卒業論文　　2003年度(平成15年度)

センサのメタ情報を利用したセンサデータ取得に関する研究

指導教員

慶應義塾大学総合政策学部

徳田 英幸

村井 純

楠本 博之

中村 修

南 政樹

慶應義塾大学　総合政策学部　総合政策学科

氏名：丸山 大佑

卒業論文要旨　　2003年度 (平成15年度)

センサのメタ情報を利用したセンサデータ取得に関する研究

論文要旨

本研究では，ユビキタスコンピューティング環境において，多様なセンサからセンサのメタ情報を利用してセンサデータを取得するミドルウェアMARSを提案する．

近年，コンピュータの小型化と処理能力の向上や通信技術の進歩により，あらゆる機器にセンサやコンピュータを埋め込み，ユーザの作業を支援するユビキタスコンピューティング環境の実現が可能となった．ユビキタスコンピューティング環境では，コンピュータが環境に埋め込まれたセンサを利用して機器やユーザの位置情報，部屋の温度情報などの環境情報を取得し，環境の変化に沿ったサービスや，ユーザや機器の状況に適したサービスを提供することができる．

現在，ユビキタスコンピューティング環境に設置するセンサとして，計算機能を持った無線センサノードが提案されている．これらは将来的に小型で安価になると予測され，屋内、屋外への大量設置が期待できる．しかし，現在の無線センサノードを対象とした研究では，設置空間上にある物や空間の名前とその位置情報との名前解決の手法が提案されていない．そのため，アプリケーションは“ユーザの周りの音量〞床全体の温度”といった物や空間の名前をもとにした環境情報の要求を無線センサノードに対して発行できない．そこでMARSでは，アプリケーションからセンシングの対象物を指定した要求を受け付け，対象物に付属するセンサや，対象物周囲の空間に設置されているセンサで取得したセンサデータを提供する．これにより，アプリケーションは個々のセンサを意識することなく，必要な範囲のセンサ情報を取得できる．

本論文では，小型センサノードを利用して，MARSを実装し，評価を行なった．その結果，MARSはアプリケーションにより，センシングの対象物を指定したセンサデータの要求を受けて，センサデータを取得，提供できた．また，アプリケーションから要求を受信してから，センサデータを提供するまでの時間を計測し，MARSが実用的な時間内にセンサデータを提供できることを示した．

キーワード：

1 ユビキタスコンピューティング環境2 ミドルウェア3 センサネットワーク4 多様性5 抽象化

慶應義塾大学　総合政策学部　総合政策学科

丸山 大佑

Abstract of Bachelor's Thesis

Sensor data acquisition utilizing sensor meta data

Academic Year 2003

Summary

In this research, we propose a middleware called MARS which acquires sensor data from sensors utilizing sensor meta data.

Recently, advances in portable devices and radio technology have made deployment of ubiquitous computing possible. In ubiquitous computing environments, services suitable for users' and devices' situations can be provided utilizing information acquired by sensors embedded in environments such as a location information of a user or a room temperature.

Currently, wireless sensor nodes with computational power are proposed as sensors used for ubiquitous computing environments. These nodes are predicted to be cheaper in the future, and expected to be in wide use. However, current research that target wireless sensor nodes do not provide name resolution scheme that relates names of object and space with its location. Thus, applications are unable to send request to wireless sensor nodes that depends on names for objects or spaces such as "the volume around the user" or "temperature of the entire floor". In contrast, MARS accepts requests with specified sensing area from application, and provides sensor data acquired from sensors in the specified sensing area. This enables applications to acquire sensor data of the target area without considering independent sensors.

In this thesis, we have implemented MARS on wireless sensor node. The evaluation results show that MARS is able to provide sensor data for requests that specify sensing area. In addition, we have evaluated the delay and have shown that MARS can provide sensor data within practical time.

Keyworkds:

1 ubiquitous computing environment 2 middleware 3 sensor network 4 diversity 5 abstraction

Keio University Faculty of Policy Management
DAISUKE MARUYAMA

目 次

図 目 次

表 目 次

第 1 章

序論

本章では，本研究の背景であるユビキタスコンピューティング環境について述べる．次に，本研究の目的であるメタ情報を利用した要求によるセンサデータの取得について述べる．最後に本論文の構成について述べる．

1.1 本研究の背景

近年，コンピュータの小型化と処理能力の向上や通信技術の進歩により，あらゆる機器にセンサやコンピュータを埋め込み，それらのセンサやコンピュータが動作することによりユーザの作業を支援するユビキタスコンピューティング環境 [3] の実現が可能となった．ユビキタスコンピューティング環境では，ユーザは実行中の作業をよりよい環境へ移動させたり [4] 作業を自動化してユーザの入力なしに機器を制御 [5] することができる．特に，作業を自動化するアプリケーションはコンテキストアウェアアプリケーションと呼ばれ，様々な研究 [6] が行われている．コンテキストとは，機器の動作状況やユーザの意図，機器やユーザの周囲の状況のことを意味する．コンテキストアウェアアプリケーションは，環境内に設置されたセンサから位置情報や温度情報などの環境情報を取得して，環境やユーザ，機器のコンテキストを推測し，コンテキストに適したサービスを自動的に提供する．

例えば，寝室においてネットワークに接続された様々なセンサが分散配置されていることを想定した場合，ユーザの周りに設置された照度センサとベッドに設置された圧力センサの値からユーザが睡眠中であるというコンテキストが抽出可能であり，ドアに設置された加速度センサでドアの開閉というコンテキストが抽出可能である．コンテキストを抽出することによって，睡眠中なら冷房の強度を弱めるというようなコンテキストに応じたサービスを提供することが可能である．このとき，アプリケーションはユーザやベッドのように，モニタリングする対象やアクチュエータが作用できる対象に関する環境情報を必要としている．

現在のコンピューティング環境では，エアコンに付属する温度センサや自動ドアに付属する人体検知用焦電赤外線センサのように，環境に設置されているセンサは，機器やアクチュエータと一対一の関係で固定的に接続されている．このため，エアコンに付属している温度センサではなく，ユーザ付近に存在する温度センサの温度情報を元にエアコンが空調を行うというような、機器やアクチュエータ自身に付属するセンサ以外から環境情報を取得することが必要とされるアプリケーションは実現が困難である．それに対して，ユビキタスコンピューティング環境では，アクチュエータとセンサ，そして，それらを制御，利用するアプリケーションを分離することで，従来のコンピューティング環境より多くの環境情報を利用することを可能とし，コンテキストアウェアアプリケーションや環境モニタリングを実現することができる．

図 1.1 に，機器とセンサが固定されている現在のコンピューティング環境と，機器とセンサ，アプリケーションを分離したユビキタスコンピューティング環境の違いを示す．

1.2 本研究の目的

ユビキタスコンピューティング環境では，様々なセンサが多数分散配置されることが考えられる．センサには温度センサや加速度センサ，照度センサなど，様々な種類の環境情報を取得することを目的としたセンサが存在する．同じ温度を取得するセン

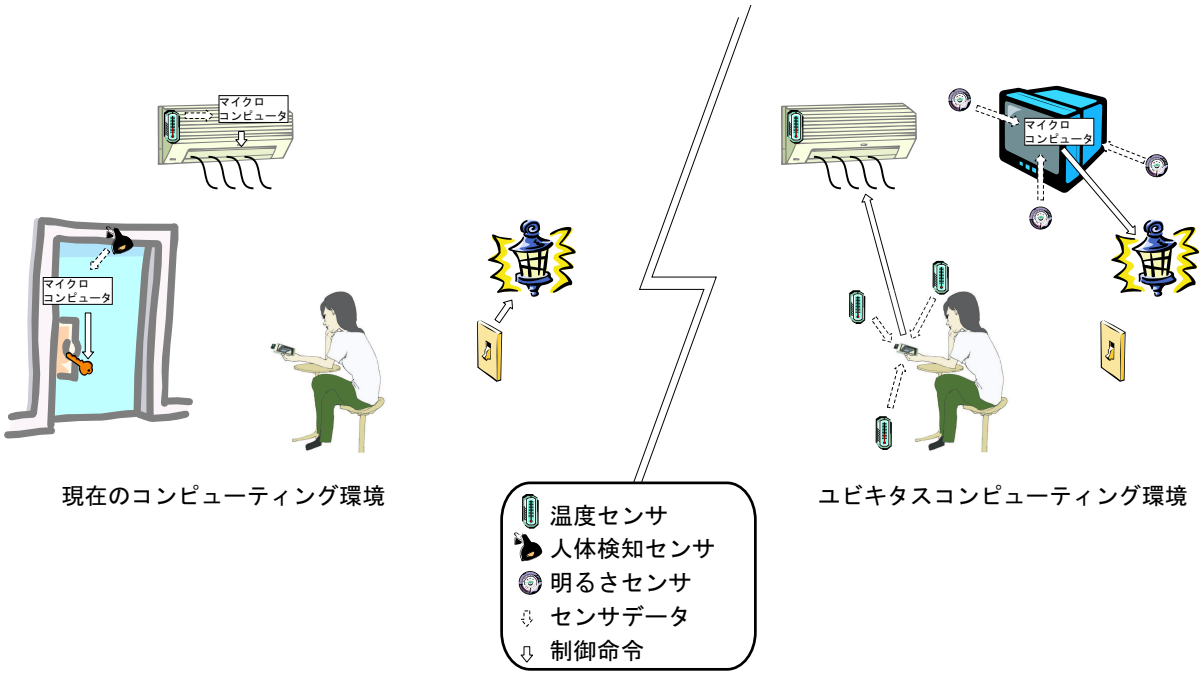

図 1.1: 既存のコンピューティング環境とユビキタスコンピューティング環境

サでも，分解能や取得範囲が異なるものが存在する．また，センサ情報を必要とした場合においても，取得できる全てのセンサデータは必要とせず，全てのセンサのうち，一部のセンサで取得したセンサデータのみを利用する事が多い．そのような場合，環境に存在する全てのセンサを各アプリケーション毎に把握し，それぞれのセンサと通信してセンサデータを利用すると，アプリケーションの処理やアプリケーション作成者の負担が増加する．これまで，一対一で固定的に接続していたアクチュエータ，アプリケーション，センサの関係を分離し，アプリケーションが様々なアクチュエータやセンサを利用できる．しかし，アプリケーションが様々なセンサを扱うことになるため，センサを利用する時に，センサの多様性や冗長性が問題となる．本研究では，センサの多様性，冗長性をアプリケーションが意識することなくセンサデータを利用できるセンサデータ取得ミドルウェアを構築する．その際，アプリケーションがユーザや機器，およびその周囲に関するセンサデータを取得することに着目したシステムを構築し，センサの種類や付属する物というようなセンサのメタ情報を利用したセンサデータの取得を実現する．これにより，アプリケーションは個々のセンサを意識せずにセンサデータを取得できる．

