Vol. 43 No. SIG 5(TOD 14) 情報処理学会論文誌：データベース June 2002

非接触スマートカード上のユビキタスデータベース

倉光君郎† 坂村健†

ユビキタスコンピューティング時代のデータは，どこにでも存在する．我々は，小型コンピュータ，特にスマートカードや組込みシステム上の非常に小型のデータマネジメントシステム（DBMS）に焦点を当てる．このようなDBMSは，実世界のモノに直接付着することができ，モノを通して異なった組織間での情報の共有が可能となる．我々は，このアイディアを電子チケットシステム実証実験で実践した．本論文では，実験の経験をもとに非接触スマートカード上でユビキタスデータベースを構築する方法について論じる．

Ubiquitous Database on a Contactless Smartcard

Kimio Kuramitsu† and Ken Sakamura†

Ubiquitous database places data everywhere. We focus on contactless smartcards combined with a small processor and memory for data storage. Very small DBMS implemented on them can interact through queries on a wireless communication. Ubiquitous database associates object information with such a DBMS physically, and then allows different organizations to share information retrieved from real-world entities (i.e., goods, materials, or persons). We combined the theory with practice in the ticket digitalization project. Through the experience of our project, this paper will discuss how to manage ubiquitous database in the context of mobile commerce applications.

1. はじめに

すべての実世界のモノ（real-world object）は，性質や他モノとの関係，状態の変化などの様々な情報を持っている．現在のデータベースシステムは，このような情報を集め，中央集中的な方法で管理している．しかし，データの集中管理は優れた効率を提供するが，異なった組織の間で情報を共有する場合に困難が残る．電子商取引やデジタルライブラリなど，今日では新しいタイプの分散・協調型アプリケーションが重要となっており，そのような環境における鍵の1つは，データの遍在性（ubiquity）であろう．

まず，各々の実世界のモノが非常に小型のデータベース管理システム（DBMS）を持ち，自分自身に関する情報を蓄積している場合を想像してみよう．企業間を移動する商品は，同時にその電子化された情報も移動される．博物館収蔵品は，来館者やキュレータに対し，それぞれ違った情報のビューを提供できる．もし人間が自分の情報を持ち歩いていたら，DBMSは個人情報にアクセスしてくる情報システムに対し，適切なプライバシ保護を提供することができるだろう．データベースで強化されたモノ（database-augmented object）は，実世界コンテクストにおいて，情報の共有を可能にする．

問題は，本当に実世界オブジェクトに付着できる小さなDBMSを作ることができるか？という点である．幸い，マイクロエレクトロニクスの革新は，その実現性を高めている．特に，非接触スマートカード技術は，カードサイズの大きさの中に，プロセッサやストレージ（不揮発メモリ），無線通信装置を備えることを可能にし，電磁波によってデータベースクエリとプロセッサ用の電力を同時に送ることができる[2]．

本論文では，非接触スマートカード技術を議論のプラットホームにして，小型のDBMSを設計，構築する方法を議論する．特に，現在のハードウェア技術の水準では，カード上に完全なDBMS機能を実装できないため，認証されたホストとカードの上で分散的にDBMSを構築する方法を提案する．そして，我々が過去にスマートカードのアプリケーション研究で行った電子チケットシステム[7),8)]によるアクセスパターンを用いて性能を評価する．

本論文は，以下のとおり構成される．2章では，ユビキタスデータベースの概念について述べる．3章で

† 東京大学大学院情報学環
Interfaculty Initiative in Information Studies, The University of Tokyo

は，ユビキタスデータベースのデータ表現，アクセス制御モデル，クエリ言語を定義する．4 章では，実装モデルについて述べる．5 章では，アクセスパターンによる評価を行う．6 章では，本論文を締めくくる．

2. ユビキタスデータベースの概念

2.1 ユビキタスコンピューティング

コンピューティングは，メインフレームからパーソナルコンピュータへと，その機能を小型化と分散化させてきた．この歴史的な傾向から予想される近未来の 1 つが，**ユビキタスコンピューティング**（ubiquitous computing）である[12)~14)]．それは，実世界環境の中にコンピュータを遍在させ，それらが相互に協調動作することで，実世界環境自体を電脳強化することを目標としている．そこでは同様に，データやデータ管理システムも実環境の中に遍在させることが可能となるだろう．

