

## 屋内ロケーションウェアサービスのための問い合わせAPIの考察

鈕 龍<sup>†</sup> 松本 真佑<sup>†</sup> 佐伯 幸郎<sup>†</sup> 中村 匡秀<sup>†</sup>

<sup>†</sup> 神戸大学 〒657-8531 兵庫県神戸市灘区六甲台町1-1

E-mail: <sup>†</sup>longniu@ws.cs.kobe-u.ac.jp, <sup>††</sup>{shinsuke,masa-n}@cs.kobe-u.ac.jp, <sup>†††</sup>sachio@carp.kobe-u.ac.jp

**あらまし** 近年、センサ技術や無線技術の飛躍的な進歩に伴い、屋内測位システム (Indoor Positioning System, IPS) の研究が盛んである。また、屋内位置情報を用いた屋内ロケーションウェアサービス (Indoor Location-Aware Service, InLAS) に注目が集まり、いくつかの実用化事例も登場している。しかしながら、現状では、IPS と InLAS が密結合しているため、InLAS が複雑化し開発効率が低下してしまう。先行研究で、IPS で得られる屋内位置情報をその目的や用途、推定手段に依存せずに、中立的に表現するためのデータモデル Data Model for Indoor Location (DM4InL) を提案している。本研究では、DM4InL を基づいて、InLAS のための屋内ロケーション問い合わせサービス (Indoor Location Query Services, ILQS) の実現を目指し、ILQS の API を検討する。これらの API を利用することで、InLAS の開発者は IPS の詳細な実装を知らなくても屋内情報を取得することができ、効率的な InLAS 開発を実現できる。

**キーワード** 屋内測位システム, 位置情報, 屋内ロケーション問い合わせサービス, ロケーション・ウェアサービス, API

## Designing Query API for Indoor Location-aware Services

NIU Long<sup>†</sup>, Shinsuke MATSUMOTO<sup>†</sup>, Sachio SAIKI<sup>†</sup>, and Masahide NAKAMURA<sup>†</sup>

<sup>†</sup> Kobe University, 1-1 Rokkodai, Nada, Kobe, Hyogo, 657-8531 Japan

E-mail: <sup>†</sup>longniu@ws.cs.kobe-u.ac.jp, <sup>††</sup>{shinsuke,masa-n}@cs.kobe-u.ac.jp, <sup>†††</sup>sachio@carp.kobe-u.ac.jp

**Abstract** Recently, with the rapid progress of sensor and wireless technology, research and development have been conducted on the Indoor Positioning System (IPS). Furthermore, indoor location information is promising to achieve sophisticated Indoor Location-Aware Services (InLAS), and some practical services come onto market. However, the conventional system does not supposed to reuse the data and program of the indoor locations of another system. This makes the system complex and difficult to manage. To cope with the problem, in our previous work, we have been presented Data Model for Indoor Location (DM4InL), which prescribes a common data model independent of implementation of IPS or the usage of InLAS. Addition, based on the DM4InL, we proposed Indoor Location Query Service (ILQS). In this paper, we designing ILQS's query API for InLAS. By utilizing these API, developers of InLAS can obtain indoor information without knowing the implementation details of the IPS.

**Key words** indoor positioning system, location information, Indoor location Query Services, location-aware service, API

### 1. はじめに

人や物の位置情報は、高度な付加価値情報サービスを実現するために重要な情報である。例えば、自身の居場所と建物情報を組み合わせ、最も近くにある病院までの場所を地図上に提示するサービスなどが考えられる。位置情報を認識する技術として、屋外では GPS や Wi-Fi を組み合わせた測位が既に実用化され、携帯電話を対象とした歩行者ナビゲーションサービスな

ど、屋外位置情報技術を利用した位置情報サービスが広く普及している。一方で、これらの測位技術は建物の出入口などを識別できるほどの精度はない。そのため、屋内や地下街と言った GPS 信号の届かない環境では利用が難しいという問題がある。近年、GPS 電波が届かない屋内において測位を可能にする重要な技術である屋内測位システム (Indoor Positioning System, IPS) の研究開発が盛んである。IPS を実現する屋内位置認識技術として、Wi-Fi [1], 赤外線 [2], 超音波 [3], IMES [4], 自律