1.3　本論文の構成

本論文では，第２章で本研究におけるユビキタスコンピューティング環境の定義，特徴について述べる．第３章では本研究の想定環境を述べ，関連研究を考察し，現状におけるセンサデータ取得に関する問題点について整理する．第４章では各問題を解決するシステムの設計について述べる．第５章では，本研究の実装を説明し，第６章で関連研究との比較，評価を行う．最後に第７章で本論文をまとめる．

第 2 章

ユビキタスコンピューティング環境におけるセンサの利用

本章では，本研究におけるユビキタスコンピューティング環境の定義を行い，その特徴について述べる．次に，ユビキタスコンピューティング環境におけるセンサの利用について考察する．

2.1 ユビキタスコンピューティング環境

本節では，ユビキタスコンピューティング環境の特徴について述べ，ユビキタスコンピューティング環境を構成する要素について論じる．

2.1.1 ユビキタスコンピューティング環境の特徴

従来のコンピューティング環境では，サーバ，パーソナルコンピュータ（以下 PC），PDA などの高機能で汎用性のあるコンピュータは，コンピュータ同士または，その入出力装置であるプリンタなどとネットワークにて接続されている．また，組み込みマイコンのような小型で単機能のコンピュータは，家電などの機器の中でセンサやアクチュエータを制御している．そして，冷蔵庫やオーディオなどの家電はネットワークには接続されず，各々が単独で動作していることが一般的である．

しかしながら，最近のデバイスの能力や通信機能の向上により，IT 冷蔵庫 [7] やネットワークテレビ [8] のように，これまでネットワーク接続性のなかった機器がコンピュータやセンサ，他の機器とネットワークにて接続され，データのやり取りを行うことが可能になった．そしてそのようなネットワーク接続性のある家電を情報家電と呼んでいる．また，これまでコンピュータが搭載されていなかった鏡や柱 [9]，タンス [10] などの家具にも小型のコンピュータやセンサ，ディスプレイなどを搭載し，ネットワークを介して他のコンピュータや機器とから操作，利用する研究も行なわれている．

デバイスの能力や通信機能が向上したことで，コンピュータやセンサや情報家電，などが相互に接続することが可能となり，これまでのコンピューティング環境で実現できなかったような，１つの機器，１つのシステムという枠を越えた協調動作が可能となる．例えば，外出先で PDA から冷蔵庫の中身を確認し，買い物を済ませて帰宅，インターネット上のレシピ通りに電子レンジを利用して調理するというようなことが可能になる．また，PC 上でビデオを再生し，PDA 上で操作し，テレビで表示するというように，機器内の一部の機能やセンサ，アクチュエータを組み合わせて１つのアプリケーションを動作させることができる．このように，環境に多数のコンピュータやセンサや情報家電，などの機器を設置し，それらをネットワークに接続することで協調動作させる環境は，ユビキタスコンピューティング環境と呼ばれ様々な研究が行なわれている．

ユビキタスコンピューティング環境では，様々なアプリケーションがネットワークに接続する様々なコンピュータや機器やセンサを利用する．このため，アプリケーションが利用できる機器やセンサの種類は数多く存在する．また，同じ目的で利用する機器やセンサでも機能や性能，インタフェースが異なるという状況が存在する．例えば，同じテレビでも，画面サイズや，入出力ポート数，形式などについて，性能が異なる．センサの場合でも，温度や明るさなど，取得できるセンサデータの種類は多様であり，同じ温度センサでも，取得できるセンサデータの分解能や取得範囲，サンプリングレートなどの性能が異なる．様々なセンサを１つのアプリケーションが利用するためには，そのようなセンサの種類や性能の違いを考慮しなければならない．どのアプリケーショ

ンからも同じようにセンサを扱うためには，それらのセンサの違いを吸収することが必要である．

ユビキタスコンピューティング環境のアプリケーションの特徴として，コンテキストアウェアネスが挙げられる．コンテキストアウェアネスとは，環境情報を取得することによって状況に適応し，自律的に提供するサービスを変化させることを指す．ユビキタスコンピューティング環境では，環境に多数のセンサを設置し利用することで，環境の変化を察知したり状況を把握することを可能とし，コンテキストアウェアアプリケーションを実現できる．

2.1.2 ユビキタスコンピューティング環境の構成要素

本研究におけるユビキタスコンピューティング環境の構成要素を示す．

- **コンピュータ**
  本研究において，コンピュータは計算機能および計算機能を有するデバイスである．コンピュータにはPCやサーバのような高機能で汎用性のあるコンピュータや，組み込みマイコンのような小型で単機能のコンピュータがある．

- **ネットワーク**
  ネットワークは，センサで取得したセンサデータやコンピュータで計算した情報やファイル，アクチュエータに対する操作命令などの情報をデバイス間で交換するインフラストラクチャである．既存の汎用コンピュータを中心としたネットワークだけではなく，ユビキタスコンピューティング環境では，１つのセンサやアクチュエータ，情報家電などがネットワークに接続する．

- **センサ**
  センサは，実世界の物理量を情報世界の情報に変換するデバイスである．センサでは，実世界の物理情報を電圧や電流，抵抗値といった電気的な信号に変換している．コンピュータで環境情報を利用するためには，センサで変換された電気的な信号をコンピュータにて計算可能なデータ形式に変換する必要がある．

  ユビキタスコンピューティング環境におけるセンサは，センサがアクチュエータと固定的に接続されていないため，取得したセンサデータを加工し，アプリケーションに送信しなければならない．このため本研究において，センサは計算機能とネットワーク接続性を持つもののことを示す．センサには照度センサや湿度センサのように，設置された空間の物理量を計測するものと，加速度センサやひずみセンサのように，設置された物に関する物理量を計測するものがある．また，温度センサのように，室内の温度やユーザの体温など，両方の物理量を計測するセンサもある．図2.1に代表的なセンサの分類を示す．

- **アクチュエータ**
  アクチュエータはコンピュータによって出力された計算結果を基に動作を行うデバイスである．アクチュエータはモータのような動力となるものだけではなく，

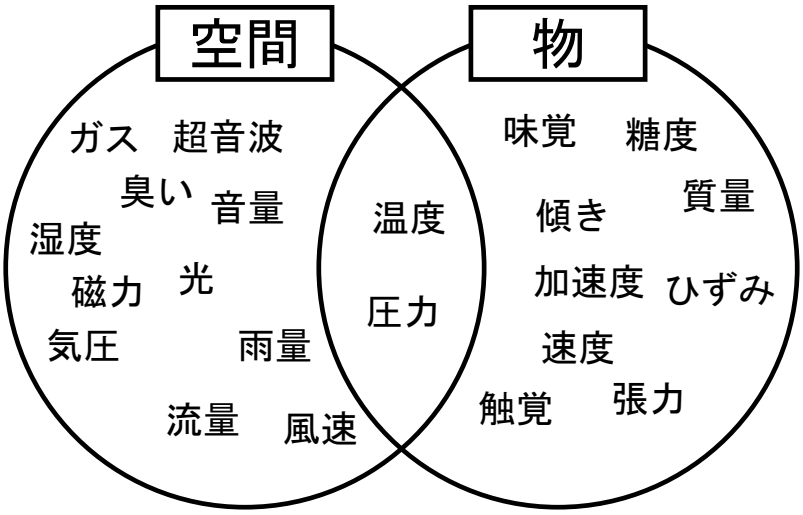

図 2.1: センサの分類

電気信号を基に物理的に動作する，電灯やスピーカ，電熱器などを含める．また，エアコンの冷房や送風，テレビの受像のような家電の機能，ディスプレイやプリンタといったPCに接続されているデバイスの機能もアクチュエータに含まれる．ユビキタスコンピューティング環境では，アクチュエータとアプリケーションを分離するため，アクチュエータを操作するアプリケーションが家電などの機器のコンピュータ上で動作するとは限らない．本研究において，アクチュエータはネットワーク接続性を持ち，ネットワークを介して操作できるものを対象とする．

- **アプリケーション**
  アプリケーションは，ユーザの作業を支援するサービスを提供するソフトウェアである．アプリケーションは，コンピュータ上でのみ動作するのではなく，ネットワークによってデータを通信したり，センサから実世界の情報を取得したり，アクチュエータを制御して実世界に物理的に働きかける．ユビキタスコンピューティング環境においてアプリケーションは，自身が動作している機器やコンピュータの持つセンサやアクチュエータだけでなく，ネットワークを介して別の機器が持つアクチュエータや単体で存在するセンサを利用する．

- **ミドルウェア**
  ミドルウェアは様々なアクチュエータやセンサの差異を吸収し，どのアプリケーションからでも統一的にアクチュエータの操作やセンサデータの取得を行うことを可能とするソフトウェアである．例えば，セルシウス温度を提供する温度センサと絶対温度を提供する温度センサを利用する場合に，ミドルウェアによってアプリケーションがその違いを意識せずにアプリケーションが要求する形式で温度情報を利用することが可能になる．ミドルウェアは，センサ，アクチュエータ，アプリケーションの分離によって生じるセンサやアクチュエータの多様性を隠蔽

する．

センシング，アクチュエーションの対象

ユビキタスコンピューティング環境において，コンテキストアウェアアプリケーションやモニタリングアプリケーションがセンシング，アクチュエーションの対象とするものには，次の２種類がある．１つは家具や家電のような機器，傘や鉛筆，財布というような道具，ユーザや人，動植物などの生物などの物である．もう１つは物の周囲や部屋，廊下，階というような空間である．ユビキタスコンピューティング環境のアプリケーションは，このようなセンシングの対象の物理量を取得し，演算や蓄積をする．またその演算結果や蓄積されたデータを元に，アクチュエーションの対象に対して暖める，照らす，動かすなどの物理的な働きかけを行う．本研究では，センシングの対象となる物，および，周囲という空間をセンシング対象とする物をセンシングの対象物と呼ぶ．

2.2 既存のセンサ環境

現在のコンピューティング環境では，アプリケーションや機器とセンサの関係は固定的である．例えば，エアコンが室内の温度を計測する場合，エアコンに付属する温度センサのみを利用し，自動ドアが人を検知する時に，自動ドアに付属する赤外線センサを用いて人がいることを検知する．エアコンが新たなセンサを利用することや，自動ドアが人がいることを検知するために他のセンサを利用するということは困難である．このような既存のセンサ環境を固定的なセンサ環境と呼ぶ．

