

卒業論文

2002年度(平成14年度)

ヘテロジニアスなセンサ環境における
位置取得システムの構築

指導教員

慶應義塾大学環境情報学部

徳田 英幸

村井 純

楠本 博之

中村 修

南 政樹

慶應義塾大学 環境情報学部 環境情報学科

氏名：青木 俊

ヘテロジニアスなセンサ環境における 位置取得システムの構築

論文要旨

近年、計算機の小型化が進み、ラップトップのような形だけではなく、家電などへの組み込みや、計算機とセンサ等との組み合わせによって独自の機能を持たせるなど、形を変えて存在するようになっている。それらの機器を接続することによって、実世界の状況の変化を取得し、状況に応じて様々なサービスを提供する環境が可能となり、このような環境をユビキタス情報環境と呼ぶ。ユビキタス情報環境構築の研究は多くなされているが、その中で、情報世界と物理世界のマッピングは大きな課題の一つである。例えばユーザや物の物理的な位置情報を取り扱うアプリケーションの実現はユビキタス情報環境に不可欠である。

本論文ではユビキタス情報環境の中で、ヘテロジニアスなセンサ群の組み合わせによる位置取得システムを提案する。今後構築されるユビキタス情報環境では、さまざまな種類のセンサが混在すると考えられ、これは位置を取得可能なセンサにおいても例外ではない。しかし、数種類のセンサを包括的に扱えるシステムは少ない。また、アプリケーションにおいては、位置情報の要求方法、取得方法において多様であるといえるが、現在個々のアプリケーションは環境に合わせて位置情報の要求、取得を行っている。本研究では様々なセンサが混在可能な位置取得システムを構築し、環境に非依存なアプリケーションの位置要求、取得を可能にするとともに、位置情報が加わることによって現実世界におけるユーザの行動を支援可能なアプリケーションの容易な実現を目的とする。

キーワード：

1 ユビキタス情報環境 2 ミドルウェア 3 位置取得システム 4 多様性 5 抽象化

慶應義塾大学 環境情報学部 環境情報学科
青木 俊

Abstract of Bachelor's Thesis

Location System under Heterogeneous Sensor Environment

Academic Year 2002

Summary

Under ubiquitous computing environment , many applications need to obtain user and device location. For example , when a user moves from a living room to a kitchen , an application must change services to continuously support the user's task such as watching TV.

Many location systems which currently exist realizes high precision. However , when an application which is build for a specific location system moves to another environment that deploys a different location system, the application cannot be used. This is because location systems , which are designed for specific sensors or environments , cannot provide appropriate location information. Moreover , many applications do not need high precision.

This system provides applications with location information to that does not depend on the environment. This system hides the diversity of the sensors by abstracting sensor data, and providing a unified API.

Keywords:

1 ubiquitous computing environment 2 middleware 3 location system 4 diversity 5 abstraction

Keio University Faculty of Environmental Information
AOKI SHUN

目次

第1章 序論	1
1.1 本研究の背景	2
1.1.1 ユビキタス情報環境と位置取得システム	2
1.2 本研究の目的	2
1.3 本論文の構成	3
第2章 ユビキタス情報環境	4
2.1 ユビキタス情報環境	5
2.1.1 現在のコンピューティング環境	5
2.1.2 ユビキタス情報環境の定義	5
2.1.3 情報世界による物理世界の把握	6
2.2 ユビキタス情報環境の構成要素	6
2.2.1 物理世界における構成要素	7
2.2.2 情報世界における構成要素	8
2.3 ユビキタス情報環境におけるミドルウェアの役割	10
2.4 ユビキタス情報環境における位置情報	11
2.4.1 位置情報の重要性	11
2.4.2 既存の位置情報ミドルウェア	11
2.4.3 ユビキタス情報環境における位置情報ミドルウェア	11
2.5 本章のまとめ	12
第3章 研究の方針と概要	13
3.1 本研究の概要	14
3.1.1 想定環境	14
3.1.2 研究概要	14
3.2 多様性の考察とその解決手法	15
3.2.1 位置情報における多様性	15
3.2.2 多様性への解決手法としての抽象化	16
3.3 3次元空間へのマッピングによる抽象化手法	17
3.3.1 解決手法	17
3.3.2 抽象化モデル	17

3.4	想定シナリオ	18
3.4.1	パーソナルシアター	18
3.4.2	動的ライト ON/OFF アプリケーション	19
3.4.3	仮想3次元空間における遠隔地制御アプリケーション	20
3.5	本章のまとめ	21
第4章 設計		22
4.1	OASISS の設計方針	23
4.1.1	必要機能	23
4.1.2	ソフトウェア構成	23
4.2	空間情報取得部の設計	25
4.2.1	空間情報の取得	25
4.2.2	空間情報の提供	25
4.3	アプリケーションインタフェースの設計	26
4.3.1	アプリケーション要求空間の構築	26
4.3.2	アプリケーション要求空間の転送	27
4.4	センシングモジュールインタフェースの設計	27
4.4.1	データ抽象化空間の構築方法	27
4.4.2	センシングモジュール固有の情報	28
4.4.3	センシングモジュールの範囲の決定	28
4.4.4	基本立体の生成	28
4.4.5	基本立体の配置	28
4.5	空間合成モジュールの設計	30
4.5.1	空間の取得及び合成	30
4.5.2	位置情報提供リージョンの決定	30
4.6	シナリオ	30
4.7	設計手法の考察	33
4.8	本章のまとめ	33
第5章 OASISS の実装		34
5.1	実装方針	35
5.2	実装環境	35
5.3	オブジェクト動作	35
5.4	アプリケーション要求空間の実装	36
5.5	データ抽象化空間の実装	36
5.6	空間合成の実装	37
5.7	アプリケーションの利用方法	37
5.8	本章のまとめ	40

第 6 章 評価	41
6.1 関連研究との比較	42
6.1.1 The Active Bat Location System	42
6.1.2 Cricket	42
6.1.3 POIX	42
6.2 基本性能の測定	43
6.2.1 測定環境	43
6.2.2 センサ数とアプリケーション粒度における合成判定時間	43
6.3 本章のまとめ	44
第 7 章 結論	45
7.1 今後の課題	46
7.1.1 ID の管理	46
7.1.2 空間の協調	46
7.1.3 アプリケーションに対する多様な API の提供	46
7.2 本論文のまとめ	46
参考文献	49

目 次

1.1	ヘテロジニアスなセンサ環境	3
2.1	情報世界と物理世界の構成要素	7
3.1	抽象化概念図	18
3.2	パーソナルシアター	19
3.3	遠隔地制御アプリケーション	20
4.1	各モジュールの関係	24
4.2	ベース直方体	25
4.3	アプリケーション座標系	27
4.4	センシングモジュールの範囲の形状	29
4.5	基本立体の生成	29
4.6	データ抽象化空間とアプリケーション要求空間の変換	31
4.7	新たなアプリケーションの追加	32
4.8	新たなセンシングモジュールの追加	33
5.1	空間情報を定義する XML ファイル	36
5.2	オブジェクト関係図	37
5.3	合成作業の GUI	38
5.4	アプリケーション実装例	39

表 目 次

5.1	実装環境	35
6.1	測定環境	43
6.2	測定結果 (ms)	44

第 1 章

序論

本章では，本研究の背景であるユビキタス情報環境と位置取得システムについて述べる．次に本研究の目的であるヘテロジニアスなセンサ環境に適した位置取得システムについて述べる．最後に本論文の構成について述べる．

1.1 本研究の背景

1.1.1 ユビキタス情報環境と位置取得システム

近年の計算機の小型化，ワイヤレス技術の発展にともない，あらゆる場所に埋め込まれた計算機やセンサがユーザを支援するユビキタス情報環境が注目を浴び，その環境を前提とした研究も多くなされている。

例えば，ネットワークに接続されたインターネット冷蔵庫や電子レンジなど，情報家電と呼ばれる機器が注目されている。これらは今までは単体で動作していたが，ネットワークに接続されることによって冷暖房を家の外から制御するなどの新たな機能が追加され，ユーザを支援できるようになった。

ユビキタス情報環境の大きな特徴として，空間に PC やセンサが埋め込まれており，ユーザやアクチュエータが小型 PC やセンサを持っていることが挙げられる。この特徴を生かすために，今まで独立して動作していたデバイス同士が結びつくことによる協調動作や，異なるデバイスによるサービスの継続性など新たな機能が考えられる。しかし，各空間を構成する機器やセンサ，ユーザが保持する小型 PC やセンサはそれぞれ異なる。また，各空間内においても様々なデバイスが混在する。よってそれらの違いを吸収し，統一した利用方法を提供するミドルウェアが必要になり，その技術がユビキタス情報環境を支える基盤となる。

ユビキタス情報環境を支えるミドルウェアは目的に応じて分類され，その中に位置情報ミドルウェアが存在する。ユビキタス情報環境では，位置情報を必要とするサービスは増えると予想される。例えば，ユーザの現在の位置やデバイスの位置関係に依存したサービスが考えられる。ユーザの周囲に存在するサービスの検索や位置情報取得のためのセンサシステムはユビキタス情報環境実現のために重要となる。

1.2 本研究の目的

ユビキタス情報環境に適した位置取得システムは，環境内の多様なセンサを利用し，アプリケーションに対して位置情報提供インタフェースを提供する必要がある。本研究では，ユビキタス情報環境の特徴である機器の混在と，位置情報の重要性，ユビキタス情報環境に適した位置システムの不足を考慮し，ヘテロジニアスなセンサ環境に対応した位置情報取得システムを構築する。図 1.1 にヘテロジニアスな位置センサ環境を示す。この例では数種類のセンサ，センサシステムが存在し，データの表現方法は座標，距離，2 値情報と多様である。

本研究では，既存の位置システムの問題を解決し，ヘテロジニアスなセンサ環境において，統一された位置取得方式を提供する。これにより，アプリケーションは環境を意識せずに位置情報を取得可能となる。

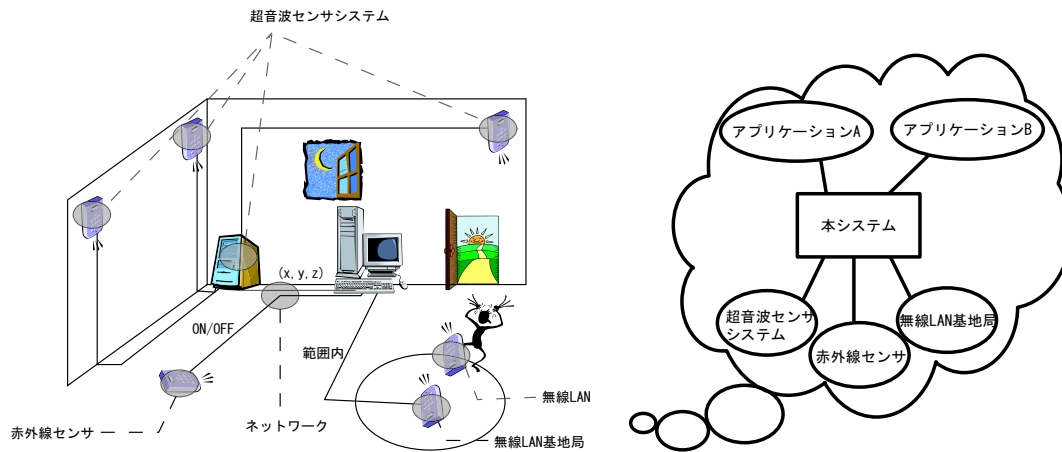


図 1.1: ヘテロジニアスなセンサ環境

1.3 本論文の構成

本論文では、第2章で本論文におけるユビキタス情報環境の定義、特徴について述べる。第3章では本研究の想定環境、解決手法、導入例をあげ、第4章で各問題を解決する設計方針について述べる。第5章では実装を動作例をふまえて説明し、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。本論文では以上のような構成を取り、ユビキタス情報環境を想定したヘテロジニアスなセンサ環境における位置取得システムについて論じる。

第 2 章

ユビキタス情報環境

本章では，本研究におけるユビキタス情報環境の定義を行い，ユビキタス情報環境の特徴について述べる．次に，ユビキタス情報環境の構成要素の一つであるミドルウェアの役割について考察し，最後に位置情報の重要性について述べる．

