

スマートスペースのプログラミングモデル

佐 藤 一 郎[†]

本論文ではユビキタスコンピューティング環境における位置依存サービスおよびユーザ依存サービスを実現するうえで基礎となる空間モデルを提案する。この空間モデルの特徴は、物理世界の建物、部屋、ユーザなどを空間的含有関係をもとにして、それぞれに対応する仮想実体の木構造として表現するだけでなく、サービスを実現するソフトウェアやそれを実行するコンピュータの位置や特性も空間モデル上で統一的に扱えることである。仮想実体はソフトウェアコンポーネントとして与えられ、位置およびユーザ依存サービスを実現するソフトウェアと区別せずに空間モデル上で扱われる。また、このモデルでは空間モデルを維持・管理するデータベースが不要なため、計算リソースが少なく、即興的なシステム構成が要求されるユビキタスコンピューティングにおいて有用となる。本論文では空間モデルの基本概念とその設計、さらに同モデルに基づくプロトタイプを実装および応用例を示す。

A Programming Model for Smart Spaces

ICHIRO SATOH[†]

This paper presents a world model for location-aware and user-aware services in ubiquitous computing environments. The model can be dynamically organized as a tree based on geographical containment, such as user-room-floor-building according and construct each node in the tree to as a self-contained and mobile software component. The model is unique among other existing approaches because it enables location-aware services to be managed without any databases, it can be managed by multiple computers, and it can provide a unified view for the locations of not only physical entities, including users and objects, and spaces but also computing devices and services. A prototype implementation of this approach was constructed on a Java-based mobile agent system. This paper presents the rationale, design, implementation, and applications for the prototype system.

1. はじめに

Mark Weiser¹⁸⁾ が提唱したユビキタスコンピューティングの目標の1つは現実世界に対応した計算サービスを提供することにある。ユビキタスコンピューティングは各種機器や環境に組み込まれたコンピュータだけでなく、各種センサも接続されることから、計算サービスは物理世界のコンテキスト、たとえばユーザは誰なのか、そのユーザがどこにいるかなどの情報を反映して、サービス内容の適応やサービスの実施場所の変更が求められる。このため、ユビキタスコンピューティングにおけるコンテキスト依存サービスを扱えるシステムが数多く提案・開発されている。しかし、その多くは道案内などの特定のサービスに特化して設計・構築されており、他のサービスへの転用や他のサービスとの同時利用は困難となっている。また、

特定のセンサ種別や製品に依存しているシステム事例も多く、設計当初想定していたセンサ以外では利用できず、さらにサービスの設計・開発においてセンサを含む低水準制御・監視に関する知識が必要であり、開発者への負担も大きかった。

本論文ではスマートスペースなどの室内のユビキタスコンピューティング環境を扱う。これはコンテキスト依存サービスのうちで最も代表的である位置依存サービスおよびユーザ依存サービスを対象に、センサや機器を抽象化するとともに、多様なサービスを統一的に実現・稼働させる空間モデルを構築し、上述の既存コンテキスト依存サービスの問題点の解決をはかっていく。

以降の論文構成を述べる。本論文の目的は空間モデルの提案と設計を示すことである。そこで次章では空間モデルに要求される表現性と機能を示し、3章では空間モデルの設計を述べ、4章ではその実現方法を概説する。そして5章では空間モデルの応用例を、6章では関連研究との比較を示し、最後の7章では結論と

[†] 国立情報学研究所

National Institute of Informatics

課題を述べる．

2. 研究背景

本論文ではユビキタスコンピューティングを前提とした空間モデルを提案し、そのモデルをもとにして位置およびユーザ依存サービスを実現する方法を考える．

2.1 要 求

これまで空間モデルはデータベースや地理情報システムを中心に研究されてきた．既存のモデルは緯度経度などの座標情報によるものと、建物や部屋などにつけられた名前や番号などの記号情報によるものに大別できる⁸⁾．本論文では室内におけるサービスを対象とするが、前者は室内では超音波センサやコンピュータビジョンなどの高精度センサが必要となるが、こうした高性能センサは一般に高価であり、さらに測定エラーが多いものや、キャリブレーションなどの管理上の手間を要するものも多く、広範な普及が望めるとはいえない．後者はRFID タグ¹⁵⁾などの安価なセンサで実現でき、その設置・維持も簡単である．さらに室内の位置およびユーザ依存サービスの多くは部屋などの所定空間におけるユーザの存在有無で十分で、室内における詳細な座標情報は不要であることが多い．

このほか、室内向けサービスの有効範囲は空間の含有関係を依存することが多い．たとえば典型的な位置依存サービスである案内システムでは、ユーザが現在いる部屋と階、建屋全体の案内図は提供されるべきだが、セキュリティや安全上の問題からその階に含まれる他の部屋や他の階の情報の提供は避けるべきである．そこで、記号による空間表現に加えて、他の位置依存サービス向け空間モデル⁸⁾と同様に、建家、階、部屋などの各空間の含有関係も扱えるようにし(図1)、さらにある空間で利用可能な情報やサービスはその空間に加えて、それを含有する空間(入れ子的にその含有する側の空間を含有する空間も含む)において利用可能なものに制限する．なお、現在の空間とは含有関係にない空間、たとえば隣接する部屋で提供される情報やサービスを利用する場合は、サービスを提供する側に対応した空間モデルを介して実現できるようにして柔軟性を維持する．

一方、ユビキタスコンピューティングにおいて位置およびユーザ依存サービスを考えた場合、下記のように一般の空間データベースや地理情報システムにはないユビキタスコンピューティング特有の制限や要求が

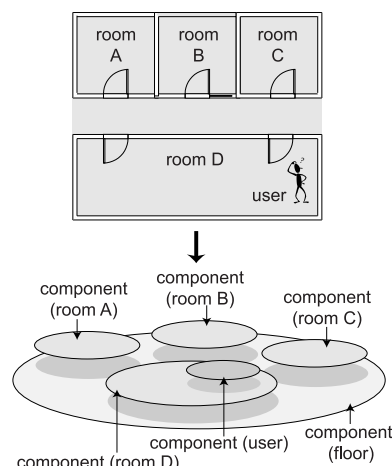


図1 物理空間・実体の配置と空間モデルの含有関係

Fig. 1 Containment relation based on spaces and objects in physical world.

