

モバイル・コンピューティング環境における
アクティブ型マルチデータベースの実現方式

倉林 修一† 石橋 直樹†† 清木 康†

†慶應義塾大学 環境情報学部

††慶應義塾大学大学院 政策・メディア研究科

〒 252-0816 神奈川県藤沢市遠藤 5322 慶應義塾大学 4 501

TEL: 0466-47-5000(53251)

E-mail: alpha@mdbl.sfc.keio.ac.jp

あらし

ネットワークで接続された異種のレガシーデータベース群の相互接続機能により、それらのデータベース群の価値は著しく高められる。一方、それらのデータベースに対して行われる更新・変更操作は、データベースの利用者にとって非常に重要な意味を持つことが多く、データベースの変化に応じたリアルタイム情報の提供が重要である。しかしながら、既存の受動的なデータベースをアクティブデータベースとして利用する為には大規模なシステムの変更を要するため、容易に実現できない。本稿では、システムへ接続された受動的、又は、能動的(active)データベースをアクティブデータベースとして利用し、動的に相互接続するアクティブ・メタレベルシステムを提案する。提案システムにより相互に連結されたローカルデータベースシステム群は、指定された情報をリアルタイムにユーザに提供することが可能となる。

キーワード マルチデータベースシステム、アクティブデータベース、分散データベース

Active Multidatabase System for Mobile Computing Environment

Shuichi Kurabayashi† Naoki Ishibashi†† Yasushi Kiyoki†

†Faculty of Environmental Information, KEIO University

††Graduate School of Media and Governance, KEIO University

5322 Endo, Fujisawa, Kanagawa, Japan, 252-0816

Keio University 4 501

TEL: 0466-47-5000(53251)

E-mail: alpha@mdbl.sfc.keio.ac.jp

Abstract

In a wide area mobile computing environment, facilities for providing real time information are very important. It is also important to provide information based on changes of databases automatically. Traditional methods to use legacy databases as active databases require the modification of entire systems. In this paper, we present a new meta-level system that integrates heterogeneous legacy databases in a sense of active databases.

key words

Multidatabase System, Active Database, Distributed Database

1 はじめに

今日、急速なネットワークの広域化とデータベース・テクノロジーの発展に伴い、膨大な数のレガシーデータベース群が広域ネットワークに接続されている。これらの多くはユーザの要求に応じて検索結果を返すことを基本とし、要求に応じて起動される受動的なデータベースである。これに対し、データベースの検索、更新などといったイベントに反応して、別の検索、更新などといったアクションを起動するイベント駆動型のデータベースシステムは、アクティブデータベースシステムと呼ばれる [6]。

本稿では、既存の受動的データベースシステム、及び、アクティブデータベースシステムを、統一的にアクティブデータベースとして利用可能にするアクティブ・メタレベルシステムの実現方式を提案する。これを、アクティブ型マルチデータベースシステムと呼ぶ。また、本稿では、既存のデータベースシステムをアクティブデータベースとして利用することを、単にアクティブ化と記す。

本方式は、次の3つの問題点を背景としている。第一に、本方式は、既存の受動的データベースシステム群にアクティブデータベース機構を適用することで、それらのデータベースシステム群をアクティブ・メタレベルシステムに連結する。従来手法により受動的なデータベースシステム群をアクティブデータベースとして利用するためには、アクティブデータベース機構を有するシステム上にデータベースを再構築する必要がある。また、既存のアクティブデータベース群は、その機能・利用方法共に大きく異なる場合が多い。そのため、既存の手法では各ローカルデータベース群のアクティブデータベース機構の利用方法、及びその特性などをメタレベルシステムが常に意識する必要がある。膨大な受動的データベース群が接続された広域ネットワーク環境では、これらの制約は極めて厳しいといえる。

第二に、今日、データベース管理機構によって管理されない膨大なデータ群が、広域ネットワークを介して利用可能である。これらは、例えば、XMLドキュメントや、PDAが格納するデータなどが挙げられる。これらのデータ群を既存データベースシステムとして、マルチデータベースシステムから統一的に利用できれば、マルチデータベースシステムが対象とするローカルデータベースシステム群の規模は大きなものとなり、既存データベースシステムの価値を飛躍的に高めることができる。しかし、従来手法によりこれらのデータ群をマルチデータベースシステムから利用するためには、データ群をデータベース管理機構を備えたデータベースシステム上に再構築する必要がある。膨大なドキュメントが存在する広域ネットワーク上では、このような再構築は困難である。

