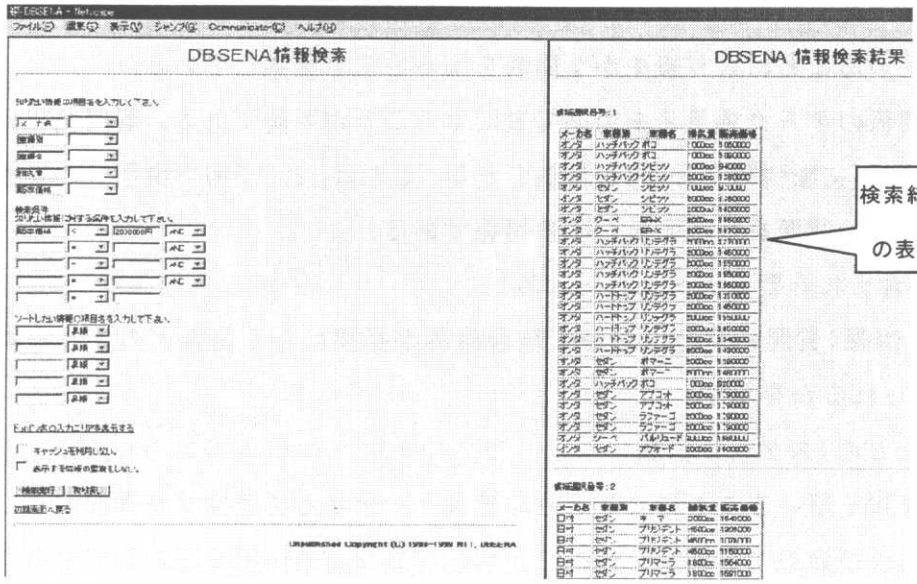


図 2.9 検索結果の表示例

2.5. 評価

提案する方式の有効性の評価を行う.

先ず，従来研究との比較について説明し，次いで普遍関係の有効性に関する定性的評価に関して説明し，次いでスキーマの試作を通しての評価について説明し，最後に本研究の成果の適用領域に関する考察について説明する.

2.5.1. 従来研究との比較

本研究の主たる特徴である，普遍関係の利用と汎用的知識を用いた類似データ項目分類支援とに関して，関連研究との比較を行う.

2.5.1.1. 普遍関係の利用

普遍関係を利用者インタフェースとして利用する研究を中心に関連研究を説明する.

Zaoら[Zhao95]は,製造業の複数社のデータベースシステムを連携する方法を提案している.そのような環境ではデータベースシステムの自律性が強く要求されることから，普遍関係を用いることにより緩やかな協調を図る方法を提案している.

利用者は普遍関係のデータ項目名を用いて質問することが可能である.質問結果には以下のような情報(文脈情報と呼ぶ)が付加される.これは，2.3.3節で説明した普遍関係における曖昧性の問題を解決するための情報である.

- データ項目が含まれる要素データベース名
- アクセスパス情報(質問の実行において結合演算が必要になる場合どのような結合演算が用いられるかを示す)
- (金額の単位などの)スケール情報

利用者は文脈情報を用いることにより自分の意図と一致する質問結果を選択する.以上により，利用者が簡単に質問文を作成し，解釈容易な結果情報を返却してもらうことが可能になる.

この方式の問題点は，二つある.一つは，データ項目レベルのスキーマ構築作業に関わるものである.この方式においては，データ項目間の関連を判定する際の支援機

能が無く，また，データ項目間の変換機能がすべて利用者記述(ルール或いは汎用プログラミング言語による関数作成)となっている.データ項目レベルのスキーマ構築作業が筆者の方式と比べて同等あるいは容易とは考え難い．もう一つは質問処理の性能に関わるものである．筆者らの提案方式では，要素データベースシステムへの問合せが実行されるのは検索候補のうちの利用者の意図に合致するものだけであるのに対して，この方式では，要素データベースシステムへの問合せが実行されるのは全ての検索候補である．これは，大きな性能劣化を招く危険性がある.

Reckら[Reck96] [Reck97]は，異種情報源アクセスへの透過的なアクセス(情報源の種類･所在などを気にせずに済むアクセス)を可能とするためのアーキテクチャを提案している.その一部として,普遍関係に基づく利用者インタフェースを提案している.

Reckらのアプローチは，スキーマを構築するアプローチではなく，メディエータ(mediator)と呼ばれる意味的異種性を解消するソフトウェアを複数構築することにより，複数情報源へのアクセスを可能にするアプローチである.

スキーマの構築にほぼ相当するのは，メディエータの作成作業と情報源をアクセス可能とするためのラッパ(wrapper)と呼ばれるソフトウェアを作成する作業とである.

この方式の問題点は二つある．一つは，データ項目レベルのスキーマ構築作業に関わるものである．ラッパ機能の一部として値の表現形式の異種性解消のための変換機能を含むとあるが,変換機能作成方法や事前の変換機能の品揃えに関する記述は無い.筆者の方式(事前に標準の変換関数を提供する方式)に比べて同等あるいは容易に変換機能を作成可能とは考え難い．もう一つは，普遍関係における曖昧性の問題を解決するための手段が示されていないことである.

また，普遍関係を用いないスキーマ構築の研究においては，要素データベースシステムのスキーマの共通データモデルへの変換作業と，実体間の関連特定作業と，関連を特定された実体を基にテーブル統合を行う作業とを行う必要があるという問題がある.

2.5.1.2. データ項目間の関連の特定

1章において汎用的な知識を利用するデータ項目間の関連特定方法として，Yuら[Yu91]の概念階層(concept hierarchies)を利用する研究と，Brightら[Bright91]の用語のネットワークを利用する研究と，Colletら[Collet91]の知識ベース (Cyc) を利用する研究とを説明した.

ここでは，上記の研究と筆者の研究とを比較する．

・ Yuらの研究との比較：

Yuらの研究に固有の問題として，データ項目毎のｎ次元ベクトルの作成を基本的には手作業で作成することを前提としており，概念数が多い実際の現場においては大きな作業量が必要になるという問題があると考える．半自動化の方式を提案しているが，方式の説明が詳細度に欠けており，その方式によってどの程度稼動が削減されるのか不明である．

・ Brightらの研究，Colletらの研究との比較：

彼らの研究も筆者らの研究も，名称の類似性に着目して，関連を有するデータ項目の候補を導く点は同じである．名称の類似性の判定方法の観点からは，Brightらの方法では，複合語の名称を扱えないという欠点がある．Colletらの研究では，名称が複合語の場合に名称を形態素解析せずに類似性を判定しており，関連を有するデータ項目が候補として提示される確率が低いという欠点がある．

また，概念階層，用語のネットワーク，知識ベースなどの汎用的な知識を用いることをしない関連特定の研究においては，汎用的な知識を用いる場合と同程度の精度および効率で類似データ項目を分類するのは困難であるという問題があると考える．

2.5.2. 普遍関係の有効性に関する評価

普遍関係に関しては，全てのデータ項目が単一の関係（普遍関係）に属するようになるため，データ（データ項目）の意味，データ間の関連などがわかりにくくなり，アプリケーションの構築が却って困難になるという批判がある[Codd88]．筆者もその批判に同意するものである．データ項目数がいくつの場合からアプリケーションの構築が困難という問題が顕在化するのかに関しては研究者間の合意はないが，筆者は，データ項目名に関して特に条件を付けない場合，数十項目を越えるとアプリケーションの構築が困難という問題が顕在化するものと考える．

アプリケーションの構築が困難という問題が存在する状況においては，スキーマ構築が容易化される効果が，アプリケーション構築稼動の増加によって打ち消され，データベースシステム全体の構築稼動は却って増加する可能性がある．このような状況においては，普遍関係を採用した効果が無くなる．

本節では，本研究の有効範囲が広いことを示すため，データ項目数が２００程度の

場合までは，普遍関係を用いるのと通常の関係を用いるのとでほぼ同程度の稼動でアプリケーションを構築可能なケースが多々あることを論じる．

データ項目の集合がある条件を満たすならばデータ項目の数が２００程度までは提案方式が有効となると言えること(1)と，その条件を満たす場合が多々あると言えること(2)との２段階に分けて説明する．

(1)社内のデータベースシステム設計・構築経験者４名にヒアリングを行った結果，彼らの設計・構築経験から判断するに以下の普遍関係の有効性に関する見解は成立するとの見解を得た．

・ 連邦スキーマのデータ項目の集合が，もし以下で定義する「理解容易性条件」を満たしかつデータ項目の数が約２００を越えないならば，各データ項目の意味は容易に理解可能であり，従ってデータベース検索・更新文の作成にあたってどのデータ項目を用いればよいかの判断が容易であり，このことより普遍関係を用いるのと通常の関係を用いるのとでほぼ同程度の稼動でアプリケーションを構築可能である．

ここで，「理解容易性条件」とは以下の２条件であると定義する．

・ 個々の用語（データ項目名を構成する用語）が，理解容易であり，かつ意味のぶれが少ない．
・ 各データ項目名がそれらの用語を基に，一定の命名規則（例：修飾語*+主要語+区分語）に基づき，命名されていること．

(2)また，データ項目名の集合が理解容易性条件を満たすように構成できる場合は多々あると考える．

企業が情報資源管理の考え方（情報を企業経営資源の一つとみなして，その完全性を維持し，有効利用を図る考え方．情報資源管理の方法論の一部に用語の統制やデータ項目名の標準化などが含まれる．）[堀内 96]に基づき，業務に用いる用語の統制及びデータ項目名命名規則の制定を行っているならば，その企業において提案方式に基づき連邦スキーマを構築する際に，統制された用語を用いることにより理解容易性条件の第１の条件が保証され，制定された命名規則を用いることにより第２の条件が保証され，従って，理解容易性条件を満たすように，データ項目名の集合を構成することが可能となる．

ここで，IRM 研究会（Information Resource Management 研究会．日本における情報資源管理の研究会．主に企業の実務担当者が会員になっている会．）[堀内 96]の活動などにみられるように，情報資源管理の考え方に基づき，業務に用いる用語の統制

及びデータ項目名命名規則の制定を行っている企業が多々あることが知られている[中村 87] [林 95].
以上をまとめると，提案方式の連邦スキーマのデータ項目の数が２００程度までは，提案方式が有効となる場合が多々あると言えるものと考える.

ここで，企業における用語が統制されている場合には，同義データ項目の発見作業においては，提案方式（類義の用語を主要語，区分語として用いるデータ項目同士をグルーピングし，それらを精査する）を用いる必要はなく，例えば単にデータ項目名の同一性だけ判断すれば十分のように一見して思えるがそうではない．そうではない理由を以下に示す．２つの理由がある.

(a)異なるデータ項目名が同義の場合がある.

大きく二つの場合がある.

- データ項目名を構成する用語同士が類義語で，精査すると同義の場合

  次節で説明する試作において実際に現われた例で説明する.代表標準用語が「識別」である類義語のグループの中には，「ID」と「コード」がある．一つの要素データベースシステムのデータ項目「顧客 ID」と，他の要素データベースシステムのデータ項目「顧客コード」とが同義であることがありうる.

- データ項目名の修飾語有無が異なるが，精査すると同義の場合

  次節で説明する試作において実際に現われた例で説明する．一つの要素データベースシステムには「通信料金」（修飾語無し）なるデータ項目が存在し，他の要素データベースシステムには「当月通信料金」,「前月通信料金」なるデータ項目が存在し，「通信料金」と「当月通信料金」が同義であることがありうる.

(b)異義のデータ項目を，連邦スキーマにおいては同一視したい場合がありうる.

次節で説明する試作において実際に現われた例で説明する．一つの要素データベースシステムには「企業コード」（修飾語無し）なるデータ項目が存在し，他の要素データベースシステムには「法人営業用企業コード」なるデータ項目が存在し，後者の値の集合が前者の値の集合の部分集合である（即ち，意味が異なる）場合がある．しかしながら，連邦スキーマでは，それぞれのデータ項目を同一視し，「企業コード」データ項目として見せたい場合がある.

(a)(b)いずれの場合も，同義データ項目の発見作業において，類義の標準用語を主要語，区分語として用いるデータ項目同士をグルーピングし，それらを精査する必要がある.

2.5.3. 試作を通しての評価

以下の方針で評価を行う.

先ず, DBSENA使用時, 未使用時のそれぞれの場合でスキーマの試作を行い, スキーマ構築稼動を評価する. 次いで, 提供した機能のうち, 必要に応じて機能の追加を予定している機能に関して, 十分性・追加容易性を評価する.

2.5.3.1. スキーマの試作

複数の既存社内データベースシステムを基に, DBSENA利用時, 未使用時のそれぞれの場合でスキーマの試作を行い, スキーマ構築稼動を評価する. なお, 一部の稼動は推定値 (例:値変換関数の作成の全稼動は, 一部の関数の作成稼動から推定) である.

既存社内データベースシステムの諸元を以下に示す.

- データベース数: 5
- 総テーブル数: 25
- 総データ項目数: 627

連邦スキーマの想定利用業務としては, 顧客管理を想定した.

DBSENA未使用時の構築方法を中心に, 試作におけるスキーマの具体的な構築方法を簡単に説明する.

(1)輸出スキーマ収集

DBSENA未使用時は各要素データベースシステムの輸出スキーマをエキスポートツールなどを用いてエキスポートし, そのデータをスキーマ入力・編集機能を用いて担当者が入力を行う.

DBSENA使用時・未使用時とも, 輸出スキーマ以外に存在する設計情報(設計ドキュメント中の, データ項目の自然言語での説明など)に関しては, スキーマ入力・編集機能を用いて担当者が入力を行う.

(2)スキーマ間関連特定

DBSENA 未使用時は担当者が自ら考案した以下のヒューリスティックスを用いて, 類似データ項目の分類を行った.

- 18のカテゴリ(例:電話, 日付, 住所)に項目を分類する.
- 各カテゴリ内では, 担当者の判断で類似項目のグルーピングを行う.

DBSENA使用時・未使用時とも，データ項目の各グループ内での項目間の関連（同義性）の判断は，ドメインの値および担当者の業務知識に基づき担当者が行う．ドメインの値および担当者の業務知識からでは判断困難な場合は，データ項目を含むデータベースシステムの管理者に，同義性の判断に必要な追加情報のヒアリングを行い，その追加情報を元に判断を行う．

なお，用語辞書の維持作業は，用語辞書の導入に伴い必要になった作業であることから，稼動の評価においては，DBSENA使用時のデータ項目の関連特定作業の一部と扱うこととする．

(3)統合スキーマ作成

DBSENA 使用時・未使用時とも，同義データ項目間の変換関数設定などを行う．必要ならば変換関数の作成を行う．

変換関数提供による効果を明確化するため，DBSENA 未使用時は変換関数をすべて新規作成要とした．新規作成要の関数の数と種類は，スキーマ間関連特定で，同義だが変換要と判明したデータ項目の組み合わせから導く．

また，普遍関係の採用による効果を明確化するため，DBSENA 未使用時はテーブル統合を行う．筆者が属する研究組織で開発した，データベース設計法[NTT96a][NTT96b]を用いてテーブル統合を行う．

なお，参照関係の投入作業は，普遍関係の導入に伴い必要となった作業であり，テーブル統合とトレードオフの関係にある作業と考えられる．従って，稼動の評価においては，DBSENA 使用時の参照関係の投入稼動は DBSENA 未使用時のテーブル統合稼動と比較することとする．

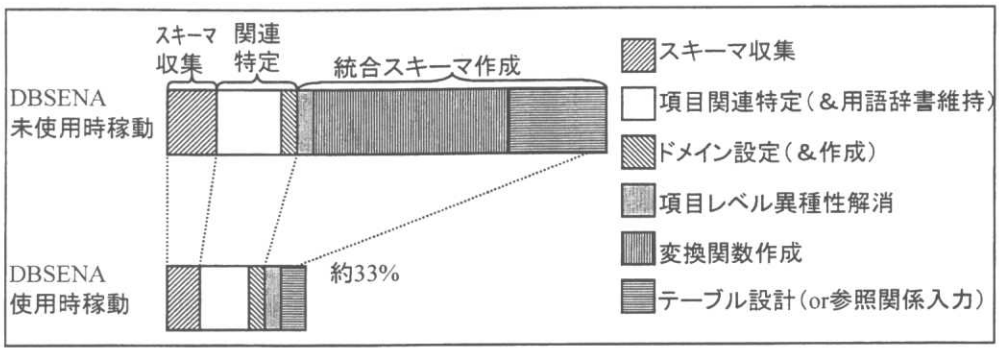


図 2.10　スキーマの構築稼動の比較

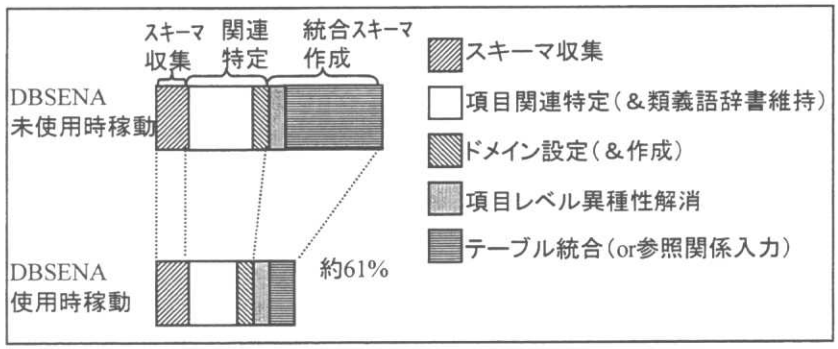


図 2.11　スキーマの構築稼動の比較（主要な特徴の効果）

2.5.3.2. スキーマ構築稼動の評価

スキーマ構築稼動の評価を２段階で説明する．

先ず，普遍関係の有効性の評価結果を踏まえ，その評価結果と整合するように稼動の補正を行ったので，そのこと（補正）に関して説明する．

次いで，補正された稼動値を用いて，試作においてスキーマ構築稼動が大幅に削減されたことを示す．

(1)稼動の補正

前節で説明したスキーマ間関連特定作業の結果得られた総データ項目数（同義データ項目の重複排除を行った結果のデータ項目数）は，普遍関係の有効性の評価の節で示した，普遍関係が有効となりうるデータ項目数の上限（約２００）を超えていた．そこで，以下のデータ項目を除外し，連邦スキーマの総データ項目数を２００以内に

収める補正を施した．スキーマ構築の全稼動の集計においては，除外したデータ項目に対応する稼動を減算して集計を行った．(2)のスキーマ構築稼動の評価においては，減算を施した修正稼動を用いて評価を行った．

・ 連邦スキーマの想定利用業務（顧客管理）とは独立な業務分野（顧客管理に属するデータ項目の集合との間に，キー・参照キーの関係にあるごく少数のデータ項目を例外として，互いに類義の関係にあるデータ項目が存在しないような業務分野．例：設備管理.）に属するデータ項目を除外した．
・ 「予備」「欠番」などの無意味なデータ項目を除外した．

除外したデータ項目は，顧客管理分野のデータ項目と独立性の高い（互いに関係の少ない）データ項目であり，従って以下の２つの稼動はほぼ等しいと言って差し支えないと考える．

・ 実スキーマ構築稼動から，除外したデータ項目に対応する稼動を減じた稼動．
・ 仮に，除外したデータ項目に対応する既存データベースシステムのデータ項目がもともと存在しないとした場合の，スキーマ構築稼動

(2)スキーマ構築稼動の評価

DBSENA未使用時と使用時のスキーマ構築稼動を比較する（図2.10).スキーマ構築稼動が大幅に削減され，約1/3になることが分かった．

提案方式の主要な特徴の効果をより明確化するため，以下の２点の補正を行った比較も併せて示す(図2.11)．提案方式の主要な２つの特徴により，スキーマ構築稼動が約４割削減されることが分かった．

・ DBSENA未使用時もスキーマ自動収集機能を用いると仮定した補正：スキーマ収集の手作業稼動を，自動収集機能の起動稼動に置換する．
・ DBSENA未使用時も項目値変換関数ライブラリを用いると仮定した補正：変換関数作成稼動を０とする．

スキーマ構築の構築プロセス毎に削減効果の分析を行う．

(a)輸出スキーマ収集

スキーマ自動収集機能の導入により，DBSENA未使用時に比べ輸出スキーマ収集の稼動を約1/3削減できた.要素データベースシステムの輸出スキーマの外部に存在する設計情報が多々ありDBSENA利用時も手入力稼動を大きくは削減できない現状においては，約1/3の削減は十分大きな削減であると考える．

(b)スキーマ間関連特定

用語辞書を用いた類似データ項目分類支援機能の導入により，DBSENA未使用時に比べスキーマ間関連特定の稼動を約1/4削減できた．

他の特徴に比べると相対的に削減効果は小さいが,約1/4の削減は有意義な削減であると考える．なお，他の特徴に比べて相対的に削減効果が小さかったのは以下の要因によるものと考える．

・ DBSENA未使用時の担当者が効率的なヒューリスティックスを考案した．
・ 収集した情報（スキーマ情報と設計ドキュメント）だけでは同義性の判断に必要な情報が不足しており，不足情報を得るために要素データベースシステムの管理者へのヒアリングが必要となったケースがいくつかあり，そのヒアリング稼動が大きかった．これは，DBSENA使用時・未使用時に共通的なオーバヘッドであるが，使用時の効果を相対的に小さくする方向に働く．

(c)統合スキーマ生成

DBSENA使用により，DBSENA未使用時に比べ統合スキーマ生成の稼動を約８割削減できた．

関数の作成稼動が，DBSENA未使用時の稼動の半分近くを占め，データ項目値変換関数ライブラリの導入による稼動削減効果は大きいと言える．10種類の関数を新たに作成した．主なものを以下に示す．

