REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
05-04-2001	CONFERENCE PROCEEDINS	

4. TITLE AND SUBTITLE
A CLIPS-BASED IMPLEMENTATION FOR QUERYING BINARY SPATIAL RELATIONSHIPS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Huiqing Yang, Maria Cobb and Kevin Shaw

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Research Laboratory
Marine Geoscience Division
Stennis Space Center, MS 39529-5004

8. PERFORMING ORGANIZATION REPORT NUMBER
NRL/PP/7440--01-1003

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. MARINE CORPS WARFIGHTING LAB

10. SPONSOR/MONITOR'S ACRONYM(S)
MCWL

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The power of spatial queries for analysis and planning purposes in many different applicaton fields has drawn significant attention within the GIS research field. The extraction of meaningful information from spatial data requires specialized data structures, query languages and query processing strategies.

This paper is primarily concerned with the binary data structures that support the fuzzy queries of spatial relationships in two dimensions. For implementation purpose, the topological relations in this model are refined from a previously defined model. This modified binary spatial model will reduce the burden of geometric computation. Based on the modified binary spatial model, a CLIPS implementation for querying binary spatial relationships is investigated. Details about the query processing strategies are also provided.

15. SUBJECT TERMS
spatial queries, GIS, binary data, two dimensions, topological relations, CLIPS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			KEVIN SHAW
Unclassified	Unlassified	Unclassified		6	19b. TELEPHONE NUMBER *(Include area code)* 228-688-4197

A Clips-Based Implementation for Querying Binary Spatial Relationships

Huiqing Yang, Maria A. Cobb
Department of Computer Science & Statistics
University of Southern Mississippi
Hattiesburg, MS, USA 39401-5601
hyan2@orca.st.usm.edu
Maria.Cobb@usm.edu

Kevin B. Shaw
Naval Research Laboratory
Mapping, Charting & Geodesy
Stennis Space Center, MS, USA 39529
shaw@nrlssc.navy.mil

Abstract
The power of spatial queries for analysis and planning purposes in many different application fields has drawn significant attention within the GIS research field. The extraction of meaningful information from spatial data requires specialized data structures, query languages and query processing strategies.

This paper is primarily concerned with the binary data structures that support the fuzzy queries of spatial relationships in two dimensions. For implementation purpose, the topological relations in this model are refined from a previously defined model. This modified binary spatial model will reduce the burden of geometric computation. Based on the modified binary spatial model, a CLIPS implementation for querying binary spatial relationships is investigated. Details about the query processing strategies are also provided.

1. Introduction

Geographic Information Systems (GIS) is an integrated technology that incorporates concept from computer graphics, spatial modeling and database management. The ability to perform queries on spatial data is essential to GIS and related systems. Due to the fact that the ability to extract information for query results is dependent on the underlying structure of data, a great deal of research efforts have focused on the modeling of spatial data. It is worth mentioning that the work in [1] provided a novel contribution to the problem of defining spatial relationships by considering inferences from topological and directional relations.

In earlier work [1], a spatial data model that represents binary topological and directional relationships between two 2-D objects was presented. A data structure called an Abstract Spatial Graph (ASG) was defined for the binary relationship that maintains all necessary information regarding topology and direction. For complete information on this model, we refer the reader to the cited references.

In this paper, we present an implementation of the modified structures based on the C Language Integrated Production System (CLIPS). CLIPS is a productive development and delivery expert system tool which provides a complete environment for the construction of rule and/or object-based expert systems [2, 3]. It is now maintained as public domain software. Because of its portability, extensibility, capabilities, and low-cost, CLIPS has received widespread acceptance throughout the government, industry and academia.

Based on the modified spatial relationship, rules are encoded using the public-domain CLIPS language. The CLIPS code is processed through the CLIPS expert systems engine to answer the topological and directional queries for binary spatial objects.

The paper is organized as follows. Section 2 describes the rules of spatial relations based on the binary spatial model, and investigates a data structure improvement for implementation purposes. Section 3 provides details about CLIPS programming strategies for query processing. The query result section follows, providing a sample of how the implementation works. Our conclusion and directions for further work are presented in section 5.

2. A Binary Spatial Model and Its Modification

For the purpose of the model, we first assume that objects involved can be enclosed in Minimum Bounding Rectangles (MBRs). Figure 1 shows two MBR objects in 2-dimensions, i.e., each object can be represented by a two-point abstraction that represents the lower-left and upper-right corners of the MBR.