あり、それらを満足する必要がある．

- サービスの実施位置：位置およびユーザ依存サービスではユーザやそのユーザの存在位置だけでなく、サービスを実現するソフトウェアを保持するコンピュータと、それらのコンピュータの存在位置や計算資源も把握する必要がある．
- 計算リソースの制限：サービス内容はコンテキスト変化に応じて変更する必要がある．しかし、ユビキタスコンピューティング環境におけるコンピュータは記憶領域を含む計算リソースが少なく、あらかじめ複数のサービスを格納・稼働しておくのは困難である．
- データベースサーバの不在：既存システムでは関係データベースなどの汎用データベースを用意し、空間モデルをそのデータベース上で維持・管理することが多い．しかし、ユビキタスコンピューティングによる計算システムでは専用データベースやそのデータベースを維持する外部サーバが利用できるとは限らない．

2.2 設計方針

上述した要求を満足するため、本論文で提案する空間モデルは物理世界の空間および実体の含有関係を扱うようにするとともに、既存研究にはない次のような特徴を与える．

- 空間モデルがサービスを実現するソフトウェアも含有できるようにする．物理空間の変化に応じて、これらのソフトウェアも自動再配置可能にする．また、サービスを実現するコンピュータの位置や移動もモデル化するとともに、コンピュータや外部システムに対応するプロキシを与えることで、

プライバシー保護の観点から、むしろ詳細な行動把握は避けたい方がよいことが多い．

コンピュータや外部システムを空間モデルの一部として扱えるようにする。

- 空間モデルはサーバなどの特定のコンピュータだけではなく、サービスを実現するコンピュータを含めた複数コンピュータ間で分担管理が可能にする。また、上述のプロキシ機能を利用することにより、あるコンピュータ上で維持・管理される空間モデルを他のコンピュータ上の空間モデルの一部として扱えるようにする。
- 空間モデルでは物理実体や空間とサービスを実現するソフトウェアを区別せずに統一的に扱う。また、物理空間上の物理実体やコンピュータの移動は、空間モデル上の仮想実体の移動として表現する。なお、空間モデルが複数コンピュータから構成される場合は、移動元の空間モデルと移動先の空間モデルが異なるコンピュータで維持されることがある。そこでモバイルエージェント技術を利用することにより、空間モデルおよびサービスを実現するソフトウェアにコンピュータ間移動性を導入する。

なお、室内空間向けの位置センサとして、RFID タグや赤外線タグ、コンピュータビジョン、超音波センサなどが利用されることが多い。しかし、これらのセンサはそれぞれ特徴があり、その特徴をすべて活かしながら統一的に扱うことは困難である。たとえば超音波センサは数センチの誤差で 3 次元座標が取得できるものがあるが、アクティブ RFID や赤外線タグなどは部屋などの空間内におけるタグの存在の有無しか判定できない。そこで、この空間モデルは、広く利用されている中でも低精度のセンサ、たとえば RFID や赤外線タグなどを前提にし、座標情報などの含有関係以外の測定情報は直接サービスを実現するソフトウェアに送る機構を用意することで、多様なセンサを扱えるようにする。

3. 空間モデルの設計

本論文の空間モデルは物理空間や物理実体だけでなく、サービスやコンピュータもモデル上で統一的に扱える必要があり、既存の関係データベースを利用することは難しいことから、ソフトウェアコンポーネントの集合体として空間モデルを構築していく。

3.1 空間モデルの構成

このモデルでは空間構成を含有関係としてモデル化すること、サービスを実現するソフトウェアを扱えること、複数コンピュータからモデルを構成できる必要があることから、コンポーネントは階層性を持つモバ

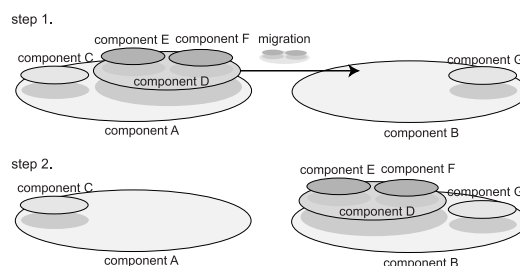


図 2 階層コンポーネントの移動

Fig. 2 Migration of hierarchical components.

イルエージェントシステムである MobileSpaces¹²⁾ を利用し、空間モデルはエージェント階層関係に基づく下記のようなコンポーネント間含有関係として実現する (図 2)。

- コンポーネントは 0 個以上の他のコンポーネントを含有できる。
- コンポーネントが移動する場合はそれに含有されているコンポーネントもともに移動する。

コンポーネント間の関係は木構造となり、たとえば建物が複数階を持つことはそれぞれの階に対応したコンポーネントが建家に対応するコンポーネントに含有され、さらに各階上の複数の部屋があるときはそれぞれの部屋に対応するコンポーネントが階に対応するコンポーネントに含有されることになる。

3.2 空間モデルの構成

コンポーネントは以降では説明を明確化するために次の 3 種類に分けて説明していく。

- Virtual Component (VC)** は空間モデル上の仮想実体であり、モノや人などの物理実体や建家や部屋などの物理空間に対応するコンポーネント。
- Proxy Component (PC)** はコンピュータのプロキシとなるコンポーネントである。
- Service Component (SC)** はサービスを実現するソフトウェアに対応するコンポーネント。

なお、各コンポーネントは各種属性を保持するデータだけでなくプログラムコードとその変数領域から構成されるが、モバイルエージェント技術を利用することにより、コンポーネントは再配置前後ではプログラムコードだけでなくインスタンス変数などの実行状態を保持し、再配置後のコンピュータでも移動前の状態からサービスを継続できるようにする。

3.2.1 Virtual Component (VC)

VC は物理空間または実体ごとに用意され、その空間的な含有関係に応じて VC 間の含有関係を持ち、空間または実体の移動はそれに対応する VC の含有先の変更としてモデル化される。たとえば人が自動車に乗

