

Device as Table: 問合せ操作によるユビキタスデバイスの管理

赤星 祐平[†] 木俣 豊^{††} 田中 克己[†]

[†] 京都大学大学院情報学研究科 〒606-8501 京都市左京区吉田本町

^{††} 独立行政法人情報通信研究機構 〒619-0237 京都府相楽郡精華町光台 3-5

E-mail: [†]{akahoshi,tanaka}@dl.kuis.kyoto-u.ac.jp, ^{††}kidawara@nict.go.jp

あらまし ユビキタスコンピューティング環境では、実世界中のあらゆる場所に無数のデバイス：ユビキタスデバイスが存在し、利用可能になると想定されている。しかし、その数の膨大さなどから、従来のデバイス管理の手法ではユビキタスデバイスを管理したり利用したりするには非常な困難が伴うと考えられる。そこで本論文では、ユビキタスデバイスをデータベースのテーブルとみなすことにより、クエリによる問合せ操作を用いてユビキタスデバイスを管理する手法について提案する。個々のユビキタスデバイスの持つ機能や状態などをデータベースにおけるテーブルとして表現することにより、クエリをテーブルに適用してデバイスを管理する事が可能になる。また、クエリによって複数の対象の集成的操作が可能のため、複数のデバイスについても一括した管理・操作が可能になる。

キーワード ユビキタスコンピューティング, 問い合わせ操作

Device as Table: Ubiquitous Device Management by Query Operation

Yuhei AKAHOSHI[†], Yutaka KIDAWARA^{††}, and Katsumi TANAKA[†]

[†] Graduate School of Informatics, Kyoto University

Yoshida-Honmachi, Sakyo-ku, Kyoto, 606-8501 Japan

^{††} National Institute of Information and Communications Technology

3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan

E-mail: [†]{akahoshi,tanaka}@dl.kuis.kyoto-u.ac.jp, ^{††}kidawara@nict.go.jp

Abstract In ubiquitous computing environment, it is assumed that there are huge number of devices: “Ubiquitous Device”, available for in the real world. However, because of their number, it is difficult for us to manage them with conventional device management mechanisms. In this paper, we propose a ubiquitous device management mechanism using query operation. In this mechanism, we consider ubiquitous devices in the world as tables in a database. Therefore, we can operate these devices by applying queries to the correspondent tables. Moreover, query language is designed to operate multiple elements at once, we can manage and operate multiple devices at once by single query operation.

Key words Ubiquitous Computing, Query Operation

1. ま え が き

現在、身の回りにはPCや携帯電話、さらにはデジタルカメラ、デジタルオーディオプレイヤーなど多種多様なデジタルデバイスが存在している。これらを利用するには、デバイス自体を直接操作したり、あらかじめ対象を限定した上でシステムを構築し同時に利用するなどのことが行われている。

近い将来には、これらのデジタルデバイスは、数がさらに増大し、ネットワークを介して相互に接続され、自由に組み合わせたりしながら利用可能になると想定されている。数千や数万といった単位で街中にデバイスが存在するようになり、そ

れらを管理、利用しようとする場合、従来の直接操作といった手法では管理に限界がある。また、ユビキタスコンピューティング環境下では、デバイスの出入りも激しくなると考えられ、従来のように対象を限定したシステム構築では、デバイスが状況に応じて制御対象になったりはずれたりするときの対応が難しいと考えられる。このようなことから、ユビキタスコンピューティング環境下の膨大な数のデバイス管理のための新しい手法が必要になると考えられる。

一方で、従来からあるデータベース管理技術について考えてみると、データベースには多数のデータが保持され、それに対してクエリを適用することによりデータの管理・操作を実現す

るものである。つまり、データベース管理技術は、元来多数のものを対象にしてその操作・管理などを行うために設計されているものであるといえる。このようなことから、この手法を実世界中に膨大な数存在するデバイスを管理利用するための仕組みに応用することで、効率的なデバイス利用環境の構築が実現されると考えられる。

