Moving objects Spatiotemporal Reasoning Model for Battlefield Analysis

GUNWOO PARK, SEONG SEUNG PARK

Department of Computer Science and Information,
Korea National Defense University,
SOUTH KOREA

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 were 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 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, proper domain is established for battlefield analysis system supporting commander’s decision making in the army operational situation and it is experimented with this domain.

1. Introduction

We call Intelligence Preparation of the Battlefield or Battlefield Analysis for evaluation and analysis of the battlefield area, based on data for terrain, climate, and unit’s maneuver and tactics basically required in battlefield simulation. For the battlefield analysis, it is required to get correct information about the identification and moving status of target units. However, since perfectly collecting all the information is actually impossible, we need spatiotemporal reasoning function that predicts and analyzes future moving status for target units by using collected information and related knowledge. Especially, moving units have a characteristic of moving objects (Erwig et al. 1997) which change its position and shape over time, and such spatiotemporal data having temporal and spatial attributes require spatiotemporal reasoning function to manage moving objects.

Observing the concepts of temporal, spatial, and spatiotemporal reasoning studied up to the present, temporal reasoning (Dutta 1991, Montanari and Pernici 1993, Vila 1994) consists of formalizing the notion of time by reasoning for various characteristics of temporal data and representing and reasoning about the temporal aspects of knowledge. Spatial reasoning (Dutta 1991, Grigni, Papadias, and Papadimitriou 1995) is an inference technique that works out several spatial problems based on the topological, directional, and distance relationship between spatial objects. Spatiotemporal reasoning (Dutta 1991, Ryu 2000) performs temporal, spatial, and spatiotemporal reasoning functions in all, integrating separate temporal or spatial reasoning by one. However, because existing temporal and spatial reasoning research has considered temporal and spatial attributes separately, spatiotemporal models that are adequate to manage and infer spatiotemporal data have not been proposed. Especially, research and development on spatiotemporal reasoning application area using moving objects (Guting 1998), which changes location and shape of spatial objects over time, is hardly progressed. By this reason, it is required to establish spatiotemporal reasoning model that is applicable to moving object application, and to implement an application system.

Therefore, this paper proposes a new type of spatiotemporal reasoning model considering spatiotemporal database for moving objects, inference engine, and GIS tool, based on the spatiotemporal reasoning concept studied until now. And, we apply
proposed model to battlefield analysis system and present experimentation results. Proposed spatiotemporal reasoning model has following characteristics. First, it uses spatiotemporal database that has not been considered in existing spatiotemporal reasoning study. Spatiotemporal database does not require storing topological and directional relations of spatial objects to knowledgebase in advance, because it can manage histories of moving objects and perform spatiotemporal operation for moving objects. Second, it can do dynamic spatiotemporal reasoning by using spatiotemporal fact data, which resulted from spatiotemporal operation and moving object operation, in the process of inference engine. Third, it can make us exactly identify movement of moving objects on the map screen using GIS system, and develop versatile spatiotemporal operators by performing spatiotemporal operation expanding time dimension to spatial operation given by GIS system.

This paper proceeds as following. Chapter 2 explains why existing temporal and spatial reasoning is not suitable for moving object reasoning, based on the study related on temporal and spatial reasoning. And, chapter 3 introduces moving objects and shows a moving objects data model for battlefield analysis, which becomes the target of spatiotemporal reasoning. In chapter 4, we describes the structure of spatiotemporal reasoning system for battlefield analysis and the procedure of inference engine on the basis of moving objects data model given in chapter 3. Chapter 5 presents implementation of the proposed model and experimentation using scenario, and analyzes the characteristics of this model. Finally, chapter 6 concludes this study and presents future directions of study.