るとは、人に対応した VC が自動車に対応した VC に含有されることになる。また、自動車の走行とは自動車に対応した VC を人に対応した VC とともに、移動元の空間に対応する VC から移動先の空間に対応した VC に移動することになる。こうした含有関係は明示的に与えることもできるが、それ以外に RFID タグなど位置センシングシステムを利用することにより、物理世界の変化に応じて VC 間の含有関係も動的に変更できるようにする。たとえば RFID タグを利用する場合、RFID リーダなどの位置検知領域も VC としてモデル化され、その領域に RFID タグが入ったときはタグが貼付されたモノに対応する VC が領域の VC に含有されることになる。

3.2.2 Proxy Component (PC)

PC はコンピュータを空間モデル上で扱うためのコンポーネントであり、空間モデル上のコンピュータのプロキシとして機能する。なお、PC はそれが参照するコンピュータの機能や役割に応じて 3 種類、PCM (PC for Model manager), PCS (PC for Service provider), PCL (PC for Legacy device) がある。なお、PC はそれが参照するコンピュータの存在位置に対応する VC 上に含有されるが、PC 側で含有位置を明示的に指定できる。この場合、PC が参照するコンピュータは空間モデルの対象となっていない外部システム上にあってもよい。

- PCM は空間モデルへの参照となるコンポーネントである。具体的には図 3 (a) のように、あるコンピュータ (computer 1) の空間モデル (model 1) の PCM が、コンピュータ (computer 2) 上の空間モデル (model 2) 上に含有されるとき、前者の空間モデル (model 1) は後者の空間モデル (model 2) の一部となり、PCM の存在位置に前者の空間モデルが含有されているとして扱う。この結果、空間モデルが異なるコンピュータにより維持されていても、1 つの連続した空間モデルとして扱えることになる。
- PCS は SC を実行できるコンピュータに対応するコンポーネントである。つまり、PCS は SC を受け取ると、それが対応するコンピュータに SC を転送して、SC の処理をそのコンピュータに代行させる。具体的には図 3 (b) のように PCS はそれが対応するコンピュータ (computer 1) に、受け取った SC を移動させて、SC を実行させる。このとき空間モデル上では SC は転送後も転送前の位置、つまり PCS 内にあるとして扱われる。
- PCL は外部サービスや SC が実行できないコン

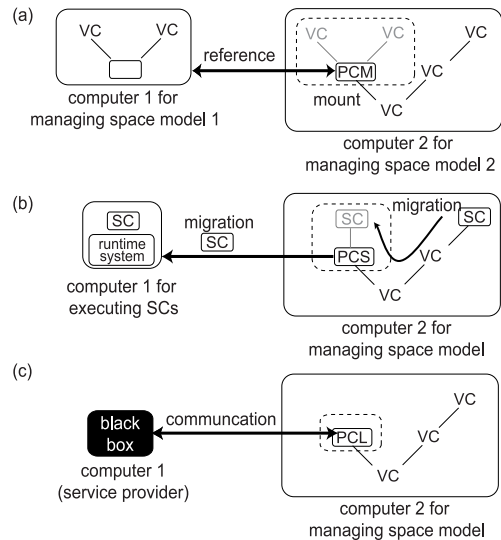


図 3 Proxy Component (PC) の動作
Fig. 3 Proxy Component (PC).

ピュータの代理となるコンポーネントである。すべてのコンピュータが SC を実行できるわけではないが、それらを空間モデル上で統一的に扱うために導入される。たとえば図 3 (c) において PCL はそれが代理となっているコンピュータ (computer 1) への処理要求を送信するとともにその処理結果を受信し、そのコンピュータがそれを含有するコンピュータ (computer 2) の空間モデル上にあるかのように見せかける。ここで PCL とコンピュータの間の通信は任意とし、PCL 側で代理となるコンピュータとの相互作用に必要な通信プロコルを実装することができる。

たとえば空間モデルを維持するコンピュータの PCM を他のコンピュータの空間モデル上に含有させることにより、前者の空間モデルは後者空間モデルの一部となり、複数コンピュータにより 1 つの空間モデルが構築できるようになる。なお、1 つのコンピュータが異なる物理空間に対応した空間モデルを維持してもよいが、このときそれぞれの物理空間を含有する物理空間に対応する VC に、それぞれの空間モデルの PCM が含有されることになる。PCS によりコンピュータに対してサービスを実現するソフトウェアを動的に配置・実行できるようになる。PCL は既存のサーバや組み込み機器などの本枠組みに直接対応できないコンピュータを扱うための機構であり、たとえば TV に対応した PCL はそのテレビへのコマンドを受け付けると、PCL が配置されたコンピュータから赤外線リモコン通信などを通じてテレビを制御する。また、PCL

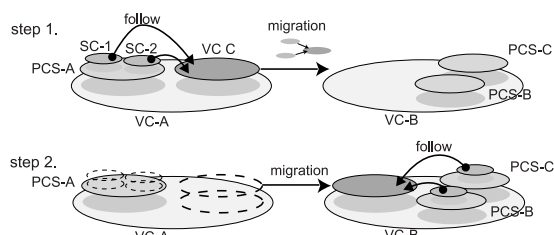


図 4 連結移動

Fig. 4 Following the movement of another component.

が参照するコンピュータは外部システム上のサービス、たとえばインターネット上のサービスでもよい。また、PCL はプログラマブルなソフトウェアであることから、相手先コンピュータの制御だけでなく、そのサービスのカスタマイズも可能である。

3.2.3 Service Component (SC)

SC は 1 つ以上のサービスを実装したプログラムコードおよびそのデータ領域の集合体となる。PCS を介してコンピュータ上に転送・実行される。なお、SC は再配置可能であり、VC に対して連結移動が可能である。たとえばユーザへの何らかの支援サービスを実現する SC はユーザに対応する VC に連結されて移動することになる。なお、実装上は VC もプログラマブルな計算実体であり、VC 上にサービスを直接定義することができるが、空間モデルとサービスの定義を分割することと、JavaBeans などの既存のソフトウェアコンポーネントを空間モデルとの親和性を保ちながら導入する方法として VC とは別のコンポーネント、つまり SC を導入している。