2.1 ユビキタス情報環境

本節では現在のコンピューティング環境について述べたあと、本研究が背景とするユビキタス情報環境の定義と特徴について述べる。

2.1.1 現在のコンピューティング環境

現在のコンピューティング環境において、ユーザに対するサービスは、ユーザ要求を情報世界で処理し、物理世界にフィードバックする形態として捉えることができる。例えばPC上で音楽を聞くというサービスでは、PCで音楽再生アプリケーションを起動し、再生する音楽ファイルを選択し、ファイル再生処理を行うという情報世界における処理を行って、PCに付属するスピーカから音として物理世界へとフィードバックを行う。

このようなサービス体系において情報世界の役割は、単に情報を処理し、ユーザに提供するという枠組みにとらわれず、ユーザを物理世界上で直接支援することにある。しかし、物理世界ではユーザの移動や物理的な状況の変化など様々な要因が存在するため、情報世界が十分な役割を果たすに至っていない。例えば、上記の音楽再生アプリケーションにおいて、ユーザの移動や状況によって、フィードバックするスピーカを変化させるというユーザ要求に答えることはできない。これは物理世界の機器の柔軟性の欠如と、物理世界の情報の取得の困難さの2点による。1点目の柔軟性の欠如とは、具体的にはユーザ入力方法と入力機器、アプリケーションと出力機器などが密に結合していることが挙げられる。柔軟性の欠如によって、物理世界に対するフィードバック方法が限定されてしまうという問題が生じる。例えば映画を提供するアプリケーションはフィードバックするディスプレイとスピーカが決まっている。2点目は、現在のセンサが、物理世界の情報取得という目的を十分に果たせていないという問題である。例えば、自動ドアに付属する赤外線センサや圧力センサはドアの前に人がいるという情報を物理世界の状態として取得しているのではなく、実際にはドアが開くトリガとしてのみ利用されている。そのため、取得した情報を再利用したり、他の機器から利用したりできない。そのため、物理世界の変化に適応したサービスの提供は非常に困難であるといえる。

2.1.2 ユビキタス情報環境の定義

前節で現在のコンピューティング環境の問題点として、機器の柔軟性の欠如と物理世界の情報収集の困難性を挙げた。

機器の柔軟性の欠如を解決するためには、機器と情報世界の処理の分離が必要である。すなわち機器は特定のアプリケーションを対象とせず、様々なアプリケーションから利用可能とすることが重要である。また、アプリケーションは特定の機器を前提とせず、様々な機器を利用できることが重要である。そのためには、機器にネットワーク接続機能と、計算処理能力を持たせることが必要となる。ネットワーク接続機能は機器における情報世界との窓口となる。また、計算処理能力を持たせることによ

り、情報世界からの要求の解釈や処理が可能となる。このようなネットワーク接続機能と計算処理能力を持った機器は情報家電と呼ばれる。

また、物理世界の情報収集の困難さを解決するためには、物理世界の情報を収集することを目的としたセンサを環境に設置しなければならない。センサにもネットワーク接続機能や計算処理能力を持たせることで、自動ドアのトリガのような機器に付随の機能ではなく、物理現象を情報世界に通知し、情報世界内で処理することが可能となる。このようなネットワーク接続機能と計算処理能力を持ったセンサをインテリジェントセンサと呼ぶ。

このようなインテリジェントセンサ、及び多様なフィードバックを可能とする情報家電で構成され、より実世界に沿った処理をアプリケーションが行い、アプリケーションの処理を物理世界の様々な状況に適した形で提供可能な環境をユビキタス情報環境と呼ぶ。

2.1.3 情報世界による物理世界の把握

ユビキタス情報環境が実現されることによって、物理世界を情報世界で把握し、情報世界の特徴を生かした実世界の支援が可能となる。情報世界の特徴は物理世界における位置、大きさなどの物理制限に依存しないことである。物理制限が存在しないことで、物理世界を情報世界内で空間的拡張、時間的拡張など、様々な拡張が可能である。それぞれの例を以下に挙げる。

- **空間的拡張**

情報世界では、物理世界のように空間の広がりには制限は存在しない。そのため、情報世界において、異なる空間を同じ空間として扱ったり、物理世界に存在しない新たな空間を創造することが可能である。

例えば、物理世界上で離れた空間にそれぞれ存在するユーザを情報世界内の同一空間に存在させ、遠隔地ミーティングを行うことが可能となる。

- **時間的拡張**

物理空間では、過去の状態を再現することは現実的に不可能である。しかし、情報空間では時間軸に沿って定期的に情報を保存しておき、過去の状態を復元できる。

例えば、部屋の家具の配置などの状態を情報空間内で保存しておき、情報世界内で任意の時間の配置を再現することが可能となる。

2.2 ユビキタス情報環境の構成要素

ユビキタス情報環境では、物理世界には、機器の柔軟性と物理世界の把握を可能とするために情報家電とインテリジェントセンサが必要となる。

情報世界では実際にユーザの入力を解釈，処理し，情報家電にフィードバックするアプリケーションが存在する．また，物理世界における情報家電とインテリジェントセンサ，情報世界におけるアプリケーションを物理世界の状態やユーザの状態など実世界の状態にあわせて結び付ける必要があり，その役割を果たすミドルウェアが必要となる．このような，情報家電とインテリジェントセンサ，アプリケーションを結び付け，ユーザの入力や状況にあわせてフィードバックを返す一連の流れをサービスと呼ぶ．図 2.1 に構成要素とサービス体系の概念図を示す．

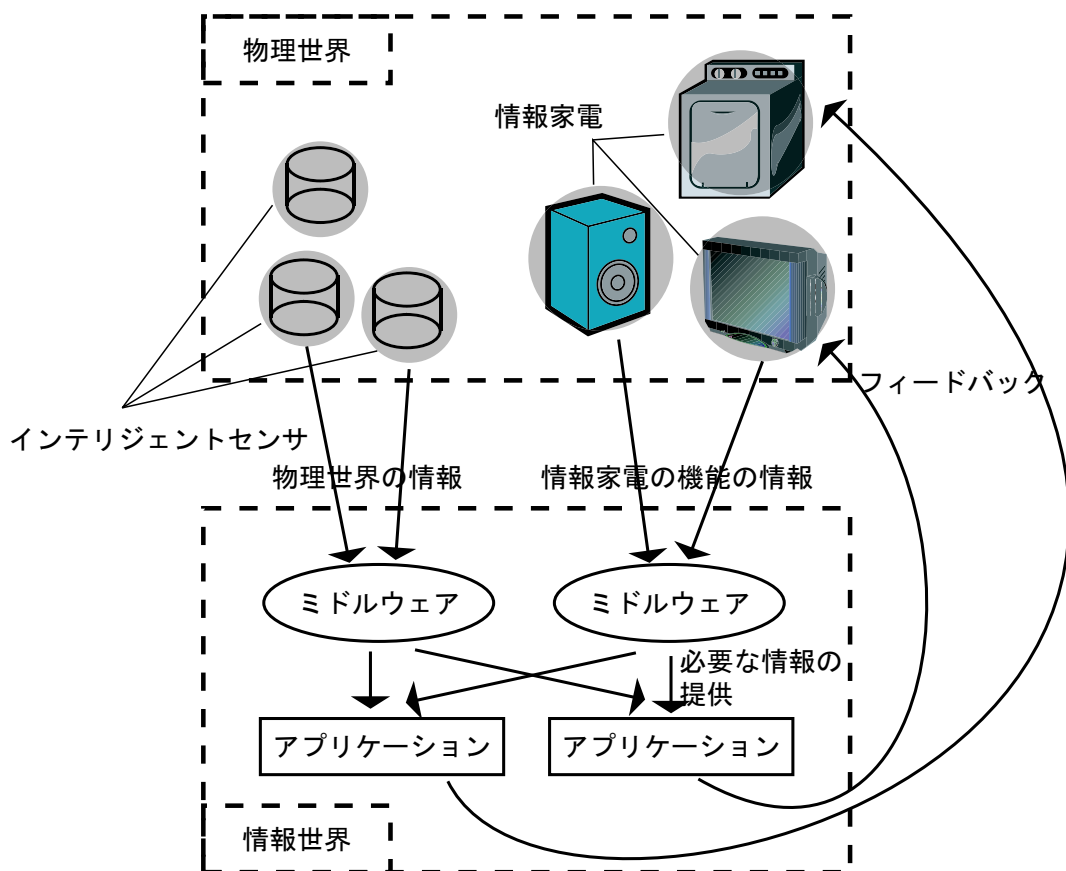


図 2.1: 情報世界と物理世界の構成要素

2.2.1 物理世界における構成要素

ユビキタス情報環境は，PC を中心とした既存のコンピューティング環境と異なり，情報家電とインテリジェントセンサから構成される．情報家電は，情報世界との窓口となるネットワーク接続機能，及び，情報世界からのフィードバック要求を解釈処理できる計算処理能力を持っているので，アプリケーションの処理と機器の分離が可能となる．

計算機の小型化は、現在一般的には導入されていない機器にも計算処理能力を持たせることも可能である。その中で、物理世界の把握という目的を達成する機器として、センサが挙げられる。ユビキタス情報環境におけるセンサは、取得した物理世界の情報を解釈し、情報世界に通知する機能を持つ必要がある。センサが取得する情報が何を表しているのかをセンサ自身が判断、通知する必要がある。そのためのネットワーク接続機能及び、計算処理能力を持っているインテリジェントセンサがユビキタス情報環境には必要である。

インテリジェントセンサにおいて物理世界を忠実に把握するためには、情報家電やユーザの様々な情報を収集する必要がある。センサには有効範囲や取得可能な情報に制限があるため、取得する情報はより多く、より正確であることが望まれる。また、物理世界は時間軸にそって刻々と変化するので、センサが固定された環境では取得できる情報に限界がある。インテリジェントセンサに移動性を持たせるなど変化に対応した環境でなければならない。

次にユビキタス情報環境における物理世界の特徴を挙げる。

- **センサの協調動作**

情報をより多く、より正確に収集するためにユビキタス情報環境ではセンサ間の協調が重要となる。情報を結合し、別の種類の情報を創出したり、他のセンサの情報と比較することにより、情報の精度を上げるなどが考えられる。

例えば、座標による位置情報を二つ組み合わせることにより、傾きや向き情報を取得したり、周囲に存在する温度センサと比較し、大きな差が生じている場合はエラー値としてはじくことができる。

- **多様性**

情報家電は冷蔵庫や電子レンジ、ディスプレイなど様々な種類存在する。また同じディスプレイであっても、解像度や色数など様々である。

またセンサにおいても、物理世界の情報をより多く集めるために、数、種類ともに多く必要である。そのため、センサの数だけではなく、センサから提供される情報の種類も多数存在する。例えば、温度を計るセンサ、圧力を計るセンサ、人体感知センサなど、センサの種類は多数存在し、それぞれ発信する情報は異なる。また同じ環境内に温度センサが多数存在し、環境内の温度変化を詳細に取得する場合など、センサの数が膨大である場合も想定される。

2.2.2 情報世界における構成要素

現在のコンピューティング環境において、ユーザからの要求を判断し、情報家電を用いて物理世界にフィードバックするという処理はアプリケーションが行っている。しかし、ユビキタス情報環境におけるアプリケーションは実世界の状態にあわせたサービス提供が求められる。また、ユビキタス情報環境では様々な情報家電やアプリケーションが存在し、情報家電とアプリケーションにおける処理が分離している。また、物

理世界の情報を取得するインテリジェントセンサも多数存在する。その中でそれぞれ独立している情報家電、インテリジェントセンサ、アプリケーションを、物理世界の状態にあわせて適時結び付け、ユーザの入力からフィードバックまでのサービス体系を作り出さなければならない。そのため、この役割を果たすミドルウェアが必要となる。次にユビキタス情報環境におけるアプリケーションとミドルウェアについて詳述する。

アプリケーション

ユビキタス情報環境は、実世界に沿った処理を情報世界で行い、実世界の様々な状況に適した形で提供可能とする。その中でアプリケーションは様々なユーザ要求方法、フィードバック方法を提供する。そのため、状況依存という特徴が生まれる。

● 状況依存

物理世界の情報を取得可能となることによって、コンテキストウェアと呼ばれる状況依存のサービス提供形態が考えられる。コンテキストウェアとは、状況に適応し、自律的に提供サービスを選択、あるいは変化させることをいう。