・ 電話番号，年号などのハイフンの削除/挿入
・ 金額の単位変換
・ 日付の書式変換

また，普遍関係の利用可能化の効果も大きいと言える．DBSENA未使用時のテーブル統合の稼動と比べると，DBSENA使用時の参照関係入力稼動は約1/4であった．

普遍関係利用可能化の効果は十分大きいものの，DBSENA使用時の参照関係入力稼動は当初予想よりは大きかった．大きくなった要因は以下と考える．

・ 収集した情報（スキーマ情報と設計ドキュメント）だけではキーに関する判断（キーはどのデータ項目か，キーが複合キーである場合構成データ項目は何か，など）に必要な情報が不足しており，不足情報を得るために要素データベースシステムの管理者へのヒアリングが必要となったケースがいくつかあり，そのヒアリング稼動が大きかった．

2.5.3.3. 機能の十分性・追加容易性の評価

提供した機能のうち，必要に応じて機能の追加を予定している機能に関して，十分性・追加容易性を評価する．

(1)用語辞書

新サービスの名称など約10語の追加が必要になった．

要素データベースシステムの管理者との確認をとりつつ，容易に，用語およびその類義語の登録を行うことができた．

今後も新製品･新サービスの追加や業務の進め方の変更などに伴い,用語は漸増し，それに伴い辞書への登録が必要となるが，スキーマの管理者が要素データベースシステムの管理者との連携を密にとることにより，用語の登録に困難は少ないと考える．

(2)データ項目値変換関数

全ての変換に必要な関数は，予め提供された関数或いは関数の組み合わせにより記述できた．

今後も必要に応じて追加する予定であるが,DB-STREAMの使用においても追加の必要が生じなかった経験をも踏まえ，追加の必要が生じる可能性は少ないと考える．

(3)スキーマ自動収集機能

既存データベースシステムの輸出スキーマは，全てスキーマ自動収集機能で収集できた．

今後も必要に応じて機能を追加する予定であるが，ビジネスで使用される関係データベース管理システムの種類が増えるとは考え難いことと，既存の関係データベース管理システムのスキーマ構造が変わることは稀であると考えられることから，機能追加の生じる可能性は殆どないと考える．

2.5.4. 適用領域に関する考察

前節までの評価結果より，提案方式は以下に示す広い適用領域を有することが分かる．

- 企業が情報資源管理を実施しており，連邦スキーマのデータ項目名を理解容易性条件を満たすように構築できる場合の，連邦スキーマのデータ項目数が約２００程度までの連邦データベースシステムの構築．

2.5.2 節で述べたように，情報資源管理を実施している企業は多々あることが知られており，このケースは多々存在すると考える．

・ （連邦スキーマのデータ項目名の集合が理解容易性条件を満たすよう構築できるか否かに関わらず)，連邦スキーマのデータ項目数が数十程度までの連邦データベースシステムの構築．

実際に社内外のいくつかのシステムに本研究の成果を適用しているが，社内の事例は第一のカテゴリに属するものであり，社外の事例は二つ目のカテゴリに属するものである．

それらのシステムにおいては，スキーマ構築が簡易化され，用語辞書の維持管理は容易であり，データ項目値変換関数とスキーマ自動収集機能の追加は不要であったことが確認されている．これは，提案方式の有効性を裏付けるものと考える．

2.6. 2章のまとめ

連邦データベースシステムにおけるスキーマの構築の簡易化を狙いとして，新たなスキーマ構築方式を提案した．その特徴は，①実体レベルの作業を簡易化することを狙いに，関係データベースにおける質問の簡易化のために導入された概念である普遍関係を採用することと，②データ項目間の関連特定作業を簡易化することを狙いに，用語辞書を用いて類似データ項目の分類を支援する機能を備えることである．

提案方式を実装する連邦データベース検索システムDBSENAを作成し，さらにDBSENAを用いて複数の既存社内データベースシステムを連携するシステムを試作し，提案方式の有効性を評価した．

提案方式の特徴を有しないスキーマ構築方式（実体レベルの構築作業を含み，類似データ項目の分類支援機能を有しない構築方式）を用いて構築した場合に比べて構築稼動が約4割削減されることを示した．

また，上記特徴の採用に伴い，用語辞書の維持管理稼動が必要になるが，その維持管理稼動はさほど大きくないと期待できるが分かった．

提案方式の適用領域に関しては，以下に示す広い適用領域を有することを示した．

・ 企業が情報資源管理（情報を企業経営資源の一つとみなして，その完全性を維持し，有効利用を図ること．情報資源管理の方法論の一部に用語の統制やデータ項目名の標準化などが含まれる）を実施しており，スキーマのデータ項目名の集合を各データ項目の意味が容易に理解可能なように構築できる場合の，総データ項

目数が約２００程度までのスキーマの構築.
情報資源管理を実施している企業は多々あることが知られており，このケースは多々存在すると考える.
（スキーマのデータ項目名の集合を各データ項目の意味が容易に理解可能なように構築できるか否かに関わらず),総データ項目数が数十程度までのスキーマの構築.

3. 企業アプリケーション統合におけるデータ交換システムの構築方式

3.1. まえがき

本章では，データ交換システム(data exchange system)の構築方式について論じる[池田97].

１章で説明したように，データ入力作業の軽減，二重入力の回避，データの品質向上，企業内を流れる情報の速度の向上などを狙いとして，企業アプリケーション統合（EAI:Enterprise Application Integration）の技術が研究・開発されている．EAIは，データレベルのEAIと，アプリケーションインタフェースレベルのEAIと，メソッド（共有および再利用可能なロジックをメソッドと呼ぶ）レベルのEAIと，ユーザインタフェースレベルの技術のEAIとに大別される．データ交換システムとは，データレベルのEAIのうち，データベースシステム間で直接データを交換する形態を言う．

様々なデータベースシステム間でのデータ交換を可能にするためには，データ交換システムは，交換元と交換先のデータベースシステム間に存在するデータ異種性を解消し，交換元と交換先のデータベースシステムの自律性を維持する必要がある．

ここで，１章でも述べたように，自律性維持の一つの形態として，交換先のデータベースシステムの既存機能や性能への影響を極小化しつつデータ交換を行う必要があるケースが考えられる．典型的なケースとして，交換先のデータベースシステムがレガシーシステム[Brodie95]であり，そのため既存機能の更改が困難であり，データ交換システムの構築時に，交換先のデータ入力方法を改造無しに用いることを要求されるケースがある．既存データ入力方法には，下記に挙げる制約を有するものがしばしば見られる．

・ 複数関連データの一括入力

特定の関連を有する，一般には可変個の関連データをまとめて入力しなければならないという制約．

例えば，参照関係の基数が１対多（多は可変とする）という意味制約を有するレコードA, Bを考える．交換元の１トランザクションで１つのAと複数（可変個）のBを作成および挿入可能であり，交換先ではそれらの関連データをまとめて入力しなければならないという制約が考えられる．

この課題に対する解決法を与えた研究・製品は見当たらない．個別にプログラミングすることによって構築されたデータ交換システムが見られるだけである．それらの

データ交換システムにおいては，プログラムロジックによって交換先のデータベースシステムでの入力方法への影響の極小化を実現している．それらのデータ交換システムは汎用言語でプログラミングされていることから，構築の稼動・期間が大であるという問題がある．

以上より，交換元・交換先データベースシステム間のデータ異種性を解消することと，交換先データベースシステムの自律性を維持すること，特に交換先のデータベースシステムの既存入力方法への影響を極小化しつつデータ交換を行うこととを可能にするデータ交換システムの簡易な構築方式が求められる．

筆者は，これらの要求に応えるために，先ず，①様々な形態のデータ交換に必要な機能のモデル（データ交換処理モデルと呼ぶ）を明確化し，次いで，②このモデルに沿ったデータ交換システムの開発・実行環境を提案する．さらに提案方式を実現するデータ交換システム開発・実行環境DB-STREAMの試作を行い，実際のデータ交換に適用し，上記①②の有効性を検証する．

データ交換処理モデルは，形式的なデータと機能のモデルの考案と，それに基づく操作機能の導出とを特徴とする．

データ交換システム開発・実行環境は，データ交換の仕様を簡易に記述できる高水準仕様記述言語を設けることと，基本的な操作機能および項目値変換関数の部品化および再利用を可能とすることとを特徴とする．

3.2. データ交換処理モデルへの要求条件

本章では，図3.1に示すように，データ交換システムを，1つ以上のデータベースシステム（交換元データベースシステムと呼ぶ）からデータ（および必要に応じて関連するデータ）を抽出・変換して，他の1つ以上のデータベースシステム（交換先データベースシステムと呼ぶ）へ該データを格納するシステムと定義する．なお，以降では「交換元データベースシステム」及び「交換先データベースシステム」をそれぞれ単に「交換元」及び「交換先」と略して使用する．

データ交換処理モデルへの要求条件は，交換元・交換先データベースシステム間のデータ異種性の解消に関する要求条件と，交換先データベースシステムの自律性の維持に関する要求条件に大別される．

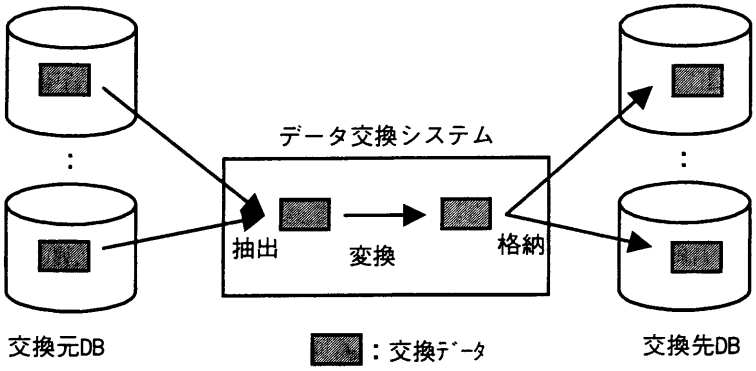


図 3.1 データ交換システム

3.2.1. データ異種性の解消

データ異種性の分類については，連邦データベースシステム[Sheth90]において，スキーマ定義言語/質問言語に要求される記述能力を明らかにすること等を狙いとして，関係データモデルにおけるデータ異種性を分類したKimの分類（１章表1.1参照）が知られている[Kim93].

データ交換システムは，表1.1に挙げたデータ異種性を解消する機能が要求される.

3.2.2. 自律性の維持

(1)交換先データベースシステムの既存機能への影響の極小化:複数関連データの交換可能化

まえがきで述べたように，交換先のデータベースシステムがレガシーシステム[Brodie95]であり，そのため既存機能の更改が困難であり，データ交換システムの構築時に,交換先のデータ入力方法を改造無しに用いることを要求されるケースがある.

既存データ入力方法のデータ構造は単純なもの（例：１レコードずつ入力）ばかりとは限らず，複雑なデータ構造の場合（例：可変個のレコードをひとまとめにして入力）もありうる．いずれの場合にも，既存データ入力方法を改造無しに利用可能とすることが求められる.

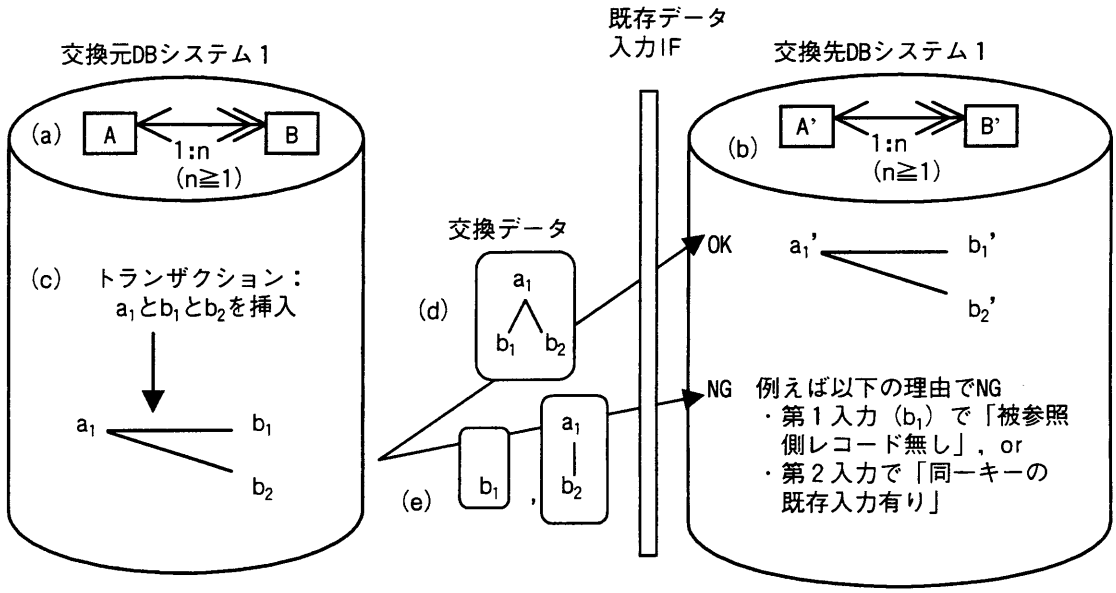


図 3.2 交換先データベースシステムでの入力方法例

例えば，図3.2の(a)のように交換元にレコードA，Bが存在し，A，B間には参照関係の基数が１対多（多は可変とする）という意味制約が存在したとする．一方，交換先にも図3.2の(b)のように，交換元と同様に，レコードA′，B′が存在するとする．ここで，図3.2の(c)のように，交換元のトランザクションにおいて，意味制約を満たす３つのレコードオカレンス（a_1，b_1，b_2の３つ．b_1，b_2はa_1への参照キーを含むとする）を作成および挿入することを考える．データ交換システム構築以前は，図3.2の(d)のように，a_1，b_1，b_2をまとめて交換する入力方法になっていたとする．この場合，データ交換システム構築後もa_1，b_1，b_2をまとめて入力させる必要がある．図3.2の(e)のように，a_1，b_1，b_2を分割して交換すると，交換先では「被参照側のレコード無し」あるいは「同一キー値での既存入力有り」などの理由でエラーとなり，データベースシステムへの反映を行うことができない．なお，この例では，A，Bが同一データベースシステムに存在する場合の例を挙げたが，実際にはA，Bが異なるデータベースシステムに存在する場合もありうる．

この例で挙げたように，複数の互いに関連するレコード群をまとめて交換し，既存データ入力方法で入力可能とすることを，これ以降，複数関連データの交換可能化と呼ぶ．

1章で述べたように，既存の研究におけるデータ定義能力は関係データベースシステムのデータ定義機能（繰返しを含まない，フラットなデータ構造の定義機能）を大きく超えるものでは無く，また，既存の研究におけるデータ変換能力は，関係データベースシステムの関係演算機能（選択，結合など）と，データ型変換機能や対応表変換機能等のデータ項目レベル変換機能とを合わせた機能を大きく超えるものでは無い．従って，既存の研究においては，この要求条件に応えるために必要な，可変回繰返し構造体を含むデータ構造の定義，可変回繰返し構造体を含むデータ構造の作成，可変回繰返し構造体間のデータ変換，などの機能は有していない．

この要求条件に対する解決法としては，個別にプログラミングすることによって構築されたデータ交換システムが見られるだけである．それらのデータ交換システムにおいては，プログラムロジックによって交換先のデータベースシステムでの入力方法への影響の極小化を実現している．それらのデータ交換システムは汎用言語でプログラミングされていることから，構築の稼動・期間が大であるという問題がある．

(2)交換元/先データベースシステムの性能への影響の極小化

(a) 交換処理得量の削減

交換処理の負荷削減のため，及び交換処理時間の短縮のためには，全体的な交換処理量を削減することがデータ交換システムに要求される．

(b) 既存センタへの影響軽減

一般にデータベースシステムは性能的に最適設計されており，データベースシステムの搭載センタによっては性能的な余裕が少ない場合がある．この場合，データ交換処理のうち既存データベースシステムの搭載センタ上で走行させる処理を最小限にする必要がある．

(c) 高負荷時間帯回避

同様に，データベースシステムによっては高負荷時間帯に処理を実行させることが許されない場合がある．この場合，既存データベースシステムの高負荷時間帯を回避して交換処理を実行させる機能が必要である．

3.3. データ交換処理モデル

本節では，前節の要求条件を満たすために，データ交換システムが具備すべき操作機能を抽出・整理することを狙いとして，データ交換処理モデルを提案する．

データ交換処理モデルは，形式的なデータと機能のモデルの考案と，それに基づく操作機能の導出とを主な特徴とする．

3.3.1. 交換データの単位

複数関連データの交換可能化のためには，関連するレコード群を一つのまとまりとして定義し管理できるようにする必要がある．そこで，交換ユニットと呼ぶ交換データの単位を新たに設ける．交換ユニットUの定義を以下に示す．なお，これ以降の形式的な記述方法は，植村[植村 79]，数学辞典[数学辞典 85]に従う．

$$U(R_1,\ldots,R_n,PredU) \equiv \{<r_1,\ldots,r_n> \mid r_1 \subseteq D_{11}\times\ldots\times D_{1m_1} \wedge \ldots \wedge r_n \subseteq D_{n1}\times\ldots\times D_{nm_n} \wedge PredU(r_1,\ldots,r_n)\}$$

・・・式(1)

ここで，$R_1,\ldots,R_n$ は関係を表わし，$D_{i1},\ldots,D_{im_i}$は関係R_iがその直積の部分集合になるような定義域を表わし，PredUは$P(D_{11}\times\ldots\times D_{1m_1})\times\ldots\times P(D_{n1}\times\ldots\times D_{nm_n})$ 上の計算可能な述語を表わす．なお，P(S)は集合Sのべき集合，×は集合の直積を表わす．

交換ユニットの個々の要素を交換ユニットオカレンスと呼ぶ．曖昧性がなければ，交換ユニットオカレンスを単に交換ユニットと呼ぶこともある．

交換元データベースシステムで送出できる交換ユニットを交換元ユニットと呼び交換先データベースシステムで受け取りたい交換ユニットを交換先ユニットと呼ぶ．

交換先ユニットの場合はデータベース中に存在しないデータであるため，交換ユニットUの定義にはR_iを用いずに$D_{i1},\ldots,D_{im_i}$を用いている．式(1)のPredUは，交換ユニットUが満たさなければならない制約を表す述語である．

交換ユニットの定義例を示す.今回のトランザクションで更新された関係Aの１レコードについて，そのレコードの更新前イメージ（これ以降BIと略す）と更新後イメージ（これ以降AIと略す）と，そのレコードを参照する関係B（AとBの参照関係の基数を１対nlとする）のレコードとを一まとめにする，交換元ユニットU(A,B,PredU)は以下のように定義できる．

先ず,トランザクションによる関係の変化を表現するために以下の概念を導入する．

R(t)≡トランザクションtのコミット時点における関係R

ここで，tはトランザクションの直列順序を表わし，tの値は１から始まる自然数値とする.トランザクションtのコミット時点で未コミットのトランザクションによる変更はR(t)には含まれないものとし，あるレコードがR(t)に含まれるか否かは，更新ロ

グなどの手段を用いることにより，交換元において判定可能とする．R(t)を用いることによりPredUを以下のように定義できる．

PredU(a,b)≡

Card(a)=2∧Card(b)=n1∧　　/*aのオカレンスは２つ．bのオカンレスはn1*/

$\exists a_i \in a(a_i \in A(current) \wedge \neg a_i \in A(current-1)) \wedge$

$\exists a_i \in a(\neg a_i \in A(current) \wedge a_i \in A(current-1)) \wedge$　/*aはBI,AIを一つずつ含む*/

$\forall a_i \in a \ \forall a_j \in a(a_i.$キー値$= a_j.$キー値$) \wedge$　　/* BI,AI のキー値は等しい*/

$\forall b_j \in b \ (b_j \in B(current)) \wedge$　　/*bのオカレンスは全てB(current)の要素*/

$\forall a_i \in a \ \forall b_j \in b \ (b_j.$参照キー値$= a_i.$キー値$)$　/* b_jの参照キー値はa_iのキー値と等しい */

ここで，Card(S)は集合Sの濃度を表わし，currentは今回のトランザクションを表わす．

3.3.2. データ交換可能性

ここでは，データ交換における入力データと出力データとの間の関係を形式的に表現する概念として，データ交換可能性の概念を導入する．

ある交換元ユニット$V_1(S_{11},\ldots,S_{1p_1},PredV_1)$,..., $V_k(S_{k1},\ldots,S_{kp_k},PredV_k)$（但しkは有限）から交換先ユニット$U(R_1,\ldots,R_n,PredU)$へデータ交換可能（単に交換可能とも呼ぶ）とは，以下の計算可能な述語Predと，計算可能な関数fが存在することを言う（図3.3，図3.4）．述語Pred，関数ｆの存在はしかるべき期間成り立つものとする．

$f:Pred(P(V_1),\ldots,P(V_k))\rightarrow P(P(D_{11}\times\ldots\times D_{1m_1})\times\ldots\times P(D_{n1}\times\ldots\times D_{nm_n}))$，かつ

$$\bigcup_{v\in Pred(P(V1),\ldots,P(Vk))} f(v) \quad \cap \quad U(R_1,\ldots,R_n,PredU) \neq \phi \quad \cdots 式(2)$$

ここで，Pred はP(V1)×...×P(Vk)上の述語を表わし，Di1,...，Dimiは関係Riがその直積の部分集合になるような定義域を表わす．なお，Pred(P(V1),...，P(Vk))は，述語Predを満たすP(V1)×...×P(Vk)の部分集合を表わす略記である．定義を以下に示す．