航法技術 [5] など様々な方式がこれまでに開発され、いくつかの商用 IPS も登場している [6] [7]。各方式の構成や特徴は、主に利用される通信方式に依存する。そのため、これらの屋内測位技術は、技術ごとに定位の精度や分解能、インフラ導入のコストといった観点について評価され、目的としているサービスごとに使い分けられている。

IPS を利用し、屋内において人や物の位置情報を利用することで、人と物の位置関係から適切なアクションを実行するサービスが実現できる。例えば、ショッピングモールのナビゲーション、美術館などでの展示物のガイダンス、位置に応じた家電機器操作、ターゲット広告、ライフログなど様々なものが考えられる。このようなサービスは**屋内ロケーション・ウェアサービス (Indoor Location-Aware Service, InLAS)** と呼ばれ、研究開発が盛んである。

InLAS に関する既存の研究では、実装する InLAS の目的や利用する IPS ごとに最適な方法で、屋内位置情報を独自に表現・管理している。この方法は InLAS の速い応答速度を得られるメリットがあるが、IPS と InLAS が密結合となり、異なる InLAS 同士でデータの共有・再利用できなくなるデメリットがある。また、システム毎に IPS の利用ロジックを実装する必要があり、InLAS が複雑化し開発効率が低下してしまうという問題もある。

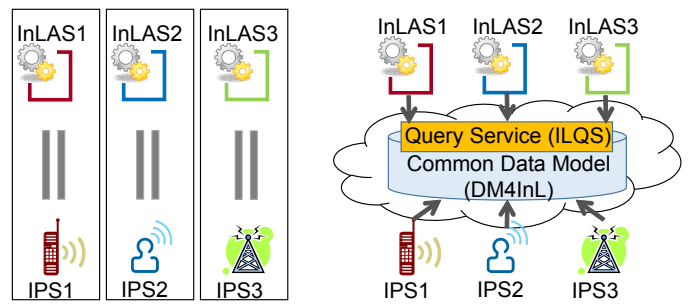
InLAS の効率的な開発を支援するために、我々は**屋内ロケーション問い合わせサービス (Indoor Location Query Services, ILQS)** の実現を目指して研究を行なっている [8]。ILQS は屋内の様々なオブジェクト (家電や設備機器、部屋など) の位置情報を何らかの IPS で収集・保存し、外部のアプリケーションから標準的な方法 (API) で取得できるようにするサービスである。ILQS によって InLAS と IPS が疎結合となるため、屋内情報の共有・再利用が容易となり、典型的な処理が共通化でき、開発効率の向上が期待できる。

先行研究では、IPS で得られる屋内位置情報をその目的や用途、推定手段に依存せず、中立的に表現するためのデータモデル **Data Model for Indoor Location (DM4InL)** が検討されている。DM4InL は、**位置モデル、建物モデル、オブジェクトモデル** の 3 つのモデルから対象とする全ての要素を表現する。本稿では、DM4InL に基づき、InLAS のための ILQS の実現を目的とし、ILQS の API を検討する。本研究では、サービスがクエリの対象によって、建物情報を問い合わせる API とオブジェクト情報を問い合わせる API を分離し設計を行う。また、同一の対象に対し、対象の属性を条件として、対象情報を問い合わせる API を検討する。更に、地物間の空間関係を問い合わせ API を提案する。これらの API を利用することで、InLAS の開発者は IPS の詳細な実装を知らなくても屋内情報を取得することができ、効率的な InLAS 開発が達成できる。

## 2. 準備

### 2.1 屋内測位システム (IPS)

IPS は、建物内部の人やオブジェクトの位置を推定するシステムの総称である [9]。GPS の電波が届かない建物内の測位技



(a) 従来型の InLAS

(b) 本研究で目指す InLAS

図 1 InLAS の実現方式の比較

術として期待され、様々な技術が研究・開発されている。IPS の実現技術として、現在までに以下のものが提案・実用化されている。

- Wi-Fi [6]
- 2次元コード [10]
- 可視光通信 [11]
- IMES (Indoor Messaging System) [4]
- RFID タグ [12]
- 複合的位置認識 [13]
- 自律航法 [5]

これらの技術は、粒度、能動性、インフラ導入のコストといった観点についてそれぞれ異なる長所・短所といった特徴があり、開発者が用途に応じ適切なものを選択する必要がある。そのため、屋外における GPS や Wi-Fi 測位 [14] のように、デファクトスタンダードとなる技術は未だ確立されていない。