2. Related Studies

2.1 Temporal Reasoning

Study about temporal reasoning includes mathematical approach that allows us to express time extending logics and the approach that applies temporal reasoning function to expert system. First, in the mathematical approach (Montanari and Pernici 1993), temporal reasoning is classified with reasoning about temporal structures and derivation from incomplete information according to modeling mechanisms of temporal data. Reasoning about temporal structures is performed using two basic algorithms for constraint-satisfaction problem solving, namely simplification and propagation. Simplification returns the minimal equivalent subset of a given set of constraints on a finite set of temporal variables. Propagation makes the deductive closure of a set of constraints, looking for all constraints that can be derived from the originally given ones. Derivation from incomplete information is performed using incomplete knowledge and different temporal granularity. The former is performed by means of forward temporal reasoning and backward temporal reasoning, and the latter is performed by means of default projection rules, downward temporal projection, and upward temporal projection.

Next, the approach that applies temporal reasoning function to expert system assumes the form of adding temporal reasoning to expert-system-building environments (Perkins and Austin 1990), and it is called Temporal Reasoning Expert System. Looking at the related studies of temporal reasoning expert system, there are some examples of aerospace applications like Wheels, REX, and NEO/Temporal. Wheels is an expert system that monitors and diagnoses problems with the Hubble space telescope’s reaction wheels (Perkins and Austin 1990). REX is an object-oriented expert system tool (Prasad et al. 1994). And NEO/Temporal is a temporal inference engine implemented by using event-based temporal reasoning method and inference engine NEO (Lee et al. 1997). Particularly, NEO/Temporal is closely related with battlefield analysis that this paper intends to apply, and is applied to a situation assessment and decision supporting system for air operation. However, it performs the inference for only such situation that a hostile plane infiltrates and moves along the routes inputted in advance, accordingly it cannot support proper inference for the situation that route of object dynamically changes.

2.2 Spatial Reasoning

Spatial reasoning has been researched about reasoning of topological and direction relations in order to mainly manage mutual relationship between spatial objects based on spatial domain established in spatial data modeling. First of all, topological relationship reasoning (Grigni, Papadias, and Papadimitriou 1995) is the inference technique using topological relations of spatial data. Topological relation between object X and Y in space can be represented by the intersection for boundary, inside, and outside. In general, only eight topological relations become to make sense such as equal, meets, overlap, contains, disjoint, properly-contains, contained-in, properly-contained-in. When there are spatial objects X, Y, and Z, if relation of X and Y, and relation of Y and Z are R1 and R2, we can derive relation R3 of X and Z using these eight topological relations. 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 derive new relation R3 of X and Z with these two 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 \((X \ R1 \ Y) \land (Y \ R2 \ Z) \rightarrow (X \ R3 \ Z)\).

Direction relationship reasoning (Hong, Egenhofer, and Frank 1995) is concerned with direction relations being a special class of spatial relations that deal with order in space. There are cone-shaped approach and projection-based approach in defining direction relations. First, direction relations in the cone-shaped approach are defined using angular regions between objects. Next, in the projection-based approach, direction relations are defined using projection lines vertical to the coordinate axes. 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). Especially, reasoning about direction and distance relation mainly qualitative. Because people can differently recognize such relations as directions of east, west, south, and north, and as distances of near and far. Separate study of topological and directional reasoning done until now cannot effectively support location inferences of present, past, and future for moving objects in space.

2.3 Spatiotemporal Reasoning

Spatiotemporal reasoning (Egenhofer 1998) is probably the most common and basic form of human intelligence. People from birth employ methods of spatial reasoning continuously to infer information about their environment, how it evolves over time, and how we change our location in space. Spatiotemporal reasoning is so common in our daily life that we rarely notice it as a particular concept of spatial analysis. When applied to computerized information systems, spatiotemporal reasoning is mainly used to solve problems that deal with spatial objects changing over time.

Studies about spatiotemporal reasoning consist of "A representational framework for approximate spatial and temporal reasoning" (Dutta 1991) and "spatial and temporal reasoning in geographic information systems" (Egenhofer 1998). Dutta presented a uniform representation schema for both spatial and temporal concepts as topological constraints. Specially, he provided a mathematical basis for representing imprecision and uncertainty by using fuzzy logic. Due to the imprecision and uncertainty, temporal and spatial reasoning becomes to use approximate reasoning. Spatial and temporal reasoning in geographic information systems is classified into spatiotemporal reasoning in computational science perspectives, spatial and temporal cognition, and spatial and temporal behaviors in social science contexts.