2.2.1 固定的なセンサ環境の利点

固定的なセンサ環境では，各センサをアプリケーションや機器に対応した取得範囲や精度などの仕様にすることにより，アプリケーションや機器に最適なセンサを利用することが可能である．例えば，自動車のエアバッグは，エアバッグ用の仕様で作られたセンサを，エアバッグが占有して利用する．これにより，他のセンサを利用するより確実に衝突を察知する事が可能になり，他のアプリケーションがエアバッグのセンサにアクセスしているために，エアバッグが衝突を察知するのが遅れるということも回避される．

2.2.2 固定的なセンサ環境の問題点

固定的センサ環境では，特定の機器に特化したセンサを使用でき，機器がセンサを占有できるという利点がある．しかし，センサの増減やセンサの変更が必要になった場合に，アプリケーションや機器自体の仕様を大幅に変更する必要が生じたり，機器全てを入れ替えなければならなかったり，センサの増減，変更を考慮した設計を行う必要が発生したりするする．また，同じセンサを利用することができるアプリケーショ

ン，機器があった場合に，それぞれの機器においてセンサを増減，変更する必要があるというような問題がある．

2.3 ユビキタスコンピューティング環境におけるセンサ環境

ユビキタスコンピューティング環境では，環境モニタリングや，コンテキストアウェアなサービスを行う為に，環境に様々なセンサを分散設置し，各センサは不特定のアプリケーションから利用される．例えばユーザの周りの温度センサのセンサデータを利用して PC 上の温度管理アプリケーションがエアコンやストーブの設定温度を変えるというように，それぞれ別の機器上のセンサやアクチュエータなどを協調させることが可能である．このような環境に存在するセンサ，アプリケーション，アクチュエータが独立して存在し，必要に応じて接続されて利用するセンサ環境を動的なセンサ環境と呼ぶ．

動的なセンサ環境では，アプリケーションは利用するセンサが不特定である．このため，アプリケーションがセンサを利用する時には利用するセンサを指定しなければ，センサからセンサデータを取得することはできない．アプリケーションがセンサデータを利用して取得したい情報は、物や人などのコンテキストである．物のコンテキストは，物に関する物理量と物の周囲の空間の物理量から得られる．多くのセンサは，センサの種類によって取得できる情報が物に関する物理量か，空間に関する物理量かが決まっている．このため，物に関する物理量も，物の周りの空間の物理量も，対象とする“ 物 ”と“ センサの種類 ”を指定することにより，アプリケーションが取得するセンサデータを取得できる．また，アプリケーションが抽出したいコンテキストの対象として，部屋の中のように物とは関係ない空間がある．そのような空間のセンサデータを取得したい場合は,“ 空間 ”と“ センサの種類 ”を指定することによりセンサデータを取得できる必要がある．

本研究では，物や空間，センサの種類のようなアプリケーションがセンサデータを要求するためのセンサに関する情報をセンサのメタ情報とする．

現在行われているセンサプラットフォームの研究により，個々のセンサに通信機能や計算機能を持たせ，不特定のアプリケーションから利用利用することが可能である．しかし，現在のセンサプラットフォームでは，次のようなハードウェアの課題が残っているため，ハードウェアの進歩やソフトウェアによる補完が必要である．

- 計算能力が非力であるため演算に時間がかかる
- 消費電力に対するバッテリ容量の制約の問題により，電池寿命が短い
- 電波や赤外線などの無線による通信技術が確立されておらず，不安定である

2.4 動的センサ環境におけるセンサを利用したアプリケーション例

本節では，動的センサ環境におけるセンサを利用したアプリケーション例を示し，その特徴について考察する．

2.4.1 ユーザのコンテキストに対応した空調

座面や背もたれに圧力センサや温度センサ，足に加速度センサなどを設置した椅子を想定する．椅子に付属するセンサから得られるセンサデータをを観察することで，ユーザが現在どのような作業を行なっているかというコンテキストが推測できる．また同様に，ベッドに圧力センサを敷き詰めたり，音量センサを設置することでユーザの睡眠状況が推測できる．このとき，ユーザの周囲の温度センサや湿度センサからセンサデータを取得することで，ユーザのコンテキストにあわせた空調を行うことができる．(図 2.2)

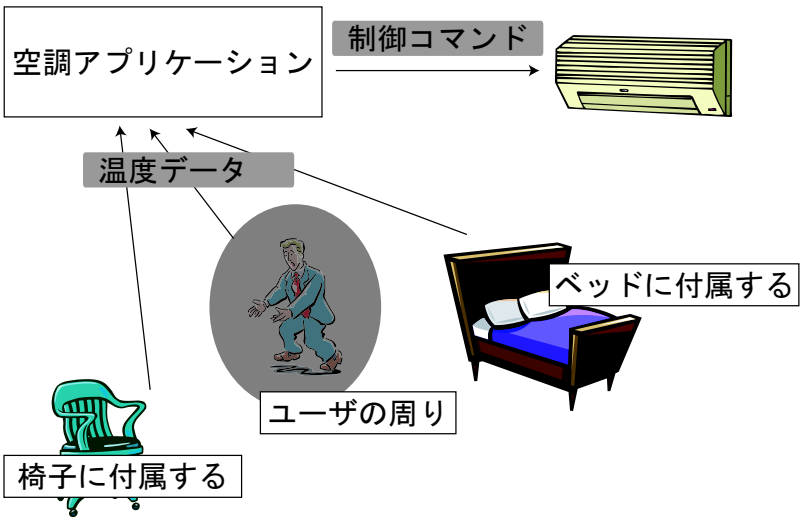

図 2.2: ユーザのコンテキストを考慮した空調アプリケーション

環境情報を利用することで，ユーザなどセンシングの対象のコンテキストを抽出することが可能になり，コンテキストアウェアアプリケーションを実現できる．

2.4.2 机上物体検知

天板に圧力センサや明るさセンサが敷き詰められ，RFID のリーダが設置されているような机を考える．鉛筆や CD ケースのように１つ１つにセンサやコンピュータが載っていないような物でも，RFID タグのようなパッシブタグを添付することは可能であ

る．RFID タグは既に数ミリ四方という小型化が実現しており [11]，全ての物が RFID 管理されることは考えられる．机の上に財布を置くと圧力や明るさが変化し，机の上に何かが置かれたという情報が得られる．また，RFID リーダにより，机付近に財布が現れた情報が得られる．これらの情報から，財布が机の上のどこにあるかという情報がわかる．アプリケーションが机の上の物とユーザの位置を把握することで，例えばこのままユーザが立ち去ろうとした時に財布を忘れていることを通知するというようなことが可能である．

2.4.3 屋内環境モニタリング

屋内環境をモニタリングするセキュリティシステムを考える．窓に振動センサを設置し，窓の振動を監視して侵入者を検知したり，室内のガスセンサや温度センサのセンサデータから，ガス漏れや火事を検知したりするためにセンサが利用することができる．このとき，窓の振動のように窓に関するセンサデータの要求と，室内のガスや温度のように室内空間に関するセンサデータの要求が生じる．

動的センサ環境におけるアプリケーションの特徴

動的センサ環境では，アプリケーションは椅子に付属するセンサ，ユーザの周りの温度センサ，机の上の圧力センサと明るさセンサというように，センシングの対象となる物毎のセンサデータを利用することが多い．なぜなら，アプリケーションが取得したいセンサデータは，アクチュエータによって作用できる対象や，コンテキストを抽出したい対象だからである．しかし，鉛筆１本やクリップ１つというように，１つ１つにセンサを付属させることが困難な物も存在する．このような場合，１つ１つのセンシング対象にセンサを付属するのではなく，環境に設置されたセンサを利用することで，センシング対象単位のセンサデータの要求が可能になると考えられる．また，空調システムや室内監視システムなど，室内というような空間をセンシング対象としてセンサデータを利用するアプリケーションも存在する．

2.5 動的センサ環境の問題点

動的なセンサ環境では，同じアプリケーションを様々な環境で利用したり，アプリケーションが動作しているセンサ環境が変化したりするため，アプリケーションを実現する際に，アプリケーション作成者がそのアプリケーションが利用される環境のセンサ環境を知ることができない．このため，センサ環境に依存せずにアプリケーションがセンサデータを取得する必要がある．その際，アプリケーションにとって，必要な情報は“ ある場所を人が通った ”や,“ テレビの周りの明るさ ”というような情報であって，その情報がどのセンサで取得されたかという情報や，各センサの情報は必要ない．

動的なセンサ環境では，センサとアプリケーションが分離しているため，既存のアプリケーションを利用すると次のような問題点が考えられる．

2.5.1 環境に存在するセンサの種類が不明であること

アプリケーション構築時に，アプリケーションが利用される環境のセンサ構成が不明である．例えば，室内の温度管理を行うアプリケーションは，室内に分散配置された温度センサを利用する．しかし，温度管理アプリケーション構築時には，利用環境に温度センサが存在するかは不明である．

2.5.2 各センサの通信手段が多様であること

センサからセンサデータを取得する際，アプリケーションは各センサと通信を行う．各センサには，RS232C や USB，Ethernet のような有線通信，赤外線や電波を利用した無線通信など，様々な通信手段が備わっていることが考えられる．動的なセンサ環境では，同じ種類のセンサにおいて，アプリケーションによって通信手段が異なることが想定される．例えば，アプリケーションがユーザの周りの温度情報を要求した場合，あるセンサはシリアル通信によってセンサデータを提供し，あるセンサは無線通信によってセンサデータを提供しているというように，１度のセンサデータ取得の際に複数の通信手段を利用してセンサデータを得るということも考えられる．このようなセンサデータの通信の差異を補完する研究 [12] が必要である．

2.5.3 センサ環境が時間経過に伴って変化すること

動的なセンサ環境では，アプリケーションの意図に関わらずにセンサが増減したり，移動したり，変更したりすることが考えられる．例えば，センサの数を増やして，センサデータを取得する粒度を細かくしたい場合，既存のアプリケーションでは，センサを増やすことは困難である．

2.5.4 必要なセンサデータの要求が困難であること

アプリケーションは環境に存在する多数のセンサによって取得したセンサデータの中から，目的に沿った一部のセンサデータを利用する．アプリケーションが必要とするセンサデータを取得可能なセンサに対して，どのようにしてアプリケーションの要求を伝えるかが問題となる．

