Journal of Korean
Data & Information Science Society
2004, Vol. 15, No. 3, pp. 663~675

Moving Object Management System for Battlefield Simulation

Yoon Ae Ahn[1]

Abstract

A battlefield simulation is the evaluation and analysis of the battlefield area, based on the data for terrain, climate, unit's maneuver and tactics basically required in battlefield simulation. Because it is difficult for the military authorities to collect all of the information perfectly for the reason of communication technology, jamming, and tactics, the military authorities need the future moving status for the target units by using acquired moving information. Therefore, we propose a moving object management system that concurrently provides domain reasoning function for the battlefield simulation. In order to implement the proposed system, we show the data modeling of the moving object for the battlefield simulation, and propose an inference engine using domain rule base and spatiotemporal operation. Also, we analyze the query response rate by inference function to verify domain reasoning of the implemented system.

Keywords : Battlefield Simulation, Inference Engine, Moving Objects, Spatiotemporal Operators

1. Introduction

For the battlefield simulation, it is required to get correct information about the location identification and moving status of the enemy units. However, since collecting all the information perfectly is actually impossible, we need to predict and analyze the future moving status for the target units by using collected information and related knowledge. Especially, certain objects of the battlefield simulation like units, tanks, and vehicles have some characteristics of moving objects which change their position and shape over time, and are included in a

---

1) Professor, Dept. of Computer Science, Chongju National College of Science & Technology, Chungbuk, 368-701, Korea
E-mail : yeahn@cjnc.ac.kr

kind of spatiotemporal data having temporal and spatial attributes altogether.(Erwig et al.(1998), Erwig et al.(1999), Guting et al.(2000), Forlizzi et al.(2000))

Therefore, the battlefield simulation prototype should have a reasoning function of unit identification and location as well as a managing function of spatiotemporal moving objects, considering limitation of time change and facing enemy.

This paper proposes a moving object management system that provides the domain inference function for the battlefield simulation. The proposed system is composed of spatiotemporal database, moving object processor, inference engine, rule base, and GIS tool. The database is stored with location information for moving the enemy units. Moving objects processor does computations using the location information stored in the database. Especially, it also predicts operations of the moving objects location. Moreover, inference engine finds out unknown units, unidentified units, and main strike direction for the battlefield analysis, using domain knowledge stored in the rule base.

The proposed system can manage all the historical information of the moving objects for a battlefield simulation, because it builds a database in the spatiotemporal database format in order to manage temporal and spatial attributes of the moving objects. Additionally, it shows a domain inference engine utilizing domain rule base and spatiotemporal operation for reasoning a specific application using the moving objects.

This paper proceeds as following. Chapter 2 reviews the concept of the moving objects. Chapter 3 explains the configuration and function of the proposed moving object management system for the battlefield simulation. Chapter 4 describes the characteristics of the implemented system. Finally, Chapter 5 draws a conclusion for this study.

2. Related Works

Moving objects belong to spatiotemporal data, having the characteristic that the spatial object changes its position and shape with move. Moving objects have two basic types, which are moving points and moving regions.(Erwig et al.(1999), Forlizzi et al.(2000)) Moving 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. Moving 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 moving regions, and express the value of the spatial region at a specific time.

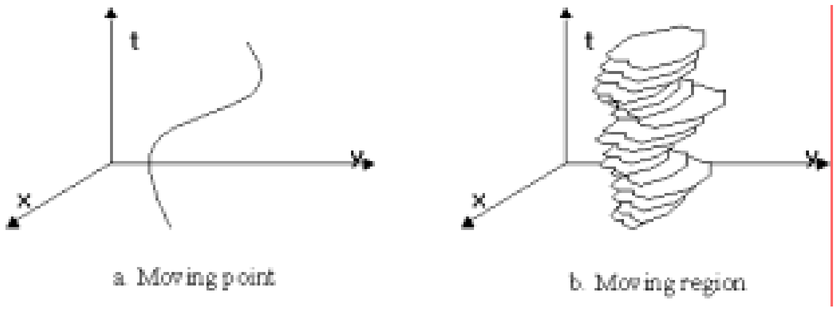

Figure 1. Moving Objects

In Figure 1, the moving 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 moving object.