These applications using spatiotemporal reasoning (Dutta 1991) are all related to both applications of spatiotemporal database and expert system. Related application areas worked so far are not very much. However, similar applications include metabolic diagnosis system, stock investment expert system, and simulator for predicting the result of professional baseball game. Moreover, applications using moving object contain navigational systems, war game models, close combat simulations, and intelligence preparation of the battlefield. However, there is hardly application system about spatiotemporal reasoning for moving objects so far.

3. Moving objects

3.1 Data types and Operations

Moving objects (Erwig et al. 1997) are geometry data, of which shape and general attribute are changing over time, and spatiotemporal data, having the characteristic that spatial object change its position and shape with move. Moving objects have two basic types, which are moving points and moving regions. Moving points are positions or locations of object changing over time. They are people, animals, cars, and tanks, and express the value of spatial point at specific time such as "Mpoint : Time → Point".

Moving regions are shapes as well as positions of objects changing over time. Also, regions can be grown or reduced. Data like glaciers, storms, and cancer belong to moving regions, and express the value of spatial region at specific time such as "Mregion : Time → Region".

Basically, we must have valid time value and spatial coordinate value to execute operations for moving objects, and store them as a format of spatiotemporal databases, representing location of a specific object on time T by using identifier of the object(Oid), location coordinate(X,Y), and time(T) data. Moving object operators studied until now are MDistance, Trajectory, Visits, Length, Minvalue, Maxvalue, Mintime, Maxtime, Velocity, and Attime. However, only five operators, which will be used in this paper such as MDistance, Trajectory, Length, Velocity, and Attime, will be described.

First, operator \(MDistance(A,B)\) computes distance between two arbitrary objects A and B that have same valid time. Input values are location coordinate(x,y) and valid time values for objects A and B. If valid time values for objects A and B are equal, we have a following formula:

\[
MDistance = \sqrt{(Oy(B)-Oy(A))^2 + (Ox(B)-Ox(A))^2}
\]

\(Ox(A)\) and \(Oy(A)\) are coordinate x and y for object A, \(Ox(B)\) and \(Oy(B)\) are those for object B. \(Trajectory(A)\) is an operator that traces and extracts the routes of an arbitrary moving object A. Object identifier and arbitrary time values are needed for input data. All the routes that an arbitrary object A
has moved during certain time interval \( <VTs, VTe> \) can be gotten as follows. The equation for a segment of a line using coordinate \( x \) and \( y \) of object \( A \) corresponding to time \( VTs \) and \( VTe \) is:

\[
Y - Oy(VTe) = \frac{Oy(VTs) - Oy(VTe)}{Ox(VTe) - Ox(VTs)} (X - Ox(VTe))
\]

Solving this equation with respect to \( X \) or \( Y \), we can get coordinate \((x,y)\) values for all points between \( VTs \) and \( VTe \), and obtain whole moving routes by connecting them and depicting with a line. \( \text{Length}(A) \) is an operator that returns total length of a moving route for arbitrary object \( A \). It returns the sum of distances of all routes extracted from Trajectory operation for the result. In case object \( A \) moves \( n \) times during time interval \( <VTs, VTe> \), length of the all-moving routes is counted as following formula.

\[
\text{Length} = \text{Dist}(O(A_1)) + \text{Dist}(O(A_2)) + \ldots + \text{Dist}(O(A_{n-1}))
\]

\[
= \sum_{i=1}^{n-1} \text{Dist}(O_i)
\]