この際，アプリケーションはセンサデータがどのセンサで取得されたかということを知る必要はなく，“ ドアの前に人がいるかどうか ”※ “ ユーザの周りの温度は何度か ” というようなセンサデータのみが必要である．

2.5.5 センサ数が十分であるか不明であること

動的なセンサ環境では，様々なアプリケーションからセンサが利用されるため，個々のセンサは特定のアプリケーションに特化せず，汎用的な作りにすることになる．このため，１つ１つのセンサはアプリケーションが必要とする精度などを満たさない場

合には，大量に分散配置されたセンサのうちの多数のセンサで取得したセンサデータを利用して精度を補う必要がある．その際，アプリケーションが動作するセンサ環境に，精度を補うのに十分な数のセンサが存在するかどうかが不明である．

本研究では，以上のようなアプリケーションとセンサの分離による問題点を解決するために，アプリケーションとセンサの間にミドルウェアを構築する．これにより，アプリケーションは個々のセンサの差異を意識することなく，センサデータを取得できる．

2.6 研究方針

前節では，動的センサ環境における既存のアプリケーションの問題点を挙げた．これらの問題を解決するために，本研究では，以下の機能を満たすシステムを構築し，運用および評価を行なった．

- 環境に存在するセンサの種類を把握すること
- センサ環境の変化に対応すること
- アプリケーションが容易にセンサデータを要求できること
- センサの数を把握すること

2.7 本章のまとめ

本章では，本研究で前提とするユビキタスコンピューティング環境を定義した．その後，ユビキタスコンピューティング環境におけるセンサ利用の特徴について述べ，既存の固定的なセンサ環境を想定したアプリケーションは，動的なセンサ環境では多くの問題があることを示した．

次章では，本研究の概要と動的なセンサ環境を実現する為の方針について説明する．

第 3 章

研究の方針と概要

本章では，本研究の想定環境について述べ，想定環境における関連研究の考察を行う．次に，関連研究の問題点を踏まえて，本研究の構築するセンサシステムの設計方針と概要について考察する．

3.1 想定環境

本節では本研究において想定するセンサやユビキタスコンピューティング環境について述べる．

3.1.1 利用するセンサ

本研究では，各センサがセンサデータを取得する機能の他に計算機能および通信機能を有することを想定する．

センサにおいても，小型化や高性能化が進み，1 立方センチメートル程度から 100 立方センチメートル程度の小型センサを環境に大量に分散配置し，環境情報を取得するようなセンサプラットフォームの研究 [13, 14, 15, 16] が多数行われている．このような研究で利用されるセンサは各センサが通信機能や計算機能，記憶装置，バッテリなどの電源装置を持ち，自律的に動作してお互いに通信することで協調作業を行うことが可能である．大量のセンサを設置することで，コンテキストアウェアアプリケーションや環境モニタリングなど，様々な分野で応用できる．本研究において想定するセンサの機能に合致する既存の小型センサの一例として，UC Berkeley の TinyOS Progect[13] で研究，開発が行われているセンサネットワーク用センサモジュールである mote の mica2[17] を図 3.1 に示す．mote は TinyOS Progect で使用されるセンサモジュールの総称で，mica2 はその中の 1 つである．

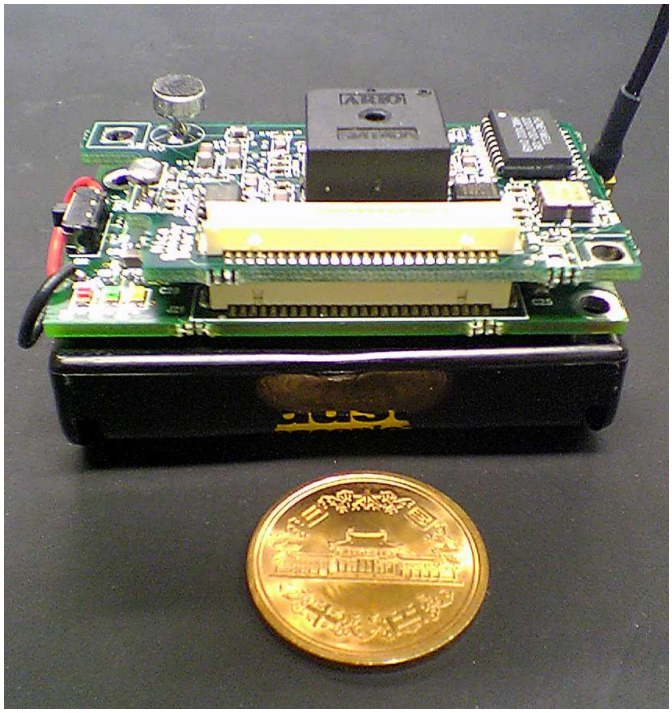

図 3.1: 無線センサネットワークプラットフォーム：mica2

mote は各センサモジュールで複数のセンサによるセンサデータの取得，CPU による演算，無線による通信を行うことが可能である．PC とのインタフェースは RS232C によるシリアル通信を用い，PC インタフェース基板に接続して利用する．本研究では，PC と接続し，センサ同士によるネットワークと Ethernet などの外部のネットワークを接続するノードをゲートウェイノードと呼び，ゲートウェイノードおよび，接続する PC の計算機能やネットワークインタフェースをまとめてゲートウェイと呼ぶ．

3.1.2 屋内ユビキタスコンピューティング環境

ユビキタスコンピューティング環境に限らず，様々な点において屋内と屋外とでは環境が大きく異なる．センサ環境について考えると，次のような違いがある．

- 相対的に屋内より屋外の方が広いため，同じ数のセンサを設置した時にセンサ間の距離，密度が異なる
- 屋外は天気や生物などの自然の影響を受けやすい
- 屋内の方がセンサの保守点検が行いやすい

このような違いから，屋内の方がセンサの設置，管理が行いやすく，また，より多くのセンサを集積させられる．本研究では屋内におけるセンサの利用について考察する．屋内では無線の電波強度が十分であるため，各センサ間の通信は 1 ホップで到達すると想定する．

3.2 関連研究

本節では本研究の関連研究を挙げ、関連研究の問題点について考察する。

3.2.1 TinyDB

本研究の関連研究である TinyDB について述べる．

TinyDB の概要

TinyDB[18] は UC Berkeley で行われている，センサネットワークからデータを抽出する研究である．センサネットワークから，宣言型クエリを記述することによりセンサデータを抽出するシステム TAG[18] が mote および TinyOS[13] 上で実装されている．アプリケーションは TinyDB のゲートウェイに対してクエリを送信する．ゲートウェイは，PC インタフェース基板に接続された mica2 および PC から構成される．

ゲートウェイが受け取ったアプリケーションの要求は，TinyDB では次のような流れで各センサに伝わる．図 3.2 に各センサへのクエリの伝播を示す．

1. ゲートウェイでクエリを分析し，最適化する

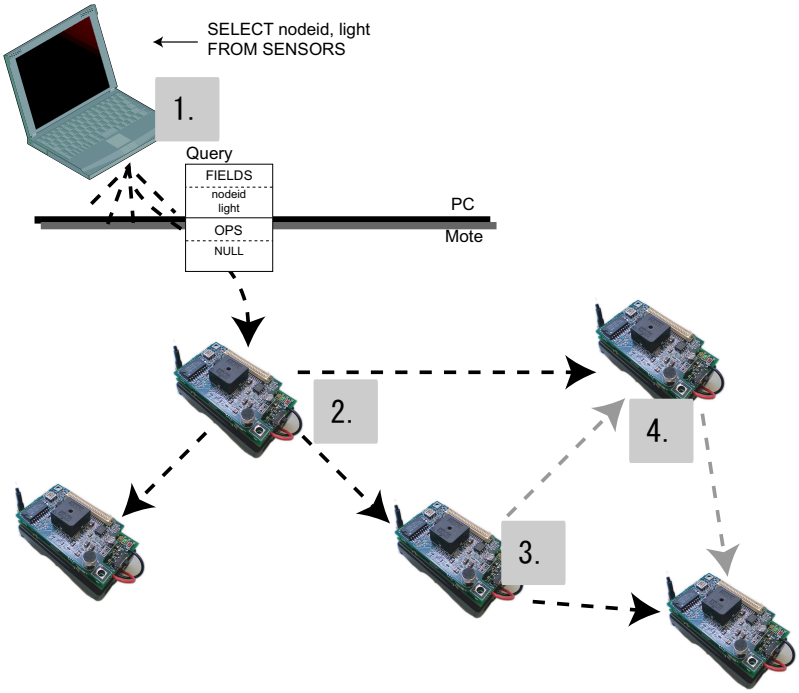

図 3.2: A query and results propagating through the network.[1][2] の図を基にした.

2. センサに対してクエリをブロードキャストする

3. クエリを受信した各センサは，受信したクエリをさらにブロードキャストする

4. 同じクエリを複数回受信した場合は，その中から親ノードを１つ選ぶ

各センサは受信したクエリがどのセンサから送信されたかという親センサの情報を保持する．各センサはクエリを分析し，クエリの対象となるセンサはセンサデータを取得する．取得したセンサデータを親センサに送信する．親センサは子センサから受信したセンサデータと自分で取得したセンサデータをまとめて，自分の親センサに送信する．センサデータをまとめて親センサに送信するため，送信量が減少し，センサのバッテリ消費を抑えることができる．TinyDB ではゲートウェイノードが全ての親センサとなるため，全てのセンサからセンサデータを収集したゲートウェイがアプリケーションに対してセンサデータを提供する．

TinyDB の目的

センサネットワークを構築するセンサの多くは電池を電源として駆動している．mica2 は，外部ＡＣ電源から電源を供給することも可能であるが，完全に自律したセンサとして動作するためには単３電池２つを利用する必要がある．このような電池駆動のセンサは電池の消費に伴い，電池を交換する必要がある．電力消費によるセンサの動作

不能状態の危険性および電池交換の労力を減らすために，消費電力を抑えることが重要である．センサの電力消費には待機状態，演算状態，受信状態，送信状態の４つのフェーズがある．４つのフェーズの内，送信状態が最も電力を消費するため，送信データを減らすことで電力消費を抑えることが可能である．TinyDB は効率的なデータアグリゲーションを行うことで各センサでの電力消費を抑えることを目的としている．