A tuple $[r_x, r_y]$ represents the relationship between the objects in both the horizontal and vertical directions. Each of r_x and r_y is one of Allen's temporal relations [4] that represents the interaction of the objects in the x direction and y direction, respectively.

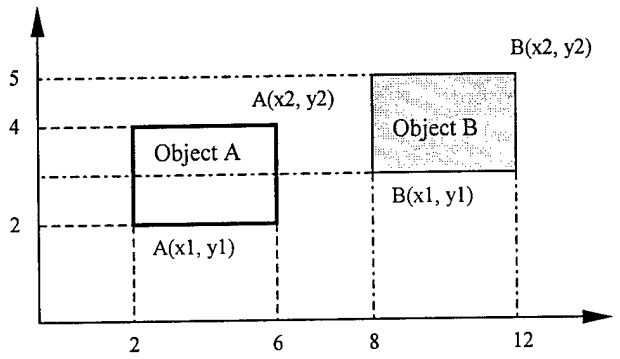


Figure 1. MBRs of two objects.

2.1 Basic Rule Sets of Binary Spatial Relations

Based on Allen's 13 temporal relations, a two-dimensional spatial data model was achieved. The result is that 85 possible relationships are deduced, which include 49 base relationships and 36 inverse relations. These relationships are then used to define topological and directional relationships.

In this paper, three rule sets are used to represent the basic structure of this model. Consider two objects:

A (Ax1, Ay1) (Ax2, Ay2)
B (Bx1, By1) (Bx2, By2)

Now, taking one-point from each object, that is,

(A1, A2) = (Ax1, Ax2) or (Ay1, Ay2)
(B1, B2) = (Bx1, Bx2) or (By1, By2),

we will present the implementation process.

Rule Set 1: *Define a set of non-ambiguous relationships.*

Consider one direction, the temporal relation between object A and object B can be defined as:

```
1. IF < A2<B1 >     THEN     < before >
   IF < B2<A1 >     THEN     < before^-1 >
2. IF < A2=B1 >     THEN     < meet >
   IF < B2=A1 >     THEN     < meet^-1 >
3. IF < A1<B1<A2<B2 > THEN < overlap >
   IF < B1<A1<B2<A2 > THEN < overlap^-1 >
4. IF < B1<A1<A2=B2 > THEN < finish >
   IF < A1<B1<A2=B2 > THEN < finish^-1 >
5. IF < B1<A1<A2<B2 > THEN < during >
   IF < A1<B1<B2<A2 > THEN < during^-1 >
6. IF < A1=B1<A2<B2 > THEN < start >
   IF < B1=A1<B2<A2 > THEN < start^-1 >
7. IF  <A1=B1<A2=B2 > THEN < equal >
```

Figure 2. Defining a set of non-ambiguous relations.

Simply, this rule set can be expressed as:

r_x = (b, m, o, f, d, s, =, b', m', o', f', s'),

where each relationship and its inverse is represented by its initial letter, e.g., 'b' → 'before.'

In the y-direction, the same rules can be applied. Moreover, there are two additional rules that apply:

1. $A(r_x^{-1}, r_y)B = B(r_x, r_y^{-1})A$
2. $A(r_x^{-1}, r_y^{-1})B = B(r_x, r_y)A$

Rule Set 2: *Define a set of topological relationships.*

Based on the eighty-five basic relationships, the topological relation set can be defined as:

T ={disjoint, tangent, surrounded-by, partially-surrounded, surrounded-by, partially-surrounds, overlapped-by, overlaps, x-subspace, y-subspace, y-subspaced-by}

Figure 3 shows a subset of the rules for topological relationships. [1] provides greater details on this.

```
IF <dd> THEN  <A surrounded-by  B>

IF <oo'|os'|of'> THEN  <A overlapped-by B>

IF <s=|d=|f=|==|o=> THEN <A x-subspace B>

IF <=s|=d|=f|==|=o> THEN <A y-subspace B>
```

Figure 3. Topological relationship rule set.

Rule Set 3: *Define the set of directional relationships.*

Directional relationships are heavily used in everyday life. The most commonly used are the cardinal directions and their refinements. In the same way as previously seen for topological relationships, the directional set can be defined as:

D ={North, East, South, West, North-East, South-East, South-West, North-West}

Figure 4 shows two of the rules for directions.

```
IF   <dd|df|fd|do|ds|ff|d=|fo|fs|
      f=|dd'|do'|ds'|fd'|df'|fo'|fs'>
THEN  < A East B >

IF    <dd|do|ds|fo|fs|db|dm|
       fb|fm|dd'|fd'|df'|ff'>
THEN  <A South East B >
```

Figure 4. Two directional relationship rules.