### 2.2 屋内ロケーションウェアサービス (InLAS)

IPS で取得した屋内位置を利用することで、位置情報に基づき様々な付加価値を提供するサービス、**屋内ロケーションウェアサービス (InLAS)** を実現できる。InLAS では、取得した屋内位置に応じて適切なアクションや振る舞いを提供する。InLAS の実用化事例として、以下のようなものが挙げられる。

- **屋内駐車場ナビゲーションサービス** : GPS 信号を受信しにくい屋内駐車場において、車の屋内位置と駐車場の空間情報を管理し、ユーザに空間情報を提供するシステムである。例えば、この駐車場に駐車できる場所はあるかどうか、その場所がどこであるか、サービス利用者の車がどこにあるか、出口へのナビゲーションなどがある。

- **自動出席登録サービス** : ユーザが部屋に入ると、スマートフォンなどの機器情報を元に個人を識別し、自動的に出席を登録するシステムである。このシステムでは、出席している人数や時間的な変化、出席者の位置など情報を提供可能である。

一般的にこうした InLAS は、サービスの目的や利用する IPS に応じた実装を行なっている。サービス内部の屋内位置情報は、標準化されておらず、InLAS 毎に独自の方式で管理・表現されている。

### 2.3 屋内ロケーション問い合わせサービス (ILQS)

図 1(a) に従来型の InLAS の実現方式を示す。2.1 と 2.2 で述べたとおり、IPS には確立した標準がなく、InLAS を実現するシステムについても、それぞれが適切な IPS を利用し、試行

錯誤的に開発されている。すなわち各システムにおいて InLAS と IPS が密結合しており、屋内位置情報のデータやそれに対する処理がシステムごとに独自に行なわれている。そのため、あるシステムのデータや処理を別のシステムで再利用することは基本的に想定されていない。この結果、InLAS が複雑となり開発効率が低下するという問題が発生する。

我々の最終的な目標は、図 1(b) に示す新たな InLAS の実現方式を確立することである。提案方式では、様々な IPS から蓄積した屋内位置情報を提供する屋内ロケーション問い合わせサービス (Indoor Location Query Service, ILQS) が重要な構成要素となっている。ILQS は、屋内の様々なオブジェクトの位置情報を標準的な形式で収集・保存し、外部の InLAS に標準的な手段で提供するサービスである。これによって InLAS と IPS が疎結合となり、データや処理の共通化・再利用が促進され、InLAS の開発効率が飛躍的に高まると期待できる。また、ILQS をクラウドに配置することで、複数の建物にまたがる屋内情報を大域的に収集・利用することが可能となる。

#### 2.4 屋内位置表現データモデル (DM4InL)

先行研究では、IPS で得られる屋内位置情報をその目的や用途、推定手段に依存せずに、中立的に表現するためのデータモデル **DataModel for Indoor Location (DM4InL)** を提案している [8]。DM4InL は、位置モデル、建物モデル、オブジェクトモデルの 3 つのモデルから構成され、建物内の様々なオブジェクトの位置情報を標準的な形式で管理するモデルである。

図 2 に DM4InL の ER 図を示す。本稿で記述する ER 図は [15] で示された記法に基づいている。四角はエンティティを表し、スキーマを構成するデータ項目を右に並べる。データ項目のうち、下線のある項目は主キー、下線のない項目は属性、括弧つき下線は二次キーを表す。各エンティティの下にはインスタンスを併記している。エンティティ間の関連については、

- (+——€) は親子関係
- (+——...) は参照関係
- (+——o+) は派生関係

を表す。以降、各モデルを説明する。

##### 位置モデル

位置モデルは建物の空間要素と建物内のオブジェクトの空間属性を定義するデータモデルである。建物内の任意の点を、その建物の基準点からの相対座標 (3 次元オフセット) で定位する。屋内における位置 (屋内位置) を議論するうえでは、その位置が属する建物が必ず 1 つ存在しなければならない。そこで位置モデルでは任意の屋内位置を、その位置が属する建物と建物の基準点からの 3 次元座標で定位する。さらに、位置モデルは上記の 3 次元座標を用いたポイント (Local Point)、ライン (Local Line)、ポリゴン (Local Polygon)、スペース (Local Space) を定義する。