なお、SC はそれを含有する VC とともに移動することになるが、本論文ではその含有関係に加えて、SC と VC 間の疎結合を実現する方法としてコンポーネント間の連結と呼ぶ機構を導入する。SC はたかだか 1 つの VC に対する連結を指定することができる。このとき連結先の VC が移動するとその SC は、連結先の VC を含有する VC にある PCS に移動することになる。たとえば図 4 をもとに説明すると、2 つの SC (SC-1 と SC-2) がコンポーネント VC-3 に連結されているとする。このとき VC-3 が現在位置の VC (VC-1) から別の場所の VC (VC-2) に移動すると、SC-1 と SC-2 は VC-2 上にあるそれぞれの要求を満足するプロキシコンポーネント (PCS-B と PCS-C) に移動し、PCS-B と PCS-C が対応するコンピュータに転送・実行されることになる。

4. 実 装

ここでは前章までに議論した空間モデルの実装について説明する。なお、空間モデルそのものはコンポーネントの実装言語を限定しないが、現在の実装は Java 言語 (J2SE 1.2 以上) と同言語によるモバイルエージェントシステムである MobileSpaces 上に構築されている。

4.1 空間モデルの維持・実行

コンポーネントの実行は MobileSpaces の実行システムを拡張して実現される。ここでは本論文と関わる部分と拡張部分を中心に説明していく。

4.1.1 コンポーネント含有関係の管理

実行システム上で各コンポーネントはプログラムコードおよびデータに加えて名称やサイズなどのコンポーネントに関する属性を保持したノードとなり、前述のようにコンポーネント間含有関係は対応するノードの木構造として実現される。また、コンポーネントの移動は、そのコンポーネントとそれに含有されるコンポーネント、つまりそのコンポーネントをルートとする部分木を他の部分木に接続することである。

なお、3 章で述べたコンポーネントの連結移動は連結する側のコンポーネントと連結される側のコンポーネントを含有する仮想コンポーネントを導入し、前者が移動するときはそれ単体で移動し、後者のコンポーネントが移動する場合は仮想コンポーネントごと移動する、つまり連結される側だけでなく、連結した側も移動することで実現する。

4.1.2 コンポーネントの実行管理

空間モデルを構成する部分木が移動・変形するとき、実行システムは表 1 に示されたようにイベントをコンポーネントに送り、コンポーネント側に定義されたコールバックメソッドを実行することになる。なお、コンポーネントは 1 つ以上の実行スレッドを持つことができる。ただし、コンピュータ間移動の前後でコンポーネント自身によりスレッドの停止・開始処理を行うとする。このほか、コンポーネントが含有されているコンポーネントに別のコンポーネントが移動してきたときにもイベントを受け取ることができる。

4.1.3 コンピュータ間移動の実現

コンポーネントが異なるコンピュータに移動すると

MobileSpaces に関する説明は論文 12) を参照されたい。なお、後述するように SC は JavaBean コンポーネントとして実現されてもよい。

コンポーネントはそれ自身およびそれに含有されるコンポーネントに移動することはできない。

なお、PCS-B と PCS-C と移動した場合は SC-1 と SC-2 は VC-3 への連結をもとに VC-2 上で再配置することになる。

表 1 コンポーネントイベント
Table 1 Component events.

イベント	タイミング	イベント対象
create	生成直後	生成対象の全コンポーネント
destroy	終了直前	移動対象の全コンポーネント
departure	移動直前	移動対象の全コンポーネント
leave	移動直前	移動元コンポーネント
join	移動直後	移動先コンポーネント
arrive	到着直後	移動対象の全コンポーネント
suspend	整列化直前	整列化対象の全コンポーネント
resume	非整列化直後	非整列化対象の全コンポーネント

きは、移動前にそのコンポーネントとそれに含有されるすべてのコンポーネントを通信可能なデータ形式に整列化 (marshalling) することになる。まず suspend イベントを整列化対象となるコンポーネントに送信して、獲得している計算リソースの解放やインスタンス変数の保持を行わせる。そして Java 言語のオブジェクト直列化 (Object Serialization) 機能を利用してヒープ領域上のインスタンス変数を整列化し、そのコンポーネントのプログラムコードともに整列化したデータを送信する。そして宛先では再びコンポーネントに逆変換し、resume イベントを整列化対象となるコンポーネントに送信して計算リソース獲得などの処理を行わせる。なお、通信には MobileSpaces が提供する TCP/IP による転送機構を利用しているが、ユーザに対応する VC などユーザプロファイルなどプライバシー保護や守秘性が要求される情報を保護することが求められる。この場合、MobileSpaces が提供する機構、たとえばエージェント単位で暗号化して転送する機構や SSL を介したモバイルエージェント転送機構¹⁴⁾ を利用することにより、転送中の情報を保護することができる。

4.2 コンポーネント

現在の実装では各コンポーネントは MobileSpaces のモバイルエージェントとして実現され、具体的には下記の Component クラスのサブクラスとして定義される。

```
class Component extends MobileAgent {
    ContextInfo getCurrentContextInfo() { ... }
    ServiceInfo[] getServices(String name) { ... }
    ServiceInfo[] getAncestorServices(
        String name) { ... }
    Object execuService(ServiceInfo si, Message msg)
        throws NoSuchServiceException { ... }
    ....
}
```

コンポーネントが提供するサービスは後述する SC として定義される。コンポーネントはそれを含有しているコンポーネントに含まれる別のコンポーネント、つま

り隣接するコンポーネントのサービスを利用することができる。コンポーネントが getServices メソッドを実行することにより、隣接コンポーネントのサービスを取得することができる。また、getAncestorServices メソッドにより、コンポーネントはそれを含有するコンポーネントについても再起的に木構造のルート方向に向かってサービスを検索できる。

4.2.1 Virtual Component (VC)

各 VC は下記のような VirtualComponent クラスのサブクラスとして定義される。

```
class VirtualComponent extends Component {
    void setPhysicalCode(PhysicalCode pc)
    void setName(String name) { ... }
    void setSize(Size s) { ... }
    void setAttribute(String attribute,
        String value){ ... }
    String getAttribute(String attribute) { ... }
    ....
}
```

ここで setPhysicalCode メソッドは VC が対応する物理空間や実体に割り当てられた識別子であり、センサなどが管理に利用する番号などが格納される。なお、特定の識別子形式は規定しないが、RFID タグが貼られた物体であればタグの識別子が格納される。また、setName メソッドは空間や実体に割り当てられた名称を、setSize メソッドは空間や実体のサイズを、setAttribute メソッドは空間または実体に固有の属性情報を登録するときに呼び出す。

