

ユビキタスデバイスのクエリを用いた集成的管理手法

赤星 祐平[†] 木俣 豊^{††} 田中 克己[†]

ユビキタスコンピューティング環境では、多種多様なデバイス（ユビキタスデバイス）が至る所に存在し、それらが相互に接続されていることで、複数のデバイス間で機能を共有し利用するような新しい使い方も可能になる。しかし、このような環境では、デバイスが多数存在するため、既存の手法では、デバイスを把握したり、デバイスを管理・操作するという点において不都合な点が多く存在すると考えられる。そのため、大量のデバイスの管理や操作に対応可能な新しい手法が必要であると考えられる。

本論文では、ユビキタスコンピューティング環境において遍在する多数のユビキタスデバイスを集成的に管理利用するための仕組みとして、クエリを用いてユビキタスデバイスを管理する手法を提案する。デバイスをデータベースのテーブルと見立てることによりデータベースへのクエリ適用でデバイスの利用や機能共有などを実現するための、テーブル、タプル、クエリなどとユビキタスデバイスの関係について言及し、その仕組みについて説明する。

Set-oriented Control of Ubiquitous Devices by Queries

YUHEI AKAHOSHI,[†] YUTAKA KIDAWARA^{††}
and KATSUMI TANAKA[†]

In ubiquitous computing environment, there are many kind of devices (ubiquitous devices) around the world. These devices are connected each other through network, which enables us to use them flexibly such as sharing functions on the respective devices one another.

In this situation, because of the large number of devices, there is much difficulty in managing and operating those ubiquitous devices. Therefore, it is required to develop a new management method that can be applied for huge number of devices.

In this paper, we propose a device management method that can be applied for ubiquitous computing environment using queries. By considering a device as a table in a database, we can use devices and share functions on devices by applying queries to the respective tables. We clarify the relationship among tables, tuples, queries and ubiquitous devices and explain our mechanism.

1. はじめに

ユビキタスコンピューティング環境においては、PC や PDA、携帯電話といったデバイスのみならず、デジタルカメラやデジタル音楽プレイヤーなど、多種多様なデバイスがネットワークを介して接続し、相互に利用できるような状況が実現される。さらに、個々のデバイスが持つ機能を広く公開し、デバイス間で共有できるようになるものと考えられる。例えば、とある音楽プレイヤーに保存した音楽を、そのプレイヤー内蔵のスピーカーで聞くのではなく、近くで使える音質

のよいスピーカーを一時的に利用するといった利用形態が想定される。

しかし、ユビキタス環境では、非常に多くのデバイスが複数の機能を共有可能な状態で提供できるようになると考えられ、これまでのデバイスの操作や管理の手法では問題点が出てくると考えられる。数百のデバイスが利用可能な場合、それぞれのデバイスの状態を調べ、その状態に応じてデバイスを操作する場合に、個々のデバイスごとにチェックをして操作をするような仕組みであると、非常にコストが高くなる。さらに、もっと大量のデバイスを利用対象とする場合、そのコストはさらに膨大となる。そのため、大量なデバイスに対応できるような仕組みが必要になると考えられる。

そこで本論文では、ユビキタスコンピューティング環境において遍在する多数のユビキタスデバイスを集成的に管理利用するための仕組みとして、クエリを用

[†] 京都大学大学院情報学研究所
Graduate School of Informatics, Kyoto University

^{††} 独立行政法人 情報通信研究機構
National Institute of Information and Communications
Technology

いてユビキタスデバイスを管理する手法を提案する。デバイスをデータベースのテーブルと見立てることでクエリ適用でデバイスの利用や機能共有などを実現するための仕組みについて説明する。