第三に、アクティブデータベース機構を適用するために記述されるデータベース間の関連性は多様であり、従来方式では異種データベース間の関連性を個々のデータアイテムに対応するパターンの対として静的に記述し、その静的な記述に対する同一性の評価により異種データベース群を結合するため、この静的な記述にオーバーヘッドが生ずる。特に、広域ネットワーク環境では、膨大なレガシーデータベース群が接続されているため、このオーバーヘッ

表 1: ローカルデータベースの異種性

異種性の段階	問い合わせ言語	DBの機能
異種性 1	同種	同種
異種性 2-1	異種	同種
異種性 2-2	同種	異種
異種性 3	異種	異種

ドはより厳しいものとなる。

また、本稿で提案するアクティブ・メタレベルシステムの特徴は、次の3つである。

第一に、本方式の適用により、既存の受動的データベースシステムをアクティブデータベースとして利用することが可能となる。また、本方式は、ローカルデータベースシステムのアクティブ化を最適化する。これは、既存データベースシステムがアクティブデータベース機構を有している場合、既存データベースシステムのアクティブデータベース機構を利用することで、各ローカルデータベースシステムに最適化されたアクティブ化を実現する。

第二に、本方式はデータベース管理機構によって管理されないデータ群を、既存ローカルデータベースシステムとして利用可能にする。これにより、XMLドキュメントなどのWWW上のデータやGPSレシーバ、PDA上に格納されているデータ群がローカルデータベースシステムとして利用可能となる。また、本方式は異種のアクティブデータベース群を統一的に扱う枠組みを提供する。これは、次の2種類の異種性を抽象化することで実現される。すなわち、1) 問い合わせ言語の異種性、及び、2) データベースシステムの機能の異種性である。ここでは、本方式が対象とするローカルデータベースシステムの異種性を3段階に分類し、表1に示した。

ローカルデータベース群が同種の問い合わせ言語を持つデータベースシステムから構成される場合とは、例えば、ローカルデータベースが方言のないSQLで問い合わせ可能であるということである。また、ローカルデータベース群が同種の機能を持つデータベースシステムから構成される場合とは、例えば、すべてのローカルデータベースがトリガ機能を備えているような場合である。また、同種の問い合わせ言語、及び、同種の機能を持つ場合であっても、多くの場合、データベースシステムのAPIはベンダごとに異なるものであり、その異種性を吸収する必要がある。このAPIの異種性を吸収するものに、JDBC[10]やODBC[11]等が挙げられる。

表1における異種性3を抽象化する枠組みと第一の特徴により、ローカルデータベースシステム群のデータベース管理機構の有無にかかわらず、また、ローカルデータベースシステム群が受動的/能動的のいずれであっても、相互に連結され、統一的にアクティブデータベースとして利用可能となる。

第三に、本方式はローカルデータベースシステム群を動的に連結する。これは、ユーザからの問い合わせに応じて、接続するローカルデータベースを動的に選択する機能により実現される。これにより、メタレベルシステムにローカルデータベースシステムの複製を作成せずにシス

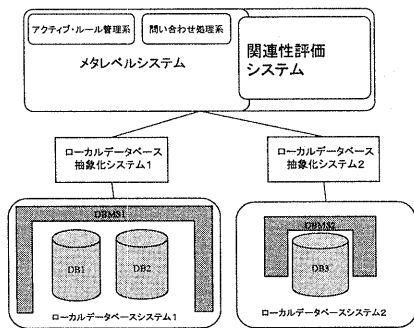


図 1: システム構成図

テムを運用することが可能となる。また、本メタレベルシステムは、データ間の多様な関連性を評価する機構を有する。これにより、データ間の多様な関連性に応じた状況を記述でき、また、多様な関連性を含む問い合わせを動的に発行できる [4]。この特徴により、データ間の多様な意味、多様な関連性を反映させたアクティブ化が実現できる。

2 提案アクティブ・マルチデータベースシステムの構成方式

本方式が対象とする既存データベースシステムは2種類に分別される。一つはデータベース管理機構により検索機能・データ管理機能を提供しているデータベースシステムであり、一つは、データ管理機構をもたないデータ群である。本方式では、これらの多種多様な既存データベースシステムの異種性を抽象化することで、統合的なアクティブ化を実現する。