There is a representative research for the application system of moving objects, which is DOMINO prototype.(Sistla et al.(1997), Wolfson et al.(1997), Wolfson et al.(1998), Wolfson et al.(1999)) It provides the deducting function of near future location using present location, speed, and direction information for the moving objects. Moreover, it proposed FTL(Future Temporal Logic) language for future queries. However, it has a critical drawback that cannot manage the full path of the moving objects including a past time point, since it does not store any historical information. Also, it cannot give a prediction method for the future location using the stored past information and historical data, because it mainly focuses on the query processing related to the future location of the moving objects. Therefore, for these reasons, this system is not appropriate to be applied to the battlefield simulation prototype that requires inferring the future moving pattern using past historical data.

CHOROCHRONOS research consortium in Europe conducted a study on the moving objects as a specific area of the spatiotemporal database. Particularly, researches about data modeling, indexing, and uncertainty processing have been worked.(Pfoser et al.(1999), Pfoser et al.(2000)) In addition, they proposed an application scenario for GPS-based transportation management system and multimedia system. However, it is also inadequate to be applied to the application system having a reasoning function, since they did not propose an application model and implementation using the moving object database yet.

3. Moving Object Management System for Battlefield Simulation

This section describes a moving object management system, which is able to support the moving objects for the battlefield simulation. This paper describes the domain inference, which only considers knowledge related with location information of the moving objects such as unit and tank used in the battlefield simulation, and designs a system that predicts the past and near future locations of the moving objects that were not in database. This system consists of user interface, database, moving objects processor, inference engine, rule base, and GIS tool, and shows the structure as shown in Figure 2.

The user interface in Figure 2 provides GUI environment. The database stores and manages temporal and spatial attributes of the moving objects, and uses the existing commercial DBMS. The moving objects processor executes and searches the operations of the moving objects stored in the database, and predicts the past and future locations of the moving objects that were not in the database by a location prediction operation.

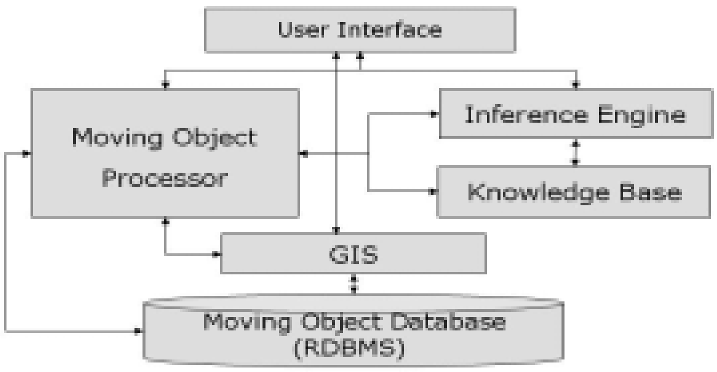

Figure 2. Moving Object Management System

In Figure 2, the inference engine performs a reasoning function for the battlefield simulation using the location data stored in database, domain knowledge in the rule base, and spatiotemporal facts processed by the moving objects processor. The rule base stores domain rules and spatiotemporal facts required in executing the inference engine. The GIS tool manages the spatial information of the moving object data, spatial indexing information, and map data.

3.1 Database for Moving Objects

We assume that the moving objects for the battlefield simulation are moving points only. The objects that can be expressed as moving points are units, tanks, missiles, and people. Also, we present the location change status of the moving object denoted by a discrete model and applied time domain. In discrete models, the route of a moving object is depicted with straight lines. This means that it approximates the location change of the object linearly. A database for the battlefield simulation is composed of initial unit data relation, moving data relation, and unknown unit relation.