TinyDB の特徴

TinyDB ではゲートウェイで SQL 文を mote の無線で通信するバイナリに変換する．クエリはブロードキャストで各センサ伝わる．各センサでは自身のセンサのメタ情報と自身の親センサはどれかという情報を保持し，クエリを受信すると，自身がクエリの対象となるかを判断し，対象となる場合はセンサデータを親センサに送信する．また，受信したクエリをブロードキャストし，全てのセンサにクエリを伝える．子センサからセンサデータを受信した親センサは，自身で取得したセンサデータと子センサから受信したセンサデータをまとめ，上流のセンサに送信する．

TinyDB の問題点

TinyDB では，ある値を示した１つのセンサとの距離が 10m というように，あるセンサの周囲の空間というセンサデータの要求が可能である．しかし，機器やユーザなどの対象物の周囲という空間のセンサデータを要求する場合，対象物とそれに対応するセンサのバインドが必要である．しかし，TinyDB では具体的な実現方法が述べられていない．TinyDB は各センサがアプリケーションの要求を解決するため，各センサが自身と様々な機器の位置を保持する方法では，スケーラビリティの点で問題がある．また，TinyDB ではセンサのメタ情報を全て自身が管理し，温度センサや照度センサのサンプリングに必要な消費電力やサンプルレートなどを定期的にゲートウェイに送信している．しかし，位置情報をメタ情報として扱うことには言及されていない．

3.2.2 Cougar Approach

本研究の関連研究である Cougar Approach について述べる．

Cougar Approach の概要

Cougar Approach[19] とは Cornell University で行われている，センサネットワークからセンサデータを抽出する研究である．センサネットワークをデータベースとして扱い，宣言型クエリを用いてセンサデータを抽出する．センサ上で使用するリソースが少なくて済むようにユーザのクエリを利用する最適化して，センサネットワークのバッテリの消費を抑える．また，宣言型クエリによる要求であるため，センサネットワークの物理的特質はユーザから隠される．

Cougar Approach の目的

Cougar Approach では，センサネットワークにおける次の４つの物理的リソースの制限を挙げている．

- 無線通信の帯域が限られていること
- バッテリの寿命が短いこと
- 計算能力，メモリ容量が乏しいこと
- センサデータが不確実であること

Cougar Approach では，特にバッテリの寿命に着目し，クエリの最適化とデータのアグリゲーションを行うことでバッテリの消費を抑え，バッテリの寿命を延長している．

Cougar Approach の特徴

Cougar Approach では次の２つのことを行なっている．宣言型言語を利用することで，ユーザやアプリケーションがセンサネットワークの構成や，データの処理過程を知らなくて済む事．各センサの持つ計算機能でセンサデータをまとめ，不要なデータを除去することで，無線通信によるバッテリの消耗を抑える．

ネットワークレイヤとアプリケーションレイヤの間に，各センサのクエリプロキシによるクエリレイヤを構成する．クエリプロキシでは，ネットワークトポロジなどから消費電力が抑えられるようなクエリプランを作成する．

Cougar Approach の問題点

Cougar Approach では，センサのメタ情報を定期的にゲートウェイに送信するなどして，ゲートウェイが管理することにより，センサに送信するクエリの最適化を行なうことが述べられている．そして，メタ情報のデータサイズやセンサ環境の動的性により，ゲートウェイで管理すべきメタ情報とセンサで管理すべきメタ情報を分ける必要があると述べられている．しかし，具体的にどのメタ情報をゲートウェイで管理し，どのメタ情報をセンサで管理するのがよいかということは述べられていない．また，センサデータを要求する際に，センシングの対象物の周囲という要求を行なうことは述べられていない．

3.3 研究概要

ユビキタスコンピューティング環境には様々なセンサが多数存在する．コンテキストアウェアアプリケーションのようなセンサデータを必要とするアプリケーションは，環境に存在する多数のセンサが提供するセンサデータの中から，利用するセンサデータを抽出しなければならない．アプリケーションのセンサデータに関する要求は，セ

ンシングの対象物の周辺の温度というように，抽象的な要求である．本研究では，アプリケーションの抽象的な要求を解決し，個々のセンサからセンサデータを取得，提供するミドルウェアを構築する．

アプリケーションは，ユーザや機器というセンシングの対象物周辺のセンサや対象物に付属するセンサからセンサデータを基に，対象物のコンテキストを抽出したり，対象物を制御する．このため，アプリケーションは，対象物を指定してセンサデータを取得することが必要である．センシングの対象物を指定してセンサデータを取得するためには，対象物やセンサと位置情報のマッピングが必要である．本研究で構築するミドルウェアは，アプリケーションにセンシングの対象物に関するセンサデータや対象物の周囲のセンサデータを提供する．センシングの対象物と各センサを同じ基準の位置情報で扱うために，同じ位置情報取得システムにより対象物とセンサの位置を取得することが必要である．センサの位置情報を他のメタ情報と共にミドルウェア内部にデータベースとして保持することで，対象物の周囲というような要求を解決する．

3.4 本研究におけるメタ情報の定義

本研究では，アプリケーションがセンサデータを要求する際に，センサの種類やセンシングの対象となる物などのセンサのメタ情報を利用する．本研究におけるメタ情報とは，センサデータに関するセンサデータ以外の情報である．表 3.1 に本研究で扱うメタ情報を示す．

メタ情報	説明	例
センサが存在する空間	アプリケーションがセンサデータを要求する空間	椅子の周り，ユーザの周り，部屋の中，共有スペース
センサが付属する物	アプリケーションがセンサデータを要求する物	椅子，ベッド，ドア，ユーザ
センサの位置	センサの存在する空間や付属する物を割り出すための座標	X，Y，Z 座標
センサの種類	アプリケーションが取得したいセンサデータの種類	温度，圧力，明るさ，加速度
データ形式	分解能やサンプリングレートなどのセンシングの形式	8byte，0.1/sec

表 3.1: センサのメタ情報

第 4 章

設計

本章ではセンサのメタ情報を利用して環境に設置されたセンサからセンサデータを取得するミドルウェアである MARS の設計を行う.

4.1 設計概要

本研究では，環境に分散配置されたセンサからアプリケーションの要求するセンサデータを取得するミドルウェア MARS を構築する．本章では，MARS の設計概要として，MARS の設計方針，MARS の想定するハードウェア構成，MARS のソフトウェア構成について述べる．

4.1.1 設計方針

ユビキタスコンピューティング環境において，アプリケーションが動的なセンサ環境に対応してセンサデータの取得を行うために，以下の要件を満たすように MARS の設計を行なった．

- **センシングの対象物の指定**
  センサはアプリケーションの要求するセンシングの対象物に関する物理量や対象物の周囲の物理量，また，室内などの空間の物理量を取得する．センシングの対象物とセンサを同じ基準の位置情報で扱うために，MARS はセンサの位置情報や種類などのメタ情報を管理するメタ情報データベースを持つ．

- **センサの多様性の吸収**
  ユビキタスコンピューティング環境には様々なセンサが存在する．アプリケーションが要求するセンサデータを取得するために，センサのメタ情報を利用した要求を受け付ける．MARS はセンサの能力などの多様性を吸収し，アプリケーションにセンサデータを提供する．

- **センサデータの加工**
  アプリケーションの要求するセンサデータは，どのセンサで取得されたセンサデータかということは重要ではないため，全てのセンサデータをそのまま提供する必要があるとは限らない．センサで取得したセンサデータは，アプリケーションの要求によって平均などの加工をして提供することが必要である．MARS はアプリケーションの要求に従ってセンサデータを加工し提供する．

4.1.2 ハードウェア構成

MARS が想定するハードウェア構成を図 4.1 に示し，各ハードウェアの説明を行う．

アプリケーションホスト

アプリケーションホストでは，センサデータを利用するアプリケーションが動作するホストコンピュータである．アプリケーションは MARS からセンサデータを取得し，取得したセンサデータを利用して機器を操作する．

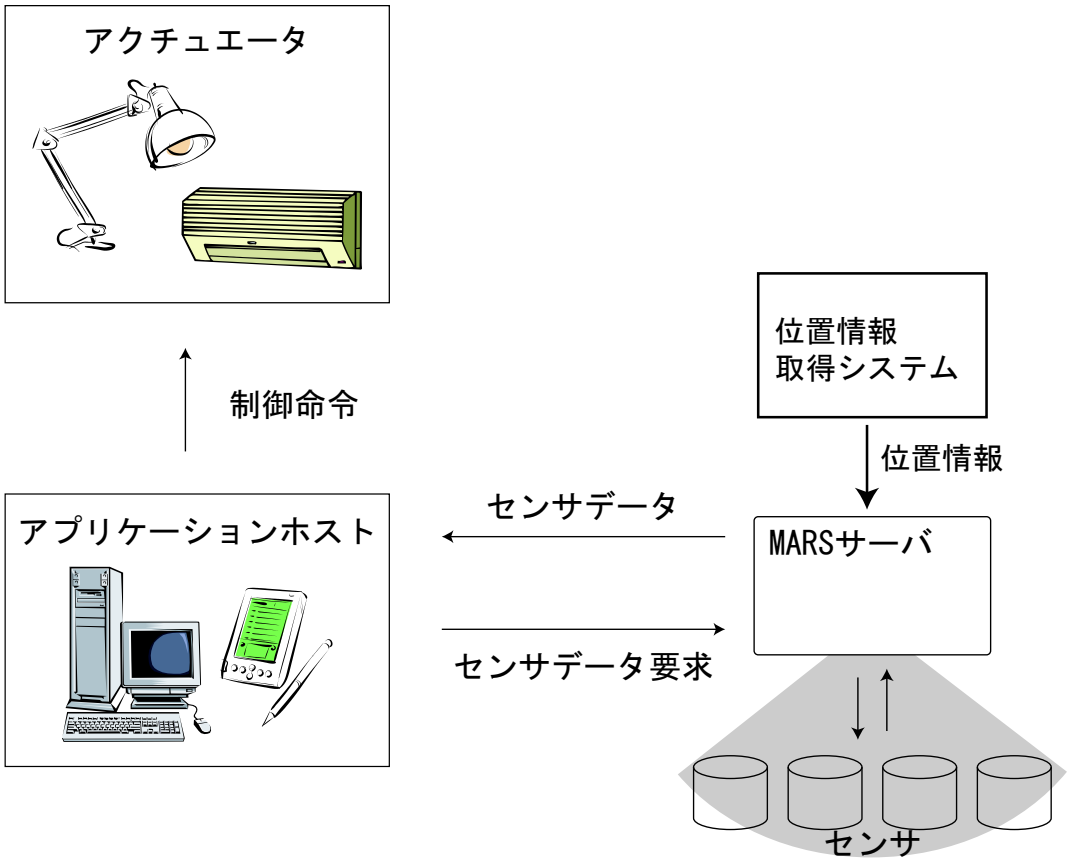

図 4.1: MARS ハードウェア構成

アクチュエータ

アクチュエータは情報家電などのネットワーク経由で操作可能な機器である．アクチュエータは外部のアプリケーションホスト上で動作するアプリケーションによって操作される．