ここで，データを実世界のモノに戻すことの意義を，まず我々のデジタルミュージアム[5)] の経験に基づいた博物館シナリオで考えてみたい．博物館の仕事は，膨大な数の収蔵品を保管するだけでなく，つねにその情報を管理することである．学芸員は，収蔵品を学術的に分析し，そのメモ☆を記録する．収蔵品は，展覧会や研究のため，外部との間で貸し借りを行う場合もある．たとえば，図録作成のため撮影業者に貸し出す場合，その撮影の指示や記録も情報として交換しなければならない．このような博物館の現場では，直観的な情報管理は，（もし可能ならば）収蔵品に直接情報を書き込むことである．実際に，収蔵品にはよくタグが付けられメモが書き込まれている．小さな DBMS は，このようなタグの代わりになるだろう．

ユビキタスデータベースは，タグとは違って，実世界のモノに情報をマークアップするだけではない．既存のデータベース技術や情報システムと簡単に統合化することを可能にする．**図 1** は，ユビキタスデータベースのコンセプトを示したものである．我々は，データを実世界のモノに戻すことによって，新しいデータベース技術やそのアプリケーションの可能性が広がるのではないかと期待している．

2.2 基盤プラットホーム

次の問題は，本当に実世界のモノに付属できる小さな DBMS を作ることができるか?という点である．幸い，マイクロエレクトロニクスの急激な発展により，

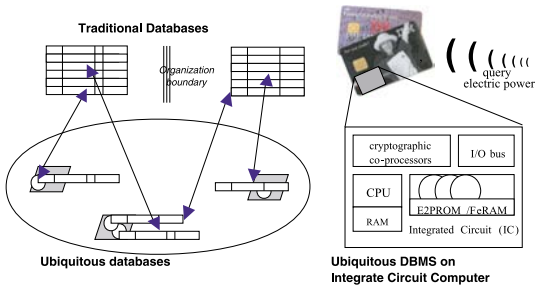

図 1 ユビキタスデータベースの概念
Fig. 1 Concept of ubiquitous database.

その可能性が現実のものとなりつつある．小型のコンピュータは，ネットワーク機能付きで，急速に身の回りの電気機器に組み込まれ始めている．近年，それらの上の組込みデータベース[9)] の研究や開発もさかんに行われ始めた．

本論文では，通常の実世界オブジェクトに付着できる可能性があるという点で，**非接触型スマートカード技術**に注目する．理由は，3 つに集約できる．1 つは，スマートカードは CPU や不揮発メモリ☆☆（ストレージ），通信システムを備えたワンチップマイクロコンピュータである点である．データマネジメントシステムを搭載できる最小限のハードウェアを備えている．第 2 は，スマートカードは物理的に耐タンパ性を有している点である．ファイヤウォールで保護されたデータベースシステム同様，すべてのアクセスを完全に制御し，必要に応じて暗号化することもできる．第 3 の理由は，非接触カードはバッテリを必要としない点である．プロセッサに必要な電力は，電磁波やマイクロ波によって，クエリと一緒に転送することができる．

これらの理由から，我々は非接触スマートカード技術を，ユビキタスデータベースの基盤プラットホームの最有力候補と考える．ただし，非接触スマートカードは，消費電力の制約のため，接触スマートカードと比べより限定されたリソースしか利用できない．本論文執筆時の標準的なカードでは，8 ビット CPU，プログラム領域 32 KB，データ領域 1〜4 KB 程度である．現在，より強力なチップ[11)] が開発中であるが，様々なモノに付着するためには製造コスト☆☆☆は無視できなく，安価なチップに対する需要は大きい．

2.3 関 連 研 究

電子タグ（アクティブタグ）は，バーコードシステ

☆ 博物館の情報管理とは，ある意味で収蔵物の分類や関係を分析し，そのスキーマを作成することである．最初からデータベーススキーマが決定できるとは限らない．

☆☆ 近年，不揮発メモリは，EEPROM に代わって，データ書き込み速度に優れた FeRAM が主流になりつつある．

☆☆☆ 現在，ほとんどの 32 ビットカードの単価は 1000 円以上で設定されている．

ムからユビキタスデータベースへの中間的なシステムといえる[4]．電子タグは，実世界のモノに付着され，RF 信号により電子的にモノを識別できるようにする．識別されたモノの情報は，その識別子をキーとして，リモートデータベースから取り出すことができる．ただし，つねにデータベースに接続可能な環境ばかりとは限らない．また組織を越えてデータを共有するのは難しい．

スマートカード上のデータ管理技術に目を向けると，まずファイルシステムの標準化が進んでいる．IEC/ISO 7816 では，UNIX のディレクトリとファイルによく似た階層構造を定義し，4 種類のファイル形式をサポートしている．さらに，ISO7816-4 では，APDUs（Application Protocol Data Units）と呼ばれるファイルシステムを操作する標準コマンドを規定している．また，SCFS（Smart Card File System）[3] では，スマートカードを UNIX のファイルシステムとしてマウントし，標準的なファイル API でアクセスできるようにしている．