\( \text{Dist}(O(A_i)) \) represents moving distance that object \( A \) moves from initial position to second position, and \( \text{Dist}(O(A_{n-1})) \) represents moving distance from \((n-1)\)-th position to \( n \)-th position. \( \text{Velocity}(A) \) is an operator that computes moving speed for arbitrary object \( A \). It returns the ratio of moving distance to total moving time for object \( A \) for the result. Object identifier and time values are needed for input data. Average moving velocity of object \( A \) during certain time interval \( <VTs, VTe> \) is computed by following:

\[
\text{Velocity} = \frac{\text{Length}(O(A))}{VTe - VTs}
\]

\( VTs \) is start time of object move and \( VTe \) is end time of it. \( \text{Length}(O(A)) \) is total distance of moving object \( A \) during \( <VTs, VTe> \). Finally, \( \text{Atttime}(A,t) \) is an operator standing for position of object \( A \) at arbitrary time \( t \). Input data use arbitrary time \( t \) and object identifier. If the time of data having inputted object identifier is equal to \( t \), this operator returns coordinate \((x,y)\) value holding that time value.

3.2 Moving Object Model for Battlefield Analysis

Here, we assume that moving objects for battlefield analysis are moving points only. The objects that can be expressed as moving points are units, tanks, missiles, and people. Also we present location-change status of moving object denoted by discrete model and applied time domain.

**Discrete Model.** For modeling moving objects (Erwig et al. 1997), there are 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 implement 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. Thus, it is very adequate to develop this system. Considering this feature, this paper will be based on discrete model that can actually be implemented.

<table>
<thead>
<tr>
<th>Time domain</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Based</td>
<td>Expresses time point and interval based on time point</td>
</tr>
<tr>
<td>Linear Times</td>
<td>Be formed of set of time points, and can be totally ordered</td>
</tr>
<tr>
<td>Discrete Times</td>
<td>Be represented in terms of Integer number</td>
</tr>
<tr>
<td>Absolute Times</td>
<td>Be based on calendar times</td>
</tr>
<tr>
<td>Temporal Granularity</td>
<td>Expresses level of time element, and differs from characteristic of application system</td>
</tr>
</tbody>
</table>

Figure 1 shows the location changes of a spatiotemporal moving object by using discrete model. In discrete models (Erwig et al. 1997), moving route of the object is described in terms of straight lines. In this figure, the moving object forms three dimensions by extending coordinate \((x, y)\) in two-dimensional space to time-axis. Time values of time-axis may be past, present, and future. In particular, future information is predictable as analyzing past information. Because the greater part of moving objects in real world moves along the movable route, and always moves with similar pattern.

**Temporal Extension of Spatial Attribute.** This paper supposes that moving objects are moving points. Also, this assumption limits spatial attribute of the moving objects to point, line, and polyline. Expanding these spatial attribute to time dimension,
we define spatiotemporal attribute of the moving objects. First, we only consider valid time for temporal attribute of moving objects. Since valid time represents time that attribute of the object is true, transaction time which data are actually stored to database is not under consideration. Time domain for valid time of moving objects is shown in Table 1.

Table 1 Time domain of Moving objects

Valid time is represented by time point or interval. Interval is denoted by a pair of time point, that is, start point and end point like \( <I^- , I^+> \). If there is a relation of \( I^- \prec I^+ \), an interval exists, and if \( I^- = I^+ \), it is a time point. Ordering relationship of valid time can be expressed by the relationship between start point and end point, and forms totally ordered set of time points by linear time domain. For example, when an interval \( I_1 \) is \( <I^-_1 , I^+_1> \) and \( I_2 \) is \( <I^-_2 , I^+_2> \), ordering relation of \( I_1 \) (before) \( I_2 \) that \( I_1 \) precedes \( I_2 \) is denoted by \( I^-_1 < I^-_2 \). Reason is that endpoint of \( I_1 \) is smaller than start point of \( I_2 \). For the operations of moving objects, this paper adopts five operators like MDistance, Trajectory, Length, Velocity, and Attime described in paragraph 3.1.

