Interoperable Spatial Information Model and Design Environment Based on ucR Technology

Yukihiro SHIGESADA†, Nonmember, Shinsuke KOBAYASHI††, Member, Noboru KOSHIZUKA†††, Member, and Ken SAKAMURA†††, Nonmembers

SUMMARY Context awareness is one of the ultimate goals of ubiquitous computing, and spatial information plays an important role in building context awareness. In this paper, we propose a new interoperable spatial information model, which is based on ucode relation (ucR) and Place Identifier (PI), for realizing ubiquitous spatial infrastructure. In addition, we propose a design environment for spatial information database using our model. Our model is based on ucode and its relation. ucode is 128 bits number and the number itself has no meaning. Hence, it is difficult to manage the relation between ucodes without using a tool. Our design environment provides to describe connection between each ucode visually and is able to manipulate data using the target space map interactively. To evaluate the proposed model and environment, we designed three spaces using our tool. In addition, we developed a web application using our spatial model. From evaluation, we have been showed that our model is effective and our design environment is useful to develop our spatial information model.

key words: spatial information system, ubiquitous computing, ubiquitous spatial infrastructure, spatial databases and GIS, web based services

1. Introduction

Many researchers are interested in ubiquitous computing, pervasive computing, or the Internet of Things [1] which is a new paradigm of information and communication technology. One of the important technologies to realize such paradigms is context-awareness. Context is defined as “any information that can be used to characterize the situation of an entity.” [2] Generally, context includes four categories; location, identity, activity and time. Especially, location plays an important role to realize ubiquitous or pervasive computing. Spatial modeling is required in order to utilize location information on computers efficiently.

Traditionally, coordinate methods have been used for modeling spaces. However, it is difficult to match human context because coordinate method can only represent physical place. It cannot represent location for human under-
difficult to develop the proposed spatial model without using our environment because our model is based on dynamic linking, so the model is not human friendly but machine friendly. In our method, connection between each ucode can be seen visually and can be manipulated using the target space map interactively.

The main differences between this work and previous ones [9], [10] are that we have shown the design environment of our spatial model, and have shown that the environment is effective to describe spatial information. ucode is 128 bits number and the number itself has no meaning. Hence, it is difficult to manage the relation between ucodes without using a tool. In experimental result, we have shown that an indoor place, such as a museum or a department store, can be described within several hours. In addition, we have developed an application using our spatial model, and showed the effectiveness of our spatial model.

The rest of the paper is organized as follows. In the next section, we discuss about related work. In Sect. 3, the proposed model is described. In Sect. 4, we describe the proposed method and environment using our spatial model. In Sect. 5, we describe experiment using our model. Finally, the conclusion is described.

2. Related Work

In this section, we describe methods of location modeling for context awareness, and international standards for location modeling.

Many researchers have already proposed methods of location modeling. Paper [11] surveys location modeling. It mentioned that four queries are needed in order to realize location-based application; position, nearest neighbor, navigation, and range queries. According to the paper, there are two categories of location modeling; one is geographic model and the other is symbolic model. Symbolic model consists of four categories; set-based, hierarchical, graph-based, combined (set-based and graph-based). In addition, hybrid models that support geometric and symbolic have been proposed. It mentioned two types of hybrid location models. The first approach, called the subspace approach, stores geometric information for every modeled location. The second approach only stores geometric information for some locations, leading to partial subspaces. Our method is one of the hybrid subspace models, and also includes combination of graph-based and set-based symbolic model. For example, Aura [12] is a hybrid subspace model. It supports hierarchical symbols and geographic coordinates. Euclidean distances between points even if they are in different CRSs (CRS: Coordinate Reference System) can be calculated in Aura. However, it does not support connection between places. Hence, it is difficult to calculate navigation path. Leonhardt [13] proposed a hybrid subspace model. It has symbolic and geographic coordinates. However, there is no discussion about navigation. The proposed modeling method can support not only navigation query among different CRSs but also other three queries; position, nearest neighbor, and range. Moreover, our method provides generating new identifiers to solve a problem of PI framework that we mentioned in Sect. 1.

ISO/TC211 discusses standardization about digital geometric information services. ISO/TC211 consists of several projects. ISO19111 is the project for spatial referencing by coordinates, and ISO19112 is the project for spatial referencing by geographic identifiers. As we mentioned before, the feature of ISO19111 is to describe the location in the globe uniquely and accurately by using coordinate. Meanwhile, coordinate such as latitude and longitude is not used in daily life, we commonly use an identifier of geographic location. Hence, geographic location based method is required. ISO19112 is one of the activities to specify geographic location based method. However, it is difficult to interoperate geographic identifiers, because there are many dedicated geographic location based specifications proposed by industries and communities.

Place Identifier is a specification to unify such identifiers, and it is discussed at ISO/TC211 as an international standard (ISO19155). In PI, the term PI is also used for the geographic identifier. Therefore, in the rest of the paper, we use the term “PI framework” for the framework of PI, and the term “PI” for the geographic identifier of the PI framework. In the PI framework, PI interfaces are defined in order to communicate between PI platform and users [14]. There are 6 kinds of PI interfaces such as “PI registration interface” to register spatial reference system and PI, and “PI conversion interface” to convert a geographic identifier to another geographic identifier which represents the same place. By implementing web API PI interfaces for every geographic database which is represented by the PI framework, we can interoperate geographic identifier and can create worldwide geographic database. In the PI framework, PI_LocationInstance is defined to describe data of PI. It can represent not only the geographic identifier, but also the position and the relationship between other PIs (Table 1). “alternativeGeographicIdentifier” is defined in order to register the alternative geographic identifier of location. However, this information cannot be used as a formal PI. When several PIs, which indicate the same place, are registered, almost of all data except for geographicIdentifier are duplicated. This is a problem of data efficiency and coherency.