そこで、本論文では、実世界中に存在するユビキタスデバイスを管理するための手法の提案を行う。1つのユビキタスデバイスを、データベース上の1つのテーブルと見なし、テーブルに対してクエリ操作を行うことで、デバイスの管理や利用を実現する仕組みを提案する。

以下、2.節で本研究の概要について述べ、3.節でクエリ操作とデバイス管理・操作の関連について述べ、4.節で関連研究に関して言及する。

2. 本研究の概要

まず本節では、本研究の基本的な考え方から提案する手法の概要について述べる。

2.1 前提とする事項

まず、本研究で対象とするデバイスについて、いくつか想定する前提を述べる。

対象とするデバイスには以下のような条件・前提を考える。

- ユーザに対して機能を提供する
- 機能を広く共有可能

まず、デバイスは何らかの機能をユーザに提供するものと考ええる。たとえば、携帯電話を考えると、携帯電話の利用者はこれを利用して、他の誰かに電話をかけることができる。また、メールを送信することも可能である。つまり、携帯電話はユーザに対して「電話をかける」機能や「メールを送付する」機能を提供してくれるデバイスであると考えられる。本研究で管理や操作の対象となるデバイスは、1つ以上の機能を必ずユーザらに提供するものである。

また、それぞれのデバイスの持つ機能は広く共有が可能であることを前提とする。どのデバイスにどのような機能が存在するかという情報が広く提供されており、かつ、それらを自由に利用するための仕組みがすでに用意されているものとする。それにより、ユーザは利用したい機能を持つデバイスを検索したり、そこから自由に複数のデバイスから適切な機能をピックアップして同時利用するなどが可能になる。

2.2 基本の考え方

まず、本研究の基本的な考え方について言及する。

現在、身の回りに多くのデジタルデバイスが普及してきている。しかし、近い将来には、ユビキタスコンピューティング環境が実現すると、さらに膨大な量のデジタルデバイスが身の回りのあらゆる場所に存在するようになり、さらにそれらがネットワークを介して相互接続するような環境が実現されると考えられる。多種多様なデバイスが存在するようになることから、しかし、数千、数万という膨大な数のデバイスを対象にして管理し利用しようとする場合、これまでのデバイスの管理の手法では限界があると考えられる。

例えば、次のような例を考える。

とある街中に多数の温度や大気センサーなどが設置され、常時気温やCO₂濃度などの測定を行っている。交通標識や信号機等についても適宜操作することによって調整が可能である。さらに、走行中の自動車からも速度や燃費などの情報を収集することが可能であり、自動車に搭載されたディスプレイや歩行中の人の持つデバイスのディスプレイに情報提供をすることが可能である。

この環境で街中の交通制御システムを構築する。大気センサーでCO₂の濃度が一定以上なった地域が見つかったら、そのエリアの近くにいるユーザのデバイスに通知を行う。その上で、交通標識や信号の制御をしてCO₂濃度が下がるように調整を行う。CO₂濃度が正常になると制御を元に戻し、ユーザへの通知も終了する。

このような仕組みでは、対象となるデバイス数が無数に存在する上、自動車や歩行者は常に対象となるエリアから出入りすることにより一定しない。このような場合、これまでの仕組みで構築を行う場合には、非常に困難になる。そのため、多数のデバイスを対象としてそれらを管理し利用するためには新しい仕組みが必要であると考えられる。

ここで、既存のデータベース技術について考えてみる。データベースは、大量のデータをタプルの形でテーブルに保存、それらに対してクエリ(SQL)を適用することでデータの追加削除、検索、アクセス制御などの管理を実現している。クエリを用いることで、たくさんのデータを集散的に扱うことが可能である。このように、データベース技術は、大量の対象を集散的に管理利用するために設計されたものであると考えられる。