4. Spatiotemporal reasoning model and application managing moving objects

Spatiotemporal reasoning system (Ryu 2000) is to perform various inference functions for spatiotemporal data changing over time flow. It must have capability of solving the problems of application domain by using spatiotemporal reasoning function as well as managing spatiotemporal data. Reflecting these considerations, we propose a spatiotemporal reasoning system on the basis of moving object data model presented in paragraph 3.2. And, in order to certify the applicability of the proposed model, we will show implementation and experiment on battlefield analysis system that supports commander’s decision making in the army operational situation. Although there are various considerations like climate, terrain, unit’s moving status, and commander’s mental state in the battlefield analysis, here we considers only location information of moving object like unit and tank which will be used in battlefield simulation.

4.1 Function and Type of Spatiotemporal Reasoning

In this paper, spatiotemporal reasoning functions are classified into General Spatiotemporal Reasoning and Domain Dependent Spatiotemporal Reasoning. General spatiotemporal reasoning means future location inference and moving time inference function that is commonly required to spatiotemporal reasoning system using moving objects. Domain dependent spatiotemporal reasoning is the function that a system must provide to solve the problems of specific application domain. And, types of spatiotemporal reasoning are classified into Forward Spatiotemporal Reasoning and Backward Spatiotemporal Reasoning. Forward spatiotemporal reasoning is a prediction technique of future spatiotemporal situation using spatiotemporal knowledge and fact data. Backward spatiotemporal reasoning is to backtrack past fact and to explain validity of present fact in order to verify currently known spatiotemporal fact.

General spatiotemporal reasoning for battlefield analysis consists of Moving Location and Moving Time inference function. And, domain dependent spatiotemporal reasoning includes functions such as Unknown Unit identification, Unidentified Unit estimation, and future Main Strike Direction prediction. Among these, Unidentified Unit estimation, and future Main Strike Direction prediction are forward spatiotemporal reasoning, and Unknown Unit identification is backward spatiotemporal reasoning.

4.2 Structure of the Proposed System

The system model in this study was designed to have following three features to manage moving objects and to perform inference function. First, it manages histories of moving objects by building spatiotemporal database, and makes moving object operations easier. Second, it performs dynamic spatiotemporal reasoning by using spatiotemporal fact data, which resulted from spatiotemporal operation and moving object operation, in the process of inference engine. Third, it utilizes GIS tools to flexibly build a spatiotemporal database.

Figure 2 shows the whole structure of a spatiotemporal reasoning system for moving object management, and it is made up of user interface, moving objects processor, inference engine, knowledgebase, GIS tool, and moving objects spatiotemporal database. User interface furnishes user with GUI environment. Moving objects processor performs operations and searches by using stored data in moving objects database. Inference engine performs inference function by using stored moving object data, domain knowledge, and dynamic spatiotemporal facts processed by moving objects processor. Knowledgebase stores knowledge and facts required to execute inference engine. Moving objects database stores and manages temporal and spatial attributes of moving objects.
4.3 Database Structure of Moving Objects

We construct initial unit data table and unit moving information history table. The former stores initial location information of the object to manage moving objects and latter stores all location changes of moving objects.

4.4 Knowledge Base

**Domain Rulebase.** Knowledge Base is composed of rulebase and dynamic spatiotemporal factbase. Rulebase is stored with domain knowledge used for inference. Rules are arranged with conditional (IF~) and active part(THEN~).

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

Figure 4 shows an example of the rule coded as "if statement" in Java. Execution rule is kept as a method type in the rule class forming rulebase. Rule_100 is a rule name given to divide rules, which is given unique rule number to every inference query. In this rule, the string beginning with underscore(_) and a capital like _Motion_Table represents spatiotemporal fact variable used in conditional and active part of the rule. On the other hand, the string beginning with capital like Main_strike represents local variable only used in active part. And, the string beginning with small letter like element_of expresses rule execution function performed in conditional or active part of the rule.