また、本方式におけるアクティブ・ルールを、メタレベル・アクティブルールと呼ぶ。これは、ローカルデータベース群よりも一段抽象度の高いメタレベルシステムにおいて管理されるアクティブ・ルールを示す。

図1に本システムのアーキテクチャを示す。また、本方式における、メタレベル・アクティブ・ルール管理方式と、ローカルデータベースの抽象化方式について述べる。

2.1 システム・アーキテクチャ

本方式では、システム全体を次の3システムにより構成する。

メタレベルシステム

メタレベルシステムは次の3機構を持つ。

1. マルチデータベース問い合わせ処理系

これは、ユーザから発行されたマルチデータベースへの問い合わせを分散並列的に処理する。また、問い合わせ処理は関数方計算の枠組みで最適化され、実行される [5]。

2. メタレベル・アクティブ・ルールの管理・適用系

これは、既存データベースよりも一段抽象度の高いメタレベルシステムにおいてアクティブルールを管理する。これにより、異種データベースより発行されるイベント間に内在する新たな意味を抽出し、その新たな意味に基づいた情報をユーザへ発信することが可能となる。このルール管理方式に関する詳細については、次節で述べる。

3. 関連性評価システム

これは、メタレベルシステムにおけるデータの計量を行う。これにより、異種のデータモデルから構築された既存データベース間に内在する多様な関連性を計算し、新たな情報生成を行うことが可能となる [4, 9]。

ローカルデータベース抽象化システム

ローカルデータベース抽象化システムは、次の3特性を持つ。1) 多種多様なローカルデータベースシステムの異種性(表1)を吸収する。2) ローカルデータベースシステムが受動的/能動的のいずれであっても、アクティブデータベースとして利用可能にする。また、機能1と機能2を組み合わせ、多種多様なローカルデータベースシステムをアクティブデータベースとして利用可能にする。この2機能を提供する機構を、アクティブ化機構と呼ぶ。3) ローカルデータベースシステムへの操作を Primitive 関数としてメタレベルシステムへ提供する。

ローカルデータベースシステム群

ローカルデータベースシステム群は多種多様なデータベースシステム群から構成される。

2.2 メタレベル・アクティブ・ルールの管理・適用方式

メタレベル・アクティブ・ルールはECAルール [6] に基づいている。また、本方式では、Event は多種多様なローカルデータベース群の更新・変更操作と定義する。Condition は異種データベース間に内在する多様な関連性により記述される状況と定義する。Action は異種データベース群に対し多様な関連性評価機能を適用し、実行される連結操作又は検索操作と定義する。このECAルールをメタレベルシステムが管理・適用を行うことにより、多種多様なローカルデータベース群を一つのアクティブデータベースとして統一的に扱うことが可能となる。

2.2.1 メタレベル・アクティブ・ルールの適用のフロー

本方式における、メタレベル・アクティブ・ルールの適用のフローについて述べる。メタレベルシステムへ入力されたメタレベル・アクティブ・ルールは、まず、対応するローカルデータベースシステムへの部分ルールへと分割される(図2)。次に、部分ルールは各ローカルデータ

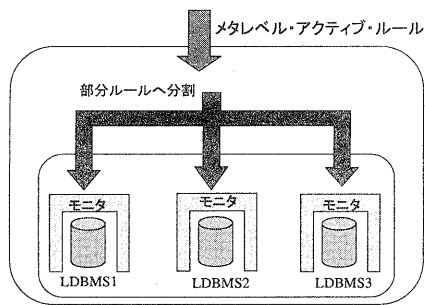


図 2: メタレベル・アクティブ・ルールの適用方式

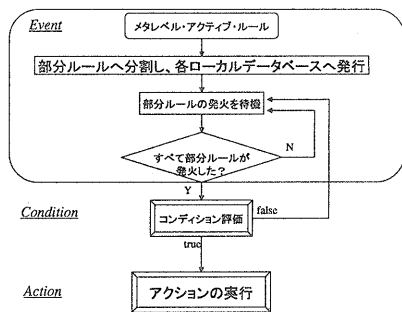


図 3: メタレベル・アクティブ・ルールの管理方式

ベースシステムに送信され、分散処理される。各ローカルデータベースシステムでの部分ルールの発火はイベントとしてメタレベルシステムへ通知される。メタレベル・アクティブ・ルールが必要とするすべてのイベントがメタレベルシステムへ通知された場合、メタレベルシステムの関連性評価システムにおいてコンディション評価が行われる。コンディション評価の結果が真の値であった場合、マルチデータベース問い合わせがアクションとして実行される。コンディション評価の結果が偽の値であった場合、次のイベント発火への待機状態へと戻る (図 3)。