ユビキタス情報環境では、インテリジェントセンサを用いて状況という物理世界の現象を情報世界に通知することで、コンテキストウェアを実現できる。

ミドルウェア

ユビキタス情報環境の構成要素である情報家電とインテリジェントセンサは多様性という特徴を持つ。例えば情報家電は、音楽再生機能を持っていたり、画像表示機能を持っているなどさまざまな機能を持つ。また、インテリジェントセンサは赤外線センサや温度センサなど多くの種類存在し、同じ温度センサであっても、摂氏と華氏など表す情報が多様である。そのような環境においてミドルウェアは、それぞれの機能を把握し、物理世界の状態にあわせてアプリケーションと結び付ける必要があり、多様性へ対応し、アプリケーションからこれらの多様性を隠蔽する必要がある。また、情報家電とインテリジェントセンサ、アプリケーションが分離して独立しているので、これらを組み合わせてサービス提供形態を形成する必要があり、合成も大きな特徴となる。

● 多様性への対応

ユビキタス情報環境において、物理世界は多様な情報家電とインテリジェントセンサにより構成されているため、両者を対象とするミドルウェアは多様性に対応する必要がある。

アプリケーションとハードウェアの中間に存在するミドルウェアは、様々な環境で様々なアプリケーションが利用するので、特定の環境やアプリケーションに依存しない必要がある。

ユビキタス情報環境におけるミドルウェアは、独立している物理世界と情報世界を結び付け、サービス体系を作り出す上で、必要不可欠なソフトウェア群である。

- 合成

物理世界では、CDプレーヤやテレビなど一つの機器として完結しているものを、情報世界で扱う際、CDプレーヤをスピーカと音楽再生部、テレビをディスプレイ、スピーカというように機能ごとに扱うことによって、さまざまな情報家電から別の新しい仮想家電を構築することが可能となる。この際ミドルウェアは情報家電がどのような働きをするのかを把握し、情報世界での処理を行うアプリケーションと情報家電の結び付けを行う。これにより、遠隔地のCDプレーヤでCDを再生し、音はユーザの近くのスピーカから出力するというようなサービスが可能となる。

2.3 ユビキタス情報環境におけるミドルウェアの役割

ユビキタス情報環境において、ミドルウェアは情報家電やインテリジェントセンサの特徴を把握し、アプリケーションと結び付ける必要がある。しかし、環境を構成するすべての情報家電とインテリジェントセンサを対象とし、アプリケーションの要求に対応する一つのミドルウェアを構築することは、それぞれの多様性から不可能である。そのために、ミドルウェアは対象とする多様性ごとに独立して存在する。具体的には、情報家電の多様性を対象とした仮想情報家電ミドルウェア、UIミドルウェア、インテリジェントセンサの多様性を対象とした位置情報ミドルウェアが考えられる。

仮想情報家電ミドルウェア

ユビキタス情報環境では、物理世界において操作される情報家電の種類は多数存在する。また、オーディオコンポからスピーカのみを用いるなど、情報家電の機能分離によって、機器を横断して構築される仮想情報家電と呼ばれる概念が登場する。環境を構成する情報家電を機能ごとに管理し、状況に応じて合成するのが仮想情報家電ミドルウェアである。仮想情報家電ミドルウェアとしてVNA[1]が挙げられる。

UIミドルウェア

情報家電などの機器を機能ごとに管理し、様々な組み合わせを提供可能とすることによって、情報家電を操作するユーザインタフェースは既存のシステムのように固定的な結合を想定した設計であってはならない。変化に対応したユーザインタフェースが必要となる。このような柔軟なUIを提供可能するミドルウェアがUIミドルウェアである。UIミドルウェアとしてICrafter[2]が挙げられる。

位置情報ミドルウェア

物理世界の把握という目的のために、位置情報は不可欠である。ユビキタス情報環境において、環境内の機器、ユーザの位置を把握し、アプリケーションの要求に応じ

て、位置情報を返すミドルウェアが位置情報ミドルウェアである。位置情報ミドルウェアとして ActiveBat[3] が挙げられる。

2.4 ユビキタス情報環境における位置情報

本節では、ユビキタス情報環境における、位置情報の重要性、既存の位置取得システムの問題点、位置情報ミドルウェアの機能について述べる。

2.4.1 位置情報の重要性

物理世界の把握をするために、位置情報は不可欠な情報である。至るところに埋め込まれている PC やセンサの位置、環境を構成する情報家電の位置、アプリケーションを利用するユーザの位置など、様々な位置情報の要求が考えられる。様々な位置情報を環境内のインテリジェントセンサから取得する情報で提供する必要がある。

2.4.2 既存の位置情報ミドルウェア

多くの既存の位置情報ミドルウェアは、固定のセンサ群に依存することにより、高精度を実現している。また、向きなどの位置情報に伴う他の情報を提供している。既存の位置取得システムの研究の目標は、精度の向上と、位置情報以外の付随情報の提供の二点である。既存の位置情報ミドルウェアをユビキタス情報環境下において使用すると、次のような問題が生まれる。固定センサ群に依存しているため、すでに位置情報ミドルウェアが設置されている空間に新たな位置情報ミドルウェアを導入した場合、既存のミドルウェアの特長や利用方法などが適用できない。この場合、新旧両方のミドルウェアが協調してさらに高い精度が得られるような構造である事が望まれる。また、ある位置情報ミドルウェアが導入された空間用に構築したアプリケーションを、他の位置情報ミドルウェアが導入されている空間で利用した場合、アプリケーションは動作しない。

このように、既存ミドルウェアの利用は、ユビキタス情報環境の特徴に対して相反する。固定センサ群への依存はインテリジェントセンサの多様性という特徴を無視しており、設置の困難さはユビキタス情報環境導入の障害となる。また、拡張性の低さや、ミドルウェアに依存したアプリケーションの要求は環境の多様性に対して対応不可能となる。

以上の理由で既存の位置情報ミドルウェアは、ユビキタス情報環境下で利用することは困難である。

2.4.3 ユビキタス情報環境における位置情報ミドルウェア

ユビキタス情報環境の特徴である、情報家電、インテリジェントセンサの多様性を考慮することは、位置情報ミドルウェアの構築において不可欠なことである。インテ

リジェントセンサから提供される様々な位置情報を統合的に扱い、アプリケーションの要求するさまざまな位置情報に対応したミドルウェアがユビキタス情報環境における位置情報ミドルウェアである。

2.5 本章のまとめ

本研究の背景となるユビキタス情報環境の定義を行い、その特徴、目的について述べたあと、ミドルウェアの役割について述べ、最後に位置情報の重要性、ユビキタス情報環境に適した位置情報ミドルウェアの定義を行った。この考察の結果、ユビキタス情報環境では、ミドルウェア、位置情報は必須であり、既存の位置情報ミドルウェアではその役割を果たすのに不十分であることを示した。

次章では、本研究の概要とユビキタス情報環境がもたらす多様性の考察と解決手法について説明する。

第 3 章

研究の方針と概要

本章では，本研究の概要を述べ，ユビキタス情報環境の特徴である多様性の考察を行い，その解決手法としての抽象化について述べる．また，本研究の用いる抽象化モデルを説明し，本研究の導入例を示す．

3.1 本研究の概要

本研究は、ユビキタス情報環境に適した位置取得ミドルウェアの構築が目的である。本節では、本研究で想定する環境、本研究の概要を述べる。

3.1.1 想定環境

物理世界を情報世界で把握することが大きな目的であるユビキタス情報環境において、位置情報は不可欠な情報である。例えば、音楽を聞くというアプリケーションにおいて、ユーザの位置情報に伴い音楽を提供するスピーカーを変更したいという要求や、ユーザの周囲の範囲を指定してスピーカーを検索したいという要求が考えられる。このような位置情報を伴ったサービスを提供するアプリケーションを位置依存アプリケーションと言う。また、ユビキタス情報環境において、位置情報を取得可能なインテリジェントセンサのような機器は多数存在し、位置依存アプリケーションが多様なインテリジェントセンサに対応することは不可能である。そのため、位置情報と共に、位置情報ミドルウェアが必要となる。

本研究は部屋などの閉鎖的な空間を想定している。そのため、本研究が構築するミドルウェアは、空間ごとに存在する。また、空間内に存在する多数の位置情報を取得可能な機器はネットワーク接続機能と計算処理能力を持っている。ミドルウェアはこれらの機器と通信し、位置情報の取得を行う。アプリケーションは本ミドルウェアに位置の要求を行うことで、位置情報を取得できる。

3.1.2 研究概要

本研究はユビキタス情報環境を想定し、本環境に適した位置情報ミドルウェアの構築を目的とする。ユビキタス情報環境において超音波センサや赤外線センサなど、位置取得可能なインテリジェントセンサは多数存在する。また、インテリジェントセンサ以外にも位置が取得可能なものが存在する。例えば、無線LAN基地局からの電波強度による位置情報や、圧力マットに反応があればそこに何かがあるという位置情報が取得可能である。その他に、すでに位置情報ミドルウェアが導入されている空間も存在する。例えば超音波センサによる位置取得システムや、RFタグによる位置取得システムなどである。本研究ではこのインテリジェントセンサと、センサ以外の位置取得可能なものをあわせて、センシングモジュールと呼ぶ。前述したように、ユビキタス情報環境における物理世界は多様であり、センシングモジュールも同様である。それらのセンシングモジュールからの情報の形式は、距離情報、座標情報などさまざまである。以降、センシングモジュールからの情報をデータと呼び、情報の形式をデータ形式と呼ぶ。

また、位置情報を利用するアプリケーションは、様々な要求単位、基準点が想定される。例えば、部屋に人がいるかいないかを知りたい場合、部屋を一つの単位とした位置情報を要求し、人の細かい移動を把握したい場合、細かい単位の位置情報を要求する。以後この単位を粒度と呼ぶ。基準点とは、位置情報を表現する際に基準となる位

置のことであり、部屋の隅の場合もあれば、ユーザの場所の場合もある。アプリケーションは、部屋の統一された基準点を利用する場合もあれば、アプリケーション独自に定めた基準点を利用したい場合もある。このようにユビキタス情報環境下の位置情報を利用するアプリケーションは多様である。

本研究は、このようなユビキタス情報環境において、センシングモジュールの多様性とアプリケーションの要求粒度、基準点の多様性を統一的に扱うことを可能とするミドルウェアを構築する。

3.2 多様性の考察とその解決手法

本節では、ユビキタス情報環境における位置情報の多様性について考察し、多様性がもたらす問題とその解決手法を述べる。

3.2.1 位置情報における多様性

ユビキタス情報環境に適した位置情報ミドルウェアを構築するために、センシングモジュール、及びアプリケーションの多様性がもたらす問題を解決する必要がある。本項では、ユビキタス情報環境における位置情報の多様性として、センシングモジュールの多様性、アプリケーションの多様性を挙げ、各々のもたらす問題点について述べる。

センシングモジュールの多様性

それぞれのセンシングモジュールは把握する有効範囲が異なる。有効範囲とはそれぞれのセンシングモジュールがデータを取得可能な範囲のことであり、センシングモジュールによって異なる。例えば、赤外線センサであればセンサから延びる直線が有効範囲であり、無線 LAN 基地局は電波が到達可能な距離を半径とした球形が有効範囲となる。その他に、センシングモジュールから送られるデータ形式は存在／非存在を表す 2 値情報、距離、座標が存在する。赤外線センサであれば 2 値情報であり、超音波センサや無線 LAN 基地局ならば距離であり、既存のセンサシステムであれば座標である。また、センシングモジュールからのデータ形式が距離、座標であった場合、データの精度もセンシングモジュールごとに異なる。

このようにユビキタス情報環境下のセンシングモジュールはデータ、データ形式において多様であるといえる。この多様性により、位置依存アプリケーションがすべてのセンシングモジュールに対応することが不可能になり、結果として位置依存アプリケーションごとに特定のセンシングモジュールに依存して位置情報を取得している。そのため位置依存アプリケーションは環境に依存し、特定のユビキタス情報環境でしか利用できない。

アプリケーションの多様性

既存の位置情報ミドルウェアは、それぞれ粒度と基準点を持っており、その粒度、基準点をもとに位置情報を提供している。ユビキタス情報環境下のアプリケーションはさまざまな粒度と基準点を持っている。例えば、部屋に入ったら電気をつけるというアプリケーションは部屋を一つの単位とする位置情報を要求し、ユーザの移動に伴って点灯する電気が追ってくるアプリケーションでは、ユーザの周囲という細かい単位の位置情報を要求する。さらに、移動するユーザの周囲から機器を検索するアプリケーションは、ユーザを基準点とする位置情報を要求する。

この形態は、ミドルウェア主導の位置提供形態であり、物理世界のさまざまな状況に適したサービスの提供を困難にする。上記の例において、既存の位置情報ミドルウェアから位置情報を取得したアプリケーションは、アプリケーション内で粒度、基準点を変更する必要がある。

