

屋内位置把握・誘導フレームワークの提案

井上 信介[†] 谷口 浩輔[†] 那 毅[†]
野口 豊司[†] 徳田 英幸^{††,†††}

近年、RFID や無線技術の進歩により、屋内で精度良く位置取得ができるデバイスが数多く開発されている。近い将来、屋内で端末や人間の位置情報を利用するようなアプリケーションが広く普及すると考えられる。しかし、そのようなコンテンツを高機能に、短時間、低コストで開発する為には、複雑なロケーションモデルの処理、多様な位置取得デバイスへの対応等の問題が障害となっている。これに対して我々は、アプリケーションの構成をいくつかの層に分割して、アプリケーションに依存する処理だけを記述することで、容易にアプリケーションを開発できるような研究を進めてる。本研究では、屋内での位置情報アプリケーションを効率よく開発する為のフレームワークと、フレームワークが利用する新たなロケーションモデルを提案する。

The proposal of the framework which locates the position and gives guidance indoors

SHINSUKE INOUE,[†] KOSUKE TANIGUCHI,[†] NA YI,[†]
TOYOSHI NOGUCHI[†] and HIDEYUKI TOKUDA^{††,†††}

In recent years, by the progress of RFID and radio technology, many devices which can locate the position efficiently indoors have been developed. And the possibility that applications which use the position information of a terminal or human indoors will spread widely in the near future is high. However, in order to develop such applications at short time and low cost efficiently, some problems have been obstacles, such as processing of complex location model or correspondence to various positioning devices. For solving these problems, we are doing the research about developing the framework which is used by the applications that locate the position and give person or object guidance indoors. In this research, we propose the framework and the new location model which is used in our framework.

1. はじめに

ユビキタスコンピューティングの一例として、Location-aware なアプリケーションが注目を集めている。人やものの位置情報を元に情報を提供するというコンセプトは、現在ではカーナビゲーションシステムや GPS 機能搭載型の携帯電話等を通じて一般ユーザにも広く普及している。

屋内での位置情報に関して言えば、衛星からの電波が届かないため GPS を利用することが困難であったが、現在では、RFID を用いる手法¹⁾ や無線 LAN を用いる手法²⁾³⁾⁴⁾⁵⁾ などさまざまなデバイスを用いた測位手法が考案されており、すでに実用化段階にまで達してきている。また、ユビキタス環境を実現するためのセンサーやモバイル端末なども数多く開発されてきており、近い将来、多くの Location-aware アプリケーションが普及するであろう。

一般ユーザーにこれらのアプリケーションを普及させる為には、さまざまな位置取得デバイスを用いて、多様な Location-aware アプリケーションを、低コスト、短時間で効率よく開発できなければならない。そのため、我々はロケーションモデルと再利用性に着目した。

ロケーションモデルとは、位置情報を表現するための現実空間のモデルである。Location-aware アプリケーションは、人やものの位置情報に応じてサービスを提供するものであるため、アプリケーションでロケーションモデルを扱う必要がある。特に、屋内の位置情報は非常に複雑な構造を持っているため、ロケーションモデルに関わる処理も複雑になる。そのため、どのようなアプリケーションにも使えるようなロケーションモデルを採用し、そのロケーションモデルに基づいた処理を抽出し、再利用することで、アプリケーション側の負担を軽減できる。

また、位置や角度の検出、経路の計算など、Location-aware アプリケーションの最も基本的なデータ取得部分を、抽象化して再利用可能なコンポーネントとして提供することで、アプリケーションは多様なモジュールに柔軟に対応することができ、また似たようなアプリケーションを開発する際、大幅に開発工数を削減できると考えられる。

[†] 株式会社内田洋行 次世代ソリューション開発センター
UCHIDA YOKO CO.,LTD. Advanced Solution R&D Center

^{††} 慶應義塾大学大学院 政策・メディア研究科
Graduate School of Media and Governance, Keio University