スマートカード上のデータベースに関する先駆的な研究としては，PicoDBMS がある[10]．この研究では，接触型スマートカード技術をベースにして，1 MB を超えるストレージ上での複雑なクエリを効率良く処理するための RDBMS の実装が論じられている☆．スマートカード上のデータベース機能の必要性は，さらに標準化プロセスでも認識され，同じく ISO 7816-7 でも SCQL（Structured Card Query Language）で定義されている．SCQL ベースの RDMBS は，Carrasco [1] の JavaCard 上の RDMBS 開発が知られている．

現在のスマートカード上のデータ管理は，ファイルシステムやリレーショナルデータベースなど，既存の手法の拡大といえる．ユビキタスコンピューティング環境におけるデータ管理という視点から見ると，ファイルは情報を探すのが非効率であるし，リレーションは情報構造の不規則性に対して弱い．我々は，過去に行った電子チケットアプリケーションの経験[7),8)] を元に，組織を越えて遍在するデータの特性に着目したデータ管理手法の構築を目指している．

3. データベースモデル

ユビキタスデータベースでは，実世界のモノ自身が自分の情報を管理するため，情報のボリュームより多様性が重要になる．そこで，我々は自己記述的なオブジェクト表現である独自データモデルを用いる．本章では，我々がすでに報告したデータモデル[6] を，スマートカード上のデータベースに最適化した拡張モデルを定義する．

3.1 データ表現

ユビキタスデータベースは，複数のオブジェクトの集合からなる．ここで，データベースを D，オブジェクトを o で表せば，データベースは次のように定義できる．

定義 *1.* あるユビキタスデータベース D は，オブジェクトの集合 $\{o_1, o_2, .., o_n\}$ である．

ただし，D とその要素 o_i の関係（セマンティクス）は，`member_of` ではなく，`has_a` でとらえられるものとする．つまり，D は，ある実世界のモノと 1 対 1 で連想されているため，モノは o で表現される特性を持っていると解釈できる．

続いて，オブジェクトの内部構造の定義する．オブジェクトの内部は，複数のエレメントから構成される．ここで，エレメントを構成する部品として，まずラベルの全集合 $\mathbf{L}$，データタイプの全集合 $\mathbf{T}$，そしてデータ値の全集合 $\mathbf{V}$ を用意しておく．すると，エレメントは次のように定義できる．

定義 *2.* オブジェクトのエレメントは，ラベル，データタイプ，データ値からなるタプル $\langle l, t, v\rangle$ である．ここで，タプルの要素はそれぞれ $l \in \mathbf{L}$，$t \in \mathbf{T}$，$v \in \mathbf{V}$ である．

最後に，オブジェクトとエレメントの関係を定式化する．今，あるオブジェクト o_i 上に含まれるラベル，データタイプ，データ値の集合をそれぞれ，$L_i \subset \mathbf{L}$，$T_i \subset \mathbf{T}$，$V_i \subset \mathbf{V}$ とする．すると，オブジェクト o_i は写像として定義できる．

定義 *3.* オブジェクト o_i は，ラベルとデータタイプのペアからデータ値への写像，つまり $o_i : L_i \times T_i \rightarrow V_i$ である．

次は，あるオブジェクト o_a の定義例である．

o_a : (`product`, `string`) → `"aCD"`
o_a : (`price`, `currency-yen`) → 1200
o_a : (`price`, `currency-usd`) → 10

注意：ユビキタスデータは，論文 6) で論じたとおり，エレメントの名前をその概念を表すラベルと実世界タイプ（単位など）に分解する．また，ユビキタスデータベースでは，扱いを単純化するため，複合オブジェクトは導入しないものとする．同時に，ここではオブジェクト識別子も省略する．それは，複数の組織の間で識別子の唯一性を保証するためには，公開鍵認

☆ ただし，非接触型カードでは，電源供給の限界から不揮発メモリの搭載量は大幅に制限される．

証系と電子証明書[8]が必要となり，本論文の範囲を超えるためである．

3.2 アクセス制御モデル

ユビキタスデータベースの特徴は，不特定多数のユーザからのアクセスが想定される点である．そのとき，ユーザは各オブジェクトごとに異なった権限でアクセスする機会も少なくない．ユビキタスデータベースでは，そのためデータベース単位で認証するのではなく，オブジェクト単位で認証することが必要となる．我々は，不特定多数のユーザと各オブジェクトの間のアクセス権限の関係を簡単に対応付けるため，ロール（role）を導入する．