4.2.2 Service Component (SC)

位置およびユーザ依存サービスを定義するコンポーネントであり、ServiceComponent クラスのサブクラスとして定義されるが、なお、JavaBean を SC として扱うコンテナ (Container) を提供しており、直列化可能な JavaBean であれば SC として扱える。

```
class ServiceComponent extends Component {
    void setName(String name)
    Host getCurrentHost() { ... }
    Hosts[] getNeighboringHosts() { ... }
    Host[] getCandidateHosts(Host[] hosts) { .. }
    void move(ComponentID id) throws
        NoComponentException { ... }
    void follow(FollowPolicy fp, ComponentID id) throws
        NoComponentException { ... }
    void setComponentProfile(
        ComponentProfile cpf) { ... }
    ....
}
```

上記の ServiceComponent クラスでは、SC は setName メソッドによりそれ自身のサービス名を登録することができる。また、SC は CC/PP (Composite Capability/Preference Profiles)¹⁹⁾ に準じた形式で、

実行に必要なコンピュータの計算リソースを記述し、`setComponentProfile` メソッドにより登録する。また、`getCandidateHosts` メソッドにより、現在の VC 上でその計算リソースを満足するコンピュータの一覧を得ることができる。`follow` メソッドにより、連結先となるたかだか 1 つの VC を指定して、その VC の移動に追従することや、`move` メソッドにより明示的にそれ自身の移動先を指定・移動することもできる。ところで、この枠組みでは部屋や RFID リーダの識別領域などの空間への含有関係を対象とし数量的な座標は扱わない。しかし一方で位置およびユーザ依存サービスはユーザとサービスを実現するコンピュータとの位置関係が重要となることがある。そこで SC は移動先の VC が `VirtualComponent` クラスの `setSize` メソッドで対応する空間のサイズを指定している場合は、たとえばユーザとコンピュータ間の距離はその空間サイズよりも小さいか否かを判断し、ユーザに近傍のコンピュータを選ぶこともできる。

4.2.3 Proxy Component (PC)

PC は空間モデル上に配置され、そのコンピュータがあたかも空間モデルの一部になっているかのように見せかける。前述のように PC には対応するコンピュータにより 3 種類があるが、そのうち SC を実行するコンピュータに対応する PC と空間モデルを維持するコンピュータに対応する PC について、`MobileSpaces` におけるコンポーネント転送機構を利用して実装される。この機構はコンポーネントを受け取ると、あらかじめ定められたコンポーネントにその受け取ったコンポーネントを転送させるものであり、PC は SC を含有するとコンピュータに自動的に転送して、コンポーネントの移動上に PC とそのコンピュータを区別できないようにする。このほか、SC を実行するコンピュータは CC/PP の形式により提供可能な計算リソースを登録することができる。また、コンピュータのプロキシとなる PC はそのコンピュータが存在する空間に対応する VC に含有され、処理要求をコンピュータに転送する。なお、コンピュータと PC 間の通信方法は個々のコンピュータの通信機能に依存することから、本枠組みはその通信方法を規定・提供しないが、TCP/IP ソケットや HTTP 接続について補助ライブラリを用意している。

4.3 物理世界の反映

建家や部屋などに対応した VC はあらかじめ含有関

係が与えられることになるが、人やコンピュータを含む物理実体の特定や位置を、`Location-aware Component Deployment Manager (LCDM)` と呼ぶ機構によって動的に空間モデル上に反映することもできる。これは著者が提案した位置依存モバイルエージェントの枠組み¹¹⁾における `Location Information Server (LIS)` と同様の機能を提供し、1 つ以上の位置センサを監視して物理実体の位置に応じて実体に対応したコンポーネントを再配置するものである。LCDM はそれが管理する 1 つ以上のセンサの情報と、センサの測定範囲に 1 つ以上の空間を定義し、その空間に対応した VC を保持するコンピュータのネットワークアドレスを管理するデータベースを持つ。なお、LIS は物理実体が RFID リーダの識別範囲内または範囲外の 2 つしか扱わなかったが、LCDM では各センサの測定領域に対応した VC を導入し、その測定領域を含有する空間に対応した VC に配置する。そして、測定領域で新しい物理実体の発見や移動を捕捉するとその領域に対応する VC 上にその実体に対応する VC または PC を (再) 配置する。なお、室内向け位置センサの場合は測定領域は部屋などの物理空間内となることが多いことから、測定領域の VC はつねに他の物理空間の VC に含まれるとする。

センサが物理実体、たとえば人やモノ、コンピュータがある物理空間から別の物理空間に移動したことを発見するとその実体に対応する VC または PC を移動元空間の VC から移動先空間の VC に移動させる。なお、ここでは RFID タグ技術により位置捕捉をする場合のコンポーネントの再配置を概説する。ここでは対象となる物理実体には RFID タグがあらかじめ貼られているとする。また RFID リーダの測定領域の重なっていない場合を扱う。

- (1) LCDM の RFID リーダがこれまで存在しなかったタグの識別子を発見すると、それ自身のデータベースからそのリーダに対応する VC の存在位置を見つける。
- (2) LCDM はマルチキャスト通信により空間モデルを維持するコンピュータに識別子に対応する VC または PC を保持しているかを問い合わせるメッセージを送信する。
- (3) 識別子に対応し VC または PC を保持しているコンピュータはそれ自身のネットワークアドレスを LCDM に返答する。逆に該当するコンピュータ

なお、`follow` メソッドの引数の 1 つである `FollowPolicy` クラスはコンポーネント再配置機構¹⁶⁾と同様であり、コンポーネント配置範囲や複製作成の有無を指定することができる。

センサの監視機構およびセンサ間相違の隠蔽機構は LIS と同様のため、これらの機構は既発表論文¹¹⁾に譲り、ここではコンポーネントの配置機構のみを説明する。

がないときは隣接する LCDM に問合せメッセージを送る。問合せメッセージを受けた LCDM は (2) から処理を行う。

- (4) LCDM は識別子に対応した VC または PC に、RFID リーダの測定範囲に対応した VC の名前とそれを保持するコンピュータのネットワークアドレスを含むメッセージを送信して、その RFID リーダの VC 上に移動するように促す。