##### 建物モデル

建物モデルは建物および建物内の地点、経路、区画を定義するデータモデルである。以下にそれぞれについて説明する。

- 建物 (Building, B)：建物そのものの存在を定義する。

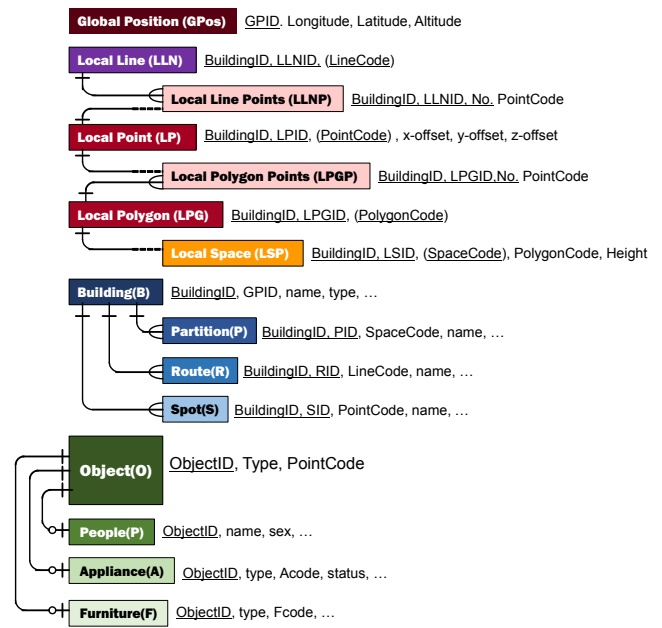


図 2 DM4InL の ER 図

各建物は、建物 ID を主キーとして識別される。属性として、その建物が存在する大域的な位置、建物名、タイプなどがある。

- 地点 (Spot, S)：建物内の任意の地点を表現する。非常口や風呂場ドアのように、ポイントとして扱いたい建物内の空間要素を表現する。
- 経路 (Route, R)：R は建物内の任意の経路を表現する。避難経路や導線のように、ラインとして扱いたい建物内の空間要素を表現する。
- 区画 (Partition, P)：P は建物内の任意の区画を表現する。リビングや寝室、キッチンのように、スペースとして扱いたい建物内の空間要素を表現する。

各エンティティの空間属性はそれぞれ位置モデルのポイント、ライン、スペースで定義される。

##### オブジェクトモデル

オブジェクトモデルは建物内の様々なオブジェクトを表現するモデルである。各オブジェクトは基本的に移動可能であり、ある建物の内部にいる場合にはその建物の屋内位置で定位される。現実世界では、人や家電機器、家具といった様々な種類のオブジェクトが存在する。そのため、オブジェクトモデルでは、定位可能な抽象的なオブジェクトエンティティを定義し、具体的なエンティティはオブジェクトを継承するサブタイプとして追加できるようなモデリングを行なっている。

### 3. ILQS の API の提案

ILQS は InLAS と IPS を疎結合化するため、DM4InL を利用し両システムに標準的なインターフェース (API) を提供するサービスである。システムを利用、あるいはシステムに働きかけるユーザもしくはシステムを、**アクター**と言う。本研究では、ILQS を利用するアクターを以下の 4 つと規定する。

- **ILQS の管理者**：InLAS と IPS を保守する。
- **InLAS の管理者**：建物を保守する。

- **InLAS:** 建物の空間要素とオブジェクトを保守する。また、建物の空間要素とオブジェクトの情報を問い合わせる。

- **IPS:** 座標の設定、屋内のオブジェクトを登録や測位や抹消を行う。

本稿では、InLAS に対し、DM4Inl を利用した建物の空間要素とオブジェクトの情報を問い合わせる API を提案する。

### 3.1 API に対する要求

これまで、様々な研究でロケーションに基づくクエリが検討されている。研究により表現の差はあるが、基本的にはロケーションに基づくクエリは以下の 4 種類である。

- 位置クエリ (Position Queries)
- ナビゲーションクエリ (Navigation Queries)
- レンジクエリ (Range Queries)
- *k*-最近傍クエリ (*k*-nearest neighbor Queries)