各オブジェクトは，その操作を行うユーザに対して，操作の目的ごとにいくつかのロールを設定できる（たとえば，creator や owner など）．各ロールにおいて，現在操作しているユーザがアクセスする権限を保有しているかどうか認証を行う．この仕組みは，パスワード認証を用いると，次のとおり定式化できる．まず，ロール名 r を有限集合 $\mathbf{R}$ の要素，パスワード p を有限集合 $\mathbf{P}$ の要素とする．ここで，ある認証系 σ は，関数 $\sigma : \mathbf{R} \times \mathbf{P} \to \{\mathtt{T}, \mathtt{F}\}$ になる．ただし，それぞれの認証系 σ に対して，$\sigma(r,p) \to \mathtt{T}$ を満たす (r,p) のペアは，1 つだけである．まとめると，オブジェクトの認証系は次のように定義することができる．

定義 4. 各オブジェクト o は，独自の認証系 σ を持つ．オブジェクト o が，$R \subset \mathbf{R}$ に対してアクセス制御が設定されているとは，それぞれの $r_i \in R$ に対して $\sigma(r_i, p_i) \to \mathtt{T}$ となる $p_i \in \mathbf{P}$ が o 上で定義されていることである．

ユーザは，ロール名とパスワードをペアにして，アクセス権限を獲得する．あるユーザが，(r,p) でアクセス権の獲得を行ったとき，あるオブジェクト o は $\sigma(r,p) \to \mathtt{T}$ により，r のアクセス権限が得られるが，別のオブジェクト o' は $\sigma'(r,p) \to \mathtt{F}$ により，権限が得られないこともある．このように，定義 4 は，同じユーザに対して，オブジェクト単位でアクセス制御を行うことを可能にする．

3.3 クエリ言語

ユビキタスデータは，ラベルとデータタイプが自己記述的に含まれているため，データ表現を指定するスキーマを必要としない．同様に，ユビキタスデータに対するクエリとその結果も，ラベルとデータタイプが含まれた自己記述的な構造となる．次は，select と update クエリの構文とその結果である．

select $\{(l_i, t_i)\}$ by $\{(r,p)\}$ where $\{(l_j, t_j)\ cmp\ v_j\}$
→ $\{\langle l_i, t_i, o_a(l_i, t_i)\rangle\ \}$ ... $\{\langle l_i, t_i, o_b(l_i, t_i)\rangle\ \}$, ..

update $\{\rangle l_i, t_i, v_i\langle\}$ by $\{(r,p)\}$ where $\{(l_j, t_j)\ cmp\ v_j\}$
→ $\{\langle l_i, t_i, o_a(l_i, t_i)\rangle\ \}$, $\{\langle l_i, t_i, o_b(l_i, t_i)\rangle\ \}$, ..

SQL のセマンティクスは，基本的に保存されている．特徴は，クエリごとに by 節によって，ロールの指定を行う．where 節の cmp は，比較演算子である．ユビキタスデータベースでは，構文を簡略化するため，$cmp = \{=, <>, <, >, <=, >=\}$ の 6 種類の演算子しかサポートしない．

注意：ユビキタスデータモデルでは，スキーマが存在しないため，ラベルやデータタイプのシンタックス的な一致によって処理される．そのため，実際に異組織間で運用するためには，同音異義語問題が発生しないように，ラベルやデータタイプのオントロジをあらかじめ構築しておく必要がある．ただし，オントロジ構築は，本論文の範囲を超えるので，我々はオントロジの共有を前提に議論を続ける．

4. 実装モデル

本章では，仮想的な非接触スマートカード $\mathcal{C}$ を導入し，特定のスマートカード製品から独立して，ユビキタスデータベースの実装モデルの議論を行う．カード $\mathcal{C}$ は，プログラム領域 32 KB，データ領域 4 KB，通信速度 106 kbps 程度と想定する．実際，我々は電子チケットアプリケーションの実験[7),8)]において，$\mathcal{C}$ と同等の性能のカードを想定し，本実装モデルのエミュレータを開発した．

4.1 システムアーキテクチャ

カード $\mathcal{C}$ 上では，明らかにすべてのデータベース機能を実現することは困難である．そのため，我々はラッパーアーキテクチャを採用し，いくつかの機能をホストコンピュータ上に外部化する．つまり，ホスト側にラッパーとしてプリプロセッサを置き，そこでクエリ言語は原始コマンドに分解され，カード上で実行可能になる．同様にプリプロセッサ上で，コマンドの実行結果はクエリの結果として再構成される．可変長のシンボル（ラベルや名前）はすべて固定長にバイトコンパイルされる．また，実行効率を最適化するため，キャッシュが統合される．