このようなことから、データベース管理の手法をユビキタスデバイスの管理利用の仕組みとして取り入れることで、効率的にデバイスの管理ができるものと考えられる。管理対象のデバイスをデータベースの要素ように見立てることにより、クエリ操作を用いた集散的な操作・管理が実現されることになる。

とあるPCとデジタルカメラを連携して使用する場合を用いて違いを説明する。図1は、従来の手法について示したもので

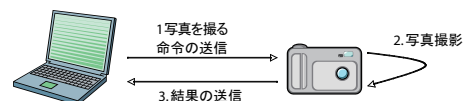


図1 従来手法：デバイスの直接操作

ある。PCからデジタルカメラの機能を利用して写真を撮りPCで見る場合、まず、APIなどを利用して直接デジタルカメラに対して写真を撮るための命令を送信する。撮影命令を受け取るとデジタルカメラで写真を撮影し、その結果を返値のような形でPCに渡し、結果を表示する。Webサービスの仕組みを利用して、別個のデバイスに存在する機能を利用するような仕組みについても、この手法と同様に、SOAPなどを用いてメッセージを送付し、その返り値としてサービスを利用するような仕組み

みとなっている。

一方で本研究で提案する手法では図2のようになる。提案す

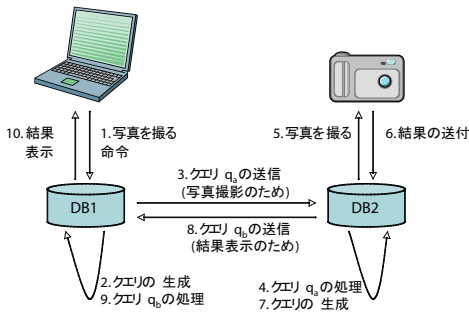


図2 提案手法：クエリによるデバイス操作

る手法では、すべてのデバイスがデータベースのようなものを持ち、それに対するクエリ操作によってデバイスを操作しようとする。そのため、PCからカメラを利用する場合、まず、PCから発せられる写真撮影の命令はPCに対応づけられたデータベース(DB1)に送られる。そこで、写真撮影のために必要なクエリ q_a を生成し、それをカメラに対応づけられたデータベース(DB2)に送る。 q_a を受け取ったDB2はそれを処理することで、実際に写真撮影が実行される。撮影された結果はDB2に反映され、それをPCで表示するために、DB2でクエリ q_b を生成する。そして、 q_b をDB1に送信し処理することで、結果をPC上で表示することになる。

データベースを介するようにすることで、クエリ生成やそのデータベースへの適用などで従来手法よりも多くの処理を必要とすることになる。しかし、多数のデバイスを同時に対象にして利用するような場合では、クエリを用いた操作の場合は、1つのクエリで多数の対象を同時に操作することも可能になる。また、対象となるデバイスが増減するような場合にも、対応するデータベースの追加削除する程度で既存のデバイスと同等に扱うことが可能になることから、このような仕組みをとることでユビキタスコンピューティング環境下のデバイス管理に非常に有用であると考えられる。

2.3 デバイスとデータベースの関係

本節では、本研究で対象となるデバイスの、データベース管理技術との対応関係について述べる。

本研究では、1つのデバイスを1つのテーブルに対応づける。この対応付けにより、管理対象となるデバイスはすべて個々のテーブルとして1つのデータベース中に保持することが可能になる。

デバイスを操作する場合、基本となる行為はデバイスの持つ機能を利用することにあると考えられる。そのため、1つのデバイスを1つのテーブルとして表現した場合、個々のテーブルを構成するフィールドは、対応するデバイスの持つ機能としている。それらの機能に対応づけられたフィールドに加えて、その機能利用の時間情報を保持するためのフィールドとして Transaction Time、と Valid Time という2つの時間情報を保持するためものを用意する。