位置クエリは移動もしくは非移動であるオブジェクトに対する屋内位置情報を問い合わせるクエリである。このクエリは、他のクエリからの依存度が高い本質的なクエリである。ナビゲーションクエリは、ユーザが探す位置までの経路を検索するクエリである。レンジクエリは、ある範囲内に存在するオブジェクトを検索するクエリである。*k*-最近傍クエリは *k* に最も近いオブジェクトを検索する機能である。ILQS において提供する API も、これらの 4 つのクエリの要求を満たすことが必要である。

### 3.2 ILQS の API

本稿では、提案する API をクエリの対象により、建物の情報を問い合わせる「建物問い合わせ API」、オブジェクトの情報を問い合わせる「オブジェクト問い合わせ API」の 2 種類に分ける。建物などの、**地物**であるオブジェクトが保有する情報は、地物固有の性質であり、属性と呼ばれる。例えば、地物の形状や位置、名称、利用目的、地物の外観が属性となる。「形状や位置」を表す属性は**空間属性** (Spatial Attribute)、「名称や利用目的等」を表す属性は**主題属性** (Subject Attribute) と言われる。そのため、同じ対象を探す方法として、主題属性あるいは空間属性で探す 2 通りが存在する。また、これら 2 種以外に、オブジェクトと建物の空間要素間の関係、建物の空間要素間の関係、オブジェクト間の関係で問い合わせる「位置モデル API」が必要となる。さらに、位置モデルで定義される 3 つのモデルを複合的に用い問い合わせを行う「複合 API」も必要となる。以下にそれぞれの API について説明する。

#### 3.2.1 建物問い合わせ API

建物問い合わせ API は、建物と建物内の空間要素の情報を提供する API である。図 3 に建物を問い合わせ API の対応表を示す。まず、検索対象の種類によって、建物クエリ、区画クエリ、経路クエリ、地点クエリに分類される。また、クエリ対象を探す条件によって、主題属性で探す API、空間属性で探す API に分類される。主題属性による問い合わせは一般的にいくつかの項目を組み合わし利用する。空間属性による問い合わせに用いる項目は地物を扱う位置モデルのエンティティで定義される。例えば、主題属性あるいは空間属性でクエリ対象を探す API、*getBuildingByBuildingID(BuildingID)* では

| Query Target | Attribute         | Item   |                         |
|--------------|-------------------|--|-------------------------|
| BUILDING     | Subject Attribute | BuildingID, Name, Type                         |                         |
|              | Spatial Attribute | GlobalPosition, Partition, Route, Spot, Object |                         |
| PARTITION    | Subject Attribute | PartitionID, Name                              |                         |
|              | Spatial Attribute | Space, Building                                |                         |
| ROUTE        | Subject Attribute | RouteID, Name                                  |                         |
|              | Spatial Attribute | Line, Building                                 |                         |
| SPOT         | Subject Attribute | SpotID, Name                                   |                         |
|              | Spatial Attribute | Point, Building                                |                         |
| Object       | Subject Attribute | NULL   | ID, Type                |
|              |                   | People   | ID, name, sex           |
|              |                   | Appliance                                      | ID, type, Acode, status |
|              |                   | Furniture                                      | ID, type, Fcode         |
|              | Spatial Attribute | Point, Building                                |                         |

図 3 建物とオブジェクトを問い合わせ API の対応表

建物 ID を用いることによって、対象となる建物を探すことが可能である。空間属性による問い合わせ API は建物と建物内の空間要素の空間属性で、地物を探す API である。例えば、*getPartitionBySpace(Space)* は区画を表現する位置モデルであるスペースによって、区画を探すことができる。建物と建物内の空間要素は親子関係であるため、例えば、*getPartitionsInBuilding()*, *getBuildingContainSpot()* など建物によって建物内の空間要素を検索できる。また、空間属性と主題属性を連携して、空間要素要素を探すことができる。例えば、建物内ある地点を探す、*getSpotByNameInBuilding(SpotName, Building)* などが可能である。

#### 3.2.2 オブジェクト問い合わせ API

オブジェクト問い合わせ API は、建物内のオブジェクト情報を返す API である。DM4InL でオブジェクトに対し、主題属性と空間属性を定義する。図 3 にオブジェクト問い合わせ API の対応表を示す。

