Design and implementation of a Moving Object Engine

Lee, Hyun Ah
Postal Technology Research Center, Electronics and Telecommunications Research Institute
161 Gajeong-dong Yuseong-gu, Deajeon 305-350, Korea
halee@etri.re.kr

Kim, Jin Suk
Postal Technology Research Center, Electronics and Telecommunications Research Institute
161 Gajeong-dong Yuseong-gu, Deajeon 305-350, Korea
kimjs@etri.re.kr

Abstract: Recently, the services using position information of moving objects is embossed. Theses services needs the moving objects databases to manage moving object data with efficiency. To build the moving object databases, we must develop the moving object engine to mange, store, and search the spatio temporal data of moving object. The moving object engine has to support query syntax to search data that suitable for user need like LBS, Telematics, ITS, vehicle management system. In this paper, we design and implement the moving object engine to support service with moving object data. The moving object engine is able to support system environment that users are able to get the moving object data easily even they don't know complex data structure.

Keywords: GIS, Moving object Databases

1. Introduction

Nowadays, it is very easy to trace vehicle position or cellular phone user from the progress of GPS/wireless communication technology, the miniaturization of wireless device, and the construction of high speed network. Hence, location based service with proper service is being studied lively, vehicle trace and management system to utilize and manage position information of vehicle is being developed to smooth traffic and logistics transportation.

The existing vehicle tracking system guides traffic volume and traffic jam based on vehicle position information received using beacon method or loop detector. However, the existing vehicle tracking system may not provide proper tracking if position data are not provided according to vehicle, location detector, and communication barrier. This is because the system focuses on providing the current vehicle position. Hence, the existing system has limitation which does not appropriately provide the query processing function and the position prediction function of future vehicle for position not to be provided. Hence, moving object engine for intelligent vehicle tracking system to store and manage historic position information of moving vehicle and to provide past and current position of moving vehicle was developed to solve this problem. The moving object engine implemented consists of moving object query processor, moving object operators, moving object index manager and moving object loader. However, this engine can not provide future position of moving object, and thus proper information is not provided by the engine when user makes future moving plan and situation decision

Hence, the following research and development are performed. (1) We design and implement MOQL to handle query about moving object data effectively; (2) we utilize moving object index to search moving object data successfully; (3) Uncertainty handling method for past and future position is used to estimate positions of moving object which are not stored in DB.

Moving object engine has the following features: (1) we use MOQL parser and query processing based multi-thread to handle moving object operators effectively; (2) past vehicle position among historical position information of vehicle is reconstructed to the moving object index; (3) we estimate past and future positions of vehicle which are not stored in DB.

Moving object data model is established in Section 2. We then design and implement moving object engine in section 3 and section 4. Finally, we make concluding remarks in section 5.

2. Data Model

We are able to classify moving object into moving point and moving region. The Vehicles is the moving points, and moving points is defined as followings;

[Definition 1](moving object) an object that changes its location as time goes by. A moving object MP has temporal, spatial, and general properties, $MP = (T_A.S_A, G_A)$. ■

[Definition 2](temporal property) The temporal property of MP is $T_A = (vt_s, vt_e)$. vt_s represents starting time, vt_e represents ending time. vt_s and vt_e are elements of the set of valid time, S_{vt}. The valid time represents real time, $S_{vt} = t_0, t_1, ..., t_k, ..., t_{now}$, and order of each element is $t_0 < t_1 < ... < t_k < ... < t_{now}$ such that $t_0 = t_0 + 1, t_0 = t_0 + k, k \leq 0$ where the elements are integer. t_{now} is a constant that represents current time. ■

The domains of valid time are linear time, discrete time, and absolute time. Each moving object data base has its own valid time granularity. In definition 1, $T_A(mp_{i_k})$ represents k[th] temporal property, $S_A(mp_{i_k})$ represents k[th] spatial property, and $G_A(mp_{i_k})$ represents k[th] general property.

[Definition 3](spatial property) The spatial property of MP is $S_A = (x, y)$, $x, y \in R$, where R is the set of real numbers. ■

[Definition 4](moving object databases) The moving object database consists of the following sets, $S_{MP} = MP_0, MP_1, \ldots, MP_k$. The history set of each S_{MP} is $H_{MP} = H_{MP_0}, H_{MP_1}, \ldots, H_{MP_k}$ which is stored in the moving object database. The history of each MP_i in S_{MP} is $H_{MP_i} = mp_{i_0}, mp_{i_1}, \ldots, mp_{i_k}$. ■

In definition 4, mp_{i_k} implies k[th] history information of MP_i, MOQL (Moving Object Query Language) is defined based on following data model.