要点の 1 つは，カード上のコマンドセットの精錬である．理論的には，このラッパーアーキテクチャは，CPU なしの安価なデータキャリアチップにも適用できる．しかしクエリ分解は，ホスト–カード間のインタラクション数は増大し，これは特に ISO14443 のような非接触通信において深刻なパフォーマンス低下の

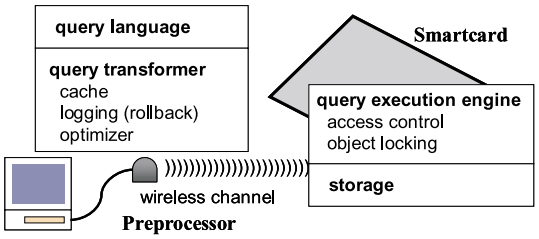

図 2 システムアーキテクチャ
Fig. 2 System architecture.

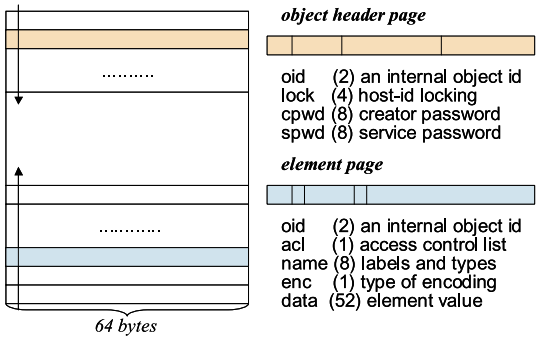

図 3 オブジェクトストレージのページフォーマット
Fig. 3 The formatting layout of object storages.

原因となる☆．つまり，カード上では，クエリ言語を効率良く処理できるコマンド設計が重要となる．

もう 1 つの要点は，データプライバシである．原理的には，ホスト–カード間は，スマートカードの認証機能によって相互に認証され，通信もすべて暗号化されている．しかしホスト上のプリプロセッサは，耐タンパ性が保証されないため，本来秘匿すべきデータがホスト側上で処理されるのは好ましくない．そこで，アクセス制御に関する機能は，外部化することはできない．**図 2** は，システムアーキテクチャと各機能分割の概要を示している．

4.2 ストレージモデル

まず，スマートカード上のストレージの構造から始めよう．カード $\mathcal{C}$ のストレージは，1 ページを 64 バイトとする．これは，EEPROM の代表的なページサイズに由来している．FeRAM 搭載カードでも，互換性のため，64 バイトページを採用することもある．

図 3 は，オブジェクトストレージのページフォーマットを示している．オブジェクトは，エレメントごとに分割される．それらを再統合するため，個々のオブジェクトには内部識別子 `oid`（> 0）が割り当てられる．エレメントページには，`oid` に続き，固定長化されたラベルとタイプ（`name` フィールド），アクセスコントロール（`acl`），データ値の int や char などのエンコーディング（`enc`），そしてデータ値そのもの（`data`）が記録される．これらのエレメント部とは別に，オブジェクトの制御情報を格納するヘッダページが導入される．そこでは，ロールを認証するパスワード（`cpwd` と `spwd`）や同期処理のためのロック（`lock`）が記録される．

我々が採用したストレージモデルは，索引（indexing）を採用しないため，フラット方式と呼ばれるものである．ただし，カード $\mathcal{C}$ のデータ領域がたかだか 4 KB 程度であり，Pucheral らの報告[10] と比較しても，索引の効果を期待できない．さらに，複雑なクエリ処理をすることなく，カード上のコマンドは 1 パススキャンで実行可能なように設計する．

表 1 原始コマンド
Table 1 Basic commands.

commands	description
ECHO	エコー検査
BYE	通信チャンネルの正常終了
SELECT	条件付き値の取り出し
OPEN	オブジェクトをロックする
READ	エレメントの値を読む
UPDATE	エレメントの値を更新する
INSPECT	オブジェクトのエレメント構造を取り出す
CLOSE	オブジェクトのロックを解除する
CREATE	新しいオブジェクト生成する
MOVE	オブジェクトを他の DB へ移動する
VERIFY	オブジェクトの証明書を検査する
DELETE	オブジェクトを消去する

4.3 原始コマンド

我々は，カード上の原始コマンドとして，オブジェクトの生成，更新，検証（証明書検査）を行うコマンドセットを設計した．**表 1** はそれらのまとめである．伝統的なカードコマンド（APDUs）との違いは，クエリを効率良く分解できるため，多重化と条件フィルタを加えた点である．

次は，エレメントに対する READ と UPDATE コマンドである．

READ `[oid]` n_1 .. n_i END.
→ $\{\langle \texttt{oid}, n_1, v_1\rangle..\langle \texttt{oid}, n_i, v_i\rangle\}$ END.
UPDATE `[oid]` n_1 v_1 .. n_i v_i END.
→ $\{\langle \texttt{oid}, n_1, o_{oid}(n_1)\rangle..\langle \texttt{oid}, n_i, o_{oid}(n_i)\rangle\}$ END.