このようにして用意されたテーブルに対して挿入するタプル

には、機能に対応するフィールドには「機能を利用するために必要な引数」、Transaction Time のフィールドには「機能の利用開始時刻」、Valid Time のフィールドには「機能の利用時間」をそれぞれ値として持つものになる。有効なタプルがテーブルに保持されている状態で、デバイスの機能が利用されている状況を表現する。

図3にて例を示す。携帯電話について、通話、テレビ電話、

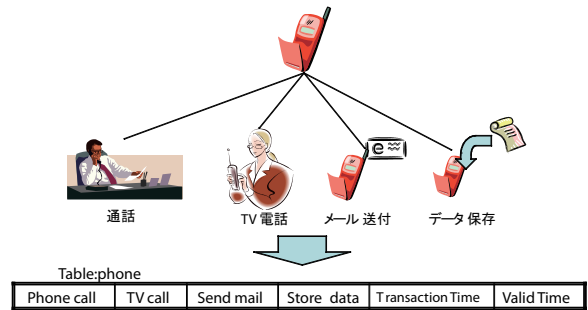


図3 デバイスとテーブルの関係

メール送付、データ保存の4つの機能を持つ場合を考える。これを管理対象とするとき、対応するテーブル phone は、それぞれの機能に対応するフィールドとそれに加えて Transaction Time と Valid Time のフィールドを加えた6つのフィールドから構成されることになる。

デバイスをデータベースにおけるテーブルで表現し、そこにタプルを保持することにより、各テーブルはデバイスの利用状態を表現することになる。デバイスに存在するそれぞれの利用状態がテーブルを見ると判明する。そして、クエリを用いてテーブルにタプルを挿入したりすることにより、テーブルの状態を変化させ、それを結果として対応するデバイスの状態として反映される。この意味から、テーブルに保持されたタプルは、機能利用のためにデバイスで実行されるプロセスに相当するものと見ることができる。

このようにして、管理対象となるデバイスをテーブルに見なすことにより、DBにクエリを適用してデータを管理するように、クエリを利用したデバイス管理が実現するようになる。クエリ操作とデバイス操作の対応関係については次節で述べる。

3. クエリ操作とデバイス管理の関連

本節では、どのようなクエリ操作を、実際のデバイス管理に対応づけるか、その関連について言及する。

3.1 デバイスの機能の利用

まずは、個々のデバイスにある機能を利用する場合について考える。

本研究においては、デバイスに対応したテーブルが存在し、そのフィールドはデバイスに存在する利用可能な機能となっている。そこで、対応したフィールドに機能を利用するための引数を値として持つタプルを挿入することでデバイスの機能利用を示すことが可能である。利用しない機能のフィールドには NULL 値を設定する。また、Transaction Time と Valid Time のフィールドの値については、機能利用の開始時刻を

Transaction Time の値に、機能利用の有効時間の値を Valid Time の値として設定する。Valid Time については、機能の利用時間を利用開始時に判明しない場合は不定値を設定する。

タブルを挿入してデバイスの機能を開始した後、機能の利用の停止をする必要が出てくる。この機能の利用停止については、挿入されたタブルの削除によって行うのではなく、タブルに存在する Transaction Time と Valid Time の値を用いて制御をする。あらかじめタブル挿入時に Valid Time の値として有限の値が設定されている場合には、機能利用の停止時刻が Transaction Time と Valid Time の値から判明するので、その時刻になり次第、自動的に機能の利用を停止させる。Valid Time の値として不定値が設定されている場合、機能の利用終了時刻が判明した時点で Valid Time の値に有限値を設定するようにタブルの更新を行う。この更新により、Transaction Time と Valid Time の値による機能の利用制御が実現可能となる。タブルを挿入せずに保持することにより、過去のデバイスの利用履歴をタブルに保存することが可能になる。

