

# Considering Common Data Model for Indoor Location-aware Services

Long Niu, Shinsuke Matsumoto, Sachio Saiki, Masahide Nakamura  
Graduate School of System Informatics, Kobe University  
1-1, Rokkodai, Nada, Kobe 657-8501, Japan  
longniu@ws.cs.kobe-u.ac.jp, shinsuke@cs.kobe-u.ac.jp,  
sachio@carp.kobe-u.ac.jp, masa-n@cs.kobe-u.ac.jp

## ABSTRACT

Indoor positioning system (IPS) identifies positions of various indoor objects, and is a key technology to achieve sophisticated Indoor Location-Aware Services (InLAS). In most conventional systems, InLAS and IPS are tightly coupled. That is, one system does not supposed to reuse indoor location data and program of another system. This makes individual systems complex and difficult to manage. To cope with the problem, we propose *Data Model for Indoor Location (DM4InL)*, which prescribes a common data schema, independent of implementation of IPS or the usage of InLAS. The proposed DM4InL represents the location of every indoor object in a standard way, by using three kinds of models: location, building and object models. We also design the fundamental API, which implements typical queries to the indoor location data from external applications. The proposed method achieves loose-coupling of InLAS and IPS, which significantly improves the efficiency and reusability in the InLAS development.

## Categories and Subject Descriptors

H.2.1 [Logical Design]: Data models; H.2.8 [Database Applications]: Spatial databases and GIS; H.3.5 [Online Information Services]: Web-based services, Data sharing

## Keywords

indoor positioning system; location information; data modeling; location-aware service; API; indoor location query service.

## 1. INTRODUCTION

With the rapid development of wireless and sensor technologies, research and development of *Indoor Positioning System (IPS)*, for short) are actively conducted. IPS identifies precise positions of people and objects in *indoor space* where Global Positioning System (GPS) is unable to work. Various enabling technologies for IPS have been developed

so far, including those using Wi-Fi [9], infrared [10], ultrasound [14], IMES [8], pedestrian dead reckoning (PDR) [13], etc. Several commercial IPS have already come onto market (e.g. PlaceEngine[6], Guardly[16]). These enabling technologies have different characteristics of accuracy, resolution and cost of infrastructure deployment.

By using IPS, various *Indoor Location-Aware Services* (we call *InLAS*, for short) can be implemented. InLAS performs appropriate actions and behaviors autonomously, according to indoor positions of users or dynamic/static objects. Typical InLAS include a navigation service of a shopping mall, an exhibition guidance service for a museum, a location-aware appliance control in smart home, location-aware targeted advertising, life log, and so on.

When we implement an InLAS with an IPS, it is necessary to determine, within the InLAS, how to represent and manage indoor location information obtained by the IPS. Many existing systems individually represent and manage the indoor position information, considering the purpose of the InLAS and characteristics of the IPS used. Such a proprietary representation and management method has an advantage of optimal performance. However, it causes *tight coupling* of the InLAS and the IPS, where indoor-location data and processes cannot be reused among different services. Thus, the proprietary method makes the implementation of InLAS complicated, and increases development cost and effort.

To improve the efficiency and reusability in the development of InLAS, we are studying to achieve a cloud service, called *Indoor Position Query Service (IPQS)*, for short). IPQS gathers indoor location information of various objects (room, equipment, appliance, people, etc.) from arbitrary IPS. It then provides *application-neutral* API, by which external systems can query indoor location of a specified object or building. Thus, IPQS achieves loose-coupling of the IPS and the InLAS, which facilitates sharing and reuse of indoor information and common procedures.

As the first step towards the IPQS, we propose *Data Model for Indoor Location (DM4InL)* in this paper. DM4InL specifies a common data schema for representing indoor location information in an application-neutral way, which does not depend on any specific IPS or InLAS.

The proposed DM4InL consists of three models: *location model*, *building model*, and *object model*. The location model represents any location in a building by a relative position (3-dimensional offset) from the base coordinates of the building. The building model defines every building with attributes and global position. It also defines geographic elements in each building such as partitions, routes and spots.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

LocWeb'14, November 3, 2014, Shanghai, China.

Copyright 2014 ACM 978-1-4503-1459-6/14/11 ...\$15.00.

<http://dx.doi.org/10.1145/2663713.2664423>.

The object model defines various objects in a building, such as people, appliance, furniture, etc. The current position of each object is represented by a point defined in location model. By using DM4InL, developers of InLAS are able to manage indoor location information independently of specific IPS, which enables to share the data among different InLAS.