^{†††} 慶應義塾大学 環境情報学部
Faculty of Environmental Information, Keio University

本研究では、上記のような考えに基づいた屋内での位置把握や誘導のためのフレームワーク、uGuidance Framework を提案する。また、uGuidance Framework が利用する新たなロケーションモデルとして GeographySpace-LocationXml(GSLX) を開発したのであわせて紹介する。

以下、第 2 節では、関連研究について説明し、第 3 節で本研究の問題意識と目的について述べ、第 4 節でアプローチについて述べる。さらに、第 5 節で GSLX の設計、第 6 節で uGuidance framework の設計について述べる。その後、第 7 節で GSLX と uGuidance Framework の評価を行い、第 8 節で本稿をまとめる。

2. 関連研究

本稿では、関連研究としてロケーションモデルと、Location-aware アプリケーションの階層モデルについて述べる。

2.1 ロケーションモデル

Location-aware アプリケーションが用いるロケーションモデルには大きく分けて 2 種類存在する。それぞれ、symbolic model と geometric model と呼ばれている。

symbolic model とは、空間をツリー構造として記述する形態のモデルである⁶⁾。このモデルの典型的な例として住所がある。例えば、“日本国東京都江東区潮見内田洋行オフィス”といったように、粒度の大きい領域から粒度の小さい領域にツリー構造を持たせることにより、人間にとって空間同士の関係が把握しやすく、検索しやすいという利点を持っている。しかし、座標系を用いた表現方法ではないため、領域の位置を表現できないという欠点も持っている。

一方、geometric model の方は、位置を縦・横・高さといった座標で表現するモデルのことで、その典型的な例として GPS 座標系がある。geometric model は、位置や距離を正確に表現することができるが、領域同士の関係は分からない。

このように、どちらのモデルにも長短があるため、現在、両者を複合した hybrid model が提案されている⁷⁾⁸⁾⁹⁾。構造としては symbolic model であるが、その属性として座標系を導入することによって、空間の関係を把握しつつ、位置や距離を正確に表現することを実現している。

2.2 Location Stack

Location Stack⁹⁾ は、Location-aware アプリケーションの機能を階層化し、アプリケーションに対して単一のプロトコルを提供しようとする考え方に基づくソフトウェアモデルである。この階層構造の考え方は、ネットワークの階層構造モデルである OSI 参照モデルと似ている。

階層ごとの機能と各階層間の通信プロトコルを標準化することによって、アプリケーションや位置取得センサはお互いを知る必要なく、開発することができる。Location Stack における階層構造を 1 に示す。

Layer	説明
Sensors	センサによって生データを取得する
Measurements	生データを位置情報に変換する
Fusion	複数の位置情報から正確な位置を推測する
Arrangements	位置情報同士の関係を推測する
Contextual Fusion	位置情報とその他のデータを統合する
Activities	実環境の状態を推測する
Intensions	アプリケーションの処理

表 1 Location Stack の階層構造

3. 問題意識と目的

本節では、本研究で対象となる想定環境を設定し、次に屋内の位置情報を表現するためのロケーションモデルの必要性について述べ、さらに、屋内で利用される Location-aware アプリケーションののためのフレームワークの必要性について述べる。その上で、本研究の目的を述べる。

3.1 想定環境

本研究は、オフィスビル、大学のキャンパス、商店街や大規模商業施設、家庭など複数の部屋が存在する環境を想定している。また、環境、もしくはその環境内に存在する人やものに任意の位置検出モジュールが設置されている。このような環境内でさまざまな Location-aware アプリケーションが多数開発されることを想定している。