MARS サーバ

MARS サーバはアプリケーションの要求を受け付け，対象となるセンサにクエリを送信する．センサからセンサデータを受信し，アプリケーションに提供する．

センサ

MARS サーバのクエリにしたがってセンサデータを提供する．MARS サーバのクエリを解釈するため，センサデータを加工するために計算機能を有する．センサは無線通信機能を有する．

位置情報取得システム

センシングの対象物とセンサの座標を取得する．MARS サーバの要求に従って，位置情報を提供する．

4.1.3 ソフトウェア構成

MARS サーバはメタ情報管理機構，アプリケーション要求受信機構，アプリケーション要求解決機構，センサデータ提供機構，メタ情報管理データベース，および，センサとの通信を行うゲートウェイノードで構成される．また，MARS サーバは外部の位置情報取得システムを利用する．図 4.2 に，MARS のソフトウェア構成を示し，次節以降で各部の説明を行う．

4.2 メタ情報管理機構

- **メタ情報の登録**
  センサは起動時に MARS サーバに対して自身のメタ情報を送信する．また，メタ情報管理機構は各センサの位置情報を外部の位置情報取得システムから取得する．メタ情報管理機構では，センサから受信したメタ情報や外部位置情報取得システムから取得したセンサの位置をメタ情報データベースに登録する．

- **センサ環境の変化への対応**
  センサのメタ情報には，センサの位置情報のように，時間の経過とともに変化するものが存在する．メタ情報管理機構では，外部位置情報取得システムから定期的にセンサの位置情報を取得し，メタ情報データベースを更新する．

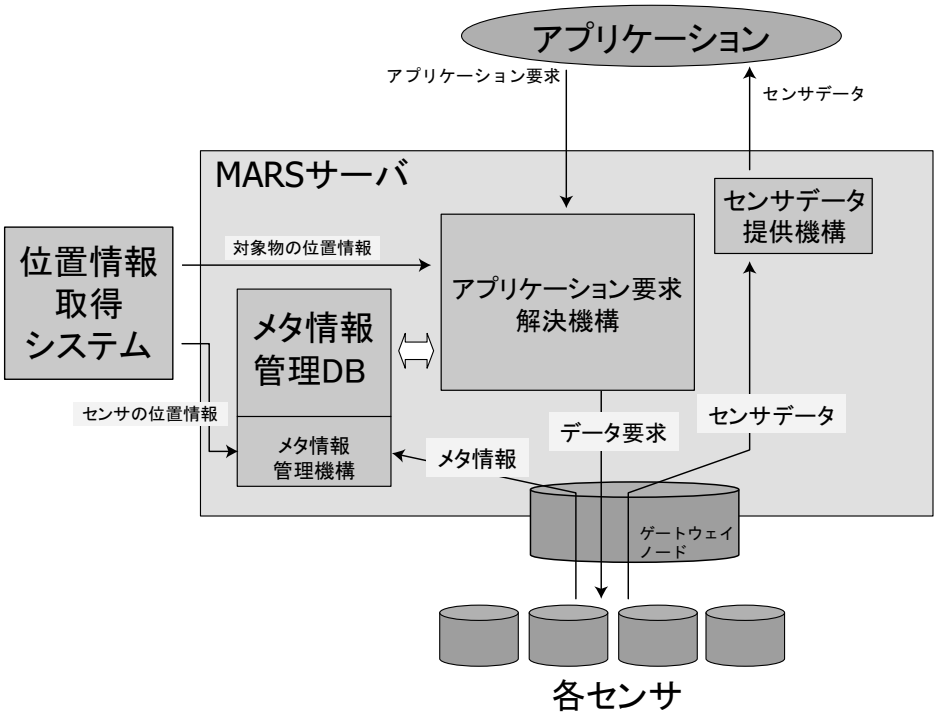

図 4.2: MARS ソフトウェア構成

4.3 アプリケーション要求解決機構

アプリケーション要求解決機構では，アプリケーションから対象物およびセンサの種類による要求を受け付ける．アプリケーション要求解決機構では，アプリケーションの要求に該当するセンサのノード ID をメタ情報データベースから検索する．また，外部の位置情報取得システムを利用して，アプリケーションが要求するセンシングの対象物の位置を取得し，要求に対応するセンサを特定する．アプリケーション要求解決機構は，ゲートウェイノードを介して対象となるセンサに対して，クエリを送信する．

4.4 センサデータ提供機構

ゲートウェイノードがセンサから受信したセンサデータはセンサデータ提供機構に渡される．センサデータ提供機構では，受信したセンサデータをアプリケーションの要求する形式に変換してアプリケーションに提供する．例えば，センサデータを平均して提供することなどが考えられる．センサデータの変換手法は，あらかじめ API として平均や合計などを用意する．

4.5　外部位置情報取得システム

MARS では，アプリケーションが要求するセンシングの対象物とセンサを同じ基準の位置情報で扱う必要がある．このため，MARS 外部の位置情報取得システムを利用する．MARS では位置情報取得システムは，１つのホストに対して問い合わせることで全ての位置情報を取得する事ができる，送信機タグ型位置情報取得システムを利用する．また，MARS は外部の位置情報取得システムを利用して，センシングの対象物である機器の位置を取得する．

4.6　センサ上で動作するソフトウェア

MARS が利用するセンサは，各センサ上でソフトウェアが動作し，無線通信によって MARS サーバと通信を行う．図 4.3 にセンサ上で動作するソフトウェアの構成を示す．

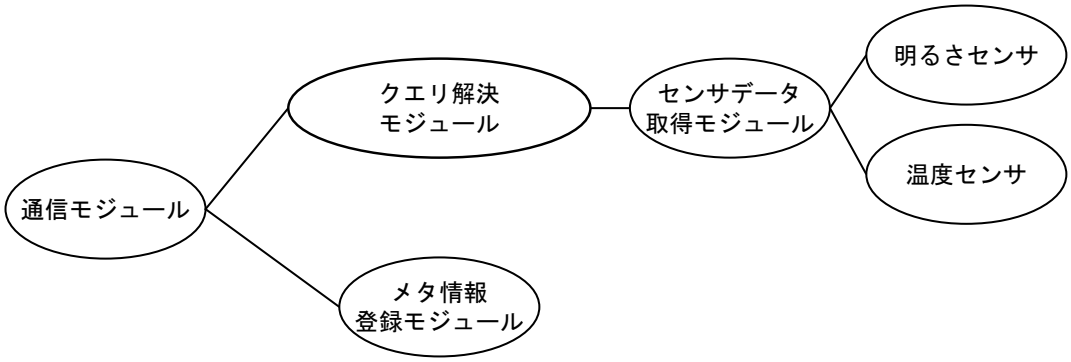

図 4.3: センサ上で動作するソフトウェアのコンポーネント関係図

センサを起動するとメタ情報登録機構はメタ情報をゲートウェイノードに送信する．また，バッテリ残量のように，時間の経過とともに変化するメタ情報は，定期的に送信する，閾値を超えた時に送信するなどして MARS サーバのメタ情報を更新する．MARS サーバからクエリを受信しすると，クエリ解決機構はアプリケーションの要求を解決し，センサデータ取得機構からセンサデータを取得する．センサデータ取得機構では，センサや A/D コンバータを利用してセンサデータを取得する．

4.7　本章のまとめ

本章では，MARS サーバの設計方針として，メタ情報を利用してアプリケーションの要求を解決するためにメタ情報を管理するデータベースを持つことを述べた．次に，本システムの概要について述べ，各機構の設計について詳細に説明した．次章では，MARS の実装について述べる．

第 5 章

実装

前章では，メタ情報を利用してセンサデータを取得するMARSの設計について述べた．本章ではMARSの実装について述べる．

5.1 実装環境

本システムの MARS サーバは表 5.1 に示す環境で実装した．本システムは様々な環境で，環境に設置されたセンサからセンサデータを取得するため，本システムが動作するハードウェアや OS も多様であると考えられる．そこで，システムのプラットフォーム独立性を確保するために，Java 言語を用いて実装した．

表 5.1: 実装環境

項目	環境
CPU	AMD Athlon(TM) XP 1800+ (1533.33MHz)
Memory	1024MB
OS	RedHat Linux 9
JDK	Java 2 SDK, Standard Edition Version Version 1.4.1
DataBase	PostgreSQL 7.3.4

また，本システムではセンサとして，UC Berkeley で研究，開発が行われている mica2 を利用し，温度と照度を取得した．mica2 は ATMEL 社の 8-bitRISC マイコンである ATmega128[20] による計算能力および Chipcon 社の無線送受信 IC である C1000[21] による無線通信能力を有し，本研究におけるセンサの想定を満たす．mica2 の仕様を表 5.1 にまとめた．

表 5.2: mica2 の仕様

CPU	ATmega128 7.4MHz
プログラムメモリ	128KB Flash
データメモリ	512KB Flash
AD Converter	10 bit x 6 ch
無線周波数	315MHz FSK
電流消費 (受信時)	8〜10mA 3Vdc
電流消費 (sleep)	<15 μ A 3Vdc
外形寸法 MICA	55 x 32 x 25 mm
ソフトウェア	TinyOS Nes-C
標準付属センサ	音/光/温度/加速度/磁気

mica2 は 128KB のプログラムメモリにソフトウェアを書き込んで動作させる．mica2 上で動作するソフトウェアは nesc を用いて実装した．nesc は TinyOS Project で開発されている，mote で動作するソフトウェアを記述するオブジェクト指向言語である．ま

た，mica2 の無線通信は 1 チャンネルの半二重通信であるため，送受信を同時に行うことはできない．

5.2 MARS サーバの実装

本節では，MARS サーバの実装について述べる．

5.2.1 メタ情報管理機構

本システムは，センサから受信したメタ情報をメタ情報データベースに登録する．メタ情報データベースのテーブルには，ノード ID，センサの種類，座標，付属物のデバイス ID という項目を用意した．メタ情報管理機構の実装として、MetaDataRegist クラスを図 5.1 に示す．

5.2.2 アプリケーション要求解決機構

GetData クラスは，対象物や空間に関する要求を受け付け，センサにクエリを送信し，取得したセンサデータを返す．以下に，GetData クラスのメソッドを示す．

double getRoundDeviceData(String sensor_type, int device_id, int radius)