2.2.2 ローカルデータベースシステムのアクティブ化方式

本方式はアクティブ化機構により、既存データベースシステムが受動的／能動的のいずれであっても統一的にアクティブ化する。アクティブ化機構は、各ローカルデータベースシステムにおける更新・検索等の操作を監視する、データベース監視機構を持つ。これを、モニタと呼ぶ。モニタはローカルデータベースの変化を検知し、メタレベルシステムへ通知する。本方式では、モニタの起動・停止・イベント発生をメタレベルシステムから制御する。これにより、ローカルデータベースシステムが能動的／受動的のいずれであっても、相互に連結されるアクティブ型マルチデータベースシステムを実現することができる。

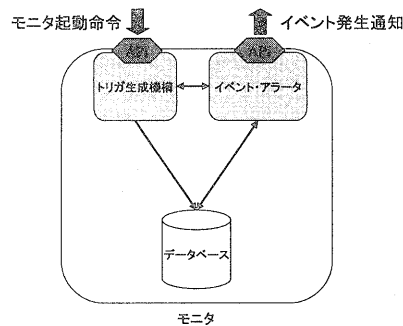


図 4: ローカルデータベースを監視するモニタ

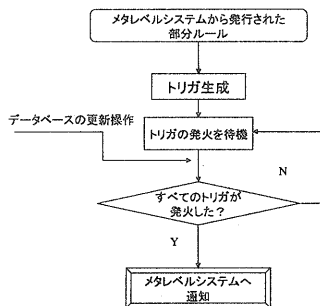


図 5: モニタの動作

モニタは図 4 のように、次の 2 システムから構成される。

- トリガ生成機構
- イベント・アラータ

トリガ生成機構はメタレベルシステムから発行された部分ルールに応じてデータベースを監視するトリガを生成する。このとき、トリガ生成機構はトリガを次の 2 通りの方法で最適化する。

- イベント検出に必要な最小限のデータを監視する最適化方式
- ローカルデータベースシステムのアクティブデータベース機構を利用する最適化方式

イベント検出に必要な最小限のデータを監視する最適化方式は、部分ルールが必要とするデータの最小限度を監視対象にする。これにより、監視コストの低いトリガを生成することが可能となる。これはローカルデータベースシステムの構成に依存しないため、ローカルデータベースシステムが能動的／受動的のいずれであっても適用可能な最適化方式である。

ローカルデータベースシステムのアクティブデータベース機構を利用する最適化方式は、ローカルデータベースシステムが備えるアクティブデータベース機構を用いてトリガを生成する。これにより、ローカルデータベースシ

システムが能動的である場合にトリガの最適化が可能となる。イベントアラータは、トリガ生成機構によって生成されたトリガの発火を検知し、メタレベルシステムへ通知するシステムである(図5)。

2.3 ローカルデータベースの抽象化方式

本方式はローカルデータベースを抽象化することでその異種性を解消する。この抽象化機能は2つに分別される。一つは、既存データベースシステムの固有の問い合わせ言語を抽象化する。これにより、統一的な問い合わせ言語を提供することが可能となる。二つは、既存データベースシステムの固有の機能を抽象化する。これにより、ローカルデータベースの振る舞いを統一化することが可能となる。この2抽象化機能を組み合わせたローカルデータベースの抽象化は次の2レベルに分けられる。