3.2.2 多様性への解決手法としての抽象化

様々な多様性は、特定の環境でしか利用できないアプリケーションや、ミドルウェア主導の柔軟でないアプリケーションの原因となり、アプリケーションプログラマの負担になる。

ユビキタス情報環境における多様性がもたらす問題を解決する手段として、ミドルウェアによる抽象化が用いられるが、抽象化手法として統合と包括を挙げ、各手法を比較する。

- **統合**

統合とは、対象とする情報を一つの形式に統一することである。例えば位置情報において、様々なデータ形式を緯度、経度で表すなどが考えられる。

統合の利点は、多様な情報を扱う際、決まった形式を対象とすればよいため、システムの構築が容易なことである。逆に問題点は、決められた形式で表現できない情報が必ず存在することである。情報を提供する側は、決められたフォーマットへの変換という作業を行う必要があり、負担が増加する。例えば、すべてのデータ形式を緯度、経度で表そうとした際、距離情報を緯度、経度で表現することは不可能である。

- **包括**

包括とは多様な情報をできる限り変化させずに、かつ一つのモデルで表現することである。例えば、電車の路線図は正確な地図ではないが、より扱いやすいようにし、すべての地下鉄路線を表現しているモデルであるといえる。

包括の利点は様々な多様な情報をそのまま利用できることにある。問題点は、包括するモデルによってはより扱いにくくなってしまうことである。

3.3 3次元空間へのマッピングによる抽象化手法

本節では、ユビキタス情報環境の特徴である多様性に対する本研究の解決方法を述べ、解決手法である抽象化のモデルを提案する。

3.3.1 解決手法

物理世界の形状、位置情報を情報世界で表現するには、数値として捉える必要があり、空間図形で表現することにより可能である。本研究では、様々な多様性を空間図形で表現することで、抽象化を行う。空間図形として扱うことにより、位置情報における多様性を包括手法を用いて抽象化可能である。統合手法を用いると、対応不可能なデータ形式が存在し、また多くのデータ形式に対応可能な統合手法は存在しない。よって、本研究では包括手法による空間図形を用いた抽象化手法を採用する。

3.3.2 抽象化モデル

本研究では、空間を外接する直方体で表し、センシングモジュールのさまざまな有効範囲を数種類の基本立体で表現する。また、アプリケーションの要求粒度を立方体の集合からなる3次元空間にマッピングする。この2つのマッピングによって抽象化を行う。センシングモジュール、アプリケーション要求の抽象化手法を以下に述べる。

データ抽象化空間

センシングモジュールをマッピングするための有効範囲を決定する要素として、指向性、始点の形状、最大距離がある。指向性とはセンシングモジュールが値を取ることができる角度であり、始点の形状とは、センシングモジュールが表すデータの基準点の形状であり、最大距離とはセンシングモジュールが値を取ることができる最大の距離である。この3つの要素からマッピングする基本立体を決定する。これらの基本立体がマップされる3次元空間をデータ抽象化空間と呼ぶ。このマッピングによって、様々なセンシングモジュールの有効範囲を抽象化し、一つの仮想空間で表現する。センシングモジュールの範囲をマッピングした基本立体の上で、データを扱う。

アプリケーション要求空間

本研究はアプリケーションに対して、一定の粒度の立方体で構成された3次元空間を提供する。この立方体の粒度はアプリケーションによって自由に決定できる。この3次元空間をアプリケーション要求空間と呼ぶ。アプリケーションは、データ抽象化空間の構成を気にすることなく、このアプリケーション要求空間の粒度を単位として自由に位置を取得できる。アプリケーション要求空間における単位立方体のことをリジョンと呼ぶ。

動作例

図 3.1 に本研究における抽象化手法を示す。データ抽象化空間内では、有効範囲をいくつかの基本立体で表している。アプリケーション A, B, C に対して、各要求粒度で構成された立方体の集合でアプリケーション要求空間を構築している。このデータ抽象化空間とアプリケーション要求空間を合成することにより、本システムではセンシングモジュールの有効範囲をアプリケーションの要求粒度へと変換する。

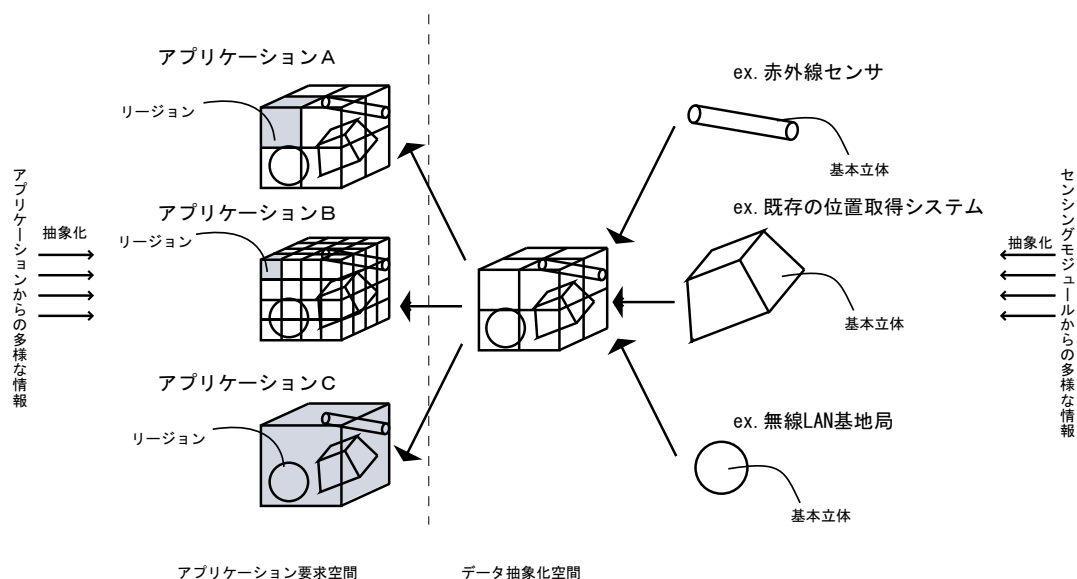


図 3.1: 抽象化概念図

ヘテロジニアスなセンサ環境において、アプリケーション要求空間内のリージョンの粒度に対してデータ抽象化空間内の各センシングモジュールの有効範囲が充分であるとは限らない。その場合、位置が単一のリージョンに決定できないため、該当するリージョンのすべてを通知する。

3.4 想定シナリオ

本節では想定シナリオを元に、本研究の構築するシステムの利用例を述べる。

3.4.1 パーソナルシアター

周囲のスピーカーとディスプレイを用い、ミニシアターを提供するアプリケーションを想定する。自分の部屋のような個人が占有可能な環境であれば部屋全体の空間から対象とする情報家電を検索し、公共の場のような多くの人が利用するような環境ではユーザ周辺の狭い範囲で情報家電を検索する必要がある。このように環境に適応して要求粒度を変更すべきアプリケーションは多く存在する。

既存の位置取得システムであればこのようなアプリケーションごとの要求粒度とは関係なしに、一定の粒度の位置情報を提供したり、アプリケーションの要求する粒度を満たすことが不可能であれば、新たなシステムの導入が必要となる。ユビキタス環境下において、アプリケーションは環境に応じて求める粒度を自由に変更できる必要がある。

図3.2に本例における導入例を示す。アプリケーションはどのような単位で位置情報を要求するかを本システムに通知することにより、目的に直結した位置情報を得ることが可能となる。

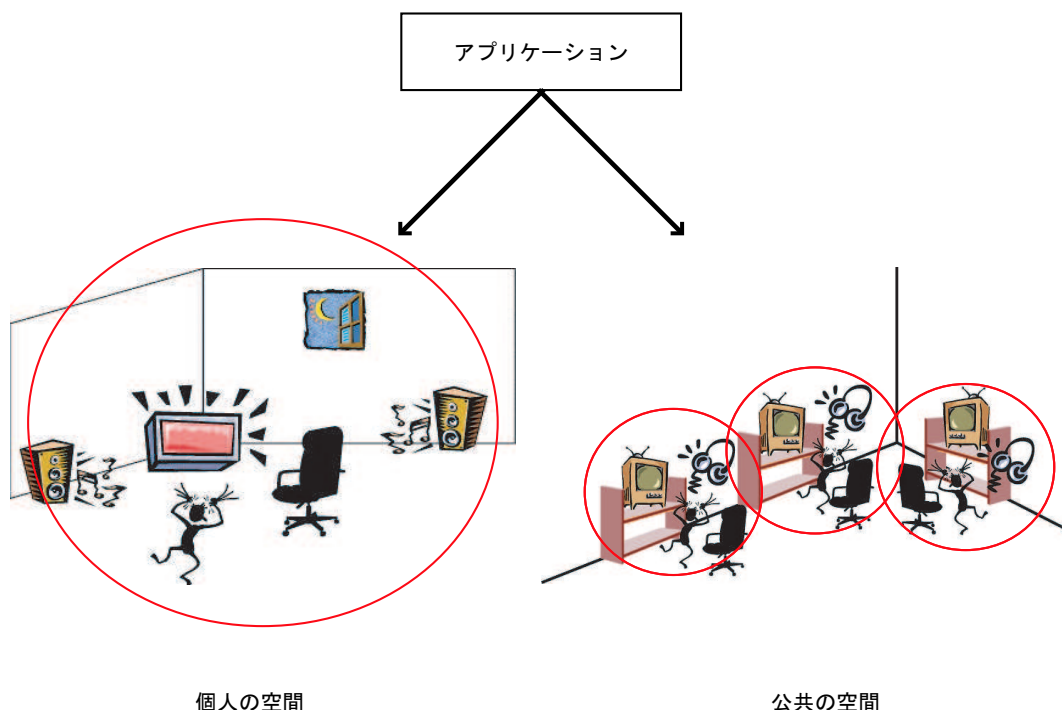


図 3.2: パーソナルシアター

3.4.2 動的ライト ON/OFF アプリケーション

部屋に入るとライトが自動的に点灯するアプリケーションを想定する。この場合、アプリケーションは細かい粒度の位置情報を必要とせず、部屋にいるかいないかを判断できれば良い。既存のシステムを用いて実現する場合、部屋の範囲と人の座標を比べることにより、部屋にどうかをアプリケーション自らが判断する手法が考えられるが、これは細かい座標からの再計算が必要となる。

位置システムには、既に部屋の入り口に設置されている赤外線センサや圧力センサなどを利用して座標から再計算することなく部屋の出入りを監視し、アプリケーションに対して部屋全体という大きな粒度の位置情報を提供することが求められる。また、

無線 LAN の範囲など粒度の粗い情報を利用して、求める情報を得ることが可能となれば、新たな位置システムの追加の必要がなくなる。このように、アプリケーションは部屋を一つのリージョンとする粒度を要求し、リージョンに人がいるかないかという目的に直結した単純な情報を得ることが望まれる。

3.4.3 仮想 3 次元空間における遠隔地制御アプリケーション

遠隔地から仮想空間で視覚的に機器を操作するためのミドルウェアを構築する際、実世界における機器の位置管理が必須になる。このようなミドルウェアはさまざまな環境で使用される。しかし、環境によって位置取得システムが異なる場合、ミドルウェアはそれぞれの位置取得システムに対応したインタフェースを用意する必要がある。

位置情報を利用するミドルウェアが多くの環境で使用される場合、それぞれの環境の位置システムを統一的に扱い、インタフェースを統合するシステムの存在は必須となる。本システムはさまざまなセンサ、センサシステムを扱うことにより、環境の混在におけるインタフェースの統合的役割もはたす必要がある。

図 3.3 に本例における導入例を示す。環境 A は超音波システムによる位置取得方式が導入されており、環境 B は RF タグによる位置取得方式が導入されている。本システムが導入されることにより、遠隔地制御を行うアプリケーションは導入されている位置取得システムの違いを考慮せずに位置情報を取得することが可能となる。

