Representing Partial Spatial Information in Databases

Thodoros Topaloglou and John Mylopoulos

Department of Computer Science, University of Toronto
Toronto, Ontario, M5S 1A4, Canada

Abstract. In this paper we present a spatial data model which facilitates the representation of and reasoning with various forms of qualitatively and quantitatively incomplete spatial information. The model is founded on a combination of object-oriented and constraint-based data modeling facilities and provides for representations of variable precision and granularity. We identify four basic reasoning tasks required for query processing operations and outline algorithms for each task. Finally, we discuss extensions of the model and outline an implementation based on the Telos knowledge base management system extended with an appropriate constraint reasoning component.

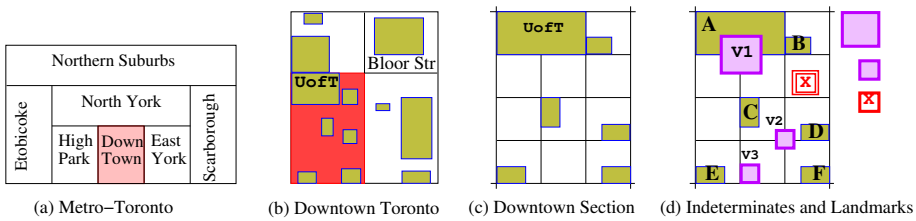
1 Introduction

Storing and manipulating spatial information is important for many database applications, including geographic information systems [MP94], vehicle navigation [Ege93], image retrieval [SYH94], protein structure prediction, environmental studies and many others. Existing spatial data models can generally be classified into two broad categories. The first category includes models that focus on explicit representations of space usually in terms of a *quantitative* formalism such as a map or a digitized array. We call these models *complete* since they represent the entire modeling space (see for example, [OM88], [RFS88], [GS95]). The second category includes models that focus on the representation of spatial features that are essential and are expressed in terms of a *qualitative* formalism such as symbolic arrays [GP92] or spatial relations [Her92]. These models are capable of reasoning about *partial* spatial information but, by-and-large, ignore quantitative spatial information and performance concerns.

This paper presents a spatial data model intended to accommodate both qualitatively and quantitatively partial spatial information. The expressiveness of the model is further enhanced by the provision of facilities for dealing with granularity and scale within a single framework. The formal tools employed in the paper include a conceptual modeling language, where the features of the proposed spatial data model are embedded, and a constraint-based language that is suitable for representing partially specified spatial information.

What are the sources of partial spatial information and what problems can it raise? Consider as an example a computerized system intended to coordinate first-aid vehicles that cover a geographic region (say, Metropolitan Toronto or Metro, for short). The region is divided into sections, which are further subdivided into subsections. For the coordinator and the vehicle drivers, *landmarks* serve as

"constants" whose locations are precisely known by all concerned. Other spatial information is represented relative to landmarks. Each vehicle, V, has a range of activity, denoted by *scope(V)*, which is the area that it can reach within, say, 2 minutes from its current position. Each vehicle reports its position to the coordinating station in imprecise terms (for example,"moving east","at Queen's Park", etc.). Hence, the vehicle position is represented by an *indeterminate* point. The scope of a vehicle is also partly known and is therefore represented by a rectangular region with an indeterminate boundary. Finally, the location of the trouble spot is reported by the coordinating station in imprecise terms, often through a reference to the nearest street intersection, and is represented by yet another indeterminate point. The reader should notice that indeterminate spatial objects in the example (vehicle positions, vehicle scopes and trouble spots) can be thought of as spatial variables which can take as value a spatial object within some spatial region. Moreover, indeterminate spatial objects are specified in varying degrees of precision, while for some of them only qualitative relationships may be known.



(a) Metro–Toronto (b) Downtown Toronto (c) Downtown Section (d) Indeterminates and Landmarks

Fig. 1. An example map at different scales

Figure 1(a) depicts the Metro region divided into sections. This information can be extracted from a city map and can be as precise as desired. To keep the example manageable, we focus on the downtown section which is further subdivided into subsections (see Figures 1(b-c)). The dividing lines are major streets of downtown Toronto. In Figure 1(d), the different drawing styles distinguish between landmark objects (denoted by A,B,...,F) and indeterminate objects, i.e, the scopes vehicles $\mathtt{V}_1$, $\mathtt{V}_2$ and $\mathtt{V}_3$ and a trouble spot $\mathtt{X}$.