Level 1-1 既存データベース固有の問い合わせ言語のみの抽象化

Level 1-2 既存データベース固有の機能のみの抽象化

Level 2 既存データベース固有の問い合わせ言語、及び、機能の抽象化

ここでは、上記2レベルの抽象化方式について、それぞれの特徴を述べる。

特徴 1-1 ローカルデータベース固有の問い合わせ言語の抽象化

既存データベース固有の問い合わせ言語を抽象化することにより、異種の問い合わせ言語を持ち、同種の機能を持つデータベースシステムから構成されるローカルデータベースシステムを統一的に利用することができる。この抽象化機構は、表1における異種性2-1を抽象化する。これにより、能動的なローカルデータベースシステムをアクティブ化し、相互に連結するメタレベルシステムを実現することが可能となる。例えば、ローカルデータベースシステムがRDBで構成される場合、各ローカルデータベースシステムの問い合わせ言語であるSQLの方言や、SQLに含まれないトリガ言語の異種性を吸収することによりアクティブ化することが可能である。

特徴 1-2 ローカルデータベース固有の機能の抽象化

既存データベース固有の機能を抽象化することにより、同種の問い合わせ言語を持ち、(または、問い合わせ言語をもたない)、異種の機能を持つデータベースシステムから構成されるローカルデータベースシステムを統一的に利用することができる。この抽象化機構は、表1における異種性2-2を抽象化する。これにより、受動的なローカルデータベースシステムをアクティブ化し、相互に連結するメタレベルシステムを実現することが可能となる。例えば、ローカルデータベースシステムがXMLドキュメント、トリガ機能を備えないRDB、GPSレーザなどで構成され

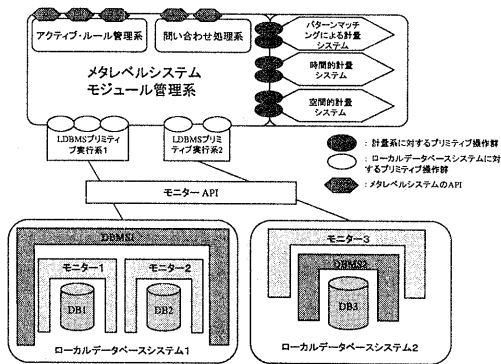


図6: システム図

る場合、各ローカルデータベースシステムのネットワーク接続機能・データ検索機能・データベース監視機能などの異種性を吸収することによりアクティブ化することが可能である。

特徴 2 ローカルデータベース固有の問い合わせ言語、及び、機能の抽象化

特徴1-1、及び、特徴1-2を組み合わせることにより、異種の問い合わせ言語、異種の機能を持つデータベースシステムから構成されるローカルデータベースシステムを統一的に利用することができる。この抽象化機構は、表1における異種性3を吸収する。これにより、ローカルデータベースシステムが受動的/能動的のいずれであっても統一的にアクティブ化し、相互に連結するメタレベルシステムを実現することが可能となる。例えば、ローカルデータベースシステムがトリガ機能を持つRDB、XMLドキュメント、GPSレーザから構成される場合、各ローカルデータベースの問い合わせ言語の異種性、及び、ネットワーク接続・データ検索・データベース監視機構の異種性を吸収することによりアクティブ化することが可能である。

3 提案アクティブ・マルチデータベースシステムの実現方式

3.1 システム構成

本方式の実現システムとして、次の4システムにより構成されるシステムを構築した(図6)。また、実装にはJava言語を用いた。

- メタレベルシステム
- 関数インタプリタ
- モニター

- ローカルデータベースシステム

また、このうちメタレベルシステムは、次の4システムにより構成される。

- メタレベル問い合わせ処理系

これは、ユーザやアプリケーションプログラムから発行された、メタレベルシステムへの問い合わせを解釈・実行するシステムである。

- メタレベル・アクティブ・ルール管理系

メタレベル・アクティブ・ルール管理系は、ユーザやアプリケーションプログラムから発行されたメタレベルシステムにおけるアクティブルールを管理・適用するシステムである。

- メタレベルシステム・モジュール管理系

これは、ユーザーやアプリケーションプログラムからの要求に応じて、メタレベル問い合わせ処理系や、メタレベル・アクティブ・ルール管理系をメタレベルシステムへ組み込むシステムであり、システムの柔軟性・拡張性を保証している。