図 4 に、デバイスの機能の利用の流れについて示す。とあ

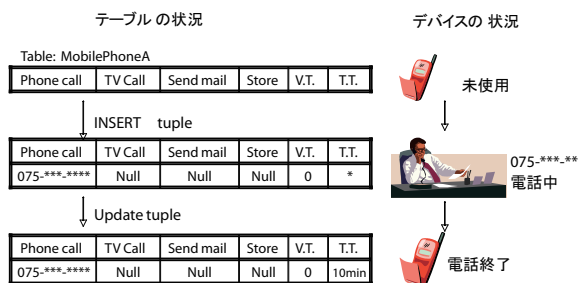


図 4 デバイスの機能の利用

る携帯電話について、「電話 (Phone Call)」「テレビ電話 (TV Call)」「メール送付 (Send Mail)」「データ保存 (store)」という機能がある場合を考える。そのとき、これらの機能に加えて Transaction Time (T.T.) と Valid Time (V.T.) をフィールドとして持つタブル MobilePhoneA が用意される。このタブルに何もタブルが存在しない状態では、対応する携帯電話は何も利用されていない状態に相当する。この携帯電話を使って、「075-***-****」という番号に電話をかける場合、タブル MobilePhoneA に Phone Call のフィールドの値が「075-***-****」という値を持つタブルを挿入する。このとき、他の機能を同時に利用することはないので、他の機能のフィールドには NULL 値を設定する。さらに、電話をかける場合、いつ終話となるかが判然としないので、V.T. のフィールドには不定値*を設定し、T.T. の値には通話開始時刻を設定する。タブルの挿入が完了すると、通話機能の利用が可能となり、設定された電話番号への通話が始まる。そして、10分後に終話となり電話を切った場合には、先ほど通話をするために挿入したタブルの値更新が行われる。不定値を設定していた V.T. の値を更新し、通話をしてきた時間である「10min」が値となるようにクエリを適用する。これにより、通話の機能が通話開始から10分間有効であったことが設定され、つまり、通話開始から10分後に電話を切った（通話機能の利用が終了した）ことが実現される。

3.2 デバイスのアクセス制御

実世界中でデバイスを使う場合、状況によってはデバイス自体、もしくはデバイスにある特定の機能についてはその利用を制限すべき状況が考えられる。例えば、電車の中では携帯電話の通話の機能のみを利用できないようにすべきである場合や、一時的に特定の機能の共有を拒否するような場合が考えられる。そのための手法について言及する。

まず、機能の利用を禁止するような場合には、対応するフィールドヘクエリを用いてアクセスができないようにする必要があるので、ビューを用いたフィールドへのアクセス制限を考える。機能の利用を制限する場合、既存のタブルに関して、アクセスを制御すべき機能に対応するフィールドを除いたビューを作成する。

図 5 でビューを用いたアクセス制御について示す。とある携

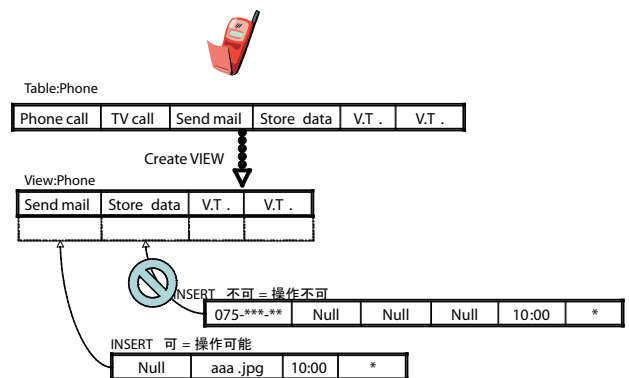


図 5 ビューを用いたアクセス制御

帯電話について、通話機能とテレビ電話機能の利用を禁止する場合を考える。そのとき、携帯電話に対応したタブルを元に、通話機能とテレビ電話機能に相当するフィールドを含まないビューを作成し、元のタブルへの直接のクエリ適用をできないようにする。それにより、クエリを用いて機能利用のためにタブルを挿入する時、もし通話機能に値を渡そうとするタブルは拒否され、一方で、写真データを保存しようとするだけのタブルは受け入れられる。