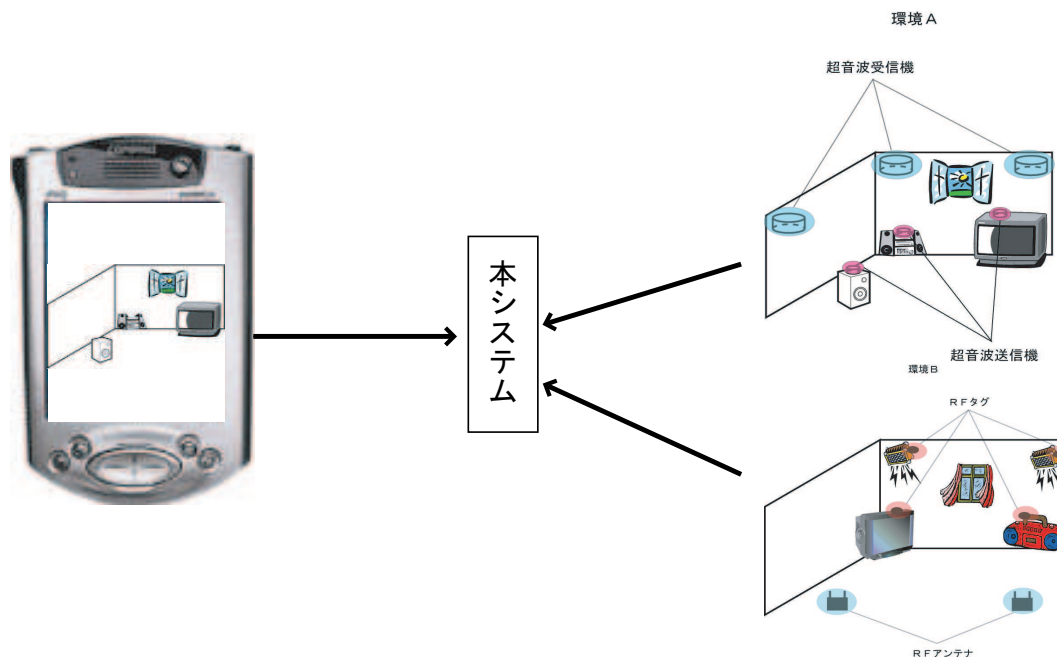


図 3.3: 遠隔地制御アプリケーション

3.5 本章のまとめ

本章では、本研究の概要を述べ、ユビキタス情報環境の特徴である多様性の解決手法について考察した。次に、本研究における解決手法である抽象化モデルを説明し、最後に本研究の導入例を示した。

次章では、本章で示した抽象化モデルを実現する OASISS システムの設計について述べ、抽象化モデルの詳細な実現手法を示し、最後に OASISS システムの考察を行う。

第 4 章

設計

本章ではユビキタス情報環境を想定した位置ミドルウェアである OASISS システムの設計について述べる。設計として方針、ソフトウェアモジュールの詳細について述べ、実際の動作例をシナリオに沿って示す。最後に設計方針の考察を行う。

4.1 OASISS の設計方針

本研究では、ユビキタス情報環境に適した位置情報ミドルウェアとして OASISS (Optimized Abstracted Sensor Information for Smart Space) を構築する。本節では OASISS を構築するために必要となる機能、ハードウェア構成、ソフトウェア構成について述べる。

4.1.1 必要機能

OASISS はユビキタス情報環境を実現するために、抽象化機能、合成機能、位置情報提供機能を必要とする。

抽象化機能

ユビキタス情報環境における位置情報ミドルウェアである OASISS は、アプリケーションの要求の多様性、センシングモジュールの多様性を考慮する必要がある。そのために、アプリケーションの要求、センシングモジュールのデータを抽象化し、OASISS において統一的に扱う。本システムでは、アプリケーション要求の多様性にはアプリケーション要求空間を構築し、センシングモジュールの多様性にはデータ抽象化空間を構築することにより抽象化を行う。

合成機能

センシングモジュールからのデータによりデータ抽象化空間を構築したあと、実際に位置情報を利用するアプリケーションに対して位置情報を提供するために、データ抽象化空間をアプリケーション要求空間に変換する必要がある。OASISS においてデータ抽象化空間とアプリケーション要求空間を合成することによって、センシングモジュールからの情報をアプリケーション要求空間上で表現可能となる。

位置情報提供機能

アプリケーションからの位置情報の要求形式は一樣ではないので、合成した後、アプリケーションに応じて位置情報を加工する必要がある。

4.1.2 ソフトウェア構成

本システムは空間合成モジュール、アプリケーションインタフェース、センシングモジュールインタフェースより構成される。各モジュールの関係を図 4.1 に示し、次節よりモジュールの詳細を述べる。

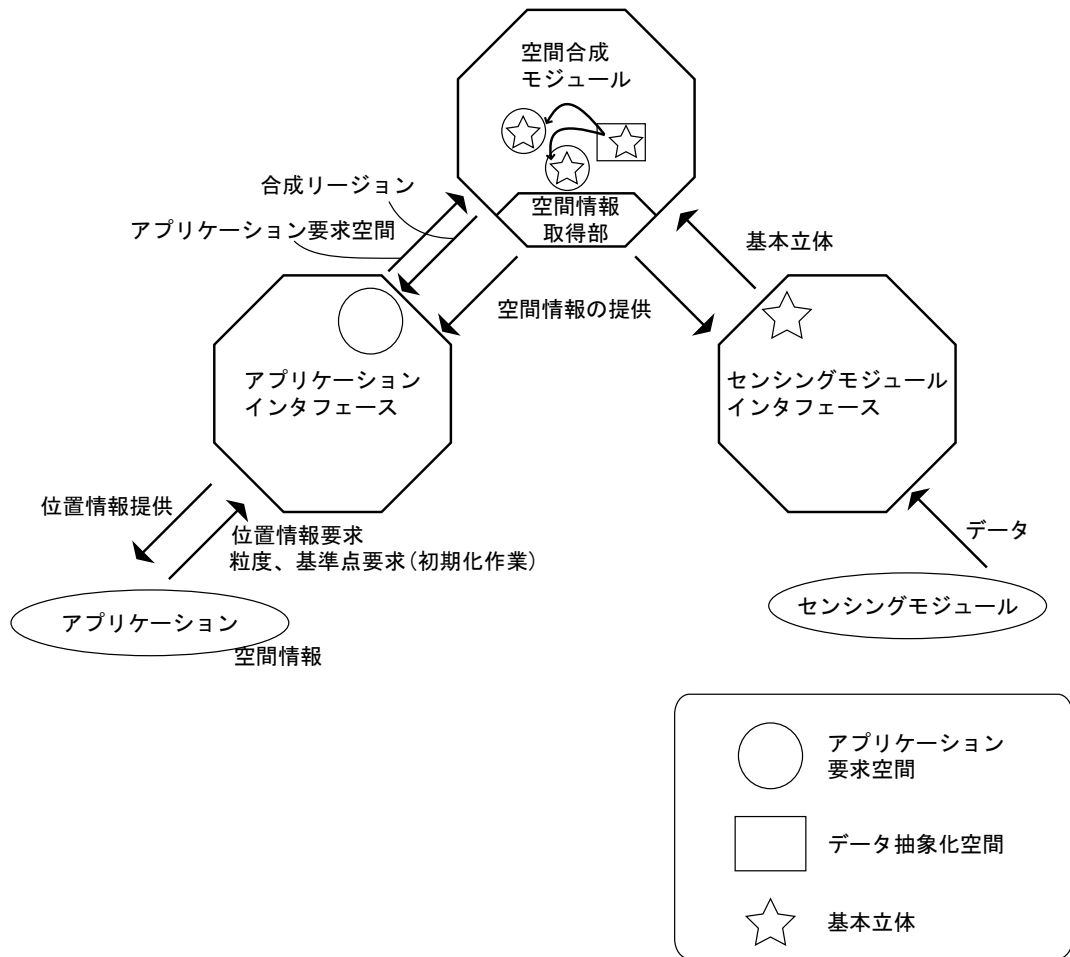


図 4.1: 各モジュールの関係

4.2 空間情報取得部の設計

本システムは部屋などの空間ごとに導入され、導入されている空間を対象とする。そのため、本システムは最初に空間情報を取得、解釈する必要がある。空間合成モジュール内の空間情報取得部では、本システムが導入された際、最初に空間情報を取得し、アプリケーションインタフェース、センシングモジュールインタフェースに対して、必要時に空間情報を通知する役割を果たす。

4.2.1 空間情報の取得

本システムでは、空間を外接する直方体で表現するため、空間の大きさを取得する必要がある。取得する情報は、外接直方体の高さ、幅、奥行きである。

空間情報取得部は、空間の大きさの情報からアプリケーション要求空間、データ抽象化空間を構成する際の、基準となる直方体を生成する。この直方体をベース直方体と呼ぶ。ベース直方体は、高さ h 、幅 w 、奥行き d の空間を図 4.2 のように表す。ベース直方体を作る際に通知する各要素は本システムであらかじめ定められた単位を元に通知する。例えば、ミリメートル (mm) を単位に持つ場合は、 w 、 d 、 h はこのミリメートルを単位とする数値である。また、図 4.2 のようにベース直方体を生成する際、一点を基準としている。この点をベース直方体における基準点と呼ぶ。このように生成されたベース直方体を座標系として捉え直すと、図 4.2 における右図のようになる。この座標系をベース座標系と呼ぶ。この座標系は、単位をミリメートルとし、ベース直方体の基準点を原点とする。

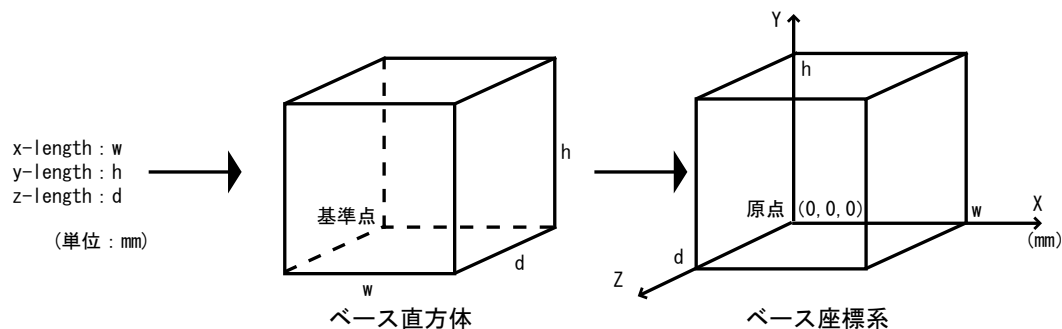


図 4.2: ベース直方体

4.2.2 空間情報の提供

OASISS ではこのベース直方体を元にアプリケーション要求空間、データ抽象化空間を構築するため、必要時に空間情報を他のモジュールに提供する必要がある。空間情報取得部では問い合わせに応じて、空間情報としてベース直方体の高さ、幅、奥行きを通知する機能が必要である。

4.3 アプリケーションインタフェースの設計

本システムでは、アプリケーションごとにアプリケーションの粒度と基準点より構築されるアプリケーション要求空間を保持する。アプリケーションインタフェースでは、アプリケーションからの接続要求時に通知される粒度と基準点、及び空間情報取得部からの空間情報を用いてアプリケーション要求空間を構築し、空間合成モジュールへ転送する。

また、アプリケーションから位置情報の要求があるたびに、空間合成モジュールで行われた合成判定の結果をリージョン番号として受け取り、アプリケーションの要求に対応した形式に変換する。例えば、あるセンシングモジュールの名前を指定し、周囲のリージョンから他のセンシングモジュールを検索する際、空間合成モジュールからは指定したセンシングモジュールに対応するリージョン番号がアプリケーションインタフェースに通知される。アプリケーションインタフェースは通知されたリージョンの周囲のリージョンを指定して再度位置要求を行い、取得した結果をアプリケーションに通知する。

4.3.1 アプリケーション要求空間の構築

アプリケーションは接続の際、本システムに要求粒度と基準点を通知する。基準点は本システム内のベース直方体における基準点からの座標で表す。基準点を通知しない場合、ベース直方体の基準点をアプリケーションの基準点とする。

本システムは、要求粒度を一辺の長さとする立方体であるリージョンの集合としてアプリケーション要求空間を構築する。このリージョンの集合は座標系として捉え直すことが可能である。アプリケーションから要求粒度を取得後、本システムではアプリケーションの座標系を生成する。この座標系をアプリケーション座標系と呼ぶ。アプリケーション座標系では、要求粒度により単位を生成する。例えばアプリケーション座標系の単位を a とし、本システムの基準となる単位を mm とする。また、要求粒度を $10000mm$ としたとき、

$$1a = 10000mm$$

となる。

図4.3にリージョンの生成、アプリケーション座標系の生成を示す。この図において、座標系の単位を a とし、要求粒度を b とする。また、ベース座標系における (x, y, z) がアプリケーション座標系における原点となる。よって、ベース座標系の任意の点 (p, q, r) をアプリケーション座標系に変換するには、

$$\begin{pmatrix} P \\ Q \\ R \end{pmatrix} = \begin{pmatrix} (p+x)/b \\ (q+y)/b \\ (r+z)/b \end{pmatrix}$$