3.2 屋内におけるロケーションモデルの必要性

屋内の位置情報は屋外と比べて非常に複雑である。屋内用のロケーションモデルが表現しなければならない、屋内に特有の構造には以下のようなものがある。

- 入れ子構造
例えば、ビルの中にはフロアがあり、フロアの中には部屋があり、部屋の中にはテーブルや椅子などの家具がある。このように、屋内では粒度の大きい領域と粒度の小さい領域が存在しており、それらの領域がツリー状の入れ子構造を形成している。
- 立体構造
建造物はビルの 1 階・2 階フロアなどのように、同じ X,Y 座標を持つ複数の領域が存在する。そのため、どのフロアかを特定する為に高さ情報を付加する必要がある。さらに、領域の中には“ビル”などのように、平面として近似するのではなく、立方体として表現しなければならない領域も存在する。
- 空間同士の位置関係
入れ子構造、立体構造によって表現される領域が、ある座標系に対してどういう位置関係で配置されているのかを表現する必要がある。
- 経路・近接性
領域内には壁や障害物などが存在しているため、領域内を人や物が通ることができる経路をあらかじめ用意しておく必要がある。さらに、例えば高いパーティションで区切られている 2 点間の場合、会話や手渡しはパーティション越しにできるが、人がそこに行くには遠回りをしなければならない。このよう

な近接性⁶⁾と呼ばれる、文脈によって変化する経路をも表現しなければならない。

汎用的に、どのようなアプリケーションの利用にも耐えうるロケーションモデルでは、以上のような屋内特有の構造を表現しなければならない。

3.3 フレームワークの必要性

アプリケーションを開発する際、以下のような要件を満たすようなフレームワークがあるなら、短時間、低コストで高性能なアプリケーションを開発できると考える。

3.3.1 重複ロジックの統合

アプリケーション側でロケーションモデルに関連する処理を行うことが非常に多いが、そういった処理は、アプリケーションによってそれほど差異はない。そこで、共通のロケーションモデルに依存した処理を汎用的なモジュールで行うことで、アプリケーション側に複雑なロジックを記述する必要がなくなる。

3.3.2 モジュールのカプセル化

位置取得デバイスには多くの種類があるが、それぞれのデバイスの出力するデータ形式やタイミングなどは、デバイスによってまちまちである。そのため、アプリケーションの実装が位置取得デバイスに強く依存してしまい、異なる位置取得デバイスへの追加変更ができない。この場合、各位置取得モジュールの差異を吸収する機構を持つことで、アプリケーションはどのような位置取得デバイスを使っても、意識することなく開発することができる。

3.3.3 モジュールの選択

各位置取得デバイスは位置検出精度やコスト面で一長一短であることが多い。そのため、状況や場所に応じて、その時点で最も適当な位置取得モジュールをアプリケーションに提供できる機構があると便利である。

例えば、屋外ではGPSモジュールを用い、屋内のフロア全体を無線LANで測位するようなモジュールを用い、特定の重要なエリアだけをRFIDモジュールを用いて位置検出できるならば、低コストで高精度に位置検知できる。

3.4 目的

本研究では、Location-awareアプリケーションの開発を容易に行う為の、屋内での位置把握、さらに人やものを誘導するアプリケーションの為にフレームワークを提供することを目的とする。

4. アプローチ

本研究では、3.4節で述べた目的を達成する為、屋内位置把握・誘導フレームワーク uGuidance Framework を開発した。本節では、uGuidance Framework のシステム構成について述べ、次に uGuidance Framework が利用するロケーションモデル、GSLX の位置付けについて述べる。

4.1 uGuidance Framework のシステム構成

我々は3.3節で論じた3つの機能を実現する為、uGuid-

ance Framework では、Location-aware アプリケーションを機能ごとに4つの層に分割した。その階層モデルを図1左に示す。

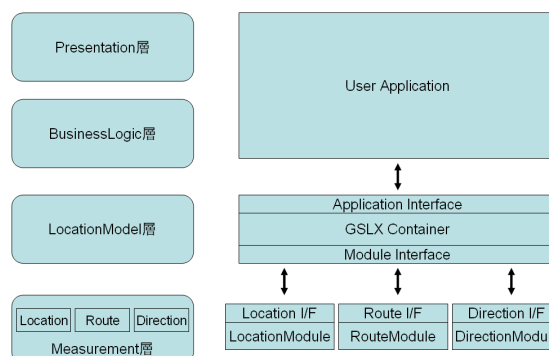


図1 階層モデルとシステム構成

- Measurement 層

Location-aware アプリケーションに必要な情報、すなわち位置情報、経路情報、角度情報を生成する層。この層に位置するモジュールは、位置取得、経路計算、角度取得モジュールのいずれかになる。