We also study some API for basic query to the IPQS. For example, the API `getObjectLocation()` returns the indoor position of a given object. Another API `getObjectsInPartition()` returns all object that exist in a certain partition. Using the API, it is possible for the developers to obtain indoor position of an object without knowing the implementation details of the IPS. Thus, the API significantly improves the development efficiency of InLAS.

## 2. PRELIMINARIES

### 2.1 Indoor Position System (IPS)

IPS is a generic name for systems that estimate position of subject or object inside building. An IPS is a solution based on magnetic, other sensor data or network of devices used to wirelessly locate objects or people inside a building [2]. Enabling technologies of IPS include the followings:

- Wi-Fi [6]
- 2D-Code [3]
- Visible light communication [4]
- IMES (Indoor Messaging System) [8]
- RFID tag [17]
- Hybrid methods of position recognition [12]
- PDR [13]

The above technologies have different characteristics in accuracy, resolution and cost of infrastructure and deployment, which are generally chosen based on requirements and cost for the target solution. More and more technologies for IPS are being developed to complement the existing GPS, which cannot reach inside buildings. Unlike the GPS, however, there is no de-facto standard for the IPS yet.

### 2.2 Indoor Location-Aware Service (InLAS)

In this paper, we refer to a service that performs appropriate actions or behaviors according to the indoor location information as *Indoor Location-Aware Service (InLAS)*. Practical InLAS have come onto market, including:

- **Shoplat** [11]: With a ultrasonic positioning technology, this service pushes coupons or loyalty program of a shop when a user approaches the shop. The user's location is estimated by broadcasting ultrasound that can be detected by microphone the user's smartphone.
- **PlaceEngine** [6]: This service visualizes motions and locations of staff working in a hospital. The IPS is implemented with beacon frame of wireless LAN.

Generally, the conventional systems of InLAS have been implemented individually, considering the purpose of service and the IPS to be used. There is no standard in how to represent and manage the indoor location information. Thus, individual systems implement own proprietary methods.

### 2.3 Geographic Information System (GIS)

GIS is a computer system designed to capture, store, manipulate, analyze, manage, and present all types of geographical data [15]. A wide range of data is used in the GIS. Meta-information is roughly divided into graphic information (maps, aerial photographs or satellite image, etc.), attribute information associate with the feature, geodetic system, projection method, reduced scale, accuracy, etc.

The GIS represents spatial data in two different formats: *vector* and *raster*. In this paper, we focus on the vector format. The vector format data consists of individual *points*, which are represented coordinates. Multiple points may be joined in a particular order to create a *line*, or joined into a closed ring to create a *polygon*. All the vector data fundamentally consist of a list of coordinates that define vertices, together with rules to determine whether and how those vertices are joined. Shapefile, a GIS file format, which uses the vector data and a variety of attribute data (such as property, feature and numeric, etc.) has become a typical standard format in the GIS industry.

### 2.4 Long-Term Goal and Scope of Paper

Figure 1(a) shows the implementation architecture of the conventional InLAS. As mentioned in Sections 2.1 and 2.2, each InLAS is tightly coupled with an IPS, and indoor location data and program are managed independently within each system. Therefore, one system cannot share or reuse the data and program of another system. As a result, each system tends to be complex and difficult to manage.

Our long-term goal is to establish an innovative architecture shown in Figure 1(b). In the proposed architecture, the *Indoor Position Query Service (IPQS)* accumulates spatial information from various IPS in a standardized way, and provides the information for various InLAS via application-neutral query API. Using the IPQS, we can achieve loose coupling of the InLAS and the IPS. This significantly improves efficiency and reusability of InLAS development. Moreover, by deploying the IPQS on cloud, it is possible to share and use the indoor information globally across multiple buildings.

As the first step towards the long-term goal, we propose a common data model, called *Data Model for Indoor Location (DM4InL)*, in this paper. As seen in the middle of Figure 1(b), DM4InL aims to prescribe a common data schema which does not depend any specific IPS or InLAS. As there is no de-facto standard format, DM4InL is developed by referencing to the GIS as introduced in Section 2.3.

## 3. DATA MODEL FOR INDOOR LOCATION

In this section, we propose DM4InL, which is a common data model for indoor location.

### 3.1 Requirements of DM4InL

The primary requirement for the DM4InL is to associate any object inside a building with location information, without depending on any purpose or application. More specifically, we aim to satisfy the following requirements R1-R4 in this paper:

- **Requirement R1:** The data model should be able to represent any position inside a building as location information with spatial and subjective attributes.