Pred(P(V1),...,P(Vk))

≡ {<v1,...,vk>| v1∈P(V1)∧...∧vk∈P(Vk) ∧Pred(v1,...,vk) }

関数fの定義域を，V1×...×Vkではなく，各Vjのべき集合を用いて定義した理由は，

同一交換元ユニットの複数オカレンスから，１つの交換先ユニットオカレンスを作る必要があり得るからである．

関数fの定義域を，述語Predを用いて定義した理由は，$P(V_1)\times\ldots\times P(V_k)$の中で交換して意味のある組み合わせを選び,そうすることによって計算可能なfが存在できなくなるのを回避するためである．例えば，実世界における一つの実体の属性群が，交換先では一つの関係で管理されているが，交換元では，異なる交換元の複数の関係で管理されている場合がある．この場合，v_1.キー値=...=v_k.キー値（但し，$\{v_i\}\in P(V_i)$）なる述語を満たす<$\{v_1\},\ldots,\{v_k\}$>の組み合わせ以外を交換する必要はない．

関数fの値域を，単なる直積$P(D_{11}\times\ldots\times D_{1m_1})\times\ldots\times P(D_{n1}\times\ldots\times D_{nm_n})$ではなくそのべき集合に含まれるとした理由は,図3.5の例のように一つの交換元ユニットオカレンスから複数の交換先ユニットオカレンスを作る必要があり得るからである．

式(2)の「かつ」以降の条件式を，$\cup f(v)=U(R_1,\ldots,R_n,PredU)$としなかった理由は以下の通りである．$U-\cup f(v)=\phi$でない理由（図3.4の領域Xが存在する理由）は，例えば，交換先データベースシステムが全国データベースシステムで交換元データベースシステムが支店データベースシステムの場合のように，交換先で必要な全オカレンスを交換元から導けるとは限らないためであり，$\cup f(v)-U=\phi$でない理由（図3.4の領域Yが存在する理由）は，実世界の意味制約のうち，交換先の方がより多くの制約を具現化している，即ち，交換先での制約が交換元よりも厳しい場合があり得るからである．

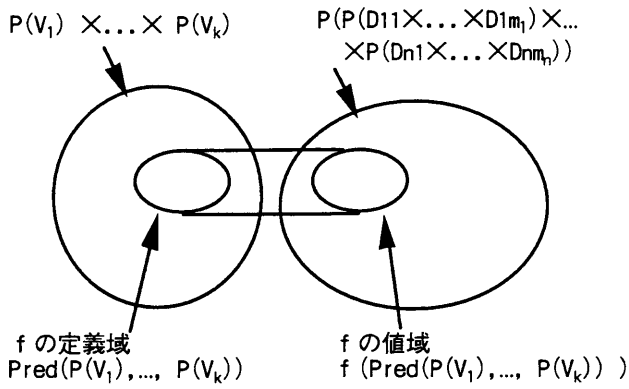


図 3.3 交換可能性の説明

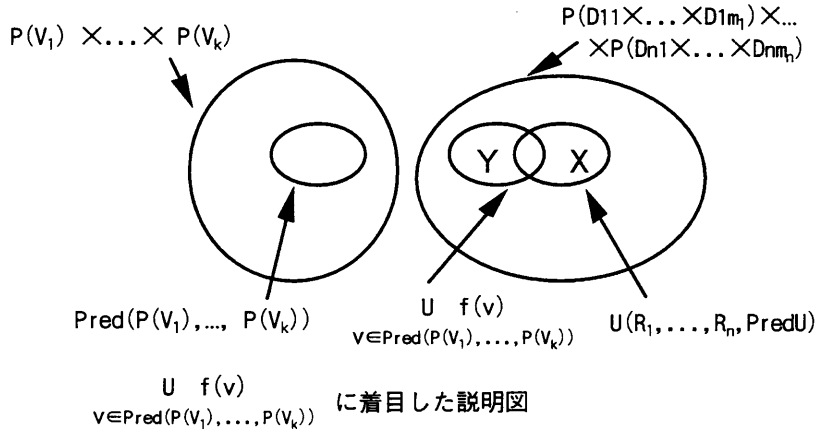


図 3.4 交換可能性の説明（続き）

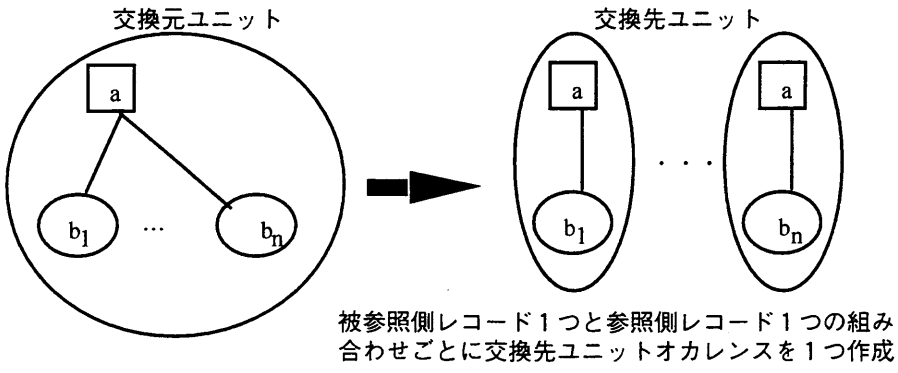


図 3.5 複数の交換先ユニットオカレンスの作成例

複数関連データの交換可能化及びデータ異種性の解消の観点からは，ここで提案した交換可能性の定義は十分汎用的であると考える．汎用性を，関数fの定義域の記述方法（記述可能な範囲）の汎用性，即ち交換先ユニットを作成可能な交換元ユニットの組み合わせが，関数fの定義域として記述可能なことと，関数fの汎用性，即ち複数関連データをひとまとめにする機能とデータ異種性解消機能とが関数fの集合に含まれることとの２つの観点から論じる．

(1)関数fの定義域の記述方法の汎用性

式(2)において交換元ユニットの数kの制約は有限であることだけであり，述語Predの制約は計算可能であることだけである．従って，１つの交換先ユニットとその元となる交換元ユニットとの対応が最も複雑な場合，即ち，全交換元の全交換元ユニットの複数オカレンスが対応し，かつそれら複数オカレンスの，特定の述語を満たす組み合わせのみが対応する場合でも，その交換元ユニットの組み合わせとその特定の述語とを関数fの定義域の定義に記述可能である．

上記より，関数fの定義域の記述方法(記述可能な範囲)は十分汎用であると考える．

(2)関数fの汎用性

関数fの制約は計算可能であることだけである．従って，複数関連データをひとまめとめにする機能とデータ異種性解消機能とは，計算機上で実現可能な機能である限りは，必ず関数fの集合に含まれる．関数fの集合は十分汎用であると考える．

ここで，交換元の実体の集合と，それらに関する統計値（平均，最大など）を含む交換先の実体との間の変換関数も関数fの集合に含まれる．従って，様々な統計的データを必要とするデータウェアハウスの構築においても，交換可能性の概念は適用可能である．

3.3.3. データ交換の操作機能

本節では，データ交換を計算機上で実現するのに必要な機能を明確化する．先ず，データ交換可能性概念を実現するのに必要な機能を明確化し，次いで，異なるセンタ間で交換ユニットを送受信する機能など，それ以外の必要な機能を明確化する．

データ交換可能性概念を計算機上で実現するためには，以下の３つの要素機能が必要である．

・ 式(1)，式(2)のPredU或いはPredを評価する，述語評価Ｐ

・ 関数ｆの計算を行う，変換Ｆ

・ 複数の交換ユニットの部分集合の直積を作成する，統合Ｃ(なお，これ以降，統合の説明において，曖昧性が無ければ，「交換ユニットの部分集合」を単に「交換ユニット」と略す．)

データ交換可能性が成立する場合の（複数の）交換元ユニットから交換先ユニットへの交換機能は，上記３つの要素機能をＣＰＦＰと並べた一塊の処理機能で実現できる．しかし，この各要素機能を固定順序に並べた処理機能だけでは，交換元ユニットと交換先ユニットの組み合わせ毎に交換システムを構築する必要が生じる．以下ではデータ交換の全体的な交換処理量を削減する（3.2.2(2)(a)項）ために，各要素機能が備えるべき機能の詳細と各要素機能間の可能な組み合わせ方とについて論じる．

(1)交換ユニットの共用化・交換処理の共通化

異なるデータ交換において設計されたそれぞれの交換元ユニットが同じであり，かつ交換処理の先頭から途中までが同じ処理である場合，交換処理の同一処理部分を共通化し途中からの分岐を可能とすることにより，交換処理量を大幅に削減できる．このためには，３つの要素機能をそれぞれ独立にし，互いに組み合わせ可能にしておく必要がある．

さらに，異なる変換関数の入力データが同一の場合，あるいは異なる述語の入力データが同一の場合，入力データの読み込み処理を共通化することにより，処理量の削減が図れる場合がある．即ち，変換Ｆにおいては複数種類の変換関数を，また述語評価Ｐにおいては複数種類の論理式の評価を行えるようにしておくことも処理量削減に効果がある．

ここで,交換元ユニット,交換先ユニット以外の中間データを「中間交換ユニット」，各要素機能の入力，出力となる中間交換ユニットをそれぞれ「入力交換ユニット」，「出力交換ユニット」と呼ぶことにする．中間交換ユニットの数学的な定義は，式(1)におけるPredUを恒真な述語に固定したものである．

(2)要素機能の実行順序を自由とする

式(2)の述語Predの積標準形(conjunctive normal form)が，以下の形になることがある．

$$Pred(v_1,\ldots,v_k) \equiv Pred_{n_1}(vn_1) \wedge \ldots \wedge Pred_{n_m}(vn_m) \wedge Pred'(v_1,\ldots,v_k)$$

但し，$v_i \in P(V_i)$，　vn_1は$v_1,\ldots,v_k$のいずれかとする．

この場合，各交換元ユニットの直積をとってからPredを評価することと，各々の

交換元ユニットに対して$Pred_{n_1}$を評価した後に直積をとりPred′を評価することが等価になる．後者により，直積を作成するCの処理量が軽減される．また，データ交換の中には，変換が恒等変換でFが不要な場合など，3つの要素機能を全ては必要としない場合もある．そこで，3つの要素機能は自由な順序で繋げるようにしておく必要がある．

(3)統合Cの機能について

式(2)のPredの積標準形が，関係演算の結合演算相当の部分とそれ以外との積の形になることがある．単純化のため，各v_i ($v_i \in P(V_i)$) が要素が一つの集合である場合を考える．

$$Pred(v_1, \ldots, v_k) \equiv J_1 \wedge \ldots \wedge J_m \wedge Pred''(v_1, \ldots, v_k)$$

$$J_i \equiv Ai_1(vi_1)\ \theta_i\ Ai_2(vi_2)$$

但し，$\{vi_j\}$は$v_1, \ldots, v_k$のいずれか，Ai_jはvi_jの定義域（属性）のいずれか，θ_iは比較演算子とする．

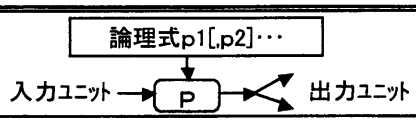
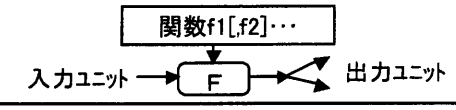
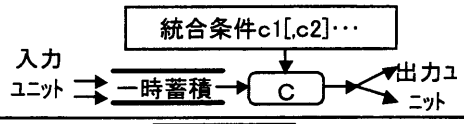
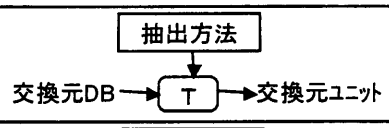
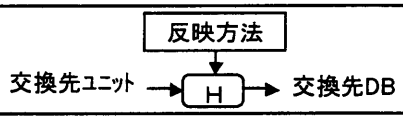
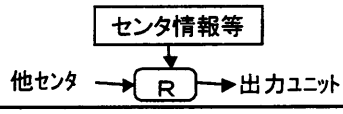
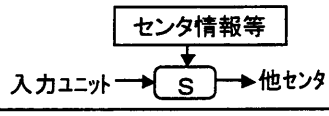
この場合，統合Cの中で結合演算相当部分（$J_1 \wedge \ldots \wedge J_m$．これ以降統合条件と呼ぶ）の評価も併せて行うことにすれば，結合演算の高速化技法（マージ結合法など[増永91]）と同様な技法を用いることにより，処理量を大幅に削減できる．統合Cでは，述語評価も併せて行えるように複合機能化しておく必要がある．また，式(2)のPredを互いに満たすような複数の交換元ユニットが同時に到着する保証はないので，入力交換ユニットの一時的な蓄積機能が必要である．

以上述べた3つの要素機能の他に，交換元データベースシステムから交換元ユニットを作成する抽出T，交換先ユニットを交換先データベースシステムに反映する反映H，および異なるセンタ間で交換ユニットの送受信を行う集信Rと配信Sの各機能が必要である．合わせて7つの要素機能をデータ交換処理モデルにおける「操作機能」と呼ぶ．操作機能の一覧を表3.1に示す．表3.1には，データフロー図にならった各操作機能の入出力関係を合わせて示す．

また，7種類の操作機能のつなぎ方の規則（操作機能のトポロジ）は以下を除いて自由とする．

- 先頭は必ず抽出，最後は必ず反映．
- 配信の直後は集信．集信の直前は配信．
- 有向サイクルになるつなぎ方は不可．

表 3.1 操作機能一覧

操作機能名	操作機能の説明	入力交換ユニットの種類数	出力交換ユニットの種類数	操作機能の繋がり規則	データフロー図
述語評価 P	入力交換ユニットから論理式を満たすものを選択する機能	1	複数	制約なし	
変換 F	入力交換ユニットに対して関数fを計算し、出力交換ユニットを作成する機能	1	複数	制約なし	
統合 C	複数の入力交換ユニットの直積から統合条件を満たすものを作成する機能	複数	複数	制約なし	
抽出 T	交換対象レコードを交換元DBから抽出し，それらのレコードをまとめ，交換ユニットを作成する機能	ー (交換元DB)	1	繋がりの先頭	
反映 H	交換ユニットを構成レコードに分解し，交換先DBへ反映する機能・	1	ー (交換先DB)	繋がりの末尾	
集信 R	交換元と交換先のセンタが異なる場合に，相手センタから交換データを受信する機能	1	1	直前は配信	
配信 S	交換元と交換先のセンタが異なる場合に，相手センタへ交換データを送信する機能	1	1	直後は集信	

3.3.4. 操作機能のセンタ配置と実行契機

3.3.4.1. 操作機能の物理的配置

既存データベースシステムの搭載センタへの交換処理の影響を減らす(3.2.2(2)(b)項）ためには，上記の各操作機能の物理的配置は，以下を除いて自由に選択できるようにしておく必要がある．

・抽出は交換元，反映は交換先のセンタに配置．

・集信と配信は，センタ間での繋がり位置に配置．

特に，交換元，交換先センタとは別の独立したセンタに操作機能を配置すれば，既存のセンタである交換元及び交換先センタの負荷を極小化できる．

3.3.4.2. 操作機能の実行契機

高負荷時間帯回避（3.2.2(2)(c)項）のためには，交換元或いは交換先センタの閑散時間帯に交換処理を開始/再開始できるようにする機能が必要である．このためには，各操作機能が，その入力データの到着時に直ちに処理を開始する内部イベント起動の他に，予め定義された契機（定時起動等）や運用者による指示等の外部イベント起動によって，各操作機能を開始/再開始する機能等が必要である．なお，統合Cでは，統合処理の開始契機の指定の他に，一時的に蓄積したデータの削除契機の指定が必要である．

3.4. データ交換システム開発・実行環境への要求条件：構築簡易化

データ交換システム開発・実行環境に要求されることは，様々な形態のデータ交換システムを簡易に構築できるようにすることである．

ここで，ソフトウェア工学において，対象領域が明確かつそれほど広くない場合の開発簡易化には，生成方式[磯田 95]，すなわち①手続き型プログラミング言語よりも高水準な言語で仕様定義を行い，②ソフトウェア部品をその中に取り込んだツールを用いて目的プログラムを自動生成する方式が有効であることが知られている．

3.3 節でのデータ交換処理モデルの整理結果より，データ交換システムの開発は，対象が明確かつそれほど広くないという生成方式が適用可能な対象領域の条件を満たすものと考える．

従って，アプローチとして生成方式を採用することによって開発簡易化を図ることとする．

3.3 節でのデータ交換処理モデルの整理結果を踏まえ，データ交換システム開発・実行環境に要求されることを整理する．

(1)高水準な仕様定義言語

(a)交換ユニットの定義

交換ユニットの定義では，3.3.1 節の式(1)に示した，複数の関係のレコードの集合を表現できること，および交換ユニットが満たすべき制約を表す述語の高水準な定義ができることが必要である．

(b)操作機能の定義

具体的な交換処理で実際に使用する操作機能と，その繋がり方，および起動契機の

指定が定義できることが必要である.各操作機能毎に必要な定義情報を表3.2に示す.特に，統合と述語評価で指定する論理式と，変換で指定する変換方法の定義とは，高水準な言語仕様で記述可能とすることが要求される.

表 3.2 定義情報の一覧

定義情報の分類＼操作機能	変換	統合	述語評価	抽出	反映	集信	配信
操作機能の繋がり方の指定	先行操作機能	先行操作機能	先行操作機能	—	先行操作機能	—	先行操作機能
入出力ユニットの指定	入力ユニット名	入力ユニット名(複数)	入力ユニット名	—	入力ユニット名	—	入力ユニット名
	出力ユニット名(複数)	出力ユニット名(複数)	出力ユニット名(複数)	出力ユニット名	—	出力ユニット名	—
起動契機の指定	開始契機	開始契機 一時蓄積削除契機	開始契機	開始契機	開始契機	開始契機	開始契機
処理の内容等	変換方法(出力ユニット対応) 入力項目と変換方法(出力項目対応)	統合条件式(出力ユニット対応)	論理式(出力ユニット対応)	DBからのデータの抽出方法	DBへの反映方法	集信元センタ情報等	配信先センタ情報等

(2)ソフトウェアの部品化と再利用

3.3.3 節で述べた各操作機能をそれぞれ，あるいは複合化した形で部品化していることが要求される.

またこれらソフトウェア部品を，指定された順序で指定された契機に実行する実行制御機能を備えていることが必要である．なお，統合における開始契機としては，

・ 入力交換ユニットが全てそろった時点，等の内部イベント起動

・ 定時起動や運用者による起動，等の外部イベント起動

などが必要であり，また，一時蓄積削除契機としては，

・ 計算可能な指定条件（例えば統合条件の成立回数など）に基づいて削除する内部イベント契機

・ 定時起動や運用者による起動，等の外部イベント起動

などが必要である.

なお，データ交換システムの開発においては，関連（同義性）を有する交換ユニッ

トの組み合わせやデータ項目の組み合わせの発見が必要となるが,本研究においては,データベース交換システムの開発者が手作業で組み合わせの発見を行うこととし，開発・実行環境において発見を支援する機能を備えることは今後の課題とした.

3.5. データ交換システム開発・実行環境

要求条件に応えるデータ交換システム開発・実行環境の一方式を提案する．説明の容易性を考慮し，先ずソフトウェアの部品化について説明し，次いで仕様定義言語について説明する.

3.5.1. メソッド

再利用可能なソフトウェア機能を部品化する.部品をメソッドと呼ぶ.メソッドは,操作機能に対応するメソッドと，変換関数として使われるデータ項目レベルの変換機能に対応するメソッドの２階層のメソッドを設ける.

(1)操作機能メソッド

(a)メソッドの種類:データ交換処理モデルの操作機能のうち,変換,統合,述語評価,集信，配信の５つの操作機能をメソッドとして設ける．抽出と反映は，交換先の既存データ入力方法の利用など交換元・交換先の多様な入出力方法に対応する必要があり汎用的なメソッドを設けるのは困難と判断し，交換元のトリガ機能を用いて半定型的な抽出を可能にするメソッド[奥村 95, 奥村 96]を設けるに止め，汎用的なメソッドは設けないこととした．なお，変換メソッド内でも述語評価を行えるように複合機能化する．これは，操作機能のつながりにおいて，変換と述語評価とが隣接する操作機能である場合に,操作機能の定義数,中間交換ユニットの定義数を削減するためである.

(b)論理式の記述能力：統合，変換，述語評価で指定する論理式の数学的な記述能力を決定するため，既存の８つのデータベースシステムの整合性チェックロジックをサンプル調査した．その結果，各システムに固有の機能に深く結びついた複雑なロジックが多く，記述能力を限定するのは困難なことが判明した．一方，比較述語を論理演算子で結んだ形の論理式の記述はほぼ共通して必要であることも判明した．そこで，比較述語を論理演算子で結んだ形の論理式を記述する機能（3.5.2 節で後述する WHERE 句及び ON 句）を設け，併せて不足機能を補うために外部プログラミングインタフェースを設ける.