2.2 Define ASG for Fuzzy Querying

Three basic rule sets can support the basic binary spatial querying, i.e. the querying without specific degree information. Researchers [5-6] have shown that the directional relationships are fuzzy concepts since they depend on human interpretation. In addition to supplementary information needed for

fuzzy query processing, a data structure, known as an abstract spatial graph (ASG), was also presented in previous work [1]. The concept is based on the tasks of defining reference areas, partitioning MBR's into object sub-groups, and assigning each object sub-group to a node on the ASG.

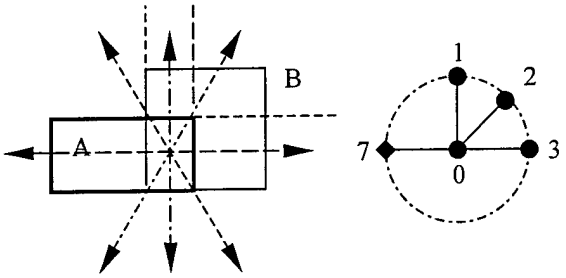


Figure 5. A [overlaps, start] B and corresponding ASG.

Figure 5 shows the geometry of the ASG, in which nodes 0-3 belong to object B and nodes 0 and 7 represent object A. Each node has associated weights that store fuzzy information.

2.3 Modifying AGS for CLIPS Implementation

The topological relations have been found useful for increasing the speed of spatial queries [5]. For implementation purpose, we analyze the geometric characteristics of topological relationships. Excepting the disjoint relation, all other relations have a similar geometry; that is, the reference area is part of both objects involved. Thus, the original topological relation set can be reduced or reclassified to a binary topological set:

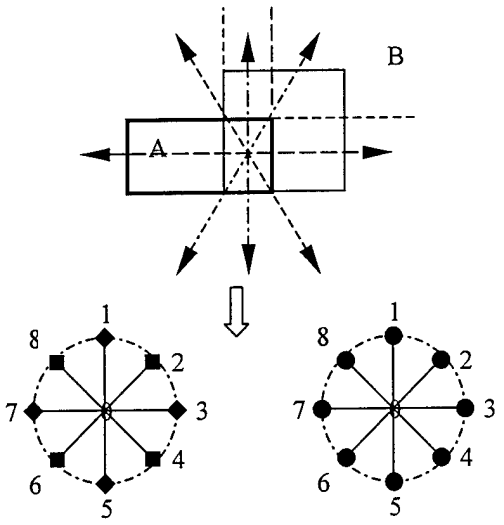
$$T \rightarrow T' = \{\text{disjoint, connected}\}$$

This new topological relation set is used in the CLIPS implementation.

For convenience of implementation and further investigation, the ASG is modified by mapping topological relationships to 9 nodes for both objects. Figure 6 represents the new ASG. Similarly, each node has associated weights. But differently, the weight in some node can be null depending on the different topological relations. In this new data structure, because each object is associated with its 9 nodes, it is not necessary to keep information related to whether a node belongs to object A or object B in the implementation. Furthermore, it is a flexible structure for fuzzy querying.

3. A CLIPS Implementation

In this section we show how CLIPS can be used to implement a the binary spatial relationships given



ASG for Object A ASG for Object B

Figure 6. A [overlaps, start] B and new ASG

earlier. Considering the amount of computation involved in implementation, we take advantage of the *deffunction* construct that allows the addition of new functions without having to recompile and relink CLIPS. Several user-defined functions are written by using the CLIPS *deffunction* construct, which can be executed by CLIPS interpretively.

As a rule-based shell, CLIPS stores the knowledge in rules, which are logic-based structures. In the implementation, the basic three rules are defined by using *defrule* constructs. They provide the basic spatial information such as, *Object A is disjoint from Object B,* or *Object A is West of Object B.* For fuzzy querying purposes, extra functions and rules are defined that will support fuzzy querying.

The implementation is directly dependent upon the reduced topological relation set and modified ASG mentioned above.

