Application of Moving Objects and Spatiotemporal Reasoning

Keun Ho Ryu and Yun Ae Ahn

May 29, 2001

TR-58

A TIMECENTER Technical Report

Title	Application of Moving Objects and Spatiotemporal Reasoning
Author(s)	Keun Ho Ryu and Yun Ae Ahn
Publication History	May 2001. A TimeCenter Technical Report

Abstract

In order to predict future variations of moving objects which general attributes, locations, and regions of spatial objects are changed over time, spatiotemporal data, domain knowledge, and spatiotemporal operations are required to process together with temporal and spatial attributes of data. However, conventional researches on temporal and spatial reasoning cannot be applied directly to the inference using moving objects, because they have been studied separately on temporal or spatial attribute of data. Therefore, in this paper, we not only define spatial objects in time domain but also propose a new type of moving objects and spatiotemporal reasoning model that has the capability of operation and inference for moving objects. The proposed model is made up of spatiotemporal database, GIS tool, and inference engine for application of spatiotemporal reasoning using moving objects and they execute operations and inferences for moving objects. Finally, to show the applicability of the proposed model, a proper domain is established for the battlefield analysis system to support commander's decision making in the army operational situation and it is experimented with this domain.

1 Introduction

Various kinds of application systems using databases simply provide the function of searching the fact that databases are stored, however the active progress of research adding knowledgebase to database requires the prediction function for the future that stored data value produces different result according to time change. This induces to do researches about several application areas which offer reasoning function as well as database. Temporal reasoning and spatial reasoning dominates research areas related to the reasoning using database. First, temporal reasoning performs inference to the changing situation over time using time data. Typical examples of temporal reasoning include metabolic diagnosis system, stock investment expert system, and simulator for predicting the result of professional baseball game. And, spatial reasoning is an inference technique to figure out several spatial problems based on the relationship between spatial objects. It has been mainly applied to site selection for commercial or industrial facility, route search, ambulance dispatch, setting up and plan of delivery truck routes, arrangement and allocation of store, and geometric inference of stratum.

By the way, all the moving things in the real world are spatiotemporal data, having time and space attributes simultaneously and having characteristic which space attribute changes over time. These data are called spatiotemporal data, especially the data that is moving with changes of its location or shape according to time on space are called moving object[Erwi99, Guti2000]. Related application areas using moving object include Navigational Systems, War Game Models, Close Combat Simulations, and Battlefield Information Analysis System. These areas require reasoning functions of its system. The reasoning of moving object must perform spatiotemporal reasoning that supports time attribute and space attribute altogether. But, existing temporal or spatial reasoning is separately proceeded, so spatiotemporal reasoning is not properly performed. Because of these reasons, developing spatiotemporal reasoning application system that can be used for actual spatiotemporal reasoning system, as well as the theoretical study about spatiotemporal reasoning to overcome the limits of temporal reasoning and spatial reasoning, is required.

Therefore, this study will proceed with the sequence of following contents so as to make the study about spatiotemporal reasoning progress concretely and realistically, with objective that develops spatiotemporal reasoning system for the battlefield analysis which applies spatiotemporal reasoning function to the battlefield analysis.

First, the study related to temporal and spatial reasoning has proceeded. Temporal reasoning is to infer several features which temporal data have, and spatial reasoning is an inference technique to figure out several spatial problems based on the relationship between spatial objects. Based on this related work, it is defined that spatiotemporal reasoning performs temporal reasoning and spatial reasoning together to the spatiotemporal data changing over time flow, and we establish application area by adopting spatiotemporal reasoning.

Second, the study about spatiotemporal data which will be used for spatiotemporal reasoning carries out. In particular, we limit spatiotemporal data to moving object and study about spatiotemporal database which is essential for managing spatiotemporal moving object. Moving object represents the spatial object that is characterized by change of its location or shape according to time.

Third, we offer general function and reasoning type for spatiotemporal reasoning system, and show spatiotemporal system model based on these function. Spatiotemporal reasoning system has functions for location inference of moving object, moving time inference, and inference to solve the problems of application domain, and must be able to perform these functions.

Finally, we develope a spatiotemporal reasoning system for battlefield analysis for the purpose of proper use of proposed system, and experimented the functions for implemented system by forming virtual scenario.

After all, spatiotemporal reasoning is to perform several inference functions for spatiotemporal data changing over time flow. And the system which manages spatiotemporal data and supports spatiotemporal reasoning function is called spatiotemporal reasoning system.

The spatiotemporal reasoning system must provide various reasoning functions for problem solving in the application area. It is the research area that can be applied to greatly wide areas, since the applications using spatiotemporal reasoning are entirely associated with applications of both spatiotemporal database and expert systems. In particular, it is expected that spatiotemporal reasoning would be very usefully employed to the application areas like navigational systems, war game models, close combat simulations, and battlefield information analysis system.

To show processed work effectively, in chapter 2, basic concepts and theories related to temporal and spatial reasoning are described. And chapter 3 expresses theory about spatiotemporal database and spatiotemporal moving object which becomes target of spatiotemporal reasoning study. In chapter 4, we propose spatiotemporal reasoning system model using spatiotemporal reasoning. And chapter 5 introduces spatiotemporal reasoning system for battlefield analysis to show the usefulness of spatiotemporal reasoning system model proposed in chapter 4. Finally, chapter 6 summarizes and concludes whole study, and presents future directions of study.

2 Spatiotemporal Reasoning

Spatiotemporal Reasoning is said to apply temporal reasoning and spatial reasoning together to the spatiotemporal data which is spatial object changed by time flow. In particular, it performs reasoning by inducing knowledge related to temporal data and spatial data, and uses spatiotemporal operation into which temporal operation and spatial operation are integrated by one. We need a preceding study about basic concept of temporal and spatial reasoning and about types of reasoning, since spatiotemporal reasoning has been worked from the separate study about temporal and spatial reasoning.

2.1 Temporal Reasoning

Temporal reasoning consists of formalizing the notion of time by reasoning for various characteristics of temporal data and providing means to represent and reason about the temporal aspects of knowledge. Moreover, it has the ability to handle dependencies among different temporal data, to handle incomplete temporal data, to determine the period of validity of values of data, to handle real and apparent contradictions, and to recognize incorrect data[Perk90, Vila94].

Temporal reasoning system using such features of temporal reasoning has basic tasks and additional tasks. First, basic tasks are Temporal Consistency Maintenance and Temporal Question-Answering. Second, additional tasks oriented reasoning about change and action consist of Explanation, Prediction, and Learning. Explanation is to produce a description of the world for past(historical) data, prediction for future data, and learning for repetitive temporal data. Some applications using temporal reasoning are as follows; medical diagnosis and explanation, planning, industrial process supervision, natural language understanding, and etc.[Vila94].

Temporal reasoning can have a different type of reasoning method depending on modeling method of temporal data, so we now discuss those methods of time modeling and types of temporal reasoning.

2.1.1 Temporal Data Modeling

Method of time modeling has the sequence of generally determining types of time domains, modeling temporal relationships, and determining temporal entities.

1. Types of Time Domains

   Time domains are divided into several types according to sort of time and aspects of representation how the time is expressed[Mont93].

   - Time Domain by time expression method
     Temporal reasoning system can be based on Time Domain in which there are Time Point and Time

Interval. Time point expresses time by a point and time interval expresses time in terms of interval. The system based on time point is called Point-based Systems, while the system based on time interval is called Interval-based Systems. However, we can use both time point and time interval by integrating them. in case of actually developing a system.

- Time Domain by sort of time
  Sort of time can be classified into linear time, branching time, and circular time. Linear time is the set of time points which express time by points, and all times can be ordered by a sequence. It may also be divided into Time Boundedness, Discrete Times, Dense Times, Continuous Times, Time Metrics, Relative Times, and Absolute times. Branching time is used defining a partial ordering of time points. Circular time represents recurrent events forming no ordering relationships among times. Determining Temporal Granularity, describing the unit of time element, is a key factor in time domain.

2. Modeling of Temporal Relationships

   Modeling of temporal relationships can be classified into three types, which are temporal logics, ordering information, and metric information[Mont93].

   - Temporal Logics
     Temporal logics are reasoning-oriented logics depending on the order column of time. A sentence clearly specified with time in temporal logics should always have the same result of true and false regardless of change of time, while the result to a sentence with inaccurate time can be mutually changed with true and false according to the change of time since true value is dependent on time.

     Mechanisms for expressing temporal logics are modal logic, modal temporal logic, topological temporal logic, and classical temporal logic. Modal logic shows the logic that true and false are changed according to possible world. Modal temporal logic, derived from modal logic, can be divided into point-based temporal logic and interval-based temporal logic. Topological temporal logic represents distance information to time. It enables one to express quantitative as well as qualitative temporal properties over both discrete and dense temporal domains. Classical temporal logics have performed temporal processes using temporal qualifier as a sort of metalanguage for describing relationships between temporal data.

   - Ordering Information
     Ordering information of time can be divided into point algebra and interval algebra, which describe time by points and by intervals respectively. The point algebra to time has the three basic relations as smaller than(<), larger than(>), and equal to(=), and two basic operations as intersection and composition. The interval algebra to time has the thirteen basic relations like before, after, meets, met-by, overlaps, overlapped-by, starts, started-by, during, contains, finishes, finished-by, and equal. Its basic operations include intersection and composition.

   - Metric Information
     Metric information of time can be represented by distance algebra and metric relation. Distance between time points in distance algebra is described as Distance, and distance between time intervals as Duration. Distance relation is the operation for the distance information of time which is expressed by distance algebra. Operations are intersection and composition.

2.1.2 Types of Temporal Reasoning

Temporal reasoning mainly handle interrelation of time on the basis of time domain established on the process of modeling time. Types of temporal reasoning can be divided into Reasoning about temporal structures and Derivation from incomplete information[Mont93].

1. Reasoning about Temporal Structures

Mechanism for reasoning about temporal structures is Constraint Satisfaction, and mainly uses algorithms of utilizing simplification and propagation for related problem solving. Simplification returns the minimal equivalent subset of a given set of constraints on a finite set of temporal variables, and uses logical equivalences as rewriting rules to remove constraints. Propagation makes the deductive closure of a set of constraints, looking for all constraints that can be derived from the originally given ones, and applies modus ponens to derive further constraints.

2. Derivation from Incomplete Information

   Incomplete information of temporal data in the temporal reasoning system is caused either by acquiring incomplete knowledge or by using system at different time granularities. Similarly, derivation from incomplete information has following two types.

   First, mechanisms for reasoning from incomplete knowledge are forward temporal projection, predicts the future by inferring consequences of specific facts, and backward temporal projection, explains the past to infer justifications of specific facts. Second, mechanisms for reasoning dealing with different time granularities are default projection rules, downward temporal projection, and upward temporal projection.

2.2 Spatial Reasoning

Spatial reasoning is a inference technique to figure out various kinds of spatial problems on the basis of relationship between spatial objects. That is, it is approached by using complex mathematical theory or operational research as well as geometry, to find out a solution among infinite possibilities. In this chapter, we will look at the types of spatial reasoning.

Spatial reasoning mainly manages mutual relationship between spatial objects based on spatial domain established in spatial data modeling. Since relations of spatial objects play a important role for query optimization in spatial database, accurate understanding of spatial relations is most essential. But, it is efficient that the system stores only what is frequently used and creates whenever needs occur, because it can hardly stores all the spatial relations.

Therefore, by storing spatial relation at minimum and offering inferable rule, new spatial relation can be inferred using well-known spatial relations. Typical reasoning types using spatial relation are reasoning about topological relations and reasoning about direction relations [Grig95, Hong95].

2.2.1 Reasoning about Topological Relations

Reasoning about topological relations is the inference technique using topological relations of spatial data. Topological relation between object X and Y on the space can be represented by the intersection for boundary, inside, and outside. In general, only eight topological relations like following table become to make sense[Grig95].

Expressions	Meaning
X equal Y	X is equal to Y
X meets Y	X meets Y
X overlap Y	X overlaps with Y
X contains Y	X contains a part of Y
X disjoint Y	X is disjoint with Y
X properly-contains Y	X properly contains Y
(Y contained-in X)	Y is partly contained in X
(Y properly-contained-in X)	Y is properly contained in X

Table 2.1: Spatial Topological Relations

Table 2.1 represents spatial topological relations. When there is spatial object X, Y, and Z, if relations of X, Y, and Z are R1 and R2, we can derivate relation R3 of X and Z using this kind of topological relation. We represent relation R1 of X and Y as (X R1 Y) and relation R2 of Y and Z as (Y R2 Z). We can use logics and transitivity in order to derivate new relation R3 of X and Z with these facts. Transitivity is the mechanism to derive the next result, if logical expression ”A -> B” is true and ”B -> C” is true, ”A -> C” is true. In the same way, if objects X, Y, and Z have relations R1, R2, and R3, we can derive this expression ”(X R1 Y) (Y R2 Z) -> (X R3 Z)”.

2.2.2 Reasoning about Direction Relations

Reasoning about direction relations is concerned with direction relations which constitute a special class of spatial relations that deal with order in space. There have been two approaches in defining direction relations. According to the cone-shaped approach, direction relations are defined using angular regions between objects. Another method is projection based approach, that is, direction relations are defined using projection lines vertical to the coordinate axes[Hong95].

There are eight primitive direction relations between spatial points, such as east(E), west(W), south(S), north(N), northeast(NE), northwest(NW), southeast(SE), and southwest(SW). Following Figure 2.1 shows these eight directions.

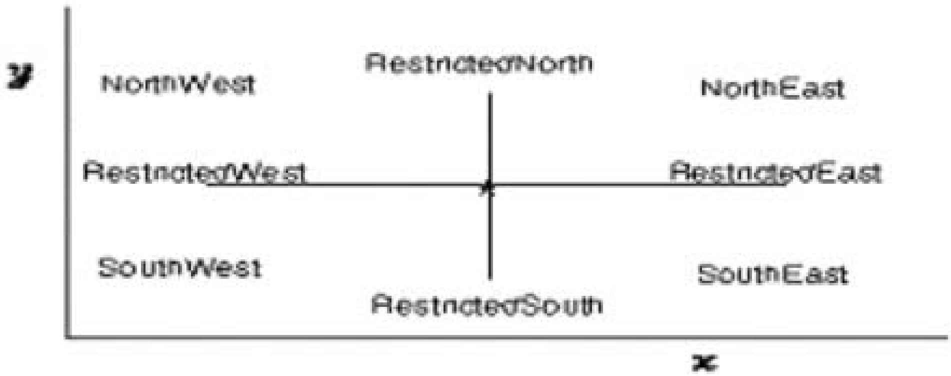

Figure 2.1: Eight directions indicating direction relations

Representation of direction relations using two points p and q, expressed by coordinate x and y like above figure 2.1, can be explained as follows.

Expressions	Meaning
E(p,q)	p is east of q. x-coordinate of p is larger than that of q.
W(p,q)	p is west of q. x-coordinate of p is smaller than that of q.
S(p,q)	p is south of q. y-coordinate of p is smaller than that of q.
N(p,q)	p is north of q. y-coordinate of p is larger than that of q.
NE(p,q)	p is northeast of q. x-coordinate of p is larger than that of q, and y-coordinate of p is larger than that of q.
NW(p,q)	p is northwest of q. x-coordinate of p is smaller than that of q, and y-coordinate of p is larger than that of q.
SE(p,q)	p is southeast of q. x-coordinate of p is larger than that of q, and y-coordinate of p is smaller than that of q.
SW(p,q)	p is southwest of q. x-coordinate of p is smaller than that of q, and y-coordinate of p is smaller than that of q.

Table 2.2: Expressions of two points with eight directions

Table 2.2 represents direction relations using two points p and q, expressed by coordinate x and y like figure

2.1. For example, the expression E(p,q) means that E expresses east direction and spatial point p is located in east of point q. Also, the direction relation between p and q is expressed by coordinate x and y, and coordinate x of p is surely greater than coordinate x of q.

3 Spatiotemporal Databases

In order to deal with spatiotemporal data applied in spatiotemporal reasoning, we may need to build a spatiotemporal database and its operation. Spatiotemporal databases store and manage geometries changing over time That is, they can effectively handle various spatial objects existing in real world as well as history of spatial object changing with time flow.

In this chapter, we are going to look at the mechanisms of spatiotemporal data modeling and the spatiotemporal operators which were studied in the past, and to explain the spatiotemporal data model and the spatiotemporal operators proposed in this study.

3.1 Spatiotemporal Modeling

Spatiotemporal data are usually modeled by extending temporal databases or spatial databases. That is, spatiotemporal data are modeled in one of two ways. First, we can add spatial properties and operations in temporal databases. The second way is to add temporal properties and operations in spatial databases. In general, the second way is used more than the first way.

3.1.1 Related Models

A classification of the spatiotemporal data model is given with snapshot data model[Arms88] , spatiotemporal composite data model[Lang88] , event-oriented data model[Peug95] , history graph model[Reno97] , three-domain data model[Yuan94, Yuan96, Yuan97] , and object-oriented data model[Worb90] , according to time adoption and its category of data occurred in the real world.