- 関連性評価システム

これは、データ間に内在する関連性を評価するシステムである。

ローカルデータベースシステムは、多種多様なデータモデル、データ操作機能を持つ異種データベースから構成される。

モニターは、次の2モジュールから構成される。

- トリガ生成機構

- モニター API

トリガ生成機構は次の2方式でローカルデータベースを監視する。1) ローカルデータベースシステムが能動的である場合は、ローカルデータベースシステムのアクティブデータベース機構を利用するトリガを動的に生成することで、データベースイベントの監視を行う。2) ローカルデータベースシステムが受動的である場合は、ローカルデータベースシステムの外部にデータベース監視プログラムを配置することでデータベースイベントの監視を行う。モニター API は関数インタプリタとモニター間の通信を抽象化し、ローカルデータベースシステム内部のモニターとローカルデータベースシステム外部のモニターの異種性を吸収する。関数インタプリタは、モニター API を経由してモニターの制御を行い、モニターはモニター API を経由してイベント情報を関数インタプリタへ伝える。

3.2 ローカルデータベースの抽象化機構の実装

実現システムでは、ローカルデータベースシステム抽象化機構の実装として、メタ JDBC を定義した(図7)。これは、通常の JDBC よりも一段抽象度の高いクラスライブラリで JDBC を補完するフレームワークである。こ

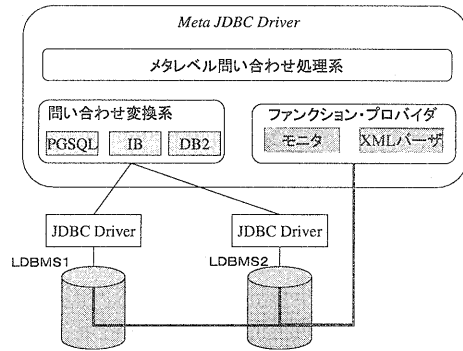


図7: メタ JDBC 構成図

れにより、ローカルデータベースシステムに固有の問い合わせ言語の異種性、及び、ローカルデータベースシステムに固有の機能の異種性を吸収している。

次に、メタ JDBC の実現する機能を示すために、標準的な JDBC との差違について述べる。標準的な JDBC は、主に次のような機能を実現している。

- システムレベルでのデータ変換
- 多種多様なデータベースシステムを対象として、統一的なプログラムコードで利用することが可能

メタ JDBC は、上記に加えて、次のような機能を実現している。

- ローカルデータベースシステムをアクティブデータベースとして利用する機能
- 多種多様なローカルデータベースシステムを対象として、統一的な問い合わせ言語で利用する機能

このように、メタ JDBC は、標準的な JDBC では実現されていない機能を、JDBC よりも一段抽象度の高いクラスライブラリで実現する。これにより、既存の JDBC に一切変更を加えずに、機能の追加や問い合わせ言語の統一が可能となる。

また、メタ JDBC は次の3モジュールにより構成される。

- メタレベル問い合わせ処理系
- 問い合わせ変換系
- ファンクション・プロバイダ

メタレベル問い合わせ処理系は、メタレベルシステムから発行された部分問い合わせを、ローカルデータベースの基本操作セットである Primitive へ分割する。Primitive がローカルデータベースシステムの問い合わせ言語へ変換可能な場合は、問い合わせ変換系が、Primitive と等価なローカルデータベースシステムに固有な問い合わせを生成する。もし、Primitive と等価な問い合わせがローカルデータベースに存在しない場合は、ファンクション・プロバイダへ Primitive の実行を依頼する。ファンクション・プ

表 2: 実験に用いたローカルデータベースの概要

DBの目的	DBMS名	DBが格納するデータ
医療情報DB (Passive)	IBM DB2 V6.1	病院名, X座標, Y座標
災害情報DB (Active)	Borland InterBase V6.0	災害名, X座標, Y座標, 半径, 災害に対応可能な病院名
利用者の位置情報DB (Passive)	仮想的なGPS	X座標, Y座標

```

connect(jdbc:idx:interbase:
//localhost/c:/database/hazard.gdb)->HAZARD_DB;
connect(jdbc:idx:gps://localhost:4987)->GPS_DB;
setClient(localhost:9876);

open(HAZARD_DB, hazard)->HAZARD_REL;
open(GPS_DB, point)->POINT_REL;
projection(HAZARD_REL, place)->HAZARD_PLACE

event(update, HAZARD_REL){
if(contains(HAZARD_PLACE, POINT_REL)){
connect(jdbc:idx:d2:hospital)->HOSPITAL_DB;
open(HOSPITAL_REL, hplace)->HOSPITAL_PLACE;
join(HOSPITAL_PLACE, HAZARD_REL, name, #, hospital)->RES1;
projection(RES1, hplace)->RES2;
selection(RES2, nearest(POINT_REL))->RES3;
pushToClient(RES3);
}
}

event(update, POINT_REL){
if(contains(HAZARD_PLACE, POINT_REL)){
connect(jdbc:idx:d2:hospital)->HOSPITAL_DB;
open(HOSPITAL_REL, hplace)->HOSPITAL_PLACE;
join(HOSPITAL_PLACE, HAZARD_REL, name, #, hospital)->RES1;
projection(RES1, hplace)->RES2;
selection(RES2, nearest(POINT_REL))->RES3;
pushToClient(RES3);
}
}
    
```