2. 関連研究

本研究では、複数のデバイスを利用するための仕組みについて述べていくが、その関連研究について述べる。

まず、複数デバイスの機能を発見し共有するための仕組みとしては、uPnP⁶⁾ や JINI⁵⁾ に代表されるようなフレームワークや、Task Computing³⁾ といった仕組みがあげられる。これらの研究は、デバイス間の接続や、サービスの発見・共有といった仕組みをミドルウェアレベルで実現する仕組みと考えることができる。また、分散して存在する Web サービス利用するための仕組みとして ActiveXML¹⁾ のような仕組みも存在する。しかし、本研究のターゲットは、ミドルウェアの上位層において、どのようにデバイスを表現したり、機能共有やデバイス管理・操作をするかといった点であり、そのための一種のインタフェースを提供するための仕組みと考えられる。つまり、本研究においては、ミドルウェアレベルでのデバイス間接続や、機能（サービス）共有などの実現が前提となっている。そのような点で大きな違いがあると考えられる。

また、センサーネットを対象としてデータベース的に管理するための手法として、TinyDB²⁾⁴⁾ があげられる。この手法では、センサーネット上の各センサをデータベースにおける 1 タプルと見なすことにより、センサーネットからの情報収集やフィルタリングなどを TinySQL と呼ばれるクエリを用いて実現できるようにしている。この手法においては、センサーネットを対象としていることから、単一種のシンプルなデバイスからデータを収集するための仕組みに特化している。しかし、本研究では、センサーといったデバイスのみならず、PC や PDA といった多種多様なデバイスを対象として管理するための仕組みとしていることから、TinyDB とは異なっている。

3. ユビキタスデバイス

本節では、まず、本研究において対象とするデバイス：ユビキタスデバイスについて、その定義や扱い方について言及する。

3.1 ユビキタスデバイスの特徴

まず、ユビキタスデバイスの特徴について説明をする。

本研究で対象とするデバイス：ユビキタスデバイス

の特徴としては次のような点がある。

- ユーザに対する機能の提供
- デバイス間での機能共有
- デバイスの遍在性と多数性

ユーザに対する機能の提供とは、デバイスは必ず、ユーザの操作に応じて何らかの結果を提供するものであるということである。例えば、カメラは、ユーザがシャッターを押すことで写真を撮ることができる。その意味で、カメラはユーザに写真撮影機能を提供することができる。つまり、PC や PDA といった比較的汎用なデバイスのみならず、デジタルカメラやデジタルオーディオプレイヤー、さらに種々のセンサー等もユビキタスデバイスの範疇として含まれるうる。

デバイス間での機能共有は、デバイス中にある機能が、そのデバイスからのみならず、そこに接続する別個のデバイスからも利用可能ということである。従来のデバイスでは、あるデバイスにある機能はそのデバイス上からしか利用ができないものであった。例えば、デジタルカメラで写真を撮る場合には、そのデジタルカメラを操作してシャッターを切る必要があるが、それがユビキタスデバイスとなる場合には、必ずしもデジタルカメラを操作する必要はない。写真撮影の機能が広く公開・共有され、デジタルカメラに接続した任意のデバイスからでもデジタルカメラの操作を行い写真を撮影するようなことが実現される。

最後に、「ユビキタス」が示すとおり、ユビキタスデバイスはあらゆる場所に遍在し、そしてその数量が非常に膨大であるという点である。そのため、身近な数台の PC や PDA といったものだけを対象とするのではなく、ある建物内に存在するデバイス群、さらには、街中に存在するありとあらゆるデバイスまでも対象となりうる。

3.2 ユビキタスデバイスと機能

前述の通り、個々のユビキタスデバイスは、機能を持ち、ユーザの操作などによってそれが利用されるが、ここではその機能についての検討をする。

ユビキタスデバイスのとある機能の利用を考えると、一般に次のような関係が考えられる。

$$x = \text{function}(A, B, \dots)$$

ここで、*function* はデバイスで提供する機能、*A*, *B*, ... は、機能を利用するために必要な引数、*x* は機能利用の結果である。例えば、ビデオレコーダについて考えてみると、ビデオレコーダに存在するテレビ録画機能の関係を考えると、テレビ録画は、チャンネル、開始時刻、終了時刻を決めることで動画データが保存されることから、*function* = 「録画機能」、その引数