対象物の ID から，getDeviceLocation() を利用して対象物の位置を取得する．センサの種類と対象物の位置から，getSphereData() を利用して対象物の周囲のセンサデータを取得する．

double getSphereData(String sensor_type, int center[], int radius)

センサの種類と中心，半径から，球形空間の内側のセンサデータを取得する．

double getCubeData(String sensor_type, int start_point[]), int end_point[]

センサの種類と始点，終点から，位置取得システムの座標軸に平行の立方体の内側のセンサデータを取得する．

Vector getNodeId(String sensor_type, int min, int max)

センサの種類と，センサデータの下限，上限から，一定の範囲のセンサデータを示すセンサのノード ID を取得する．

double getAllData(String sensor_type)

一種類のセンサに関して，室内全てのセンサからセンサデータを取得する．

```
public class MetaDataRegist{
    //センサから受信したメタ情報 MessageFromNode とメタ情報データベースが動作するホスト名から，メタ情報を登録する
    public static void regist(MessageFromNode am_mes, String host){
        int nodeadder; //センサのノード ID
        int stype; //センサの種類

        //ノード ID を取得
        nodeadder = am_mes.payload[0]+am_mes.payload[1]*256;
        //センサの種類を取得
        stype = am_mes.payload[2];
        getSensorType(stype);
        //センサの位置を取得
        getSensorLocation(nodeadder);

        // データベースへの接続
        db_access_init();
        // 同一のセンサを一度削除する SQL 文を作成
        String sql = "DELETE FROM MARS_NODE " +
                        "WHERE NODEADDER = "+ nodeadder + " AND STYPE = " + stype;
        // クエリを実行
        int result = stmt.executeUpdate(sql);

        // NODEADDER が am_mes.nodeadder の行を追加する SQL 文を作成
        sql = "INSERT INTO MARS_NODE " +
                "values("+nodeadder+
                ",'"+sensorname+
                "',"+nodelocation[0] +
                ","+nodelocation[1] +
                ","+nodelocation[2] + ")";
        // クエリーを実行
        result = stmt.executeUpdate(sql);

        //終了処理
        db_access_close();
        return;
    }

    //stype から，データベースに登録するセンサ名を取得するメソッド
    public static void getSensorType(int stype){
      //省略
      ....
    }
}
```

図 5.1: MetaDataRegist クラス

5.2.3 センサデータ提供機構

本論文における実装では，対象物の周囲のセンサで取得したセンサデータを，アプリケーションの要求に従って，平均値や最大値，最小値に加工して提供した．

5.3 センサの実装

センサは，ゲートウェイノードから受信したセンシング命令に従い，センサデータをゲートウェイノードに送信する．

メタ情報登録メッセージ

destination address (4byte)	handlerID (2byte)	gourpID (2byte)	message length (2byte)	source address (4byte)	metadata type (2byte)	metadata (4byte)

クエリメッセージ

destination address (4byte)	handlerID (2byte)	gourpID (2byte)	message length (2byte)	source address (4byte)	command (2byte)

センサデータメッセージ

destination address (4byte)	handlerID (2byte)	gourpID (2byte)	message length (2byte)	source address (4byte)	value (2byte)

図 5.2: センサ間通信の形式

センサとゲートウェイノード間の通信の形式を，図 5.2 に示す．センサは 4byte のノード ID および，2byte のグループ ID を持つ．handlerID はメッセージの種類を示し，メタ情報登録メッセージを 0，クエリメッセージを 1，センサデータメッセージを 2 とした．metadata type はメタ情報の種類を示し，metadata は具体的なメタ情報を示す．例えばセンサの種類というメタ情報は metadata type を 0 とし，温度センサは metadata を 0，明るさセンサは metadeta を 1 とした．command はセンサデータ取得命令を示すし，温度の取得を 0，明るさの取得を 1 とした．

通信モジュールからクエリを受信すると，クエリ解決モジュールがセンサデータ取得モジュールに対してセンサデータ取得命令を行う．センサデータ取得モジュールは A/D コンバータの値を読む．図 5.3 に，クエリ解決モジュールとセンサデータ取得モ

```
//クエリを受信したことにより発生する割り込み処理
  task void cmdInterpret() {
    //受信したクエリ
    struct SimpleCmdMsg * cmd = (struct SimpleCmdMsg *) msg->data;

    //センサデータの取得をセンサデータ取得モジュールに命令する
    switch (cmd->action) {
      case PHOTO_SENSING:
        //LED による表示
call Leds.greenToggle();
       //照度取得イベントを発生させる
        signal PhotoComd.fire();
      case TEMP_SENSING:
        call Leds.redToggle();
        signal TempComd.fire();
    }
    //割り込み終了を通知
    signal ProcessCmd.done(msg, SUCCESS);
  }

  //照度取得イベント
  event result_t PhotoComd.fire() {
    //照度センサが接続されている A/D コンバータの値を読み込む
    call P-ADC.getData();
    return SUCCESS;
  }

  event result_t TempComd.fire(){
    call T-ADC.getData();
    return SUCCESS;
  }
```

図 5.3: クエリ解決モジュールおよびセンサデータ取得モジュール

ジュールの概要を示す.

5.4 アプリケーションによる利用方法

本研究で実装した MARS は慶應義塾大学 SFC Open Research Forum 2003[22] において，u-Photo[23] のデモンストレーションで使用した．u-Photo は,“ 撮る ”という行為によって，画像だけでなく，環境の情報も取得できる写真である．u-Photo が取得する情報は機器の作業状態や利用できるサービス，機器周辺のセンサデータなどである．ORF2003 では，u-Photo がセンサデータを取得する際の，センサデータ取得システムとして MARS を利用した．u-Photo は u-PhotoCreator によって生成される．図 5.4 に，u-PhotoCreator からセンサデータの要求を受信してから，センサデータを提供するまでの MARS の動作を示す.

1. アプリケーション要求解決機構は，u-PhotoCreator からセンサの種類，センシン

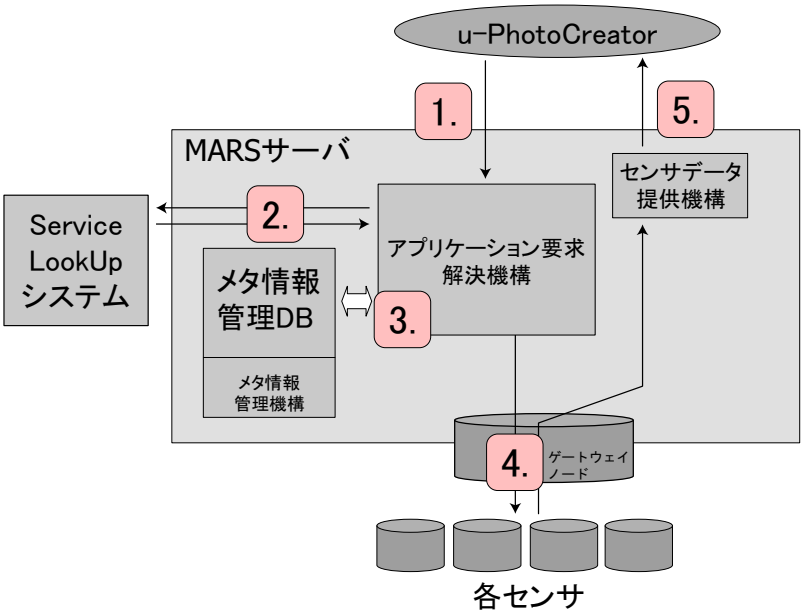

図 5.4: u-Photo における MARS の動作

グの対象となるデバイスの ID を指定した要求を受信する．

2. u-Photo が作業状態などの情報を取得する対象となる機器を ServiceLookUp システムで管理した．アプリケーション要求解決機構は，デバイス ID を元に ServiceLookUP システムから各デバイスの座標を取得する．

3. アプリケーション要求解決機構は，デバイスの座標，センサの種類，半径を元に，メタ情報データベースから対象となるセンサのノード ID を取得する．u-PhotoCreator の要求に従い，機器の周囲を半径 1 メートルの範囲のセンサからセンサデータを取得した．

4. アプリケーション解決機構は，取得したノード ID のセンサに対してセンサの種類を指定したクエリを送信する．クエリを受信したセンサは，センサデータを取得し，センサデータ提供機構に送信する．

5. u-PhotoCreator の要求に従い，取得したセンサデータをセンサデータ提供機構で平均し，u-PhotoCreator に提供した．

図 5.5 に，u-Photo を操作する様子を示す．

図 5.6 に MARS を利用して取得したセンサデータが u-Photo で表示されている様子を示す．u-Photo では，写真に写っている部屋全体の温度情報，照度情報と，各デバイスの周囲の温度情報，照度情報を MARS を利用して取得した．

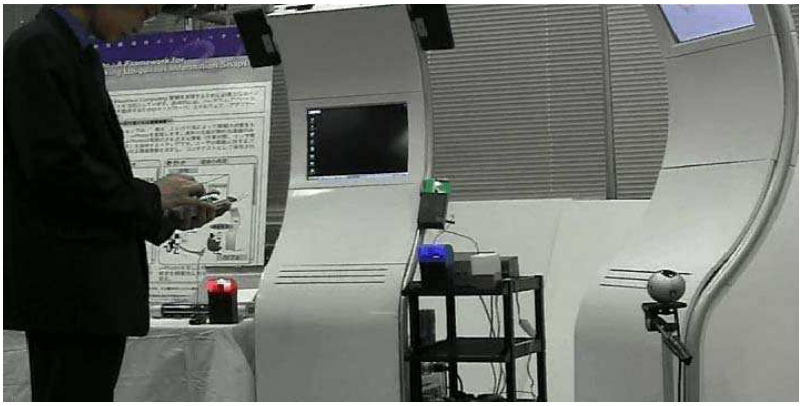

図 5.5: u-Photo を操作する様子

部屋全体の環境情報の取得　　機器の周囲の環境情報の取得

図 5.6: MARS で取得されたセンサデータの表示

5.5　本章のまとめ

本章では，MARS のプロトタイプ実装について述べた．実装環境，実装手法を示し，実装例として ORF2003 で u-Photo での利用方法について示した．次章では，MARS の評価を行う．