**Spatiotemporal Factbase.** Dynamic Spatiotemporal Fact Base stores spatiotemporal facts data required executing the rule. The kind and function of are differently defined according to application domain. In the rule of Figure 4, Mechanized unit, _Motion_table, _N_L, and _S_L were used as spatiotemporal facts variables. Spatiotemporal facts are created by spatiotemporal facts generating operator whenever new inference query is executed and are disappeared after the query execution.

Figure 5 is drawn for the information concerning distance and direction of moving objects having same valid time. Figure 5(a) adds valid time that is attributes of objects. Table (b) is the history table that manages moving information of units stored in initial location data table. This table only has the valid time and location coordinate for moving units.
temporal attribute of moving object to qualitative spatial distance relations like Near, Far, and VeryFar (Hong, Egenhofer, and Frank 1995, Sharma, Flewelling, and Egenhofer 1994), and uses operator T_Near, T_Far, and T_VeryFar. Operators T_Near, T_Far, and T_VeryFar search the moving objects that have the relations of Near(object B and C), Far(object D), and VeryFar(object E) for a target moving object A.

Figure 5  Spatiotemporal facts data for moving objects during the same valid time \(<VT_s, VT_e>\). But, the notion of distance such as Near, Far, and VeryFar has qualitative spatial characteristic, since it cannot be represented by one numerical value due to the difference of the degree of feeling depending on the individual. Therefore, actual system has been implemented by the quantitative operations, and previously allocates the scope of quantified values for Near, Far, and VeryFar.

Figure 5(b) adds temporal attribute to directional relation of spatial objects and searches spatiotemporal fact data of moving object during the arbitrary time interval \(<VT_s, VT_e>\). It uses operator T_Mdirection. Like this figure, when a moving object A moved from location P1 to P2, T_Mdirection operation can find moving direction and this operation is done by using azimuths of P1 and P2.

4.5 Inference Engine
Inference engine performs spatiotemporal inference queries concerning moving objects, is composed of rule classifier, fact data generator, and rule executor.

**Algorithm ReasonQuery(q)**
q : 'query' of string type
n : 'rule number' of integer type
f : 'fact set' of array type
1) begin
2)  invoke ExtractRule(q,n)
3)  invoke CreateFact(n,f)
4)  invoke ExecuteRule(f,n)
5) end

**Algorithm ExtractRule(q,n)**
q : 'query' of string type
n : 'rule number' of integer type
1) begin
2)  n ← get Rule Number from the first character of the query string q
3)  return n
4) end

**Algorithm CreateFact(n,f)**
n : 'rule number' of integer type
f : 'fact set' of array type
1) begin
2)  p ← extract fact operator list from the stored table in the knowledge base
3)  i ← total number of elements of array p
4)  for (j=0 to i-1 step 1)
5)    begin
6)      process the fact operation of p[j]
7)      f[j] ← store the result of fact operation
8)    end
9)  return f
10) end

**Algorithm ExecuteRule(f,n)**
f : 'fact set' of array type
n : 'rule number' of integer type
1) begin
2) invoke rule_n() : rule execution method
correspond to rule number n
3) mapping the fact set f to fact variables of
the rule_n()
4) check condition part and process action part
of the rule_n()
5) r ← store result of the rule_n()
6) invoke DisplayResult(r)
7) end

Figure 7 Algorithms of inference execution

Figure 7 is the algorithm that processes inference
queries. ReasonQuery(q) receives q(query) for input
value, and calls ExtractRule, CreateFact, and
ExecuteRule in order. ExtractRule(q,n) returns
the result extracting n(rule number) from inputted
q(query). CreateFact(n,f) extracts p(fact production
operator list) stored in knowledgebase using input
value n(rule number). Array p has all the number of
elements at l. If j repeats from 0 to i-1, the execution
result of operator kept at p[j] is stored at f[j](j-th
element of fact set f). After executing all the operators
of p, it returns f(fact set). ExecuteRule(f,n) is taken
over f(fact set) and n(rule number) for input value.
First, it calls rule execution method rule_n() corresponding rule number n. Fact set f is mapped to
fact variable of rule_n(). It checks conditional
part(if~)of rule_n(), processes active part(then~),
stores result to r(inference result), and calls result
output method DisplayResult(r). Rule execution
method rule_n() is stored as the format of production
rules and uses forward reasoning method. Production
rule (Dutta 1994) is said to represent or to infer
knowledge as a simple sentence IF ~ THEN. For
instance, if the rule of Figure 4 has the same structure
of Figure 7, and items A, B, C, D, E, a, b, c, and d of
conditional and active part have following contents,