として、「チャンネル」「開始時刻」「終了時刻」を与えて、録画機能の結果 x は実際に撮られた動画データということになる。機能利用の結果は、その機能がいつ実行されたかによって変わってくるため、操作の結果を保存する必要がある場合には、その結果自体も保存する必要がある。

つまり、デバイスの操作の状態と結果を保持するためには、「デバイスの機能」「機能利用における引数」「機能利用の結果」を保持することが必要になる。

3.3 ユビキタスデバイスの扱い

本節では、ユビキタスデバイスの扱い方について説明する。

前節までで述べてきたとおり、ユビキタスデバイスはその数の多さや機能利用の柔軟度の高さなどから、従来のような管理や操作のメカニズムでは管理しきれないことが考えられる。そのため、本研究では、ユビキタスデバイスの管理のためにデータベース管理の技術を応用する。データベース管理技術を応用する理由は「クエリによる集成的操作」にある。データベース管理技術は、そもそも大量なデータを管理するために設計された仕組みであり、クエリを用いてデータベース中の要素を集成的に操作・管理することが可能である。そのため、この仕組みをユビキタスデバイスに適用することで、集成的に管理・操作ができると考えられる。

次に、データベース管理技術を応用するに当たって、デバイスとデータベースの要素の関係を設定する必要がある。そこで本研究では、「1つのデバイスごとに1つのテーブル」を設定する。つまり、10台のPCに対して10個のテーブルが生成され、それぞれ1対1で対応する。テーブルのフィールドには以下のような要素を含む。

- デバイスの機能
- デバイスの機能利用のための引数
- 時間情報

まず、デバイスの機能では、個々のデバイスに存在し、デバイス間で共有可能な機能の情報をそれぞれのフィールドに保持する。そのフィールドの値を参照することにより、その機能を利用した結果を知ることが可能になる。1つの機能に対して1つのフィールドを設定する。そのため、PCのように多数の機能を提供可能なデバイスの場合、機能の数に応じてフィールド数は増える。

その上で、機能利用のための引数では、実際に機能を利用する際に必要となる引数をそれぞれテーブルのフィールドに設定をする。例えば、録画機能を利用し

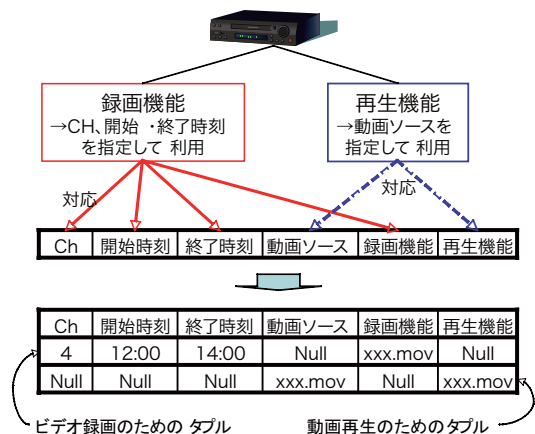


図 1 デバイスのテーブル化

てとある TV 番組を録画する場合には、録画すべきチャンネルや開始時刻、終了時刻を指定する必要がある。これらの情報は、機能利用の引数であり、それぞれにフィールドを設定する。

最後に、時間情報では、機能利用の開始時刻と終了時刻に関する情報を保持する。これにより、テーブルに対応する操作がいつ行われたか、履歴の管理が可能になる。この情報については、一部は機能利用の引数に含まれるものである。

これらの情報をテーブルの個々のフィールドに設定することで、テーブル中の1つのテーブルがデバイス上での1回の操作と対応することになる。これは、1つのテーブルで、機能利用のための引数と機能利用の結果がわかるため、このテーブルを保存しておくことによって、デバイスの操作履歴もテーブルの形で保持することが可能になる。

例えば、「チャンネルと開始・終了時刻を指定することで録画をする機能」と「何らかの動画ソースを指定することで再生する機能」を持つようなデバイスでは、テーブル表現は図1の通りになる。