- LocationModel 層

ロケーションモデルに従って、位置情報に関する処理を行う層。BusinessLogic 層から Measurement 層を隠蔽する役割も担っている。

- BusinessLogic 層

アプリケーションの仕様に従って処理を行う層。アプリケーション開発者が開発しなければならない、システムの本体部分である。

- Presentation 層

ユーザーとのインタラクションを行う層。

アプリケーションを階層モデルすることにより、3.3節で論じた問題が解決できる。3.3.1節で論じた重複ロジックの統合は、フレームワークが LocationModel 層の機能を提供することで実現できる。また、3.3.2節で論じたモジュールのカプセル化は、それぞれの層を標準インターフェース化することで実現できる。さらに、3.3.3節で論じたモジュールの選択を実現するためには、センサーデータを直接 BusinessLogic 層に渡すのではなく、Location-Model 層を介在させることで実現できる。

4.2 ロケーションモデルの位置付け

uGuidance Framework はロケーションモデルに依存する、屋内での Location-aware アプリケーションのための汎用的なフレームワークである。従って、uGuidance Framework が利用するロケーションモデルは、屋内の構造を正確に表現でき、かつ処理しやすいものでなければならない。そのため、我々は uGuidance Framework のために GSLX という新たに開発したロケーションモデルを採用した。

5. GSLX の設計

本節では、uGuidance Framework が利用するロケーションモデル、GSLX の設計について述べる。まず GSLX の全体的な構造を述べ、次に GSLX の各要素について述べる。

5.1 GSLX の全体的な構造

フレームワークは、屋内で利用されることを前提としているので、そこで採用されるロケーションモデルは、3.2 節で論じたような構造を表現できなくてはならない。そのため、今回採用した GSLX では、2.1 節で述べた hybrid model を採用した。なお、GSLX は XML Schema¹⁰⁾ として定義されており、各要素には出現回数が設定された子要素と属性を持つ。

図 2 に GSLX の構造を示す。

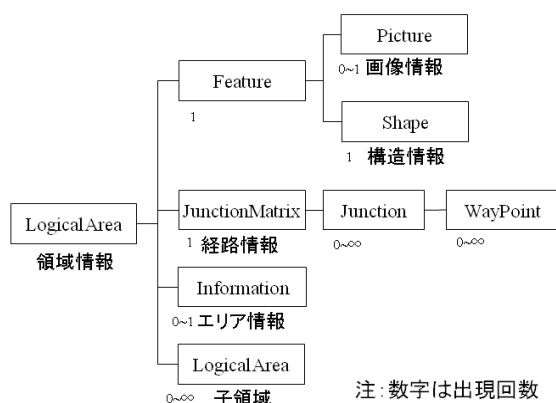


図 2 GSLX の構造

5.2 GSLX の要素

GSLX での実世界の領域は、LogicalArea という単位で定義される。LogicalArea とは、地域、建造物、部屋、家具、地点など、任意の粒度をもつ高さ属性を持っている 2 次元平面、もしくは立方体として表現される領域を意味する。GSLX は LogicalArea のツリー構造として定義される。

LogicalArea には子要素として 4 つの要素を持つ。各要素については、次の通りである。

- LogicalArea
子要素として Feature・JunctionMatrix・Information・ChildArea を、さらに属性として AreaId・Layer を持つ。AreaId はその領域を識別する為の識別子、Layer は高さを表す。Layer は、2 階・3 階といった抽象的な高さ、もしくはビルのような立方体を示す。
- Feature
LogicalArea の形を表す要素。自エリアが親エリアのどの部分にあたるのかを指定することで形を表現する。
表現方法には 2 通りのやり方があり、それぞれ Shape