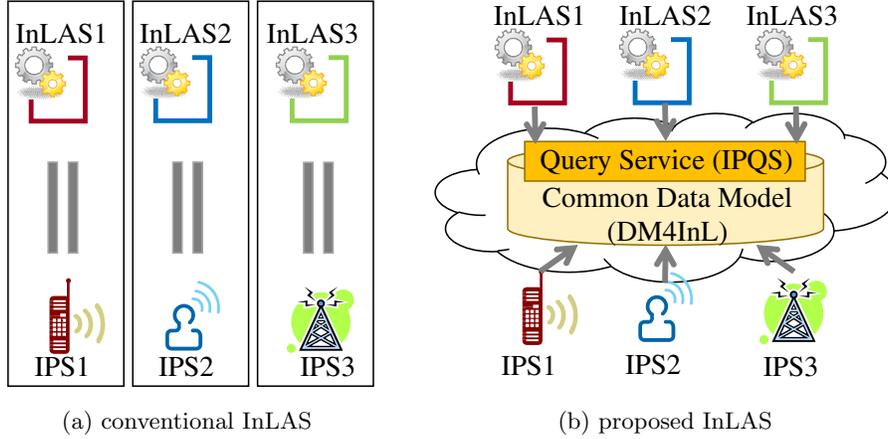


Figure 1: Two different architectures of InLAS

- **Requirement R2:** The data model should be able to associate any object inside a building (person, device, furniture and so on) with the location information.
- **Requirement R3:** The data model should be able to represent spots, routes and partitions as geographic elements within a building.
- **Requirement R4:** The data model should be able to provide seamless positioning, integrating indoor and outdoor location information.

### 3.2 Three Models of DM4InL

In order to satisfy the requirements, we construct DM4InL as a composition of the following three data models:

- **Location Model:** It defines indoor location information. Every position in a building is represented as a *relative coordinate* (3-dimensional offset) from a *reference point* of the building. Using the coordinate, we construct four geometric primitives: *local point*, *local line*, *local polygon* and *local space*. It also defines *global positions* by triplets of [longitude, latitude, altitude].
- **Building Model:** It represents spots, routes and partitions as geographic elements within a building. Each spot (route or partition) is located by a local point (a local line or a local space, respectively) in the location model. It also identifies every building with a reference point represented by a global position.
- **Object Model:** It represents various objects (people, furnitures, appliances, etc) within a building. Each object refers to a local point in the location model to represent its current position.

We will explain details of each model in the following subsections.

### 3.3 Location Model

The location model defines *indoor positions* and *geometric primitives*. First of all, indoor position must be defined on the foundation that every position belongs to a single building. Also, as discussed in [1], the 3-dimensional Cartesian coordinate is convenient for applications to use and represent the indoor locations. Based on the two reasons, we

represent every indoor position by 3-dimensional Cartesian coordinate system, and define each position as *relative coordinate* (3-dimensional offset) from a *reference point* of the building which the position belongs to. Furthermore, borrowing the idea of GIS, we define *local point*, *local line*, *local polygon* and *local space* in the location model.

Figure 2 shows the ER diagram of the proposed location model. The diagram follows the notation defined in [18]. A square represents an entity, whose schema is defined by multiple data items arranged in the right. An underlined item represents a primary key, and an underlined item with bracket defines a secondary key. Other items represent attributes. Data instances are listed under each entity. A relationship may be defined between a pair of entities, where

- $(+ \text{---} \in )$  represents a parent-child relationship,
- $(+ \text{---} \dots)$  represents a reference relationship,
- $(+ \text{---} \circ +)$  represents a sub-type relationship

The details of each entity are explained as follows.

#### (1) Local Point (LP)

LP defines every indoor position by a building to which the position belongs, and a 3-dimensional offset  $(x, y, z)$  from the reference point of the building. The primary key is a composite key consisting of a building ID (defined later) and a point ID. It also has a sequential code as a secondary key, so that external entities can easily refer to the point. The attributes are the coordinate values of LP in the 3D Cartesian system, by which external applications can locate the point in the building.

Figure2 contains five instances of LP, where two points belong to building B0001 and three points belong to building B0002. Points p0001 and p0002 are depicted in the right side of the ER diagram. Point p0001 represents a position, whose coordinate is 1.10m in x-direction, 7.50m in y-direction and 4.30m in z-direction from a reference point (GP01) of the building B0001.