A frequent task for the first-aid coordinator is to determine at any time which aid vehicle will serve a call, taking into account information about the relative positions of vehicles and the trouble spot. The problem is obviously dynamic, since the position and the scope of vehicles change over time, and a solution to the coordinator's query involves resolving efficiently a set of variably "grained" constraints. The primitive operations that a back-end spatial database needs to support to accommodate such queries are various topological (e.g., overlaps and contains), directional (e.g., north/south, left or above) and distance-based (e.g., within 1Km) predicates between operands with mismatched degrees of precision. It would also be desirable to be able to evaluate queries which receive an intentional answer such as "the places that can be served equally well by two given vehicles $\mathtt{V}_1$ and $\mathtt{V}_2$".

The rest of this paper is organized as follows. Section 2 includes an informal presentation of the features of the proposed spatial data model using the running example. Section 3 integrates the proposed features with the conceptual modeling language Telos. Section 4 presents a formalization of the spatial constraint language employed by the data model and outlines the algorithmic complexity of the query processing operation. Section 5 discusses related work and investigates directions in which the proposed model can be extended. Finally, Section 6 summarizes the contributions of the paper and briefly discusses future research.

2 Partial Information in Spatial Representations

This section first introduces the notions of scale and grain as basic concepts emerging from the example of the introductory section and then presents two fundamental constructs of the proposed spatial data model: the *spatial envelope* and the *map* structure.

2.1 Basic Concepts

A *spatial object* is a symbol structure representing a point or region of space.

Spatial Object Types. As suggested by the example, spatial objects can be either *landmarks*, in which case there is complete information about the point or region of space being represented, or *indeterminate* objects (or indeterminates, for short), for which there is only partial information about the point or region being represented. Indeterminates are related to landmarks through constraints expressed in a qualitative constraint language.

Scale. In a representation space, a scale is a system of ordered marks used as a reference standard in determining the relationships between representations. For metric domains, a scale is defined as an ascending set of point values which differ by a fixed interval, called *grain* (or *unit*), denoting distance from a fixed constant of the system. In qualitative domains a scale is defined as a fixed order between landmarks which may differ by a variable size qualitative interval.

Scale Hierarchy. A scale hierarchy is an ordered set of scales, $S = \{s_1, s_2, \ldots, s_n\}$, such that for each spatial object α at scale s_i there exists a container object $container(\alpha)$ at scale s_{i-1} that contains α, i.e., $inside(\alpha, container(\alpha))$, for $i = 2 \ldots n$. The existence of a unique container requires that the scales are not overlapping and that scale s_{i-1} is "coarser" than s_i. The coarser relation is a total order. The ordered set $\{city, section, division\}$ of the above example, defines such a scale hierarchy.

Haze. Haze is a region which encloses an indeterminate spatial object (point or region) and models its indeterminacy. The size of the haze area accounts for the degree of indeterminacy associated with the object. In the example, assume that the position of $\mathtt{V}_2$ is specified as "at University Ave. and Queen St." where $\mathtt{V}_1$ is said to be "near College and St. George St.". In effect, $\mathtt{V}_2$'s position is specified more precisely than that of $\mathtt{V}_1$. Consequently, the haze size for $\mathtt{V}_1$ is greater than the haze size for $\mathtt{V}_2$.

2.2 Spatial Constraints

Constraints have been shown to be very successful in representing qualitative and quantitative temporal information [vB90], [KL91], [Kou94b]. This section

considers a particular class of *spatial constraints*, which provide a convenient syntactic facility for expressing partial and relative information about spatial objects.

Spatial constraints are conjunctions of atomic formulas expressed in a constraint language such as the one developed in Section 4 that relates indeterminates and landmarks in one or two-dimensional space. In Section 4, we also define a set of higher level topological and directional spatial relations which can serve as basic vocabulary for the constraint language. For example: "$close(V2$, "University Ave. and Queen St.")", or "$south_side(V1$, "UofT") $\wedge\, east_of(V1$, "Spadina Ave.")".