表 3: ローカルデータベースのサンプル

医療情報データベース

hosname	x	y
A 病院	60	30
B 病院	40	40
C 病院	40	80
D 病院	10	34

災害情報データベース

hazname	x	y	radius	hosname
住宅火災	20	80	0.5	A 病院, C 病院, D 病院
交通事故	30	40	0.05	A 病院, B 病院, D 病院
光化学スモッグ	25	40	5	C 病院
核施設の事故	90	90	3	D 病院

ロバイダは、ローカルデータベースのAPIや外部モジュールを利用することによって、Primitiveを実行する。これにより、ローカルデータベースシステムの問い合わせ言語、及び、機能の異種性を吸収することが可能になる。

4 評価実験

ここでは、本方式の有効性を示すための、評価実験について述べる。

4.1 実験環境

本実験では、ユーザのルール定義に応じて多種多様なローカルデータベースを動的に連結する本方式の、モバイル・コンピューティング環境における有効性を示す。実験環境として、3ローカルデータベースシステムを構築した(表2, 3)。

医療情報データベースは、ある特定の地区の病院の位置情報を格納している、受動的なデータベースである。利用者の位置情報データベースは利用者の位置をリアルタイムに更新するシステムである。実験の利便性を図るため、実際のGPSレシーバではなく、仮想的なものをJava言語により実装した。これは、x軸100Km × y軸100Kmの合計10,000Km²を認識可能なシステムである。ユーザが移動するごとにGPSの保持する値が変化する。災害情報データベースは、ある特定の地区における災害情報、及び、その災害に対応できる病院名を格納する能動的なデー

図 8: システムへの入力

タベースである。これは、災害の発生した場所を、x座標、y座標、半径を用いて表現する。

4.2 実験方法

ここでは、上記の実験環境を対象とした、評価実験の方法について述べる。本方式のモバイルコンピューティング環境における有効性を示すために、次の2つの状況を設定した。

1. x=20, y=80, 半径0.5Kmの場所で住宅火災が発生する。
2. x=90, y=90, 半径3Kmの場所で核施設の事故が発生する。

災害が発生した場合、災害情報データベースの情報が更新され、メタレベルシステムへ通知される。また、ユーザが移動した場合、ユーザの現在位置がメタレベルシステムへ通知される。これにより、ローカルデータベースシステム抽象化機構、及び、アクティブ化機構の有効性を検証する。

以上のようなローカルデータベースシステム、及び、イベント発生状況を対象として、次のメタレベル・アクティブ・ルールを定義した。

- 利用者の位置情報データベースシステムが格納する、ユーザの現在位置が、災害情報データベースシステムが格納する最新の災害の発生場所に含まれている場合、利用者に通知せよ。そのとき、災害情報データベースシステムに格納されている1)緊急医療に対応可能な病院名と、医療情報データベース中に格納されている2)病院の位置情報、そして3)利用者の現在位置を結合し、「緊急医療に対応可能な、ユーザから最寄の病院の名前と位置」をユーザへ通知せよ。

また、このメタレベル・アクティブ・ルールの、実現システムへの入力を図8に示す。

このメタレベル・アクティブ・ルールは、災害が発生、もしくは、ユーザの移動をイベントとして処理する。いずれかのイベントが発生した場合、ユーザの現在位置が災害の発生地域に含まれるかどうかの判定を行う（コンディション判定）。この判定が真の値を返したとき、メタレベルシステムは、ユーザの現在位置と、医療情報、そして災害情報を結合する。これにより、「ユーザに最寄の災害に対応可能な病院名とその場所」をユーザへ通知する。また、この情報は、災害が発生した地区に存在するユーザにのみリアルタイムに提供され、その他のユーザへは通知されない。これにより、メタレベル・アクティブ・ルールの有効性を検証する。