第 6 章

評価

前章ではメタ情報を利用してセンサデータを取得する MARS の実装について述べた．本章では MARS の評価を行う．

6.1 定量的評価

本節では，MARS の定量的評価を行なう．MARS がアプリケーションから要求を受信してから，センサデータを提供するまでの時間を計測した．センサが 1 個から 8 個まで増えた時にセンサデータを提供するまでの時間がどのように増加するかを観察した．計測の結果を図 6.1 に示す．

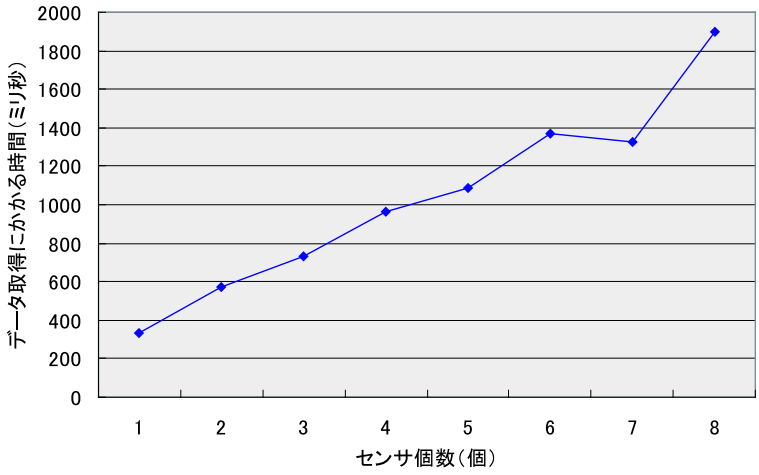

図 6.1: 測定結果

本測定により，センサの個数が増加するに連れて，センサデータを提供するまでの時間が増加していることがわかる．平均 2 秒以内で，8 個のセンサからセンサデータを取得し，アプリケーションに提供できた．ユーザの周囲の温度情報を利用した空調や，ユーザが睡眠中であるというコンテキストを抽出し，利用するようなアプリケーションにおいては，実用的に利用可能である．

6.2 本章のまとめ

本章では，MARS の評価として，アプリケーションの要求を解決し，センサデータを提供する時間を計測した．

第 7 章

結論

本研究では，センサのメタ情報を利用し，センシングの対象物を指定したセンサデータの取得を行うミドルウェア MARS を設計し実装した．本章では本研究の今後の課題を示し，最後に本論文のまとめを行う．

7.1 今後の展望

本節では，MARS の今後の課題について述べる．

7.1.1 スケーラビリティ

本論文で行なった実装では，センサを８個しか使用しなかった．その理由としては，現在は通信機能や計算機能を持つセンサが高価で大量に配置できないことや，mica2 がセンサプラットフォームの研究用プロトタイプであるため，１つ１つのセンサにソフトウェアを書き込む必要があり，大量消費の障害となっていることが挙げられる．今後センサの価格が下がり，センサを分散配置することが実用化されると，１つのセンサデータ取得システムで１００個や１０００個以上のセンサを扱うことも考えられる．大量のセンサからセンサデータを取得する場合に，アプリケーション要求の解決やセンサデータの取得にかかる時間がアプリケーションの動作に支障をきたさないように設計，実装を工夫しなければならない．

また，MARS は屋内でセンサ間の無線通信が１ホップで到達することを想定しているため，現状では各部屋毎に MARS サーバが設置される必要がある．各部屋に設置されている MARS サーバを横断的に利用することや，各部屋にはゲートウェイノードのみを設置し，MARS サーバは階や建物に毎に１つ設置することなどの考察を行う．

7.1.2 センサデータの加工

本論文で行なった実装では，センサデータは平均，最大値，最小値をアプリケーションに提供することが可能である．しかし，アプリケーションの要求するセンサデータの加工は，平均や最大値，最小値だけではなく，極端なエラー値を除去して平均するなどが考えられる．また，アプリケーションの要求によっては，各センサで取得したセンサデータとセンサの位置情報により，環境情報のマップを生成するように，全てのセンサデータを加工せずに提供する必要がある．今後，実際にコンテキストアウェアアプリケーションや環境モニタリングアプリケーションにより MARS を利用し，アプリケーションの要求するセンサデータの加工方法を考察する．

7.1.3 多様なセンサプラットフォームへの対応

本論文では，mica2 を用いて実装を行なった．しかし，実際の環境には，機器の製作時に各メーカで埋め込まれるセンサなど，通信形態や計算能力が異なる様々なセンサが存在すると考えられる．このため，異なるプラットフォームのセンサが混在する環境でも，同様にアプリケーションの要求するセンサデータを取得することが必要である．

7.2 本論文のまとめ

本論文では，センサのメタ情報を利用してセンサデータを取得するミドルウェアMARSを設計，実装し，評価を行なった．本システムを利用することで，アプリケーションは環境に設置されたセンサからコンテキストを抽出する対象物やアクチュエータにより制御する対象物の周囲に存在するセンサや対象物に付属するセンサからセンサデータを取得できる．

近年のコンピュータの小型化と処理能力の向上や通信技術の進歩により，あらゆる機器にセンサやコンピュータを埋め込み，ユーザの作業を支援するユビキタスコンピューティング環境の実現が可能となった．ユビキタスコンピューティング環境では，コンテキストアウェアアプリケーションが，コンテキストを抽出するために，対象物に関するセンサデータを取得する事が必要である．様々なコンテキストアウェアアプリケーションが，様々な対象物のコンテキストを抽出するため，ユビキタスコンピューティング環境では，アプリケーションとセンサが動的に接続する．このため，多様なセンサからアプリケーションの要求するセンサデータを取得するためのミドルウェアが重要となる．本研究では，センサのメタ情報を管理することにより，アプリケーションの要求を解決し，センサデータを提供できた．

謝辞

本研究を進めるにあたり，ご指導を頂きました，慶應義塾大学環境情報学部教授徳田英幸博士に深く感謝致します．

慶應義塾大学徳田，村井，楠本，中村，南研究会の諸先輩方には，お忙しい中貴重な示唆や御助言，御指導を頂きました．特に，徳田研究室の先生方や先輩方，HORN研究グループの方々に深く感謝いたします．また，慶應義塾大学政策・メディア研究科後期博士課程3年の岩本健嗣氏，同政策・メディア研究科後期博士課程1年の神武直彦氏，同政策・メディア研究科修士課程1年の青木俊氏，同鈴木源太氏には，本論文執筆にあたって励ましと御指導を頂きました．ここに深い感謝の念を表します．

最後に，研究の日々を共に過ごした，幸田拓耶氏，出内将夫氏，佐川昭宏氏その他多くの友人に深く感謝し，謝辞と致します．

平成15年 12月 29日
丸山 大佑

参考文献

[1] Samuel Maddern, Michael J. Franklin, J. M. H. and Hong, W.: The Design of an Acquisitional Query Processor For Sensor Networks, *SIGMOD* (2003).

[2] Maddern, S.: Query Processing for Streaming Sensor Data, *Ph.D. Qualifying Exam Proposal* (2002).

[3] Weiser, M.: The Computer for the Twenty-First Century, *Scientific American*, pp. 94–10 (1991).

[4] Takeshi Iwamoto, Nobuhiko Nishio, H. T.: Wapplet: A Media Access Framework for Wearable Applications, *International Conference on Information Networking*, Vol. II.

[5] 桐原幸彦, 由良淳一, 楠本晶彦, 徳田英幸: 情報家電機器におけるプロアクティブな利用支援機構, 情報処理学会システムソフトウェアとオペレーティングシステム研究会 (OS 研), pp. 105–112 (2001).

[6] Chen, G. and Kotz, D.: A Survey of Context-Aware Mobile Computing Research, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College (2000).

[7] 東芝: 東芝ネットワーク家電 フェミニティ IT 冷蔵庫. http://feminity.toshiba.co.jp/feminity/series/refrigerator/index.html.

[8] SONY: プラズマテレビ WEGA KDE-P61HX2N. http://www.ecat.sony.co.jp/wega/products/index.cfm?PD=15220&KM=KDE-P61HX2N.

[9] 青木崇行, 村瀬正名, 松宮健太, 中澤仁, 西尾信彦, 高汐一紀, 徳田英幸: Smart Furniture: Improvising Ubiquitous Hot-spot Environment (2002).

[10] Itiro Siio, J. R. and Mynatt, E.: Peek-A-Drawer: Communication by Furniture, *Conference Extended Abstracts on Human Factors in Computer Systems*, pp. 582–583 (2002).

[11] HITACHI: μ-chip The World's Smallest RFID IC. http://www.hitachi.co.jp/Prod/mu-chip/index.html.

[12] 岩崎弾: 知的環境を支援する情報収集型デバイスの設計と実装 (2002).

[13] J. Hill, R.Szewczyk, A. W. S. H. and Pister, D. C. K.: System architecture directions for networked sensors, *ASPLOS 2000*, pp. 93–104 (2000).

[14] Joshua Lifton, Deva Seetharam, M. B. and Paradiso, J.: Pushpin Computing System Overview: A Platform for Distributed, Embedded, Ubiquitous Sensor Networks, *Pervasive 2002*, pp. 139–151 (2002).

[15] J. M. Kahn, R. H. K. and Pister, K. S. J.: Mobile Networking for Smart Dust, *ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom 99)*, pp. 17–19 (1999).

[16] L.E. Holmquist, F. Mattern, B. S. P. A. M. B. and Gellersen, H.: Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts, *Proc. of UBICOMP 2001* (2001).

[17] Hill, J. and Culler, D.: A wireless embedded sensor archtecture for system-level optimization, *In UC Berkeley Technical Report*.

[18] Samuel Maddern, Michael J. Franklin, J. M. H. and Hong, W.: TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks, *Operating Systems Design and Implementation* (2002).

[19] Philippe Bonnet, J. E. G. and Seshadri, P.: Querying the Physical World, *IEEE Personal Communications*, Vol. 7, pp. 10–15 (2000). Special Issue on Smart Spaces and Environments.

[20] AtmelCorporation: 8-bit RISC ATmega128. http://www.atmel.com/dyn/products/product_card.asp?part_id=2018.

[21] Chipcon: RF-IC C1000. http://www.chipcon.com/index.cfm?kat_id=2&subkat_id=12&dok_id=14.

[22] 慶應義塾大学: SFC Open Research Forum 2003. http://orf.sfc.keio.ac.jp/.

[23] 鈴木源太, 岩本健嗣, 神武直彦, 青木俊, 丸山大佑, 幸田拓耶, 高汐一紀, 徳田英幸: u-Photo: 家電機器操作や環境情報取得を直感的に実現するユビキタス情報スナップショット (2004).