We categorize spatiotemporal data model into snapshot data model, spatiotemporal composite data model, event-oriented data model, history graph model, three-domain model, and object-oriented data model by the time that is occurred in the real world.

1. The Snapshot Model

   The snapshot model is one of the simplest spatiotemporal data models. Temporal information has been incorporated into snapshot spatial data model by time-stamping layers. Every layer in this model is a set of temporally homogeneous units of one theme. It could not describe temporal relations among layers, but it only shows the states of a geographic distribution at different times. Time is considered as an attribute of the location. Time intervals between any two layers may vary and there is no implication for whether changes occur within the time lag of any two layers [Arm88].

   The time dimension of the snapshot model is based on the linear, discrete, and absolute time model. Only valid time is supported while the model supports multiple granularities. This model is the simplest method for describing spatiotemporal information, but its function to support complex queries is almost limited. Therefore, the snapshot model can not properly describe changes in space through time, and is inadequate to manage history by units of the object.

2. The Spatiotemporal Composite Data Model

   The spatiotemporal composite data model, proposed by Gail Langran[Lang88], is a mechanism of overlapping multiple snapshot layers with one spatiotemporal composite layer, and of expanding snapshot data model. However, it contains the problem of data duplication and impossibility in dividing by history for spatial objects holding same history.

3. The Event-Oriented Data Model

   Peuquet[Peug95]'s Tempest model is applied object-oriented model for non-spatial objects and position-based model for spatial objects. The record of change history at the time of spatiotemporal object is composed of linked lists. In general GIS, more approaches are not made for the past state of history, but for the latest state. However, the event-oriented data models have a problem with expression of latest state, for the reason that they have to synthesize all the state change from the past data for confirmation of current data.

4. The History Graph Model

   Understanding temporal behavior in History Graph Model is one of the most fundamental issues in spatiotemporal systems. A simplistic view that many researchers seem to adopt, is to represent objects only in terms of static representations, viewing changes as sudden events. However, we know that many changes in the real world have duration. Actually, features in the real world exhibit a wide range of temporal behavior. These behaviors are outlined to three types as in [Sege93, Mont93, Reno97].

   First, Continuously changing objects. These are modeled with transitions whose duration span the whole extent of time. Snapshots that are versions which represent the state of the object at an instant of time are given between transitions as intermediate states. In this case, the state of an object between two snapshots is not known. Second, objects that are basically static, but which are changed by events that have duration. As an example, a settlement area holds small-scale map in the beginning, but it is gradually increased by population rise. Third, object that are always static and which are changed by sudden events. Administrative units and cadastral maps, which can be conceptually divided into domains, are typical examples of such behaviour. The main purpose of the history graph model is to identify the properties of changes in a spatial data set.

5. The Three Domain Model

   Yuan[Yuan94, Yuan96, Yuan97] describes a three-domain model for spatiotemporal modeling. This model represents semantics, space and time separately and provides links between them to describe geographic processes and phenomena. The semantic domain holds uniquely identifiable objects that correspond to human concepts independent of their spatial and temporal location. For example, this is in contrast to other models where a land owner is represented as an attribute of a land parcel. In the three domain model, the land owner is a semantic entity that is linked to a land parcel(spatial object), with changes to the parcel associated with dates(temporal objects). The advantage of this vector model is the ability to handle movement as well as change.

   The classification of change is given with semantic change, spatial change, and temporal change. Semantic changes include variations in attributes over time and the static spatial distribution of a geographic phenomenon. Spatial changes may be static, looking at variations of a geographic phenomenon at a snapshot, or transitional, comparing states of an event at different sites. Temporal changes are either spatially fixed mutations of an event or the actual movement of it from one place to another[Abra96].

6. The Object-Oriented Data Model

   The Object-Oriented Data Model, based on the object-oriented paradigm, makes it possible to embed all historical versions of the same object into one single entity. Worboys defines a spatiotemporal object as a unified object with both spatial and bitemporal extents on the basis of object-oriented concept[Worb90]. Worboys' model is based on the three approaches. The first is to add the temporal concept to timestamps of all geographic objects. The second approach is to add the temporal concept to the spatial object at the spatial object level. The last is to extend spatiotemporal classes by adding the temporal to the spatial object at the point level. This model represents change history of a object with temporal and spatial information by object-oriented approach, but it doesn't represent the change of a spatiotemporal object such as division and split.

7. Advantages and Disadvantages of Related Model

   Each spatiotemporal data model has advantage and disadvantage. The snapshot model is very simple but has the data duplication problem. The spatiotemporal composite data model solved the data duplication problem but doesn't distinguish objects which have the same histories. The event-oriented data model has disadvantage to combine the change state to get the recent state because history objects are connected by a linked list. The object-oriented data model easily represents spatiotemporal objects but its implementation is difficult. So the spatiotemporal data model depends on applications that apply it.

3.1.2 Spatiotemporal Operator

Spatiotemporal operator processes spatial operations for some objects which are satisfied with given temporal topological operation and have valid and transaction time. There are two main operators such as temporal topological operator and spatial topological operator.

1. Temporal Topological Operator

   Temporal topological operations represent topological relationship between time values of target objects. Allen suggested 13 operations[Alle83] that are available and the figure 13 shows these operations.

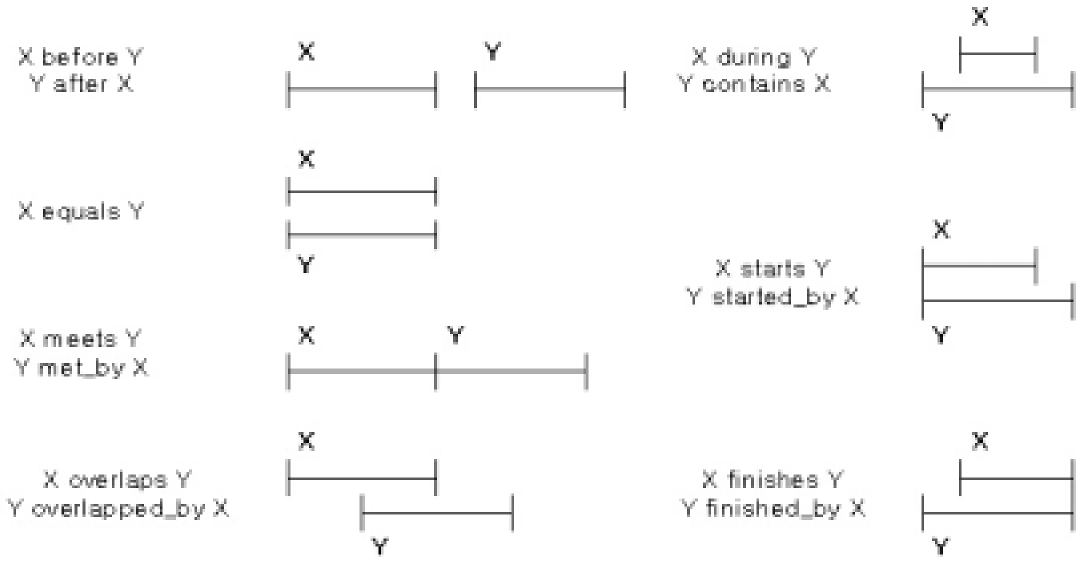

Figure 3.1: Allen's Temporal topological operator

   Figure 3.1 shows topological relationship between the time interval. which has start and end time of each operation. The operations, such as before and after, during and contains, have the inverse meaning except equals operation.

2. Spatial Topological Operator

   The spatial topological operations support different functions according to spatial management system. Geowin spatial management system supports the operations such as Enclosure, Identification, Inside, Intersection, Overlap, and Proximity. These operations return boolean value after analyzing topological relationship between two objects. Queried objects are point, chain, and ring and the operation between different objects is possible.

Operations	Functions
Enclosure	Inclusion Relation
Identification	Equivalent Relation
Inside	Located in a specific area
Overlap	Overlap Relation
Proximity	Adjacency Relation

Table 3.1: Examples of spatial operators in Geowin

   In table 3.1, each operator is used as format of 'A Operator B'. 'A Enclosure B' returns true when A includes B. 'A Inside B' returns true when A exists inside B. 'A Identification B' is equal relation and returns true when A's coordinates are equal with B's coordinates. Intersection and Overlap check the relationship such as inclusion or overlap between point, line, or area.

3.2 Moving Object Modeling

Moving objects are the object of which spatial data is changed in sequence over time. It can be largely divided into moving point and moving region[Erwi99, Guti2000].

3.2.1 Moving Points

Moving points are positions or locations changing over time. They are people, rocket, missile, tank, submarine, and etc. and have the following characteristics.

mpoint : time -> point

That is, moving point is changed according to time. Query for certain time returns points which describe existing position of moving object at that time. Query for moving point asks for the position of moving object, and does not ask regions of moving object. The queries relating to moving points are as follows.

People
Movement of a terrorist/spy/criminal
Animals
Determine trajectories of birds, whales,...
Which distance do they traverse, at which speed?
Where are the whales now?
Did their habitats move in the last 20 years?
Cars
Taxis : Which one is closest to a passenger request position?
Trucks : Which routes are used regularly ?
Planes
Were any two planes close to a collision ?
Are two planes heading towards each other(going to crash) ?
Did planes cross the air territory of state X ?
Ships
Are any ships heading towards shallow areas ?
Rockets, missiles, tanks, submarines
All kinds of military analyses

Table 3.2: Moving point queries

Table 3.2 shows queries which are related to path, direction, or distance of moving objects. The query related to a moving point doesn't include a query about moving object's shape.

3.2.2 Moving Regions

Moving regions are positions as well as shapes of objects changing over time. These include administrative area, progress of forests, influence of storms, or racial movement. They have characteristic as following.

mregion : time -> region

That is, position and shape are changed according to time. Query for certain time returns regions which describe existing position and shape of moving region at that time. Query for moving region asks for the position or shape of moving object at specific time. The queries relating to these are as follows.

Countries
What was the largest extent ever of the Roman empire ?
Which states split into two or more parts ?
Forests, Lakes
How fast is the Amazon rain forest shrinking ?
Is the Dead Sea shrinking ?
What is the minimal and maximal extent of river X during the year ?
Storms
Where is the tornado heading ?
When will it reach Florida ?
People
Movement of the Celt etc.
Troops, armies
Hannibal going over the alps. Show his trajectory.
When did he pass village X ?
Cancer
Can we find in a serious of X-ray images a growing cancer ?
How fast does it grow ?
How big was it on Jan 1, 1995 ?
Continents
History of continental shift.
High/low pressures areas, temperature, etc.

Table 3.3: Moving region queries

Table 3.3 shows queries which related to the shape of moving objects. The query of moving area includes a query about path, direction, or distance of a moving object, as well as a query about a moving point and in addition, it includes a query about the shape of a moving objects.

3.2.3 Method of Moving Objects Modeling

The positions of moving objects are continuously changing over time. For modeling these moving objects, we can consider both continuous and discrete models.

Continuous models allow us to represent the moving object in terms of infinite sets of points, and to view a moving point as a continuous curve in the 3D space. This model can accurately describe the motion information, but it is inadequate to be implemented since we cannot store and manipulate infinite points in computer.

On the other hand, Discrete models allow us to describe the moving object in terms of finite sets of points, and to view a moving point as a polyline in the 3D space. This can be implemented by showing the motion information of moving object in terms of approximate values. Although change of positions for the moving object is a continuous concept, it is necessary to use the discrete concept for modeling a system that can express change of positions for the moving object, since computer systems which we use have limited resources.

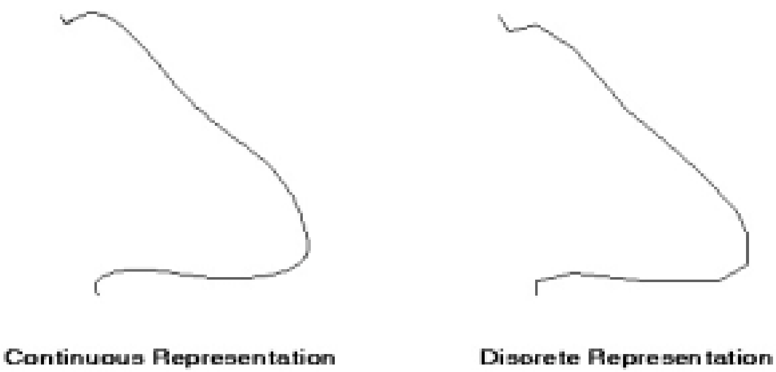

Figure 3.2: Continuous and Discrete Representations

In figure 3.2, the continuous model is represented on the left and able to be depicted by continuous curve. However, the discrete model is represented on the right and is depicted by polyline.

3.2.4 Moving Object Spatiotemporal Data Model

This section will describe MOST(Moving Object Spatio-Temporal) data model which is used in DOMINO(Database fOr MovINg Object) Project as an example of data model for spatiotemporal moving object[Sist98, Wolf98].

- Design Background

  Existing database management systems(DBMS's) are not well equipped to handle continuously changing data, such as the position of moving objects.

  Since the location of moving object is continuously changing over time, modifying operation is frequently needed and method to overcome it is required. Moreover, because existing DBMS only records current status of moving object, the query for expectation of future location of moving object cannot be processed.

- Characteristics of MOST Model

  Attributes of moving objects are classified and managed with dynamic attribute and static attribute. Dynamic attribute is the attribute that its value changes over time without explicit modifying operation. The value of dynamic attribute can be counted by means of function at specific time.

  For an example, the position of moving object can be calculated by time, speed, and route when we know its speed and route. The speed and route are managed as dynamic attributes, and are the motion information of moving object. If the motion information is changed, explicit modifying operation in database would be executed. The modifying operation for the motion information requires less operation frequency than explicitly changing the position of moving object.

- Structure of MOST Model

  MOST Model is implemented on the basis of existing DBMS. Its data are stored in DBMS, and its part of managing dynamic attributes is built on DBMS. By way of this, the queries for expectation of next location of moving object besides query processing function of existing DBMS can be performed.

3.2.5 Moving Object Operators

Various operators are used for moving object operators according to the kinds of moving object and the application areas adopted.

Operations	Functions
GetDistance	Obtain the distance between moving points for all time. ex) What's the distance between A and B for all time ? MDistance(A,B)
IsInside	Get the position of moving object when it is located inside of a specific area. ex) Is A located inside of B ? IsInside(A,B)
GetTrajectory	Get the trace of a given moving point, and describe it with line ex) What is A's Trajectory ? MTrajectory(A)
GetLength	Obtain total length of a given line. ex) obtain total length of a line L. Length(L)
GetMinValue, GetMaxValue	Get the minimum or maximum value among given values.

GetMinTime, GetMaxTime	Obtain the time when a moving object has a min. or max. value.
GetSpeed	Get the speed of moving object which corresponds to a given time value. (expect next position)
PositionAtTime	Seek the position of a moving object at specific time.

Table 3.4: Moving Object Operators

Table 3.4 shows an example of moving object operators, which will be used in this paper, on the basis of moving object operators proposed by Erwig[Erwi99].

Each operator has been added time concept to the existing spatial or arithmetic operator.

4 Spatiotemporal Reasoning System

It is a research area that can be applied to greatly wide areas, since the applications using spatiotemporal reasoning are entirely associated with applications of both spatiotemporal database and expert systems. Similar applications studied until now include navigational systems, war game models, and close combat simulations. We can find that data which become target for applied study are wholly moving object whose location is changed over time.

In this study, applied target for spatiotemporal reasoning is limited to moving object[Park2001]. And we propose features of spatiotemporal reasoning system for moving object and spatiotemporal reasoning system model.

4.1 Spatiotemporal Reasoning

Spatiotemporal reasoning is to perform several inference functions for spatiotemporal data changing over time flow. And the system which supports spatiotemporal reasoning function is called spatiotemporal reasoning system.

It must be designed and developed so as to support spatiotemporal reasoning function. Therefore, the very first thing to do for the development of spatiotemporal reasoning system is defining functions of spatiotemporal reasoning.

4.1.1 Functions

In developing spatiotemporal reasoning system, reasoning function and method depend on applied area and characteristics of data. But, in the application of spatiotemporal reasoning using moving data, there is a spatiotemporal reasoning function that has to be commonly performed. It means that the function reflects the general features of moving object. Therefore, functions which spatiotemporal reasoning system must provide can be classified into general function and function dependent on domain that is different from each other according to the features of applied area.

1. General Spatiotemporal Reasoning Function
   Application systems using moving object can have basic reasoning functions which must be commonly provided. That is, those are functions which greatly reflect the characteristics of moving object. Since the moving object has the characteristic that location of object on space is changing with time change, following two reasoning functions are needed. First one is location reasoning function that solves curiosity for where the moving object will move, and second is reasoning function for moving time that is required to get certain location. Because these two functions can affect user's decision making, they become very essential basic functions which system should have.

2. Spatiotemporal Reasoning Function dependent on domain
   Spatiotemporal reasoning system using moving object have to provide additional reasoning function required on applied domain besides common basic reasoning function. For what reason, since type and method of reasoning depend on the characteristics of application area, reasoning function for problem solving in the application area must be furnished. For instance, reasoning functions dependent on domain, which is provided in spatiotemporal reasoning system for battlefield analysis, include unknown unit inference, unidentified unit inference, main strike direction inference, and etc.