3.1 Store All Facts in CLIPS

The facts are the critical resources for the querying. All details for binary spatial relations are contained in *deftemplate* facts. The type of information stored in the database includes the positions of two objects, the reference object, non-ambiguous relations, and topological relationship and directional relationships. Figure 7 shows the information stored in facts using CLIPS syntax. The corresponding data structures are declared by using *deftemplate* syntax.

```
(object-position (objectname A)
       (x1 2 ) (y1 2 ) (x2 3 ) (y2 5 ) )

(2D-relation (object1 A)
             (relations bd) (object2 B))

(topological-relationship (object1 A)
    (t_relation disjoint) (object2 B) )

(directional-relationship (object1 A)
        (d_relation West) (object2 B)  )

(nodes (objectname  A)
       (Central   area_weight)
       (N  area_weight  direction_weight)
        :       :                :
       (NW area_weight  direction_weight)
)
```

Figure 7. Facts stored in database.

3.2 Representing 2-D Relation in CLIPS

To represent 2-D temporal relations extended from Allen's relations, the *deffunction* construct in CLIPS is utilized. With this construct, a new function that implements Allen's relations in 1-D is defined directly in CLIPS. Figure 8 shows the *deffunction* for Allen's internal relations. The knowledge of the rules implemented in step one that define a set of non-ambiguous relationships is built by the *defrule* construct shown in Figure 9.

```
(deffunction AllenRelation
                      (?A1 ?A2 ?B1 ?B2)
(if(< ?A2 ?B1) then (bind ?relation b))
(if (= ?A2 ?B1) then (bind ?relation m))
(if (and(< ?A1 ?B1)(< ?B1 ?A2)(< ?A2 ?B2))
 then   (bind ?relation o) )
(if (and (< ?B1 ?A1) (= ?A2 ?B2))
 then (bind ?relation f))
(if (and (< ?B1 ?A1) (< ?A2 ?B2))
 then   (bind ?relation d))
(if (and (= ?B1 ?A1) (< ?A2 ?B2))
 then   (bind ?relation s))
(if (and (= ?B1 ?A1) (= ?A2 ?B2))
 then   (bind ?relation =))
return ?relation
)
```

Figure 8. *Deffunction* for Allen's interval relations.

The function AllenRelation() develops a set of temporal relations in 1-D. It accepts four arguments from a CLIPS program. When it is called, it returns the temporal relation that can be used in the rule for the application.

The *defrule* collects the relation facts in 2-D by calling AllenRelation( ), and then puts the 2-D relation knowledge into facts.

```
(defrule define-2D-relation
 ?f3 <-(object-position (objectname ?A&A)
       (x1 ?Ax1)(y1 ?Ay1)(x2 ?Ax2)(y2
?Ay2))
 ?f4 <-(object-position (objectname ?B&B)
       (x1 ?Bx1)(y1 ?By1)(x2 ?Bx2)(y2
?By2))
=>
(bind ?x(AllenRelation ?Ax1 ?Ax2 ?Bx1
?Bx2))
(bind ?y(AllenRelation ?Ay1 ?Ay2 ?By1
?By2))
        :
(bind ?r (sym-cat ?x_relation
?y_relation))
(assert (2D-relation      (object1 ?A )
      (relations ?r) (object2 ?B ))))
```

Figure 9. *Defrule* to implement Rule Set 1.

3.3 Basic Binary Spatial Querying Using CLIPS

The basic queries are based on the primary topological set (Rule Set 2) and directional set (Rule Set 3). In this kind of querying, the degree to which one object lies in a particular direction with respect to a second object is not of concern. Figures 10 and 11 show CLIPS rule structures for topological relationship and directional relationship, respectively.

```
(defrule define-topological-relation
         (relationship (object1 ?A&A)
         (relations ?r) (object2 ?B&~A))
=>
  (if (eq ?r dd)
   then (bind ?tr "is surrounded by"))
               :
               :
  (if (numberp (member$ ?r
                 (create$ oo' os' of')))
   then (bind ?tr "is overlapped by"))
  (assert (topologic-relationship
          (object1 ?A)(t-relation ?tr)
          (object2 ?B))   )
```

Figure 10. Defrule for topological relationship.

```
(defrule define-directional-relation
         (relationship (object1 ?A&A)
         (relations ?r) (object2 ?B&~A))
=>
 (if (numberp (member$ ?r (create$ od of
                sd sf dd df fd  ff =d =f
                ob' om' oo' os' ..... )))
  then  (bind ?dr1 North))
               . . . . . .
 (loop-for-count (?count 1 8) do
    (bind ?dr (nth$ ?count (create$ ?dr1
              ?dr2 ?dr3  ..... ?dr7 ?dr8)))
    (if (numberp (member$ ?dr (create$
                 North East ..... West )))
     then
        (assert (directional-
relationship
            (object1 ?A)(d-relation ?dr)
             (object2 ?B)) ) ) )    )
```

Figure 11. Defrule for directional relationship.

3.4 Fuzzy Querying of Binary Spatial Relationships

Based on the new topological relation set and modified ASG data structure, we define three rules and four functions to support the processing of fuzzy queries. Query processing strategies are described as follows:

Step1. Find the reference area

Fuzzy variable *weights* store all fuzzy query information. In order to get weights for each node in the ASG, a reference area must first be found. Based on the fact that there are four points in the x-direction or y-direction, given two objects, a simplified approach to determine the reference area can be given.

Approach: The reference area is also treated as an MBR object. We take two middle points among the four points in each direction as the reference object position. It can be represented as R =(Rx_1, Ry_1) (Rx_2, Ry_2).

get-reference-object Rule and *reference* Function

Given two objects, the *get-reference-object* rule calls *reference* function to get the reference object position. The *reference* function accepts eight arguments that represent positions of two MBR objects, and finds the position for the reference object. Finally, it places the position information into the corresponding *object-position* fact .

Step2. Calculate weights

Based on the binary topological relations, a general method developed for connected relations is shown in Figure 12.

N_area	$= (Rx_2 - Rx_1)(Oy_2 - Ry_2)$
NE_area	$= (Ox_2 - Rx_2)(Oy_2 - Ry_2)$
E_area	$= (Ox_2 - Rx_2)(Ry_2 - Ry_1)$
SE_area	$= (Ox_2 - Rx_2)(Ry_1 - Oy_1)$
S_area	$= (Rx_2 - Rx_1)(Ry_1 - Oy_1)$
SW_area	$= (Rx_1 - Ox_1)(Ry_2 - Oy_1)$
W_area	$= (Rx_1 - Ox_1)(Ry_2 - Ry_1)$
NW_area	$= (Rx_1 - Ox_1)(Oy_2 - Ry_2)$

Figure 12. Formulas for area weight calculation.

In the figure, R represents the reference object, and O represents the one of two objects investigated. By adding some constraints, the general method for connected relations can also be applied to disjoint relations.

get-weight Rule and *weights* Function

Given two objects and their reference object, the *weights* function maps the object sub-group into 9 nodes for each object, and calculates the area weights and node weights. The CLIPS program passes nine arguments to *weights* function, that is, one for object identifier, four for object position, and four for reference position. The function asserts area weights to the corresponding nodes for fuzzy querying. The basic *weights* function structure is shown in figure 13. A related rule that activates the *weights* function.

```
(deffunction weights (?object ?Ox1 ?Oy1
          ?Ox2 ?Oy2 ?Rx1 ?Ry1 ?Rx2 ?Ry2)

(bind ?Total_area (* (- ?Ox2 ?Ox1)
                     (- ?Oy2 ?Oy1)))
(bind ?C_area  (/ (* (- ?Rx2 ?Rx1)
           (- ?Ry2 ?Ry1)) ?Total_area))
(if (and (<= ?Ox1 ?Rx1) (>= ?Ox2 ?Rx2))
 then
    (bind ?N_area  (/ (* (- ?Rx2 ?Rx1)
           (- ?Oy2 ?Ry2)) ?Total_area))
 else
    (bind ?N_area 0); for disjoint case)
       . . . . . .
;get the node weight for north direction
(if (> ?N_area 0)
 then (bind ?N_len(- ?Oy2(/(+ ?Ry2
                            ?Ry1)2)))
)
(if (< ?N_len 0) then (bind ?N_len 0))
(if (> ?N_len ?Longest)
 then (bind ?Longest ?N_len) )
     . . . . . .
(assert (nodes (objectname ?object)
        (C ?C_area )
        (N ?N_area (/(* ?N_area ?N_len)
                   ?Longest))
         . . . . . .   )          )
```

Figure 13. Function to calculate weights.

Step3. Get qualifier to implement Fuzzy querying

To provide support for fuzzy query processing, the fuzzy variable *weights* is assigned to the corresponding linguistic terms qualifier. The *fuzzyTq* function defines the topological qualifiers that represent the linguistic terms for area weight. Similarly the *fuzzyDq* function defines the directional qualifiers that represent the linguistic terms for node weight.