となる。

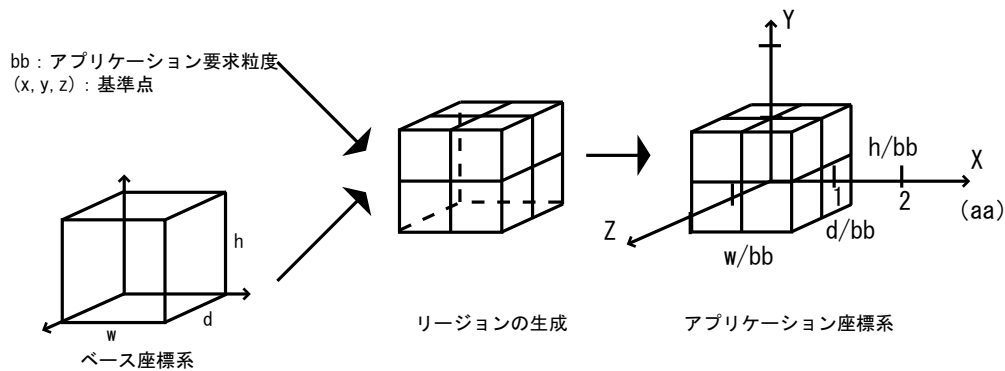


図 4.3: アプリケーション座標系

4.3.2 アプリケーション要求空間の転送

アプリケーションインタフェースで空間を構築後、空間の保持は空間合成モジュール内で行うため、アプリケーション要求空間は空間合成モジュールへと移動する。

4.4 センシングモジュールインタフェースの設計

本システムでは、初期作業としてデータ抽象化空間を構築する。このデータ抽象化空間は、空間合成モジュール内に保持する。また、センシングモジュールは、センシングモジュールインタフェースに接続する。センシングモジュールインタフェースでは、センシングモジュールが最初に本システムに通知する Character-Info と Establish-Info を用いて、次のことを行う。

- センシングモジュールの有効範囲をもとに基本立体を生成
- 基本立体を複数のオブジェクトに分割
- データ抽象化空間に各基本立体を配置

この作業によりデータを統一的に扱うことを可能とする。本節では、データ抽象化空間の構築方法、Character-Info、Establish-Info、基本立体の生成方法、基本立体の分割方法、基本立体のデータ抽象化空間配置方法について述べる。

4.4.1 データ抽象化空間の構築方法

データ抽象化空間は、本システムの初期作業として生成され、座標系としてとらえた場合、ベース座標系と単位、軸の角度、基準点は同じである。このデータ抽象化空間における座標系をデータ座標系と呼ぶ。

4.4.2 センシングモジュール固有の情報

本システムへ接続されるセンシングモジュールは本システムが Character-Info, Establish-Info と呼ぶ2つの情報を明示する必要がある。この2つの情報は、センシングモジュールが本システムに接続する際に通知する。

- **Character-Info**

Character-Info はセンシングモジュールの始点の形状、最大距離、指向性、データタイプから構成され、必要があればデータ精度も含む。始点の形状は点、円、長方形、最大距離は長さ、指向性は角度、データタイプは2値情報、距離、座標、データ精度は長さで表される。Character-Info は基本立体を決定する要因として利用される。

- **Establish-Info**

Establish-Info はセンシングモジュールの設置の際の情報であり、始点の頂点座標情報と向き情報から構成される。これらを数値で指定する。この Establish-Info をもとに、本システムは基本立体をデータ抽象化空間にマッピングする。

4.4.3 センシングモジュールの範囲の決定

センシングモジュールはそれぞれ物理世界の情報を取得可能な範囲を持つ。本システムでは Character-Info の中の始点の形状、最大距離、指向性からセンシングモジュールの範囲を決定する。本システム内では、始点の形状を、点、円、長方形に限定する。各情報と、範囲の形状を図4.4に示す。本システムで扱うセンシングモジュールは、範囲をこの形状に抽象化する。

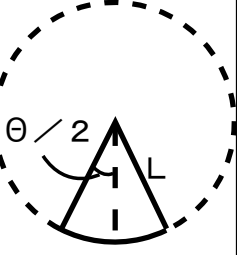

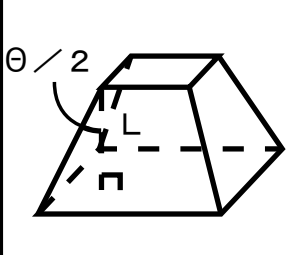
4.4.4 基本立体の生成

本システム内で扱うことが可能なデータタイプは2値情報、距離、座標である。距離、座標の場合はデータ精度を明示する必要がある。データタイプが2値情報の場合は範囲を一つの基本立体として、データタイプが距離の場合は範囲を始点からデータ精度で分割した複数の基本立体として、データタイプが座標の場合は範囲をデータ精度を一辺とする立方体で近似した複数の基本立体として扱う。最終的にすべてのデータタイプにおいて、データは各基本立体毎の存在／非存在を表す2値情報で表される。図4.5に基本立体生成の概念図を示す。

本システムでは、基本立体に分割することにより異なるデータタイプのセンシングモジュールを統一的に扱うことを可能とする。

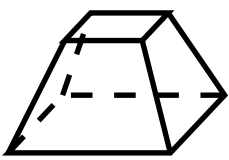
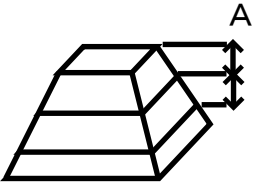
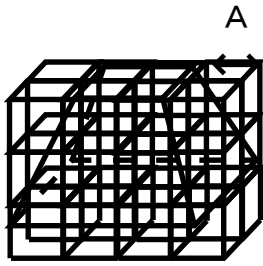
4.4.5 基本立体の配置

基本立体を生成したあと、センシングモジュールインタフェースは空間合成モジュール内のデータ抽象化空間に基本立体を配置する。基本立体の配置には、Establish-Info

始点の形状	点	円	長方形
センシング モジュールの 範囲			
センシング モジュール例	<ul style="list-style-type: none"> ・超音波センサ ・赤外線センサ ・無線LAN基地局 	<ul style="list-style-type: none"> ・焦電型 赤外線センサ ・既存の位置 提供システム 	<ul style="list-style-type: none"> ・圧力マット ・既存の位置 提供システム

最大距離 : L
指向性 : θ

図 4.4: センシングモジュールの範囲の形状

データタイプ	ON or OFF	距離	座標
基本立体			
センシング モジュール例	<ul style="list-style-type: none"> ・赤外線センサ ・圧力センサ 	<ul style="list-style-type: none"> ・超音波センサ ・レーザー 	<ul style="list-style-type: none"> ・既存の位置取得 システム

データ精度 : A

図 4.5: 基本立体の生成

を用いる。Establish-Infoによって通知される頂点座標，向き情報より，本システムはデータ抽象化空間に対する基本立体の位置，傾きを取得し，配置することが可能となる。

4.5 空間合成モジュールの設計

空間合成モジュールには，アプリケーションインタフェースからアプリケーション要求空間を取得し，保持する。また，センシングモジュールインタフェースよりデータ抽象化空間を取得し，各アプリケーション要求空間と合成を行う。その他に，アプリケーションからの位置要求を取得し，要求に沿った形で位置情報を提供する。

4.5.1 空間の取得及び合成

アプリケーションインタフェースより通知されたアプリケーション要求空間は，アプリケーションごとに空間合成モジュールで保持される。また，データ抽象化空間を取得し，各アプリケーション要求空間に変換する。データ抽象化空間のデータ座標系は，ベース座標系と単位，各軸の傾きは同一であるため，データ座標系からアプリケーション座標系への変換は，ベース座標系からアプリケーション座標系への変換と同じである。その後，データ抽象化空間における基本立体と，アプリケーション抽象化空間におけるリージョンを対応させる。実際には基本立体をさらに細分化した単位オブジェクト単位でリージョンと関連づける。図4.6において空間合成モジュールにおける一連の作業の流れを示す。

4.5.2 位置情報提供リージョンの決定

合成作業によってリージョンというアプリケーション要求に近い位置情報となるが，実際の位置情報の要求は様々である。例えば，アプリケーションがIDを指定して位置の要求を行った時に，同じIDを持った単位オブジェクトが2個存在し，それぞれいくつかのリージョンに対応していたとする。また，それぞれの単位オブジェクトは1個の単位オブジェクトで交わっていた場合，アプリケーションに通知するリージョンは1個に限定可能である。このように最終的にアプリケーションの位置要求に応じて加工を行う必要がある。