マルチ引数をサポートすることで，複数のエレメントを 1 つのコマンドで処理できるようになっている．結果は，$\langle \texttt{oid}, n, v\rangle$ のシケーンスで受け取る（ここで，n は，エレメント部の name に相当し，v は写像 $o_{oid}(n)$，

☆ 我々の経験[7] では，デバイスのベンダにも依存するが，1 インタラクションのコストは 100 ms を超える．

つまりエレメント（oid と n）上に記録されていた値である．UPDATE では，更新する前の値を返す）．

oid は，カード $\mathcal{C}$ 上の内部 ID であるが，プリプロセッサ上でも共有する．各コマンドの最初の引数である [oid] は，ある特定のオブジェクトに対してフィルタをかけて実行するときに指定する．省略（oid = 0）すれば，すべてのオブジェクトがコマンド処理の対象となる．同様に，エレメント名（n）でもフィルタ処理を指定できる．

そしてより複雑なフィルタを可能にするのが，次の SELECT である．ただし，メモリストレージ上をワンパススキャンで実行するため，検索結果はエレメントの和（disjoint）として返される．

SELECT [oid] n_1 *CMP* v_1 .. n_i *CMP* v_i END.
→ {⟨oid, $n_1, o_{oid}(n_1)$⟩..⟨oid, $n_i, o_{oid}(n_i)$⟩} END.

4.4 アクセス制御

アクセス制御を行うため，4 種類のロール（creator，service，owner，others）を固定的に導入する．各エレメント上に記録される acl では，これらのロール対して，それぞれ read/write パーミッションが指定できるようになっている．また，creator と service のパスワードは，それぞれコントロール部の cpass と spass に記録される．owner は，データベースの所有権を表す特別なロールであり，パスワードはデータベース上で一元管理される．パスワード認証されなかった場合，デフォルトで others でアクセスすることになる．原始コマンドでは，引数の前でオプションとしてロールの指定が行える．ロールの指定が省略された場合は，前回のコマンドのロールとパスワードがそのまま継承される．

COMMAND BY (r, p) ..

注意：本論文で定義したロールは，5 章で述べる電子チケットアプリケーションに由来している．creator は，チケット発券者であるし，service は発券者以外のチケットサービスを提供する機関に該当する．明らかに，博物館やヘルスケアなど，アプリケーションによって，必要なロールは変わってくる．ロールの拡張性については将来の検討課題として残っている．

4.5 コンカレンシとロールバック制御

カード $\mathcal{C}$ は，物理的な限界のため，マルチユーザアクセスは存在しない．ただし，非接触通信は突然のセッション切断が発生しやすく，そのときのトランザクションの一貫性を保証する必要がある．

まず，各コマンド実行のアトミック性を保証する．ここで，非接触カードでは通信が切断されたとき，プロセッサへの電力供給が断たれることに留意しなければならない．したがって，受け取った命令と実行結果は，非接触メモリでバッファ処理することにする．UPDATE 実行のアトミック性が保証されれば，あらかじめ決定している複数のエレメントの更新はマルチ引数で実行することができる．

さらなる複雑なトランザクションに対しては，オブジェクト単位のロック機構を導入する．ロック（lock）には，ホスト–カード間でセッションが開くときに認証されるホストの（論理）ID が記録される．したがって，セッション切断後，通信が再開したとき，ホスト ID によってトランザクションの再開を判定することが可能になる．

また，ホスト上のプリプロセッサでは，リモートデータベースとデータの同期をとるため，2 相コミットスキームを構成できる．UPDATE は，更新前の値を返すため，これらをホスト側でログとして記録することで，ホスト側はロールバック処理を制御できるようになる．

5. 評　　価

本章では，我々が過去にスマートカードのアプリケーション研究で行った電子チケットシステム[7),8)] におけるアクセスパターンを用いて，実装モデルを評価する．

5.1 電子チケットアプリケーション

電子チケットシステムは，チケット発券から譲渡・流通，さらに入場チェックまで，すべて電子的に行うシステムである．我々は，複数の業者が自由に発券できるマルチチケットシステム☆として開発した．

チケットは，ユビキタスデータベース上のオブジェクトとして，約 20 エレメント程度で表現できる．これは，ストレージ上のサイズとして，大体 1 KB 強に相当する．つまり，カード $\mathcal{C}$ 上では，3 つほどのチケットが格納できることになる．**図 4** は，我々が実験で用いたチケットのサンプルを示している．