The fuzzy set for topological qualifiers is:

{all (0.96 – 1), most (0.6 –0.95), some (0.3 – 0.59)
little (0.06 –0.29), none (0 –0.05) }

The fuzzy set for directional qualifiers is:

{directly (0.96 – 1), mostly (0.6 –0.95), somewhat (0.3 – 0.59), slightly (0.06 –0.29), not (0 –0.05) }

The *fuzzy-query* rule in figure 15 provides the fuzzy querying information by calling *fuzzyTq* and *fuzzyDq* functions.

```
(defrule fuzzy-query
 ?f3 <-(nodes (objectname ?A&A)
              (C ?C_area )
              (N ?N_area ?N_len)
                 . . . . . .
              (NW ?NW_area ?NW_len) )
=>
 (if (neq ?A B ) then (bind ?obj B) )
 (loop-for-count (?count 1 8) do
   (bind ?dir (nth$ ?count (create$ North
                          ... North_West)))
   (bind ?area_w (nth$ ?count (create$
                 ?N_area ?NE_area ?E_area
                 . . .?SE_area ?NW_area)))
   (bind ?node_w (nth$ ?count (create$
                 ?N_len ?NE_len ?E_len
                 . . .?W_len ?NW_len)))
   (bind ?tq (fuzzyTq ?A ?area_w
                              ?dir  ?obj))
   (bind ?dq (fuzzyDq ?A ?node_w
                               ?dir ?obj))
   (if (and (neq ?tq non) (neq ?dq non ))
    then
     (printout t "query information" crlf)
) ) )
```

Figure15. Fuzzy query rule.

All of the CLIPS code is processed through the CLIPS expert systems engine to answer the topological and directional queries for binary spatial objects.

4. Query Results

Consider two objects:

object A (1 , 1) (5 , 3) and
object B (4 , 1)(8 , 7).

When the *define-2D-relation* rule is fired, calling AllenRelation (1 5 4 8) will return 'o', and the second calling of AllenRelation (1 3 1 7) will return 's.' Finally, the relation 'os' is added to the *temporal-relation* fact.

When the define-topological-relation rule is fired, the topological information 'Object A overlaps Object B' is displayed. When the define-directional-relation rule is fired, 'Object A is South Object B, Object A is South West of Object B, and Object A is West of Object B' are provided for directional relations. When the *reference* rule is fired, the reference object R(4, 1) (5, 3) is asserted into the fact database. While

the *get-weight* rule is firing, area weights and node weights are assigned into 9 nodes for each object.

Finally, the *fuzzy-query* rule fires, providing the following fuzzy querying information:

Most of Object A is West of Object B
Object A is mostly West of Object B
⇒ *Most of Object A is mostly West of Object B*

5. Conclusion and Directions for Further Works

In this paper, the capabilities of a binary spatial data model and a CLIPS tool to support fuzzy topological and directional queries have been shown. The results demonstrate that CLIPS is a flexible, powerful, and intuitive tool that can be successfully applied to spatial database analysis.

Because the querying involves handling concepts expressed by verbal language, such as direction, area weights and node weights, this kind of query is illustrative of problems that involve uncertainties. However, in this implementation, the representation of the fuzzy variable weight is based on classical set theory where the membership can be clearly defined by a set. It simply performs a low level fuzzy query.

In the future we plan to continue research on the use of CLIPS in spatial data analysis. We intend to investigate also the use of FuzzyCLIPS for high level information queries, in which the representation of weights information is based on the concept of fuzzy set theory.

6. References

[1] M. A. Cobb, "Modeling Spatial Relationships Within a Fuzzy Framework," *Journal of the American Society for Information Science,* Vol.49, No.3, 1998, pp 253-266.

[2] CLIPS Reference Manual, Version 6.10, August 5th, 1998.

[3] R. M. Wygant, "CLIPS – A Powerful Development and Delivery Expert System Tool", *Computers in Engineering,* Vol. 17, No. 1-4, 1989, pp 546-549.

[4] J. F. Allen, "Maintaining Knowledge about Temporal Intervals," *Communications of the ACM*, Vol. 26, No. 11, 1983, pp 832-843.D

[5] S. Winter, "Topological Relations between Discrete Regions," *Proceedings of 4th International Symposium, SSD'95,* Portland, ME, USA, August, 1995, pp 310-327.

[6] J. Peuquet and Z. Ci-Xiang, "An algorithm to Determine the Directional Relationship Between Arbitrary-shaped Polygons in the Plane," *Pattern Recognition,* 20(1), 1987, pp 65-74.