The discussion in the rest of this section is restricted to constraints on a single dimension, that are conjunctions of the following two types of atomic constraints: $x\ R\ c$, and $x - y\ R\ c$; where x, y are variables representing indeterminates, c is a constant representing a landmark, and R is one of $\{=, <, \leq\}$.

2.3 Spatial Envelopes

Spatial constraints can be used to define arbitrary types of spatial indeterminacy. *Spatial envelopes* provide a convenient mechanism for defining a useful and often-occurring type of spatial indeterminacy. In particular, spatial envelopes constrain an indeterminate spatial object to fall within a rectangular region. If x is an indeterminate spatial object, its spatial envelope is denoted by $env(x)$. The spatial envelope of a two-dimensional object is a rectangle characterized by two one-dimensional envelopes. A one-dimensional envelope constrains the exact position of an indeterminate spatial interval.

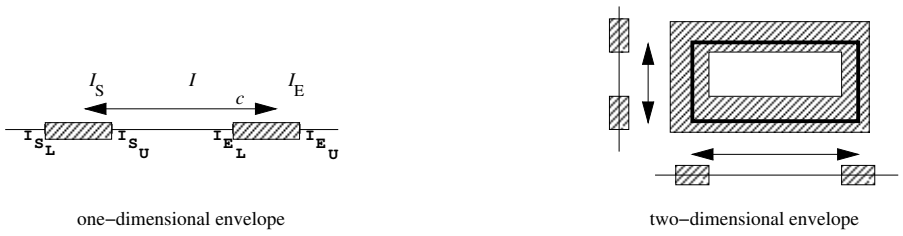


Fig. 2. Spatial Envelopes

Indeterminacy in one-dimension is formulated as follows. Let I be an indeterminate interval whose (partly known) start and end points are denoted respectively by I_S and I_E. The length of I is denoted by some constant c, defined with respect to the scale of the metric domain of I. The envelope interval of I, $env(I)$, is a pair of point envelopes, $env(I_S)$, $env(I_E)$ (see Figure 2 for a graphical illustration). In a discrete domain, each point envelope, $env(P)$, is represented by two points, $\mathtt{P_L}$ and $\mathtt{P_U}$ which impose lower and upper bounds on the position of point P. Assuming that the size of the haze is g, then points $\mathtt{P_L}$ and $\mathtt{P_U}$ are related by the constraint $0 \leq \mathtt{P_U} - \mathtt{P_L} \leq 2g$, and the envelope definition $\mathtt{P_L} \leq P \leq \mathtt{P_U}$

for some point P. For interval envelopes, the length of the interval poses an additional metric constraint, i.e., $I_E = I_S + c$. As a result, the one-dimensional *interval envelope*, $env(I)$, can be characterized by four variables, $\mathtt{I}_{S_L}$, $\mathtt{I}_{S_U}$, $\mathtt{I}_{E_L}$, and $\mathtt{I}_{E_U}$, related by the following constraints:

$$\{\ 0 \leq \mathtt{I}_{S_U} - \mathtt{I}_{S_L} \leq 2g,\ 0 \leq \mathtt{I}_{E_U} - \mathtt{I}_{E_L} \leq 2g,\ \mathtt{I}_{S_L} \leq I_S \leq \mathtt{I}_{S_U},\ \mathtt{I}_{E_L} \leq I_E \leq \mathtt{I}_{E_U},\ I_E - I_S = c\}$$

2.4 The Map Structure

The *map* structure is a logical data structure used to define a collection of spatial objects and their inter-relationships. Formally, a map structure m is a quintuple

$$m = \langle\ L,\ I,\ C,\ s,\ g\ \rangle$$

where L is a finite set of landmarks, I is a finite set of indeterminate spatial objects, C is a set of spatial constraints and s and g are its scale and grain, respectively. As is often the case, the same block of space may be represented by multiple maps of variable granularity.

Example 2.4.1 Figure 1(d) defines a map m with $L = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}\}$, $I = \{\mathtt{V}_1, \mathtt{V}_2, \mathtt{V}_3, \mathtt{X}\}$ at "city-division" scale and grain size g_m. The constraints of m are discussed in the next two examples. □

A valid map definition must be such that its grain size can accommodate the haze size of its indeterminates. More formally, if *haze* is a function returning the haze size of an indeterminate and g_m is the grain size of map m, then the condition is written as: $g_m \leq min\{haze(i) | i \in I\}$.