3. Moving Object Engine

1) Conceptual Modeling

Moving object engine is designed to provide the following functionalities for efficient search of vehicle location.

Firstly, it provides moving object index to quickly respond to user's query so that the historical location that is stored in the database can be managed efficiently.

Secondly, it predicts location of the time user's query is requested, when there is no certain location information respect to user's query for past or near future timestamp stored in the database.

Thirdly, it provides MOQL parser and its processing module. MOQL is a query language based on SQL which is independent to various vehicle tracking systems nor its various schema structure [5] .

Fourthly, in order to rapidly process each MOQL query we support multi-thread. It also supports multi-user environment.

Fifthly, it supports spatio-temporal operator to discover object to object and object to region relational analysis.

Sixthly, it supports moving object engine API so that other vehicle tracking system to easily use moving object engine.

2) System Design

Figure 1 shows the moving object engine that consists of MO Query Processor, MO core, MO Data Loader.

A. MO Query Processor

MOQL based moving object query processor supports query syntax respect to user's query and increases system interoperability. Multiple queries can be processed efficiently by supporting thread based query processor.

With the following type of moving object query language from the user interface, the syntax is being checked, and the meaning of each token is analyzed.

Example) "Find the trajectory of vehicle number SL81BA3578 that departs from (243461,349089) and arrives at (244032,350806) in the time interval from 2004 March 5 1:24 pm to now."

MOSQL) SELECT TRAJECTORY FROM VEHICLEHISTORY WHERE ID = 'SL81BA3578' AND VALID FROM '200403051324' TO NOW AND SECTION FROM (243461,349089) TO (244032,350806);

This example MOSQL is able to analysis following;

TRAJECTORY	- query type
'SL81BA3578'	- ID
'200403051324'	- temporal predicate
NOW	- temporal predicate
SECTION	- spatio-temporal operator
(243461,349089)	- spatial predicate
(244032,350806)	- spatial predicate

B. MO Operator

MO Operator does not only support basic operation[1] to find spatio-temporal location and trajectory but also operator with spatio-temporal predicate, BUFFER and BOUNDARY.

Specially, to resolve uncertainty operations for predicting past location and near future location, we have applied interpolation (1), (2)[2] and trigonometry (3), (4) respectively.

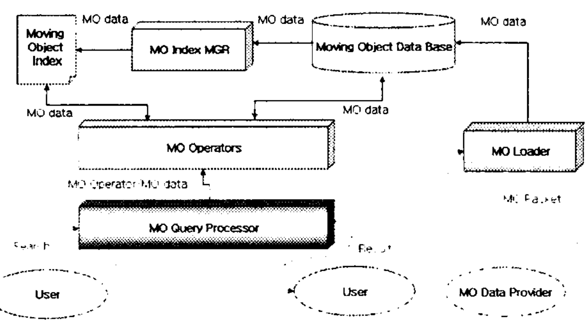


Fig. 1. Moving Object Engine

$$x_{t_p} = \frac{x_{t_{i+1}} - x_{t_i}}{t_{i+1} - t_i}(t_p - t_{i+1}) + x_{t_{i+1}} \quad (1)$$

$$y_{t_p} = \frac{y_{t_{i+1}} - y_{t_i}}{t_{i+1} - t_i}(t_p - t_{i+1}) + y_{t_{i+1}} \quad (2)$$

$$x_{t_p} = x_{t_i} + (\frac{t_{now}}{s_{t_{now}}} \times \cos(90 - angle_{t_{now}})) \quad (3)$$

$$y_{t_p} = y_{t_i} + (\frac{t_{now}}{s_{t_{now}}} \times \cos(90 - angle_{t_{now}})) \quad (4)$$

t_p is time for location prediction, and x_{t_p} , y_{t_p} are predicted location. $S_{t_{now}}$ is current velocity, and $angle_{t_{now}}$ is current direction.

C. MO Index Manager

For rapid moving object data search in the moving object engine, it constructs TB-tree[3,4] based moving object index and queries the index.

D. MO Data Loader

Moving object data is stored in the moving object database depending on the user defined interval. At this time, the moving object data collector supports GPS, Beacon. Newly collected data is stored in VehicleTemp table and historical data is stored in VehicleHistory table.

4. Implementations

Moving object engine is developed under JDK 1.3.1, Oracle 9i, Windows XP, and there is no further installation requirement for system execution. The moving object engine does not provide separate GUI so a web presentation for moving object query and its result has been developed. The following shows the initial execution screen shot.