(c)メソッドの起動契機：変換，統合，述語評価，集信，配信の各機能に外部イベント契機での起動機能を設ける．

統合の開始契機は，入力交換ユニットの全てがそろった時点と，指定時刻による起動の機能を，また一時蓄積削除契機は，指定された統合条件の成立回数によって削除する機能を設ける．

(2)項目レベル変換機能メソッド

変換の詳細機能（すなわちデータ項目レベルの変換機能）を部品の組み合わせで開発可能とすることを狙いに，データ異種性の分類（表 1.1）を基に，項目レベルの変換関数を導いた（表3.3）．

なお，誤データ異種性のうちの入力遅延に起因する不整合は，データ交換を行うこと自体で解消されることより，表3.3から対応する項番を省いた．

データ的異種性（値の表現方法の差異，および誤データ異種性のうちの入力誤りに起因する不整合）に関しては，原理的には対応表変換関数を設ければ十分である．しかしながら，値の変換が規則的な場合は，その規則を記述できる関数（の組）を予め設けておけばデータ交換システムの開発が一層効率化できる．開発の一層の効率化を狙いとして，予め設けておくことが望ましい関数を，既存の８つのデータベースシステムのサンプル調査結果に基づき導いた．関数例を以下に示す．

・ 電話番号の文字列操作に有用な関数（特定文字間文字列抽出関数など）．
・ NTT固有外字のコード体系（EUC，SJISなど）間の変換関数．
・ IF THEN ELSE構文，等．

なお，本研究においては，項目レベル変換機能として，本来同一実体の同一データ項目に対応するデータ項目間の変換機能のみを備えることとし，データウェアハウス構築ツールなどに有用な集約関数などを備えることは今後の課題とした．

表 3.3 項目レベル変換機能メソッド(抜粋)

項番	異種性の分類			項目レベル変換機能メソッドの分類			関数名
	大分類	中分類	小分類	大分類	中分類	小分類	
1	スキーマ的異種性	1レコード対1レコード	名称の差異	レコード名とそのデータ項目名を，シナリオの流通ユニット定義構文のユニット名とユニット内のデータ項目名に対応づけることにより解決.			−
2			構成の差異	規定値設定によるデータ項目の作成	単純代入	単純代入	move
3		多レコード対多レコード		同一ユニット内のレコードの対応の場合は項番1に同じ. 異なるユニットの場合は、"統合"メソッドにより解決.			−
4		1データ項目対1データ項目	名称の差異	シナリオの変換定義構文における<変換指定>の<項目変換指定>で、入力データ項目と出力データ項目を対応づけることにより解決			−
5			データ型の差異	データ型の変換	データ型変換	INT->CHAR	altic
						CHAR->INT	altci
						CHAR->SHORTINT	altcs
						その他30種	省略
6		複数データ項目対複数データ項目		データ項目の統合・分解	文字列操作	文字列結合	couple
						部分文字列	substr
7		レコード対データ項			算術演算	四則演算	+, -, *, /
						剰余	mod
						最大, 最小	max, min
8	データ的異種性	誤データ	誤入力データ	誤入力データの検出	対応表変換	対応表変換	altcode
9		値の表現方法の差異	異なる単位	データ型の値の体系の変換	算術演算	四則演算	+, -, *, /
10			異なる単位			剰余	mod
			異なる精度			最大, 最小	max, min
					文字列操作	文字列結合	couple
						部分文字列	substr
						左シフト/右シフト	lshift, rshift
						特定文字間文字列取出	getstr
						文字列比較	like
						その他11種(文字探索/置換等)	省略
11	異種性なし			無変換代入	単純代入	単純代入	move

注1)表1.1の項番9は除いた．また,異種性が存在しない場合を追加した.

```
<交換ユニット定義> ::=
UNIT <識別子> "{" <ユニット要素>... "}"

<ユニット要素> ::= <データ項目> | <名前付き構造体>
<データ項目> ::= <識別子> [<配列指示>] <データ型>                    ・・・(1)
<名前付き構造体> ::= <識別子> [<配列指示>] "{" <ユニット要素>... "}"  ・・・(2)
<配列指示> ::= "[" <繰り返し指定> "]"
```

図 3.6 交換ユニット定義構文（抜粋）

3.5.2. シナリオ言語

高水準仕様定義言語（シナリオ言語と呼ぶ）を設ける．

(1)交換ユニット定義

以下の構成要素を記述することにより，様々な分野のデータ交換に適用できるようにする（図 3.6）．

・データ項目とデータ項目の繰返し（図 3.6 の(1)）

・構造体と構造体の繰返し，ネスト（図 3.6 の(2)）

なお，交換ユニットの制約を表す述語は，統合，変換，述語評価の中で記述できるため，交換ユニット定義の中に述語の定義用の構文は設けなかった．

(2)操作機能の詳細定義

表 3.2 にまとめた定義情報のうち，交換処理の中核である変換，統合，述語評価の定義文を説明する．

変換については，出力ユニット対応に，WHERE 句（図 3.7 の(1)）で述語評価用の論理式を指定し，構文要素<変換指定列>（図 3.7 の(2)）で変換方法を指定する．また，構文要素<項目変換指定>（図 3.7 の(3)）で項目レベルの変換方法を指定する．<項目変換指定>の中で記述する関数は,項目レベル変換メソッドに対応する.構文要素<LOOP 指定>（図 3.7 の(4)）は，可変回構造体間の変換を可能にするためのものである．

統合については，ON 句（図 3.8 の(1)）で統合条件を指定し，WAKE 句（図 3.8 の(2)）で起動契機（定時または入力が全て揃った時点）を指定し，WAIT 句（図 3.8 の(3)）で一時蓄積削除契機（統合条件の成立回数が指定回数に達した時点）を指定する．構文要素<交換元ユニット参照列>の*記法（図 3.8 の(4)）と<統合論理式>の*記法（図 3.8 の(5)）とは，可変回繰返し構造体の作成を可能にするためのものであり，それぞれ可変個の交換元ユニットの指定とそれらのユニットに対する述語の指定とに用いる．

```
<変換定義> ::=
CONV <処理名>
PREDECESSOR <先行処理名>
FROM <交換元ユニット参照>
   {MAP <交換先ユニット参照>
     WHERE <論理式>                                          ・・・(1)
     <変換指定列>   }...                                     ・・・(2)

<変換指定列> ::= "{" <変換指定>... "}"
<変換指定> ::= <項目変換指定> | <LOOP指定>
<項目変換指定> ::= <データ項目>:=<式>;
<式>::=<定数> | <データ項目> | <関数> | (<式>)  } ・・・(3)
<関数>::=<関数記号>([<式>[,<式>]...])
<LOOP指定> ::=                                              ・・・(4)
  LOOP STRUCT_ARRAY <名前付き構造体識別子> FROM STRUCT_ARRAY <名前付き構造体識別子>;
    <項目変換指定>...
  LOOPEND;
```

図 3.7 変換定義構文（抜粋）

```
<統合定義> ::=
INTEGRATE <処理名>
PREDECESSOR <先行処理名>{, <先行処理名>}...
FROM <交換元ユニット参照列>
   {MAP <交換先ユニット参照>
   ON <統合論理式>                                          ・・・(1)
   WAKE {<起動時刻指定>| WHEN_ALL_REACH }                   ・・・(2)
   WAIT <交換元ユニット参照> UNTIL <一時蓄積削除条件>       ・・・(3)
     {, <交換元ユニット参照> UNTIL  <一時蓄積削除条件>}...}...

<交換元ユニット参照列> ::= <交換元ユニット参照>{, ITG <交換元ユニット参照>}...
   |<交換元ユニット参照> {,ITG <交換元ユニット参照>}...(ITG <交換元ユニット参照>)*
                                                            ・・・(4)
<統合論理式> ::= <論理式> | <論理式> (AND <論理式>)*       ・・・(5)
```

図 3.8 統合定義構文（抜粋）

```
<述語評価定義> ::=
DISTRIBUTE <処理名>
PREDECESSOR <先行処理名>
FROM <交換ユニット参照>
   {MAP <交換ユニット参照>
   WHERE <論理式>}...                                       ・・・(1)
```

図 3.9 述語評価定義構文（抜粋）

述語評価については，変換と同様の WHERE 句（図 3.9 の(1)）で出力交換ユニット対応に論理式を指定する．

各操作機能の繋ぎ方は，先行機能を指定することにより指定する（図3.7, 3.8, 3.9のPREDECESSOR句）．

3.6. データ交換システム開発・実行環境 DB-STREAM

提案した方式を実現するデータ交換システム開発・実行環境（DB-STREAM と呼ぶ）を試作した．

3.6.1. アーキテクチャ

DB-STREAM 及び DB-STREAM を利用したデータ交換システムのアーキテクチャを図 3.10 に示す．

DB-STREAM では，既存データベースシステムのセンタへの負荷を極小化するため，抽出・反映（及びそれらに繋がる配信・集信）以外の操作機能を全て独立センタで実行する物理的配置を採用した．また，実行制御機能は，実行時の独立機能（実行制御部）として実現した．

DB-STREAM は大きく開発環境部（図 3.10 の(a)）と実行環境部（図 3.10 の(b)）とからなる．

データ交換システム（図 3.10 の(c)）の開発時には，開発者が作成したシナリオ言語ソース（図 3.10 の(d)）を基に，目的プログラム生成部（図 3.10 の(e)）が，実行制御部（図 3.10 の(f)）で解釈実行可能な目的プログラム（シナリオオブジェクトと呼ぶ）（図 3.10 の(g)）を生成する．

データ交換の実行時には，交換元（図 3.10 の(h)）からのデータの集信を契機に，実行制御部が該当するシナリオオブジェクトを選択し，そのシナリオオブジェクトに従って必要なメソッド（図 3.10 の(i)）を順次コールすることにより，交換処理全体が実行される．

3.6.2. 機能

DB-STERAM の機能は，3.5 節で述べた機能を実現している．なお，採用したアーキテ

クチャに起因して，制約しても実用上問題無い機能については，DB-STREAM の試作稼動削減を狙いとして制約を設けた．制約を設けた機能を以下に示す．

・ メソッドの起動契機:配信と統合のみに外部イベント契機での起動機能を設けた．配信に設けた理由は，抽出・反映以外の操作機能を全て独立センタ配置としたので，既存センタの高負荷時間帯の回避のためには配信にのみ外部起動契機を設ければ十分と考えたからである．

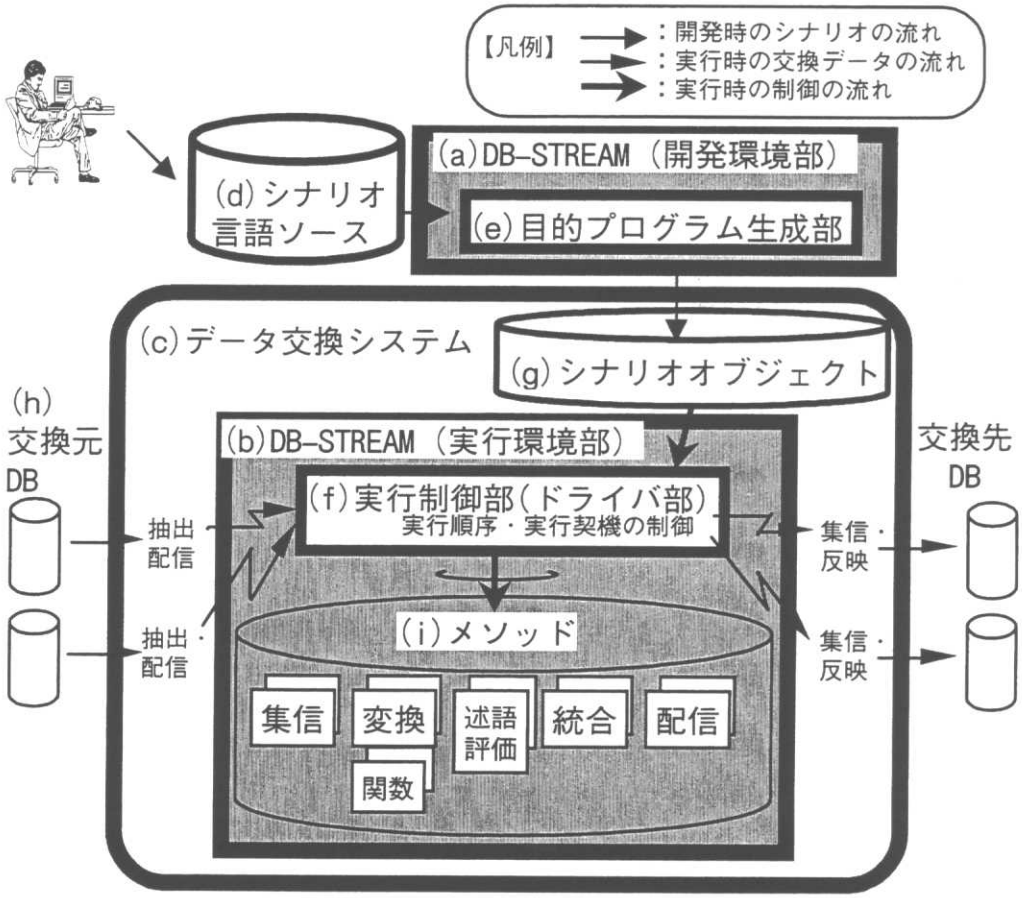


図 3.10 DB-STREAM 及びデータ交換システムのアーキテクチャ

3.7. 評価

提案するデータ交換処理モデルとデータ交換システム開発・実行環境との有効性の評価を行う.

先ず，従来研究との比較を通しての定性的評価について説明し，次いでデータ交換システムの試作を通しての評価について説明し，最後に本研究の成果の適用領域に関する考察について説明する.

3.7.1. 従来研究との比較

先ず,データ交換処理モデルの主要な特徴である,①形式的モデルに基づくデータ，機能の明確化と，②生成方式による開発簡易化とについて，従来研究と比較しての評価を行う.

次いで，開発・実行環境においては，データ交換処理モデルで明確化した機能の全てを実現してはいないことから，実現した機能の十分性を論じる.

3.7.1.1. データ交換処理モデルの評価

(1)形式的モデルに基づくデータ・機能の明確化

従来のデータ交換システムの研究において，入力データと出力データとの間の関係を形式的にモデル化した研究は見当たらない.

提案した形式的モデルは，3.3 節で論じたように，以下のモデルとして十分汎用的であると考える.

- データ異種性解消用の機能とデータ
- 複数関連データの交換可能化用の機能とデータ

汎用的な形式的モデルの考案は，①本研究においてデータ交換処理に必要な機能を導く基盤となったこと，および②今後データ交換システムの研究を進める際の基盤となりうること，の２つの観点から有用であると言えると考える．汎用的な形式的モデルの考案は，本研究の一つの意義と考える.

(2)生成方式による開発簡易化

従来のデータ交換システムの研究において，複数関連データの交換可能化用の機能

とデータを含むデータ交換システムの開発・実行環境の研究は見当たらない.

生成方式の研究は，ソフトウェア工学の分野において専ら行われている[磯田 95]が，そこでの知見として，生成方式が成立するためには，適用領域（ドメイン）が，①適度に狭い，②十分に理解されている，③技術が静的で短期間に変化しない，という良い性質を満たす必要があることが知られている.

データ交換システムという適用領域が上記の性質を満たすか否かは自明なことではなく，本研究（特に，形式的モデルに基づくデータ・機能の明確化）によって，①②の性質を満たすことが実証され，従って，生成方式の適用が可能になったと考える.

生成方式の適用が実際に可能であることを示したことは，本研究の一つの意義と考える.

なお，市販のデータベース移行システムやデータベース複製システムの中には，その構築が本研究よりも容易なもの（例：簡単なパラメタ値の指定のみで構築可能なもの）が見られるが，それらはいずれも提供する機能が本研究より劣っている（例：本研究の変換メソッド及び項目レベル変換関数に匹敵する変換機能を提供するものは見当たらない）. 即ち，それらの製品は，機能を犠牲にして簡易化を達成しているものである.

3.7.1.2. 開発・実行環境の実現機能の十分性の評価

開発・実行環境においては，データ交換処理モデルで明確化した機能の全てを実現してはいないことから，実現した機能の十分性を論じる.

データ交換処理モデルに含まれる機能で，かつ従来研究の中にその機能を実現したものが見られる機能のうち，本研究での開発・実行環境では実現対象外とした機能の一覧を先ず説明する.

(1)項目レベルの変換機能関連

以下の機能は開発・実行環境では実現対象外としている.

- 本来同一実体の同一データ項目に対応するデータ項目間の変換機能以外の機能
  例）データウェアハウス構築などに有用な集約機能
- 知識ベースなどの高度な技術を必要とする入力誤り検出機能
  例） 1章の 1.3.3 節のガン患者の症例レコードの誤り検出

(2)抽出・反映機能関連

交換元のデータベース管理システムのトリガ機能を用いて抽出メソッドを簡易に構

築する機能[奥村 95, 奥村 96]を設けるに止め，汎用的な構築機能は設けていない．異種情報源連携技術の研究（1.3.5 節）において，情報ブローカシステムにおけるラッパを簡易に構築する方法の研究（TSIMMIS[Garcia-Molina97], DISCO[Tomasic96]など）など，抽出・反映（特に抽出）機能を簡易に構築する技術の研究が見られる．

実現対象外の項目レベル変換機能は，当初想定していたデータ交換システムの範囲（次節で述べる設備データ交換システムを典型例とする定常的なデータ交換システム，あるいはシステム新規構築・更改時に必要となる初期データ構築・移行システム）では特に必要ないことから実現しなかった．

抽出・反映メソッドの汎用的な構築機能は，交換先のレガシーシステムでの既存データ入力方法の利用など交換元・交換先の多様な入出力方法に対応する必要があり，汎用的な構築機能を設けるのは困難と判断したため，実現しなかった．

以下の理由により，上記機能を実現対象外としたことが本研究の成果の適用範囲を大きく狭めることはないと考える．

- 実現対象外の変換機能は，これまでのデータ交換システム適用検討の経験を通して必要になる確率はそれほど高くないと考える．
- 抽出・反映メソッドの構築稼動は次節で見るように全稼動の高々数割である．

それぞれの実現対象外機能に関して，特記事項を述べる．

同一実体の同一データ項目に対応するデータ項目間の変換機能以外の機能((1)の第1の機能）に関しては，研究の適用範囲をデータウェアハウス構築など向けにさらに拡張するためには，あると望ましい機能であり，実現は今後の課題と考えている．

知識ベースなどの高度な技術を必要とする入力誤り検出機能（(1)の第2の機能）に関しては，既存のデータクリーニング製品を本研究の開発・実行環境（DB-STREAM）の前処理或いは後処理として適用することで十分であり，本研究の延長で新たな機能を実現することは不要と考える．

抽出・反映の汎用的な構築機能に関しては，一層のデータ交換システム構築簡易化のために，あると望ましい機能であり，その実現は今後の課題と考えている．なお，上で述べた情報ブローカシステムにおけるラッパを簡易に構築する方法の研究を元に，抽出・反映機能を簡易に構築する方法は考案可能と考えている．

3.7.2. 試作を通しての評価

3.7.2.1. DB-STREAM の適用事例

DB-STREAM を，通信ネットワークを構成する設備（回線，パス，関連する装置等）の設計業務を行う３つのデータベースシステムから，運用中の設備の監視業務を行う２つのデータベースシステムへ，データを交換させるデータ交換システム（設備データ交換システムと呼ぶ）に適用した（図 3.11）[横山 94]．以降，図 3.11 の各データベースシステムをそれぞれ DB-A～DB-E と呼ぶ．これらのデータベースシステムは，回線，パス，装置等の設備の種別に応じて段階的に個別に構築されてきた既存のシステムである．ネットワーク型データベースシステムの DB-E を除いて，他は全て関係データベースシステムである．

DB-STREAM 適用前の，設計系のデータベースシステム（DB-A，DB-B，DB-C）から監視系のデータベースシステム（DB-D，DB-E）へのデータ交換は，設計系データベースシステムの更新データを参考に，人手で監視系データベースシステムのデータ形式に変換し，入力していた．そのため，無駄な入力稼動とともに，たびたび変換誤りや入力誤りによるデータベース品質の劣化を招いていた．

設備データ交換システムのねらいは，既存データベースシステムを連携するデータ交換システムを構築することにより，データ投入稼動削減とデータ品質向上とを実現することにある．

設備データ交換システムの設計においては，レガシーシステムである DB-E への入力インタフェースとして DB-E の既存の入力インタフェースを利用することを要求された．具体的には，データ交換システムにおいて，DB-E での１つの設備情報入力トランザクションの入力レコード群に対応する交換ユニットをまとめあげてから，DB－E に入力することを要求された．