本論文では，このようなチケットに対する**表示**と**入場チェック**のクエリ処理に注目して議論をする☆☆．我々は，これらの処理はユビキタスデータベースの基

☆ 電子チケットシステムは，ユビキタスデータベースのコンテクストから見れば，スマートカード上の DBMS はカードの所有者に付着され，データオブジェクト（チケット）は所有者のサービスを受ける権利を表現することになる．

☆☆ チケットの発券や譲渡，その一貫性の保証に関しては，論文 8) で報告済みである．

o : (type, string) → "ticket"
o : (id, serial) → 100001
o : (title, string) → "今井美樹コンサート"
o : (orgnizer, string) → "キョードー"
o : (orgnizer, phone) → 03-xxxx-xxxx
o : (place, string) → "日本武道館"
o : (place, phone) → 03-yyyy-yyyy
o : (date, date) → 1999-10-07
o : (seat, string) → 指定 A00132
o : (enable, boolean) → *true*
o : (checkin, date) → *null*

図 4 サンプルチケット o
Fig. 4 A sample ticket o.

本的な操作を含んでいると考える．チケットの表示は，データベースから多くのエレメント読み出す操作が必要である．また入場チェックでは，複数のチケットの中から妥当なチケットを探して，入場済みを記録しなければならない．特に後者は，自動改札機の例が示すとおり，リアルタイム処理が求められる場合もある．

5.2 電子チケットの表示

スマートカード上のチケットは，ホスト側のコンピュータ上で表示される．チケットの所有者（ロール owner）は，チケットの一覧を次のクエリ（$Q1$）で得られる．

select $(id, serial), (title, string), (date, date)$
by $(owner, p1)$ where $(type, string) =$ "$ticket$"

続いて，あるチケットを選んで，その情報を取り出すクエリ（$Q2$）は次のとおりである．

select $(*, *)$ by $(owner, p1)$ where $(id, serial) = 100001$

5.3 電子チケットの入場チェック

入場チェック処理は，まずカード上から妥当なチケットを探し，そのうちの 1 つに入場記録を更新することである．次は，未使用のチケットを探すクエリ（$Q3$）である．

select $(id, serial)$ by $(service, p2)$ where
$(type, string) =$ "$ticket$" $and (enable, boolean) = true$

ここで各チケットは，入場サービスに対して，異なったパスワードを設定していることに注意しよう．つまり，オブジェクトの認証系 $\sigma(service, p2)$ → authorized となるチケットしかアクセスすることができない．入場サービスに対して妥当なチケットが複数見つかった場合，次のクエリ（$Q4$）によって，1 つ選んで入場記録を行う．

update $\langle enable, boolean, no\rangle, \langle checkin, date,$ 2001-12-20$\rangle$
by $(service, p2)$ where $(id, serial) = 100001$

表 2 コスト見積りの比較
Table 2 Results of cost estimations.

commands	*Q1*	*Q2*	*Q3*	*Q4*
フィルタと多重化	2	2	1	1
フィルタのみ	4	7	3	2
APDUs（参考）	12	7	15	2

5.4 解析と比較

クエリ $Q1$〜$Q4$ のアクセスパターンを使って，コマンドセットの設計のパフォーマンス解析を行う．まず，各コマンドごとの実行時間の違いは無視できるものとする．これはストレージのスキャンに比べ，コマンドに内在する比較演算は比較的に軽い処理と見なせるためである．したがって，コマンドの実行時間は，ストレージ上のオブジェクトの数に依存することになる．加えて，通信バッファのサイズも 256 バイトと仮定する．このサイズは，APDUs メッセージの大きさに由来するが，多くのカードが採用している．

ここで，カード上に 3 種類の同一ロールでアクセス可能なチケットが存在する場合に対して，クエリを実行するのに必要なインタラクション数の見積りを行う（各メッセージ長は，256 バイト以下になるように分割されている）．**表 2** は，クエリを実行するとき，コマンドセットの機能別に必要なインタラクション数をまとめたものである．たとえば，$Q1$ は，フィルタと多重化の機能を使えば，次の 2 つのコマンドに分解して実行することができる．

SELECT by (owner, $p1$) $\langle n_{type,string}, =,$ "$ticket$"$\rangle$ END.
READ 0 $n_{id,serial}, n_{title,string}, n_{date,date}$ END.