ここで、各コンピュータには DHCP によりネットワークアドレス (IP アドレス) が割り当てられているとする。なお、不要なメッセージを最小化するため、問合せメッセージには最大転送ホップ数とメッセージ識別子が与えられており、LCDM はすでに受け取ったメッセージおよび転送ホップ数に達している場合はさらなる転送を行わない。また、タグが不在を発見した場合はタグに対応する VC または PC (特に PCM と PCS) にメッセージを送り、その VC または PC に指定されているホームコンピュータの所定コンポーネントに移動または休止させる。また、PCL に対応するコンピュータの場合は、コンピュータはメッセージが受信・処理できるとは限らないことから、それを参照する PCL にコンピュータの後処理を任せる。なお、物理実体が空間モデル上で発見されたとき、それに対応する VC の発見は、最悪の場合、空間モデル上の全検索になる可能性があるが、実際には物理実体の移動は建物の出入り口などに限られることから、そうした出入り口を対象としている LCDM が、建物への入出が想定される物理実体に対応した VC の待機場所、つまり VC のホームコンピュータを把握していれば十分である。

最後に RFID 以外によるセンサによる位置捕捉についても概説しておく、たとえば対象物の空間座標を測定するセンサの場合は、部屋などの空間の物理的な制約、または座標空間上に明示的に与えられた空間範囲に対して名前を与え、その空間範囲にその物理実体が含有されたときは、その空間に対応した VC に物理実体に対応するコンポーネントが含有されているとして扱う。なお、数値座標から含有関係への変換はセンサの測定精度を下げることになるが、センサを管理する LCDM は上記の手順 (2) と手順 (4) におけるメッセージを通じてセンサの測定情報、たとえば物理実体の座標情報を、その物理実体を含有する空間の VC および物理実体に対応したコンポーネントを保持するコンピュータに送り、そのコンピュータの実行システムが、上述の移動前後に発行するイベントを通じて、これらの VC およびコンポーネントにその測定情報を通

知することができる。このため、これらの VC およびコンポーネントは数値座標などの詳細な測定情報をもとに処理を行うこともできる。

4.4 基本性能評価

ここで基本性能を示しておく。LCDM が管理するセンサが物理実体の移動を発見後、その実体に対応した VC を移動元空間の VC から移動先空間の VC に移動させるコスト、つまり上述の手順 (1) から (4) の処理に要する時間は、移動先および移動元の VC が同一コンピュータ内にある場合は 210 msec であり、逆に異なるコンピュータ内にある場合は 420 msec である。なお、LCDM を維持するコンピュータおよび移動先または移動元に対応した VC を保持するコンピュータはそれぞれプロセッサとして Pentium M-1.4 GHz を持ち、OS は Windows XP Professional である。LCDM および実現システムの Java (J2SE ver.1.4) 上で実行される。また、コンピュータ間の通信接続は Fast Ethernet を通して実現され、手順 (2) のメッセージ送信は UDP マルチキャスト、他の通信は TCP を用い、上述のコストには TCP 接続に要した時間も含まれることになる。なお、ここで基本イベントに対するコールバックメソッドだけを定義したコンポーネント (移動時のサイズは約 3 KB) がコンピュータ間移動に要する時間は 45 msec であることから、残りの時間が LCDM と実行システム間の通信処理に要した時間となる。なお、コンピュータが移動した際は、そのコンピュータを参照する PC も移動することになるが、その移動に要する時間は上述の VC の移動と同様になる。また、SC のコンピュータ間移動に要する時間は上述のコンポーネントのコンピュータ間移動に要する時間は SC と VC のサイズが同等であればほぼ同じになる。ところで、上述の移動検知後の対応する VC の移動に要する時間 (420 msec) であるが、人間の歩行速度程度であれば十分な応答時間である。

5. 応用事例

ここでは本論文で提案した枠組みによる実現例を示す。

5.1 位置依存サービス

ここでは GUIDE²⁾ や NEXUS⁶⁾ と同様に移動コンピュータによる位置依存サービスを取り上げる。これは PDA の移動に従って、PDA 上でその現在地点を明示した地図表示を行うものである。図 5 の左のよ

現在の実装は必ずしも実行性能に最適化してあるわけではない。多くの位置センサの周期測定間隔が数秒単位であり、空間モデルによるオーバヘッドは無視できることが多い。

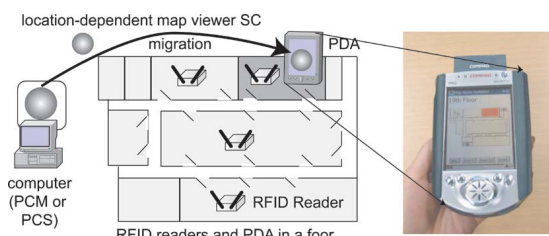


図 5 RFID システムを使った位置依存地図表示サービス
Fig.5 RFID-based location-aware map-viewer service.

うに各部屋にはアクティブ RFID リーダが設置され、それらは 1 つ以上の LCDM により管理されている。空間モデル上には各部屋に対する VC が与えられ、さらに各 VC には位置依存案内サービスとなる SC が連結されている。この SC は階の地図を表示するとともにそれぞれの部屋を地図上で明示する。そして PDA が移動してある部屋に到着すると、LCDM により管理された RFID リーダは PDA 裏面にはアクティブ RFID タグを識別し、PDA の到着を判定する。そして LCDM は PDA に対応した PC をネットワーク上から発見し、その部屋に対応した VC 上に PC を配置する。また、この際、その VC に連結された SC には PC が配置された旨のイベントが送信され、その SC は要求する計算リソースと PC に定義された PDA の計算リソースを比較し、要求を満足すれば SC は PC に移動する。そして PC は到着した PDA に SC を転送して、PDA 上で階の地図とともに現在地点を示す(図 5 の右の写真)。この応用例のプロトタイプ実装では PDA として HP iPAQ、アクティブ RFID システムとして RF Code Spider を用いている。本論文の枠組みを利用することにより、サービスを定義・実現する地図表示 SC は自動的に PDA に配置されるため、SC 自身は位置センサなどの監視が不要となり、位置センサの監視プログラムが不要となる。また、SC は空間モデルを通じて適切なコンピュータに再配置されるため、SC がサービス実現位置を管理する必要がない。このほか、PDA はそれが提供する計算リソースを、SC はその実行に必要な計算リソースを明示的に定義できることから、PDA の種別に応じて SC を選択することもできる。