と Picture という子要素内で表現する。Shape は、親エリアの LogicalArea からみた自エリアの形を、矩形・多角形・円・点を用いて表現する。

一方、Picture では自エリアの平面図の画像を、親エリアの画像にマッピングすることによって、形を表現する。ただし、その領域の画像が存在しない場合は、Picture 要素を指定しなくてもよい。

- JunctionMatrix
LogicalArea 内の移動可能な通路網を表す要素。通路を表現するために、JunctionMatrix の子要素として Junction、Junction の子要素として WayPoint という要素を用いる。
Junction とは交差点を意味する要素で、通路の始点を表現する。Junction から延びている通路の終着点を、WayPoint 要素で指定することによって、一本の道を表示する。このとき、道の距離や通り抜け難さを表現するために、その道の移動にかかる時間を Cost という単位で示す。
- Information
その LogicalArea 固有の情報を表す。具体的な内容は定められておらず、それぞれの LogicalArea によっても情報は異なる。
例えば、店舗を表現している LogicalArea の場合、Information には、その店が扱っている品物についての情報や開店時間などが入る。
- ChildArea
LogicalArea 内に存在している子領域群を表す。ChildArea の構造は LogicalArea の構造と同一であり、GSLX 全体としてツリー状の入れ子構造を構成する。

6. uGuidanceFramework の設計

本節では、uGuidanceFramework の設計について述べる。まず、本システム全体のソフトウェア構成について述べ、次に各層の処理について述べ、最後に処理の流れについて述べる。

6.1 ソフトウェア構成

uGuidance Framework のソフトウェア構成を図 1 右に示す。この中で uGuidance Framework が提供するものは、LocationModel 層の処理を行うモジュールである GSLX Container と、各層の標準インターフェースである。

このフレームワークによって各層間をやり取りするデータは、領域、位置、経路、方向を意味する 4 つのエンティティオブジェクトである。これらのエンティティオブジェクトはそれぞれ、LogicalArea、Position、Route、Direction と呼ばれている (表 2 参照)。

6.2 Measurement 層

Measurement 層の役割は、センサからの情報や、計算によって生成された情報を LocationModel 層に受け渡すことである。センサから得られるような生データは、Lo-

エンティティ	要素	説明
LogicalArea	GSLX の要素と同一	
Position	X	x 座標
	Y	y 座標
	LogicalArea	座標系の属する領域
	Layer	その地点の高さ
Route	Precision	地点を示すセンサの精度
	Junction 配列	経路
Direction	Direction	角度

表 2 エンティティオブジェクトの構成要素

メソッド	機能
getCurrentPosition	現在地の取得

表 3 LocationInterface の機能

メソッド	機能
getRoute	経路の取得

表 4 RouteInterface の機能

メソッド	機能
getDirection	角度の取得

表 5 DirectionInterface の機能

cationModel 層で計算を行うことによって、GSLX によって定義されるような形に整形された後、LocationModel 層に返却される。

Measurement 層のインターフェースは、位置検出、角度検出、経路計算それぞれのモジュールごとに提供される。それぞれのインターフェースの機能を表 3、表 4、表 5 に示す。

6.3 LocationModel 層

LocationModel 層の役割は、ロケーションモデルに従って位置情報に関する処理を行うことである。なお、uGuidance Framework では GSLX Container というコンポーネントによって LocationModel 層の処理を行っている。

LocationModel 層のインターフェースには ModuleInterface、ApplicationInterface、BaseInterface の 3 種類がある。ModuleInterface は Measurement 層との間のインターフェース、ApplicationInterface は BusinessLogic 層との間のインターフェースである。アプリケーションが ApplicationInterface を介して位置情報を取得することによって、BusinessLogic 層から Measurement 層を隠蔽している。