A = element_of(a,b)
B = exist_on(a,c)
C = is_determined(e,c)
D = opposite_direction(c,d)
E = is_determined(e,c) &
is_determined(f,d)
a = '00-Mechanized Unit'
b = 'history_info'
c = 'load_1'
d = 'null'
activation of rule and reasoning process is shown in
Figure 8. Here, A, B, C, D, and E stand for one
condition described in conditional part of rulebase,
and element_of(a,b), exist_on(a,c), is_determined(e,c),
and opposite_direction(c,d) represent rule execution
function. And, a, b, c, d, and e are spatiotemporal fact
variables.

Figure 8 Rule activation and reasoning execution

Figure 8 shows the process of reasoning execution
that actual spatiotemporal facts are accomplished the
mapping to rule “A & B → C” and “A & B & D →
E”. Actually, the inference process via mapping from
the spatiotemporal facts proceeds in the same way as
solving logic equation. Here, we can know from this
instance that the contents of active part is performed
only if the contents of conditional part are all true,
and its result is also true.

5. System Implementation and Experimentation

5.1 Implementation Environment and
Inference Function

The battlefield analysis system using spatiotemporal
reasoning has been realized with client/server
structure. Client was implemented with Java 1.2 on
Window NT 4.0 and server was implemented with
Oracle 7 and Geowin space management tool on
UNIX. Moving object data used for battlefield
analysis adopt one Army battalion unit for the basic
unit, assuming that a battalion is a moving point.
There are five spatiotemporal inference functions
developed for battlefield analysis. First, unknown unit
inference reasons unit's name and assignment data,
when date and location among observed moving
information are correct, but unit's name and
assignment data aren't. Second, unidentified unit
inference reasons the name and location coordinate of
unit which has absolutely no observed moving
information. Third, main strike direction inference
reasons unit's moving direction in the future by using
unit's moving information and former knowledge.
Fourth, moving location inference predicts location
coordinate where a unit will move on certain date in
future. Fifth, moving time inference receives a future
location coordinate for a unit, and shows predicted time to arrive at that position.

5.2 Applied Scenario and Result of Experiments

We composed simulated battlefield scenario to experiment with the reasoning function of implemented system. This scenario makes it possible to establish the best corresponding strategy by analyzing current situation and by predicting attack direction when an army division is moving. First, we inputted observed data on May 1, 2000 to initial unit data table, supposing that twenty battalions of one division are the moving objects. Next, moving information during ten days for twenty battalions was generated. Now we suppose that accurate moving information is 80 percent (160 times) and the remaining 20 percent (40 times) is inaccurate moving information. Based on this scenario, we can perform inference queries through the following query input window.

Figure 9 is the user interface screen and inference query input window. Objects in an ellipse show the initial unit location from 2000/05/01 to 2000/05/02. Inference query inputs a query after choosing one from the six items registered in the query input window.

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

Figure 11 is the result screen of reasoning unidentified units corresponding to the valid time between 2000/05/13 and 2000/05/14. Since actual data stored in the moving objects database only exist until 2000/05/12, locations of all objects corresponding to valid time after 2000/05/13 are displayed on the screen as inferred by the system.
Figure 12  Inference query and result for main strike direction

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

Figure 13  Inference result for moving location

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

Figure 14  Inference query and result for moving time

Figure 14 is the result of reasoning the taking time for 22th-infantry battalion to move from current location to location (620, 520). The inside of tetragon is the route that 22th-infantry battalion has moved from 2000/05/01 to 2000/05/11, and the inside of circle shows the expected time and location required to get the location (620, 520).