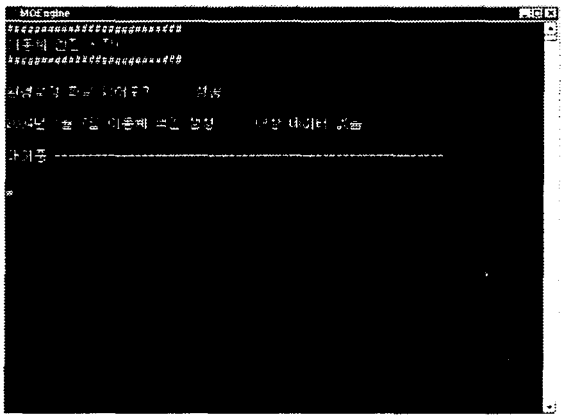


Fig. 2. the initial screen of the moving object engine

Web presentation includes user query input functionality. Figure 3 shows query input screen shot.

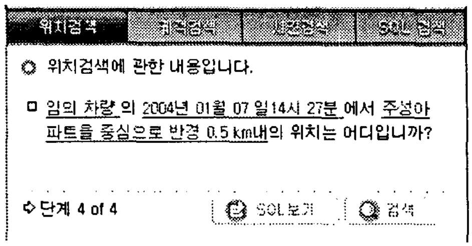


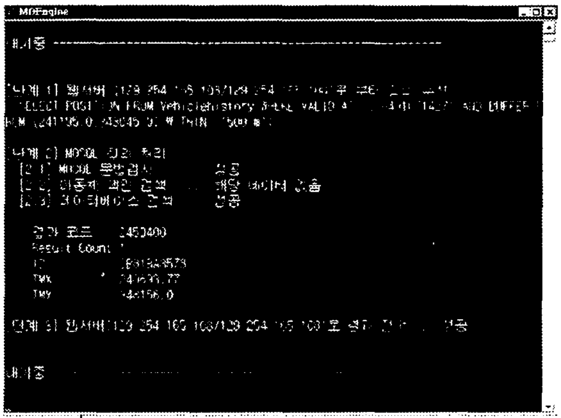
Fig. 3. the screen of user query insertion

The query in Figure 3 can be expressed as the following MOQL and this is sent to the moving object engine.

SELECT POSITION FROM Vehiclehistory WHERE VALID AT '200401071427' AND BUFFER FROM (241195,348045) WITHIN '500 m';

Figure 4 shows the execution state after moving object engine's MOQL analysis and operation.

Fig. 4. Execution screen of the moving object engine



Execution result is constructed with Stream and is provided with we presentation. Web presentation analyses Stream and displays the result of the analysis.

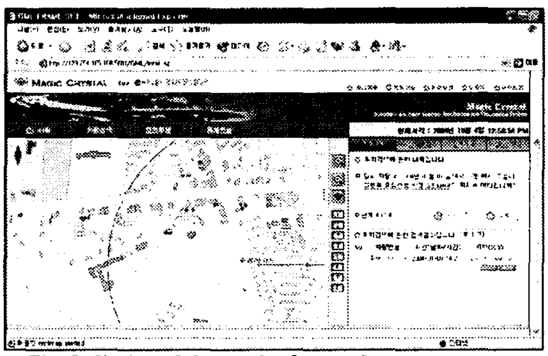


Fig. 5. display of the result of example

5. Conclusions

Existing vehicle tracking systems only provide current vehicle location and in case of malfunctioning of location detector we may not get the right location information and lose its trajectory. In order to solve such problems, we have developed a moving object engine that effectively manages vehicle location information.

The moving object engine has functionalities of existing vehicle tracking system as well as providing location information of near future. Thus this can be applied to intelligent logistics management system, location based service, intelligent traffic system, etc. and process efficient spatio-temporal queries.

References

[1] Timos K. Sellis, 1999. CHOROCHRONOS - Research on Spatio-temporal Database System, *Integrated Spatial Database*, p308-316

[2] Ouri Wolfson, A. Prasad Sistla, Bo Xu, Jutai Zhou, Sam Chamberlain, 1999. DOMINO: Databases fOr MovINg Objects tracking, *SIGMOD Conference*, p547-549

[3] Mohamed F. Mokbel, Thanaa M. Ghanem, Walid G. Aref, 2003. Spatio-Temporal Access Methods. *IEEE Data Eng. Bull.* 26(2): 40-49

[4] Dieter Pfoser, Christian S. Jensen, Yannis Theodoridis, 2000 Novel Approaches in Query Processing for Moving Object Trajectories. *VLDB 2000*: 395-406

[5] Lee, Hyun Ah, Lee, Hye Jin, Kin, Dong Ho, Kim, Jin Suk, 2003. Moving Object Query Language Design for Moving Object Management System *한국정보과학회 2003 추계학술발표대회* 제30권 제2호(II)