オブジェクトの空間属性はオブジェクトの位置を表現するポイントで定義する。そして、位置モデルで定義される BuildingID によりポイントが表現され、オブジェクトはどの建物内に存在するかが検索される。空間属性でオブジェクトを探す API として、ポイントによりオブジェクトを検索する、*getObjectByPoint()*、建物内のオブジェクトを検索する、*getObjectsInBuilding()* などが挙げられる。

オブジェクトモデルでは、具体的なエンティティはオブジェクトを継承する、また、オブジェクトのタイプによって、具体的な主題属性を定義する。主題属性でオブジェクトを探す API としては、オブジェクトの種類により検索する、*Object[] getObjectsByType(Type)*、名前により人を検索する *Object[] getPeopleByName()* などが挙げられる。

更に、空間属性と主題属性を複合し、オブジェクトの問い合わせを行う API も提案する。例えば、建物内のある種類のオブジェクトを探す API、*getPeopleBySexInBuilding()* などが挙げられる。

| POINTENTITY1 | ENTITY2      | TOPOLOGY |
|--------------|--------------|----------|
| Spot, Object | Spot, Object | Distance |
|              | Route        | Through  |
|              | Partition    | Contain  |

図 4 複合 API の対応表

### 3.2.3 位置モデル API

位置モデル API は、建物やオブジェクトなど様々な要素の空間関係を返す API である。本稿では、ポイントとポイント、ポイントとライン、ポイントとスペースの関係を検討し、以下基本的な空間関係を考える。

- ポイントとポイント：距離
- ポイントとライン：介す
- ポイントとスペース：含む

ポイントとポイントの空間関係について、2つのポイントの距離を計算する API を、ポイントとラインの空間関係について、ラインがポイントを紹介するかどうかを判断する API を、ポイントとスペースの空間関係について、ポイントがスペースの内外を判断する API を提案する。

例として、ポイントとポイントの距離を計算する API、*getDistanceBetweenPoints(Point1, Point2)*、ポイントと他のエンティティの関係を判断する API、*isRouteThroughPoint(Route, Point)*、ポイントがスペースの内外を判断する API、*isPointInSpace(Point, Space)* などが挙げられる。

### 3.2.4 複合 API

複合 API は位置モデルで定義された3つのモデルを連携して複合情報を提供する API である。例えば、キッチンに人が居るかどうかを問い合わせる API などである。図4に複合APIの対応表を示す。本稿では、複合APIとして3.2.3で述べた、関係を判断するAPIと建物やオブジェクトの問い合わせAPIを連携し、ポイントとして扱いたいエンティティの空間のトポロジー情報を提供するAPIを提案する。ポイントとして扱いたいエンティティは地点とオブジェクトである。例えば、*Boolean isObjectInPartition(Object, Partition)* は区画にあるオブジェクトがあるかどうかを判断する、*Double getDistanceBetweenSpots(Spot1, Spot2)* は二つの地点の距離を計算する、*Boolean isRouteThroughSpot(Route, Spot)* はある経路が地点を紹介するかどうかを判断するAPIである。

以上、ILQSの基本的なAPIを提案した。これらのAPIを活用して、複合情報を問い合わせるAPIが実現できる。例えば、ホテルでテレビを点けている部屋が何室あるか検索する場合、まず、ホテルにあるオブジェクトを探す。*getObjectsInBuilding()*。そして、探したオブジェクトの中の、点いているテレビを検索する。*getApplianceByTypeAndStatus()*。最後に、戻り値として、探したテレビを収容する区画をAPIが返す。

## 4. ケーススタディ

本章では、3. で提案した基本的なAPIの実用性を、以下の

```

Partition[] getParkableSpaces(Parking){
    Partition[] parkableSpace[], allParkSpace[];
    allParkSpace[] = getPartitionsByNameInBuilding(Parking, parkSpace);
    for(i=0; i<allParkSpace[].length; i++){
        if(allParkSpace[i].parkable)
            parkableSpace[].add(allParkSpace[i]);
    }
    return parkableSpace[];
}

ParkSpace getMyCarPosition(Car mycar){
    ParkSpace myCarParkSpace;
    Partition partition = getPartitionContainObject(mycar.object);
    myCarParkSpace = getParkSpaceByPartition(partition);
    Return myCarParkSpace;
}

countParkingCars(Building Parking){
    Object[] car[] = getObjectByTypeInBuilding(Parking);
    Return car[].length;
}