5.2 デスクトップテレポーティング

次にユーザ依存サービスの実現例として、ユビキタスコンピューティングにおける典型的なサービスであるデスクトップテレポーティングを取り上げる。これはユーザの移動とともにデスクトップアプリケーションをユーザの近くにあるコンピュータに移動させるもので、ユーザは移動先でも自身のアプリケーションを

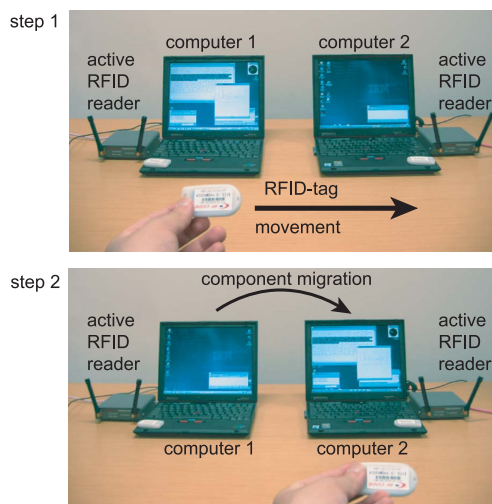


図 6 デスクトップテレポーティング
Fig. 6 Follow-Me Desktop Applications.

継続的に利用できるようになる。本論文の枠組みではアプリケーションを SC として実現し、それをユーザに対応した VC に連結して移動させることにより、容易に実現できる。図 6 は本応用例のプロトタイプ実装である。ここでコンピュータ 1 と 2 はそれぞれ PCS の参照先となるコンピュータ(プロセッサは Pentium-III, OS は Windows 2000)であり、LCDM は図 6 には表れていないコンピュータ(プロセッサは Pentium-M, OS は Windows XP)で実現されている。ユーザが移動してある空間から別の空間に移動すると、LCDM はそれが管理する RFID リーダ(RF-Code Spider)により移動元空間の VC から移動先空間の VC にユーザの VC を移動させる。それとともに SC も移動先空間の VC に移動し、その VC 内に配置された PCS を通じて SC は空間内のコンピュータに転送されて、そのコンピュータ上で実行される。この際に、SC は移動前の状態から処理を継続することになる。なお、図 6 における LCDM のユーザ移動発見からデスクトップアプリケーションの移動に要する時間は 0.5 秒である。

5.3 サービスの開発試験

コンテキスト依存サービスはその開発・試験が難しくなることが知られている。著者は移動コンピュータのエミュレータをモバイルエージェントとして実装する方法を提案している¹³⁾。一方、この空間モデルではコンピュータの移動はコンピュータに対応した PC が空間に対応した VC 間を移動するとしてモデル化され、その PC の移動を通じてサービスを切り換える。このため本章で取り上げた移動コンピュータによる位置依

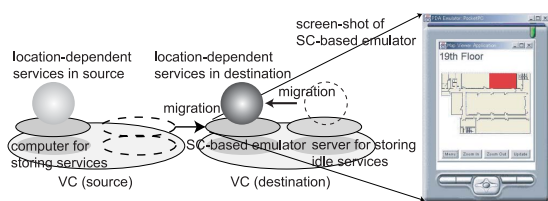


図 7 位置依存地図表示サービスの試験

Fig. 7 Testing for location-aware map-viewer service.

存サービスの場合、PDA の動作をエミュレーションする SC を用意し、PC の代わりにその SC を空間に対応する VC 間を移動させれば、あたかも PDA が移動しているかのようにサービスを実現する SC の動作試験を行うことができることになる。また、この空間モデルが物理実体やコンピュータ、サービスをコンポーネントして統一的に扱えることから、PC や VC の代わりにコンポーネントとして実装されたエミュレータを利用する限り、試験対象の SC の修正や再コンパイルなどは不要となる。また、図 7 に先述の移動コンピュータによる位置依存サービスの試験に行った際の PDA のエミュレータの実行画面を示す。このエミュレータは既発表論文¹³⁾の PDA 再現機構を SC として実装したものであり、PDA およびその PC として動作・移動することができる。

6. 関連研究

本論文の枠組みは、著者が提案した位置依存モバイルエージェントシステム¹¹⁾の問題点を解決することが研究動機となっている。これはユーザやモノに支援・注釈するエージェントを用意し、そのユーザやモノの移動に合わせてユーザの近傍に位置するコンピュータに移動させる枠組みであった。しかし、このシステムは多くの既存研究事例、たとえば GUIDE²⁾や Context Toolkit¹⁰⁾などと同様にセンサの識別可能領域をそのまま利用しており、サービスが依存すべき空間構成と測定可能な領域が一致しない可能性がある。このほかにもユビキタスコンピューティングにおける位置・ユーザ依存サービスのシステムおよびアプリケーション事例が数多く発表されているが、アドホックに実装されることが多く、体系的な空間モデルを提供する事例は少ない。

その中でも ParcTab システムでは本論文と同様に建家や部屋、人などの空間的な含有関係を木構造に変換した空間モデル¹⁷⁾を利用している。NEXUS⁶⁾や Cooltown⁷⁾は GPS による数値座標情報を基本にして、建家や道などの空間に当たられた属性や空間どうしの関係を組み合わせることにより柔軟な位置依

存の道案内サービスを提供している。また、Sentient Computing⁴⁾や EasyLiving¹⁾では部屋などの特定空間の名称とその空間内の相対的な座標情報による空間モデルを提供している。しかし、これらの既存システムでは空間情報は関係データベースなどで集中管理されることを前提にしており、人やモノなどの物理実体やコンピュータの位置は扱えるが、サービスを実現するソフトウェアの配置や、自動車などの移動空間は扱えない。一方、本論文の枠組みはサービスを実現するソフトウェアをコンテキストに応じて動的に再配置することができる。これは計算資源の少ないユビキタスコンピューティングでは有用となる。なお、ETH の Virtual Counterpart⁹⁾は Jini を利用することにより物理的実体に対応した Java オブジェクトを仮想オブジェクト導入し、さらに仮想オブジェクトにより直接サービスを実現できる点で本論文の枠組みとの類似性が高いが、Virtual Counterpart は RFID で捕捉された物理実体だけをモデル化対象にするのに対し、本論文では物理実体だけでなく建物や部屋などの物理空間も空間モデル上で統一的に扱える点が重要な差異となる。