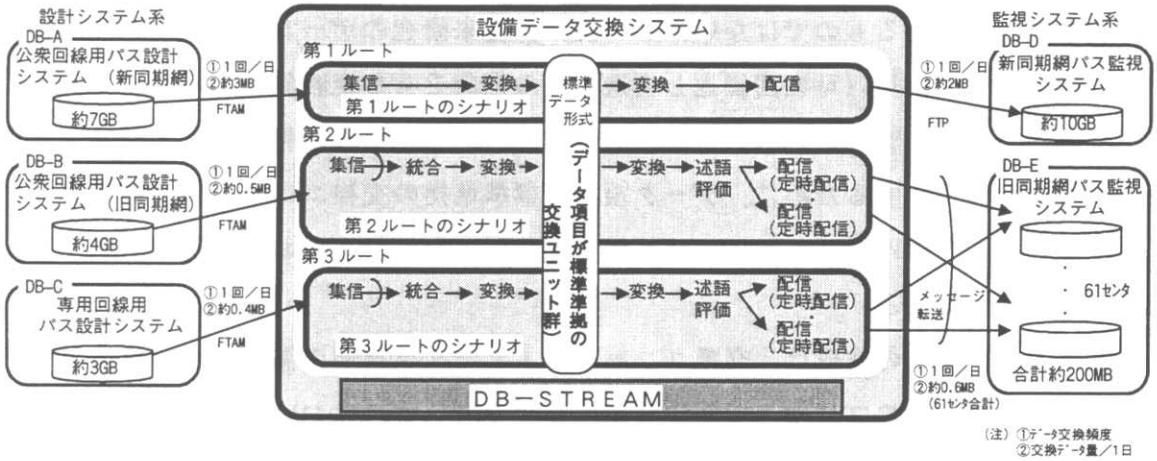


図 3.11　設備データ交換システム

設備データ交換システムは図 3.11 に示すように以下の3つのルートのデータ交換をサポートしている．

(1)第1ルート：DB-A のデータを DB-D へ交換．

(2)第2ルート：DB-B のデータを DB-E へ交換．第2ルートでは，複数種類の交換元ユニット（後述するように，それぞれが DB-B での異なる設計トランザクションの変更レコード群に対応）を基に，1つの交換先ユニット（DB-E での1つの設備情報入力トランザクションの入力レコード群に対応）を作成しなければならないケースがあり，図 3.11 に示すようにシナリオの中で統合操作機能を使用している．また，交換先の DB-E が1システム複数センタ構成をとる分散型システムであるため，シナリオ内で述語評価操作機能を用いて分解を行っている．交換は閑散時間帯である夕刻に定時配信を行う．

(3)第3ルート：DB-C のデータを DB-E へ交換．第3ルートも第2ルートと同様の理由で，図 3.11 に示すようにシナリオの中に，統合，述語評価，定時配信を使用している．

各ルートにおける交換ユニットは以下のように設計した．

各交換先データベースシステムでは，設備情報の入力トランザクション毎に，そのトランザクションに必要な入力レコード群を交換先ユニットとした．一方，各交換元

データベースシステムでは，設備の設計トランザクション毎に，そのトランザクションで変更されるレコード群を交換元ユニットとした．

交換ユニット定義の例を図3.12に示す．1章で説明したように，従来のデータ交換システムの研究におけるデータ定義機能は，関係データベースシステムのデータ定義機能を大きく超えるものではない．この例は，従来研究のデータ定義機能では，記述不可能なデータ構造（可変回繰返し構造体）の定義を含む定義例である．

交換元，交換先データベースシステムの追加，削除，更改に伴うデータ交換システムの更改を簡易化するために，データ項目が標準準拠の交換ユニット群（標準データ形式と呼ぶ）を設けた．これは，連邦データベースシステムにおける連邦スキーマに類似した概念である．

標準データ形式の作成は，交換ユニット間・データ項目間の関連特定，標準データ形式の作成の2つのプロセスで実現した．2つのプロセスについて説明する．

(1)交換ユニット間・データ項目間の関連特定

同義のデータ項目を発見するタスクと，関連（レコード集合の直積同士の等価，包含，オーバラップ）を有する交換ユニットを発見するタスクとからなる．

連邦データベースシステムのスキーマ構築における実体・データ項目の関連特定に類似したプロセスである．

データ交換システムの設計者が，交換元と交換先のスキーマ情報，設計ドキュメントを基に，手作業で同義データ項目の発見，関連を有する交換ユニットの発見を行った．スキーマ情報，設計ドキュメントだけからでは発見が困難な場合は，交換元あるいは交換先のデータベースシステムの管理者から補足情報をヒアリングすることにより，発見を行った．

(2)標準データ形式の作成

関連を有する交換ユニットを基に標準データ形式を作成するタスクと，標準データ形式と交換先ユニット，交換元ユニット（または中間ユニット）とのマッピング方法を整理するタスクと，データ項目間の変換方法を決定するタスクとからなる．

連邦データベースシステムのスキーマ構築における統合スキーマ（連邦スキーマ）作成に類似したプロセスである．

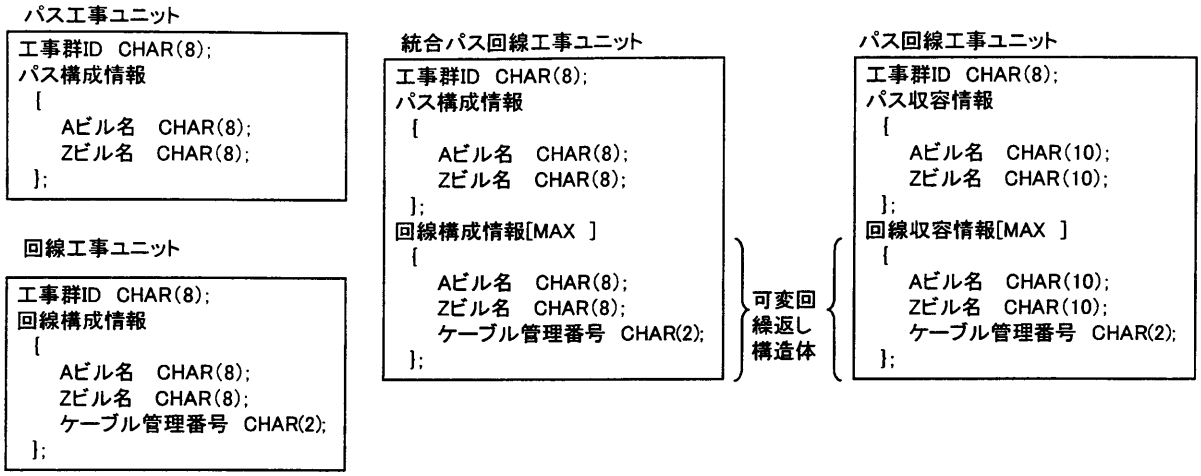


図 3.12 交換ユニット定義の例

統合定義(一部)

パス工事ユニット1個と回線工事ユニット可変個とを統合して，統合パス回線工事ユニット1個を作成.

```
FROM パス工事 ITG 回線工事 K1 (ITG 回線工事)*
{MAP 統合パス回線工事
 ON パス工事.工事群ID=K1.工事群ID
 (AND パス工事.工事群ID=回線工事[*].工事群ID
  AND if *>1 then回線工事[*].ケーブル管理番号>回線工事[*-1].ケーブル管理番号
             else回線工事[*].ケーブル管理番号>K1.ケーブル管理番号
 )*
WAKE WHEN_ALL_REACH
}
```

図 3.13 統合定義の例

変換定義(一部)

統合パス回線工事ユニットを変換して，パス回線工事ユニット1個を作成.

```
FROM 統合パス回線工事
{MAP パス回線工事
 ・・・
 パス回線工事.工事群ID := move(統合パス回線工事.工事群ID);
 ・・・
 LOOP STRUCT_ARRAY 回線収容情報 FROM STRUCT_ARRAY 回線構成情報;
  回線収容情報[*].Aビル名 := altcode (ビル名変換表,dest, source,回線構成情報[*].Aビル名);
  回線収容情報[*].Zビル名 := altcode (ビル名変換表,dest, source,回線構成情報[*].Zビル名);
  回線収容情報[*].ケーブル管理番号 := move(回線構成情報[*].ケーブル管理番号);
 LOOPEND;
}
```

対応表を用いた変換

可変回繰返し構造体間の変換

図 3.14 変換定義の例

標準データ形式は以下のように作成した.

- 構造は，対応する交換先ユニットの構造をそのまま採用した．これは，交換先ユニットの構造が標準になっていれば，交換元データベースシステムの追加・更改などのデータ交換システム更改の際に，交換先データベースシステムの自律性を維持すること（既存入力方法への影響極小化）が容易に実現できるからである．
- データ項目に関しては，社内標準規程に同義のデータ項目が存在する場合は，社内標準規程のデータ項目を標準データ項目とし，社内標準規程に同義のデータ項目が存在しない場合は，社内標準規程のデータ項目作成方針（名称は主要語・区分語を有すること[Durel187]，データ型はハード依存性の少ないものであること，など）に近いものを同義データ項目の中から選択し，標準データ項目として設定した．

標準データ形式と交換先ユニット，交換元ユニット（または中間ユニット）とのマッピング方法は，(1)で特定した交換ユニット間の関連（等価，包含，オーバラップ）を基に整理した．整理したマッピング方法は，標準形式を入力あるいは出力とする操作機能の定義文に，マッピング関係を有する＜交換元ユニット参照＞と＜交換先ユニット参照＞を記述することなどにより，シナリオに反映した．

データ項目間の変換方法は，(1)で特定した同義のデータ項目に関して，それらのデータ型や値の表現方法を基に，適切な変換関数（或いはその組み合わせ）を選択することによって決定した．決定した変換方法は，標準形式を入力あるいは出力とする変換操作の定義文の＜項目変換指定＞構文に記述することにより，シナリオに反映した．

統合定義，変換定義の具体例を図 3.13，図 3.14 に示す．図 3.12 で示した交換ユニットを用いた統合定義例および変換定義例である．１章で説明したように，従来のデータ交換システムの研究におけるデータ変換機能は，関係データベースシステムの関係演算機能と，データ型変換機能や対応表変換機能等のデータ項目レベル変換機能とを合わせた機能を大きく超えるものではない．統合定義例と変換定義例は，それぞれ，可変回繰返し構造体を含むデータ構造の作成，可変回繰返し構造体間の変換，という従来研究のデータ変換機能では記述不可能な機能の定義を含む定義例である．

3.7.2.2. データ交換処理モデルの評価

適用事例で述べたように，３つのルートのデータ交換において操作機能を組み合わ

せることにより，抽出・反映用のプログラムを例外として，プログラムを開発することなくDB-STREAMのメソッドのみで交換システムを構築でき，データ異種性の解消および交換先データベースシステムの自律性維持が実現できた.その詳細を以下に示す.

(1)データ異種性の解消

(a) 誤データ異種性解消

交換先データベースシステムに交換されるデータ量は図 3.11 に示すように一日当たり約 2.6MB である．自動データ交換の実現により，多重投入稼動を 0（従来は平均 250 人時/日）にでき，交換先データベースシステムでの投入誤り/投入漏れを 0（従来は全交換データの 5～10%）にでき，交換先データベースシステムでの投入遅延を高々1日（従来は1日から1月）にすることができた.

(b)（誤データ異種性以外の）データ異種性の解消

各データベースシステムとも個別に開発されてきた古いシステムであるためデータのデータ異種性は大きかった．3つのルートにおける，交換対象のデータ項目総数約 2100 のうちの約 2/3 が何らかの変換を必要とした．分析の結果，DB-STREAM の提供する 58 種の変換関数のうち約 70%（40 種）を使用して全ての変換を実現できた.

また，前節で説明したように標準データ形式を介する交換シナリオを採用することにより，交換元データベースシステム，交換先データベースシステムの更改，追加，削除に伴うデータ交換システムの更改を簡易なものとしている．この特徴は，大企業などで，異なる開発・更改時期のデータベースシステムを多数有し，それらのデータベースシステム間でデータ交換を実現したい場合に有効であり，現在，この標準データ形式を別の新たに追加した監視系のデータベースシステムへ交換することも行っている.

(2)交換先データベースシステムの自律性維持

(a) 交換先データベースシステムの既存機能への影響の極小化：複数関連データの交換可能化

交換先データベースシステムの一つはレガシーシステムであり，既存のデータ入力方法を用いることを要求された．該データ入力方法は，複数の互いに関連を有するデータをひとまとめにして入力する方法であった．交換ユニットおよび統合機能を用いてデータ交換を行うことにより，交換先データベースシステムの入力データ方法に必要なデータ群をひとまとめにして渡すことができた.

(b) 交換元/先データベースシステムの性能への影響の極小化

図 3.11 における第２ルート及び第３ルートの交換元/交換先センタはいずれも新たな処理を追加できる性能的な余裕が少なく，交換処理の負荷をできるだけ軽減することが望まれた．DB-STREAM が独立センタ構成であることより以下のように既存センタへの影響を最小限にすることができた．第２ルートでの処理時間は，交換ユニットオカレンス数が約 40 の時のサンプルで，交換元センタでの抽出処理時間を１とすると，交換元センタでの配信処理時間が約 0.05，データ交換システムセンタでの統合，変換，述語評価，複数センタへの配信の処理時間が約 4.5 であり，独立センタ化により，反映以外の処理を全て交換元センタで行う場合に比べ，交換元センタの負荷を約 19%（≒(1+0.05)/(1+4.5)）に軽減できた．データ交換システムの独立センタ構成を可能とし，操作機能の自由な配置を可能とする処理モデルの有効性が確認できた．

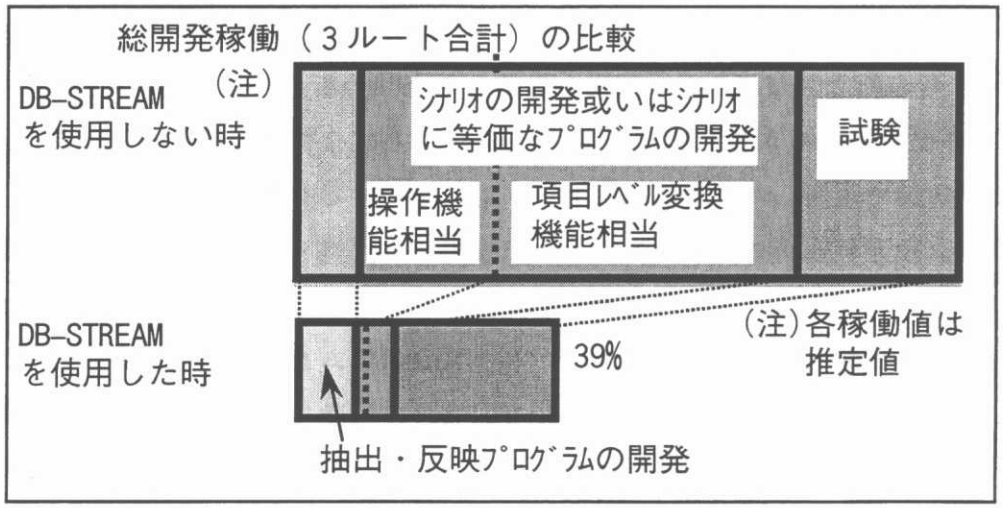


図 3.15　データ交換システム開発稼働の比較

3.7.2.3. データ交換システム開発・実行環境の評価

試験稼働も含めたデータ交換システム構築稼働は，図 3.15 に示すように全体で約 60％を削減できた．また，データ交換システム構築期間は約 1/3 に短縮できた．このうちシナリオ開発稼働については約 10 倍の生産性向上を達成し，大きな効果があった．

分計はしていないが，データ交換システム構築稼働の中には，交換ユニットの構成データ項目の追加/変更や，変換方法の修正等の稼働も含まれており，これらの交換定義仕様の修正が，シナリオの部分修正のみで可能だったことの効果が大きい．データ

交換システム開発・実行環境として高水準な仕様記述言語を提供し，シナリオ定義のみで構築可能とすることの重要性が確認できた．

3.7.3. 適用領域に関する考察

前節までの評価結果より，本研究の成果の適用領域に関して整理する．
本研究が有効となる明確なケースは，以下である．

① 交換先の既存データ入力方法への影響極小化のために交換ユニット（複数の関連するレコードの集まり）が必要になるデータ交換システムの構築．

ところで，一般に，歴史のある中・大規模の会社におけるデータベースシステムの構築は以下の特性を持つと言われている[Brodie93]．

・ 多数のデータベースシステムを有する．
・ 各業務分野毎に独自の設計指針に基づいて設計されたデータベースシステムの割合が多い．全社共通的な指針に基づいて設計されたものはさほど多くない．

上記の特性を持つということは，多くのデータベースシステム間に多様なデータ異種性が発生している可能性が高いことを示し，また，既存データベースシステムを含む情報システム群の段階的更改などにおいて，既存入力方法への影響極小化などの自律性維持要求が発生する場合が少なからずあることを示すものと考える．

従って，一般に歴史のある中・大規模の会社において，①のケースはしばしば生じるものと考える．

実際の社内適用事例においても，前節で述べた設備データ交換システム以外にも，上記①に分類される事例が数例ある．これらの適用事例においては，データ交換システムの構築が簡易化されたことと，3.7.1 節で今後の課題として述べた機能を除き新たな機能要求はなかったことが確認されている．これは，本研究の成果の有効性を裏付けるものと考える．

また，適用対象のデータ交換システムで必要となる機能の多寡に依存して，競合製品・研究と比べて，本研究の成果が最適となりうる（稼動最小で構築できる）ケースが存在する．以下に示す．

② 業務として定常的にデータ交換を行うシステムの構築．
③ 情報システムの更改において，初期データの構築および移行を行うシステムの構築

④ データウェアハウス構築システムの構築.

・データローダ

・データクリーニング（データクレンジング）システム

具体的社内事例において，上記③，④に分類される事例が数例ずつある．これらの社内事例においては，開発組織が機能と構築容易性とを総合的に判断して本研究の成果（DB-STREAM）の適用を決めた．また，適用の結果，データ交換システムの構築が簡易化されたことと，3.7.1 節で今後の課題として述べた機能を除き新たな機能要求はなかったこととが確認されている．このことは，本研究の成果の有効性を裏付けるものであり，本研究の成果が②～④のケースで最適になる場合が少なからぬ割合であることを示すものと考える．

以上をまとめると,本研究の成果は一定の適用領域を有すると言えるものと考える.

3.8. 3章のまとめ

交換元・交換先データベースシステム間のデータ異種性を解消することと，交換先データベースシステムの自律性を維持すること，特に交換先のデータベースシステムの既存入力方法への影響を極小化しつつデータ交換を行うことを可能にするデータ交換システムの簡易な構築方式が求められる．

本章では，上記の要求条件を満たすデータ交換システムの構築方式を確立することを目的に，①データ交換処理モデル，ならびに②これに沿ったデータ交換システム開発・実行環境を提案した．

データ交換処理モデルは，形式的なデータと機能のモデルの考案と，それに基づく操作機能の導出とを特徴とする．

データ交換システム開発・実行環境は，生成方式を適用すること，即ち，データ交換の仕様を簡易に記述できる高水準仕様記述言語を備えることと，基本的な操作機能およびデータ項目値変換関数をソフトウェア部品として備えることとを特徴とする．

形式的なデータと機能のモデルの考案と，生成方式が実際に適用可能なことを示したことは，従来の類似研究・製品には見られない有意義な特徴である．

データ交換処理モデルの機能のうち，①同義のデータ項目間の変換以外の項目レベル変換機能（集約機能など），②知識ベース技術などを必要とする高度な入力データ誤り検出機能，③汎用的な抽出・反映メソッド，は開発・実行環境で未実現であるが，

未実現であることが研究の有効性を少なからず損なうものでないと考え，それらの機能の実現は今後の課題とした．

データ交換システム開発・実行環境の実装（DB-STREAM と呼ぶ）を行い，さらに DB-STREAM を用いて複数データベースシステム間のデータ交換システムを試作し，データ交換処理モデルおよびデータ交換システム開発・実行環境の有効性を評価した．

交換元・交換先データベースシステム間のデータ異種性を解消し，交換先データベースシステムの自律性を維持するデータ交換システムが構築できた．データ交換処理モデルはデータ異種性解消，自律性維持に有効である．

データ交換システム構築稼動に関しては，個別プログラミングによって構築した場合に比べて構築稼動が約 4 割になった．データ交換システム開発・実行環境は構築簡易化に有効である．

提案した開発・実行環境の適用領域に関しては，以下が明確な適用領域である．歴史のある中・大規模の会社においては，以下のケースがしばしば生じうると考える．

① 交換先の既存データ入力方法への影響極小化のために交換ユニット（複数の関連するレコードの集まり）が必要になるデータ交換システムの構築．

また，以下も適用領域になる．適用対象のデータ交換システムで必要となる機能の多寡に依存して，本研究の成果が最適となりうる場合が少なからぬ割合で存在すると考える．

② 業務として定常的にデータ交換を行うシステムの構築．

③ 情報システムの更改において，初期データの構築および移行を行うシステムの構築

④ データウェアハウス構築システムの構築．

・データローダ

・データクリーニング（データクレンジング）システム

4. データウェアハウスにおけるオンラインデータローダ高速化方式

4.1. まえがき

本章では，データウェアハウス(DWH:Data Warehouse)におけるオンラインデータローダ高速化方式について論じる[池田 88].

データウェアハウスの一つの形態にオペレーショナルデータストア（ODS：Operational Data Store）がある．ODS は，遅延の少ない意志決定支援を可能にすることなどを目的に，基幹系データベースシステムと同じ環境に構築されるデータウェアハウスである（1.3.4 節参照).

ODSへのデータローダは，高性能な（遅延の少ない）ロードを可能にするため，しばしばトランザクション技術を用いて構築される．トランザクション技術を用いて構築されるデータローダをオンラインデータローダと呼ぶ.

ODSのオンラインデータローダのアクセス特性は,多量のデータを基幹系データベースシステムから検索し，少量の更新をODSに行うことである．これ以降，ODSのデータローダを始めとする,多量の検索と少量の更新を行うグローバルトランザクション(複数サイトにまたがるトランザクションをグローバルトランザクションと呼ぶ）を大量検索型グローバルトランザクションと呼ぶこととする.

ここで，少量のデータの検索・更新を行うトランザクション（グローバルとローカルの両方を含む）と，大量検索型グローバルトランザクションとが共存する環境を考える．前者のトランザクションを「通常のトランザクション」と呼ぶこととする．両者が共存する環境において，両者を高性能に実行させることは簡単ではないことが知られている[Pirahesh 90]．例えば，単純に２相ロック方式を適用すると，大量検索型グローバルトランザクションの大量READが，通常のトランザクションを多数かつ長時間ウェイトさせるという問題を生じる.