In the initial unit data relation, there are general attribute, location coordinate, and valid time (actually observed date) of the enemy units that were first observed. The initial data of the unit is stored in GIS for spatial indexing. This paper uses Geowin for the GIS tool.(Geostechnology (2002))

Table 1. Initial unit data relation

Id	Code	Name	VTs	VTe	X_vts	X_vte	Y_vts	Y_vte	Type	Reg
Key	Code of object	Name of object	Valid time (start)	Valid time (end)	X coord At VTs	X coord at VTe	Y coord at VTs	Y coord at VTe	Unit type	Regiment info
int	int	string	string	string	float	float	float	float	int	string

Table 1 shows an initial unit data relation which consists of object ID, Code, Name, VTs, VTe, X_vts, X_vte, Y_vts, Y_vte, Type, and Reg. The moving data relation is the history relation that records all the moving data of the units stored in the initial unit data relation, and stores the location coordinate and moving date of units.

Table 2. Moving data relation

Code	VTs	VTe	X_vts	X_vte	Y_vts	Y_vte
Code of object	Valid time (start)	Valid time (end)	X coord at VTs	X coord at VTe	Y coord at VTs	Y coord at VTe
int	String	string	Float	float	float	float

The moving data relation consists of moving object Code, VTs, VTe, X_vts, X_vte, Y_vts, and Y_vte. The unknown unit relation does not have any unit identification data, but stores only the unit location coordinate and time value.

Table 3. Unknown unit relation

VTs	VTe	X_vts	X_vte	Y_vts	Y_vte
Valid time (start)	Valid time (end)	X coord at VTs	X coord at VTe	Y coord at VTs	Y coord at VTe
string	string	Float	float	float	float

The unknown unit relation cannot get unit identification. Therefore, all the items in the moving data relation except Code are the elements of this relation.

3.2 Moving Object Processor

Moving object processor is the part of the processing operation using moving objects, which are stored in a database. Operation types are general operations of the moving object and prediction operation for past and future locations which are not stored in the database. We define moving object operators for the battlefield simulation. We do not need a search function in addition, since the commercial DBMS provides it for the general attribute of a moving object. Table 4 shows moving object operations based on operators given on the existing study of moving objects.(Guting et al.(2000))

Table 4. General operations of a moving object

Operator	Input	Output	Equation
MDistance	OA(x,y), OB(x,y) <VTs, VTe>	Distance between OA & OB during <VTs, VTe>	$\sqrt{(Oy(B) - Oy(A))^2 + (Ox(B) - Ox(A))^2}$
Trajectory	Oid_OA, <VTs, VTe>	Moving line of OA during <VTs, VTe>	$Y - Oy(VTe) = \frac{Oy(VTe) - Oy(VTs)}{Ox(VTe) - Ox(VTs)}(X - Ox(VTe))$
Length	Oid_OA	Total moving distance	$\sum_{i=1}^{n-1} Dist(Trajectory(O_{Ai}))$

Also, we define some operations for the battlefield simulation, which is called "spatiotemporal fact generation operator", besides the general operations of the moving object in Table 4. These operations generate some facts required to execute rules of the inference engine.

Table 5. Spatiotemporal fact generation operations

T_Mdirection(P1,P2)	Finds moved direction from P1(x1,y1) to P2(x2,y2)
T_Near(A,t1,t2)	Finds all the objects having Near relation with object A during valid time <t1,t2>
T_ Far(A,t1,t2)	Finds all the objects having Far relation with object A during valid time <t1,t2>
T_VeryFar(A,t1,t2)	Finds all the objects having VeryFar relation with object A during <t1,t2>

T_Near, T_Far, and T_VeryFar in Table 5 have spatial characteristic that cannot be expressed in one mathematical value, since people have different distance senses from each other.(Lee et al.(1997), Sharma et al.(1994)) Therefore, we actually assign quantitative values for them in advance.