4.6 シナリオ

本節では，いくつかのシナリオを例に各モジュールの動作を示す。

新たなアプリケーションの追加

本システムに新たなアプリケーションの接続があった場合，以下の手順を踏む。

1. アプリケーションから粒度と基準点を取得

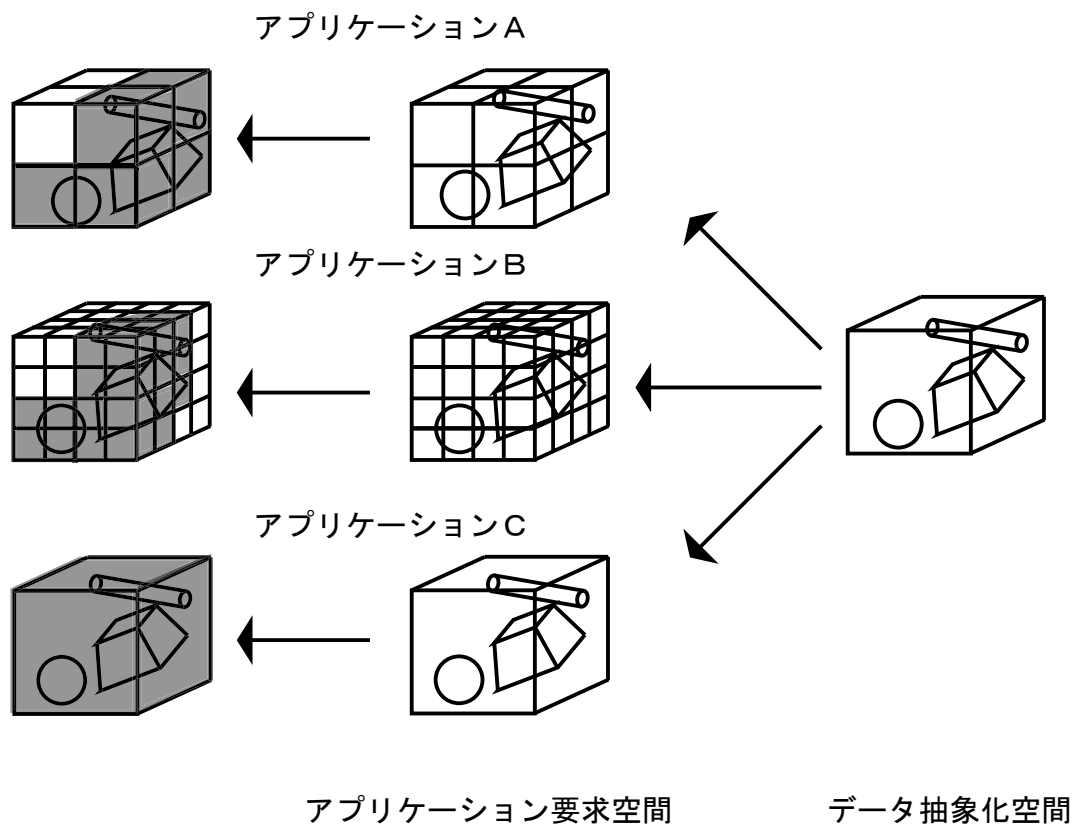


図 4.6: データ抽象化空間とアプリケーション要求空間の変換

2. アプリケーション要求空間の構築
3. 空間合成モジュールにアプリケーション要求空間を転送
4. データ抽象化空間と合成
5. 衝突リージョンを計算

以下の図 4.7 にアプリケーションの追加の概要を示す。

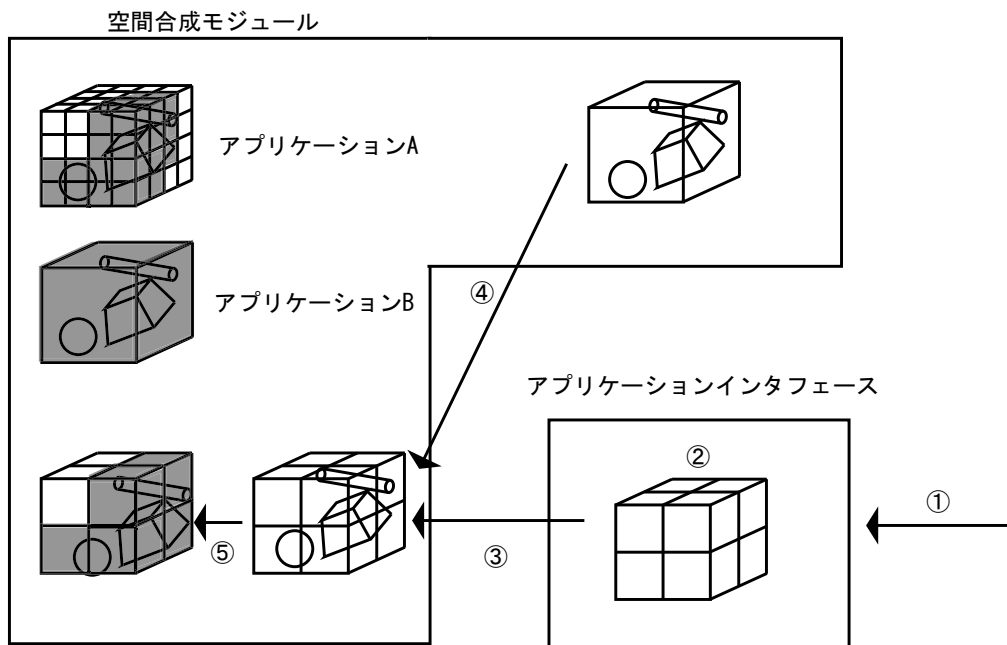


図 4.7: 新たなアプリケーションの追加

新たなセンシングモジュールの追加

本システムに新たなセンシングモジュールの追加があった場合以下の手順を踏む。

1. センシングモジュールから, Character-Info, Establish-Info を取得
2. データ抽象化空間に基本立体を配置
3. 各アプリケーション要求空間に基本立体を追加
4. 合成判定のやり直し

以下の図 4.8 にセンシングモジュールの追加の概要を示す。

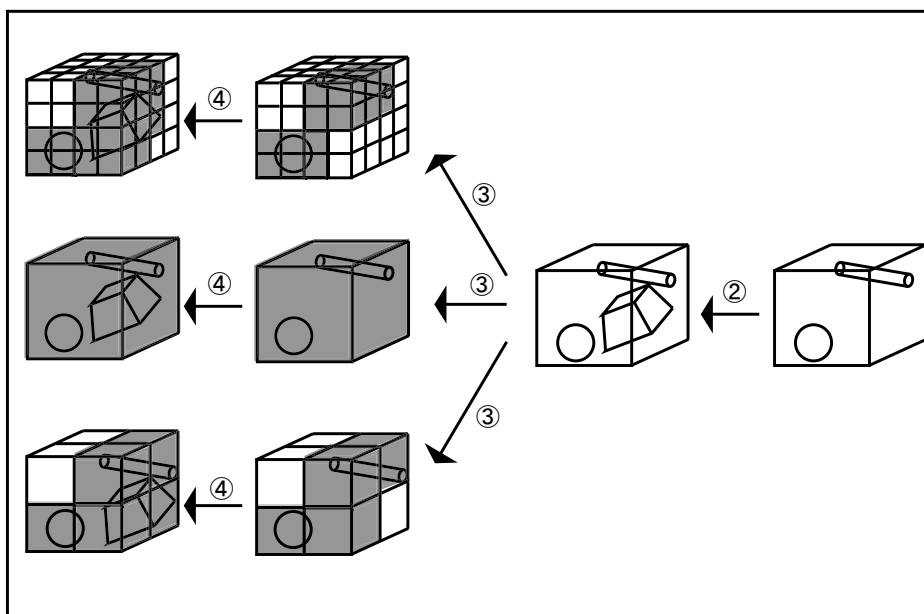


図 4.8: 新たなセンシングモジュールの追加

4.7 設計手法の考察

一般的に本抽象化モデルを実現する際、本システムにおけるベース座標系において、アプリケーション要求とセンシングモジュールからの情報を対応させる方法が採用される。すべての情報をベース座標系で扱うことで、統一的に扱うことが可能である。しかし、統一的に計算されたアプリケーションの要求とセンシングモジュールの情報を、アプリケーションの位置要求の度にアプリケーションが要求する粒度に変換する必要があるという点が問題点として挙げられる。しかし、本システムはアプリケーション毎に保持されるアプリケーション要求空間を用いて計算処理を行っている。そのためアプリケーションへの高い応答性が期待できる。反面、各アプリケーション要求空間を保持し、センシングモジュールから生成された基本立体をすべてのアプリケーション要求空間に配置しなおす手間が増え、システムリソースの面で問題がある。

4.8 本章のまとめ

本章では、ユビキタス情報環境における位置情報ミドルウェアである OASISS システムの設計を行い、本システムにおける位置情報とアプリケーション要求の空間図形への変換方法、取り扱いについて述べた。

次章では、OASISS システムの実装を行う。

第 5 章

OASISS の実装

本章では，本研究における抽象化モデルを実現する位置情報ミドルウェアである OASISS システムの実装について述べる．

5.1 実装方針

OASISS のプロトタイプ実装は Java 言語を用いて行った。本システムはユビキタス情報環境を想定しているため、様々な OS や CPU を想定する必要がある。Java 言語 [4] は JavaVM が動作する環境で動作可能であり、本システムの想定環境において最適である。実際に本システムは Windows2000, linux 2.4.18-0v13 上で同様の動作を示した。

また、システムの実装の他に、システムの動作状況を表示する GUI の実装を行った。GUI を実装することにより、本システムの内部処理を視覚的に確認可能となった。

5.2 実装環境

本実装は、Windows2000 上で、J2SDK1.4.0[5] を用いて行った。センサ情報を抽象化する基本立体は、直方体に限定して行った。さらにデータ抽象化空間と、アプリケーション要求空間を視覚的に確認するための GUI を Java3D ライブラリ [6] を用いて実装した。

表 5.1: 実装環境

CPU	PentiumIII 1200MHz
主記憶	768MB
OS	linux 2.4.18-0v13
JavaVM	J2SDK1.4.0
GUI	java 3D 1.3(OpenGL)SDK

5.3 オブジェクト動作

本システムでは、システム初期化作業として空間情報を取得する必要がある。本実装において空間情報は XML [7] を用いて記述する。また、空間の情報を視覚的に確認するために、部屋のテクスチャ画像を XML の記述を用いて取得する。図 5.1 に本実装における XML ファイルの記述例を示す。

空間情報を取得すると同時に、本システムは、アプリケーション要求空間、データ抽象化空間を管理する RoomData オブジェクトを生成する。本システムでは RoomData オブジェクトを中心とし、アプリケーション要求空間ごとに生成されるアプリケーション要求空間オブジェクト、データ抽象化空間を表すデータ抽象化空間オブジェクトから構成されている。各オブジェクトの関連を図 5.2 に示す。また、アプリケーション要求空間オブジェクトとデータ抽象化空間オブジェクトはそれぞれ、空間内のセンシングモジュールを表す基本立体オブジェクトを持っている。

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE roomdata SYSTEM "roomdata.dtd">
<roomdata>
  <name>SSLab</name>
  <size>
    <unit>mm</unit>
    <x-length>540</x-length>
    <y-length>360</y-length>
    <z-length>540</z-length>
  </size>
  <image>./image/SSLab.gif</image>
</roomdata>

```

図 5.1: 空間情報を定義する XML ファイル

アプリケーションが本システムに新たに接続する際、アプリケーション要求空間はデータ抽象化空間からすべての基本立体情報を取得する必要がある。また、データ抽象化空間に新たな基本立体が追加された場合、すべてのアプリケーション要求空間に基本立体の情報を通知する必要がある。このとき、RoomData オブジェクトは、すべてのアプリケーション要求空間オブジェクトとデータ抽象化オブジェクトを一元的に管理しているため、各オブジェクトは、他のオブジェクトの状態を RoomData オブジェクトから取得することができる。

本実装における主なオブジェクト関係は、図 5.2 のようになる。

5.4 アプリケーション要求空間の実装

本システムは、アプリケーションが新たに接続する度にアプリケーション要求空間オブジェクトを生成する。アプリケーションが接続の際に本システムに通知する粒度に基づいて空間を立方体に分割し、各立方体を一つのリージョンとして構築する。さらに、各立方体ごとにハッシュテーブルを用意し、RoomData オブジェクトのセンシングモジュール情報に基づきデータの格納場所を用意し、アプリケーション要求空間内の基本立体の高さ、幅、奥行き、回転角、頂点座標のデータを表すオブジェクトを代入する。

5.5 データ抽象化空間の実装

センシングモジュール情報を表現する基本立体を、始点の形状を長方形に限定してデータ抽象化空間を構築した。また、データタイプは座標、2 値情報に対応して実装を行った。センシングモジュールが本システムに接続される度に、基本立体を生成し、データ抽象化空間に追加する。センシングモジュールを追加の際にすでに存在するアプリ

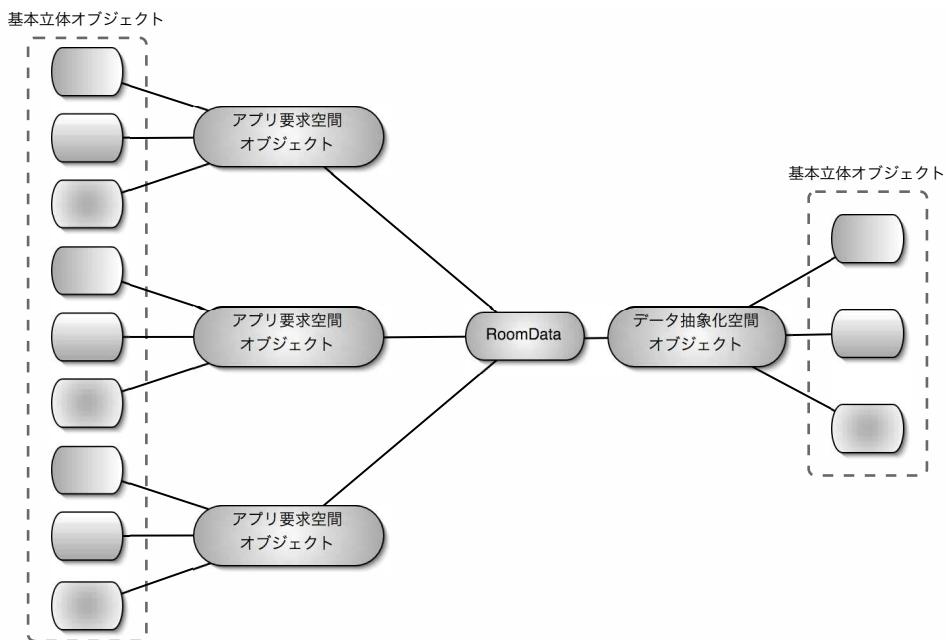


図 5.2: オブジェクト関係図

ケーション要求空間オブジェクトのハッシュテーブルにセンサデータ領域を確保し、アプリケーション要求空間内の基本立体の高さ，幅，奥行き，回転角，頂点座標のデータを表すオブジェクトを代入する．新たにセンサが追加される度にこの作業を繰り返す．

5.6 空間合成の実装

構築した二つの空間を合成することにより，基本立体と各リージョンが交差しているかを判断し，基本立体として表されている各センシングモジュールのデータをアプリケーションの要求するリージョン情報へと変換する．アプリケーションは，データ抽象化空間の構成を知らずにリージョン単位で位置情報を取得可能である．

5.7 アプリケーションの利用方法

アプリケーションは本システムが提供する API を利用することによって，容易に実装することが可能となる．以下の図 5.4 に Java 言語を用いたアプリケーションの実装例を簡易的に示す．

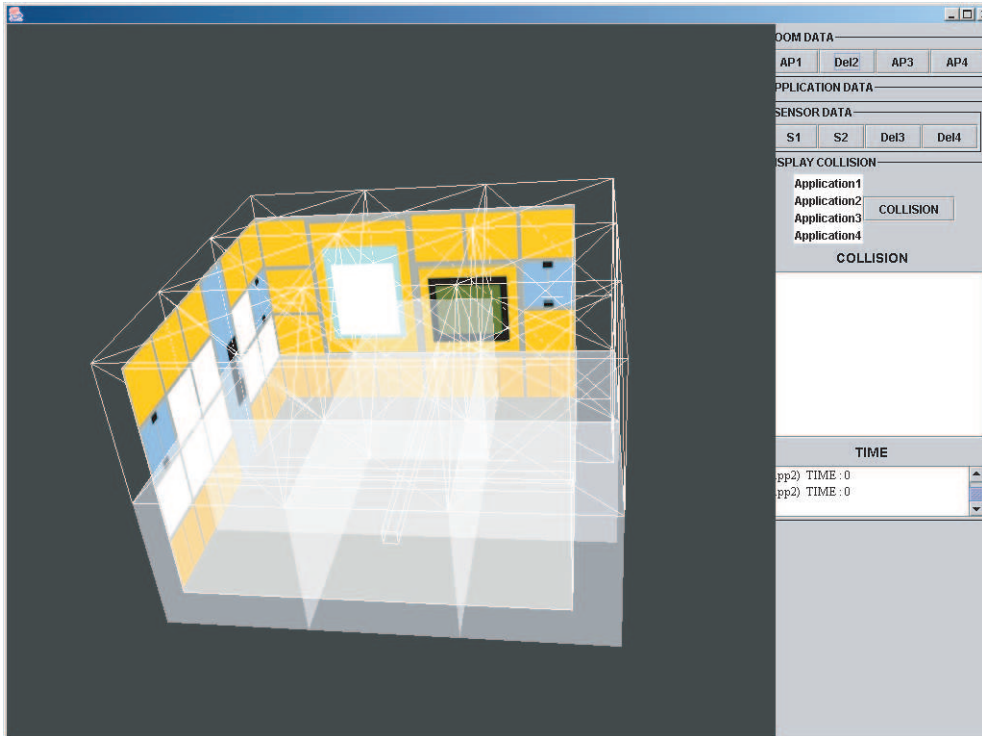


図 5.3: 合成作業の GUI