１章で述べたように，上記環境向けのトランザクション同期制御方式は確立していない．従来提案されている方式には，正当性（直列可能性）が保証されない，古いデータを読むことになる，等の問題がある.

本章では，上記環境向けの高性能なトランザクション同期制御方式を論じる．特に，通常のトランザクションにはミッションクリティカルなものが一般に多数存在することを考慮して，通常のトランザクションの高性能な実行を可能にするトランザクション同期制御方式を論じる.

検討においては，分散データベースシステムのトランザクション同期制御方式の主流である２相ロック方式と，グローバルトランザクションが検索専用の場合は，同期制御に起因する副作用が小さい（アボートは一切発生せず，他のトランザクションとの間で待ちが発生するのはREADの実行時に,未完了のトランザクションによって作成された版を選択しそのトランザクションの完了を待つ場合のみ）という特長を持つ方式である多版時刻印方式とに着目して，２相ロック方式と多版時刻印方式とを統合した新たなトランザクション同期制御方式を提案する．

以下，4.2 節では，トランザクション同期制御方式への要求条件を説明する．次いで，4.3節では，新たな方式の提案を行う．4.4節では，提案方式の有効性の評価を行う．

4.2. トランザクション同期制御方式への要求条件

先ず，トランザクション同期制御方式の従来の研究状況の説明を行い，次いで，本研究における要求条件について説明する．

従来の研究状況の説明は，本研究でベースとする方式の説明と，ODS 向けトランザクション同期制御方式の研究状況の説明とからなる．

要求条件は，高性能化に関わる条件と，異種性解消に関わる条件との２つがある．

4.2.1. 従来の研究

(1) ２相ロック方式と時刻印方式

本研究において提案方式のベース方式として採用した，２相ロック方式(two phase locking method)と時刻印方式(timestamp ordering method)とについて説明する．

２相ロック方式は,検索／更新(READ／WRITE)要求毎に対象データにロックをかけ，コミットあるいはアボート時にロックを解放するという２相ロックの原理[Eswaren76]でトランザクションの正当性基準（直列可能性）を保証する方式である．２相ロック方式では，デッドロックが生じうる．デッドロックへの対処法としては，トランザクション待ちグラフの中の有向サイクルの有無を判定することによってデッドロックを検出し，サイクル中のトランザクションをアボートすることによってデッドロックを解消する方式が一般的である[Obermarck82] [Stonebraker77]．この方式には，アボートの発生頻度が少ないという大きな長所がある．そのため，集中形データ

ベースシステムおよび分散データベースシステムのトランザクション同期制御方式の主流となっている．しかしながら，通常のトランザクションと大量検索型グローバルトランザクションが共存する環境では，双方を高性能に実行させるのは困難であるという問題点がある．

時刻印を用いるトランザクション同期制御方式は，競合するトランザクションを時刻印順に実行させることによって,直列可能性を保証する方式である.時刻印方式は，基本時刻印方式(basic timestamp ordering method)と，多版時刻印方式(multiversion timestamp ordering method)とに大別される．

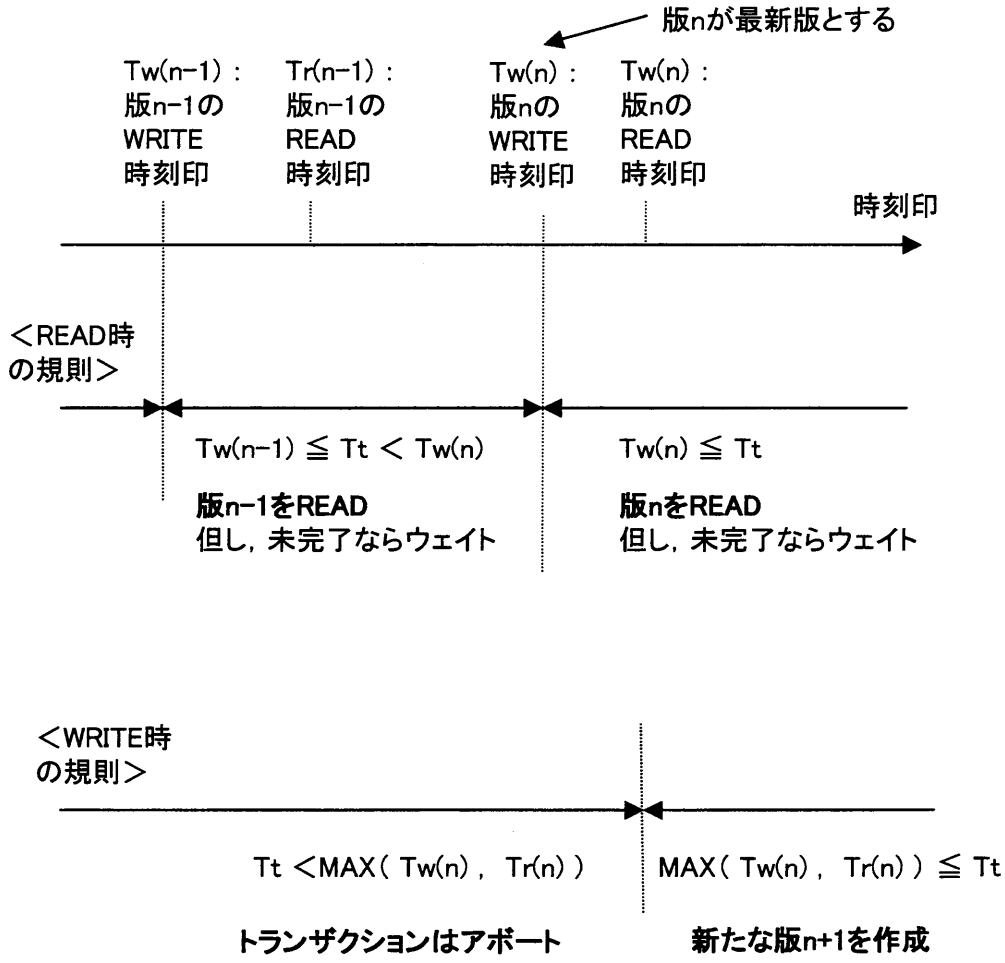


図 4.1 多版時刻印方式

基本時刻印方式[Bernstein80]は，トランザクションの時刻印 Tt と，データの READ 時刻印 Tr と，WRITE 時刻印 Tw とを用いてトランザクションの実行を制御する．READ 時は，Tt>Tw の時は読み込み可，それ以外は，トランザクションのアボートになる．

WRITE 時は，Tt>MAX (Tw, Tr) の時は書き込み可，それ以外は，トランザクションのアボートになる．この方式は，トランザクションの並列走行性が高いという長所を持つが，アボートが多発するという問題点を持つ．

多版時刻印方式[Reed83] [Chan83]は，データの WRITE 毎に新たな版を作成する点が異なる．各版毎に WRITE 時刻印 Tw(n) (但し，n は版を表す) と，READ 時刻印 Tr(n) とが付随する．WRITE 時刻印は，その版を生成したトランザクションの時刻印と等しく，READ 時刻印は，それを読んだトランザクションの内，最新のトランザクションの時刻印と等しい．

READ 時および WRITE 時の規則を図を用いて説明する (図 4.1)．トランザクションがデータを READ する時は，WRITE 時刻印が Tt より小さい版のうち最大の時刻印を有する版を選択する．例えば，版 n が最新版とすると，Tw(n)≦Tt ならば版 n が選択され，Tw(n-1)≦Tt＜Tw(n)ならば版 n-1 が選択される．該版がコミット済みの場合は，直ちに読み込みが行われ，未コミットの場合は，ウェイトする．WRITE 時は，最新版 n の時刻印と比較して，MAX (Tw(n) , Tr(n)) ≦Tt の時は，書き込みと新たな版 n+1 の作成が可能となり，それ以外は，トランザクションのアボートとなる．性能特性は，基本時刻印方式とほぼ同様であるが，READ 時にアボートが発生しない分、アボート発生頻度が小さくなり、ウェイトが存在する分、並列走行性が低くなる．この方式の特徴は，READ 時に同期制御に起因する副作用が小さい（アボートは一切発生せず，他のトランザクションとの間で待ちが発生するのは READ の実行時に，未完了のトランザクションによって作成された版を選択しそのトランザクションの完了を待つ場合のみ）という点である．

(2)ODS 向けのトランザクション同期制御方式

少量のデータの検索・更新を行う通常のトランザクションと大量検索型グローバルトランザクションとが共存する環境向けに従来提案されているトランザクション同期制御方式としては，２相ロック方式を基本にしつつ，大量検索型グローバルトランザクションのREADにのみ特別な規則を適用する方式が知られている．代表的な２つの方式を説明する．

(a)ダーティリード

大量検索型グローバルトランザクションのREADにおいて，READ対象ページを他トランザクションが既に排他ロック（WRITEロック）していても，その排他ロックを無視してREADする．ACID特性のうち，分離性を犠牲にして，性能向上を図るアプローチである．

(b)予備系のREAD

耐障害性の向上を狙いとして，現用のデータベースとは別に予備系のデータベース（即ち，現用データベースのバックアップ）を設けている場合がある．この時，大量検索型グローバルトランザクションのREADにおいて，現用データベースの代わりに，予備系のデータベースをREADする方式が知られている．

上記２つの方式は両方とも問題がある．(a)には，未コミットの更新途中のページをREADすることから，正当性（直列可能性）が保証されないという問題がある．(b)には，現用データベースに比べると，古いデータ（例えば１日前のデータ）を読むことになるという問題がある．

4.2.2. トランザクション同期制御の高性能化

通常のトランザクションと，大量検索型グローバルトランザクションとが共存する環境での，高性能なトランザクション同期制御方式を導くこと．特に，ミッションクリティカルなものが多数存在する通常のトランザクションの高性能な実行を可能とする方式を導くことが要求される．

4.2.3. 異種性解消

ローカルなデータベース管理システム間の異種性を簡易に解消して，提案するトランザクション同期制御方式が実装できること．すなわち，ローカルなデータベース管理システムの機能への影響（改造）を少なくしつつ，提案するトランザクション同期制御方式が実装できることが要求される．

4.3. 提案方式

4.3.1. トランザクション同期制御方式

4.3.1.1. 検討の前提条件

同期制御方式の正当性の基準としては，通常の直列可能性を用いる．これは，提案方式の適用範囲をできるだけ広いものとするためには，直列可能性を緩和することなく採用するのが望ましいと判断したからである．

ローカルデータベースシステムとしては，現状の集中形データベースシステムおよび分散データベースシステムの技術動向を考慮して，以下の条件をみたすものを前提とする．

- トランザクションの原子性，耐障害性保証には，2相コミットメント・プロトコル(two phase commitment protocol)[Gray79]を使用
- トランザクションの同期制御には，2相ロック方式を使用
- デッドロックは，トランザクション待ちグラフの解析により検出

4.3.1.2. トランザクション同期制御方式の提案

READ 時に同期制御に起因する副作用が小さいという特長を持つ方式である多版時刻印方式と，分散データベースシステムの同期制御方式の主要な方式である2相ロック方式とのそれぞれの長所を組み合わせることにより，高性能な同期制御方式を導くこととする．

大量検索型グローバルトランザクションについては多版時刻印方式を修正適用し，ローカルトランザクションと通常のグローバルトランザクションについては2相ロック方式を修正適用することによって，新たな方式を作成，提案することを検討する．

以下，検討の内容を，以下の順に述べる．

- 時刻印の管理方法
- 通常のトランザクション（ローカルとグローバルの両方を含む）に関する規則
- 大量検索型グローバルトランザクションの READ に関する規則
- 大量検索型グローバルトランザクションの WRITE に関する規則
- デッドロックの解消規則

なお，大量検索型グローバルトランザクション全体が検索操作のみの場合は，検索対象のテーブルのうちの一つを仮に更新対象とみなして規則を適用することとする．

(1) 時刻印の管理方法

トランザクションの直列順序を管理する手段として，時刻印を用いる．本研究のこれ以降の検討内容と整合させるため,時刻印の管理は以下の条件を満たすものとする．

・ 時刻印の付与用の時計がグローバルに管理されており，その時計を用いて，時刻印払い出し要求順に合わせて昇順に時刻印が払い出される．

(2)通常のトランザクションに関する規則

通常のローカルトランザクションの READ・WRITE，通常のグローバルトランザクションの READ・WRITE は２相ロック方式によって制御する．但し，WRITE 時に WRITE 対象データの新たな版を作成する点が単純な２相ロック方式と異なる．

READ 時は，READ 対象データの最新版に READ ロックをかけて，READ を行う．WRITE ロックがかかっている場合は，ウェイトする．READ ロックの解放は，トランザクションのコミットあるいはアボート時に行われる．WRITE 時は，WRITE 対象データの新たな版を作成し，WRITE ロックをかける．対象データに既に READ ロックあるいは WRITE ロックがかかっている場合はウェイトする．WRITE ロックの解放は，トランザクションのコミットあるいはアボート時に行われる．

作成した版には時刻印を付与する．時刻印の付与方法は(3)で説明する．

(3)大量検索型グローバルトランザクションの READ に関する規則

多版時刻印方式に従う．

版の時刻印付与方法と選択方法とが，オリジナルな多時刻印方式と異なる．

版の時刻印の付与方法を先ず説明し，ついで版の選択方法を説明する．

(a)版の時刻印の付与方法

ローカルトランザクションによって作成された版と通常のグローバルトランザクションによって作成された版は，そのトランザクションの完了時点のグローバルな時刻を時刻印として付与する．従ってそれらのトランザクションによって作成された版の時刻印は，トランザクションの完了時点までは未確定である．なお，版の時刻印の値の比較が必要な場合は，未確定の時刻印の値は無限大であるとみなす．大量検索型グローバルトランザクションによって作成された版は，そのトランザクションの開始時刻印を時刻印として付与する．開始時刻印の厳密な定義は，(4)で述べる．

(b)版の選択方法

READ 対象レコードの含まれる複数の版のうち，大量検索型グローバルトランザクションの開始時刻印よりも小さい時刻印を持つ版の中で最大の時刻印を持つ版を READ 対象の版として選択する．READ 時の規則を図を用いて説明する（図 4.2）．トランザクションの開始時刻印を Tt とする．例えば，版 n が最新版とすると，$Tw(n) \leqq Tt$ ならば

版 n が選択され，Tw(n-1)≦Tt＜Tw(n)ならば版 n-1 が選択される．以下に示す理由により，オリジナルな方式と異なり，選択された版は必ず READ 可能である．

- 通常のトランザクションによって作成された版が選択された場合は，その版の時刻印が既に確定していることを意味する．このことは，版の作成方法で述べたようにそのトランザクションがコミット済みであることを意味する．
- 大量検索型グローバルトランザクションによって作成された版が選択された場合は，(4)の規則で述べるように，版を作成したトランザクションがもし未完了ならば，その WRITESET（(4)参照）を READ しようとすること自体がウェイトさせられるはずなので，版を作成したトランザクションは必ずコミット済みである．

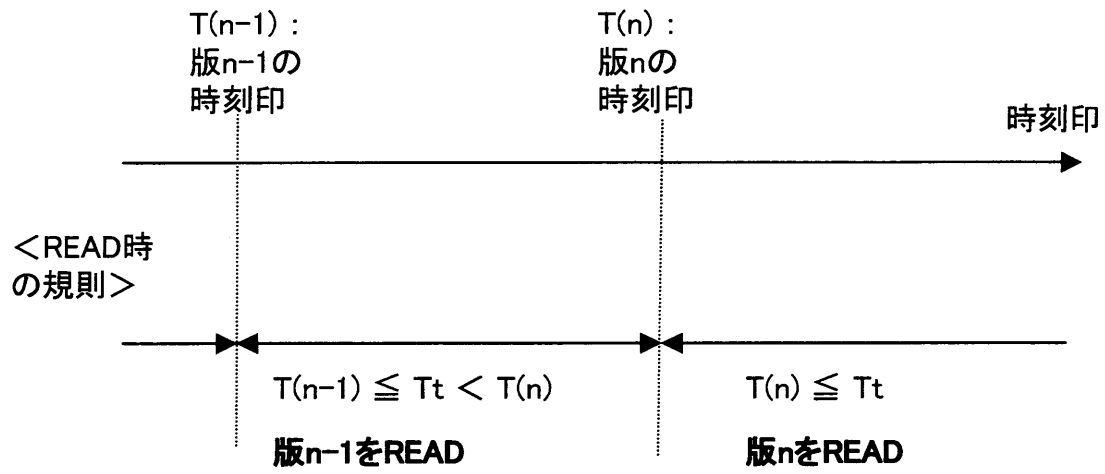


図 4.2 READ 時の版の選択方法

上記の時刻印付与方法と版選択方法との狙いは，トランザクションの実行履歴において，大量検索型グローバルトランザクションの全操作を大量検索型グローバルトランザクションの開始時刻印以前に完了済みの全トランザクションの全操作より後に位置付け可能とし，かつ大量検索型グローバルトランザクションの開始時刻印以降に完了する全トランザクションの全操作より前に位置付け可能とすることにある．

(4)大量検索型グローバルトランザクションの WRITE に関する規則

上に挙げた規則だけでは，直列可能性を保証することはできない．

矛盾が生じる例を示す．

［例］

以下の前提の下での，データベースシステム A,B における実行は矛盾を生じさせる．

T1：更新対象がデータベースシステム A の大量検索型グローバルトランザクション

T2：更新対象がデータベースシステム B の大量検索型グローバルトランザクション

T1A：データベースシステム A での T1 の更新操作

T2A：データベースシステム A での T2 の検索操作

T1B：データベースシステム B での T1 の検索操作

T2B：データベースシステム B での T2 の更新操作

T1 の時刻印< T2 の時刻印

＜データベースシステム A＞	＜データベースシステム B＞
T2A：READ　X	T1B：READ　Z
（T1 作成版より古い版を READ）	（T2 作成版より古い版を READ）
T1A：WRITE　X	T2B：WRITE　Z
T1A：COMMIT	T2B：COMMMIT
データベースシステム A での直列順序	データベースシステム B での直列順序
T2＜T1	T1＜T2

この問題に対処するため，大量検索型グローバルトランザクションをその WRITESET に関して，他のトランザクションと直列に実行させる規則を追加することとする．ここで，WRITESET とは，そのトランザクションでの更新（追加・変更・削除）操作の対象となる全てのテーブルの和集合とする．

規則の詳細を説明する．

(a)大量検索型グローバルトランザクションの WRITE と，大量検索型グローバルトランザクションの READ 以外の操作との規則

大量検索型グローバルトランザクションの開始から完了までの間，WRITESET 全体の排他ロック（WRITE ロック）を保持する．即ち，先ず，大量検索型グローバルトランザクションの開始時は，WRITESET にアクセス中の全ての他トランザクションの完了を待ち，また WRITESET に新たにアクセスしようとする他トランザクションの実行を待たせる．

さらに，大量検索型グローバルトランザクションの実行開始後は，その WRITESET にアクセスしようとする他トランザクションの実行をウェイトさせる．

ここで，WRITESET が互いにオーバラップする大量検索型グローバルトランザクションの排他ロック取得に関しては，トランザクションの開始時刻印順に排他ロックが取

得されるように，排他ロックの取得順序を制御することとする．これは，排他ロックの取得順序がテーブル間で逆転することに起因して大量検索型グローバルトランザクション間でデッドロックが発生するのを，予め回避するためである．

(b)大量検索型グローバルトランザクションのWRITEと,他の大量検索型グローバルトランザクションのREADとの規則

大量検索型グローバルトランザクション（仮にAと呼ぶ）のWRITESETと他の大量検索型グローバルトランザクション（仮にBと呼ぶ）の検索対象がオーバラップする場合の制御は以下のように行う．

Aの開始時刻印＞Bの開始時刻印の場合は，Bによる検索を許す．BはA以前のトランザクションで作成された版を読むことになる．

Aの開始時刻印＜Bの開始時刻印の場合は，Aの開始時点にBが既にAのWRITESETをアクセス済みならば，AはBの完了をウェイトする．(c)で述べるように，Aはウェイト完了後に,Bよりも大きい開始時刻印を再度払い出してもらい,実行を開始する．Aの開始時点にBがAのWRITESETに未アクセスならば，Aの完了までBの実行をウェイトさせる．

(c)開始時刻印の決定規則

大量検索型グローバルトランザクションの開始時刻印は，トランザクション開始時のグローバルな時刻とする．但し，WRITESETでの排他ロック取得のために先行トランザクションの完了待ちが発生した場合は，待ちが完了し実行が開始する時点のグローバルな時刻とする．

(5)デッドロックの解消規則

大量検索型グローバルトランザクションはその開始時に,WRITESETにアクセス中の他のトランザクションを待ち，さらに開始から完了までの間，WRITESETに新たにアクセスしようとする他のトランザクションを待たせる．従って，大量検索型グローバルトランザクションを有向サイクルの一部に含むような，トランザクション待ちグラフの有向サイクルが発生しうる．即ち，大量検索型グローバルトランザクションを含むデッドロックが生じうる．

ところで，大量検索型グローバルトランザクション同士で待ちが発生するのは，双方のWRITESETがオーバラップする場合あるいは片方のWRITESETと他方の検索対象がオーバラップする場合に，先に開始したトランザクションを後続のトランザクションが待つ場合だけである．よって，大量検索型グローバルトランザクションだけで，有

向サイクルを構成することはありえない．従って，大量検索型グローバルトランザクションを含むデッドロックが発生した場合は，大量検索型グローバルトランザクション以外のトランザクションをアボートすることにより，デッドロックを解消することが常に可能である．