また、Measurement 層と BusinessLogic 層のいずれの層でも必要な機能は BaseInterface が提供する。主に BaseInterface ではロケーションモデルに関する処理を行う。

表 6 に BaseInterface の、表 7 に ModuleInterface の、表 8 に ApplicationInterface の主要な機能を示す。

なお ApplicationInterface には、3 つのイベント通知機能が備わっている。表 9 にイベント通知を行うイベントリスナの機能を示す。

AreaComesToNearEvent は、例えば商店街の広告配信アプリケーションを想定している。商店街を歩いていると、端末を持っている客の近くにある店舗の情報がイベントとして通知されるので、容易にアプリケーションが開発できる。

AreaInOutEvent は、部屋に入ると自動的にライトが

getRootArea	ルートエリアの取得
findAreaById	エリアの検索
getParentArea	親エリアの取得
getAreasFromPosition WithCost	近傍にあるエリアの取得
getAreasFromPosition	Position の存在しているすべてのエリアの取得
convertPositionOfArea	Position のエリアを変換
getChildAreasFromArea	子エリアの取得

表 6 BaseInterface の機能

installModule	モジュールを LocationModel 層にインストール
getModules	インストールされたモジュールを取得

表 7 ModuleInterface の機能

getCurrentPosition	現在地を取得
getRoute	経路を取得
getDirection	角度を取得
addEventListener	イベントリスナを BusinessLogic 層にインストール

表 8 ApplicationInterface の機能

AreaComesToNearEvent	
notifyAreaComesToNear	近くにエリアが近づいてきたことを通知
AreaInOutEvent	
notifyAreaEnter	エリアに入ったことを通知
notifyAreaExit	エリアから出たことを通知
RouteTraceEvent	
courseSeparated	ルートから離れたことを通知
junctionPassed	中継点を通じたことを通知
destinationArrived	目的地に着いたことを通知

表 9 イベントリスナの動作

点くなどの、領域への出入りがトリガとなって動作するアプリケーションを想定している。

RouteTraceEvent は、カーナビゲーションシステムのような使い道を想定している。最初に、目的地までの経路を設定しておく、曲がり角のような方向を指示する必要のある中継点や、目的地に到達したことを通知してくれる。また、ルートから離れたことも通知してくれるので再設定することもできる。

このようなイベントリスナ機能を GSLX Container に付加することで、イベントドリブなアプリケーション開発を容易にする。

なお、GSLX Container は GSLX によって記述された位置情報を各層に提供する機能も持っている。

6.4 処理の流れ

本節では、uGuidance Framework における処理の流れについて述べる。

アプリケーションの起動シーケンス図を図 3 に示す。

Startup モジュールによって各層に位置するモジュールが生成された後、BusinessLogic 層に位置するアプリケーションに対して init メッセージを送ることによって、処理をアプリケーションに渡す。

7. 評価と課題

我々は、以上のような設計に基づいて、GSLX と uGuid-

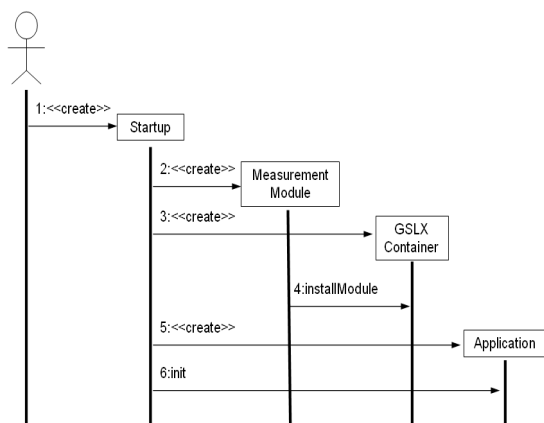


図3 アプリケーションの起動シーケンス

ance Framework の実装を行った。また、SmartPAO¹¹⁾と呼ばれる多くの情報機器が埋め込まれたユビキタス空間で動作する、簡単なアプリケーションを実装し、実証実験を行っている。