```

//初期化作業
public void oasis_init(){
    //オブジェクトの取得
    oasis = createOasis();
    //粒度の送信
    oasis.setAccuracy(1000,"mm");
    //基準点の設定
    oasis.setPoint(3000,2000,3000,"mm");
}

public static void main(String args){
    Reasion reasion; //オブジェクト識別子
    ID id; //オブジェクト識別子
    ID ids[];

    oasis_init();

    //アプリケーション依存コード
    //..... 省略
    //.....

    //センシングモジュール"id"の位置情報をリージョンオブジェクトとして取得
    reasion = oasis.getLocation(id);

    //指定したリージョン内にあるすべてのオブジェクトの ID を配列で取得
    ids = oasis.getLocation(1,2,2);

    //終了処理
    oasis.exit();
}

```

図 5.4: アプリケーション実装例

5.8 本章のまとめ

本章では，OASISS システムのプロトタイプ実装について述べ，実装環境，実装手法，実装例を示した．次章では，本実装を用いた OASISS システムの評価を行う．

第 6 章

評価

本章ではOASISSの評価を関連研究との比較, 基本性能の測定
において行う.

6.1 関連研究との比較

既存の位置取得システムを、センサの多様性への対応を目的とするものとそれ以外に分類し、前者の例として、ActiveBat と Cricket、後者の例として POIX を挙げる。

6.1.1 The Active Bat Location System

ActiveBat[3] はオフィスなどの環境を想定した超音波センサを用いた位置システムである。位置を知りたい対象に超音波を発信する小型デバイスを取り付け、天井に設置された複数の超音波受信機からの距離より位置を算出する。誤差は 10 センチ以内で取得可能であり、これらの位置情報を用いて、ユーザに位置適応的なサービスを提供する。

しかし、Active Bat は、天井に複数の超音波センサを設置する必要があり、設置の際に多大な労力を必要とする。また、Active Bat は特定の超音波センサに特化してつくられており、他のセンサの利用は考慮に入れていない。したがって、アプリケーションは ActiveBat に依存した形で書く必要がある。

6.1.2 Cricket

Cricket[8] は超音波と無線を用いた、センサのヘテロジニティに対応していない位置取得システムである。屋内での位置取得を想定している。ActiveBat との大きな違いはユーザが送信機を持ち、これによりプライバシーの保護が可能となることである。

Cricket beacon と呼ばれる機器を建物内に複数配置し、無線で超音波を定期的に発信する。Cricket compass と呼ばれる機器を小型機器につけ、compass の位置を取得する。無線は同期を取るために用い、無線を受信してから超音波を受信するまでの時間差によって、距離を求める。また、超音波センサを複数用いることにより、向き情報も取得できる。

想定アプリケーションとして、Cricket compass から距離と方向を指定し、その範囲内にあるデバイス一覧を取得する ViewFinder、位置情報と向き情報を利用して、ユーザに建物内の道案内をする WayFinder などがあげられる。

しかし、ユーザが送信機を持っていることにより、送信機の向きと天井に設置される CricketBeacon の配置により位置情報が取れなくなる場合がある。また、ActiveBat と同様に、設置の困難さがあげられる。

6.1.3 POIX

POIX[9] は XML で記述されたインターネット上における位置情報の交換フォーマットである。位置の表現だけでなく、位置を中心として様々な情報を包括的に提供可能である。例えば、対象が移動体であった場合、移動手段、移動速度、移動方向などが提供される。位置は、緯度、経度、平面誤差、高度、高度誤差で表現される。

しかし、想定以外のデータで表現されるセンサを追加することが不可能となるため、緯度、経度などで表すことができない位置情報は対象外となる。ユビキタス環境下に

において、緯度、経度で表すことができないセンシングモジュールは多数存在する。また、閉鎖的な空間を想定していないので、ユビキタス環境における位置記述フォーマットとしては不十分である。また、アプリケーション要求の多様性は想定していない。

6.2 基本性能の測定

アプリケーション要求から、その要求に適した形で位置情報を計算するまでの時間を把握するための定量的評価を行った。本システムが利用するセンサの個数、アプリケーションの要求粒度を変化させることにより、これらの値と本システムの負荷の関係を測定した。

6.2.1 測定環境

測定を行ったハードウェアおよびソフトウェアの環境を表 6.1 に示す。

表 6.1: 測定環境

CPU	PentiumIII 1200MHz
主記憶	768MB
OS	linux 2.4.18-0v13
JavaVM	J2SDK1.4.0

6.2.2 センサ数とアプリケーション粒度における合成判定時間

本システムでは、アプリケーション要求空間を構成する立方体と、データ抽象化空間内の基本立体の合成判定を行い、アプリケーションに提供する位置情報のリージョンを計算する。この合成判定の実行速度を決定する要因として、以下の要素が考えられる。

- アプリケーション要求空間の粒度
- センシングモジュールの数
- 基本立体の形状
- 基本立体を占める単位オブジェクトの数

本測定では、アプリケーションの要求粒度と、センサ、センサシステム数における実行速度の変化を測定した。

表 6.2: 測定結果 (ms)

要求粒度 (cm) センサ個数 (個)	120 (5×5×4)	100 (6×6×4)	80 (7×7×5)	60 (10×10×7)
1	2.05	3.9	6.1	14.8
2	2.05	8.8	5.25	22.2
3	3.35	12.15	9.15	24.7
4	4.65	13.15	15.5	54.3

実際には 540cm × 540cm × 360cm の直方体の空間において、アプリケーション要求空間の粒度を 60cm, 80cm, 100cm, 120cm, センサノード数を 1 から 4 と変化させて行った。表 6.2 に測定結果を示す。

本測定により、要求粒度、センサ個数が増加するにつれて、交差判定時間が増加していることがわかる。空間を構成するリージョン数は要求粒度に対して三次関数的に増加するので、要求粒度の細かさは、リージョン数において大きく反映される。今後、空間合成における交差判定の計算方法を工夫する必要がある。

60 センチ単位で位置を取得できれば、人の動き、ものの動きを判断する際、実世界の動きを判断可能である場合が多い。以上のことをふまえて、測定結果よりユビキタス環境下において利用可能であると考ええる。

6.3 本章のまとめ

本章では、OASISS システムを関連研究と比較し、基本性能を測定した。本評価により本システムがユビキタス情報環境における位置情報ミドルウェアとしての機能を果たし、実用に耐えうることを示した。次章において本研究の結論を述べる。

第 7 章

結論

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 今後の課題

本節では OASISS システムの課題として今後の拡張すべき項目について述べる。

7.1.1 ID の管理

本システムの課題として情報家電やユーザの ID と、位置情報との関連付けが挙げられる。情報環境内でデバイスやユーザの ID 付けによる識別が可能となりつつあり、Auto-ID[10] などの技術が研究されている。このような実世界のオブジェクトを ID で管理するとき、管理情報に位置情報を付加することによりより物理世界に忠実な管理情報となる。今後ディレクトリサービスなどの利用により、OASISS システム内において ID の管理を行う。

7.1.2 空間の協調

OASISS システムは部屋などの閉鎖的な空間を想定し、空間内の位置情報の把握、提供を目的としている。今後、ユビキタス情報環境の普及に伴い、空間横断的なサービスが必要となる。例えば、ある部屋から違う部屋に移動する際、位置アプリケーションにとって位置情報が途切れずに提供されることが望まれる。位置情報の連続性を提供するために、異なる空間に導入されている OASISS システム同士の協調が必要となる。

7.1.3 アプリケーションに対する多様な API の提供

位置アプリケーションにおいてさまざまな位置の要求が想定される。例えば、不特定のユーザの位置情報を要求するアプリケーションと、特定のユーザの位置情報を要求するアプリケーションが考えられる。また、特定の ID を指定して位置情報を要求するアプリケーションもあれば、スピーカーやテレビというような特定の属性を指定して位置情報を取得したいアプリケーションも存在する。今後実際の運用を重ねていくことで、アプリケーションの要求方法の考察、API の追加を行う。

7.2 本論文のまとめ

本論文では、ユビキタス情報環境における位置情報ミドルウェアである OASISS システムを提案し、設計と実装、評価を行った。本システムを利用することにより、ユビキタス情報環境下のセンシングモジュールからの多様な情報、位置アプリケーションからの粒度や基準点の多様な要求への対応が可能となる。

今後普及していくと予想されるユビキタス情報環境では、ますます多様性への対応方法が重要となり、本システムのような多様性を吸収することを目的としたミドルウェアの役割は大きい。また、ユビキタス情報環境の目的である物理世界の状況把握を行

上で、位置情報は欠かせない。よって多様性への対応、位置情報の提供という面において本システムは今後大きな役割を果たす。

本研究において OASISS システムは、センシングモジュールの多様性を吸収し、アプリケーションに環境非依存な位置取得方式を提供できた。

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年の由良淳一氏には、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、門田昌哉氏、高橋元氏、田丸修平氏、鈴木源太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 2月 22日
青木 俊

参考文献

- [1] Nakazawa, J., Tobe, Y. and Tokuda, H.: On Dynamic Service Integration in VNA Architecture, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. 7, No. E84-A, pp. 1610–1623 (2001).
- [2] Shankar R. Ponnekanti, Brian Lee, A. F. P. H. and Winograd, T.: ICrafter: A Service Framework for Ubiquitous Computing Environments, *UbiComp 2001: Atlanta, GA, USA*, 56-75 (2001).
- [3] Andy, W., Alan, J. and Andy, H.: A New Location Technique for the Active Office, *IEEE Personal Communications*, No. 5, IEEE, pp. 42–47 (1997).
- [4] Sun Microsystems Inc.: *Java Technology*.
<http://java.sun.com/>.
- [5] Sun Microsystems Inc.: *Java 2 SDK, Standard Edition, Version 1.4.0*.
<http://java.sun.com/products/archive/j2se/1.4.0/>.
- [6] Sun Microsystems Inc.: Java 3D(TM) API Home Page.
<http://java.sun.com/products/java-media/3D/>.
- [7] W3C: Extensible Markup Language (XML). <http://www.w3c.org/XML/>.
- [8] Nissanka, B, P., Anit, C. and Hari, B.: The Cricket Compass for Context-Aware Mobile Applications, *Proc. 7th ACM MOBICOM* (2001).
- [9] MOSTEC: POIX Point Of Interest eXchange language. <http://mostec.aplix.co.jp>.
- [10] Center, A.-I.: AUTO-ID Center (1999). <http://www.autoidcenter.org/>.