以上より，大量検索型グローバルトランザクションを含むデッドロックが発生した場合は，大量検索型グローバルトランザクション以外のトランザクションをアボートすることにより，デッドロックを解消することとする．

(6) トランザクション同期制御方式のまとめ

- ローカルトランザクションと通常のグローバルトランザクションとのトランザクション同期制御方式は2相ロック方式とする．但し，WRITE 時は新しい版を作成する．
- 大量検索型グローバルトランザクションの READ は多版時刻印方式に従う．但し，版の時刻印付与方法と選択方法とがオリジナルな方法と異なる．ローカルトランザクションと通常のグローバルトランザクションによって作成された版の時刻印はトランザクションの完了(COMMIT)時点のグローバルな時刻を付与する．大量検索型グローバルトランザクションによって作成された版は，そのトランザクションの開始時刻印を版の時刻印として付与する．版の選択方法は，READ 対象レコードの含まれる複数の版のうち，大量検索型グローバルトランザクションの開始時刻印よりも小さい時刻印を持つ版の中で最大の時刻印を持つ版を READ 対象の版として選択する．
- 大量検索型グローバルトランザクションはその WRITESET（トランザクションでの更新対象テーブルの和）に関して，他のトランザクションと直列に実行させる.即ち,WRITESET に既にアクセスしている他トランザクションの完了を待ち，また WRITESET に新たにアクセスしようとする他トランザクションの実行を待たせる．大量検索型グローバルトランザクションの実行中は，その WRITESET にアクセスしようとする他トランザクションの実行をウェイトさせる．但し，他の大量検索型グローバルトランザクションの検索操作が WRITESET 内のレコードを検索しようとする場合は，他の大量検索型グローバルトランザクションの開始時刻印が大量検索型グローバルトランザクションの開始時刻印よりも小さい場合に限り，待たせずに検索を可能とする．
- ロックの両立性規則を表 4.1 にまとめる．

・ デッドロックの発生時は，通常のトランザクションをアボートすることによってデッドロックを解消する．デッドロック解決時に大量検索型グローバルトランザクションがアボートされることはない．

表 4.1 ロック両立性規則

後行操作 ＼ 先行ロック	通常READ	WRITE	大量検索型TR内READ
READ	可 ロック	不可 ウェイト	可 ロックはかけない
WRITE	不可 ウェイト	不可 ウェイト	可 ロックはかけない 注）

注）古い開始時刻印を持つ大量検索型グローバルトランザクションの完了（コミットあるいはアボート）をウェイトする場合がある．

(7)提案方式の正当性

全てのトランザクションは直列化可能である，即ち，本研究で提案したトランザクション同期制御方式は正しい同期制御方式であることを示す．

先ず，通常のトランザクションに関しては，２相ロック方式を用いることにより直列化可能である．

また，大量検索型グローバルトランザクションの READ・WRITE に関しては以下が成立する．

(a)READ では，大量検索型グローバルトランザクションの開始時刻印よりも小さい時刻印を持つ版のうち最大の時刻印を持つ版を読む．

また，以下に述べる２つのことより，大量検索型グローバルトランザクションが READ した版よりも後に作成される版の時刻印は，必ず大量検索型グローバルトランザクションの開始時刻印よりも大きくなる．

・ローカルトランザクション及び通常のグローバルトランザクションによって作成

される版の時刻印はそのトランザクションの完了時点の時刻である．

・他の大量検索型グローバルトランザクションの WRITESET を READ する際は他の大量検索型グローバルトランザクションと開始時刻印の大小順に直列実行させられる．

(b)WRITE で作成される版に関しては，WRITESET へアクセスする他トランザクションの完了を待って大量検索型グローバルトランザクションを開始することから，大量検索型グローバルトランザクションの開始時刻印はそれ以前に作られた版の時刻印よりも大きい．

また，大量検索型グローバルトランザクション実行中は（開始時刻印の小さい他の大量検索型グローバルトランザクションの多版 READ を除き）他トランザクションの実行を待たせることから，大量検索型グローバルトランザクションの完了後に，WRITESET 内に新たに作成される版の時刻印は大量検索型グローバルトランザクションの開始時刻印よりも必ず大きい．

以上より，個々の大量検索型グローバルトランザクションの全操作は，トランザクション実行履歴において，その開始時刻印よりも小さい完了時刻印を持つ全ての通常のトランザクションの全ての操作よりも後ろに位置付け可能であり，その開始時刻印よりも大きい完了時刻印を持つ全ての通常のトランザクションの全ての操作よりも前に位置付け可能である．また，大量検索型グローバルトランザクションの操作同士は開始時刻印順に直列化される．

これらのことと，通常のトランザクションの実行は2相ロック方式により直列化可能となっていることより，全トランザクションの実行は直列化可能である．

これは，提案する方式が正しい方式であることを示す．

4.3.2. 異種性解消方式

次に，異種性の解消方式について説明する．

提案するトランザクション同期制御方式を，ローカルデータベース管理システム上に簡易に（即ち少ない改造で），実現する方式を検討した．

以下の方針で検討した．

・　外付けの機能（グローバルな質問処理機能およびトランザクション同期制御機能）と，ローカルデータベース管理システムのオリジナルな機能（質問処理機能およびトランザクション同期制御機能）を改造した機能とが協調動作して，提

案方式を実現することを狙う．

・　ローカルデータベース管理システムのオリジナルな機能（質問処理機能およびトランザクション同期制御機能）への改造は極小化する．

上記方針に従い考案した異種性解消方式を説明する．

検討にあたり，以下を前提条件に追加する．

・　ローカルなデータベース管理システムにおいて，テーブルレベルの排他ロック（WRITE ロック）命令を発行可能なこと．

現在，業務用に使用されている主要な関係データベースシステムにおいては，必ず備わっている機能であり，妥当な仮定である．

また，煩雑になるのを避けるため，グローバルな質問処理機能とグローバルな同期制御機能との間のインタフェース機能の説明は省いている．質問処理機能で利用可能な情報は，同期制御機能でも利用可能とする．

(1)グローバルな質問処理機能

主な機能は，コンパイル時にグローバルトランザクション（通常のグローバルトランザクションと大量検索型グローバルトランザクション）の各操作を個別のデータベースシステムが実行可能な形式に分解・変換することと，実行時にグローバルトランザクションの各操作を適切な個別のデータベースシステムに処理を依頼しつつ実行することである．

提案方式の実現に必要となる機能を説明する．

(a)大量検索型グローバルトランザクションの識別機能

トランザクションが大量検索型グローバルトランザクションであることを識別できることが必要である．具体的な実現方法としては，大量検索型グローバルトランザクションの開始は，独自コマンドで開始する，などが考えられる．

(b)大量検索型グローバルトランザクションと他トランザクションとの直列実行機能

WRITESET へのアクセスが競合する他のトランザクション（但し，開始時刻印の小さい他の大量検索型グローバルトランザクションの検索操作を除く）に関して，それらのトランザクションの実行と，大量検索型グローバルトランザクションの実行とが直列になるように制御することが必要である．具体的な実現方法としては，テーブルレベルの排他ロック（WRITE ロック）を用いる．WRITESET に属するレコードが属するテーブル（複数ありうる）全てに関して，大量検索型グローバルトランザクションの開始時に排他モードでのテーブルロックを取得し，完了時にロックを開

放することにより直列実行を実現する.

また，大量検索型グローバルトランザクションの排他ロック要求状況・取得状況を管理して,WRITESETが互いにオーバラップする大量検索型グローバルトランザクションの排他ロック取得はその開始時刻順になるよう制御することとする.

なお，大量検索型グローバルトランザクションの WRITESET と他の大量検索型グローバルトランザクションの検索操作の検索対象がオーバラップする場合の制御はグローバルな同期制御機能で制御する.

(c)ローカルな質問処理機能とのインタフェース機能

①(大量検索型グローバルトランザクションの READ 操作を除く)操作の処理依頼. ここで言う操作には，WRITESET に属するレコードの属するテーブルに関する，排他ロック命令も含む.

②依頼した操作の処理結果（ロック取得，検索結果，等）の通知の受信.

③トランザクション(ローカル及びグローバルの双方を含む)による更新発生時の，更新の発生と更新対象ページへのポインタとの通知の受信.

④ローカルトランザクションの完了通知の受信.

(2)グローバルな同期制御機能

提案方式の機能のうち，グローバルな同期制御機能で実現する部分を説明する.

(2-1)特定のサイト（中央サイトと呼ぶ）でのみ実現する機能

(a)グローバルな時計の管理機能.

(b)トランザクションの時刻印付与機能

①大量検索型グローバルトランザクション開始通知，ローカルトランザクションの完了通知，あるいは通常のグローバルトランザクションの完了通知を受信する.

②グローバルな時計を用いて時刻印を払い出す.

(c)グローバルデッドロック検出＆解消機能

①各サイトから定期的にトランザクション待ちグラフ情報を受け取る.

②トランザクション待ちグラフ情報を解析し，有向サイクル有無を確認し，もしサイクルがあれば，サイクル中の１つのトランザクションをアボートする．なお，大量検索型グローバルトランザクションはアボート対象外とする.

(2-2)各サイトで実現する機能

(a)大量検索型グローバルトランザクションの検索操作の処理（多版 READ）

適切な版を選択し，検索を実行する．この時，評価の節で説明するように，性能

向上のため，多版管理用インデクスを利用することも考えられる．

大量検索型グローバルトランザクション（仮にAと呼ぶ）のWRITESETと他の大量検索型グローバルトランザクション（仮にBと呼ぶ）の検索操作の検索対象がオーバラップする場合の制御は以下のように行う．

Aの開始時刻印＞Bの開始時刻印の場合は，Bによる検索を許す．(A以前のトランザクションで作成された版を読むことになる)

Aの開始時刻印＜Bの開始時刻印の場合は，Aの開始時点にBが既にAのWRITESETをアクセス済みならば，AはBの完了をウェイトし(Aはウェイト完了後に，Bよりも大きい開始時刻印を払い出してもらい，実行を開始する)，Aの開始時点にBがAのWRITESETに未アクセスならば，Aの完了までBの実行をウェイトさせる．

(b)版の作成・更新．

トランザクション(ローカル及びグローバルの双方を含む)による更新発生時に，版を新たに作成する，即ち更新対象ページを更新に先立ちコピーすることにより一つ前の版を作成し，更新対象ページを最新版として設定する．

大量検索型グローバルトランザクションによって版が作成される場合は，トランザクションの開始時刻印（中央サイトに依頼して払い出してもらう）を最新の版の時刻印として設定する．大量検索型グローバルトランザクションの完了時は，該トランザクションで作成された全ての最新版に関して完了情報を「完了」に設定する．

ローカルトランザクション及び通常のグローバルトランザクションの完了時は，該トランザクションで作成された全ての最新版に関して完了情報を「完了」に設定し，併せてトランザクションの完了時刻印（中央サイトに依頼して払い出してもらう）を版の時刻印として設定する．

なお，多版管理用インデクスを用いる場合は，版の作成に応じてインデクスのエントリを追加する．また，版管理情報（完了情報，時刻印）の変更に応じて，対応するインデクスエントリの版管理情報をも変更する．

(c)グローバルトランザクションのコミット機能

グローバルトランザクションに関して，2PCを用いて原子性の保証を行う．

(d)ローカルなトランザクション同期制御機能とのインタフェース機能

①サイト内のトランザクション待ちグラフを受信する．

②グローバルデッドロック解消のためのアボート命令を送信する．

③グローバルトランザクションのコミット用の命令を送信する．

(3)ローカルな言語処理機能の改造

提案方式実現のために以下の機能が必要となる．(a)(b)は，提案方式固有で必要となる機能ではなく，ローカルな言語処理機能を分散型の言語処理機能に拡張する場合は一般に必要となる機能である．一方，(c)は提案方式に固有で必要となる機能である．

(a)グローバルな言語処理機能から処理依頼を受けた操作（但し，大量検索型グローバルトランザクションのREAD操作を除く）の実行．

(b)グローバルな言語処理機能とのインタフェース機能

①操作の処理依頼の受信．

②依頼された操作の処理結果（ロック取得，検索結果，等）の通知．

(c)提案方式固有で必要となる，グローバルな言語処理機能とのインタフェース機能

①トランザクション(ローカル及びグローバルの双方を含む)による更新発生時の，更新の発生と更新対象ページへのポインタとの通知．

②ローカルトランザクションの完了の通知．

(4)ローカルなトランザクション同期制御機能の改造

提案方式実現のために以下の機能が必要となる．なお，これらの機能は，提案方式固有で必要となる機能ではなく，ローカルな同期制御機能を．(２相ロック方式と２相コミットメント方式とを用いる）分散型の同期制御機能に拡張する場合は一般に必要となる機能である．提案方式に固有で必要となる機能は特に無い．

(a)コミットメント対応機能，特に２PC対応機能

commit/abortのいずれにもできる状態を保持する機能，commit/abort命令に対応してcommit/abortする機能などからなる．なお，本機能は，前提条件より，各ローカルデータベース管理システムが備えていることを前提済みである．

(b)グローバルなトランザクション（但し，大量検索型グローバルトランザクションのREAD操作を除く）とローカルトランザクションとの，２相ロック方式を用いた同時実行の制御．

(c)グローバルな同期制御機能とのインタフェース機能

①サイト内のトランザクション待ちグラフの送信．

②グローバルデッドロック解消用のアボート命令の受信．

③グローバルトランザクションのコミット用の命令の受信．

4.4. 評価

提案方式と従来方式の性能比較，異種性解消方式の評価，及び提案方式の適用領域の考察を行う．

4.4.1. 従来方式との性能比較

先ず，性能評価モデルについて説明し，次いで性能評価結果について説明する．

4.4.1.1. 性能評価モデル

性能評価モデルについて説明する．

比較対象とする方式としては，ローカルデータベース管理システムの同期制御方式として２相ロック方式を前提条件としたことと，分散データベース管理システムの同期制御方式として２相ロック方式は一般に性能の良い方式として知られていることから，２相ロック方式を選択した．性能比較の方法としては，シミュレーションによって２相ロック方式と提案方式との比較を行った．シミュレーション言語は，Imagine That, Inc.の Extend を用いた．Extend は，GUI によるモデルの簡易な構築を可能とすることを特徴とするシミュレーション言語である．

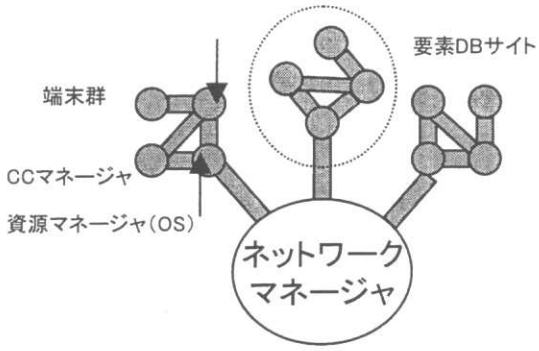


図 4.3 性能評価モデルの構造

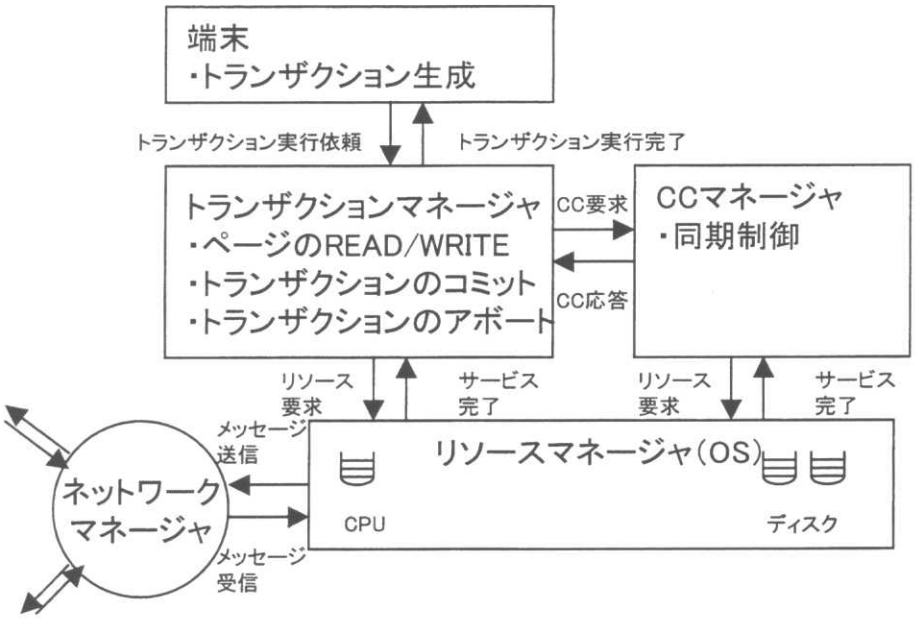


図 4.4 モデルの詳細構造

評価基準は，トランザクションのスループットとした．

モデルの構造としては，各サイト毎に①端末群，②トランザクションマネージャ，③同期制御マネージャ（CC マネージャ），④リソースマネージャの要素機能を有する構造とした（図 4.3，図 4.4）．なお，サイト毎の要素機能以外にネットワークマネージャがある．各要素機能の働きは以下の通りである．

・ 端末群：トランザクションの生成.
・ トランザクションマネージャ：トランザクションの実行の制御.
・ CCマネージャ：同期制御. 提案方式と2相ロック方式とで異なる.
・ リソースマネージャ：CPUとディスクの制御.
・ ネットワークマネージャ：通信の制御.

本モデルにおいては，ファイルアクセス性能向上と版アクセス性能向上を目的として，インデクスと多版管理用インデクスとの使用を仮定する.

以下，版の作成方法と，インデクスと多版管理用インデクスの構成と，ローカルトランザクションとグローバルトランザクションの振舞いと，ローカルトランザクションとグローバルトランザクションの組み合わせ方と，その他のパラメタの設定方法とついて順次説明する. なお，いずれのモデルも，以下の2つの観点から決定した.
・ 現実的であること
・ 提案方式が比較方式（2相ロック方式）と比べて有利にならないこと

(1)版の作成方法

版の作成方法に関しては，最も単純な方法である，各トランザクションで更新するレコードが含まれるページ毎に新たな版をページ単位で作成する方法，即ち更新対象ページを更新に先立ってコピーすることにより一つ前の版を作成し，さらに更新対象ページを最新版として設定する方法を採用することとする. なお，同一トランザクションが同一ページ内の複数レコードを更新しても作成される版は一つだけである.

(2)インデクスと多版管理用インデクスの構成

ファイルアクセス用のインデクスは，各ファイル（テーブル）毎に一つ存在し，その構造はB^+木構造であり，クラスタリングインデクス（インデクスのキー値順にレコードがファイル内で格納されているインデクス）であるとする. 木の段数は3段であり，インデクスをREADする際は，上位2段分は主記憶中に常にバッファリングされている，すなわちディスクアクセスが発生するのは最下段のみと仮定する.

多版管理用インデクスも，その構造はB^+木構造であるとする. ファイルアクセス用インデクスとの違いは，インデクス最下段ページにおいて各キー値対応に格納される情報が，ファイルアクセス用インデクスの場合はレコードへのポインタであり，多版管理用インデクスの場合は複数の版管理情報であることである[Lomet90],

[Kolvson89]．版管理情報は，版の格納アドレス，(版内の) レコードへのポインタ，版の時刻印，版の作成トランザクションの識別子などからなる．インデクスの容量が版の増加と共に増加するのを避けるため，いかなるトランザクションによってもアクセスされる可能性のなくなった古い版に対応する版管理情報は，適切な契機に削除されるものとする．この処理をごみ集め(garbage collection)と呼ぶ[Chan85][Bober92]．ごみ集めの詳細は本論文では省略する．

多版管理用のインデクスも木の段数は3段であるとする．インデクスをREADする際は，上位2段分は主記憶中に常にバッファリングされている，すなわちディスクアクセスが発生するのは最下段のみと仮定する．

(3)トランザクションの振舞い

先ず，両方の同期制御方式に共通的なトランザクションのアクセス特性，発行方法をトランザクション種別毎に説明する．次いで，トランザクションでのREADとWRITEの振舞いの詳細を，同期制御方式毎，トランザクション種別毎に分けて説明する．

なお，イントラネット環境においては，通信による遅延は大きくないことから，アクセスページ数 (READページ数とWRITEページ数) がほぼ同等の通常のグローバルトランザクションとローカルトランザクションとは，ほぼ同等な性能特性を示すと考えられる．このことより，シミュレーションの簡単化のため，ローカルトランザクションのみをモデルに含めることとし，通常のグローバルトランザクションはモデルには含めないこととした．

(3-1)トランザクションのアクセス特性と発行方法

(a)ローカルトランザクションのアクセス特性と発行方法

方式の性能差を明確にするためには，トランザクション同士の競合が無視できない割合で発生する必要がある．これを考慮し，以下のアクセス特性を持つトランザクション及びその発行方法を設定した．

- 1トランザクション当たりREAD10レコード，WRITE20レコード．
- READ/WRITE対象ファイルは各サイトの2つのファイル（第0，1ファイル）からランダムに選択． READ/WRITE 対象レコードの選択は，クラスタリングインデクスのキー値に関してランダムに選択．WRITE は，クラスタリングインデクスのキー値を変更しない更新（UPDATE）とする．
- 各端末では，トランザクションを一つずつ発行．先行するトランザクションの完

了後に，ThinkTimeだけ待ってから次のトランザクションを発行する．

(b)大量検索型グローバルトランザクションのアクセス特性と発行方法

ODS 用の情報の検索においては，データベースの全ページ数の数割に及ぶページを検索することがありうる．そこで，検索ページ数がデータベースの全ページ数に占める割合（選択度と呼ぶ）を測定パラメタとすることとした．

また，大量検索型グローバルトランザクションの更新対象ファイルは，同一サイトを更新対象サイトとする大量検索型グローバルトランザクション以外の他のトランザクションのアクセス対象ファイルと排他的となるように設定した．これは，意志決定支援用に収集された情報を，他のトランザクションがアクセスすることは現実的にはありえないと判断したためである．(なお,意志決定用に格納された情報のみを検索する新規トランザクションの存在が考えられるが，同期制御方式の性能比較の観点からは，そのようなトランザクションの有無は殆ど影響を与えないと判断し，モデルに陽に入れることはしなかった.)