このデバイスを表現するテーブルには、録画機能の引数となる「チャンネル」「開始時刻」「終了時刻」および再生機能の引数となる「動画ソース」の各フィールド、および、録画機能の結果と再生機能の結果を保持するための各フィールドが存在する。そして、1つめのテーブルでは、「12時から14時まで4チャンネルで放送された番組を録画し xxx.mov という動画データを得る」という操作の結果を表現し、2つめのテーブルでは「xxx.mov という動画データを再生する」という操作の結果が表現される。

最後に、ユビキタスデバイスとデータベース管理技術の対応関係をまとめると表1の通りになる。

表 1 データベース管理技術と本研究との対応関係

データベース管理技術	ユビキタスデバイス管理 (本研究)
テーブル	ユビキタスデバイス
フィールド	機能 or 機能利用のパラメータ
タプル	1回の機能 (デバイス) 利用
タプルの各フィールドの値	機能利用の結果 or パラメータ値
クエリ	デバイスに対する操作

4. クエリによる集成的操作

前節までで、ユビキタスデバイスをテーブルの形で表現する仕組みについて言及した。本節では、そこにクエリを適用してユビキタスデバイスを管理するための仕組みや、クエリの持つ意味について言及する。

4.1 タプルの挿入

まずは、タプルの挿入について言及する。

データベースにおいては、INSERT 句を利用することで、テーブルに対してタプルを挿入することができる。本手法では、各テーブルにおいて1つのタプルは、対応するデバイスでの1回の操作に相当する。そのため、タプルを挿入することは、新たにデバイスを操作することと同義である。

タプルの挿入においては、「機能利用の引数」に対して適切な値を設定したタプルを挿入する必要がある。これは、機能によっては引数を受け取って機能を利用して、初めて機能利用の結果が決まるものがあるためである。

このタプル挿入のクエリは、外的要因によって発行されて実行される場合と、自律的に発行される場合がある。前者の例としては、写真撮影が挙げられる。写真はユーザがシャッターを切ることにより写真撮影機能が駆動する。そのため、写真撮影のためのクエリが発行されるタイミングは、「ユーザがシャッターを切る」という外的要因に基づく。後者の例としては、センサによる周期的なサンプリングなどが挙げられる。1時間おきに温度センサで温度を検知するような場合、写真撮影の場合のようなユーザなどからの外的操作は与えられない。このような場合には、デバイス自体に内蔵するタイマーなどを利用して自律的に駆動し、クエリを発行するようにする必要がある。

図 2 でタプル挿入の流れについて例を用いて説明する。

この例では、とある番組を録画するためのタプル挿入の流れを示している。まず、録画機能を利用するために、利用のために必要となる引数「チャンネル (CH)」「開始時刻」「終了時刻」に対応するフィールドに正しい値を含むタプルを用意する。その際、他のフィー

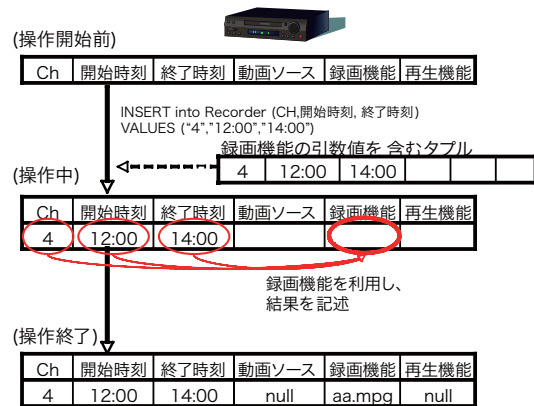


図 2 タプルの挿入

ルドの値は指定をしない (空値)。次にそのタプルをテーブルへ INSERT 句を用いて挿入する。タプル挿入が実行されると、与えられた引数の値を基に録画機能を利用する。録画機能の利用が終了すると、利用の結果として得られる動画データ (aa.mpg) の情報を、録画機能のフィールドに書き込む。書き込みの終了とともに一連の録画機能利用が終了し、その履歴がテーブルにタプルの形で保持される。

4.2 タプルの検索

データベースにおいては、SELECT 句を使うことでタプルの検索を行うことができる。本手法において、1つのタプルはデバイスにおける1回の操作 (履歴) を意味することから、検索のクエリによって、デバイスの利用状況や利用履歴をチェックすることが可能になる。

4.3 ビューの利用

テーブル操作の特徴の1つに、ビューの利用がある。ビューを利用することにより、テーブルから一部分を抽出したり、複数のテーブルから1つの仮想的なテーブルを生成することが可能である。その仕組みもユビキタスデバイスの管理において重要な役割を果たす。

本手法におけるビューには2種類が考えられる。

1つめは、単一のテーブルから生成するビューである。単一のテーブルからビューを生成する場合には、そのテーブルの一部が切り出されることになる。そのため、ビューを用いることで、デバイスの持つ機能ごとの状態を表現可能である。

2つめは、複数のテーブルから生成するビューである。これは、複数あるテーブルから、適切なフィールドを寄せ集めて仮想的なテーブルを生成することになる。ここで、テーブルが1つのデバイスと見なせることを考えると、ビューを使って生成される仮想的な

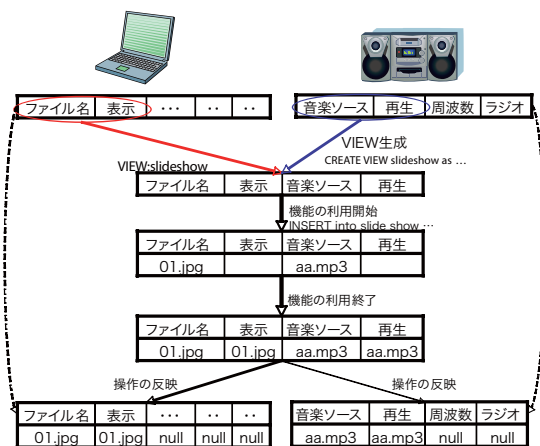


図 3 ビューの利用

テーブルは、一種の仮想デバイスと見ることができる。つまり、デバイス間で機能共有が可能な状況において、適宜必要な機能を寄せ集めて同時利用する時に、その機能の集合を仮想的なデバイスと見ることによって、あたかも1つのデバイスと同じように操作する場合に、このビューを利用することが可能である。

複数デバイスから機能共有による仮想的な複合デバイスを構成して利用するためにビューを形成する場合、そのビューに対してタプルを挿入することにより、その複合デバイスの操作が可能になる。しかし、複合デバイスを操作することは、実際には個々のデバイスをそれぞれ動かしていることと同義であり、デバイスの操作履歴の情報は、個々のデバイスに対応するテーブルにタプルの形で保持されているため、一貫性を保つために、ビューに対してタプルが挿入された場合には、その情報を、ビューに含まれるフィールドが本来含まれるテーブルに反映する必要がある。

図3で機能共有におけるビューの利用について説明をする。

この例では、とあるPCで写真のスライドショーをしながら、別のコンポで音楽をBGMとして流す場合を考える。このとき、必要となる機能は、PCにある「表示機能」とコンポにある「音楽再生機能」である。そこで、これらに関するフィールドを寄せ集めたビュー (Slideshow) を CREATE VIEW 句により生成する。つまり、このビューのフィールドは、PCから利用する「表示機能」と表示機能利用のための引数としての「ファイル名」、コンポから利用する「再生機能」と再生機能利用のための引数としての「音楽ソース」から構成される。ビューが生成されると、そこにタプルを挿入することでPCでの表示機能とコンポの音

楽再生機能を同時利用することが可能になる。例えば、01.jpg という写真を表示しながら aa.mp3 という音楽をBGMで再生するのであれば、「ファイル名=01.jpg, 音楽ソース=aa.mp3」というタプルを INSERT 句により slideshow に対して挿入する。挿入後、操作を実行し、表示機能と再生機能のフィールドに実行後の結果として 01.jpg と aa.mp3 をそれぞれのフィールドの値として保持することで操作が終了する。操作終了後、PCとコンポのそれぞれで操作された情報に対応するテーブルに反映して一貫性を保持するために、それぞれのテーブルにタプルを挿入する。

4.4 タプルの更新・削除

通常のデータベースにおいては、タプルの更新や削除といったクエリが用いられる。しかし、本研究において、個々のタプルは、デバイスの操作の履歴と見ることができる。そのため、ユーザの操作によって直接的にタプルの削除や更新を実行する必要性は非常に低い。むしろ、一度テーブルに挿入されたタプルは削除せずに保持しておくことで、操作履歴に応じたデバイスの管理等に利用できることから、本手法においては、タプルの削除を実行することは原則としてない。ただしタプルの更新については、機能を利用するために INSERT 句によりタプルを挿入するが、その後に機能を利用し結果に対応するフィールドに書き込む際に実行することになる。

4.5 トリガ

最後に、トリガについて言及する。

データベースにおいては、テーブルにトリガを設定することで、クエリの連続発行が可能になる。これを本研究の仕組みに適用すると、「デバイスの連続的操作」の実現が可能になる。テーブルにトリガを設定することでクエリが連続的に発行できることから、デバイスの状況に応じてクエリを発行し、他のテーブルにタプルを挿入することで機能を利用したり、ビューを生成することで機能共有を実現するといったことが可能になるためである。

5. 考 察

前節までで、ユビキタスデバイスをクエリにより集合的に扱うための手法について述べてきた。本節ではそれに関する考察を行う。

まず、本研究の大きな特徴として、1つのデバイスを1つのテーブルとして表現し扱うことを行う。そして、テーブルのフィールドには、機能や機能利用のための引数などの情報を含み、操作履歴はタプルの形でテーブルに蓄積されていく。このような仕組みにする

ことで次のような点が利点としてあげられる。

- デバイス状態が一覧可能
- デバイスの機能とコンテンツ（データ）の同等な操作環境の提供
- 操作履歴によるデバイス利用の実現

まず、デバイスに存在する機能や機能利用のために必要な情報（引数）、デバイスの利用履歴が1つのテーブルにまとめて保持されていることになるため、テーブルを参照することでデバイスの状態が一覧することが可能である。また、クエリによりテーブルにアクセスすることにより、かなりの部分のデバイス管理・利用を表現できると考えられる。

そして、デバイスの機能とコンテンツ（データ）を同等に扱うことが可能なことも特徴と考えられる。これは、テーブル中にフィールドとして現れる「デバイスの機能」とそこで利用するための「コンテンツ（データ）」がタブルの値として現れ、それらをクエリという同一の手法によって操作できることである。従来の仕組みでは、デバイスの管理やデバイス上でのコンテンツ利用といった仕組みは別個のものにされることが多かった。しかし、本手法では、機能やそこで利用するコンテンツはテーブル上に同等に置かれ、その種類に関係なくクエリによってアクセスすることが可能になる。それにより、別個にデバイス管理とコンテンツ管理をするために必要になるコストよりも低いコストにて大量のデバイスに対するデバイス管理やコンテンツ利用制御が可能になると考えられる。このようにデバイスやそこにある機能、その上で利用するコンテンツ（データ）を統一したフレームワークで扱うことができるという点は、本手法における利点と考えることができる。

また、操作履歴がタブルの形で保持されることで、それらの情報を利用した操作が可能である。これにより、「最近にとある書類を編集したPCから、その文書をこれから使うPCに転送する」といったデバイス操作も、テーブルを検索して操作履歴を参照し、その上で条件に合うテーブルに対して新たにタブルを挿入することで実現される。

一方で、解決すべき問題点も存在する。

- デバイス名とテーブル名の対応関係の解決方法
- 位置情報の扱い

まず、デバイス名とテーブル名の対応関係を解決するための方法がある。既存のSQLのようなクエリの仕組み・仕様では、「条件を指定した上でのテーブル名の検索」ということができない。対処の方法としては、「SQLの拡張」と「テーブル名とデバイス名の関係を

解決するためのテーブルの用意」の2点が考えられる。SQLの拡張については、現在SELECT句による抽出対象は、テーブルの一部、もしくは、テーブル中のタブルから計算可能な値（平均値、タブルの数など）を利用したものであり、例えば、特定の値を持つ「フィールド名」や「テーブル名」を検索することはできない。そのため、その点を拡張し、タブルの状態などからフィールド名を検索したり、特定のフィールドを含むテーブル名を検索できるようにすることが考えられる。この手法の利点として、テーブル名やフィールド名が検索できるようになることで、クエリの記述能力が向上し、単純な機能利用以上により高度な機能利用が実現されると考えられる。一方で、「テーブル名とデバイス名を解決するテーブルの用意」という手法も考えられる。このテーブルにデバイスの名称（ID）やテーブル名、そこに存在するフィールド名称などまで保持することにより、デバイス名とテーブル名の関係の解決は可能である。非常にシンプルな解法ではあるが、管理対象のデバイスが膨大になると、デバイス名とテーブル名の解決のためのテーブルと個々のデバイス情報を保持するテーブルを結合して利用するような場合に、膨大なコストがかかる可能性が考えられる。

また、現在は位置情報についての扱いをまったく行っていない。ユビキタスコンピューティングにおいては、時間情報のみならず位置情報も非常に重要になると考えられる。時間情報は各テーブルにフィールドの形で持たせておくことで操作の開始・終了時刻が保持できるが、位置情報については、操作をしながら移動をする場合にその情報をどのように保持するかといった点で、時間情報ほど簡単ではないと考えられ、今後検討する必要があると考えている。

6. ま と め

本論文では、ユビキタスコンピューティング環境下で利用可能な多数のデバイス：ユビキタスデバイスを対象として、それらを集散的に管理・利用するための仕組みについて提案をした。データベース管理技術を応用し、個々のデバイスをテーブルに、テーブルのフィールドにはデバイスの持つ機能や機能利用のために必要な情報（引数）を割り当てることにより、テーブルに保持するタブルでデバイスの操作や、操作履歴を管理することが可能になる。そして、テーブルに対してクエリを適用することで、デバイスの操作や機能共有の実現、履歴の検索といったことが実現される。本手法の特徴としては、テーブル参照によるデバイス状態の一覧性や、デバイス・機能とコンテンツのクエリによ

る同等な操作環境の提供といった点がある。

現在、これらの仕組みを利用したシミュレーションのためのシステムを構築中であり、今後それを用いた本手法の有効性の検証を行う予定である。

謝 辞

本研究の一部は、文部科学省 21 世紀 COE プログラム「知識社会基盤構築のための情報学拠点形成」(リーダー：田中克己)、および、文部科学省科学研究費補助金特定領域研究「情報爆発時代に向けた新しい IT 基盤技術研究」計画研究「情報爆発時代に対応するコンテンツ融合と操作環境融合に関する研究」(研究代表者：田中克己、A-01-00-02、課題番号 18049041)、文部科学省科学研究費補助金特定領域研究「情報爆発時代に向けた新しい IT 基盤技術研究」公募研究「ユビキタス情報社会での情報爆発に適した検索・分類・統合化手法」(研究代表者：木俣豊、A01-24) によります。ここに記して謝意を表します。

参 考 文 献

- 1) S.Abiteboul, O.Benjelloun, I.Manolescu, T.Milo, R.Weber, “Active XML: Peer-to-Peer Data and Web Services Integration”, Proc. of the 28th Int’l Conf. on Very Large Data Bases (VLDB2002), pp.1087–1090, August 2002
- 2) S.R.Madden, M.J.Franklin, J.M.Hellerstein, W.Hong, “An Acquisitional Query Processing System for Sensor Networks”, ACM Transaction on Database System (TODS) Vol.30, Issue 1, pp.122–173, March 2005
- 3) R.Masuoka, B.Parsia, Y.Oabrou, “Task Computing – The Semantic Web Meets Pervasive Computing”, Proc. of The 2nd Int’l Semantic Web Conf(ISWC2003), pp.866-881, 2003
- 4) TinyDB <http://telegraph.cs.berkeley.edu/tinydb/>
- 5) JINI <http://www.jini.org>
- 6) uPnP Consortium <http://www.upnp.org>