#### (2) Local Line(LLN)

LLN represents a line constructed by two or more local points. Similar to LP, LLN have a composite key consisting of a building ID and a line ID. It also has a sequential code

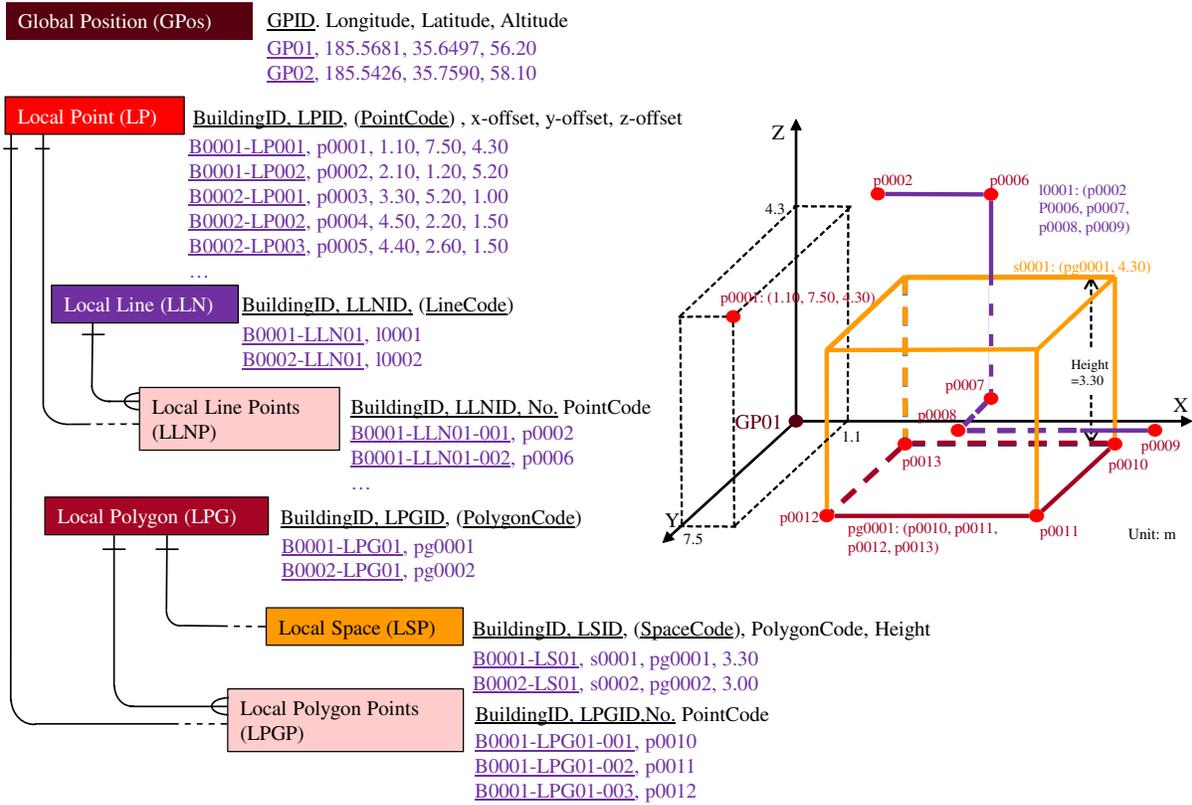


Figure 2: ER diagram of location model and schematic representations of instances

as a secondary key. Each line is defined by multiple points called local line points (LLNP). Each LLNP has a composite key consisting of an ID and a sequence number of the line, and a point code referring to a local point to locate the LLNP. Thus, a parent-child relationship exists from LLN to LLNP, and a reference relationship exists from LLNP to LP.

Figure 2 contains an instance of LLN (10001) that belongs to building B0001, constructed by p0002, p0006, p0007, p0008 and p0009. The line is depicted in the right figure.

### (3) Local Polygon (LPG)

LPG represents a polygon constructed by three or more local points. Similar to LLN, each polygon is defined by multiple points called local polygon points (LPGP). Each LPGP has a composite key consisting of an ID and a sequence number within the polygon, and a point code referring to a local point to locate the LPGP. Thus, a parent-child relationship exists from LPG to LPGP, and a reference relationship exists from LPGP to LP.

Figure 2 contains an instance of LPG (pg0001) that belongs to building B0001, constructed by p0010, p0011, p0012 and p0013. The polygon is drawn as a square in the right figure.

### (4) Local Space (LSP)

LSP represents a 3D space in a building. A 3D space is generally constructed by several polygons. However, taking convenience and characteristic of ordinary indoor space into account, we define a local space as a pillar-shaped space made by extending a local polygon to the z-axis direction.

LSP has a composite key consisting of a building ID and a space ID. It also has a sequential code as a secondary key. Attributes are a reference to a local polygon as a bottom of the space, and a height of the pillar from the bottom.