- 1トランザクションでの検索サイト数は2サイトで，更新サイト数は1サイト．
- 各サイトでのREADは，各サイトの2つのファイル（第0，1ファイル）の総ページのうち選択度分を，クラスタリングインデクスのキー値順に連続的にREADするとする．
- WRITEはクラスタリングインデクスのキー値順のINSERTであり，挿入レコードは3ページ分（既存1ページを満杯にし，さらに空きページ2ページにレコードを追加する）とする．WRITE対象ファイルは第2ファイルとする．
- 各端末では，トランザクションを一つずつ発行．先行するトランザクションの完了後に，ThinkTimeだけ待ってから次のトランザクションを発行する．

(3-2)トランザクションのREAD/WRITE時の振舞い詳細

CASE　1．2相ロック方式の場合の振舞い

(a)ローカルトランザクションの振舞い

①READ時の振舞い

レコードのREAD時は，先ずインデクスをその最上位ページを起点にアクセスし，次いでレコードの含まれるページにアクセスする．

インデクスの上位ページは主記憶にバッファリングされているという仮定より，1レコードのREAD時に必要なディスクアクセスは，インデクスの最下段1ページ分と，

ファイル１ページ分の計２回である．

②WRITE 時の振舞い

レコードの WRITE 時は，先ず，WRITE 対象ページを読み込む．即ち，インデクスをその最上位ページを起点にアクセスし,次いでレコードの含まれるページを読み込む．

WRITE 対象ページを主記憶上で更新した後の処理として，WRITE 対象ページと，更新を反映したインデクスページのディスク書き込みを行う．ここで，本モデルでの WRITE は，キー値を変更しない更新（update）であると仮定したことより，インデクスページのディスク書き込みは不要となる．

従って，１レコードの WRITE 時に必要なディスクアクセスは，読み込み時の，インデクスの最下段１ページ分と，ファイル１ページ分，及び書き込み時のファイル１ページ分の計３回である．

(b)大量検索型グローバルトランザクションの振舞い

①READ 時の振舞い

最初のレコードの READ 時は,インデクスをその最上位ページを起点にアクセスし，検索範囲の先頭キー値に対応するインデクス最下段ページを読み込む．次いで，インデクス最下段ページをシーケンシャルに読みこみつつ，対応するレコードの含まれるページを読み込む．

インデクスがクラスタリングインデクスであることから，ファイル内ページの READ もシーケンシャルアクセスとなる．

従って，１トランザクションでの各サイトでの範囲指定検索に必要なディスクアクセスは，以下の和である．

インデクス最下段総ページ数＊選択度

サイト内通常アクセス総ページ数＊選択度

サイト内通常アクセス総ページ数は，２ファイルの総ページ数である．インデクス最下段総ページ数は，サイト内通常アクセス総ページ数の１割と仮定した．これは，キー長+ポインタ長が，レコード長のほぼ１割に相当することを意味する．これは，現実にありうる妥当な仮定である．

②WRITE 時の振舞い

前述したように，WRITE 対象ファイルは，大量検索型グローバルトランザクションの WRITE 専用とした．

最初のレコードの WRITE 時は，先ず，WRITE 対象ページを読み込む．即ち，インデ

クスをその最上位ページを起点にアクセスし，次いでキー値最大のレコードの含まれるページを読み込む．

その後は，WRITE対象ページ内にレコードを追加し，ページが満杯になったならば，空きページを払い出してレコードを追加する処理を繰り返す．これらの処理の後処理として，追加のあったページのディスク書き込みと，追加を反映したインデクスページのディスク書き込みを行う．ファイルページのディスク書き込みは(3-1)(b)で説明したように3ページである．インデクスページのディスク書き込みは，インデクスページの分割無しと仮定し，1ページとする．

従って，1トランザクションでの全レコードのWRITE時に必要なディスクアクセスは，読み込み時の，インデクスの最下段1ページ分と，ファイル1ページ分，及び書き込み時の，ファイルページ3ページ分とインデクス最下位1ページ分の計6回である．

CASE　2．提案方式の場合の振舞い

(a)ローカルトランザクション及び通常のグローバルトランザクションの振舞い

①READ時の振舞い

レコードのREAD時は，先ずインデクスをその最上位ページを起点にアクセスし，次いでレコードの含まれるページにアクセスする．

インデクスの上位ページは主記憶にバッファリングされているという仮定より，1レコードのREAD時に必要なディスクアクセスは，インデクスの最下段1ページ分と，ファイル1ページ分の計2回である．

②WRITE時の振舞い

レコードのWRITE時は，先ず，WRITE対象ページを読み込む．即ち，インデクスをその最上位ページを起点にアクセスし,次いでレコードの含まれるページを読み込む．

WRITE対象ページを主記憶上で更新した後の処理として，WRITE対象ページと，更新を反映したインデクスページのディスク書き込みを行う．ここで，本モデルでのWRITEは，キー値を変更しない更新（update）であると仮定したことより，インデクスページのディスク書き込みは不要となる．

提案方式においては，さらに版の作成と，多版管理用インデクスの更新とが必要になる．版の作成は，読み込み済みの更新対象ページを更新に先立ってコピーすることにより一つ前の版を作成し,更新対象ページを最新版として設定することにより行う．多版管理用インデクスの更新においては，先ずキー値に対応するインデクス最下段ペ

ージにアクセスし，次いで最下段ページのキー値に関するエントリの修正（版管理情報の追加）を行う．ここで，インデクス最下段ページの内容更新に伴い，１０回に１回の頻度でインデクスページの分割が発生するものとする．以上より，版管理に必要なディスクアクセスは，読み込み時の，多版管理用インデクス最下段１ページ分と，書き込み時の多版管理用インデクス最下段ページ１．１ページ分と，版１ページ分との計３．１回である．

従って，１レコードのWRITE時に必要なディスクアクセスは，読み込み時のインデクスの最下段１ページ分とファイル１ページ分と，書き込み時のファイル１ページ分と，版管理用の３．１回との計６．１回である．

(b)大量検索型グローバルトランザクションの振舞い

①READ時の振舞い

レコードの READ は，各サイトのファイル内の一定の範囲をキー値順に連続的に多版READすると仮定する．

最初のレコードの READ 時は，多版管理用インデクスをその最上位ページを起点にアクセスし，検索範囲の先頭キー値に対応するインデクス最下段ページを読み込む．次いで，インデクス最下段ページをシーケンシャルに読みこみつつ，対応するレコードの含まれるページ（版）を読み込む．

ファイル内のレコードがキー順に格納されていることから，ファイル内ページのREADもシーケンシャルアクセスとなる．但し，選択された版が最新版（ファイル内ページ）でない場合は，古い版を代わってアクセスする．

従って，１トランザクションでの各サイトでの範囲指定検索に必要なディスクアクセスは，以下の和である．

多版管理用インデクス最下段総ページ数＊選択度

サイト内通常アクセス総ページ数＊選択度

サイト内通常アクセス総ページ数は，２ファイルの総ページ数である．

多版管理用インデクス最下段総ページ数は，サイト内通常アクセス総ページ数の５割と仮定した．これは，キー毎に管理されている複数の版管理情報が，レコード長のほぼ５割に相当することを意味する．これは，現実にありうる妥当な仮定である．

②WRITE時の振舞い

前述したように，WRITE 対象ファイルは，大量検索型グローバルトランザクションのWRITE専用とした．

最初のレコードのWRITE時は，先ず，WRITE対象ページを読み込む．即ち，多版管理用インデクスをその最上位ページを起点にアクセスし，次いでキー値最大のレコードの含まれるページにアクセスする．

その後は，WRITE対象ページ内にレコードを追加し，ページが満杯になったならば，空きページを払い出してレコードを追加する処理を繰り返す．これらの処理の後処理として，追加のあったページのディスク書き込みと，追加を反映したインデクスページのディスク書き込みを行う．ファイルページのディスク書き込みは(3-1)(b)で説明したように3ページである．インデクスページのディスク書き込みは，インデクスページの分割無しと仮定し，1ページとする．

提案方式においては，さらに版の作成と，多版管理用インデクスの更新とが必要になる．版の作成は，読み込み済みの更新対象ページを更新に先立ってコピーすることにより一つ前の版を作成し，更新対象ページを最新版として設定することにより行う．なお，空きページに新たに追加を行った分に関しては，古い版（空きページを内容とする版）の作成は不要と仮定した．多版管理用インデクスの更新においては，先ずキー値に対応する多版管理用インデクス最下段ページにアクセスし，次いで最下段ページのキー値に関するエントリの修正（版管理情報の追加）を行う．ここで，多版管理用インデクス最下段ページの内容更新に伴い，インデクスページの分割が1回発生するものとする．以上より，版管理に必要なディスクアクセスは，読み込み時の，多版管理用インデクス最下段1ページ分と，書き込み時の多版管理用インデクス最下段ページ2ページ分と，版1ページ分との計4回である．

従って，1トランザクションでの全レコードのWRITE時に必要なディスクアクセスは，読み込み時の，インデクスの最下段1ページ分とファイル1ページ分と，書き込み時のファイル3ページ分とインデクス最下位1ページ分と，版管理用の4ページ分との計10回である．

(4) ローカルトランザクションとグローバルトランザクションの組合せ方

端末の一定割合をローカルトランザクションのみ発行する端末とし，残りをグローバルトランザクションのみ発行する端末とする．

(5) その他のパラメタの設定

その他のシミュレーションパラメタは，前述した2つの観点（現実的であることと提案方式が比較方式（2相ロック方式）と比べて有利にならないこと）に加えて，ト

ランザクション同士の競合が無視できない割合で起こり得るということの計3つの観点から設定した.

サイト数は3, サイト当りページ数は4800 (3ファイル, 各ファイル1600ページ) とした.

グローバルトランザクションとローカルトランザクションの ThinkTime は平均 10 秒とし, 逆指数分布に従うものとした.

サイト当りディスク数は2とし,それぞれ2400ページずつの容量を持つものとした. 各ディスクは並行動作可能とした.

選択度はシミュレーションの実行毎に一定の値を用いるのではなく,区間[0.5,1.5]の一様乱数を乗じた値を用いることとした.

ディスクアクセス性能は, 最小10ミリ秒, 最大20ミリ秒の間で一様分布する時間とした.

ディスクアクセスに伴うCPU時間は8ミリ秒とした.

メッセージ処理CPU時間, デッドロック処理CPU時間はそれぞれ1ミリ秒, 5ミリ秒とした.

同期制御に要するCPU時間 (CCマネージャでの処理に要するCPU時間) は無視可能 (0秒) とした.

デッドロックのチェック間隔は, 0.3秒とした.

シミュレーションのパラメタを表4.2にまとめる.

表 4.2　シミュレーションの主要パラメタ

サイト数	3
サイト当りページ数	4800
端末数	13(ローカル12, グローバル1)
ローカルトランザクションのThinkTime	平均10秒(逆指数分布)
グローバルトランザクションのThinkTime	平均10秒(逆指数分布)
サイト当りディスク数	2
ディスクアクセス性能	10～20ミリ秒(一様分布)
ディスクアクセス関連CPU処理時間	8ミリ秒
メッセージ処理CPU処理時間	1ミリ秒
同期制御要求CPU処理時間	0ミリ秒
デッドロックチェック間隔(2PLの場合のみ)	0.3秒
デッドロックチェックCPU時間	5ミリ秒

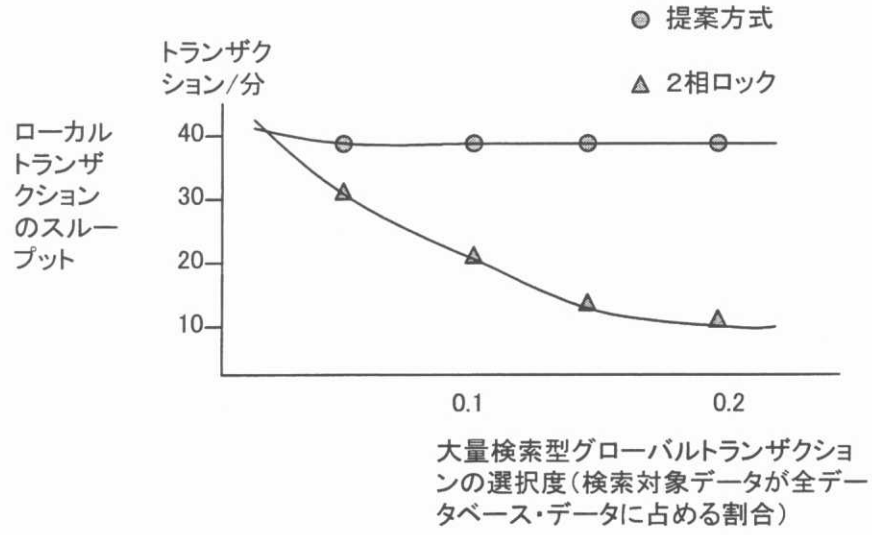


図 4.5 ローカルトランザクションのスループット

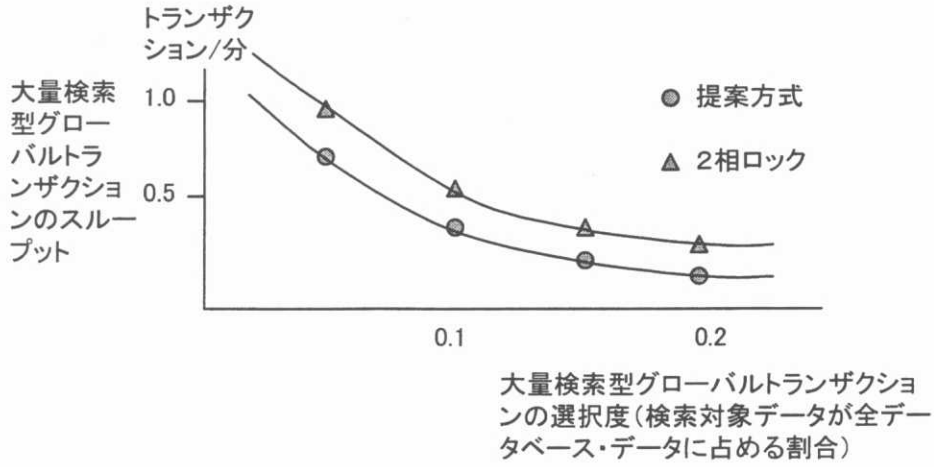


図 4.6 大量検索型グローバルトランザクションのスループット

4.4.1.2. 性能評価結果

シミュレーション結果を図 4.5～4.6 に示す.

シミュレーション結果より以下が分かる.

(1)ローカルトランザクションのスループットに関しては,提案方式の方が2相ロック方式よりも大きく（最大約7割）優れる.

特に選択度（検索対象データが全データに占める割合）が大きい場合に，差が顕著となる.

(2)大量検索型グローバルトランザクションのスループットに関しては,2相ロック方式の方が提案方式よりも多少（最大約3割）優れる.

ローカルトランザクションのスループットに関して提案方式の方が優れること(1)の原因は，2相ロック方式の場合は，大量検索型グローバルトランザクションのREADによって多数の通常のトランザクションが長時間ウェイトさせられるのに対して，提案方式の場合は,大量検索型グローバルトランザクションのREADによって他の通常のトランザクションがウェイトさせられることが無いからである.

大量検索型グローバルトランザクションのスループットに関して2相ロックの方が優れること(2)の原因は，2相ロック方式の場合は，大量検索型グローバルトランザクションのREADがある程度進んで来ると,その大量検索型グローバルトランザクションにウェイトさせられるローカルトランザクションの数が増えてくるため，大量検索型グローバルトランザクションに資源（ディスク，CPU）の使用順番が直ぐ回ってくるのに対して，提案方式の場合は前記ウェイトが生じないので大量検索型グローバルトランザクションに資源の使用順番がなかなか回って来ないためである.

以上より，提案方式は，多数の通常のトランザクションと少数の大量検索型グローバルトランザクションが共存する環境において，双方を高性能で走行させる方式であること，特に，ミッションクリティカルなものが多数存在する通常のトランザクションの高性能な実行を可能とする同期制御方式であることが示されたと考える.

4.4.2. 異種性解消方式の評価

質問処理機能とトランザクション同期制御機能とは，データベース管理システムの機能の中でもアルゴリズム的に難度の高い機能であり，改造を加えることは一般には容易ではない.

しかしながら，提案した異種性解消方式における，既存の質問処理機能とトランザクション同期制御機能とで実現すべき改造機能は，①アルゴリズム的にさほど困難ではなく，②規模はさほど大きくなく，かつ③質問処理機能とトランザクション同期制御機能との残余の機能と独立性の高い（即ち残余の機能への影響の小さい）機能であ

ると言えると考える．

従って，提案する異種性解消方式により，既存データベース管理システムの既存機能に大きな影響を与えることなく，提案するトランザクション同期制御方式を実現可能と言えるものと考える．

4.4.3. 適用領域に関する考察

性能評価結果より，提案方式は，通常のトランザクションと大量検索型グローバルトランザクションが共存する環境において，双方を高性能で走行させる方式であること，特に，ミッションクリティカルなものが多数存在する通常のトランザクションの高性能な実行を可能とする同期制御方式であることが示された．

また，データベースシステムの設計・構築の経験または設計・構築の支援の経験のある者（3名）にヒアリングを行い，以下の見解を得た．

データウェアハウスの重要性は広く認識されつつあり，社内外でODSが構築されるケースは今後増えるものと考える．

それらのケースのうち,遅延の少ないデータロードを可能にすることを狙いとして,データローダをトランザクション技術を用いて実現するケースは少なからぬ割合で存在すると考える．

従って，通常のトランザクションと大量検索型グローバルトランザクションが共存する環境はしばしば生じうると考える．

以上をまとめると,本研究の成果は一定の適用領域を有すると言えるものと考える．

4.5. 4章のまとめ

ODSのオンラインデータローダのアクセス特性は,多量のデータを基幹系データベースシステムから検索し，少量の更新をODSに行うことである．ODSのデータローダを始めとする，多量の検索と少量の更新を行うグローバルトランザクションを大量検索型グローバルトランザクションと呼ぶこととする．

本章では，通常のトランザクションと大量検索型グローバルトランザクションとが共存する環境向けのトランザクション同期制御方式を検討した．特に，ミッションク

リティカルなものが数多く存在する通常のトランザクションの高性能な実行を可能とする方式を検討した.

併せて，異種性解消方式，即ちローカルなデータベース管理システム間の異種性を簡易に解消して,提案するトランザクション同期制御方式を実現する方式を検討した.

グローバルトランザクションが検索専用の場合は，同期制御に起因する副作用が小さい（アボートは一切発生せず，他のトランザクションとの間で待ちが発生するのはREADの実行時に，未完了のトランザクションによって作成された版を選択しそのトランザクションの完了を待つ場合のみ）という特長を持つ方式として多版時刻印方式があることに着目し，該方式と同期制御方式の主流である２相ロック方式とをベースに新たな同期制御方式を提案した．提案した方式は以下の特徴を有する.

- ローカルトランザクションと通常のグローバルトランザクションとのREAD／WRITEの制御には２相ロック方式を修正適用する.
- 大量検索型グローバルトランザクションのREADの制御には多版時刻印方式を修正適用する.
- さらに，大量検索型グローバルトランザクションはそのWRITESET（トランザクションでの更新対象テーブルの和）に関して，他のトランザクションと直列に実行させることによって，全トランザクションの直列化可能性を保証する.

また，異種性解消方式として，ローカルなデータベースシステムの質問処理機能および同期制御機能に改造を加え，外付けの機能と協調動作させることによって提案同期制御方式を実現する方式を提案した.

提案方式と従来方式（２相ロック方式）との比較をシミュレーションによって行い，以下の結果を得た.

- 通常のトランザクションの性能（スループット）は提案方式の方が大幅に（最大約７割）良い．但し，大量検索型グローバルトランザクションの性能は提案方式の方が多少（最大約３割）悪い.

提案方式は，通常のトランザクションと大量検索型グローバルトランザクションとが共存する環境において優れた同期制御方式であること，特に，ミッションクリティカルなものが多数存在する通常のトランザクションの高性能な実行を可能とする同期制御方式であることが示された.

また，異種性解消方式に関しては，既存機能への改造が，①アルゴリズム的にさほど困難ではなく，②規模はさほど大きくなく，かつ③質問処理機能とトランザクション同期制御機能との残余の機能と独立性の高い（即ち残余の機能への影響の小さい）

機能であり，従って簡易に異種性を解消可能であることを示した．

今後ODSが構築されるケースは増えるものと考えられ，それらのケースのうちデータローダをトランザクション技術を用いて実現するケースは少なからぬ割合で存在すると考えられ，従って通常のトランザクションと大量検索型グローバルトランザクションが共存する環境はしばしば生じうると考えられる．このことから，提案方式は一定の適用領域を有すると言えるものと考える．