4.1.2 Types of Spatiotemporal Reasoning

In general, types of reasoning are classified into forward reasoning and backward reasoning according to the proceeding direction of reasoning. Similarly, spatiotemporal reasoning can be also classified into forward spatiotemporal reasoning and backward spatiotemporal reasoning.

1. Forward Spatiotemporal Reasoning
   Forward spatiotemporal reasoning is a prediction of future spatiotemporal situation using spatiotemporal knowledge and fact data. Battlefield analysis system, for example, consists of unidentified unit inference and main strike direction inference.

2. Backward Spatiotemporal Reasoning
   Backward spatiotemporal reasoning backtracks past fact and explain it in order to verify current recognized spatiotemporal fact. In battlefield analysis system, for example, there is a kind of unknown unit inference.

4.2 System Model

Spatiotemporal reasoning system is combined spatiotemporal database with knowledgebase system. It additionally has knowledgebase dependent on domain and inference engine to the elements of existing spatiotemporal database. It processes general spatiotemporal operations by using spatiotemporal data, and performs inferences by using spatiotemporal operators and knowledgebase.

4.2.1 Structure

Spatiotemporal reasoning system model proposed in this study is composed of as following figure 4.1.

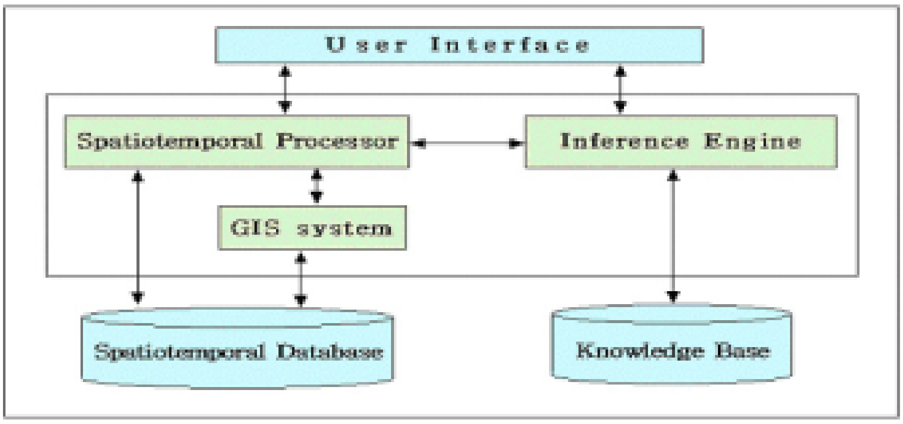

Figure 4.1: The proposed model for spatiotemporal reasoning system

Spatiotemporal reasoning system is made up of user interface, spatiotemporal processor, inference engine, GIS system, knowledgebase, and spatiotemporal database, as shown on figure 4.1. In this study, the system was designed to have following four features to perform the inference function for moving object.

1. To manage historical data for valid time and motion information of moving objects using spatiotemporal database.

2. To perform spatiotemporal reasoning by introducing spatiotemporal operator to inference process.

3. To store knowledgebase with domain knowledge relating to time and space of moving objects.

4. To utilize GIS tools to build a spatiotemporal database.

4.2.2 Elements

Spatiotemporal reasoning system consists of user interface, spatiotemporal processor, inference engine, GIS system, knowledgebase, and spatiotemporal database. They have their own features as follows respectively.

1. User Interface
   User can request the general queries relating to search and operation using spatiotemporal database as well as the queries related to spatiotemporal inference through GUI, and return their results.

2. Spatiotemporal Processor
   Spatiotemporal processor manages spatiotemporal database and performs spatiotemporal operation. It processes spatiotemporal operation required to spatiotemporal inference as well as general spatiotemporal processing.

3. Inference Engine
   Inference engine is an engine which infers changing characteristics and coordinate values of spatial objects by using data stored in spatiotemporal database, domain knowledge stored in knowledgebase, and spatiotemporal facts. Inference engine, at this time, can also call and use spatiotemporal processor to perform spatiotemporal operation as a preprocessor.

4. GIS system
   GIS system is a spatial database management system used to build a spatiotemporal database. A spatiotemporal database is built by using GIS system.

5. Knowledge Base
   Knowledge base is a key element deciding performance of reasoning system. It is a place where knowledges dependent on domain are stored. That is, it can be a database that stores rules together specifying the analysis method of data.

6. Spatiotemporal Database
   Spatiotemporal database stores general attribute, time attribute, and space attribute of moving object. Especially, it is stored with motion information of objects as historical database according to time, and stored with all spatial attributes by using GIS system.

5 Spatiotemporal Reasoning System for Application of Battlefield Analysis

This study adopted a business about battlefield analysis to a specific application, and developed a prototype of spatiotemporal reasoning system in order to show the effectiveness of spatiotemporal reasoning system model which was previously proposed. In this chapter, we will explain basic concept of the battlefield analysis and the design and implementation of prototype of spatiotemporal reasoning, and present the result experimented by applying a virtual scenario.

5.1 Battlefield Analysis

Battlefield Analysis is to collect and analyze information about enemy, terrain, and weather in an expected battlefield, and to study systematically battlefield's impact on friendly and enemy operations. In order to evaluate and analyze battlefield, various kinds of functions are required. However, the basic functions must be the identification of key terrain features, avenues of approach, and maneuver space, as well as the changes of unit position according to time change. That is, it is necessary to identify spatial relation of data, and to predict and analyze the future object which attribute value changes over time. For doing these, temporal reasoning, spatial reasoning, or integrated spatiotemporal reasoning becomes required.

In this research, we designed and implemented the system, focused on the features of spatiotemporal reasoning for moving units, that is, moving object among several considerations of battlefield analysis.

5.2 System Design

Spatiotemporal reasoning system for battlefield analysis has been designed and implemented on the basis of general spatiotemporal reasoning system model which was proposed in chapter 4.

5.2.1 System Configuration

The whole configuration of system designed for battlefield analysis is as following figure. This structure was designed on the basis of spatiotemporal reasoning model which was explained in figure 4.1.

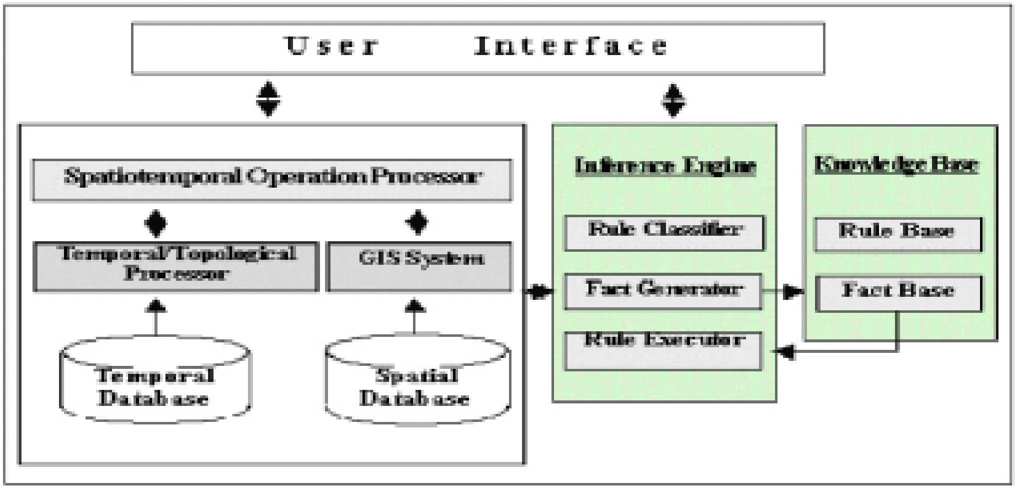

Figure 5.1: Spatiotemporal Reasoning System Configuration for Battlefield Analysis

1. Spatiotemporal database

   Spatiotemporal database for battlefield analysis is made up of initial unit data table, motion information table, and unknown unit data table.

   First, general attribute, location coordinate, and real observed date of observed units for the first time are recorded in initial unit data table. The general attribute of units include name, assigned regiment, and type. Second, Motion data table is the history table that records all motion processes of units stored in initial unit data table. In this table, general attribute, location coordinate, and date of units are recorded. Third, unknown unit data table is stored with unit data that its name or assignment is not identified, but location value is clear among observed moving units. This table is not stored with general attribute value, but only recorded with location coordinate and date.

   (a) Initial unit data table

   Unit's initial data are kept in Geowin, GIS tool. Data storage in Geowin is composed of three parts. They are Attribute Data table, which stores attribute data for object, Spatial Indexing Data Table, which stores indexing data, and Spatial Data Table, which stores spatial data. Among these, the table in which user can arbitrarily create and change desired attribute data is the attribute table. Therefore, additional name, so called ADT, is attached to its table name.

   - IPB_ADT(Attribute table for initial object data)
     Table IPB_ADT is an attribute table which stores initial data for units. This table is created when Geowin load initial data, and can add additional attribute fields.

id	Code	X_coord	Y_coord	VTs	VTe	Reg	Name
OID(Key)	target object code	X coord	Y coord	start time	end time	Regiment	object name
int	int	int	int	String	string	Int	string

Table 5.1: Structure of Initial unit data table (Attribute information)

Attribute data consists of oid, code, x coordinate, y coordinate, valid time value (start, end), regiment data and name of object.

Table 5.1 consists of object identifier(id), code name(code), x coordinate (x_coord), y coordinate(y_coord), start value of valid time(VTs), end value of valid time(VTe), Regiment information(Reg), and object name(Name) to express units initial information.

- IPB_SIT(Spatial indexing data table)
  Table IPB_SIT is an spatial indexing data table. This table is also created at data loading time as table _ADT, and generates and stores indices of R*-Tree for spatial objects.

NID	PID	SID	HGT	OID	XMIN	YMIN	YMAX	XMAX

Table 5.2: Structure of Initial unit data table(Spatial Indexing Information)

Table 5.2 includes the fields of NID, PID, SID, HGT, OID, XMIN, YMIN, XMAX, and YMAX. As shown in Table 5.2, XMIN, XMAX, YMIN, and YMAX fields express the boundary information of the area partitioned from spatial indices of R*-Tree. PID, SID, and NID fields have ID information for the upper and lower nodes.

- IPB_SOT(Spatial data table)
  IPB_SOT is an spatial data table and includes the fields of OID, ELEMENTS, EID, GEOTYPE, SEGMENTS, SID, POINTS, XO, and YO. It is the table which stores real coordinate data for spatial objects in space. One record has ten points and one spatial object can be composed of plural records. X0 and Y0 are mandatory input fields. It means that one record must have the minimum of one or more coordinates. This table is automatically created at the load time of initial data.

OID	ELEMENTS	EID	GEOTYPE	SEGMENTS	SID	POINTS	XO	YO
X1	Y1	X2	Y2	X3	Y3	X4	Y4	X5
X6	Y6	X7	Y7	X8	Y8	X9	Y9	

Table 5.3: Structure of Initial unit data table(Spatial Information)

In Table 5.3, fields $X0 \sim X9$, and $Y0 \sim Y9$ store coordinates of points, field of Segments indicates in which this record is included, and Geotype describes type of object that this record is included, that is, point, line, and area.

Next table 5.4 shows an example of initial unit data table for battlefield analysis stored in Geowin system.

ID	CODE	NAME	TYPE	VTs	VTe	X_Coord	Y_Coord	REG
1	4913	00-기계화부대	1	2000/05/01	2000/05/02	349	9	0
2	4919	14-100sp-포대	2	2000/05/01	2000/05/02	182	26	10
3	4925	15m-포대	2	2000/05/01	2000/05/02	166	48	10
4	4937	10-보대	3	2000/05/01	2000/05/02	152	73	10
5	4931	16r-포대	2	2000/05/01	2000/05/02	233	46	10
6	4976	12-보대	3	2000/05/01	2000/05/02	241	69	10
7	4943	11-보대	3	2000/05/01	2000/05/02	204	76	10
8	4982	44-200sp-포대	2	2000/05/01	2000/05/02	304	52	40
9	4988	45-200sp-포대	2	2000/05/01	2000/05/02	330	51	40
10	5041	32-보대	3	2000/05/01	2000/05/02	324	66	30
11	5000	47r-포대	2	2000/05/01	2000/05/02	396	41	40
12	5035	24-100sp-포대	2	2000/05/01	2000/05/02	531	15	20
13	5067	26r-포대	2	2000/05/01	2000/05/02	538	44	20
14	5061	25m-포대	2	2000/05/01	2000/05/02	518	35	20
15	5079	20-보대	3	2000/05/01	2000/05/02	506	75	20
16	5107	22-보대	3	2000/05/01	2000/05/02	573	69	20
17	5093	21-보대	3	2000/05/01	2000/05/02	535	65	20
18	4994	46-100sp-포대	2	2000/05/01	2000/05/02	363	49	40
19	5027	31-보대	3	2000/05/01	2000/05/02	355	61	30
20	5013	30-보대	3	2000/05/01	2000/05/02	386	59	30

Table 5.4: Initial Unit Information Data

Initial unit data table as shown in table 5.4 is entered with valid time information which is actual observed date for unit, coordinate value which represents unit location on space, and general attribute value related with unit assignment information.

(b) Motion Information Table

Motion information table manages historical data by keeping moving status of units, that is, moved location coordinate and date. Motion information are utilized for tracking movement routes of units, searching unit data by date, reasoning unknown unit and unidentified unit, and reasoning main strike direction. Motion information history table is stored and managed in Oracle database.

Code	X_coord	Y_coord	VTs	VTe	Reg	Name
target object code	X coord	Y coord	start time	end time	regiment	object name
int	int	Int	string	string	Int	string

Table 5.5: Structure of Motion Information Table

Motion information table, as shown in table 5.5, consists of object code(Code), x and y coordinates(x_coord, y_coord), start and end values of valid time(VTs, VTe), regiment data(Reg), and object name(Name).

Next table 5.6 shows an example of motion information table stored in Oracle database.

CODE	NAME	TYPE	VTs	VTe	X_Coord	Y_Coord	REG
4913	00-기계화부대	1	2000/05/02	2000/05/03	349	69	0
4913	00-기계화부대	1	2000/05/03	2000/05/04	382	99	0
4913	00-기계화부대	1	2000/05/05	2000/05/06	352	153	0
4913	00-기계화부대	1	2000/05/07	2000/05/08	272	211	0
4913	00-기계화부대	1	2000/05/08	2000/05/09	232	210	0
4913	00-기계화부대	1	2000/05/09	2000/05/10	198	212	0
4913	00-기계화부대	1	2000/05/10	2000/05/11	155	215	0
4913	00-기계화부대	1	2000/05/11	NOW	143	266	0
4937	10-보대	3	2000/05/02	2000/05/03	153	111	10
4937	10-보대	3	2000/05/03	2000/05/04	157	137	10
4937	10-보대	3	2000/05/05	2000/05/06	172	209	10
4937	10-보대	3	2000/05/07	2000/05/08	179	268	10
4937	10-보대	3	2000/05/08	2000/05/09	184	296	10
4937	10-보대	3	2000/05/09	2000/05/10	191	328	10
4937	10-보대	3	2000/05/10	2000/05/11	200	359	10
4937	10-보대	3	2000/05/11	NOW	211	386	10
4943	11-보대	3	2000/05/02	2000/05/03	206	122	10
4943	11-보대	3	2000/05/03	2000/05/04	209	144	10
4943	11-보대	3	2000/05/04	2000/05/05	216	179	10
4943	11-보대	3	2000/05/06	2000/05/07	225	248	10

Table 5.6: Motion Information Table

Motion information of units are stored in motion information table like table 5.6. They are classified with values of valid time, and describe the information of location for a corresponding unit at specific time. If the value of valid end time is 'NOW', it indicates that motion information of corresponding data are effective up to now.

(c) Unknown Unit Data Table

Unknown unit data table is stored with location coordinate and date information when the general attribute of unit can not be identified among the unit that motion information were observed. It is utilized for reasoning unknown unit, unidentified unit, and main strike direction after assembled disposition and uses Oracle database.

X_coord	Y_coord	VTs	VTe
X coordinate	Y coordinate	start time	end time
Int	int	String	string

Table 5.7: Structure of Unknown Unit Data Table

Unknown unit data table as shown in table 5.7 consists of x coordinate, y coordinate, and start time and end time of motion. Unknown unit is the information that general attribute was not identified from observed motion information, so only the location coordinate and observed date for unit are stored.

Next table 5.8 shows an example of unknown unit data table stored in Oracle database.

VTs	VTe	X_Coord	Y_Coord
2000/05/06	2000/05/07	307	188
2000/05/06	2000/05/07	176	241
2000/05/05	2000/05/06	217	216
2000/05/05	2000/05/06	256	214
2000/05/09	2000/05/10	277	326
2000/05/04	2000/05/05	174	160
2000/05/04	2000/05/05	166	182
2000/05/08	2000/05/09	134	372
2000/05/08	2000/05/09	129	363
2000/05/06	2000/05/07	521	238
2000/05/04	2000/05/05	600	165
2000/05/09	2000/05/10	639	313
2000/05/02	2000/05/03	546	58
2000/05/09	2000/05/10	574	324
2000/05/10	2000/05/11	568	345
2000/05/04	2000/05/05	420	123
2000/05/04	2000/05/05	385	124
2000/05/08	2000/05/09	291	206
2000/05/05	2000/05/06	412	144
2000/05/09	2000/05/10	305	187

Table 5.8: Unknown Unit Data Table

Table 5.8 is the data type that is observed and stored as unknown units. Unknown unit data table is stored with location coordinate and date of the unit. Since the minimum information about observed units as unknown is needed in order to infer a unknown unit, unknown unit data table is managed apart from motion information table.

2. Spatiotemporal Operation Processor

   Spatiotemporal operation processor performs spatiotemporal operation and search by using temporal and spatial attributes of object stored in spatiotemporal database. Next figure shows the structure of spatiotemporal operation processor.

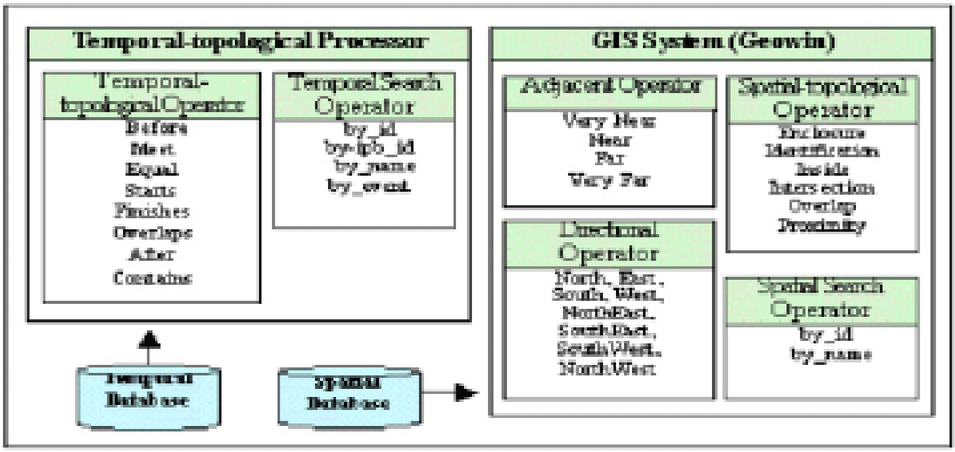

Figure 5.2: Spatiotemporal Operation Processor

Spatiotemporal operation processor in figure 5.2 is made up of temporal- topological operator, temporal search operator, spatial-topological operator, adjacent operator, directional operator, and spatial search operator. Temporal topological operator does operation by using both before and after relation and inclusion relation of time. Temporal search operator searches by using time attribute of data. Spatial topological operator performs operation by using topological relation of spatial data. Adjacent operator uses adjacent relation between spatial data, while directional operator uses directional relation of them. Spatial search operator searches by using space attribute of data.

3. Inference Engine

   Inference engine consists of inference engine and knowledgebase as a part of performing inference query. Moreover, inference engine is composed of rule classifier, fact data generator, and rule executor, and knowledgebase is composed of domain rule base and spatiotemporal fact base.

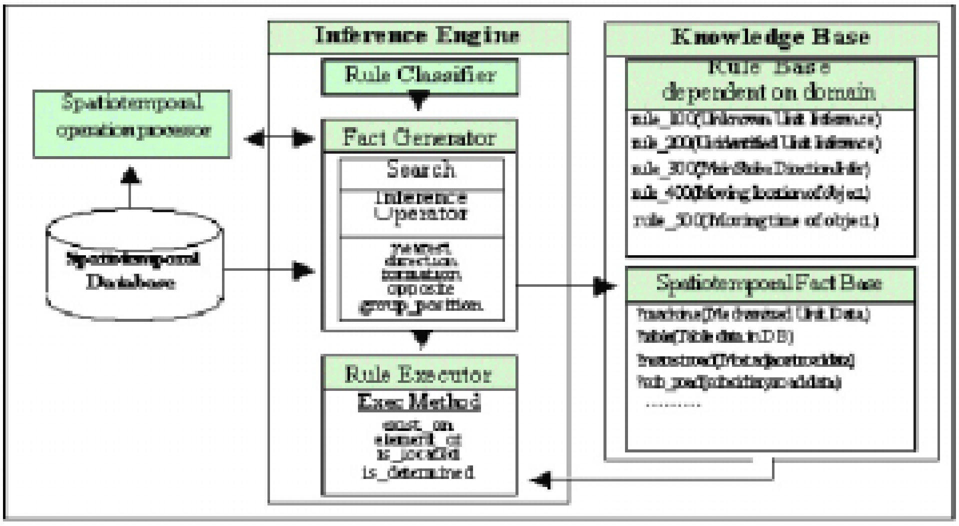

Figure 5.3: Inference Engine

Figure 5.3 describes the whole structure of inference engine. It is composed of inference engine part and knowledgebase part which manages knowledge used for inference.

(a) Inference Engine

Inference Engine consists of rule classifier, fact data generator, and rule executor.

First, the inference query is entered, then inference engine executes rule classifier. Rule classifier extracts an metadata to perform spatiotemporal reasoning. Metadata includes rule name which will be used for performing inference query, and fact data generating operator for performing creation of fact data.

Second, fact data generator becomes performed. Fact data generator creates spatiotemporal facts by using fact data generating operator among extracted metadata from rule classifier. Fact data generating operator executes additional spatiotemporal operators only applied to inference like nearest, direction, and formation as well as general spatiotemporal operators. Generated fact data are stored in factbase.

Third, it is the process of executing rules. That is, the rule extracted from the rule classifier is executed. Rule execution is the process of extracted rule from the rule classifier. Rule is existed in sequential enumeration of rules as ”if (condition part) then (action part)”[Dutt94, Stef95]. Fact information are obtained by fact data generator and stored in the fact base. When rules are executed, fact information are corresponded to the condition part of each rule. If rule is satisfied with all condition parts, the result of rule execution is ”true”, and otherwise the result is ”false”. Therefore, when rule is satisfied with all condition parts and the result of rule execution is ”true”, the action part of rule is only executed and displayed proper result of each query.

(b) Knowledge Base

Knowledge base is separately stored with rulebase and spatiotemporal fact base. Rule base collects and stores domain knowledges used for inference as rule type, and is inspected in sequence at time of rule execution. Spatiotemporal fact base is the place in which the results of spatiotemporal operations required to execute corresponding rule for inference execution are gathered. After new inference query is performed, the stored spatiotemporal facts will be totally erased, and different facts will be created and utilized.

- Rule Base

  Rules required to perform each inference query are assorted and stored with mutually different rule numbers. Stored rules include unknown unit inference rule, unidentified unit inference rule, and main strike direction inference rule. Following table shows an example of stored rules. Several rules for inferring main strike direction among these are as follows. First, rule that infers the main strike direction for the units on the move is described as shown in figure 5.4.

```
Rule_300() {
if(element_of(_Machine, _Table) && exist_on(_Machine, _Nearest_load)) {
_Main_attack_direct = is_determined(_Nearest_load);
draw3_1(_Main_attack_direct); }
if(element_of(_Machine, _Table) && exist_on(_Machine, _Nearest_load) &&
opposite_direction(_Sub_load, _Nearest_load)) {
_Sub_attack_direct = is_determined(_Sub_load);
draw3_2(_Sub_attack_direct); }
  }
```

Figure 5.4: Rule of Main Strike Direction Inference(on the move)

  Figure 5.4 represents the rule of main strike direction inference on the move. Rule_300 is number of the rule which consist of 'if (condition part) then (action part)'. Among these rules, _Machine, _Table, _Nearest_load, and _Sub_load are variables which represent spatiotemporal facts obtained by fact data generator. Also, element_of, exist_on, is_determined, and draw3_1 are names of method which are executed in condition and action part of rules.

  There are three considerations on determining main strike direction such as locations of mechanized units, formation of forward regiments, and direction of rear area regiment. Determining main strike direction on the move explained before considers only the locations of mechanized units out of these three considerations. However, main strike direction after assembled disposition is inferred by considering all three considerations. Therefore, the feature applying arbitrary weights from user's desire is added. Applying default weight is to apply no weights to three considerations, and applying user weight is that user arbitrarily applies weights to three considerations.

```
Rule_400() {
if(element_of(_Machine, _Table) && exist_on(_Machine, _Nearest_load)) {
_Attack_1 = is_determined(_Nearest_load);
draw4_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);
draw4_2(_attack_2, _Weight_2); }
if(element_of(_Jede_2, _Table) && leaned_to(_Jede_2, _Direction)) {
_Attack_3 = is_determined(_Direction);
draw4_3(_Attack_3, _Weight_3); }
}
```

Figure 5.5: Main Strike Direction Inference Rule(assembled disposition)

  Figure 5.5 shows the rule related to main strike direction after assembled disposition. Rule_400 is a number of rules which consist of 'if (condition part) then (action part)'. Among these rules, _Machine, _Table, _Nearest_load, _Attack_1, and _Weight_1 are variables which represent spatiotemporal facts obtained by fact data generator. Also, form_of, is_located, and leaned_to are names of method which are executed in condition and action part of rules.

  Expression of rules used in figure 5.4 and 5.5 follows the IF-statement format in JAVA language, and each rule is formed with one method in class rule.

- Spatiotemporal Fact Base

  Spatiotemporal fact base is the place where fact data for executing inference query is stored. In general, fact base tends to have same data at all times. However, spatial data according to time, that is spatiotemporal facts, can be changed, since objects are occasionally moving in case of using spatiotemporal data. Therefore, spatiotemporal fact base always has the characteristics of changing dynamically. Especially, fact base is to form different factbase with each other according to type of each inference query, and it is created when one inference query is executed, while it is gone after the query execution. Next table 5.9 is an example of fact variable which forms fact base.

Fact Variable	Meaning
_Machine	Spatiotemporal data of mechanized unit
_Table _Nearest_load	Presence of target table Most adjacent movement route
_Jede_2	Presence of second echelon regiment
_Form_jede_2_reg	Formations of second echelon regiment
_Position	Location data for second echelon regiment
_Direction	Biased direction data for second echelon regiment

Table 5.9: Spatiotemporal Facts

Table 5.9 shows examples of the name and mean of spatiotemporal fact variable. Especially, in order to differentiate spatiotemporal fact variable from general variable, we make a variable name by add underscore(_) to the first part of fact variable. For example, a variable by the name of _Machine is a spatiotemporal fact which is extracted location value from the spatiotemporal information of mechanized unit. This fact is most important data for the inference of main strike direction. Because of the most important factor, in order to decide main strike direction, is information of the location of mechanized unit.

- Fact Data Generating Operator
  Fact data generating operator is told operators used to form spatiotemporal factbase. Fact data generating operation is to use different operation with each other according to the type of each inference query. Fact data generating operators representatively include operators like nearest, direction, and group_formation.

  Operation nearest is to search a unique object most adjacent to certain object during valid time. For example, it is the operator that can perform such query as which road is most adjacent to 10-infantry battalion on May 5, 2000 ?

  Operation direction is to search direction information that certain object holds for valid time. This operation is a very abstract concept, and so it must be simplified and implemented in order to be suitable for application system. For example, it is the operator that can perform such query as which direction is the rear area regiment inclined on May 5, 2000 ?

  Operation group_formation is to identify formation shape(type) of the group when several units are assembled and make a specific formation. This operator is designed for applying to special purpose such as battlefield analysis. For example, it is the operator that can perform such query as of which type of formation is the battalions of 10-regiment made on May 5, 2000 ?

5.3 Implementation

5.3.1 Environment and Extents

Implementation environment for battlefield analysis system is as following table.

Classes	Tools
Operating System	Windows NT 4.0
Programming Language	Java

Database Management System	Oracle
Spatial Management System	Geowin

Table 5.10: Implementation Environment

Table 5.10 represents implementation environment of total system. In this study, we got the advantage of visualization of unit's movement by using the JAVA language. Also, we chose Geowin system for spatial management system, because of Geowin is developed to implement GIS applications using JAVA.

The extent of total system implementation can be divided into spatiotemporal query and inference query by function as following table 5.11.

Functional Classification	Kinds of Functions
Inference Queries	Unknown Unit Inference Unidentified Unit Inference Main Strike Direction Inference (on the move, assembled disposition) Unit's Moving Location Inference Unit's Moving Time Inference
Spatiotemporal Queries	Unit Movement Route Tracking Unit Data Inquiry by Date Adjacent Object Search
Spatiotemporal Operator	nearest : Adjacent Object Search trajectory : Movement Route Tracking attime : Unit Data Inquiry by Date

Table 5.11: The extent of implementation

Table 5.11 shows the functional classification which user actually asks query to the system and obtains its result. Even though not suggested in table 5.11, there are internal operators and queries that are internally implemented and executed to perform actual queries.

5.3.2 Function Explanation

The implemented system classify queries according to function into two types which are inference query and spatiotemporal query. Although this is a spatiotemporal system for battlefield analysis, this system needs spatiotemporal query to understand characteristics of spatiotemporal data.

1. Inference Queries

   Inference queries consist of reasoning function for battlefield analysis and general spatiotemporal reasoning function. There is Unknown Unit Inference, Unidentified Unit inference, and Main Strike Direction Inference Query implemented for the special purpose of battlefield analysis. And general spatiotemporal reasoning queries consist of Moving Location Inference and Moving Time inference. The meaning of each inference query is as follows.

   (a) Unknown Unit Inference
   Unknown unit inference is the function of reasoning unit's name and assignment data, when observation date and location coordinate among acquired motion information are correct, but unit's name and assignment data aren't. Unknown unit inference adopts backward spatiotemporal reasoning among spatiotemporal reasoning. The backward spatiotemporal reasoning proves that the current fact is valid by finding unit's general attribute data using the past motion information based on current fact data of unit's location.

(b) Unidentified Unit Inference
Unidentified unit inference is the function of reasoning name and location coordinate of unit which has absolutely no acquired motion information. Unidentified unit inference adopts forward spatiotemporal reasoning among spatiotemporal reasoning. Forward spatiotemporal reasoning is the inference processing using acquired motion information until now in order to predict location of unit which has to be exist but, has no acquired motion information. Therefore the results of unidentified unit inference differ from correctness according to acquired unit's motion information and additional related knowledge.

(c) Main Strike Direction Inference
Main strike direction inference is the function of reasoning major unit's moving direction in the future by using unit's motion information until present and other former knowledge. Main strike direction on the move is inferred by only using motion information and location data for mechanized units. However, main strike direction after assembled disposition is inferred by considering formation of forward regiment and location data for rear area regiment, as well as motion information and location data for mechanized units. Main strike direction inference adopts forward spatiotemporal reasoning among spatiotemporal reasoning.

(d) Unit's Moving Location Inference
Unit's moving location inference predicts location coordinate where a unit will move in the future and prints out the result on the screen. The moving location inference is needed for all general spatiotemporal reasoning system using moving objects. In this study, in order to inference unit's moving location, we assumed that moving objects move in the regular direction and along the movable road.

(e) Unit's Moving Time Inference
Unit's moving time inference receives a future location coordinate for a unit, and shows predicted time to arrive at that position. In order to inference unit's moving time, we assumed that moving objects move in the regular direction and along the movable road.

2. Spatiotemporal Queries

   - Unit Movement Route Tracking
     Unit movement route tracking displays whole movement routes of one or more units on the screen. Whole movement route is to be tracked to the coordinates moved from initial location to current location of a unit and to be shown.
   - Unit Data Inquiry by Date
     Unit data inquiry by date is the function of searching unit's location coordinate corresponding within scope of dates which user input, and of displaying the result, unlike unit movement route tracking. Time point or time interval may be feasible for user input date.
   - Adjacent Object Search
     Adjacent object search is the function of searching unit or road data to which a unit is adjacent. Provided adjacent relations include operators like nearest, very near, near, far, and very far. Among these operators, nearest searches the only one closest object, and very near, near, far, and very far find out all the objects contained within the scope of adjacent relation operator fixed earlier, and give us the results.

5.3.3 Inference and Query Processing

1. Inference Queries Processing

   Inference queries suggested in this system are unknown unit inference, unidentified unit inference, main strike direction inference, unit's moving location inference, and unit's moving time inference. Inference procedure which is commonly applied in each inference query is performed as three steps.

Figure 5.6: Flowchart of Unknown Unit Inference Execution

Figure 5.6 depicts the inference process that performs rule classifier, fact data generator, and rule executor in regular sequence as follows.

Step 1 : Rule Classifier Processing
First, inference engine will execute rule classifier as soon as an inference query is inputted. Rule classifier extracts metadata to perform spatiotemporal reasoning. Metadata include rule name and fact data generating operator, which will be used for inference query execution and be executed at time of fact data creation, respectively.

Step 2 : Fact Data Generator Processing
Second, inference engine executes fact data generator. Fact data generator creates spatiotemporal facts by using fact data generating operator among metadata extracted from rule classifier. Fact data generating operator performs general spatiotemporal operator as well as additional spatiotemporal operator only applied to its inference. Generated fact data are kept in factbase.

Step 3 : Rule Executor Processing
Last step of inference engine is the process for executing rules. Rule execution is to execute the rule extracted from rule classifier. Rules are continuously arranged with rules like if (conditional clause) and then (active part). At this time, fact data, which acquired from fact data generator and stored in factbase, are mapped to conditional clause of each rule. If conditional clause of the rule is entirely satisfied, it would be ”true”, but even if one is not satisfied, it would be ”false”. Active part of the rule can be executed only when all conditional clauses are satisfied and ”true”, and is to print out the result which corresponds to each inference query.

Unknown unit inference, unidentified unit inference, and main strike direction inference, which will be explained later, have different type of execution algorithm with each other according to the characteristics of each query. However, inference performing procedure in figure 5.6 are always applied as the same.

Therefore, we will omit the common inference procedure in figure 5.6, and importantly explain the inference algorithms, such as the order in which the conditions forming each real inference rule are inspected, and at which condition the set of fact data will be used.

(a) Unknown Unit Inference
Unknown unit inference is the function of reasoning unit’s name and assignment data, when observation date and location coordinate out of observed motion information are correct, but unit’s name and

assignment data aren't. The sequence of rule execution and inference is depicted as figure 5.15.

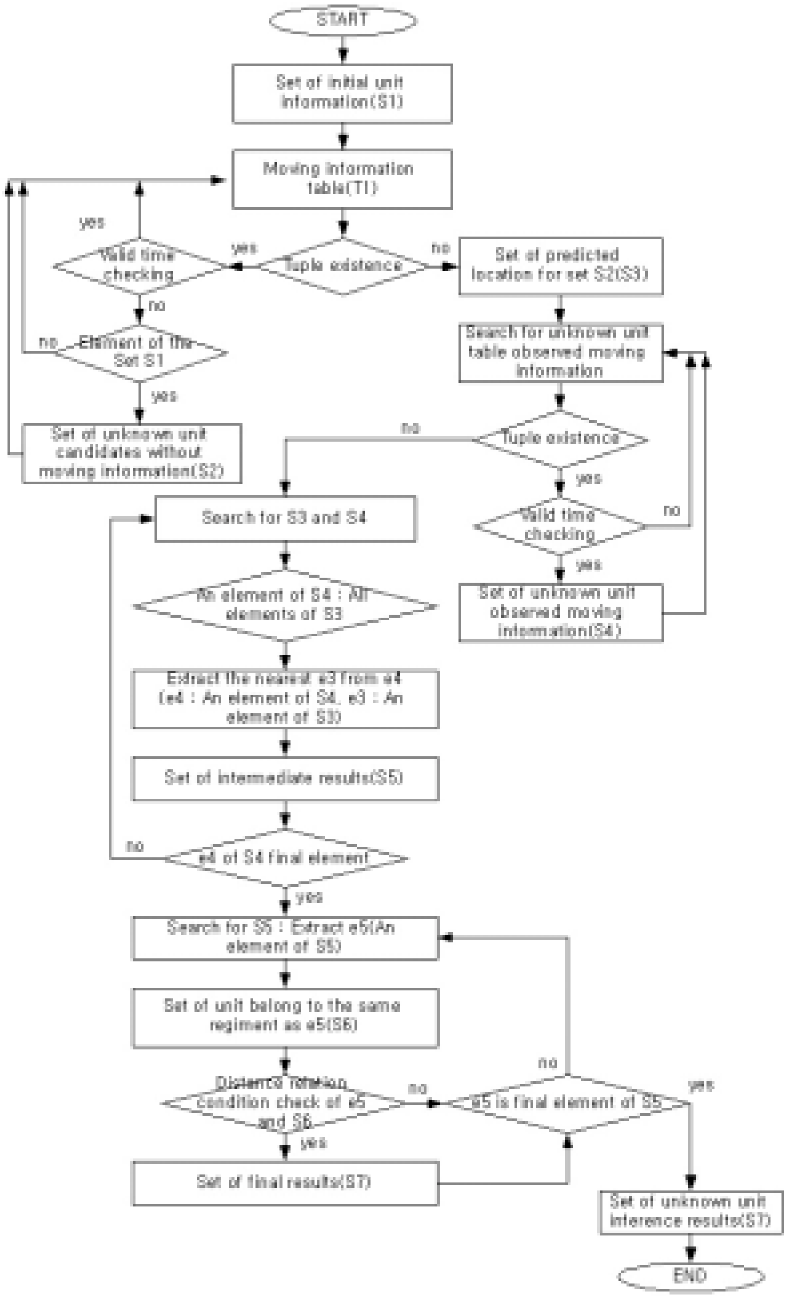

Figure 5.7: Flowchart of Rule execution for Unknown Unit Inference

Figure 5.7 shows execution sequence of rule which is adopted to unknown unit inference. Unknown unit means that unit information out of observed motion information is incorrect, and the location and date information are only stored in unknown unit data table. Figure 5.7 is performed as follows.

Step 1 : Create a set of unknown unit candidates
This step creates all units that do not have actual motion information as a set of unknown unit candidates. The set of unknown unit candidates is a set of extracted units that exist in the initial unit data table, but do not exist in the motion information table.

Step 2 : Create a set of expected locations for unknown unit candidates

This step creates a set of expected locations for unknown unit candidates by means of moving location inference function for all elements of the set of unknown unit candidates created in step 1.

Step 3 : Create a set of unknown units whose motion information is observed
This step creates a set of unknown units whose motion information is observed by extracting all location information which corresponds to inputted date from unknown unit data table.

Step 4 : Extract most probable unknown unit candidate and Create a set of intermediate results
This step selects the closest unit among the set of expected locations in step 2 for each element of the set of unknown units whose motion information is observed in step 3, and stores the set of intermediate results.

Step 5 : Create a set of units assigned to same regiment
This step creates a set of units assigned to same regiment for each element of the set of intermediate result created in step 4.

Step 6 : Inspect distance relation condition and Create a set of final results
This step inspects distance relation condition between elements of the set of units assigned to same regiment in step 5 for each element of the set of intermediate result in step 4. If the unit extracted as intermediate result is satisfied with this condition, it would be stored to the set of final results.

(b) Unidentified Unit Inference
Unidentified unit inference is the function of reasoning name and location coordinate of unit which has absolutely no acquired motion information. The sequence of rule execution and inference is depicted as figure 5.8.

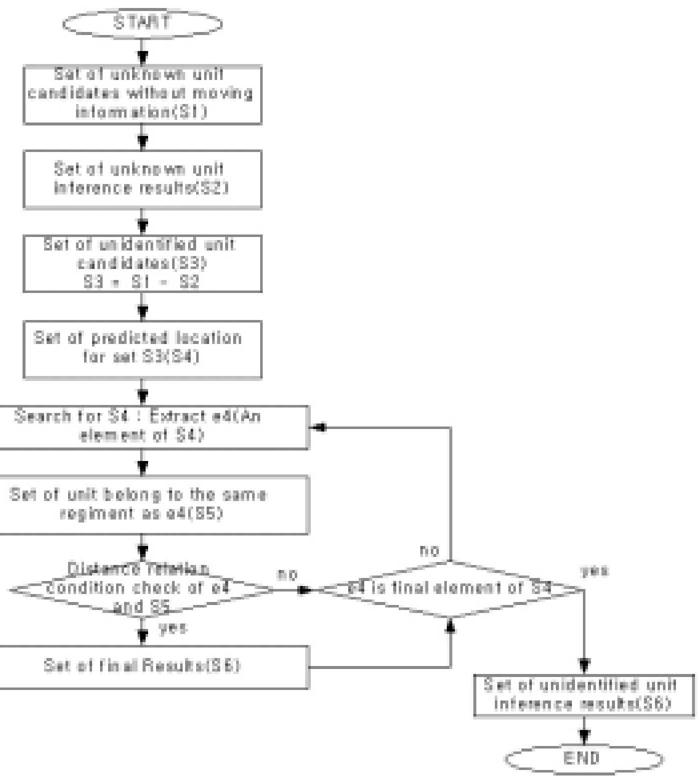

Figure 5.8: Flowchart of Rule execution for Unidentified Unit Inference

Figure 5.8 shows the execution sequence of the rule which is adopted to unidentified unit inference. Unidentified unit is a unit which has no motion information at all. Therefore, it means the unit which is not stored to the set of unknown unit inference results among the elements of the set of unknown unit candidates created in unknown unit inference. Figure 5.8 is performed as follows.

Step 1 : Create a set of unidentified unit candidates
The set of unidentified unit candidates is created by eliminating elements that belonged to the set of unknown unit inference results from the set selected as unknown unit candidates.

Step 2 : Create a set of expected locations of unidentified unit candidates
This step creates a set of expected locations by means of moving location inference function for each element of the set of unidentified unit candidates in step 1.

Step 3 : Create a set of units assigned to same regiment
This step creates a set of units assigned to same regiment for each element of the set of expected locations created in step 2.

Step 4 : Inspect distance relation condition and Create a set of final results
This step inspects condition for distance relation between elements of the set of units assigned to same regiment in step 3 for each element of the set of expected locations in step 2. At this time, if the unit's location information is included within the scope of distance relation condition, it would be stored to the set of final results.

(c) Main Strike Direction Inference
Main strike direction inference is the function of reasoning major unit's moving direction in the future by using unit's motion information until present and other former knowledge. The sequence of rule execution and inference is depicted as figure 5.9.

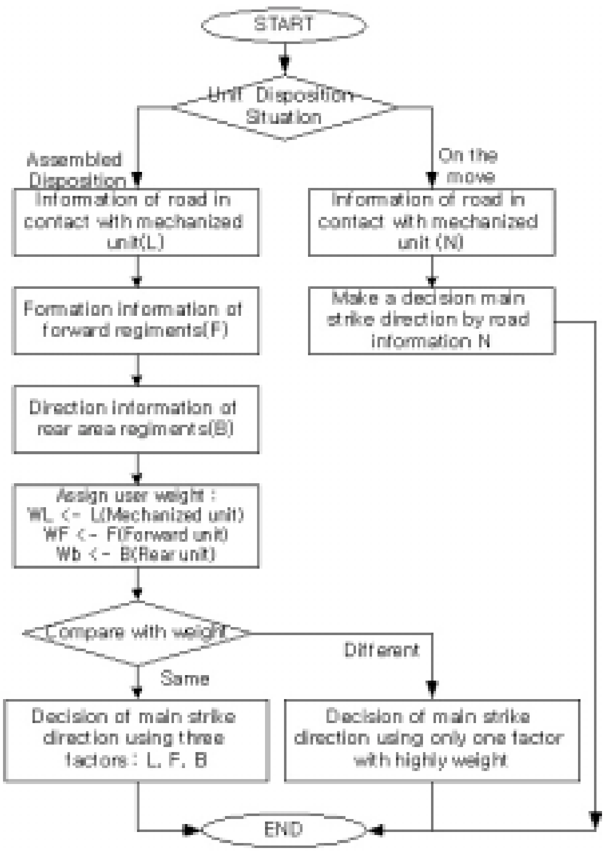

Figure 5.9: Flowchart of Rule Execution for Main Strike Direction Inference

Figure 5.9 shows the execution sequence of the rule which is adopted to main strike direction inference. Main strike direction inference is divided into on the move and after assembled disposition. Main strike direction inference on the move is a simplification of main strike direction inference after assembled disposition, and so here we will explain the process for main strike direction inference after assembled disposition only. The process for main strike direction inference after assembled disposition is as follows.

Step 1 : Create road information most adjacent to mechanized unit
This step extracts mechanized unit data which are contained within valid time of inputted date from motion information table, and searches and creates road information most adjacent to that mechanized unit.

Step 2 : Create formation information of forward regiments
This step extracts all location data for forward regiment which is included within valid time of inputted date from motion information table. From the created location data, the formation of forward regiment is classified into either triangle or line.

Step 3 : Create direction information of rear area regiments
This step extracts all location data for rear area regiment which is included within valid time of inputted date from motion information table. From the location data for rear area regiment, the road formation most adjacent to the inclined direction of rear area regiment is extracted.

Step 4 : Compare with weight values and Create the result
Above all, this step compares weight values with each other for the information created in step 1, 2, and 3, and then determines main strike direction on the basis of largest weighted data and most adjacent road. If three weight values are all the same, main strike direction for each three condition must be presented respectively.

2. Spatiotemporal Queries Processing

   We describes movement route tracking and object data inquiry by date algorithm.

   (a) Movement Route Tracking

      Movement route tracking algorithm is the function of searching whole route moved from initial location to latest location for one or more units, and of showing the result to users. Movement route tracking is processed with the sequence like following figure.

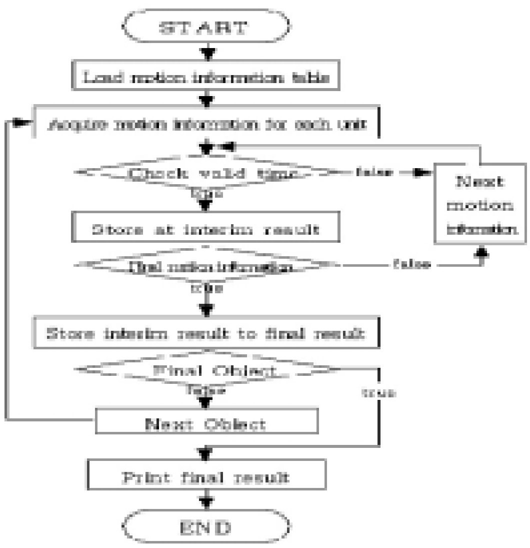

Figure 5.10: Flowchart of Movement Route Tracking

Figure 5.10 is the flowchart of processing the movement route tracking. First, all motion information from motion information table is loaded, and motion information for each unit is acquired. Acquired motion information for each unit is sorted by valid time, and unit location is stored to intermediate result storage variable until processing unit's all motion information. After processing one unit, intermediate result is stored to final result storage variable, and then the same procedure for next unit will be performed. After the procedures for all units are finished, the final result will be output.

(b) Object Data Inquiry by Date

Object data inquiry by date searches data for all units which are included within the scope of valid time user wants, and presents its result. Object data inquiry by date is processed with the sequence as following figure.

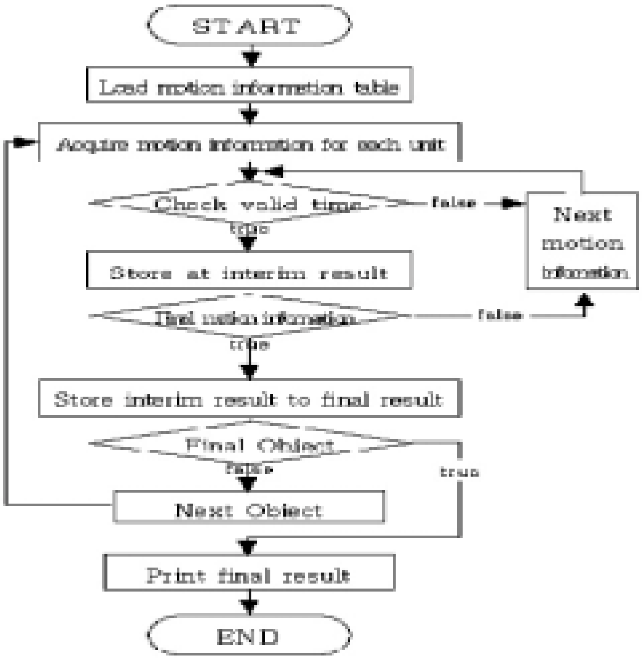

Figure 5.11: Flowchart of Object Data Inquiry by Date

Figure 5.11 explains the flowchart of processing the inquiry of object information by date. First, all motion information from motion information table is loaded, and valid time for this is inspected. The object which has valid time desiring inquiry is stored to the result storage variable, and repeated until inspecting all motion information. After the procedures for all motion information are finished, the final result will be output.

5.4 Experimentation

A virtual scenario has been applied to carry out the experimentation in order to show total features and procedure of implemented Battlefield Analysis System.

5.4.1 Applied Scenario

Scenario goes on with Create Initial Unit Data, Create Motion Information, Acquire and Input Motion Information, and Perform Queries in sequence.

- Step 1 : Create Initial Unit Data
  We created one division data as a target to create initial unit data. It was named ”1st Division”. There are three regiments and twenty battalions in 1st Division. In total, twenty units data would be generated because

the storing unit of an object data is based on the battalion. Location coordinates of initial unit data were arbitrarily set and its observation date was put on May 1, 2000.

- Step 2 : Create Motion Information
  Motion information for ten days are created by using twenty battalions made as initial unit data. Thus, total number of times occurring motion information becomes 200 times, and accurate motion information are made to create 80%(160 times), while the remainder of 20%(40 times) is to created as inaccurate motion information.

- Step 3 : Acquire and Input Motion Information
  User observes the created motion information and inputs them. After user input data by using input menu for motion information, motion information are stored in database. We suppose that motion information observation dates would be from May 2, 2000 to May 11, 2000.

- Step 4 : Execute Queries
  Based on scenarios created and stored until now, user executes several queries system provides. Execution of queries proceeds with the order of movement route tracking, unit data inquiry by date, adjacent object search, unknown unit inference, unidentified unit inference, main strike direction inference, moving location inference, and moving time inference.

5.4.2 Result of Scenario Execution

Experimentation result shows the query execution process of the scenario. It goes in sequence with motion information input, movement route tracking, unit Data search by date, adjacent object search, unknown unit inference, unidentified unit inference, main strike direction inference, moving location inference, and moving time inference. Created scenario is numbered and divided into, that is, scenario 1 and scenario 2. User must choose scenario to be executed first before the query input.

1. Motion Information Input

   Observed motion information can be inputted using input window for motion information as follows. User query is as follows.

   - Query : "Input the motion information of 00-mechanized unit observed at location (345, 560) on May 6, 2000 for scenario 1"

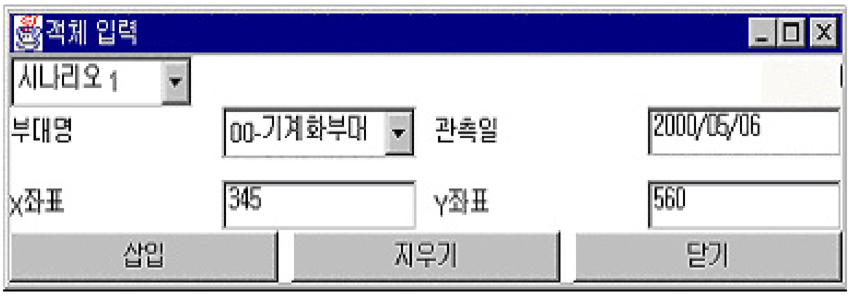

Figure 5.12: Input Motion Information

   Figure 5.12 shows an example of inputting observed unit data into the database of scenario 1. It shows to input the result that '00-mechanized unit' is observed at location of (345, 560) on May 6, 2000.

2. Movement Route Tracking

   The window of inquiring the movement route and its query is as follows.

   - Query : "Track the whole movement route for 2nd regiment."

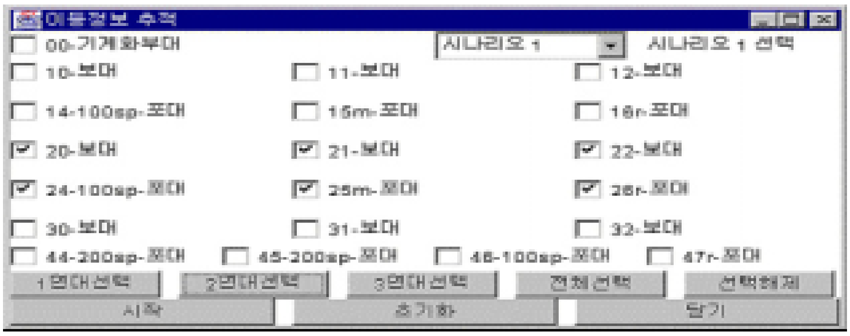

Figure 5.13: Making a query of Movement Route Tracking

Figure 5.13 is the query input window that tracks movement route for 2nd regiment among unit data kept in scenario 1. This is executed by selecting scenario and corresponding unit and pressing start button. In order to help user's convenience, select button for each regiment or whole select button for all units can be used. Result of executing the movement route tracking function is as following figure 5.14.

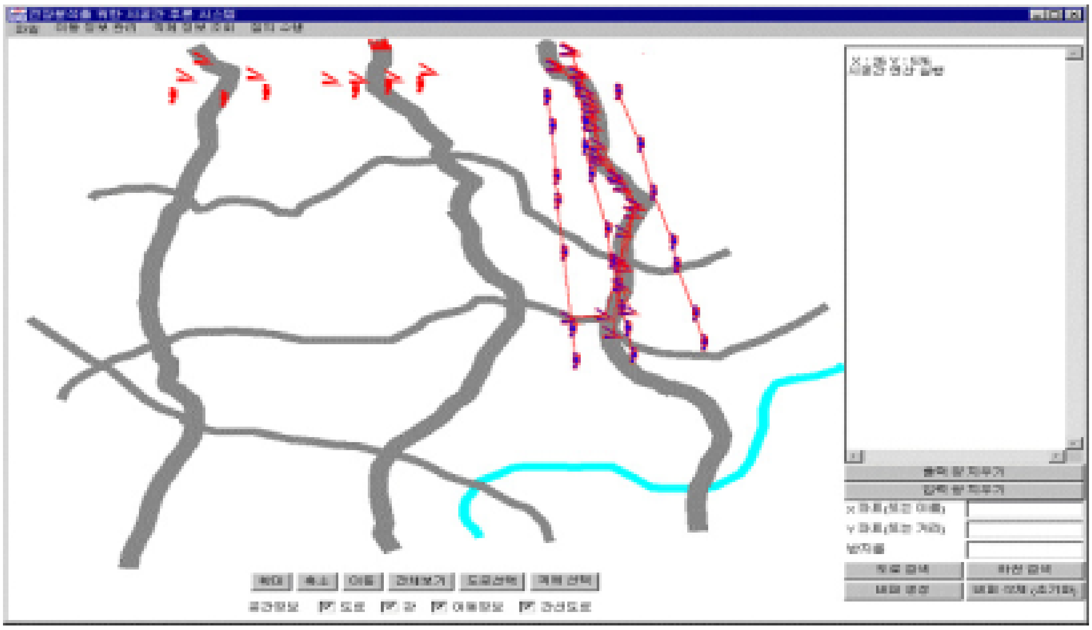

Figure 5.14: Result of the Movement Route Tracking

Executing movement route tracking function as figure 5.14, the moving location by date for the inquired unit come out to the screen, after initial unit data are output. The movement route comes out based on observation date inputted as motion information, and links with straight line to display before and after relation of moving locations clearly. Especially, it displays the whole movement route not at once, but one at a time in unit's moving sequence to show user more actual result screen.

3. Unit Data Search by Date

   Unit data inquiry by date is the function of searching unit's location coordinate corresponding within scope of dates which user input, and of displaying the result. Following figure 5.15 is the query input window that inquiries unit data by date.

   - Query : "Inquiry unit information from May 5, 2000 to May 8, 2000 for scenario 1."

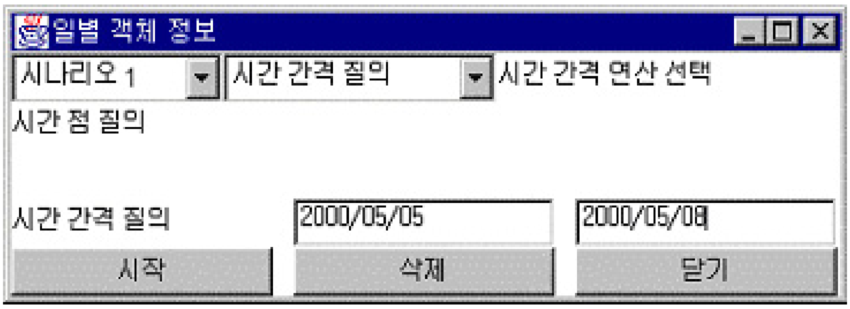

Figure 5.15: Making a query of Unit Data Inquiry by Date

Figure 5.15 is a query input window that inquiries unit data from May 5, 2000 to May 8, 2000 for scenario 1, and the result is as follows.

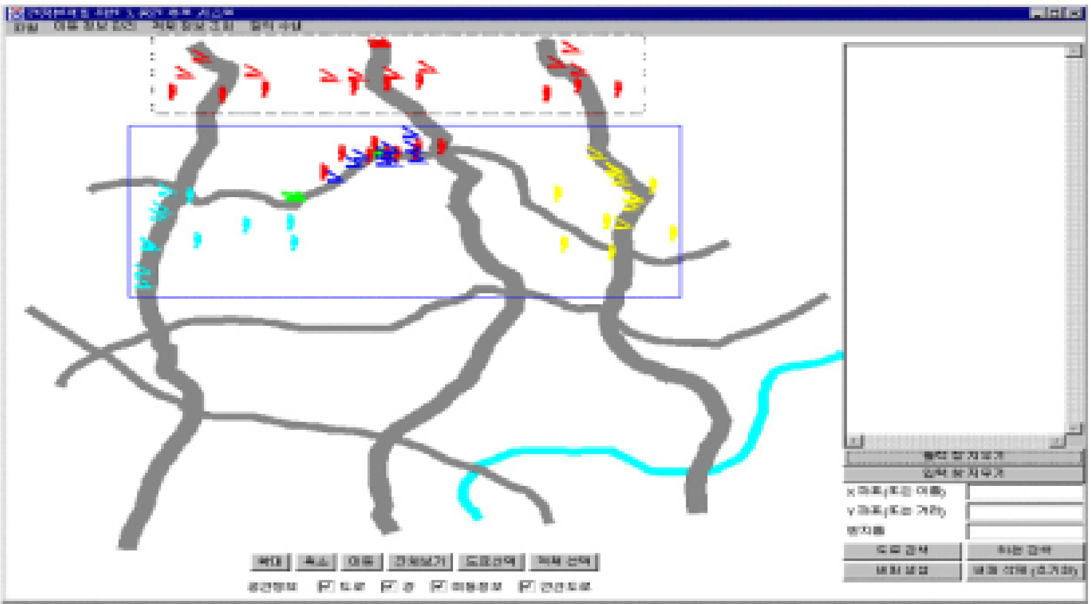

Figure 5.16: Result of the Unit Data Inquiry by Date

Figure 5.16 is the result that unit data inquiry by date was performed. After performing the query, the result that can be provided to user is two kinds.

First one is initial location screen of corresponding unit. Initial locations of units are displayed with dotted line on the upper side of the screen. Initial locations are the position information that was observed at the beginning. Location information observed after this is called motion information.

Second is location information of unit which corresponds to inputted date to query and is depicted with solid box. Figure 5.16 outputs more units than the number of initial units, since all unit's locations from May 5, 2000 to May 8, 2000 are come out. And, we indicate regiments with different colors for user to see assigned unit distinctively.

4. Adjacent Object Search

   Adjacent object search is the function of searching unit or road data to which a unit is adjacent. Query execution of searching adjacent object proceeds with two steps.

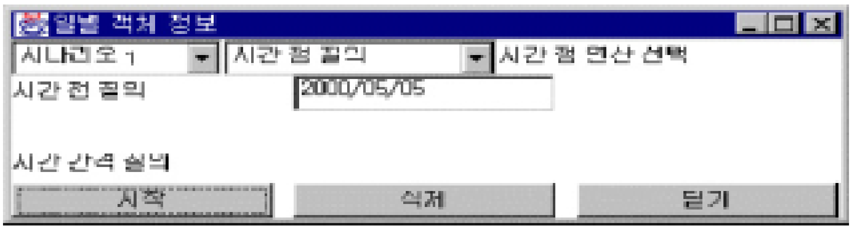

Figure 5.17: Making a query of Unit Data Inquiry by Date

First one is to inquiry object data by date for desired date and the second choose the adjacent relation query area and performs the query. First query execution searches the object data by date for desired date, and using query is as follows.

- Query : "Inquiry all the unit information that motion information was observed on May 5, 2000 for scenario 1."

Figure 5.17 is a query input window that inquiries object data by date on May 5, 2000 for scenario 1, and the result is as follows.

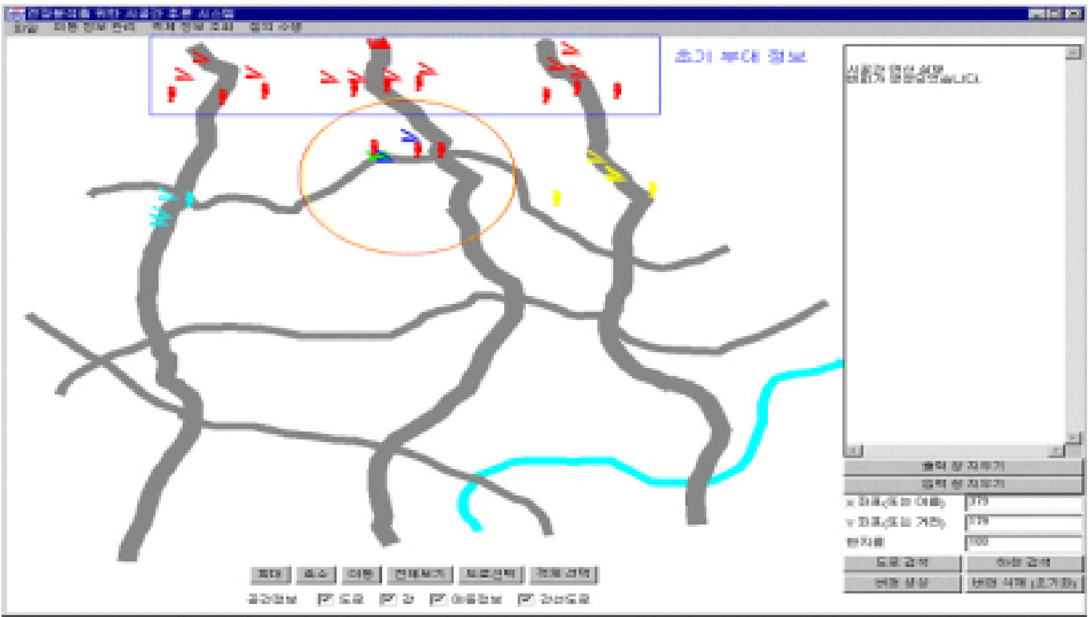

Figure 5.18: Result of the Unit Data Inquiry by Date

The box in figure 5.18 describes initial location data of units, and the circle shows query area that user creates. It says that query area is a buffer area which is inputted and created location coordinate and radius for a certain point to search close object. Second, we perform desired search for adjacent object data by using query input window in figure 5.19.

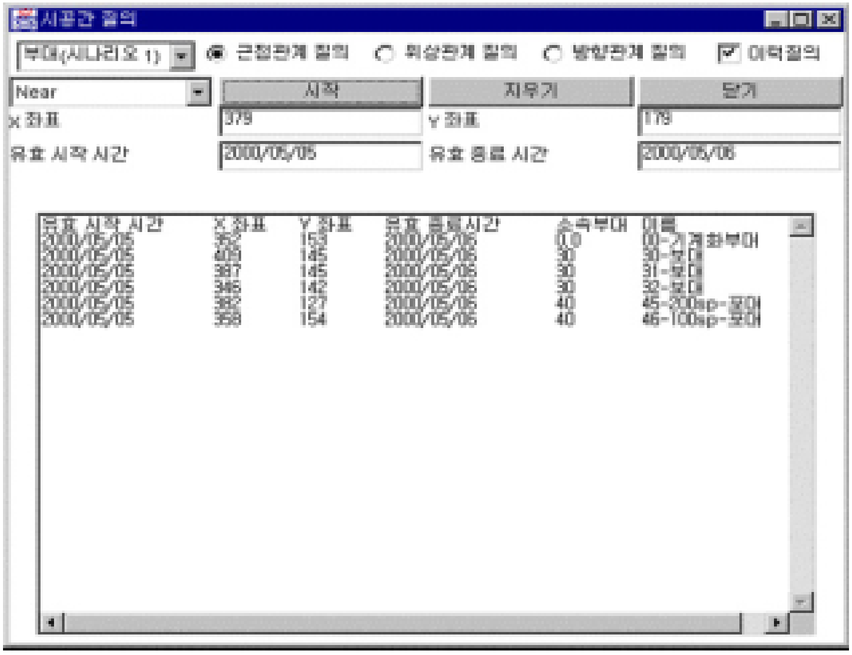

Figure 5.19: Making a query and the result of Adjacent Object Search

Figure 5.19 is a query and result screen for searching units near to coordinate(379, 179) from May 5, 2000 to May 6, 2000 for scenario 1. As this figure, adjacent object search outputs name, assigned unit, valid time, and location coordinate for the units existing in query area.

5. Unknown Unit Inference

   Unknown unit inference is the function of reasoning unit's name and assignment data, when observation date and location coordinate among acquired motion information are correct, but unit's name and assignment data aren't. Unknown unit inference is performed by inputting desired date and choosing query menu. Following figure 5.20 is the query input window that inquiries unknown unit inference and using query is as follows.

   - Query : "Infer unknown unit from May 9, 2000 to May 10, 2000 for scenario 1."

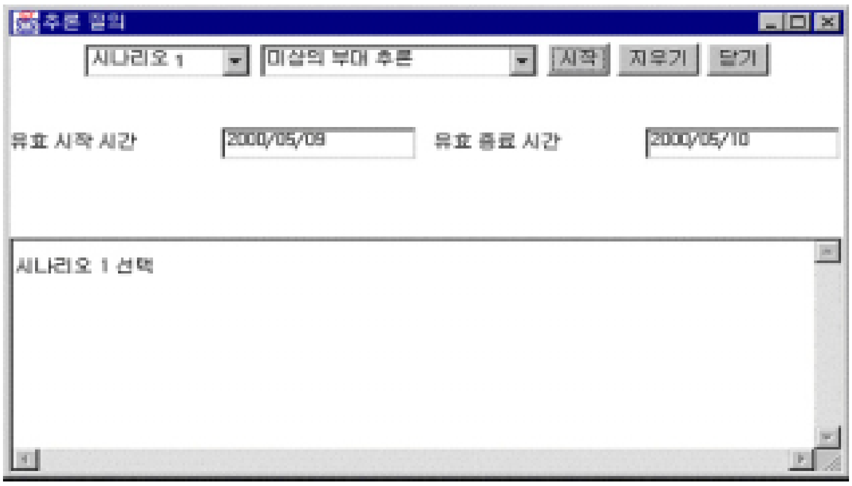

Figure 5.20: Making a query of Unknown Unit Inference

Figure 5.20 is a query input window that performs unknown unit inference from May 9, 2000 to May 10, 2000 for scenario 1, and the result is as follows.

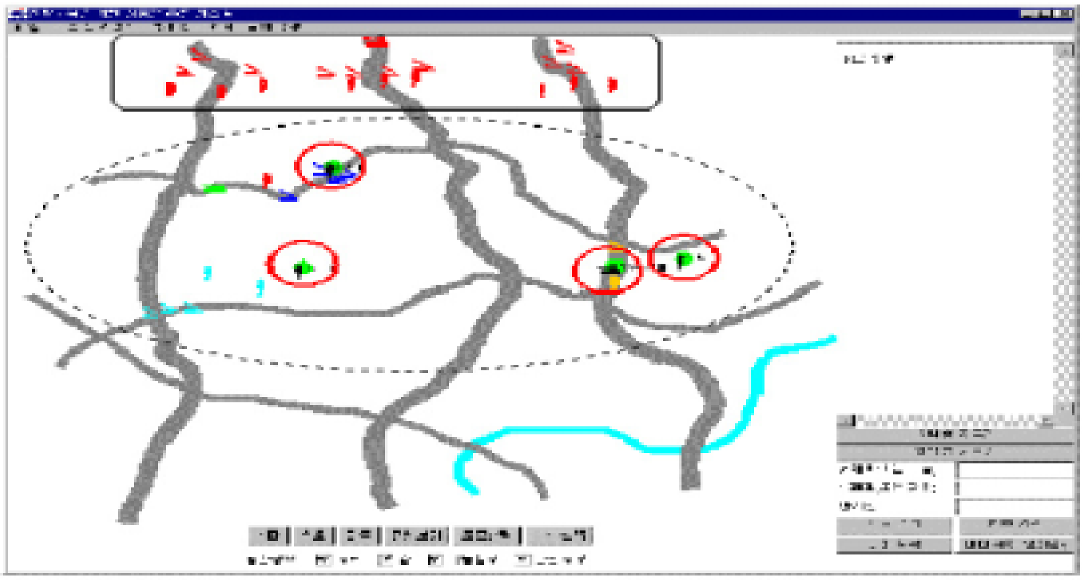

Figure 5.21: Result of the Unknown Unit Inference

Figure 5.21 is the result of the unknown unit inference. The box in upper part of this figure describes initial location data of units, and the units expressed in the dotted big circle below the box shows object data by date from May 9, 2000 to May 10, 2000. The encircled part by small circles depicted with solid line is the inferred result as unknown units.

6. Unidentified Unit Inference

   Unidentified unit inference is the function of reasoning name and location coordinate of unit which has absolutely no acquired motion information. Figure 5.22 is a query window which infers unidentified unit and using query is as follows.

   - Query : ”Infer unidentified unit from May 9, 2000 to May 10, 2000 for scenario 1.”

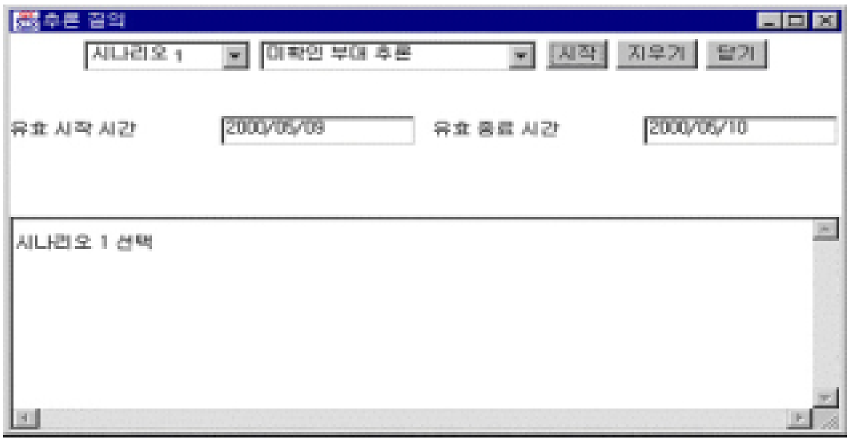

Figure 5.22: Making a query of Unidentified Unit Inference

Figure 5.22 is a query input window that performs unidentified unit inference from May 9, 2000 to May 10, 2000 for scenario 1, and the result is as figure 5.23.

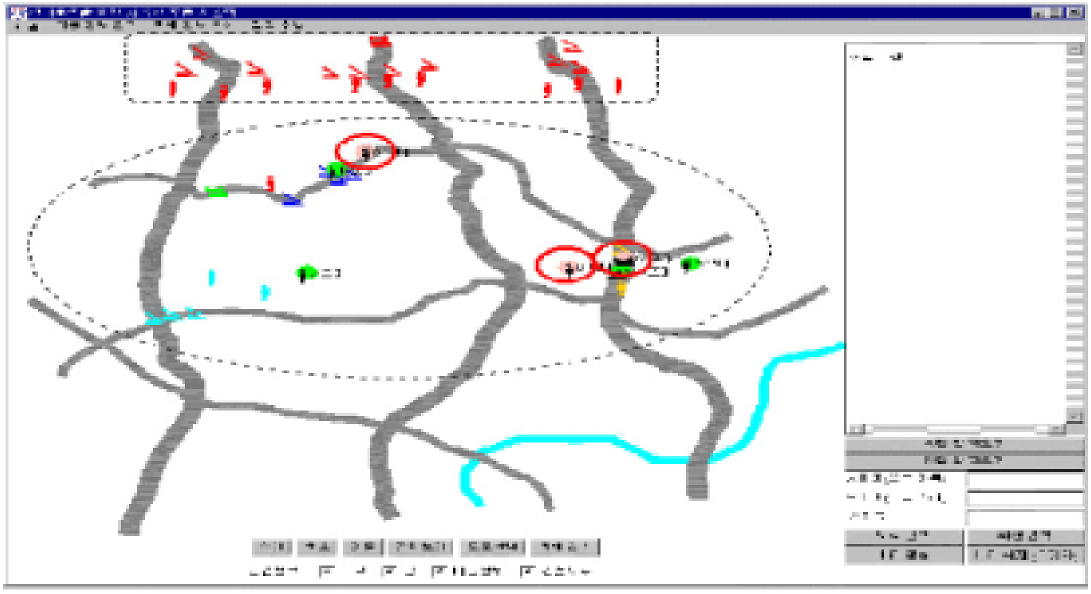

Figure 5.23: Result of the Unidentified Unit Inference

Figure 5.23 is the result of the unidentified unit inference. The box in upper part of this figure describes initial location data of units, and the information expressed in the dotted big circle below the box shows unit's location data from May 9, 2000 to May 10, 2000. The units inside small circles with solid line are the inferred result as unidentified units. The locations of inferred units as unidentified units can have difference from real information, because the unit location is computed by the inference using motion information for the units.

7. Main Strike Direction Reasoning (After Assembled Disposition)

   Main strike direction inference is the function of reasoning major unit's moving direction in the future by using unit's motion information until present and other former knowledge. It can perform the query on the move or after assembled disposition. Here, the experiment will present the result of performing main strike direction inference after assembled disposition.

   Main strike direction inference after assembled disposition is divided into two cases, to apply default weight and for user to apply arbitrary weight. However, main strike direction after assembled disposition is determined by considering location of mechanized unit, formation of forward regiment, and direction of rear area regiment. Applying default weight is to apply no weights to three considerations, and applying user weight is that user arbitrarily applies weights to three considerations.

   (a) Applying Default Weight

      The main strike direction inference query using default weight which user applies no weight is made as follows.

      - Query : "In case of default weight, infer main strike direction after assembled disposition from May 11, 2000 to May 12, 2000 for scenario 2."

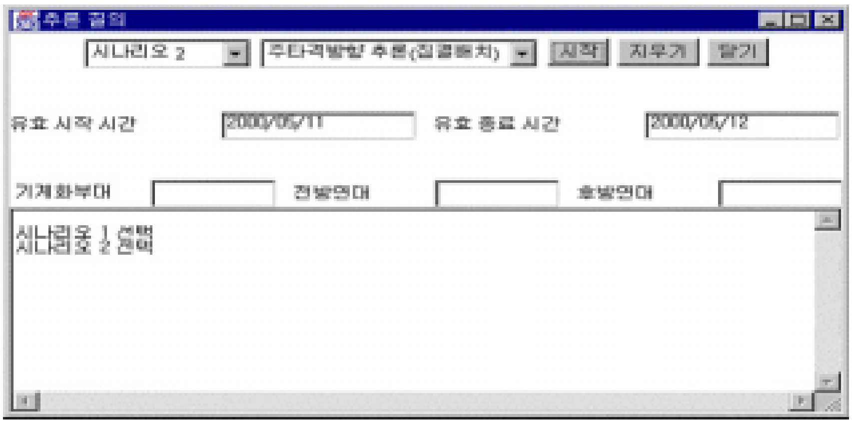

Figure 5.24: Making a query of Main Direction Inference
after Assembled Disposition(default weight)

Figure 5.24 is a query window that infers main strike direction from May 11, 2000 to May 12, 2000 for scenario 2. In querying for main strike direction inference, we can give separate weight values to three considerations. This query is a screen to show the result for giving no weights, and the result is as follows.

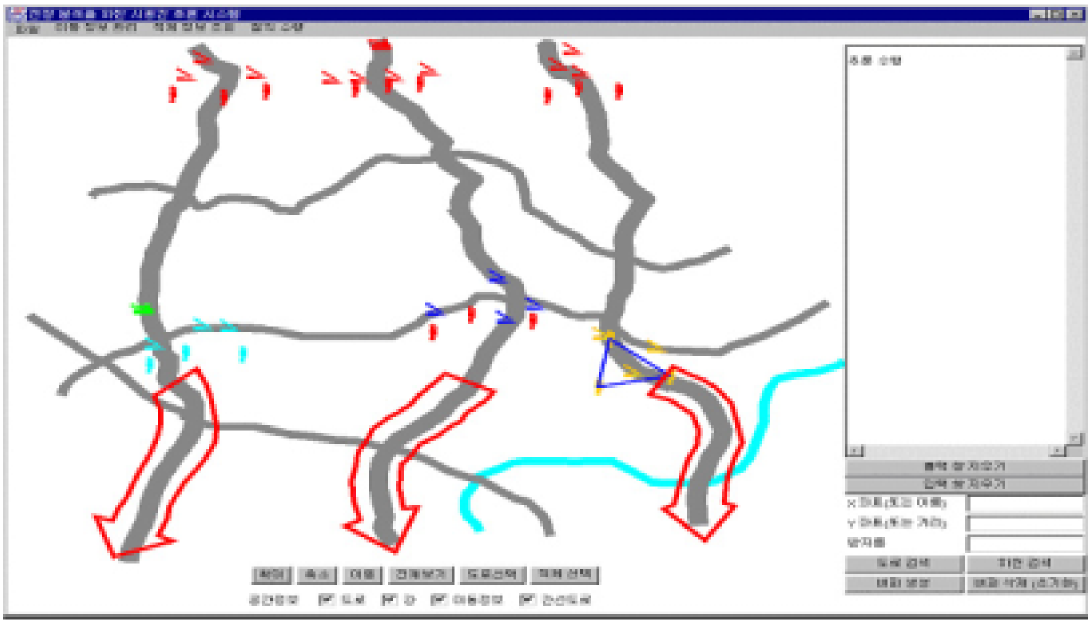

Figure 5.25: Result of the Main Direction Inference
after Assembled Disposition(default weight)

Above figure 5.25 is the result screen of inferring main strike direction after assembled disposition in case of giving no weights. If user do not give any weight, the arrows showing main strike direction would be come out in different three directions with each other since the same weight is automatically given to three considerations by the system.

(b) Applying User Weight

The main strike direction after assembled disposition that user arbitrarily applies weights is made as follows.

- Query : "In case of user weight, infer main strike direction after assembled disposition from May 11, 2000 to May 12, 2000 for scenario 2.Only, the weight of mechanized unit is 10%, forward regiment is 30%, and rear area regiment is 60%."

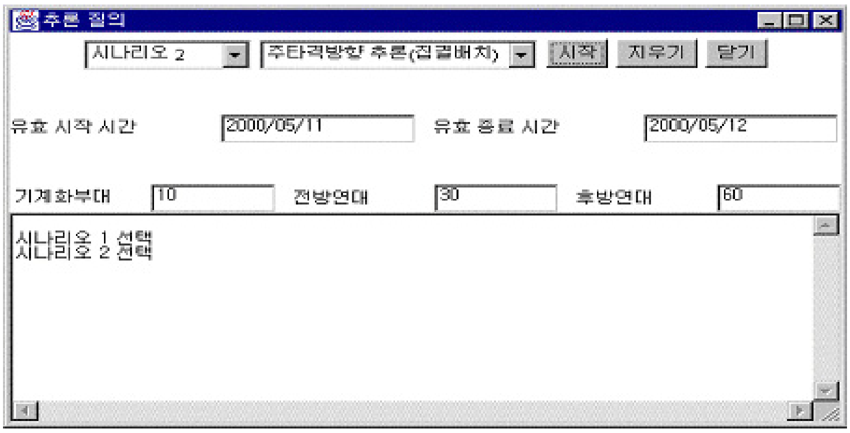

Figure 5.26: Making a query of Main Direction Inference
after Assembled Disposition(user weight)

Figure 5.26 is the query that infers main strike direction from May 11, 2000 to May 12, 2000 for scenario 2. At this time, user queries by giving different weights with each other for three considerations, and the result applying 10% to weight of mechanized unit, 30% to weight of forward regiment, and 60% to weight of rear area regiment is as following figure 5.27.

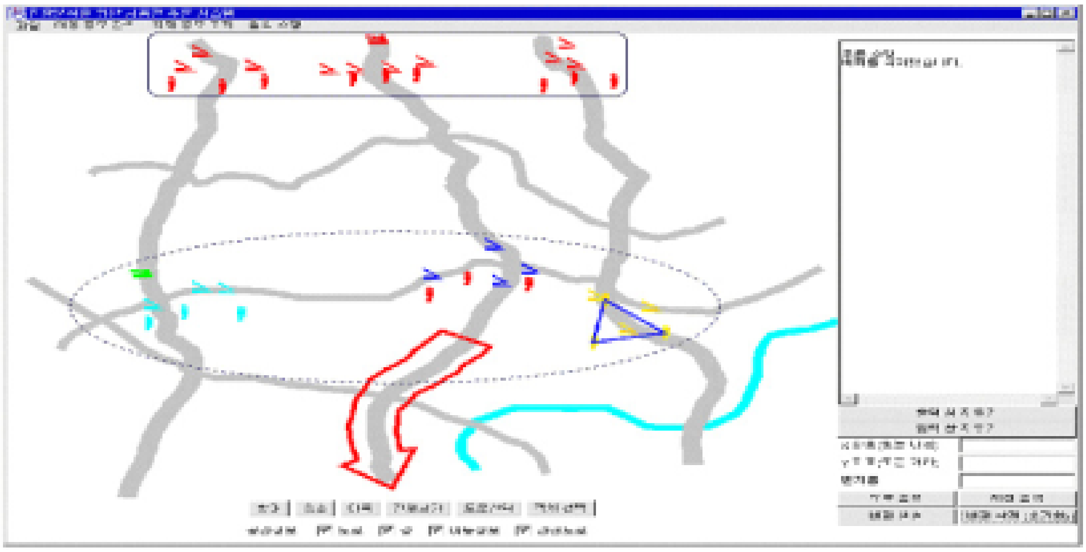

Figure 5.27: Result of the Main Direction Inference
after Assembled Disposition(user weight)

Figure 5.27 is the result window for the main strike direction query in case of applying different weights with each other. At this time, the system will infer the main strike direction and show the inferred result on the screen by using the item that the most weight value is given. The box depicted with solid line in figure 5.27 describes initial location data of units, and the unit information inside circle with dotted line shows location data from May 11, 2000 to May 12, 2000. And, after searching the location of rear area regiment given 60% weight, the arrow part shows the result of establishing main strike direction along with the road closest to that location.

8. Unit's Moving Location Inference

   Unit's moving location inference predicts location coordinate where a unit will move in the future and prints out the result on the screen. The moving location inference is needed for all general spatiotemporal reasoning system using moving objects. Following 5.28 is a query window that infers where a unit will be located on what date in future. And using query is as follows.

- Query : ”Infer next position for 11 infantry battalion in scenario 1 to move from May 13, 2000 to May 15, 2000.”