Figure 2 contains an instance of LSP (s0001) in building B0001, which is defined as a pillar of 3.30m height made from a local polygon pg0001. The space is drawn as a cube in the right figure.

### (5) Global Position (GPos)

GPos represents a global position, used for the reference point of a building. Since Requirement R4 suggests the seamless positioning among indoor and outdoor locations, we locate the reference point of every building as a global position. Each global position is represented by a triplet of longitude, latitude and altitude. It is referenced from a building in the building model, as is explained later.

## 3.4 Building model

The building model defines spots, routes, partitions within a building. A building can be regarded as a container that can contain various objects. From this viewpoint, the building model should represent the container itself, and geographic elements inside the container. The location of each geographic element is associated to a corresponding entity in the location model. Figure 3 shows the ER diagram of the building model. Each entity is explained below.

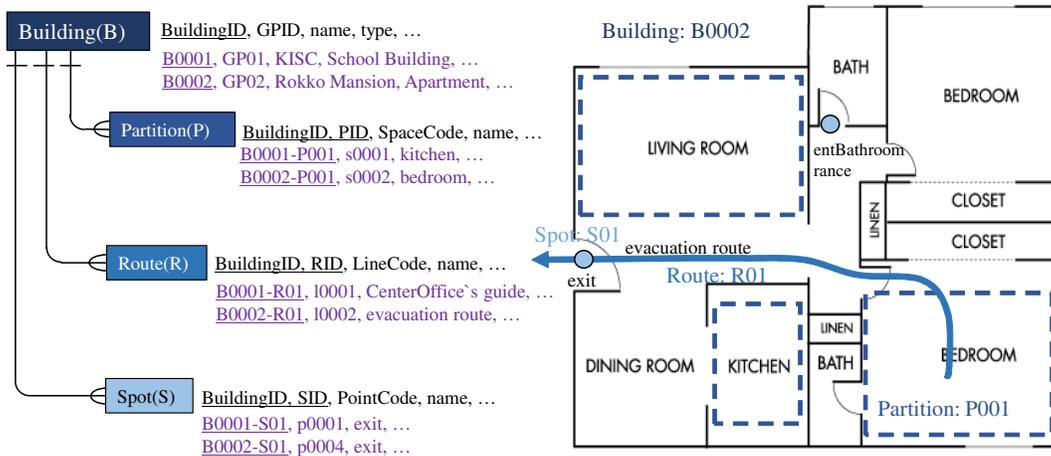


Figure 3: ER diagram of building model and schematic representations of instances

### (1) Building (B)

The building entity represents the existence of a building. A building is identified by building ID, which is a primary key. Attributes include a reference to a global position where the building exist (see (5) of Section 3.3), building name, type, etc. As mentioned in Section 3.3, every entity in the location model is associated with a building ID, which constructs a parent-child relationship. In other word, when a building is eliminated, all the location information in the building are also removed. The global position of a building serves as a reference point of the indoor position. Since every local position is represented by a relative coordinate from the reference point, we can convert the local position into a global position. This satisfies Requirement R4.

In the real world, a building can take various granularity. For example, when we consider a collective housing of apartments, there are choices: a building refers to the whole building, or a building refers to an apartment within the building. For this, we consider it reasonable to define a building for every living unit which a single IPS can cover.

### (2) Spot (S)

The spot entity represents a geographic element treated as a point, including a bathroom entrance, emergency exit, and so on. A spot has a composite key consisting of a building ID and a spot ID. Attributes include a reference to a local point that locates the spot, the name of the spot, etc.

### (3) Route (R)

The route entity represents a geographic element treated as a line, such as an evacuation route, a traverse of a residence, etc. A route has a composite key consisting of a building ID and a route ID. Attributes include a reference to a local line that draws the route, the name of the route, etc.

### (4) Partition (P)

The partition entity represents a geographic element treated as a space, such as a living room, a bedroom, a kitchen, etc. A partition has a composite key consisting of a building ID and a partition ID. Attributes include a reference to a local space that surrounds the partition, the name of the partition, etc.

The right side of Figure 3 shows schematic representations of instances over a floor plan of a house. A circle represents a spot, an arrow represents a route, and a dotted rectangle represents a partition.

## 3.5 Object Model

The object model represents various objects in a building. We assume that every object is movable, and is located by an indoor position when the object is in a building. In the real world, there are various kinds of objects (e.g., people, appliance, furniture) whose attributes may vary. Therefore, we first define an abstract entity that associates any object with the location information. We then define each concrete entity as a sub-type of the abstract entity. Further sub-types can be added when needed. Figure 4 shows the ER diagram of the object model. Each entity is explained below.

### (1) Object (O)

The object entity abstracts an arbitrary object in a building. An object is identified by an object ID, which is a primary key. Attributes include a type of the object, and a reference to a local point where the object currently exists. The type is a reference to a concrete object entity. The coordinate of the local point deduces a positional relationship between the point and a space (or a line). Hence, we can deduce a geographic element of the building (i.e., a spot, a route, or a partition), where the object currently exists.

The right side of Figure 4 illustrates four instances of the object entity. We can identify the current position of each object from its local point. We can also deduce a space from the position. For example, we can see that an object exists in the bed room. If the type of the object is people, the object is explained in details by a people sub-type entity.

### (2) Sub-Types of Object

A concrete object (e.g., people, appliance, furniture, etc.) is defined by a sub-type entity of the abstract object. A sub-type object defines attributes necessary for the type of the object. The sub-type and the abstract have the same object ID as a primary key. Therefore, we consider that a sub-type extends an object based on the type attribute.

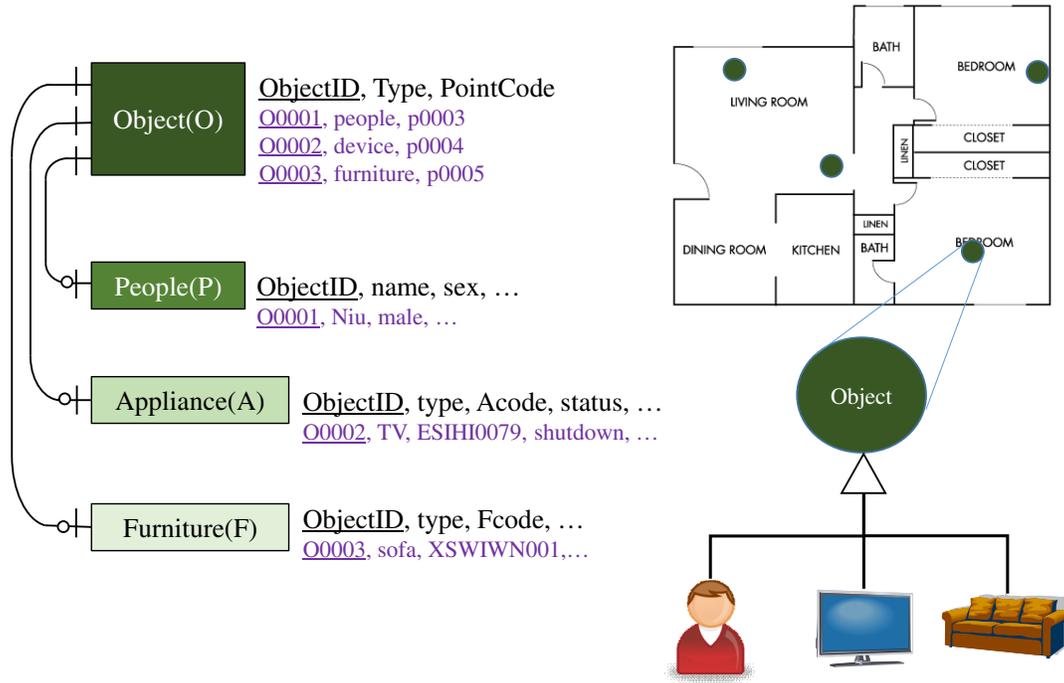


Figure 4: ER diagram of object model and schematic representations of instances

Figure 4 illustrates three sub-types: people, appliance and furniture. For instance, the people entity defines attributes of the name and gender. The appliance includes attributes of the appliance type, model number and state, etc. The furniture entity has the type and marker as its attributes. Note that further sub-types can be added as needed.

### 3.6 DM4InL

DM4InL is composed of the above three data models (i.e., location model, building model, and object model). Figure 5 shows an ER diagram of DM4InL, representing relationships among the three models. Every indoor location (i.e., LP, LLN or LSP) is associated with a single building, whereas every building involves more than one indoor locations. A building (B) is located by a global position (GPos). A geographic element in a building (i.e., spot, route or partition) refers to a location entity (LP, LLN or LSP, respectively). An object is located by a local point. These observations suit well the intuition of the indoor locations, objects, and buildings in the real world.

## 4. DISCUSSION

### 4.1 Sufficiency of Requirements

We here discuss how the proposed DM4InL satisfies Requirements R1-R4 in Section 3.1. First, by the definition of the location model (using the relative coordinate bound with a building) we can represent any position inside any building. Also, an element of the location model can be an attribute of a geographic element in a building. Hence, Requirement R1 is satisfied. By the definition of the object model, we can associate every object with a local point. Thus, Requirement R2 is satisfied. Since the building model represents spots, routes and partitions within a building, Re-

quirement R3 is satisfied. Every indoor location is associated with a single building, and every building has a reference point located by a global position. Combining the global position and the relative coordinate of an indoor location, we can convert the indoor location into a global position. Therefore, we can satisfy Requirement R4.

Note that Requirements R1-R4 are derived, purely considering how the common data model for the indoor location should be. They do not depend on any specific application or purpose. Thus, we believe that the constructed DM4InL is a *neutral* data model, which can be shared and reused by various IPS and InLAS.

### 4.2 Towards Query API for DM4InL

Using DM4InL, we can represent indoor location information of objects and geographic elements in an application-neutral form. To allow a variety of services and applications to use the data easily and efficiently, it is essential to implement API that implements typical queries for the DM4InL. For this, we are currently developing the following API:

(1) **Position API:** It queries location information of a given spatial element or an object. For instance, getObjectPosition(objectID) returns the current position of a given object, and getBuildingPosition(buildingID) returns a global position of a given building.

(2) **Attribute API:** It queries attributes with respect to an object or a spatial element. For instance, getObjectsInPartition(partitionName) returns all objects within a given partition.

Design and implementation of the API will be left for the next step of our research project.

### 4.3 Limitations

Indeed, there are a few limitations in the current version of DM4InL. A limitation is that we do not yet consider the

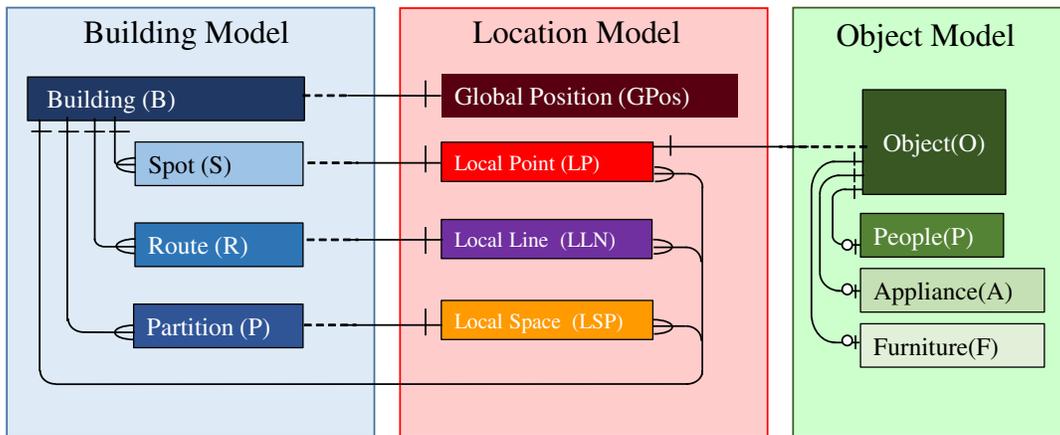


Figure 5: DM4InL as a composition of 3 models

time concept in the model. The proposed model is not supposed to represent past or future locations. Introducing the time concept allow us to derive *uncertainty* [7] of the location data, or to estimate the future location of the object. Although the time concept enhances the expressivity of the model, it increases the complexity as well. We are investigating whether or not the time should be included in DM4InL.

Another limitation is in the representation of local space. As seen in Section 3.3, we defined each space as a pillar-shaded space just for convenience. We need to evaluate that our definition can sufficiently cover practical InLAS.

Finally, the object model may be refined further so as to represent dynamic *context* of every object. Typical contexts include the current activity of a person, a status of a device, and direction and placement of a furniture.

#### 4.4 Related Work

Leonhardt et al. [7] proposed an approach that constructs a *generic query layer* between location sensors and positioning systems. Although their approach is similar to our long-term goal (see Figure 1(b)), it should be applied to a lower layer between the IPS and physical sensor devices. Their hierarchical model also deals with *uncertainty* and *data conflicts* under an unreliable indoor positioning environment. These issues are not yet considered in our DM4InL. Together with the time concept discussed in Section 4.3, we will investigate them in our future work.

Kim et al. [5] proposed *Indoor Spatial Data Model (ISDM)*, which uses CityCML to define location data for 3D indoor location-based services. ISDM can define *topologies* among spatial elements using the 3D object topology model, which is more expressive than our building model. However, it lacks a way to explicitly associate indoor objects with locations, as defined in our object model.

## 5. CONCLUSION

In this paper, we proposed *Data Model for Indoor Location (DM4InL)*, which prescribes a common data schema to represent indoor location information. By composing three data models (i.e., the location model, the building model and the object model), DM4InL represents location information of various objects inside a building in a standard format. The proposed method contributes to loose coupling of In-

LAS and IPS, which will significantly improve the efficiency and reusability in the InLAS development.

Our future work includes the evaluation of the data model with practical use cases of InLAS, as well as the design and implementation of the query API for DM4InL. Using the DM4InL and the query API, we will also develop the indoor position query service (IPQS), which is our long-term goal. Finally, the extension of DM4InL (w.r.t. the time concept, object contexts, local space, etc.) should be considered carefully in order to address pragmatic issues (such as uncertainty and data conflicts).

## 6. ACKNOWLEDGMENTS

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (C) (No.24500079, No.24500258), Scientific Research (B) (No.26280115), Young Scientists (B) (No.26730155)] and Kawanishi Memorial ShinMaywa Education Foundation.

## 7. REFERENCES

- [1] I. Afyouni, C. Ray, and C. Claramunt. Spatial models for context-aware indoor navigation systems: A survey. *J. Spatial Information Science*, 4(1):85–123, 2012.
- [2] C. Kevin, F. Eoghan, J. Lunney, J. Santos, D. Woods, and A. McCaughey. An evaluation of indoor location determination technologies. *J. Location Based Services*, 5(2):61–78, Jun 2011.
- [3] K. Hattori, T. Fujii, Y. Kado, and B. Zhang. Verification of the indoor user position and direction presumption system by using the two dimensional marker (in Japanese). *J. Information Processing Society of Japan. Ubiquitous Computing Systems (IPSJ. UBI)*, 2008(18):203–207, feb 2008.
- [4] H.-S. Kim, D.-R. Kim, S.-H. Yang, Y.-H. Son, and S.-K. Han. An indoor visible light communication positioning system using a RF carrier allocation technique. *J. Lightwave Technology*, 31(1):134–144, Jan 2013.
- [5] Y.-J. Kim, H.-Y. Kang, and J. Lee. Development of indoor spatial data model using CityGML ADE. *J. ISPRS - International Archives of the*

- Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-2/W2:41–45, 2013.
- [6] Koozyt Inc. About placeengine, 2011. Retrieved September 7, 2014 from <http://www.placeengine.com/showe/about>.
- [7] U. Leonhardt and J. Magee. Multi-sensor location tracking. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom’98, pages 203–214, New York, USA, 1998. ACM.
- [8] D. Manandhar and H. Torimoto. Opening up indoors: Japan’s indoor messaging system, IMES, 2011. Retrieved September 11, 2014 from <http://gpsworld.com/wirelessindoor-positioningopening-up-indoors-11603/>.
- [9] Navizon Inc. Navizon indoor triangulation system, 2014. Retrieved September 8, 2014 from <http://www.navizon.com/product-navizon-indoor-triangulation-system>.
- [10] NEC Corp. Smartlocator (in Japanese), 2014. Retrieved September 7, 2014 from <http://jpn.nec.com/engsl/pro/smartlocator/index.html>.
- [11] NTT Data MSE Corp. Shoplat (in Japanese), 2014. Retrieved September 7, 2014 from <https://www.nttdocomo.co.jp/service/convenience/shoplat>.
- [12] T. Ogasawara, H. Igaki, A. Inoue, and T. Hoshi. Velsy: Development support system for indoor location platform (in Japanese). *J. Human Interface Society*, 15(2):131–150, 2013.
- [13] A. Pratama, Widyawan, and R. Hidayat. Smartphone-based pedestrian dead reckoning as an indoor positioning system. In *System Engineering and Technology (ICSET), 2012 International Conference on*, pages 1–6, Sept 2012.
- [14] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom’00, pages 32–43, New York, USA, 2000. ACM.
- [15] S. Shekhar and H. Xiong. *Encyclopedia of GIS*. Springer Publishing Company, Incorporated, 1st edition, 2007.
- [16] J. Sookman. Guardly releases industry-first integrated indoor positioning system to provide responders with floor and room-level accuracy of mobile emergency callers, February 2013. Retrieved September 7, 2014 from <http://blog.guardly.com/guardblog/2013/02/26>.
- [17] S. Ting, S. Kwok, A. H. Tsang, and G. T. Ho. The study on using passive RFID tags for indoor positioning. *J. Engineering Business Management*, 3(1):9–15, 2011.
- [18] K. Watanabe. *Introduction to Data Modeling for Database Designing (in Japanese)*. Nippon Jitsugyo, Tokyo, 2003.