3.3 Domain Rule Base

The rule base consists of a domain rule set and a fact set. In the domain rule set, domain rules are stored and used for inferring an unknown unit, unidentified unit, and main strike direction. Rules are arranged with conditional (If～) and active parts (Then～).(Dutta (1994))

```
Rule_100() {
  if (element_of(_Mechanized_unit, _Motion_Table)
      && exist_on(_Mechanized_unit, _N_L)
      && opposite_direction(_N_L, _S_L))  {
           is_determined(Main_strike, _N_L);
           is_determined(Sub_strike, _S_L);
           display(Main_strike);
           display(Sub_strike);
  }
}
```

Figure 3. An example of rule in JAVA

Figure 3 shows an example of the rule coded as "if statement" in Java. It is a rule that determines the future main strike direction of the unit. Rule_100 is a rule number, and Mechanized_unit, Motion_Table, _N_L, _S_L, Main_strike, and Sub_strike are all variables. In addition, element_of, exist_on, opposite_direction, is_determined, and display are rule execution methods.

3.4 Execution Model of Inference Engine

The inference engine performs a domain inference function for the battlefield simulation, and consists of rule classifier, spatiotemporal fact generator, and rule executor.

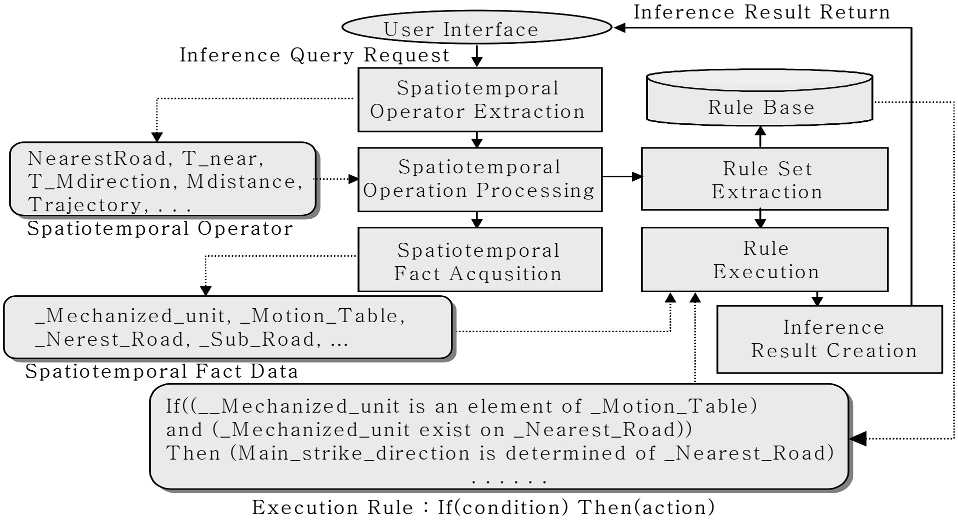

Figure 4. Inference model using domain rule

The execution model of an inference engine is shown in Figure 4 and it is executed as the following steps.

- Step 1: *Extraction of spatiotemporal operator*
  - Extracts spatiotemporal operator needed to process inputted inference query.
  - Executes different spatiotemporal operation with each other according to query type.
  - Ex.) Mechanized_unit, _Motion_table, _Nearest_Road, _Sub_Road, etc.

- Step 2: *Processing of spatiotemporal operation*
  - Acquires spatiotemporal fact data after processing spatiotemporal operation extracted from Step 1.
  - Assigns spatiotemporal fact data to value of conditional part at rule execution.
  - Ex.) Mechanized_unit, _Motion_table, _Nearest_Road, _Sub_Road, etc.

- Step 3: *Extraction of rule set*
  - Extracts rule set needed to query type inputted from rule base.
  - Rule set is the gathering of rules that are arranged with If(conditional part) and Then (active part).

- Step 4: *Execution of rule*
  - Uses spatiotemporal fact data generated from Step 2 and rule set acquired from Step 3.
  - Creates result after rule execution in working memory.

3.5 GIS Tool

Here, the system uses the Geowin spatial management system for the GIS tool. The data storage of Geowin consists of three parts. They are attribute data table(_ADT), spatial indexing data table(_SIT), and spatial data table(_SDT). (Geostechnology (2002)) _ADT stores non-spatial attribute data of the object, _SIT stores indexing data, and _SDT stores spatial data. The table where a user can create and change an attributes arbitrarily is attribute data table. It puts an additional name "ADT" to its table name. The spatial indexing data table of Geowin creates and stores R*-Tree index for the spatial objects.