我々の設計したコマンドは，クエリ処理を効率良くラップできることが分かる．また，リアルタイム処理に関しては，実際のメモリアクセスやプロセッサ，通信システムの性能にも依存するが，最悪時間がインタラクション数の比例で見積もることができる．

6. 結　　論

データベースの研究分野では，正しく実世界の構造をとらえることができるデータモデルや，ある特定のモデルの上で効率良く大量のデータを扱う手法について論じられてきた．その歴史から見ると，ユビキタスデータベースは実世界に情報を戻し，実世界のコンテクストの中でデータ処理をすることは新しいデータアプリケーションの可能性を開拓するだろう．

我々は，電子チケットアプリケーションの実験を行った経験に基づいて，非接触スマートカード上の超小型の DBMS に注目してきた．そのうえで，実オブジェ

クトが持つ情報をよく表現し，複数の組織が共有・管理するためのいくつかの新しい手法を開拓した．オブジェクト単位のロールベースのアクセス制御やホスト側でトランザクションのロールバック制御を行う手法が含まれている．本論文では，それらの機能をホスト側とカード側に分割して，効率良く実現する方法を示した．

現在，我々のユビキタスデータベースはエミュレータとしての実装が行われただけである．これは，実際に実装可能な非接触スマートカードが入手できなかったためである．しかし，今年に入ってから，32 bit（CPU）/64 KB（FeRAM）搭載のカードも発表され始めた．今後は，ユビキタスデータベースのフル機能を搭載も視野に入れながら，DBMS のカード上で実装を目指していきたい．

謝辞 本研究を進めるにあたり様々なご意見，ご討論をいただいた今田庸介君と新堂克徳君（東京大学大学院理学系研究科）に深謝いたします．

参 考 文 献

1) Carrasco, L.C.: RDBMS's for Java cards? What a senseless idea! (1999). http://www.smartcard central.com/technical/articles/rdbms/rdbm[1].asp
2) Hansmann, U. (Ed.): *Smart Card Application Development Using Java*, Springer Verlag (1999).
3) Itoi, N., Honeyman, P. and Rees, J.: SCFS: A UNIX Filesystem for Smartcards, *Proc. of USENIX Workshop on Smartcard Technology* (1999).
4) 梶尾一郎，早坂 達：モノに情報をはりつける—RFID タグとその応用，情報処理，Vol.40, No.8, pp.846–850 (1999).
5) Koshizuka, N. and Sakamura, K.: Tokyo University Digital Museum, *Proc. 2000 Kyoto International Conference on Digital Libraries*, pp.179–186 (2000).
6) 倉光君郎，坂村 健：半構造オブジェクト：ユビキタスデータモデル，情報処理学会論文誌：データベース，Vol.42, No.SIG 15 (TOD12), pp.40–49 (2001).
7) Kuramitsu, K. and Sakamura, K.: Towards Ubiquitous Database in Mobile Commerce, *Proc. ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp.84–89 (2001).
8) Kuramitsu, K. and Sakamura, K.: Tamper-Resistant Network: an infrastructure for moving electronic tokens, *TOWARDS THE E-SOCIETY E-Commerce, E-Business, and E-Government*, pp.113–129, Kluwer Academic Publisher (2001).
9) Ortiz, S.: Industry Trends Embedded Databases Come out of Hiding, *IEEE COMPUTER*, Vol.33, No.3, pp.16–19 (2000).
10) Pucheral, P., Bouganim, L., Valduriez, P. and Bobineau, C.: PicoDBMS: Scaling down database techniques for the smartcard, *The VLDB Journal*, Vol.10, No.1, pp.120–132 (2001).
11) Sakamura, K. and Koshizuka, N.: The eTRON Wide-Area Distributed-System Architecture for E-Commerce, *IEEE Micro*, Vol.21, No.6 (2001).
12) Weiser, M.: The Computer for the 21st Century. *Scientific American*, pp.66–75 (September, 1991).
13) Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. *Comm. ACM*, Vol.36, No.7, pp.75–84 (1993).
14) Wellner, P. (Ed) : 電脳強化環境 — どこでもコンピュータの技術と展望，パーソナルメディア (1994).

(平成 13 年 12 月 20 日受付)
(平成 14 年 4 月 3 日採録)

(担当編集委員 宝珍 輝尚)

倉光 君郎（正会員）

1972 年生．1996 年東京大学工学部卒業（機械情報工学）．1998 年東京大学大学院理学系研究科修了（情報科学）．2000 年同大学院博士課程中退．現在，東京大学大学院情報学環助手．電子商取引，半構造データ，インターネットマーケッティング技術に興味を持つ．

坂村 健（正会員）

東京大学大学院情報学環教授．1984 年より TRON プロジェクトリーダーとして新しい概念に基づくコンピュータ体系の構築に精力を注ぐ．最近はプロジェクトの最終目的である超機能分散システムの研究を進めている．