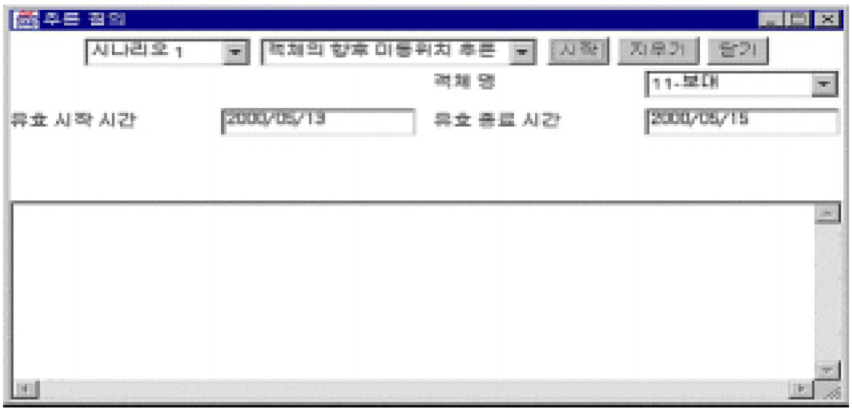

Figure 5.28: Making a query of Future Moving Location Inference

Figure 5.28 is a query window that infers next moving location from May 13, 2000 to May 15, 2000 for 11-infantry battalion in scenario 1, and the result is as follows.

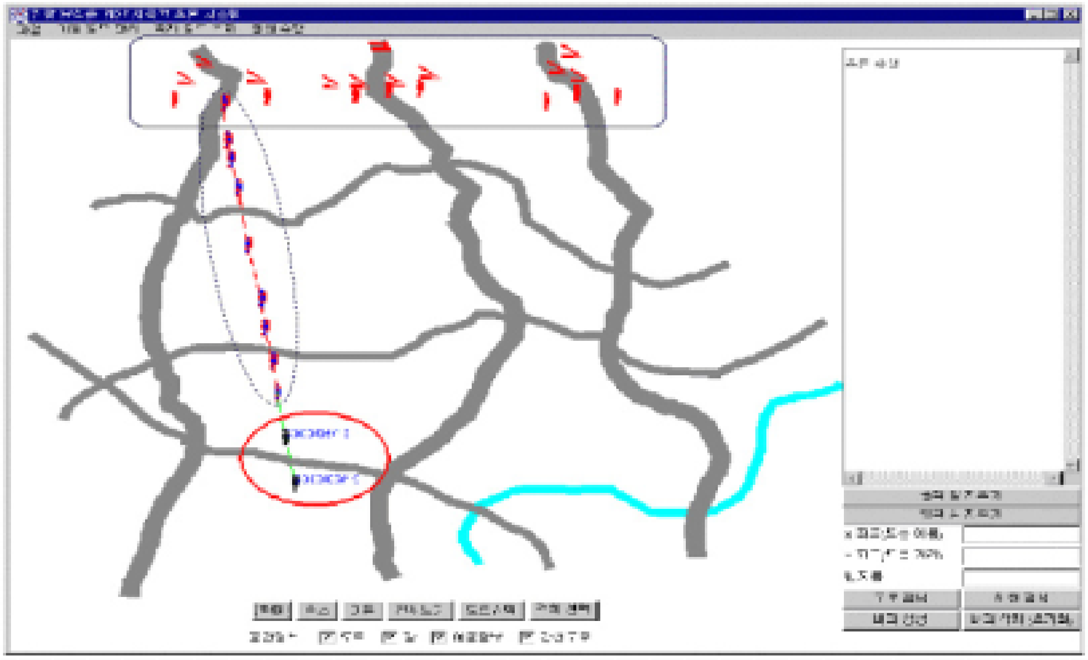

Figure 5.29: Result of the Future Moving Location Inference

Figure 5.29 is the result of the future moving location inference. The box depicted with solid line describes initial location data of units, and the ellipse with dotted line shows the route of unit’s movement from initial to current location. And the solid circle indicates next moving location and expected movement route for the unit from May 13, 2000 to May 15, 2000.

9. Unit’s Moving Time Inference Unit’s moving time inference will offer expected moving time required for a unit to arrive at certain location, after user input future location coordinate for a unit. Next screen is a query window that infers moving time required for a unit to arrive at certain location.

   - Query : ”Infer expected time for 22-infantry battalion in scenario 1 to move from current location to location (620, 520).”

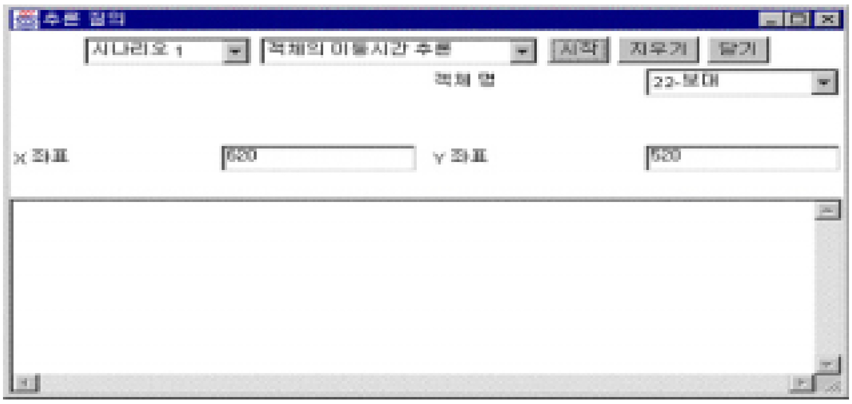

Figure 5.30: Making a query of Moving Time Inference

Figure 5.30 shows a query of inferring required time for 22-infantry battalion in scenario 1 to move from current location to location (620, 520), and the result is as following figure 5.31.

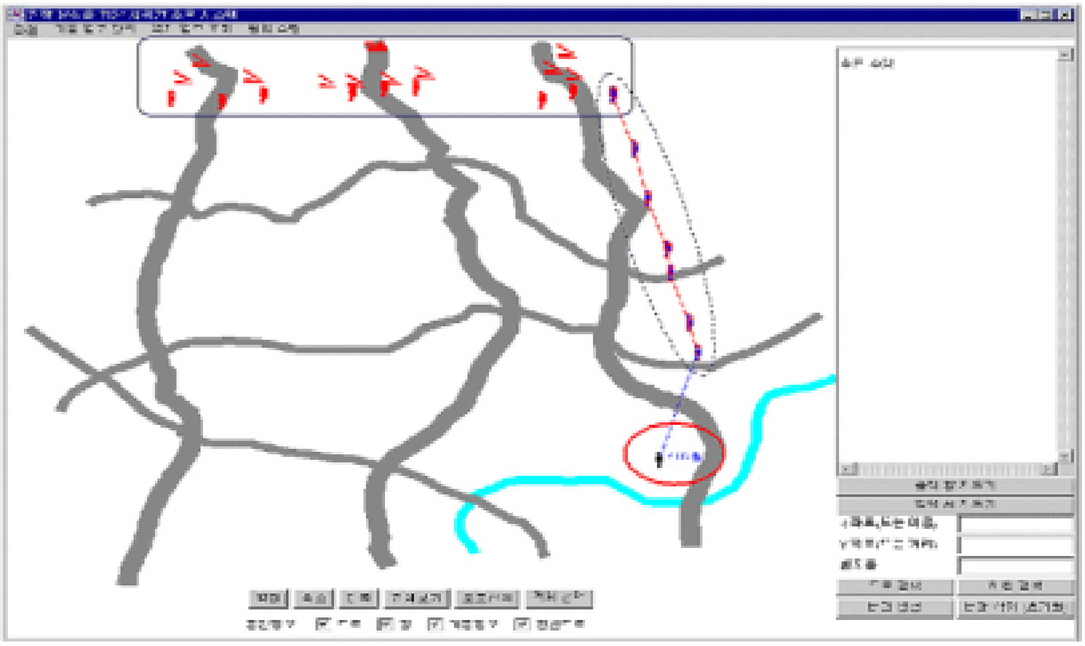

Figure 5.31: Result of the Moving Time Inference

Figure 5.31 is the result of inferring required moving time for 22-infantry battalion in scenario 1 to move to future location (620, 520). The box depicted with solid line describes initial location data of units, and the ellipse with dotted line shows the route of unit's movement from initial to current location. And the solid circle indicates expected moving time for the unit which will exist at location (620, 520).

6 Conclusion

The separate studies about temporal reasoning and spatial reasoning have been long, and have obtained a great amount of theoretical outcome. But, the studies about spatiotemporal reasoning are not very much proceeded. Especially, research and development on spatiotemporal reasoning application area using moving object, which changes position and shape of spatial objects, is hardly progressed. By the way, applications like navigational system, war game model, and close combat simulation, and battlefield analysis system mainly use moving objects such as vehicle, tank, ship, and unit. They requires simple fact data of moving object as well as spatiotemporal reasoning function to solve the specific problem.

Therefore, theoretical study and developing application system about spatiotemporal reasoning integrated temporal and spatial reasoning are required. This study has proposed spatiotemporal reasoning system model using moving object by expanding basic theories about temporal and spatial reasoning to spatiotemporal reasoning. Moreover, to reduce the interval between theoretical study and applicative study about spatiotemporal reasoning, we have constructed a prototype of Battlefield Analysis Spatiotemporal Reasoning System, based on the proposed spatiotemporal reasoning system model.

The spatiotemporal reasoning system model proposed in this study is made up of user interface, spatiotemporal processor, inference engine, GIS system, knowledgebase, and spatiotemporal database. User requests general spatiotemporal query and reasoning query to the system and confirms the result of requested query through the user interface. Spatiotemporal processor manages spatiotemporal database and processes spatiotemporal operations. Inference engine infers location and properties of spatiotemporal data using the knowledge base and database. GIS system is a spatial data management system used to build spatiotemporal database. Knowledge base is a storage in which domain dependent knowledge and rules used to infer are stored. Spatiotemporal database stores general, temporal, and spatial attributes of moving objects. Also, motion information is stored in the history database changed over time.

In order to proper use the proposed spatiotemporal reasoning system model, we have designed and implemented a prototype of spatiotemporal reasoning system for battlefield analysis by using spatiotemporal motion information related to battlefield analysis as a specific application. And, to show systematic function of implemented system, we experimented spatiotemporal operation and reasoning function by building virtual scenarios related to battlefield analysis. As the result of experimentation, the proposed spatiotemporal reasoning system model is indicated that it can be applied to several applications and can be actually developed, and can be greatly utilized to develop the applications like navigational system, war game model, and close combat simulation, and battlefield analysis system mainly use moving objects.

The spatiotemporal reasoning system implemented in this study has four features as follows. First, it manages history of valid time and motion information for moving objects by using spatiotemporal database. Second, it performs spatiotemporal reasoning by introducing spatiotemporal operators to reasoning process. Third, it stores knowledgebase with domain knowledges, which are related to time and space of moving objects, and applies them to reasoning. Fourth, it utilizes GIS tools to build the spatiotemporal database.

Imperfection from this study is that we cannot use the method of processing uncertainty of spatiotemporal data in actual reasoning process, since uncertainty characteristic of spatiotemporal moving objects was not applied to actual system development. In general, reasoning system can not guarantee with 100% accuracy for inferred result, even though data and knowledge used in inference engine are wholly correct. Since the data used in this study have its own uncertainty factor, inferred result have occurred a little error. From now on, various processing methods to minimize uncertainty of spatiotemporal data applied to spatiotemporal reasoning must be worked, and the study raising accuracy by applying to reasoning process at actual system development also has to be progressed.

Acknowledgments

This rearch was funded by Agency for Defense Development and KOSEF(Korea Science and Engineering Foundation.)

References

[Abra96] T.Abraham and J.F.Roddick, "Survey of Spatio-temporal Databases", Research Report CIS-96-011, School of Computer and Information Science, University of South Australia, 1996.

[Alle83] J.F.Allen, "Maintaining Knowledge about temporal intervals", Communication of the ACM, Vol. 26, Nov. 1983.

[Arms88] M.P.Armstrong, "Temporality in Spatial Databases", Proceedings: GIS/LIS, 1988.

[Dutt91] Soumitra Dutta, ”Topological Constraints : A Representational Framework For Approximiate Spatial And Temporal Reasoning”, Second International Symposium on Large Spatial Databases, SSD ’91, Zurich, Switzerland, 1991.

[Dutt94] S. Dutta, ”Knowledge Processing & Applied Artificial Intelligence”, Butterworth Heineman Publisher, PP.31-54, 1994.

[Erwi99] M. Erwig, R.H. Guting, M. Schneider, and M. Vazirgiannis, ”Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases”, GeoInformatica 3:3 (1999), pp. 269-296.

[Gere93] A. Gerevini and L. Schubert, Efficent temporal reasoning through timegraphs. In Proceedings of the Thirteenth IJCAI, page 648-654, Chambery, Savoie, France, 1993.

[Grig95] Grigni, M. Papadias, D., Papadimitriou, C. ”Topological Inference”, Proceedings of the International Joint Conference of Artificial Intelligence(IJCAI), Montreal, Canada, AAAIPress, 1995.

[Guti2000] R.H.Guting, M.H.Bohlen, M.Erwig, C.S.Jensen, N.A.Lorentzos, M.Schneider, and M.Varzirgiannis, ”A Foundation for Representing and Querying Moving Objects”, ACM Transactions on Database Systems, Vol.25, No.1, Mar. 2000, pp.1-42.

[Hong95] J.-H. Hong, M. Egenhofer, and A. Frank, ”On the Robustness of Qualitative Distance and Direction-Reasoning”, D. Peuquet (Ed.), Autocarto 12, Charlotte, NC, February 1995.

[Horn97] Kathleen Hornsby and Max J. Egenhofer, ”Qualitative Representation of Change”, 1997.

[Lang88] G.Langran, ”A Framework for Temporal Geographic Information Systems”, Cartographica, 25(3), 1998.

[Mont93] A. Montani and B. Pernici, ”Temporal Reasoning”, In Temporal Databases: Theory, Design and Implementation, The Benjamin Cummings Publishing Company Inc., 1993.

[Papa94] Dimitris Papadias and Yannis Theodoridis, ”Spatial Relations, Minimum Bounding Rectangles, and Spatial Data Structures”, TR KDBSLAB-TR-94-04, Dept. of Computer Science and Engineering, Univ. of California, San Diego, 1994.

[Papa95] Papadias, D., Theodoridis, Y., Sellis, T., Egenhofer, M. ”Topological Relations in the World of Minimum Bounding Rectangles: a Study with R-trees”, Proceedings of the ACM Conference on the Management of Data(SIGMOD), San Jose, CA, ACM Press, 1995.

[Park2001] Park S.S., Ahn Y.A. and Ryu K.H., ”Moving Objects Spatiotemporal Reasoning Model for Battlefield Analysis”, Proceedings of Military, Government and Aerospace Simulation part of ASTC 2001, pp.108-113.

[Perk90] W. Perkins and A. Austin, ”Adding Temporal Reasoning to Expert System Building Environments”, IEEE Expert, Vol.5, No.1, pp.23-30, Feb., 1990.

[Peuq95] Donna J. Peuquet and Miu Duan, ”An event-based spatiotemporal data model(ESTDM) for temporal analysis of geographic data”, International Jounal of Geographic Information Systems, 9(1), 1995.

[Reno97] A. Renolen, ”The Spatio-temporal Object Model: A Conceptual Model Based on Ontology”, article submitted to Geoinformatica, 1997.

[Sege93] A. Segev and A. Shoshani, ”A Temporal Data Model Based on Time Sequences”, In Temporal Databases: Theory, Design, and Implementation, The Benjamin/Cummings Publishing Company Inc., 1993.

[Sist98] P. Sistla, O. Wolfson, S. Chamberlain, S. Dao, ”Querying the Uncertain Position of Moving Objects”, Springer Verlag Lecture Notes in Computer Science number 1399, 1988, pp. 310-337.

[Stef95] Mark Stefik, ”Introduction to Knowledge Systems”, Morgan Kaufmann Publishers, Inc., 1995.

[Vila94] L. Vila, ”A Survey on Temporal Reasoning in Artificial Intelligence”, AI communication, Vol.7, Num.1, Mar., 1994.

[Wolf98] O. Wolfson, B. Xu, S. Chamberlain, L. Jiang, ”Moving Objects Databases: Issues and Solutions”, Proceedings of the 10th International Conference on Scientific and Statistical Database Management (SSDBM98), Capri, Italy, July 1-3, 1998, pp. 111-122.

[Worb90] M. F. Worboys, H. M. Hearchshow and D. J. Maguire, ”Object-Oriented Modelling for Spatial Databases”, Int. journal of GIS, 4(4), 1990.

[Yuan94] M. Yuan, ”Wildfire Conceptual Modeling for Building GIS Space-Time models”, Proceedings: GIS/LIS’94, 1994.

[Yuan96] M. Yuan, ”Modeling Semantical, Temporal and Spatial Information in Geographic information systems”, M. Craglia and H. Couclelis eds. Geographic Information Research: Bridging the Atlantic, Taylor&Francis, 1996.

[Yuan97] M. Yuan, ”Use of a Three-Domain Representation to Enhance GIS Support for Complex Spatio-Temporal Queries”, In Press by Transactions in GIS, 1997.