4. Implementation

The proposed moving object management system for battlefield simulation has been realized with a client/server structure. A client was implemented with JDK1.3 on Window 2000 and a server was implemented with Oracle 8 and the Geowin space management tool on UNIX. The functional scope of the system implementation is divided into an inference query and a general query. There are five inference queries for the battlefield simulation, which are the functions of an unknown unit inference, unidentified unit inference, main strike direction inference (on move, after assembled disposition), moving location inference, and moving time inference. Meanwhile, general queries are tracing the moving route, viewing the unit data by date, searching the close unit, and searching the buffer, and provide an input function of the moving data by the user interface. We show the implementation results and query response rate.

4.1 Applied Scenario and Implementation Result

We composed a battlefield simulation scenario to experiment with the moving object operators of the implemented system. This scenario makes it possible to establish the best corresponding strategy by analyzing the current situation and by predicting the attack direction when an army division is moving. First, we input the observed data on May 1, 2000 to the initial unit data table, supposing that twenty battalions of one division are the moving objects. Next, the moving data during ten days for twenty battalions were generated. Now we suppose that accurate moving data are 80 percent (160 times) of the total and the remaining 20 percent (40 times) is inaccurate. Under this scenario, the results of several query samples are shown as follows.

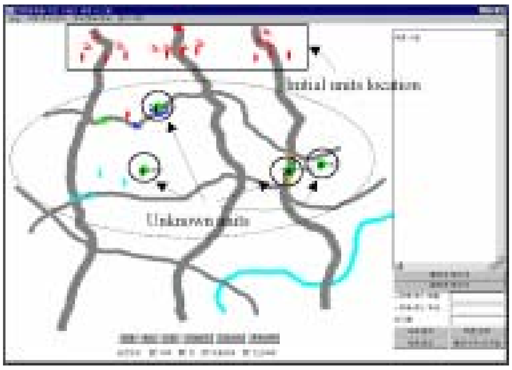

Figure 5. Unknown unit inference

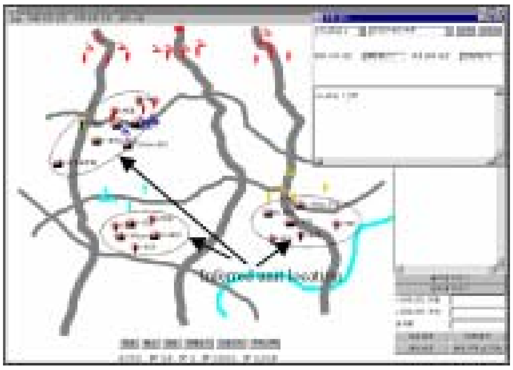

Figure 6. Unidentified unit inference

Figure 5 is the result screen from reasoning the unknown units corresponding to the valid time between 2000/05/09 and 2000/05/10. Inside of the dotted ellipse represents the unit location information from 2000/05/09 to 2000/05/10. The small circle with the solid line inside of the ellipse represents the location information and the name of the unit inferred as an unknown unit.

Figure 6 is the result screen from reasoning the unidentified units corresponding to the valid time between 2000/05/13 and 2000/05/14. Since the actual data stored in the moving object database only exist until 2000/05/12, locations of all objects corresponding to the valid time after 2000/05/13 are inferred as unidentified units.