3.3 複数デバイス・機能の同時利用

ユビキタスコンピューティング環境下では、非常に多くのデバイスが身の回りに存在するようになることから、周囲にある複数のデバイスから適宜適切な機能だけを寄せ集めて同時に利用するような利用形態が生まれると考えられる。その実現方法について述べる。

個々のデバイスは、デバイスの持つ機能をフィールドとしたタブルとして表現されており、そのタブルにタブルを挿入することでデバイスの機能を利用することは、前に示している。ここで、データベース技術について考えてみると、複数のタブルから適宜フィールドを寄せ集めて仮想タブルを作成するビューがある。

そこで、複数のデバイスに分散して存在している機能を同時に利用するために、ビューを用いて仮想的なデバイスを作成し、そのビューに対してクエリ操作を行うことで分散して存在

するデバイスの機能を同時に利用することを実現する。作成するビューのフィールドには、利用するデバイスの機能を寄せ集め、さらに Transaction Time と Valid Time のフィールドを付加する。このビューに対してタブルの挿入を行うことで複数のデバイスに分散して存在している機能についても単一クエリを用いて機能の利用をすることが可能となる。

このとき、ビューに対してクエリ操作を行う時、元のテーブルに対してもタブル操作の結果を反映する必要がある。これは、個々のテーブルが対応するデバイスの状態を表すことから、その整合性をとる必要があるためである。例えば、あるビューに対してタブルが挿入された場合には、ビューを構成するフィールドの元となるテーブルに対してタブルを挿入する。

図 6 にて複数デバイス・機能の同時利用の例を示す。とある

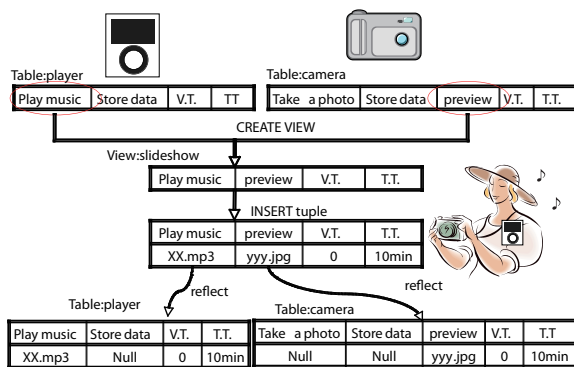


図 6 複数デバイス・機能の同時利用

ユーザがデジタルカメラとデジタルオーディオプレイヤーを別個に所持している状況を考える。その時、とある音楽をオーディオプレイヤーを使って聞きながら、デジタルカメラを利用してスライドショーを見る場合を考える。このとき、オーディオプレイヤーの音楽再生機能とデジタルカメラの写真プレビュー機能を同時に利用することになる。そのため、これら2つの機能と Transaction Time, Valid Time の4つをフィールドに持つビューを生成する。実際に音楽を聴きながら写真を見る場合には、このビューに対してタブルの挿入のための INSERT クエリを適用する。XX.mp3 という音楽ファイルを再生しながら yyy.jpg の画像をある基準時刻から10分間見る場合、音楽再生のフィールドに XX.mp3、プレビューのフィールドに yyy.jpg、T.T. フィールドに基準時刻 (0)、V.T. フィールドに 10min という値を設定したタブルを用意し、ビューに対して挿入する。ビューに対してタブルが挿入されると、その情報を元のデバイスのテーブルに対して反映するために、デジタルカメラとオーディオプレイヤーのテーブルにそれぞれタブルを挿入する。オーディオプレイヤーのテーブルには、利用しないデータ保存機能フィールドは null 値、他のフィールドには、ビューに挿入したタブルの値と同じ値を持つタブルを挿入する。またデジタルカメラのテーブルにも同様にタブルを挿入する。その際、カメラ上で利用しない写真撮影機能とデータ保存機能のフィールドの値は NULL、他のフィールドにはビューに挿入したタブルの値と同値を設定したものとなる。