```

図 5 屋内駐車場ナビゲーションサービスのプログラム

2つのInLASを対象としケーススタディを行い確認する。

### 4.1 屋内駐車場ナビゲーションサービス：

屋内駐車場ナビゲーションサービスは、GPS信号を受信しにくい屋内駐車場において、車の屋内位置と駐車場の空間情報を管理して、ユーザに駐車場と車の空間情報を提供するサービスである。例えば、この駐車場に駐車できる場所があるかどうか、その場所はどこであるか、ユーザの車がどこにいるか、出口までのナビゲーション、などのサービスを提供する。

サービスには、駐車場所や通りを区画と入口で定義し、駐車場所の利用状況と言う具体的な属性も定義する。以下ILQSを利用してC1, C2, C3の3つのケースを検討する。

C1: 駐車できる場所はどこにあるか

C2: ユーザの車がどこにいるか

C3: 駐車場に駐車している車がいくつあるか

以上のケースで必要となるAPIのJavaによる実現方法を図5に示す。C1に対し、まず、*getPartitionsByNameInBuilding()*を使い、建物内の全ての駐車場所を探す。その後、各駐車場所に対し状況をチェックし、全ての駐車できる場所を返す。C2に対し、複合API、*getPartitionContainObject()*を使い、ユーザの車が止まる場所を返す。C3に対し、空間属性と主題属性によるオブジェクト問い合わせAPI、*getObjectsByTypeInBuilding()*を使い、駐車場に止まっている全ての車を返す。そして、返ってきた数をカウントする。

以上のAPIで屋内駐車場ナビゲーションサービスで必要となるAPIを実現できることが確認できた。

### 4.2 自動出席登録サービス：

自動出席登録サービスは、ユーザが教室に入ると、サービスがスマートフォンを認識し、スマートフォンに対応したユーザが自動的に出席登録されるサービスである。出席以外にも、現在出席している人数や、出席者の位置などの情報を提供できる。以下ILQSを利用してC4, C5の2つのケースを検討する。

```

Void autoLogin(){
    Appliance[] phone[] = getApplianceByTypeInPartition(Partition,Phone);
    For(i = 0; i<=phone.length; i++){
        getPhoneOwner(phone[i]).login();
    }
}
}
Void autoLogout(){
    User[] LoginUser[];
    ...omission...//select all logged users and add to LoginUser[];
    For(i = 0; i<=LoginUser.length; i++){
        Phone phone = LoginUser[i].phone();
        if(!isObjectInPartition(Partition, phone))
            LoginUser[i].absence();
    }
}
}

```

図 6 自動出席登録サービスのプログラム

C4: ユーザが部屋に入ると、自動的に出席登録する  
 C5: ユーザが部屋に出ると、自動的にログアウトする  
 C4 に対して、getApplianceByTypeInPartition() を使い、室内の携帯を探して、全て出席ユーザの携帯を収集する。次に、携帯のユーザの出席状況を修正する。C5 に対して、まず、全てログインしたユーザを収集する。次に、isObjectInPartition() を使い、退席したユーザの出席状況を修正する。

以上のケースが提案する API の JAVA による実現方法を図 6 に示す。

以上の API で自動出席登録サービスで必要となる API を実現できることが確認できた。

## 5. 考 察

ILQS の API が、現在 InLAS に必要とされる一般的なクエリを満たしているか、モデルの完備性と応用性を検討する。InLAS で必要とされる API は 3.1 で述べた通り、位置クエリ、ナビゲーションクエリ、レンジクエリと  $k$ -最近傍の 4 種類である。

位置クエリはデータモデルの定義によって実現する。建物内の空間要素やオブジェクトを主題属性で探すことができる。そして、返り値として地物の空間属性を返す。先行研究で提案した DM4InL がこのクエリを実現できるを確認している。

レンジクエリは、区画に何を収容するか、オブジェクトがどこに存在するか、などの問い合わせを行うクエリである。3.2.3 で述べた位置モデルの問い合わせ API で実現できる。

$k$ -最近傍クエリの実現は、ポイントとして扱いたいエンティティを設定し、 $k$  とポイントとして扱いたいエンティティ間の距離を計算することで、最も近くに存在するエンティティを検索することができる。