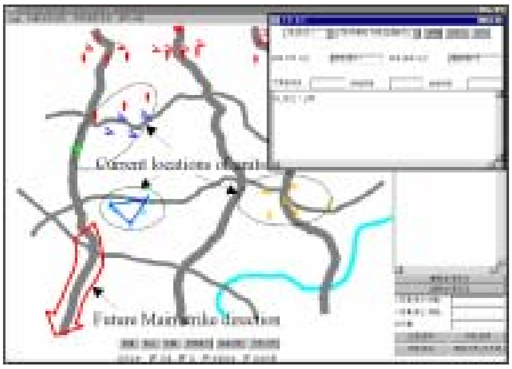

Figure 7. Main strike direction inference

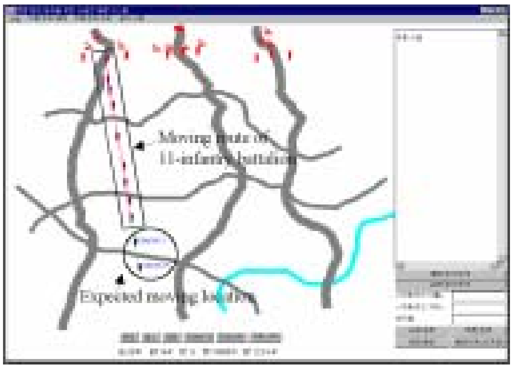

Figure 8. Moving location inference

Figure 7 is the result from reasoning the main strike direction after the assembled disposition from 2000/05/11 to 2000/05/12. Inside of the circle represents the locations of objects corresponding to the valid time input for the query, and the arrowed part is the result from reasoning the main strike direction to which units will move after 2000/05/13.

Figure 8 is the result from reasoning the moving location which 11th-infantry battalion will move from 2000/05/13 to 2000/05/15 in the future. Inside of the tetragon is the route where 11th-infantry battalion has moved from 2000/05/01 to 2000/05/11, and inside of the circle shows the expected moving routes from 2000/05/13 to 2000/05/15.

4.2 Query Response Rate by Inference Engine

In order to analyze and consider the characteristics of the proposed system, we analyzed query response rate by inference engine and by date. The query response rate by date for the system was considered as a factor of the characteristic evaluation. It is represented by the percentage for the number of system response over the number of total queries by date. We used five query types for this experiment and obtained the result shown in Figure 9 after the execution of 500 queries. They were made by creating 50 queries per date for ten days with respect to five reasoning functions such as the unknown unit inference (Query 1), unidentified unit inference (Query 2), main strike direction inference (Query 3), unit's moving location inference (Query 4), and unit's moving time inference (Query 5). Queries were built by assigning ten different valid times by date for each reasoning function.

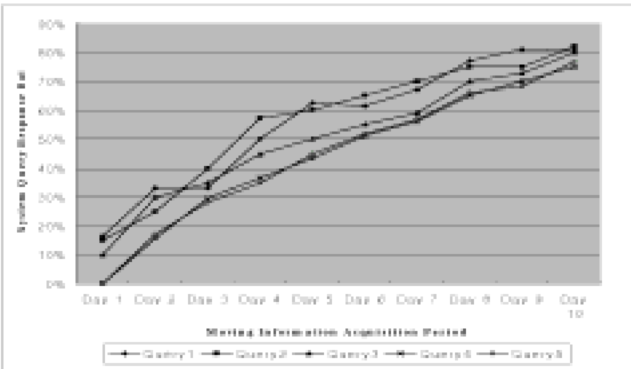

Figure 9. Query response rate

Approximately, as the acquisition period of the system becomes longer, the query response rate is inclined to increase. Especially the response rates of Queries 1, 2 and 3 show relatively higher distributions than those of Queries 4 and 5. The reason is that Queries 1, 2, and 3 are the inference function limited to a specific application, which is called the battlefield simulation. Therefore, since the system utilizes a domain rule regarding Queries 1, 2, and 3, it can return the result even though there are no historical data. On the contrary, Queries 4 and 5 are related to the moving location and time of an object. Thus, the system cannot return the result if there are no historical data.

5. Conclusion

The battlefield simulation prototype requires a proper corresponding strategy of friendly forces according to the moving status of enemy forces. For this purpose, prediction of the moving status and location for enemy units is necessary. This paper implemented a domain dependant inference engine and a moving object management system such as units and tanks applied to the battlefield simulation. Moreover, we found that the implemented system could be applicable to develop similar applications like navigation system, war game model, and close combat simulation, through the simulated experiment using the scenario.

The implemented system manages all the historical data for the moving objects by building a spatiotemporal database, in order to control temporal and spatial attributes of the moving objects. In addition, the system proposed the domain rule base for reasoning of the specific application using the moving objects, and the domain inference engine model using the spatiotemporal operation.

As the result of the analyzing query response rate of the implemented system, the accumulated amount of the moving data influences the system query response rate, and decision of the moving data acquisition interval becomes a key variable.

Moving objects always have uncertainty due to either sampling error or measurement error, because their attributes, locations, and shapes are changing over time. Generally speaking, although the data or knowledge used in the inference engine are wholly correct, the reasoning system cannot fully guarantee the correctness to the inferred result. Therefore, it is in progress that the study about various processing methods that can minimize the uncertainty of the spatiotemporal data applied to moving object reasoning, and the study that raises the correctness of result by applying to the actual reasoning process.

References

1. Erwig, M., Guting, R.H., Schneider, M. and Vazirgiannis, M. (1998). Abstract and Discrete Modeling of Spatio-Temporal Data Types, Chorochronos Technical Report, CH-98-14.
2. 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.
3. Guting, R.H., Bohlen, M., Erwig, M., Jensen, C.S., Lorentzos, N., Schneider, M. and Vazirgiannis, M. (2000). A Foundation for Representing and Querying Moving Objects, ACM Transactions on Database Systems, Vol. 25, No. 1, 1-42.
4. Forlizzi, L., Guting, R.H., Nardelli, E. and Schneider, M. (2000). A Data

Model and Data Structures for Moving Objects Databases, Proc. of the ACM SIGMOD Conference, Dallas, Texas, 319-330.
5. Sistla, P., Wolfson, O., Chamberlain, S. and Dao, S. (1997). Modeling and Querying Moving Objects, Proc. of the 13th International Conference on Data Engineering (ICDE13), Birmingham, UK.
6. Wolfson, O., Chamberlain, S., Dao, S. and Jiang, L. (1997). Location Management in Moving Objects Databases, Proc. of the 2nd International Workshop on Satellite-Based Information Services (WOSBIS'97), Budapest, Hungary.
7. Wolfson, O., Xu, B., Chamberlain, S. and Jiang, L. (1998). Moving Objects Databases: Issues and Solutions, Proc. of the 10th International Conference on Scientific and Statistical Database Management (SSDBM98), Capri, Italy, 111-122.
8. Wolfson, O., Sistla, P., Xu, B., Zhou, J., Chamberlain, S., Rishe, N. and Yesha, Y. (1999). Tracking Moving Objects Using Database Technology in DOMINO, Proc. of the 5th Workshop on Next Generation Information Technologies and Systems, Israel, 112-119.
9. Pfoser, D. and Jensen, C.S. (1999). Capturing the Uncertainty of Moving Object Representations, Proc. of the Advances in Spatial Databases, 6th International Symposium, SSD'99, Hong Kong, China.
10. Pfoser, D. and Tryfona, N. (2000). Fuzziness and Uncertainty in Spatiotemporal Applications, Chorochronos Technical Report, CH-00-4.
11. Geostechnology. (2002). Geowin System, http://www.geowin.net.
12. Lee, J.H., Lee, M.S., Woo, Y.W., Park, C.S. and Lim, J.H. (1997). An Event-based Temporal Reasoning Method, Journal of the Korea Institute of Telematics and Electronics, Vol. 34, No. 5, 337-346.
13. Sharma, J., Flewelling, D. and Egenhofer, M. (1994). A Qualitative Spatial Reasoner, Proc. of the 6th International Symposium on Spatial Data Handling, Edinburgh, Scotland, 665-681.
14. Dutta, S. (1994). Knowledge Processing & Applied Artificial Intelligence, Butterworth Heineman Publisher, 31-54.
15. Oh, I.B., Ahn, Y.A., Lee, E.J., Ryu, K.H. and Kim, H.G. (2002). Prediction of Uncertain Moving Object Location, Proc. of International Conference on East-Asian Language Processing and Internet Information Technology, EALPIIT'02, 51-58.
16. Huh, M.H., Ahn, Y.A. and Ryu, K.H. (2002). Moving Object Location Change Function using Cubic Spline Interpolation, Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics, SCI'02, Vol. V, Computer Science I, 336-341.

[received date : May. 2004, accepted date : Aug. 2004]