G/S (Geo-Browser/Server) model [15] is the distributed network structure system under the internet environment based on the geographic information exchange
standard—HGML (Hyper Geographic Markup Language) which is XML-based standard. HGML is compatible with the open OGC [16] geographic information standards—GML [17] and KML [18]. HGML describes the spatial entity geometric elements (point, line, surface, arc, etc.) and their attributes as well as interrelationships. However, users cannot directly use authorized identifier in this approach. Comparing it to web service, users have to use search engine such as Google to refer a web page in this approach. In our approach, users are able to refer a web page directly using URI. This means that users can refer the spatial information with a little overhead.

Geoads [19] is a location-based service, which is a streaming music service with geographical advertisement. To realize the system, they propose Geoinfo gateway. Geoinfo gateway uses two standard specifications. One is the Mobile Location Protocol (MLP) published by Open Mobile Alliance (OMA). The other is Terminal Location API, which is one of Parlay X Web Services published by the Parlay Group. Location is based on coordinate and shape of place. Hence, they do not consider about geographic identifier.

### 3. Proposed Spatial Model

In this section, we describe the proposed spatial model. Before describing our model, we explain ucode and ucR which are basic technologies of our approach.

#### 3.1 ucode and ucR

“ucode” is an identification number which can be issued by anyone, anytime for anything. “ucode” can be issued for content, information which does not exist in the real world, and for more abstract concepts as well as tangible objects and places in the real world. The ucode system is a 128 bit fixed length identifier system.

“ucode” is simply an identification number. There is no relationship between the number and the attribute and the meaning of the target to which the ucode was assigned. Our system stores such information as attributes and meanings in databases. The attribute and the meaning can be retrieved from the databases by using the ucode as a key.

The model, which describes the context of the real world, is called “ucode Relation model” (ucR model). The ucR model can represent the context using the relationship among objects and places. In the model, ucodes are assigned to objects and places. “ucode” which identifies relationships between objects and places is called “relation ucode”. Information which can become the attribute value of objects and places to which ucode is assigned such as strings, web page URLs, numerical values, etc. is called an atom in the ucR model.

The triplet of (ucode, relation ucode, ucode) or (ucode, relation ucode, atom) is called ucR unit. This is the basic unit used in the ucR model. In addition, if a triplet is compared to a sentence where the relation ucode is the predicate, the ucode that corresponds to the subject of the sentence is called a subject ucode, and the ucode that corresponds to the object or the complement of the verb is called an object ucode. The structure of triplet is quite simple. Yet it has a high representation power for context in the real world.

#### 3.2 Spatial Model Using ucR

Spatial ucode is an identification number that indicates a location. Spatial ucode has relations among other spatial ucodes. “isPartOf” and “hasPartOf” relations define hierarchy of location. For example, suppose a shop of UNIQLO that is located in the sixth floor of ABC department store. In our spatial model, each location such as building, floor and room is defined using a spatial ucode. In Fig. 1, each edge such as “isPartOf”, and “region” is relation ucode. When UNIQLO is in the sixth floor, the spatial ucode of the UNIQLO connects to the spatial ucode of the sixth floor, and the spatial ucode of the sixth floor connects to the spatial ucode of the ABC Department Store. The relation can be defined using triplets (“#100321”, “isPartOf”, “#100320”),

![Fig. 1 Example of spatial information model using ucR.](image-url)
and ("#100322", "isPartOf", "#100321"). Atoms related to a spatial ucode are geographic identifiers of the spatial ucode. A relation ucode between a spatial ucode and an atom represents the location type of the geographic identifier.

In Fig. 1, relation ucodes such as “address”, “building”, and “floor” stand for the location type of geographic identifiers. Atoms such as “Shinagawa Tokyo”, “ABC Department Store”, and “6th floor” are geographic identifiers of spatial ucodes. In addition, spatial ucode has a relation of region ucode. Region ucode indicates coordinate location using coordinate reference system (CRS). In the example, representative point of “ABC Department Store” is POINT (35.625913, ...). This point is represented using WGS84 CRS in this example.

To specify the CRS of a location, the relation ucode “crs” is used. The reason why we adopt this data model is to realize defining CRS flexibly. There are many CRSs. They are not only international CRS but also domestic CRS. In Fig. 1, the CRS ucode, which is related to the spatial ucode of the 6th floor, specifies a domestic CRS that represents the floor map of the ABC department store. Hence, specifying various CRS is important.

Moreover, in this model, it is very easy to register new geographic identifiers that represent the same location of another geographic identifier stored in the database. For example, in the case of Fig. 1, suppose the ABC Department Store has a nickname like “XYZ Tower”, the geographic identifier “XYZ Tower” can be efficiently registered to the database by storing the triplet ("#100320", “building”, “XYZ Tower”) into the database. Unlike RDB, there are no coherency problems.

3.3 Extension of Our Spatial Model for PI

In our model, CRS can be specified flexibly by using relation ucode “crs”. Using the same approach, other spatial information model can be extended by adding a relation ucode to a spatial ucode. When considering the PI framework, Fig. 2 shows an example of extension of our ucR spatial model to interoperate with the PI framework. Geographic identifiers such as “Shinagawa Tokyo” and “UNIQLO” in Fig. 2 correspond to the PI of the PI framework. Therefore we call these geographic identifiers as “PI” in the rest of this paper. Spatial ucodes have a piLRS (PI Location Reference System) relation. LRS is a spatial reference system by geographic identifiers based on ISO19112, which is specified in the PI framework. ucode of PI LRS has locationType relations which correspond with the property of location types of the geographic identifiers. In this example, there are six locationTypes.

Users can refer PIs by using our system as follows. The first step is getting the list of location type from PI LRS. The second step is getting the list of PIs which has the same location type specified by users. This operation is called as “PI search”. For example of Fig. 2, suppose that users specify “building” location type for search. Users obtain two results; “23rd XX building” and “ABC Department Store”.

Table 2 shows correspondence between our model and PI LocationInstance. In our model, geographicIdentifier, which is equivalent with PI, is represented by the property of spatial ucode. These properties are represented using relation ucodes, which are specified in PILRS. “alternativeGeographicIdentifier,” which means a same place, is not used in our model because multiple PIs can be registered to a spatial ucode. Position information can be obtained from region ucode. Relations of parent and child are represented using relation ucode “isPartOf” and “hasPartOf”, respectively.

3.4 Creating New Identifiers

Using our extended ucR spatial model, we realize creating new geographic identifiers by combining with other PIs. For example of Fig. 2, geographic identifier of “UNIQLO” can be represented by the combination of PIs such as “UNIQLO at 23rd XX building”, “Room 603 at 23rd XX building”, and “UNIQLO at 6th floor at ABC Department Store”. All of them are the same location, but they are not registered in the database directly. In our method, combination of multiple PIs can be used as a new identifier. Then, users can refer various locations by specifying combined identifiers even if they are not registered in the database.

In the PI framework, PIs must be registered in PI database when using PIs. This operation keeps consistency and accuracy of PIs. However, it is not easy to register other representations of a PI flexibly after the PI has been
registered in the database. The reason why we adopt this approach is that required geographic identifier depends on the type of application. When you look at pedestrian navigation system, the name of shop or department store is an important identifier. When you look at building management, the name of building or room number is important. Taking variety of identifiers into consideration, it is difficult to register enough possible identifiers. Hence, combination of identifiers is useful in order to solve the problem.

3.5 Connection

Euclidean distances between locations can be calculated by coordinates. However, information for connection between locations is required for navigation. In our model, connection ucode represents the connection between two locations. A connection ucode has four properties; “length” is the length of the path between two locations, “type” is type of the path such as passages, stairs and elevator, “spatial ucode” is spatial ucodes of the two locations, “path” is the shape of the path. If the “spatial ucode” property contains only one spatial ucode, it means that the connection is one way. If the “path” property is omitted, it means that the shape of the path is straight line. If spatial ucode A and B are connected, both spatial ucode A and B have a “connection ucode” property to the same connection ucode. If there are two paths between two locations, you can create two connection ucodes and set different length, type and path attributes to each connection ucode.

For example, in the case of Fig. 3, spatial ucodes “place A” and “place B” belong to the CRS “2nd floor” and the spatial ucodes “place C” and “place D” belong to the CRS “1st floor”. There are three connections; “place A between place B”, “place A between place C” and “place B between D”. The connection between “place B” and “place D” is one way connection.

Because the value of the property “length” of the connection ucode is independent from CRS, the distance between spatial ucodes can be calculated even if these spatial ucodes belong to different CRSs.

3.6 Queries for Location-Aware Computing

Paper [11] surveys location model for ubiquitous computing, and mentioned the following functions were required for location model; Position queries, nearest neighbor, navigation, and range queries. In this section, we describe these queries using the proposed model.

Position queries are basic queries to get object position. Generally, the definition of a position requires some form of coordinates that are not only global but local CRSs. In our model, each spatial ucode represents a position of geographic coordinates using one or more region ucodes. Spatial ucode has hierarchical structure, and the highest spatial ucode indicates a global position. The other spatial ucodes indicate a local position. Spatial ucode has not only region ucode but also has relation ucode of locationType. The region ucode indicates geographic coordinates, and the locationType indicates symbolic coordinates. Both of them can be converted.

Nearest neighbor query is the search for the closest object to a certain location. In our model, each spatial ucode has connections of spatial ucode by connection ucodes. These connections indicate a distance between spatial ucodes. Connection ucode can connect spatial ucodes even if they are in different CRSs. The distance information of connection ucode is independent from CRSs. Hence, the distance between locations of different CRSs can be calculated.

Navigation is to find paths between locations. In our model, we use connection ucodes to find paths. Since connection ucode has length, the length of path can be calculated using connection ucode. In addition, using type of connection ucode, context aware navigation, such as selecting elevator path for wheel chair, can be realized.

A range query returns all objects within a certain geographic area. In our model, region ucode has an attribute of geographic range. In addition, topological relation “contain” can be represented because spatial ucode has hierarchical structure. We realized the range query using region ucode and spatial ucode.

4. Design Flow and Development Environment

Our spatial model can represent many kinds of places, however, it is costly to develop our spatial model because we need to describe relationships between places and hierarchical information such as building, floor, room number and so on. Describing relationships, visualization is important because it helps us to understand the location of place in a map and relationships between each place. We developed design flow and development environment to describe our spatial model. Our development environment is a web application, so you are able to use it on any kinds of platforms. In the following sections, we describe the design flow and environment.
4.1 CRS and LRS Definition

In order to define a place, we need to define CRS firstly. When you define the place in outside, you use CRS such as WGS84. However, when you look at building, we have no common CRS. Hence, we need to define a CRS when you want to describe a space. In CRS definition, name and a URL of a base map image for a space are needed.

LRS is also needed to describe a place. In our model, LRS needs to define location types such as “address”, “building”, “floor”, and “room number”. LRS represents the PI LRS of the PI framework (Fig. 2). When you register location types in LRS definition, you can describe location attributes using the location type.

In our development environment, you can easily input CRS and LRS definition by edit form of the web application.

4.2 Overview of the Place Editor

In the next step, we define places. Firstly, we select a CRS that is defined in the previous step. Figure 4 shows the place editor. Left hand side of this figure is a map that you specified at the CRS definition. You can scroll and zoom up and down the map by mouse operation. Right hand side is a list of registered places. Blue lines of the list such as “elevator” and “gallery” are location types which you registered in the LRS definition. Other black lines are registered places. A red line represents a selected place. You can select a place from the list or blue and green circles on the map by click operation. In this figure, a place named “Egyptian Funerary Arts” is selected. You can select a circle of selected place, circle of every place which is connected to the selected place, and arrow of every path which is connected to the selected place by click operation. For example in Fig. 4, a blue circle and three green circles and three red arrows are highlighted because the blue circle (which is “Egyptian Funerary Arts”) is selected and green circles (which are passages) are connected to the blue circle by three red arrows (which are paths). To distinguish connected places, each connected circle has a colored border. The border color of the selected circle corresponds with the text color of the path data in the attributes of selected place (bottom left of Fig. 4, and Fig. 5).

Figure 5 shows attributes of the selected place.

coordinates: (1015, 1111)
selected data (click here to delete (d))
(1007, 837) ucode=6132, parent ucode=6070 (click here to edit (p))
click here to insert new name (n)
Egyptian Funerary Arts (gallery) (click here to edit)
path data (click here to insert new path (i))
number of path = 3 (click below list to delete path)
to 6126 ucode = 6134 length = 64 type = 0
to 6124 ucode = 6133 length = 54 type = 0

Fig. 4 Place editor.

Fig. 5 Attributes of selected place.

In our development environment, you can easily input CRS and LRS defintion by edit form of the web application.
a ucode. Parent ucode is the last four digits of the parent spatial ucode. The fifth line indicates the name of the place and its location type. In this example, the selected place has a name “Egyptian Funerary Arts” whose location type is “gallery”. The seventh line indicates the number of path connected to the selected place. The list of the path data is displayed under the seventh line. In this example, there are three path data. The first four-digit of the path data is the spatial ucode of the connected place, “ucode” is the connection ucode, “length” is the length of the path, “type” is the type of the path (in Fig. 5, type attribute is not set in each path), and the name of the connected place is displayed in the bracket. In this example, every name of the connected place is empty because every connected place is passage and has no name. The color of path data corresponds with the border color of the circle in the map that represents the connected place.

### 4.3 Place and Relationship Registration

In order to register a new place, select a location type listed at right hand side in the place editor (Fig. 4). When you select a location type, a pop up view to input new place name, coordinates, and parent ucode is displayed. The coordinate values and parent ucode are automatically filled in the input box. The default values of the coordinates are the center position of the displayed map. After registering a place, you can adjust the coordinates by drag and move operation to the circle displayed in the map, or change coordinates by keyboard. The default value of the parent ucode is the value of the parent ucode which was used to register previous place. This is because most of the places in the same CRS have common parent ucode. When you want to register a new place which has the same location type that you have registered previous time, you can press Enter key instead of selecting a location type. It can save input time because you do not have to specify the same location type.

After place registration, you can create a path between places. The step to create a path is as follows. First, select a target place from the editor, and then click the sixth line of the Fig. 5. Instead of the mouse operation, you can press i or [ key which are displayed in the bracket in the sixth line for short cut operation. When you move the pointing device to the destination place in the map editor, a purple arrow appears which connects between the target place and the destination place (Fig. 6). After selecting the destination place, a pop up view is displayed to input attributes which are the length of the path and the type of the path. The default value of the length is automatically calculated by the coordinates of the target place and the destination place. You can also create a unidirectional path.

In order to add a relationship to a place of other CRS which is not displayed in the map, for example elevator, stairs, and escalator and so on, you can specify spatial ucode to make a relationship. In our environment, orange slant arrows represent these relations (Fig. 6).

### 4.4 Edit Operation

To edit the name of the place, double-click the name of the place in the right side of the place editor, or select the place and click the name of the place in the Fig. 5. To edit the coordinates of the place, double-click the coordinates of the place in the right side of the place editor or double-click the circle of the place in the map of the place editor. You can also edit the coordinates of the place by dragging the circle of the place.

To define another name to a place, select the place and click the fourth line in the Fig. 5. A pop up view to input the location type and the name is displayed. For example, you can define two names to a place such as “Japanese Gallery” whose location type is “gallery” and “603” whose location type is “room” (Fig. 7).

### 4.5 Inter-floor Definition

Figure 8 shows the inter-floor definition view. You can register elevation view as a CRS, and define relationships among

---

**Fig. 6** Relationship registration.

**Fig. 7** Multiple name definition.

**Fig. 8** Inter-floor definition.
floors. When you look at high-rise building, all elevators may not stop every floor. A part of elevators is used for lower floors, and a part of elevators is used for higher floors. In our environment, you can specify such cases using elevation view. In Fig. 8, vertical red arrows in the left, middle, and right represent elevators. The orange slant arrows represent relationships between an inside of elevators and an entrance of the elevators for each floor.

Other blue circles represent spatial ucodes of the floor such as “First floor”, “Women’s bag floor” and so on. For example, every place in the first floor of the building has the parent ucode of the “First floor.”

4.6 Register Multiple Data by Web Browser

If multiple place data are stored in a database or in a data file such as CSV format, it is not efficient to register such data by our GUI place editor. The place editor uses query string of the URL to request update query to the ucR database. Therefore, it is easy to register multiple place data in a database or a data file to the ucR database by translating place data to the update query string of the URL and send it to the CGI of the place editor by HTTP.

5. Experiment

To evaluate the efficiency of our model, we implemented APIs for PI operations in our extended ucR spatial information system. In addition, we described spatial data of three spaces.

5.1 Implementation of PI APIs

The specification of the PI framework has been discussed in Japan and JIPDEC (Japan Information Processing Development Corporation) [20] creates detailed specification of the PI framework every year. In the 2009th “Report of detailed design specification of PI” [21], it is specified that every PI spatial information services must implement several interfaces called PI interface. We implemented major PI interfaces (Table 3) in our extended ucR spatial information system. We also implemented extended PI interface called Convert compound PI API which converts a single PI to combination of several PIs.

For implementation, we used Sesame 2.6.8 [22] for database and implemented Web APIs for spatial database. The reason why we selected Web API is that Web API is widely used in Web 2.0 technologies. You can mash-up our system with many services. The specification of an execution machine was quad core Intel Xeon 2.67 GHz with 16 GB memory, and the operating system was Linux version 2.6.18-238 (CentOS 5.4).

For experimental setup, we described spatial data of Ginza Mitsukoshi department store [23], Museum of Fine Arts, Boston [24], and Tokyo Station [25].

Figure 9 shows a part of data we stored. We designed three layers in Ginza Mitsukoshi. The building layer consists of the data of building, and in this case, the building of the Ginza Mitsukoshi is the only data stored in this layer. There are three PIs that represent the Ginza Mitsukoshi building. One is the address, and two are the name of the building (in Japan, Ginza Mitsukoshi is also called “Mitsukoshi Ginza Store”).

The floor layer consists of the data of the floor. In the Ginza Mitsukoshi, “Women’s wear floor” is placed in the fourth floor and the fifth floor. Therefore, we created a spatial ucode, which represents the “Women’s wear floor” and related the spatial ucode of the fourth floor and the fifth floor as the child of it. The “floortype” property represents the type of commodity displayed in the floor. The spatial ucode of each floor has two PIs such as “5th floor” and “5F”.

The shop layer consists of the data of the shop in each floor. Every spatial ucode of the shop layer is the child of a spatial ucode of the floor layer.

5.2 Specifications of Implemented PI Interfaces

To show the efficiency of our implementation, we performed some experiments by using the implemented PI interfaces. First, we describe the specification of searchPI interface used in the experimentation.

SearchPI performs search query to the database to get place identifiers. Parameters of searchPI are “search word”, “CRS”, “location type”, and “geographic extent”. Search word specifies the name of a place, CRS specifies a CRS name which includes the PI to search, location type specifies the name of a location type such as “shop”, and geographic extent specifies the bound of searchPI. In this
implementation, geographic identifier that has building location types can be specified as a geographic extent.

The spatial server returns the result using XML format. “pi” is place identifier. “count” is the number of obtained result. We adopt tag URI as the format of PI because PI specification describes tag URI. Figure 10 shows an example of PI description using tag URI. The first column is URI scheme. This column is fixed value because we use tag URI. The second one is SRS (Spatial Reference System). In this case, we can select CRS as a SRS. The third one is year. In the PI specification, place name may change with time even if they are the same place. For example, several hundred years ago, we call Tokyo as Edo. The fourth one is geographic identifier of PI, which is the name of a place such as Ginza Mitsukoshi, Tokyo Station and so on.

For example, when you specify the “location type” is “shop”, and the “geographic extent” is “Ginza Mitsukoshi”, you will get the result as follows. This query requests to get every shop PI of Ginza Mitsukoshi. The list of shop PIs of Ginza Mitsukoshi and the number of result are returned. We stored a CRS for each floor in the Ginza Mitsukoshi named such as “Ginza Mitsukoshi 1F:ubin.jp” which is used in the SRS part of the PI.

```
<xml version="1.0" encoding="utf-8">
  <result>
    <pi>tag:Ginza Mitsukoshi 6F:ubin.jp,2011:pi:Neil barrett</pi>
    <pi>tag:Ginza Mitsukoshi 10F:ubin.jp,2011:pi:Burberry</pi>
    <pi>tag:Ginza Mitsukoshi 3F:ubin.jp,2011:pi:Tony burch</pi>
    ...
    <count>341</count>
  </result>
```

5.3 Evaluation of Development Environment for ucR Spatial Model

To evaluate our development environment, we made spatial data from three spaces as follows; “Ginza Mitsukoshi Department Store,” “Museum of Fine Arts, Boston,” and “Tokyo Station.” In the evaluation, we defined all places and connections that include floor, elevator, escalator, stairs, entrance, restroom, information, and rooms described in the map on each Web site.

Table 4 shows registered places and connections using our spatial model. It took about several hours to describe each building using our environment. When we make spatial model, coordinate information is important. Using our environment, it is easy to describe coordinate value because you are able to describe the spatial model using a map. In addition, you can validate described spatial model because the place and connection are visualized. Visualization is important to describe spatial information.

We also stored about 500 place data of buildings in Ginza which are stored in the relational database. As we have described in the previous section, the place editor uses query string of the URL to request query to the ucR database. We created a script to translate the building data in the relational database to the query string of the URL and send it to the CGI of the place editor by HTTP. In the experiment, we registered over 500 places of buildings by this method in a few minutes. Moreover, we registered test data for performance checking. The test data includes 10 CRSs, 10240 places, and 10558 connections. Each CRS has 1024 places and these places are arranged in the pattern of grid (x, y). These horizontal and vertical lines are 32 places, and outer periphery and horizontal lines are connected. The point (0, 0) of CRS(n), which means the number n CRS, connects to the point (0, 0) of CRS(n – 1) and CRS(n + 1), respectively. The name of CRS(n) is test n. The location type of places is shop and names of places are shop n, which means the place number m of CRS(n).

5.4 A Case Study: Art Navigator Application

In order to evaluate our spatial information model, we have developed an application that is a navigation system for art. This application provides art information that includes title, artist, year, location where you can see, and so on. The location is defined by using our spatial information system. The art information is also defined by using ucR database system. One of the strong benefits of our ucR database system is extendibility. We only add an ucR relation between the art information ucode and the location information ucode. It is similar to adding a link of web. Hence, it is very simple to describe location information for each item once you have developed spatial information using our model.

We developed an iPhone application shown in Fig. 11. When user specifies search parameter: title, artist, or classification of art (Fig. 11 (a)), user can retrieve a list of art information from our server.

After user specified search parameters and executed search function, he can select an art from the list (Fig. 11 (b)). He will see the information of art in next view. The view includes art information such as title, artist, year, and location where the art is displayed (Fig. 11 (c)). When he touches a pin that is displayed on a map, he can see a pop that includes the name of location. In addition, user will see the indoor position when he touches the indicator that is included in the pop of location (Fig. 11 (d)). In Fig. 11 (d), a red pin represents the place of the selected art and blue...
pins represent the places of arts placed in the same room.

The following shows a result of search query. In search query, we realized the following procedure. The first step is searching art data from the database, which is defined using a Museum schema. The second step is searching spatial ucodes which are included in the art data, and getting the spatial information of the art. Next step is making URL that indicates the location of art. We developed a CGI to show a location of indoor place. The URL to display a location like Fig. 11 (d) includes the CGI and parameters to indicate the specified location.

Finally, we obtain the XML data. In the result, art information contains the following information: title, artist, year, classification, image, ucode of this art, and place information that includes a link to the proposed special information model. Place tag contains CRS, PI, URL that indicates indoor place information, the name of building, and coordinates location of the building. The application uses this result to display art information.

```xml
<art>
  <title>Homme nu debout</title>
  <artist>Pablo Picasso</artist>
  <year>1930</year>
  <classification>Paintings</classification>
  <image>http://www.example.com/ex.jpg</image>
  <artUcode>0001C000000000010000001B7F05</artUcode>
  <place>
    <<crs>The National Art Center Tokyo Gallery 2E</crs>
    <p1>Homme nu debout</p1>
    <url>http://www.example.com/result.php?num=2&amp;crs=The National Art Center Tokyo Gallery 2E&amp;p1=Homme nu debout</url>
    <building>National Art Center Tokyo</building>
    <coordinateOfBuilding>POINT (35.666198 139.726517)</coordinateOfBuilding>
  </place>
</art>
```

5.5 Discussion

In this section, we discuss about the ability of the model in making context aware applications, granularity, and feature of our spatial information model.

As we described in Sect. 3, four queries are needed in order to realize location-based application; position, nearest neighbor, navigation, and range queries. From the paper [11], in order to realize four queries, a location model should provide: Object positions, distance function, topological relations, and orientation. In our method, object positions can be modeled using geometric and symbolic coordinates, and multiple local and global CRSs. Distance can be also calculated using connection ucode in our model. Topological relation includes spatial containment and spatially connected, and our method supports both of topological relation. In addition, orientation is also included in our model. Connection ucode includes orientation to support one way like escalator.

As mentioned in paper [11], the location model should support requirements that mentioned above with minimal modeling effort. Actually, modeling a location using our method is a difficult task when you do not use any tools. That is because designers should make a huge number of relations and coordinate positions for modeling. To cope with the problem, we proposed a development tool. Coordinate information can be set using map graphically. Since some relations such as hierarchical information are generated using other relation information, you can omit to describe the information. In our evaluation, we modeled three places, and each modeling time was several hours. This means that our tool reduces modeling effort.

In our spatial information model, granularity of space can be defined variously such as country, city, building, and room using ucode. In Ubiquitous ID architecture [7], things and locations in the real world have identifiers called ucode. When you want to add location information to a thing, you can define a link between the location and the thing directly. It is useful that you are able to make a database of things, which is related to space information. In addition, you can flexibly set attributes to it, such as name, input date, price,
and so on, because we adopt dynamic linking approach. Hence, our model is useful when you consider information system of space and things.

For example, when you look at supply chain management system, you would like to manage the location of products. Attributes of products include name of products, lot number, production date, and so on. Products move from place of production to a shop through markets. To manage current location of products, you should change location information of products. In our model, it is easy to change the location information because you just change a relation ucode to another location ucode. In another example, management of sensor network node is realized by using our information model. Each sensor node has ucode, sensor information such as temperature, humidity, atmosphere pressure, etc., and location of sensor. When you make database of sensor nodes, you add ucode relations between sensor node ucode and information. It is the same manner to realize supply chain management system that we mentioned before. Please note that “location” is modeled as same as object positions mentioned above in our method. Currently, some delivery service companies use private location ID[27] to manage items of customers and their delivery. Location IDs are assigned to their branches, delivery areas, etc., and develop management systems for goods of connecting customers and their current locations. However, these systems usually use one global CRS like WGS84, postal address, and common knowledge of local area to manage their locations. When supposing a huge building, a delivery staff does not have knowledge of the building, and he has a location ID that has attributes of postal address, latitude and longitude, it is difficult to find the destination. Furthermore, when they have a local navigation system with the building CRS, there is no interoperability among other information such as postal address, latitude, and longitude. In our method, a spatial ucode has relations of multiple local and global CRSs. That is a difference between a private location ID of current delivery service companies and our method. In addition, the location ID itself includes not only number like spatial ucode but also name described in Sect. 3.4. Users can specify the place using spatial ucode or location name such as “UNIQLO at 23rd XX building”, and “Room 603 at 23rd XX building”.

The reason why our model is effective is that spatial information is separated from other information such as name, date, and so on. Hence, you can maintain spatial information easier than other systems. When you consider a system that is developed by using ad-hoc manner, which means that they do not consider well-defined spatial model, it is difficult to manage the location information. For instance, if you manage the location by using name, such as building, city, and so on, it is difficult to keep consistency because the name of building or city might be changed. Suppose you manage the location by using a location number, it is better than using name of location because it is easy to change the meaning for each number when the name of location is changed. That is why our system is easy to maintain database of spatial information.

In addition, since each space is defined by ucode, the length of data for each space is fixed 128 bits. In previous work, space ID such as geographic identifier, or URI is used to describe space. As we mentioned in Sect. 1, geographic identifier is useful to specify a location. However, data length is not fixed size. Taking RFID tag into account, the size of data is very important. Using our spatial model, RFID tag can be used as a location tag. When you read an RFID tag that is put on a door, you will get your location accurately.

Table 5 shows execution time of searchPI. “Geographic extent”, “search word”, “location type”, and “CRS” are place parameters explained in the previous chapter. “# of results” is the number of search result of searchPI API. “Ave. time” is average search time when we executed searchPI ten times. “Std. Dev.” is standard deviation. We executed three cases of searchPI for each geographic extent. First case is to specify only “location type”, second is to specify only “search word”, and third is to specify both “search word” and “location type”. We also executed two cases of searchPI for test data which stores more than 10,000 places and connections (Table 4) to evaluate the performance of searchPI for huge number of data. First case is to specify a “CRS” to search every location stored in the “test1” CRS. Second case is to specify a search word “shop1_l” to search every location whose name contains “shop1_l”. As you can see in Table 5, we realized most of search function within a second. The forth and tenth result show that, when the number of places was more than three hundred or one thousand, search time was longer than others. In the fifth result, it took more than three seconds.

Table 5 shows execution time of searchPI. “Geographic extent”, “search word”, “location type”, and “CRS” are place parameters explained in the previous chapter. “# of results” is the number of search result of searchPI API. “Ave. time” is average search time when we executed searchPI ten times. “Std. Dev.” is standard deviation. We executed three cases of searchPI for each geographic extent. First case is to specify only “location type”, second is to specify only “search word”, and third is to specify both “search word” and “location type”. We also executed two cases of searchPI for test data which stores more than 10,000 places and connections (Table 4) to evaluate the performance of searchPI for huge number of data. First case is to specify a “CRS” to search every location stored in the “test1” CRS. Second case is to specify a search word “shop1_l” to search every location whose name contains “shop1_l”. As you can see in Table 5, we realized most of search function within a second. The forth and tenth result show that, when the number of places was more than three hundred or one thousand, search time was longer than others. In the fifth result, it took more than three seconds.

The reason why it took three seconds is that location type was not specified as a parameter in this API. On the other hand, for example, the sixth result was 0.683 seconds, when the parameter of “location type” was specified to “shop.” Specifying “location type” parameter reduces candidates of PI. Hence, the execution time is faster than the fifth search result.

Table 6 shows execution time of four functions: position, navigation, nearest neighbor, and range queries. We applied two or three locations for each query. “Avg. time” is average time when executing ten times and “Std. Dev.” is standard deviation. As shown in Table 6, results can be

| geographic extent | search word | location type | CRS | # of results | Ave. time (sec) | Std. Dev. |
|-------------------|-------------|---------------|-----|--------------|----------------|-----------|
| Boston            | Japanese    | gallery       |     | 66           | 0.489          | 0.111     |
| Boston            | Japanese    | gallery       |     | 5            | 0.568          | 0.128     |
| Mitsubishi       | shop        |               |     | 5            | 0.045          | 0.010     |
| Mitsubishi       | Celine      | shop          |     | 3            | 2.825          | 0.589     |
| Mitsubishi       | Celine      | shop          |     | 3            | 2.395          | 0.437     |
| Tokyo Station    | elevator    |               |     | 29           | 0.193          | 0.036     |
| Tokyo Station    | Shinkansen  | elevator      |     | 62           | 1.443          | 0.331     |
| Tokyo Station    | Shinkansen  | elevator      |     | 5            | 0.061          | 0.008     |
| shop1_l          |             |               |     | 1024         | 4.791          | 0.433     |
| test1             |             |               |     | 135          | 1.548          | 0.406     |
two places which belong to different CRSs. Moreover, because the distances between two places are longer than the first search, it took more than 5 seconds for search. However, these searches showed that our method was able to search route between two places which belong to different CRSs. Figure 12 shows the result of the second search. We might improve the performance when applying other algorithms, however, route search algorithm is out of scope in this paper.

6. Conclusion

We proposed a new spatial information model. The main contribution of this work is making an interoperable spatial information model, which can extend to use other spatial reference system specification. In addition, we achieved making a flexible spatial information system. In our system, combination of location identifiers can be used as a new location identifier. It gives a powerful solution when users would like to refer various geographic identifiers without registering every possible identifier. Modeling effort is required when using our modeling method because there are many relations in the model. To solve the problem, we proposed the design environment for our model. Through the evaluation, we have shown that our environment is useful and efficient to make it easy to describe places and their relationships.

In the case study, we showed that our spatial information model and ucode relation model was effective to manage information of things. The reason why our model is effective is that spatial information is separated from other information. It provides to maintain spatial information easier than other systems. In addition, we showed that the proposed location model includes enough information to realize location queries for ubiquitous computing.

Acknowledgments

This work is partly supported by SCOPE (Strategic Information and Communications R&D Promotion Program) from Ministry of Internal Affairs and Communications of Japan.

References

[1] L. Yan, Y. Zhang, I.T. Yang, and H. Ning, The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems (Wireless Networks and Mobile Communications), Auerbach Publications, 2008.
[2] A.K. Dey, “Understanding and using context,” J. Personal and Ubiquitous Computing, vol.5, issue 1, pp.4–7, Feb. 2001.
[3] ISO 19111:2007, Geographic information — spatial referencing by coordinates, 2007.
[4] ISO 19112:2003, Geographic information — spatial referencing by geographic identifiers, 2003.
[5] R. Plews and S. Kawano, “Place identifiers: Describing position, location, and virtual world references,” IEEE Pervasive Computing, vol.10, no.1, pp.90–93, Jan. 2011.
[6] ISO/TC 211: Geographic information/Geomatics, http://www.isotc211.org/, accessed Feb. 19, 2012.
[7] T. Kamina, N. Koshizuka, and K. Sakamura, “Embedding legacy keyword search into queries for the ubiquitous ID database,” Proc. 2nd International Conference on Network-Based Information Systems, pp.263–272, Springer-Verlag, 2008.
[8] W3C, “Resource description framework (RDF): Concepts and abstract syntax,” http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/, accessed Feb. 2012.
[9] Y. Shigesada, S. Kobayashi, N. Koshizuka, and K. Sakamura, “ucR based interoperable spatial information model for realizing ubiquitous spatial infrastructure,” Proc. IEEE COMPSAC 2010, pp.303–210, July 2010.
[10] S. Kobayashi, Y. Shigesada, N. Koshizuka, and K. Sakamura, “Interoperable spatial information system architecture based on ubiquitous ID infrastructure,” IEEE Computer Software and Applications Conference Workshops, pp.435–440, July 2011.
[11] C. Becker and F. Durr, “On location models for ubiquitous computing,” Personal and Ubiquitous Computing, vol.9, no.1, pp.20–31, 2005.
[12] C. Jiang and P. Steenkiste, “A hybrid location model with a computable location identifier for ubiquitous computing,” Proc. 4th International Conference on Ubiquitous Computing, pp.246–263, 2002.
[13] U. Leonhardt, Supporting location-awareness in open distributed systems, PhD thesis, Department of Computing, Imperial College London, 1998.
[14] R. Plews and S. Kawano, “A conceptual framework for the description of place identifiers,” ISO Workshop on Address Standards: Considering the Issues Related to an International Address Standard, Copenhagen, Denmark, pp.71–79, May 2008.

[15] X. Guo, F. Miao, H. Wang, and G. Du, “Initial discussion on the architecture of a new spatial information network service model based on the digital earth,” IEEE International Conference on Environmental Science and Information Application Technology, pp.406–410, July 2009.

[16] OGC, “Open geospatial consortium,” http://www.opengeospatial.org/, accessed Feb. 19, 2012.

[17] C. Portele, “OpenGIS® geography markup language (GML) encoding standard,” OGC 07-036, OGC standard, 2007.

[18] T. Wilson, “OGC® KML™,” OGC 07-147r2, OGC standard, 2008.

[19] S. Kuo, S. Lin, B. Wu, Y. Tseng, and C. Shen, “GeoAdS: A middleware architecture for music service with location-aware advertisement,” IEEE International Conference on Mobile Adhoc and Sensor Systems, pp.1–3, 2007.

[20] JIPDEC, “Japan institute for promotion of digital economy and community,” http://www.jipdec.or.jp/, accessed Feb. 19, 2012.

[21] JIPDEC, “Report of detailed design specifications of PI” (in Japanese), http://www.jipdec.or.jp/archives/dpc/gxml/contents/pi/spec/pi-specification_report_2009.pdf, accessed Feb. 19, 2012.

[22] OpenRDF, “Sesame,” http://www.openrdf.org/, accessed Aug. 1, 2012.

[23] Ginza Mitsukoshi, http://www.mitsukoshi.co.jp/, accessed Feb. 19, 2012.

[24] Museum of Fine Arts Boston, http://www.mfa.org/, accessed Feb. 19, 2012.

[25] JR East Tokyo Station, http://www.jreast.co.jp/estation/stations/1039.html, accessed Feb. 19, 2012.

[26] IETF, “Tag URI,” RFC 4151, Oct. 2005.

[27] SG Systems, “Biz-BLUE,” http://www.sg-systems.co.jp/service/images/info_biz-blue.pdf, accessed Aug. 16, 2012. (in Japanese)

Shinsuke Kobayashi received M.E. and Ph.D. degrees in computer engineering from Osaka University, Japan in 2000 and 2003, respectively. He was a research associate in interfaculty initiative in information studies, the University of Tokyo from 2003 to 2006. He was an associate professor in the University of Tokyo from 2006 to 2011. Currently, he is a senior researcher at YRP Ubiquitous Networking Laboratory. His research interests include computer architecture, VLSI design, hardware/software co-design, and embedded system design. Currently, he is interested in ubiquitous computing technology such as RFID, sensor network, smart environment, and so on. Dr. Kobayashi is a member of IEEE Computer and Communication Society, and Information Processing Society of Japan.

Noboru Koshizuka is Professor in Department of Applied Computer Science, the University of Tokyo, Japan. He is also Vice Director of YRP Ubiquitous Networking Laboratory. He was born in Tokyo, Japan in 1966. He received the B.S., M. S. and D. S. degrees in information science in 1989, 1991, and 1994 from the University of Tokyo, Japan. Since 1990, he has been participating in the TRON (The Realtime Operating system Nucleus) Project. For about 15 years, he has been researching ubiquitous computing, embedded systems, human interface, and computer networks. Currently, his main research interests are ubiquitous computing, embedded real-time systems, operating systems, and computer networks. Prof. Koshizuka is a member of the IEEE, ACM, and Information Processing Society of Japan.

Ken Sakamura was born in Tokyo in 1951. At present, he is a professor of the Interfaculty Initiative in Information Studies at the Graduate School of the University of Tokyo, director of the YRP Ubiquitous Networking Laboratory, the chairman of the T-Engine forum and the uID center. He has been elected as fellow and the golden core member of the IEEE Computer Society. He has been the leader of the TRON project since 1984. In this capacity, he has designed the TRON open computer system architecture, which will be useful for ubiquitous computing of the future. Today, the real-time operating systems based on the TRON specification are used for engine control on automobiles, mobile phones, digital cameras, and many other devices, and are believed to be the most popular such operating systems of the kind all over the world. Aside from the computer systems, he has been involved in the artistic design of electronic appliances, furniture, houses, buildings, urban landscape, and museums, all of which contain the networked embedded computers based on the TRON specification. Prof. Ken Sakamura, Ph.D. was awarded Takeda Award 2001, the Medal with Purple Ribbon from Japanese government in 2003, Okawa Prize in 2004, and Japan Academy Prize in 2006.

Yukihiko Shigesada was associate professor in Department of Faculty of Intercultural Communication, Hosei University, Japan, from 2005 to 2011. He is currently professor in Hosei University since 2011. He was born in Japan in 1971, and received the B.S., M.S and Ph.D. degrees in information science in 1994, 1996 and 2002 from the University of Tokyo, Japan respectively. He is also a researcher at YRP Ubiquitous Networking Laboratory. His research interests include distributed operating system, embedded computer, and ubiquitous computing. Currently he is mainly interested in ubiquitous computing technology such as geographic information system. Prof. Shigesada is a member of the IEEE, ACM, and information Processing Society of Japan.