3.4 複数デバイスの連続利用

たくさんのデバイスが利用可能な場合、前節のように同時に利用する場合もあれば、連続して利用する場合も考えられる。その実現方法について述べる。

連続的にデバイスを利用するためには、あるデバイスにタブルが挿入された際に、そのイベントに基づいて、次に利用するデバイスにタブル挿入を行うようにする必要がある。これは、ECA ルールを用いてテーブルに対してトリガを設定することで実現可能である。トリガを用いて発行するクエリによっては、単一のデバイスに対してのみ発行したクエリを適用するのではなく、多数の条件にあったデバイスに対してクエリを発行・適用すると行ったことも可能である。

図 7 で複数デバイスの連続利用の例を示す。ある PC で受信

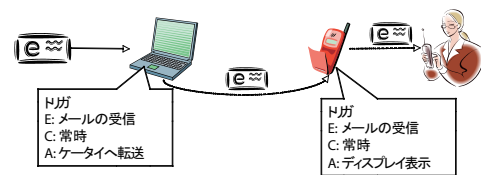


図 7 複数デバイスの連続利用

したメールを順次携帯電話に転送し表示するような場合を考える。このとき、PCで受信したメールを転送するためには、PCのメール送付機能を利用して、メールを受信したときに、携帯電話に送るためのメールアドレスを送付先としてメールを送付するためのクエリを発行するトリガを設定する。また、携帯電話では、受信したメールを即ディスプレイで表示させるには、メールを受信したときにそのメールを携帯電話の表示機能を通して表示させるためのクエリを発行するトリガを設定する。これにより、PCにメールが送られてくると、まずPCに設定されたトリガにより、そのメールを携帯電話に送付するためのクエリが発行され、メールが転送される。そして、転送されたメールが携帯電話で受信すると、携帯電話に設定されたトリガにより、受け取ったメールが自動的に携帯電話のディスプレイに表示するためのクエリが発行され、メールがディスプレイに表示されるようになる。

4. 関連研究

TinyDB [1][2] では、センサーネット上に多数に存在するセンサー (mute) からデータを取得し利用するための仕組みを提案している。ここでは、1つのセンサーネットを1つのデータベースのテーブル、1つの mute をデータベースのテーブルにおける1つのタブルと見ることにより、TinySQL と呼ばれる SQL 言語を用いて mute からのデータ抽出などを行うことが可能となっている。この手法においては、システムに与えられるクエリによって、各センサで取得されるデータの種類・送信量などのフィルタリングを行うことや、クエリに関数を埋め込むことにより mute に付属する機能の操作等が可能になっている。しかし、本研究においては、クエリ自体が機能操作等を意味しており、TinyDB で実現される仕組みの上位レイヤーの仕組み

を指向している．提案する手法によって TinyDB で実現されるセンサーネットワーク自体をテーブル表現にし，管理することが可能になると考えられる．

また，SuperSQL [3] では，SQL を拡張することで，データの管理のみならず，取得したデータの出力形式の制御までを実現する仕組みが提供されている．クエリの利用目的は本研究と大きく異なっているが，クエリ操作によって本来のデータ管理以外の仕組みを実現するという点については本研究と共通した点があると考えられる．しかし，データベースに保存されたデータの出力制御という点に着目している SuperSQL とは違い，本研究では，クエリで実現されることはデバイス（機能）の利用や管理といったものになる．そのような点で大きな違いがあると考えられる．

Web サービスの仕組みを利用した機能連係などの仕組みをいくつか取り上げる．

ActiveXML [4] では，XML 文書の中に Web サービスの呼び出しの記述を追加することにより，動的にコンテンツの生成・提供する仕組みが提案されている．この仕組みは，見方を変えると，分散して存在するデバイスで提供される機能呼び出しで利用するためのフレームワークとして見る事ができる．この仕組みでは，あらかじめどこでどのような名前でのどのようなサービスが提供されているかを知っている必要があり，Web サービス検索のための仕組みなどは別途必要となる．しかし，本研究における手法においては，各デバイスで提供される機能はテーブルとして表現されることから，検索のクエリを用いることで検索も可能である．さらに，機能を利用する場合も，ActiveXML でのサービス呼び出しのように 1 つの呼び出し命令で単一の機能を利用するだけでなく，WHERE 節で条件を指定することで単一のクエリで同時に複数のデバイスに存在する機能を利用するような事も，本研究で提案する手法では可能である．

また，Task Computing [5] では，Semantic Web 技術を用いて，デバイス間連携を実現する仕組みが提案されている．この仕組みにおいては，デバイスで提供される各機能は Web サービスとして扱われ，その機能の提供する内容などが OWL-S を用いて記述される．これにより，ユーザが何らかのタスクを実行する際には Semantic Web 技術により各デバイスで提供されるサービスの中から必要なものを選択し利用することが可能となる．この仕組みでは，基本的に可能なことはサービスの検索，合成，利用といった点である．しかし，本研究で提案する手法においては，これらの機能（サービス）利用といった事に加えて，機能の利用制限をはじめとするような機能の管理までもがクエリを用いて実現が可能である．また，クエリを用いることで集合的な操作が可能になることから，特に膨大な数のデバイスを対象として機能連係をするような場合には，Task Computing における OWL-S による機能に関する記述の解釈やそれに基づく機能合成や利用と比べて，クエリを用いた検索や機能利用・合成の方が同時に多数の対象に適用することが可能になることから強力であることが見込まれる．

5. おわりに

本論文では，実世界中に膨大な数存在するデバイスを効率的に管理利用するための仕組みについて提案を行った．個々のデバイスをデータベースにおけるテーブルとし，テーブルのフィールドにはデバイスの提供する機能および Transaction Time と Valid Time という時間情報を設定することで，テーブルに対してクエリ操作を行うことでデバイスの利用などが実現されることになる．機能の利用にはクエリを用いたタブルの挿入を利用し，機能のアクセス制御をするためにはビューを用いることで実現される．複数のデバイスに存在する機能を同時利用する場合には，ビューを用いて仮想のデバイスを構成し，そのビューに対するクエリ適用により複数のデバイス・機能の同時利用を実現することができる．さらに，デバイスの連続的利用の場合は，テーブルに対してトリガを設定することにより ECA ルールに基づいた連続的クエリ発行・適用により実現される．

このような仕組みにより，実世界中にあるデバイスを集合的に管理・操作することが可能になると考えられ，ユビキタスコンピューティング環境下でのデバイス管理に有用であることが見込まれる．

謝 辞

本研究の一部は，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」，文部科学省科学技術振興費プロジェクト「異メディア・アーカイブの横断的検索・統合ソフトウェア開発」（代表：田中克己），および，平成 17 年度科研費特定領域研究 (2)「Web の意味構造発見に基づく新しい Web 検索サービス方式に関する研究」（課題番号：16016247，代表：田中克己）によるものです．ここに記して謝意を表すものとします．

文 献

- [1] TinyDB: <http://telegraph.cs.berkeley.edu/tinydb/>
- [2] S.R.Madden, M.J.Franklin, J.M.Hellerstein, W.Hong, “An Acquisitional Query Processing System for Sensor Networks”, ACM TODS Vol.30, Issue 1, pp.122–173, March 2005
- [3] M.Toyama, “SuperSQL: An Extended SQL for Database Publishing and Presentation”, Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, pp.584–586, 1998
- [4] Active XML: <http://activexml.net>
- [5] R.Masuoka, B.Parsia, Y.Labrou, “Task Computing - The Semantic Web Meets Pervasive Computing”, Proceedings of The 2nd International Semantic Web Conference, pp.866–881, 2003