ナビゲーションクエリの実現は InLAS のデータ構造によって、実現できる。例えば、駐車場所や通りを区画と入口であらかじめ定義する。そして、入口を地点で定義する。入口によって通行可能な区画を連携する。このデータ構造によって、道路のネットワークを作成できる。その後、アルゴリズムで区画 A から区画 B までの経路を計算できる。ここで、道路ネットワー

クの経路探索については、本研究では検討を行わない。

以上の考察によって、ILQS が一般的な InLAS のクエリを実現でき、また、クエリ同士の複合により応用的な利用も可能であることを確認した。

## 6. ま と め

本稿では、先行研究で提案したデータモデル、DM4InL を利用した屋内ロケーション問い合わせサービス、ILQS の基本的な API を提案し、ILQS を利用した、2 つのユースケースを通じ API の有効性を検討した。また、InLAS に必要とされるの 4 つの基本クエリの充足性を確認することにより、ILQS の API の実用性に対する考察を行い、提案 API は InLAS 実現に対し、要件を満足していることが明らかになった。これらの API を利用して、屋内ロケーションウェアサービスの開発効率が飛躍的に高まると期待できる。今後の課題として、本研究の最終目的である、提案する API を用いた屋内位置問い合わせサービス (ILQS) を開発していく必要がある。

### 謝辞

この研究の一部は、科学技術研究費（基盤研究 C 24500079、基盤研究 B 26280115、基盤研究 C 24500258、若手研究 B 26730155）の研究助成を受けて行われている。

### 文 献

- [1] “Google location based services.” April 2010. [http://www.google.com/googleblogs/pdfs/google\\_submission\\_dpaw\\_wifi\\_collection.pdf](http://www.google.com/googleblogs/pdfs/google_submission_dpaw_wifi_collection.pdf).
- [2] “Smartlocator(r),” 13 2006. <http://www.nec.co.jp/press/ja/0603/1301.html>.
- [3] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location-support system,” *MobiCom*, pp.32–43, 2000.
- [4] “準天頂衛星システム ユーザインタフェース仕様書 (is-qzss),” March 2013. [http://qz-vision.jaxa.jp/USE/is-qzss/DOCS/IS-QZSS\\_15\\_J.pdf](http://qz-vision.jaxa.jp/USE/is-qzss/DOCS/IS-QZSS_15_J.pdf).
- [5] 吉澤菜津子, 遠藤貴裕, 永見健一, “屋内位置情報における推定技術の開発と新しいサービスの展開について,” *INTEC TECHNICAL JOURNAL*, vol.13, pp.44–51, 2013.
- [6] “Placeengine”. <http://www.placeengine.com/show/about>.
- [7] “Guardly’s indoor positioning system”. <https://www.guardly.com/technology/indoor-positioning-system>.
- [8] 鈕 龍, まつ本真佑, 佐伯幸郎, 中村匡秀, “屋内ロケーションウェアサービスに向けた位置表現データモデルの提案,” 電子情報通信学会技術研究報告, pp.101–106, 2014.
- [9] K. Curran, E. Furey, T. Lunney, J. Santos, D. Woods, and A.M. Caughey, “An evaluation of indoor location determination technologies,” *Journal of Location Based Services*, vol.5, pp.61–78, 2011.
- [10] 服部聖彦, 藤井哲也, 門 洋一, 張 兵, “2 次元マーカを用いた屋内ユーザー位置・方向推定システムの検証,” *IPJS SIG Technical Report*, vol.2008-UBI-17, pp.203–207, 2008.
- [11] “Bytelight’s indoor location system”. <http://www.bytelight.com/>.
- [12] 中田豊久, 伊藤日出男, 金井秀明, 國藤 進, “既知タグとの共変化を利用した active rfid タグの測位方法,” 第 5 回知識創造支援システムシンポジウム報告書, 第 108 巻, pp.381–386, IN2008-196, 2008.
- [13] 小笠原貴洋, 井垣 宏, 井上亮文, 星 徹, “屋内位置情報サービス開発支援環境の構築,” *IPJS Interaction*, pp.331–336, 2012.
- [14] “Skyhook,” 2014. <http://www.skyhookwireless.com/>.
- [15] 渡辺幸三, データモデリング入門, 日本実業出版社, 2003.