7. おわりに

本論文ではユビキタスコンピューティング環境における位置およびユーザ依存サービスを前提に、新しい空間モデルを提案した。これは建家や部屋などの物理空間、人やモノなどの物理実体だけでなく、サービスを定義するソフトウェアやそれを実行するコンピュータの位置や特性を空間モデル上で統一的に扱えるという従来にはない特徴を持っている。また、ユーザや空間の移動に応じてサービスをコンピュータ上に配置できることと、空間モデルを維持・管理するためのデータベースサーバがいらないことから、計算資源に制限の多いユビキタスコンピューティングにおいて有用な枠組みとなっている。

最後に今後の課題を述べる。現在の実装では Java 言語および同言語に基づくモバイルエージェントシステムを利用して実装しているが、今後は XML 形式など汎用性の高い表現形式を利用した実装も検討している。また、検索機構の高速化や API などの整備なども重要となる。

謝辞 担当編集委員および査読者から数多くの貴重なコメントをいただいた。つつしんで感謝の意を表する。

参 考 文 献

- 1) Brumitt, B.L., Meyers, B., Krumm, J., Kern, A. and Shafer, S.: EasyLiving: Technologies for Intelligent Environments, *Proc. Intl. Symp. HUC'2000*, pp.12–27 (2000).
- 2) Cheverst, K., Davis, N., Mitchell, K. and Friday, A.: Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project, *Proc. Mobicom'2000*, pp.20–31 (2000).
- 3) Garlan, D., Siewiorek, D., Smailagic, A. and Steenkiste, P.: Project Aura: Towards Distraction-Free Pervasive Computing, *IEEE Pervasive Computing*, Vol.1, No.2, pp.22–31 (2002).
- 4) Harter, A., Hopper, A., Steggeles, P., Ward, A. and Webster, P.: The Anatomy of a Context-Aware Application, *Proc. Mobicom'99*, pp.59–68 (1999).
- 5) Hightower, J. and Borriello, G.: Location Systems for Ubiquitous Computing, *IEEE Computer*, Vol.34, No.8, pp.57–66 (2001).
- 6) Hohl, F., Kubach, U., Leonhardt, A., Rothermel, K. and Schwehm, M.: Next Century Challenges: Nexus — An Open Global Infrastructure for Spatial-Aware Applications, *Proc. Mobicom'99*, pp.249–255 (1999).
- 7) Kindberg, T., et al.: People, Places, Things: Web Presence for the Real World, Technical Report HPL-2000-16, Internet and Mobile Systems Laboratory, HP Laboratories (2000).
- 8) Leonhardt, U. and Magee, J.: Towards a General Location Service for Mobile Environments, *Proc. Workshop on Services in Distributed and Networked Environments*, pp.43–50 (1996).
- 9) Römer, K., Schoch, T., Mattern, F. and Dübendorfer, T.: Smart Identification Frameworks for Ubiquitous Computing Applications, *Proc. IEEE Intl. Conf. Pervasive Computing and Communications (PerCom'03)*, pp.253–262 (2003).
- 10) Salber, D., Dey, A.K. and Abowd, G.D.: The Context Toolkit: Aiding the Development of Context-Enabled Applications *Proc. Conf. CHI'99*, pp.434–441 (1999).
- 11) Satoh, I.: Bridging Physical and Virtual Worlds with Mobile Agents, *情報処理学会論文誌*, Vol.44, No.8, pp.2218–2229 (2003).
- 12) Satoh, I.: MobileSpaces: An Extensible Mobile Agent System, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E86-A, No.11, pp.2782–2790 (2003).
- 13) Satoh, I.: A Testing Framework for Mobile Computing Software, *IEEE Trans. Softw. Eng.*, Vol.29, No.12, pp.1112–1121 (2003).
- 14) Satoh, I.: Configurable Network Processing for Mobile Agents on the Internet, *Cluster Computing*, Vol.7, No.1, pp.73–83, Kluwer (2004).
- 15) 佐藤一郎: RFID タグ: 技術動向と影響, *情報処理*, Vol.45, No.1, pp.56–62 (2004).
- 16) Satoh, I.: Dynamic Federation of Partitioned Applications in Ubiquitous Computing Environments, *Proc. IEEE Intl. Conf. Pervasive Computing and Communications (PerCom'2004)*, pp.356–360 (2004).
- 17) Schilit, B.N. and Theimer, M.M.: Disseminating Active Map Information to Mobile Hosts, *IEEE Network*, Vol.8, No.5, pp.22–32 (1994).
- 18) Weiser, M.: The Computer for the 21st Century, *Scientific American*, No.9, pp.94–104 (1991).
- 19) World Wide Web Consortium (W3C): Composite Capability/Preference Profiles (CC/PP) (1999).
<http://www.w3.org/TR/NOTE-CCPP>
(平成 16 年 4 月 5 日受付)
(平成 16 年 10 月 4 日採録)



佐藤 一郎 (正会員)

1991 年慶應義塾大学理工学部電気工学科卒業。1996 年同大学院理工学研究科計算機科学専攻後期博士課程修了, 博士 (工学)。1996 年お茶の水女子大学理学部情報科学科助手, 1998 年同学科助教授。2001 年より国立情報学研究所助教授。このほか, 1994 年～1995 年 Rank Xerox Grenoble 研究所客員研究員, 2002 年より総合研究大学院大学数物科学研究科助教授 (併任)。1999 年～2001 年科学技術振興事業団さきがけ研究 21 (「情報と知」領域) 研究員。1996 年度情報処理学会論文賞, 1999 年度同学会山下奨励賞受賞。分散システムおよびユビキタスコンピューティングの理論, プログラミング言語, ミドルウェア等の研究に従事。電子情報通信学会, IEEE, ACM 各会員。