本節では、3.3節で挙げた機能要件に基づいて、uGuidance Framework の定性的な評価と今後の課題について述べる。

7.1 評価

● 重複ロジックの統合

アプリケーションは、処理の大部分を Application Interface を通して、uGuidance Framework に委託することができた。特に、フレームワークのイベント通知機能が、アプリケーションの開発にとって有効であった。

● モジュールの統合

SmartPAO など、多くのセンサデバイスが埋め込まれた環境では、新しいセンサデバイスが追加されることが多い。Measurement 層のモジュールをアプリケーション開発者から隠蔽したことは、運用コストを下げることができ、また保守性にも優れていることが実証された。

● モジュールの選択

現状では、無線 LAN を用いた低コストだが精度の悪い測位手法と、RFID を用いた高コストだが精度の良い測位手法を、場所によって使い分けるようなアプリケーションが考えられる。そういったとき、アプリケーション側でモジュールの使い分けを意識する必要がなくなる。

7.2 課題

● モジュールの受け渡し

複数のモジュールからのデータを BusinessLogic 層に受け渡すとき、アプリケーションの要件によっては、LocationModel 層でマージするのではなく、それぞれ単体での結果を受け渡したい場合がある。例えば、経路モジュールの場合、最短経路を取得するモジュールと、最も人通りの少ない経路を取得する

モジュールは、どちらが適切なのかは BusinessLogic 層の判断である。

● 位置検出モジュールの取得タイミング

例えば、無線 LAN を用いた測位方法のように、常に現在地を取得できるモジュールと、RFID を用いた測位方法のように、ある領域に入ったときのみ現在地を取得できるモジュールがある。

これら取得タイミングの異なるモジュールをどのようにして BusinessLogic 層に渡すべきなのかについては、議論の余地がある。

8. 終わりに

本稿では、Location-aware アプリケーションのためのフレームワーク、uGuidance Framework と、フレームワークが利用するロケーションモデル、GSLX を提案した。今後は、uGuidance Framework を利用したアプリケーションの開発を引き続き行い、実証実験を進めていくことで、より現実的な環境での本フレームワークの有効性を検証していく。

参考文献

- 1) 矢部宏樹, 佐野睦夫. 複数センサを用いた信頼度マップに基づく個人同定・状態検出. 情報処理学会第 66 回全国大会, 2004.
- 2) Ekahau Positioning Engine. <http://www.ekahau.com>.
- 3) AirLocation. <http://www.hitachi.co.jp/Prod/vims/solutions/ssup/airlocation>.
- 4) AeroScout. <http://www.aeroscout.com>.
- 5) 石井健一, 森崎充敬, 松田淳一, 倉島顕尚, 山崎俊太郎. ハイブリッド無線 lan 測位システムの提案 (2) ~ 照明位置タグシステムの提案 ~. 電子情報通信学会, 2004.
- 6) 高田敏弘, 青柳滋己, 栗原聡, 光来健一, 清水奨, 廣津登志夫, 福田健介, 菅原俊治. センサデータのマイニングによるロケーションモデルの構築とその応用. 情報処理学会 ユビキタスコンピューティングシステム研究会研究報告, No. 39(2003-UBI-1), pp. 87-94, Apl 2003.
- 7) Changhao Jiang and Peter Steenkiste. A hybrid location model with computable location identifier for ubiquitous computing.
- 8) U. Leonhardt. Supporting location-awareness in open distributed systems, 1998.
- 9) J. Hightower, B. Brumitt, and G. Borriello. The location stack: A layered model for location in ubiquitous computing. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002)*, pp. 22-28, Callicoon, NY, June 2002. IEEE Computer Society Press.
- 10) XML Schema. <http://www.w3.org/XML/Schema>.
- 11) Smart PAO. <http://www.uchida.co.jp/ubiquitous>.