4.3 結果と考察

上記の実験を行い、メタレベル・アクティブ・ルールが発火し、ユーザへ通知した情報を表4に示す。

表 4: 実験結果

状況 1

hname	x	y
A 病院	60	30

状況 2

hname	x	y
D 病院	10	34

上記の結果により、災害が起きる場所と、ユーザの現在位置によって、ユーザへ通知される情報が異なることが確認された。また、仮想 GPS レシーバと災害情報データベースが、双方ともアクティブデータベースとして動作することも確認された。また、ユーザが災害とは関連性のない場所を移動している場合は、ユーザへの通知は一切行われなかった。これにより、メタレベル・アクティブ・ルールが適切に管理、及び、適用されていることが確認された。以上により、モバイルコンピューティング環境における有効性を示すことができた。

5 結論

本稿では、既存の受動的データベース、及び、能動的データベースを対象として、統一的に利用可能とする、アクティブ型マルチデータベースシステムの実現方式を提案した。本方式の特徴は、既存データベースシステム群をアクティブデータベースとしてマルチデータベースシステムに接続する場合において、個々の既存データベースシステムは受動的であっても、マルチデータベースシステム全体としてはアクティブデータベースとして利用可能であるという点、及び、複数のデータベース間におけるイベント、コンディション、アクションを統一的に扱うことが可能であるという点にある。また、提案方式を実現した実験システム上でのルール適用処理実験の結果を示し、本方式の有効性を明らかにした。

今後は、メタレベル・アクティブルールの定式化、及び、

イベント情報の伝達方法の定式化を行い、それらを反映したシステム的设计、及び、実現を行う予定である。

参考文献

- [1] Allen, J.F.: "Maintaining Knowledge about TemporalIntervals", *Communications of the ACM*, No. 26, pp. 832-843(1983).
- [2] Bright, M.W., Hurson, A.R and Pakzad, S.: "A Taxonomy and Current Issues in Multidatabase Systems", *Computer*, Vol.25, No.3, pp.50-60 (1992).
- [3] Egenhofer, M.J.: "Spatial Relations: Models, Inferences, and their Future Application", *Proceedings of Advanced Database Symposium, Tokyo, Japan, December 2-4*, separate volume (1996).
- [4] 細川 宜秀, 石橋 直樹, 八代 夕紀子, 清木 康: "マルチデータベース環境における時間的・空間的関連性評価によるデータ結合方式", *情報処理学会論文誌*, Vol.40, No.SIG 8(TOD4), pp.95-111 (1999).
- [5] 細川 宜秀, 清木 康: "関数方計算によるマルチデータベースシステムの問い合わせ処理方式", *情報処理学会論文誌*, Vol39, No7, pp.2217-2230(1998).
- [6] 石川 博: "アクティブデータベース", *情報処理*, Vol.35, No.2, pp.120-129 (1994).
- [7] Arne Koschel, Ralf Kramer, Gunter von Bultzingsloewen, Thomas Bleibel, Petra Krumlinde, Sonja Schmuck, Christian Weinand: "Configurable Active Functionality for CORBA", *11th ECOOP'97 Workshop #7: CORBA: Implementation, Use, and Evaluation*. Jyvaskyla, Finland, June 10th (1997).
- [8] Kitagawa, T. and Kiyoki, Y.: "The mathematical model of meaning and its application to multidatabase systems", *Proc. 3rd IEEE Int. Workshop on Research Issues on Data Engineering: Interoperability in Multidatabase Systems*, pp.130-135 (1993).
- [9] Kiyoki, Y. and Kitagawa, T.: "A metadatabase system supporting interoperability in multidatabases", *Information Modeling and Knowledge Bases*, Vol.5, pp.287-298 (1993).
- [10] Seth White, Maydene Fisher, Rick Cattell, Graham Hamilton, Mark Hapner: "JDBC(TM) API Tutorial and Reference, Second Edition: Universal Data Access for the Java(TM) 2 Platform (Java Series)", Addison-Wesley (1999).
- [11] Microsoft: "Microsoft Odbc 3.0 Software Development Kit and Programmer's Reference", Microsoft Press (1997).