5.3 Characteristic Analysis of Implemented System

In order to analyze and consider the characteristics of the proposed model, the reasoning system implemented for battlefield analysis was used to analyze the results. Query response rate by date for the system was considered as a factor of characteristic evaluation. It is represented by the percentage for the number of system response over the number of queries by date.

Queries used for this experiment are total of 500. They were made by creating 50 queries per date for ten days with respect to five reasoning functions such as unknown unit inference, unidentified unit inference, main strike direction inference, unit's moving location inference, and unit's moving time inference. Queries were built by assigning ten different valid times by date for each reasoning function. After executing 500 queries, query responses having the distribution like Figure 15 were obtained.
5.4 Characteristics of the Proposed Model

Existing temporal and spatial reasoning methods have some problems that cannot directly apply to the spatiotemporal reasoning application using moving objects. This paper worked out these problems in following ways by using the proposed model.

First, since the existing temporal or spatial reasoning separately considers temporal attribute or spatial attribute respectively, spatiotemporal attribute of moving objects can not be considered. The proposed model introduced spatiotemporal databases to figure out this problem. Spatiotemporal databases have a strong point that manages all histories of moving objects changing over time, since temporal and spatial attributes can be stored in one database.

Second, methodologies regarding representation of temporal and spatial knowledge are proposed in large numbers. But, the introduction of spatiotemporal operators, which can be applied to process spatiotemporal data, has not been considered at all. Therefore, this paper allows us to do dynamic spatiotemporal reasoning by using moving object operations and spatiotemporal operators in the process of inference engine. Moreover, there is no necessity for storing all the knowledge for topological and directional relations of moving objects, since spatiotemporal operation for moving objects can be performed.

Third, theoretical researches of spatiotemporal reasoning have obtained a lot of results for the last several decades, nevertheless there were so few cases applied to some application area. However, the need of applications using spatiotemporal reasoning is increasing in many fields. Particularly, applications such as navigational system, wargame model, close combat simulation, and battlefield information analysis system chiefly handle moving objects. Thus, this paper proposed a spatiotemporal reasoning system model using moving objects, implemented, and experimented with it applying to battlefield analysis. Consequently, we showed the applicability of the proposed spatiotemporal reasoning model for actual applications.

6. Conclusions

It is called moving object that data having a feature either which changes shape or general attributes of spatial objects over time or which changes location and shape of spatial objects with move. This paper proposes user interface, moving object databases, moving object processor, inference engine, knowledgebase, and a new type of spatiotemporal reasoning model using GIS tool to infer moving objects. Especially, we proposed the use of moving objects spatiotemporal databases so that the temporal and spatial attributes theoretically dealt in existing spatiotemporal reasoning study are applicable to the actual system development. And, we made dynamic spatiotemporal reasoning possible using spatiotemporal facts data acquired as the result of spatiotemporal operation and moving objects operation to the reasoning process.

In order to show the applicability of moving objects spatiotemporal reasoning model proposed in this paper, we applied it to battlefield analysis, implemented and experimented spatiotemporal reasoning system for battlefield analysis which conducts reasoning unknown unit, unidentified unit, and main strike direction, and predicts unit's moving location and moving time. Thus, we found that the proposed spatiotemporal reasoning model is applicable to the application system development using moving objects such as navigational system, wargame model, and close combat simulation.

Moving objects always have uncertainty due to sampling error or measurement error, because their attributes, locations, and shapes are changing over time. This paper has not applied the methods of processing this uncertainty of spatiotemporal moving objects to the reasoning process. Generally speaking, although the data or knowledge used in the inference engine is wholly correct, the reasoning system cannot guarantee the 100% correctness to the inferred result.
Therefore, it is necessary that the study about various processing methods which can minimize the uncertainty of spatiotemporal data applying to spatiotemporal reasoning and the study that raises the correctness of result by applying to actual reasoning process should be proceeded.

References:


