Journal of Korean
Data & Information Science Society
2007, Vol. 18, No. 4, pp. 1157 ~ 1170

A Location Context Management Architecture of Mobile Objects for LBS Application[1)]

Yoon Ae Ahn[2)]

Abstract

LBS must manage various context data and make the best use of this data for application service in ubiquitous environment. Conventional mobile object data management architecture did not consider process of context data. Therefore a new mobile data management framework is needed to process location context data. In this paper, we design a new context management framework for a location based application service. A suggestion framework is consisted of context collector, context manager, rule base, inference engine, and mobile object context database. It describes a form of rule base and a movement process of inference engine that are based on location based application scenario. It also presents an embodiment instance of interface which suggested framework is applied to location context interference of mobile object.

Keywords : Inference Engine, LBS, Location Context, Mobile Object

1. Introduction

In ubiquitous environment, location based service offers value added service on the base of user location. Location based service satisfies several requests which are needed to build ubiquitous computing environment, so it will be the best test not only for itself but for realization of ubiquitous computing environment for everybody in everyday life in the future. Especially, in order to meet the expectation of the people who want to live quite convenient daily life, it first should consider personalized context awareness service on demand. Rough location

1) The research was supported by a grant from the University Restructuring Program (funded by the Ministry of Education and Human Resources Development) of Chungju National University.

2) Professor, Dept. of Multimedia Engineering, Chungju National University, Chungju, Chungbuk, 380-702
E-mail : yeahn@cjnu.ac.kr

information sensed by sensor is meaningless data that gives no effect to user, so it gives some meaning of context to the data by converting location information and offers suitable context awareness service to user.

Currently location based service system must manage various location context data and make the best use of this data for application service in ubiquitous environment. But, conventional mobile object data management architecture did not consider process of context data. Therefore a new mobile data management framework is needed to process location context data.

In this paper, we design a location based context management framework for various location based application service in ubiquitous environment. A suggestion framework is consisted of context collector, context manager, rule base, inference engine and mobile object context database. It describes a scenario which is applied the suggestion framework to a specific application and presents an instance by embodying an application interface which offers users a service. In chapter 2, we discuss related study of location based service and context awareness system. We design a location based context awareness framework in chapter 3. In chapter 4 we suggest an instance by using an application scenario and meet the conclusions in chapter 5.

2. Related Works

Location information managing system of mobile object for location based service manages and stores a process of continuous location change of objects like vehicles, planes, ships, PDAs, laptops, etc, and operates operation function to offer various query processing function to users by using stored location data (Wolfson, 2001). DOMINO is a real-time location tracking prototype of mobile object applied location inference technique of dead reckoning (Wolfson, 1999). CHOROS studies problems which the design and embodiment of spatio-temporal database system has, suggests the structure of spatio-temporal database system, and embodies it partly (Lema, 2001). DEDALE is a developed prototype in order to do modeling spatio-temporal data and processes queries by using constraint database (Grumbach, 1999). MOMS is a mobile object managing engine for distribution vehicle management (Kim, 2002). This prototype offers not only the function of pre-existing vehicle tracking system, but also location information of the mobile vehicle in the past and at present.

There are some by Cliff Randell & Henk L. Muller (Randell, 2000) and Farringdon (Farringdon, 1999) as instances of study on human's kinetic state inference by using an acceleration sensor among location based context awareness systems. An instance of context recognition system which activates by sensor-networking of central processing management is in Priyantha(2000) and Harter(1998). There is a research on user behavior state inference for

context-aware computing in mobile device in Baek(2006). In Park(2004) for ubiquitous environment, a structure of location based context awareness system is proposed and it classifies a context recognition hierarchical structure into sensor layer, qualitative relation layer, spatio-temporal layer and semantic layer, and designs a system. There is a proposal of indoor location based context supporting system for health-care home service in Ahn(2006). The service system is consisted of a group manager, a dynamic binder, a mobile proxy, a real-time manager and a scheduler.

Mobile objects belong to spatio-temporal data, having the characteristic that the spatial object changes its position and shape with move. Mobile objects have two basic types, which are mobile points and mobile regions. Mobile points are the positions or locations of the object changing over time. They are people, animals, cars, and tanks, and express the value of spatial point at specific time. Mobile regions include shapes as well as positions of objects changing over time, and they can be grown or reduced. Data like glaciers, storms, and cancers belong to mobile regions, and express the value of the spatial region at a specific time. Mobile object forms three dimensions by extending the coordinate (x, y) in the two-dimensional space to the time axis t. Time values of the time axis may be past, present, and future. In particular, the future value can be obtained by a certain operation, while past and present values are stored in the database. Here, we only consider the location coordinate (x, y) for the spatial attribute and the valid time for temporal attribute of the mobile object (Erwig, 1999; Forlizzi, 2000).

3. Location Context Management Architecture for LBS

3.1 Context Management Framework

In this paper, we design a location based context management framework for location information management of mobile object which is an issue in location based service application. We in the proposed location based context management framework can use sensors like GPS, mobile communication network in order to achieve location information of mobile object. The proposed framework is consisted of context collector, context manager, rule base, inference engine, and mobile object context database. And it also has a structure that enables to serve users by embodying context awareness application interface for particular application.

In Figure 1 context collector manages location context received through sensor and location awareness system. Context manager classifies context which sensing done and stores it in the database and manages. It integrates sensed context into a structured form in the base of user ID, device ID, and location ID. Rule base is consisted of the domain knowledge and location based space knowledge. Domain knowledge stores knowledge information of particular situation which is applied in

the application system. Location based space knowledge stores operating rules of spatio-temporal operators. Inference engine plays a role that infers new location context and offers whenever user situation or environment changes. Inference engine uses domain knowledge and location knowledge stored in knowledge database. Mobile object context database stores context with sensing. Sensor data is consisted of ID of object, X coordinate value and Y coordinate value on the 2-dimension space, direction of object, and so on from real sensor and is different up to application.

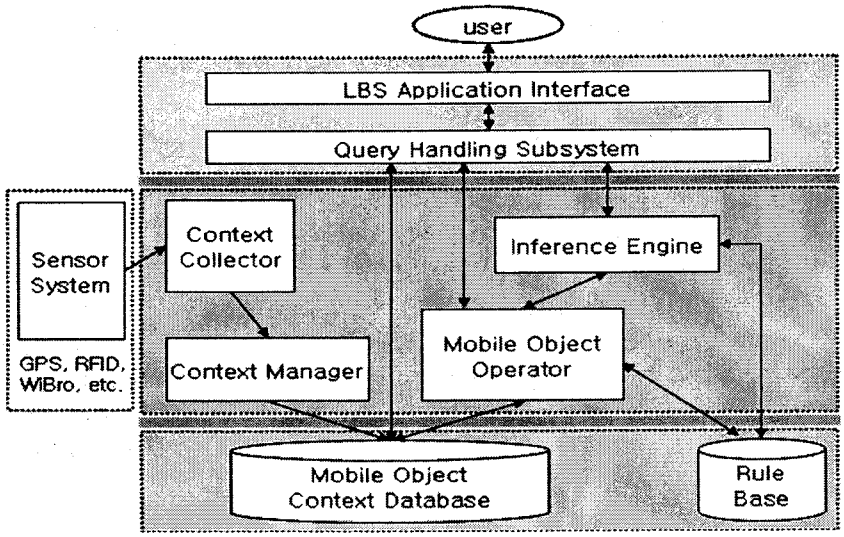


Figure 1. Context Management Framework for LBS

3.2 Context Collector

Context data of the mobile objects are received by context collector from the outside location supply server. The context collector read context data packets of the mobile objects that are transferred real-time through TCP/IP. The received location context packets are consisted of hexadecimal type. The received values are changed into a text type data by packet converter and stored in the context database. Processing is the same as following algorithm in Figure 2.

Algorithm ContextCollector
Input: serverIp(address for the real time location supply server)
Output: parsedPacket(packet data for the database)
Begin
    socket ← new Socket(serverIp, portNum)
    byte[] loadPacket ← new byte[32]
    in ← new BufferedInputStream(socket.getInputStream())

```
    For i = 0, 31
        readOneByte ← (byte) in.read()
        loadPacket[i] ← readOneByte
    mo_id ← ((loadPacket[4]*255+loadPacket[5])*255+loadPacket[6])*255+loadPacket[7]
    x ← ((loadPacket[8]*255+loadPacket[9])*255+loadPacket[10])*255+loadPacket[11]/100
    y← ((loadPacket[12]*255+loadPacket[13])*255+loadPacket[14])*255+loadPacket[15]/100
    time ← ""+loadPacket[16]+"-"+loadPacket[17]+"-"+loadPacket[18]
    parsedPacket[0] ← mo_id
    parsedPacket[1] ← x
    parsedPacket[2] ← y
    parsedPacket[3] ← time
    Return parsedPacket
End
```

Figure 2. Algorithm for the Context Collector

In Figure 2, the ContextCollector module gets a URL of a sensor system for input data and suggests a packet array of the received context data for output value. The ContextCollector module extracts mo_id, x, y, and time value in the received context data packet to store actually in the context database. The extracted value of the hexadecimal type is changed into decimal type data and is stored in the parsedPacket for returning.

3.3 Context Manager

When a loss of real-time location data for a mobile object is occurring, the context manager infers location information of the mobile object. The context manager estimates a location loss value by location inference module. The final location data is modeled adequately for a table schema of the database and stored in the context database. The Figure 3 explains process of the context manager.

```
Algorithm ContextManager
Input: parsedPacket(packet data for the database), type(moving trajectory)
Output: contextTuple(the final location data array)
Begin
    inputData ← parsedPacket
    mo_id ← parsedPacket[0]
    temp_x ← parsedPacket[1]
    temp_y ← parsedPacket[2]
    t_now ← parsedPacket[3]
    If (temp_x == 'null' or temp_y == 'null') Then
        If (type == 1) Then
            future_location ← futureLinear(mo_id, t_now)
            inputData[1] ← future_location[0]
```

```
            inputData[2] ← future_location[1]
        If (type == 2) Then
            future_location ← futureSpline(mo_id, t_now)
            inputData[1] ← future_location[0]
            inputData[2] ← future_location[1]
        Else
            inputData[1] ← temp_x
            inputData[2] ← temp_y
    mo_id ← inputData[0]
    table ← searching a table in the MobileObject database for mo_id
    target ← "MobileHistory_"+table
    time ← inputData[3]
    x ← inputData[1]
    y ← inputData[2]
    Return contextTuple
End
```

Figure 3. Algorithm for the Context Manager

In Figure 3, the ContextManager module get an inputData array for input and append the inputData to the history table. Many tables are simultaneously stored in the MobileDB. Therefore a table name of the MobileObject which have an object identifier is searched using a mo_id in the inputData. The target variable has a Mobile_History table name. This target value is a history location data table name in the database.

3.4 Mobile Object Context Database

Context database stores and manages attributes of time and space of mobile object and uses established common DBMS. Context database is consisted of initial mobile object context table and mobile history context table. Attribute table, general attribute, location coordinates, and valid time the date observed actually of mobile objects observed at the beginning are recorded and stored in the form of (id, Code, Name, Type, VTs, VTe, X, Y) in the initial mobile object context table.

Table 1. Initial Mobile Object Context

id	Code	Name	Type	VTs	VTe	X	Y
100	901	13-corps	3	2007/05/01	2007/05/02	204	76
101	902	23-corps	2	2007/05/01	2007/05/02	304	52
102	903	33-corps	2	2007/05/01	2007/05/02	330	51

Table 1 is consisted of object identity(id), code(Code), object name(Name), X coordinate(X), Y coordinate(Y), the beginning point of valid time(VTs), the ending

point of valid time(VTe), and type(Type).

Moving history context table is a history table that records all mobile processes of objects which are stored in the initial mobile object context table and it contains location coordinates of objects and moving dates. It will be stored in the form of (Code, VTs, VTe, X, Y) and is consisted of object code(Code), X coordinate(X), Y coordinate(Y), the beginning point of valid time(VTs), and the ending point of valid time(VTe).

Table 2. Moving History Context

Code	VTs	VTe	X	Y
901	2007/05/02	2007/05/03	206	122
901	2007/05/03	2007/05/04	209	144
901	2007/05/04	2007/05/05	216	179

3.5 Rule Base

Rule base is consisted of a domain rules group and a facts group. Domain rules that are used to LBS application interfaces are stored in the group of domain rules. Rules are composed of conditional part (part of If~) and behavior part(part of Then~) like in Figure 4.

```
rule_exam() {
    if (element_of(_machine, _table) && exist_on(_machine, _nearest_load)) {
        _attack_1 = is_determined(_nearest_load);
        draw5_1(_attack_1, _weight_1);
    }
    if (element_of(_jede_2, _table) && form_of(_jede_2_reg, _triangle)
       && is_located(_jede_2_reg, _position)) {
        _attack_2 = is_determined(_position);
        draw5_2(_attack_2, _weight_2);
    }
    if (element_of(_jede_2, _table) && leaned_to(_jede_2, _direction)) {
        _attack_3 = is_determined(_direction);
        draw5_3(_attack_3, _weight_3);
    }
}
```

Figure 4. Example of Domain Knowledge

Figure 4 is an example of rule expressed by If statement in JAVA language and it is a rule that decides further main moving direction of object. Rule_exam presents a rule number and _machine, _table, _nearest_load, _attack_1, and _weight_1 show variables. The element_of(), exist_on(), is_determined(), form_of(),

and is_located() are rule execution methods. The fact group is a place that all the spatio-temporal computing results which are needed to operate rules for inference operation. Spatio-temporal facts stored like this will be deleted if new inference query is operated, and other facts will be generated and used.

3.6 Mobile Object Operators

Mobile object operator uses temporal and spatial attribute of object and operates spatio-temporal operation and search. Temporal topology operator operates an operation by using a front/rear relation and an inclusion relation of time. Temporal retrieval operator operates a search by using temporal attribute of data. Spatial topology operator operates an operation by using a topological relation of spatial data. Adjacent operator operates an operation by using an adjacent relation of spatial data and directional operator operates by using a directional relation of spatial data. Spatial searching operator operates a search by using a spatial attribute of data. Operating types managed by mobile object operator are the same as in Figure 5.

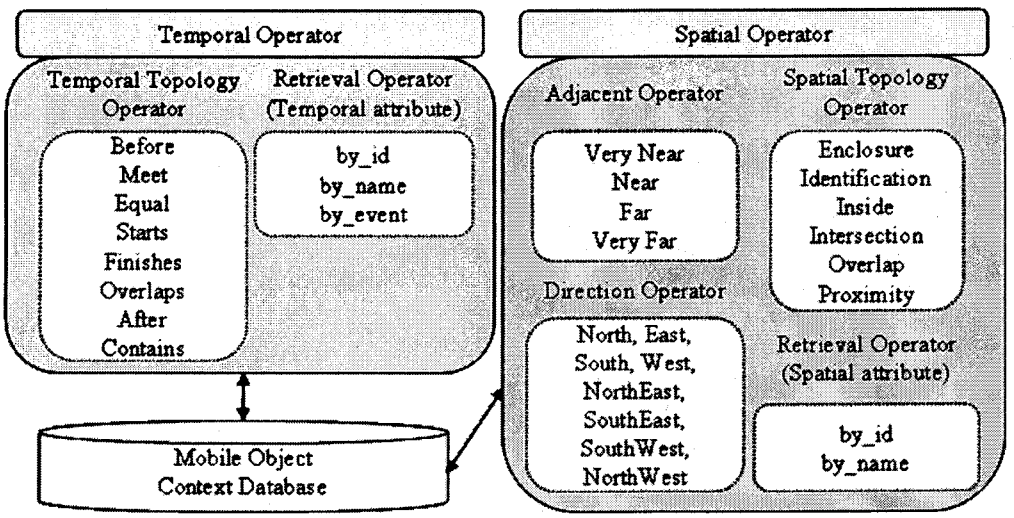


Figure 5. Spatio-temporal Mobile Object Operators

In this paper we define some operations that will be able to apply directly for LBS as Table 3. There are mdistance, visit, trajectory, length, minvalue, maxvalue, mintime, maxtime, velocity, and attime.

Table 3. Definition of Context Process Operators

operator	input	output	mean
mdistance	mpoint × mpoint	mreal	- a distance between two moving points at every time - mreal("moving real")
visit	mpoint × region	mpoint	- a position of moving point
trajectory	mpoint	line	- line(a curve data type on two-dimensional space) - trajectory : a projection of a moving point onto the plane
length	line	real	- length : a total length of line value
minvalue, maxvalue	mreal	real	- min or max moving real value
mintime, maxtime	mreal	time	- min or max time value
velocity	mpoint	mpoint	- production of time-dependent vector r(t) - moving point location coordinates
attime	mpoint × time	point	- point location coordinates at some special time t

For example, the attime operator computes location coordinates of mobile object at special past or future query time point as following Figure 6.

```
Algorithm AtTime
Input : mo_id(object identifier), time(a query time point),
        table(a table name), type(moving trajectory)
Output : location(a loacation coordinates at the time point)
Begin
    target ← "MobileHistory_"+table
    x, y ← searching a location coordinates at the time point in the target table
    If (search result == null) Then
        t_now ← the end time point in the target table
        If (type == 1) Then //a linear moving trajectory
            If (time < t_now) Then //a past query time point
                location ← pastLinear(mo_id, time) //estimate a past location
            Else location ← futureLinear(mo_id, time) //estimate a future location
        If (type == 2) Then //a curve moving trajectory
            If (t_now < time) Then //a future query time point
                location ← pastSpline(mo_id, time) //estimate a past location
            Else location ← futureSpline(mo_id, time) //estimate a future location
    location[0] ← x
    location[1] ← y
    Return location
End
```

Figure 6. Algorithm for attime Operator

In Figure 6 the AtTime module calculates the past, current, and future location value by comparing input time value with mobile objects in the MobileDB.

3.7 Inference Engine

Inference engine operates inference function by using spatio-temporal fact information managed by mobile object location context stored in database, domain knowledge stored in rule base, and mobile object operator. The process of inference engine is the same as Figure 7.

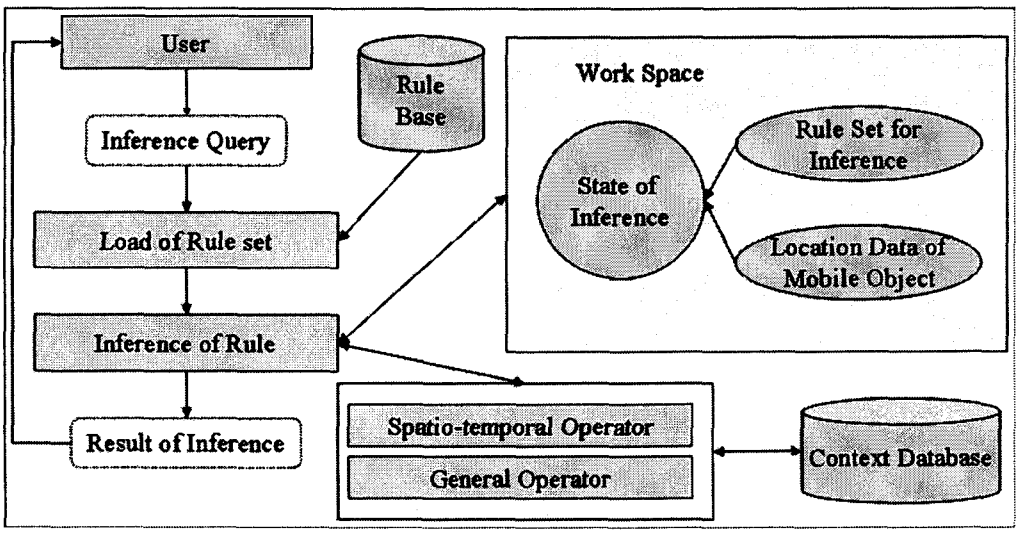


Figure 7. Process of Inference Engine

Inference engine operates work through mobile object operator. Inference query management algorithm using spatio-temporal operation and rule base is the same as in Figure 8.

Algorithm InferenceEngine

```
q : 'query' of string type
n : 'rule number' of integer type
f : 'fact set' of array type
Begin
    n ← get Rule Number from the query string q
    p ← extract fact operator list from knowledge base
    i ← total number of elements of array p
    For j=0, i-1
        process the fact operation of p[j]
        f[j] ← store the result of fact operation
    invoke rule_n() : rule execution method
    mapping the fact set f to fact variables of the rule_n()
    check condition part and process action part of the rule_n()
    r ← store result of the rule_n()
```

```
    invoke displayResult(r)
End
```

Figure 8. Inference Query Process Algorithm

Inference engine in Figure 7 and 8 operates domain inference function to infer location context of object and is processed as follows. First, it extracts spatio-temporal operator which is needed to operate inference query input and operates different operations according to query type. Second, it operates spatio-temporal operation extracted from first step and achieves spatio-temporal fact information. Spatio-temporal fact information is allocated as a value of condition when rules are practiced. Third, it extracts rules group needed for query type input from knowledge base. Rules group is a group of rules that is consisted of If(conditional part), Then(behavior part). Finally, it uses spatio-temporal fact information generated from second step and rules group achieved from third step, operates rules in working memory, and generates results.

4. Application of LBS

4.1 Scenario

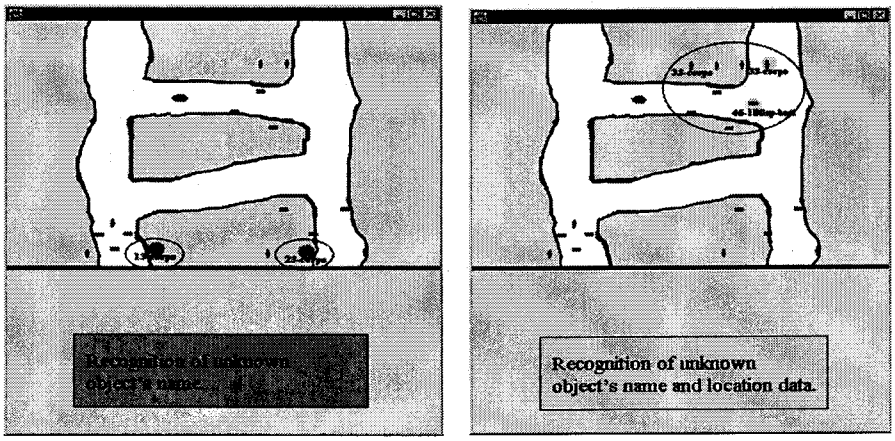
For application of context awareness of location based service, we operate inference which considers domain knowledge related to location information of mobile object that is used in the imitation battlefield and embody a module that infers location context of mobile object which is not stored in context database. For the experiment of location context inference function of embodied module, an imitation battlefield scenario has been composed. This scenario analyzes the present mobile context when enemy objects move, predicts the direction of their movement, and offers basic information to map out a correspondence strategy.

First, let's suppose 20 of battalions as a mobile object. Observed information from that is input on the initial object information attribute table. For those 20 battalions, there happens moving information for 10 days. Among the total 200 times of moving information, it supposes that 80%(160 times) of them are accurate and the rest(40 times) of them are inaccurate. Mobile object used in scenario is supposed to be a moving point. Mobile objects which can be expressed as a moving point are such as a unit, a tank, a missile, a human, etc., and information of moving change of object is expressed as a discrete model.

4.2 Implementation of Application Interface

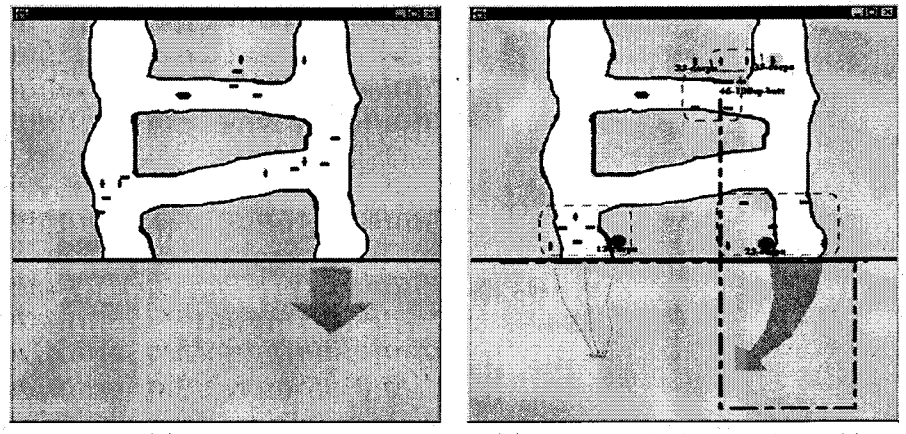
Application interface for location based context management is embodied by

Java and uses Oracle DBMS. In order to show an application of context management, we embodied inference query and embodied inference function can be confirmed through inference query. The result of query management of inference engine will be offered as a form of as follows.



(a) Inference of Unknown Object (b) Inference of Unidentified Object

Figure 9. Context Inference of Mobile Object



(a) In Moving (b) After Deploying an Assembly

Figure 10. Inference of Main Direction of Object

(a) in Figure 9 is the result screen that inferred unknown object during the valid time from May 9 and May 10. The content inside circle shows location information and name of mobile object inferred as unknown. (b) is the result screen that inferred unidentified object during the valid time from May 13 to May 14. If we suppose that there is moving information stored in the database till May 12, locations of all objects corresponded to the valid time after May 13 will be inferred as unidentified objects.

Figure 10 is the screen result that inferred which direction the object would go afterward from May 11 to May 12. (a) is the result of prediction of further predictable moving direction while the object moves and (b) is the prediction of moving direction after the object stopped moving and deployed an assembly in an area.

5. Conclusion

Location based service is what offers service added on the base of user's location and tries to offer context awareness LBS on demand according to sex, age, taste of individual. Therefore it first should consider personalized context awareness service on demand and could offer appropriate context awareness service to user by working on location information measured by sensor. Also location based service system must manage various location context data and make the best use of this data for application service in ubiquitous environment. But existing mobile object data management architecture did not consider process of context data. Therefore a new mobile data management framework is needed to process location context data.

This paper has proposed a location based context management framework for various location based application services. It has also described a scenario by applying proposed framework to a specific application and presented an application example by embodying application interface which is able to serve it to user. We ascertain that the proposed framework can supply application service by implement application interface to process a virtual scenario. In the future, there will be a research on context modeling technique of proposed framework and a research on extended architecture based on OSGI framework.

References

1. Ahn, D. I., Shin, C. S., and Joo, S. C. (2006). Context Information Supporting System Based on Indoor Location for Healthcare Home Service, *Proceedings of the Korean Information Science Society Conference*, Vo. 33, No. 1(D), 64-66.
2. Baek, J. H. and Yun, B. J. (2006). Estimation of User Activity States for Context-Aware Computing in Mobile Devices, *Journal of the Institute of Electronics Engineers of Korea SP*, 1229-6384, Vol. 43, No. 1, 67-74.
3. Erwig, M., Guting, R. H., Schneider, M. and Vazirgiannis, M. (1999). Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases, *GeoInformatica*, Vol. 3, No. 3, 269-296.

4. Farringdon, J., Moore, A. J., Tilbury, N., Chruch J., and Biemond, P. D. (1999). Wearable Sensor Badge & Sensor Jacket for Context Awareness, *Proceedings of the 3rd International Symposium on Wearable Computers*, 107-113.
5. Forlizzi, L., Guting, R. H., Nardelli, E. and Schneider, M. (2000). A Data Model and Data Structures for Moving Objects Databases, *Proceedings of the ACM SIGMOD Conference*, Dallas, Texas, 319-330.
6. Grumbach, S., Rigaux, P., Scholl, M., and Segoufin, L. (1999). The Design and Implementation of DEDALE.
7. Harter, A., Hoper, A., Steggles, P., Ward, A., and Webster., P. (1998). The Anatomy of a Context-aware Application, *Proceedings of 5th ACM MOBICOM Conf.* Seattle, WA.
8. Kim, D. H., Kim, J. S., Ahn, Y. A., and Ryu, K. H. (2002). Moving Objects Relational Model and Design for e-Logistics Applications, *Proceedings of International Conference*, ICITA2002.
9. Lema, J. A. C., Forlizzi, L., Guting, R. H., Nardelli, E., and Schneider, M. (2001). Algorithms for Moving Objects Databases, Fern University, Hagen, *Informatik-Report* 289.
10. Park, J. S. and Park, Y. T. (2004). The Architecture of Location-based Context Awareness System for Ubiquitous Environment, *Proceedings of the Korean Information Science Society Conference*, Vol. 31, No. 2, 172-174.
11. Priyantha, N., Chakraborty, A., and Balakrishnan., H. (2000). The Cricket Location-Support System, *6th ACM International Conference on Mobile Computing and Networking* (ACM MOBICOM), Boston, MA.
12. Randell C., and Muller, H. L. (2000). Context Awareness by Analyzing Accelerometer Data, *The Metadata International Symposium on Wearable Computers*, 175-176.
13. Wolfson, O. (2001). Moving Objects Databases: Issues and Possible Solutions, Keynote Address, *Proceedings of the Mobile Data Management.*
14. Wolfson, O., Sistla, P., Xu, B., Zhou, J., Chamberlain, S., Rishe, N., and Yesha, Y. (1999). Tracking Moving Objects Using Database Technology in DOMINO, *Proceedings of the 4th Workshop on Next Generation Information Technologies and Systems*, 112-119.

[received date : Oct. 2007, accepted date : Nov. 2007]