The construction of a map involves two phases: First, the set of landmarks of the map are represented, followed by a definition of the map indeterminates. For the first phase, we assume that the input is a "segmented image", e.g., an $n \times m$ occupancy array, containing a set of landmarks, L. The grain, g, for the map is captured by the size of the array cells. A set of X and Y constraints can then be defined so that each one contains all the known order and distance relationships between landmarks along the X and the Y axes, respectively. Indeterminacy can now be circumscribed for map indeterminates through spatial envelope constraints.

Example 2.4.2 This example shows a constraint representation created for the static part of Figure 1(d). As indicated earlier, landmarks are approximated by their minimum bounding rectangle; we therefore need four parameters for representing them, namely, A_{S_X}, A_{S_Y}, A_{E_X} and A_{E_Y}. For some grain g, the landmarks in the map are represented by the following two sets of equality constraints (i.e., the X and Y axis projections of its array representation):

$\{E_{S_X}{=}0, A_{S_X}{=}0, E_{E_X}{=}3, C_{S_X}{=}5, C_{E_X}{=}7, A_{E_X}{=}9, B_{S_X}{=}9, F_{S_X}{=}11, D_{S_X}{=}11, B_{E_X}{=}12, D_{E_X}{=}15, F_{E_X}{=}15\}$
$\{E_{S_Y}{=}0, F_{S_Y}{=}0, E_{E_Y}{=}2, F_{E_Y}{=}2, D_{S_Y}{=}5, D_{E_Y}{=}7, C_{S_Y}{=}7, C_{E_Y}{=}10, A_{S_Y}{=}15, B_{S_Y}{=}15, B_{E_Y}{=}17, A_{E_Y}{=}20\}$
□

Indeterminates are introduced next. According to the earlier discussion, a two-dimensional indeterminate is represented by a spatial envelope which consists of two coordinate one-dimensional envelopes. Each one-dimensional interval (resp. point) envelope is specified by four (resp. two) parameters, which are related by a fixed set of constraints as presented in Section 2.3. The notion of envelope parameters used here (in `courier` font) is similar to Koubarakis' *e(xistential)-variables* [Kou94a], intended to represent values which are not completely known

but for which a global constraint exists. The indeterminate parameters (in *italics* style) are also *e-variables* since they name a specific indeterminate and their possible values are bounded by envelope constraints.

Example 2.4.3 The insertion of a rectangular indeterminate into the above map is demonstrated next: Let $V1 = \langle V1X_S, V1X_E, V1Y_S, V1Y_E \rangle$, $grain(V1) = g_1$, $size(V1) = c_1$; c_1 is a constant that denotes the size of $V1$'s scope. As for the discussion of Section 2.3, constraints 1-10 become available. In addition, the position of $V1$ in the representation space is specified by constraints 11-14 (on the envelope parameters):

1. $0 \leq \mathtt{V1X}_{S_U} - \mathtt{V1X}_{S_L} \leq 2g_1$
2. $0 \leq \mathtt{V1X}_{E_U} - \mathtt{V1X}_{E_L} \leq 2g_1$
3. $\mathtt{V1X}_{S_L} \leq V1X_S \leq \mathtt{V1X}_{S_U}$
4. $\mathtt{V1X}_{E_L} \leq V1X_E \leq \mathtt{V1X}_{E_U}$
5. $V1X_E - V1X_S = c_1$
6. $0 \leq \mathtt{V1Y}_{S_U} - \mathtt{V1Y}_{S_L} \leq 2g_1$
7. $0 \leq \mathtt{V1Y}_{E_U} - \mathtt{V1Y}_{E_L} \leq 2g_1$
8. $\mathtt{V1Y}_{S_L} \leq V1Y_S \leq \mathtt{V1X}_{S_U}$
9. $\mathtt{V1Y}_{E_L} \leq V1Y_E \leq \mathtt{V1X}_{E_U}$
10. $V1Y_E - V1Y_S = c_1$
11. $(E_{E_X} = \mathtt{V1X}_{S_L}) \wedge (\mathtt{V1X}_{S_U} < C_{S_X})$
12. $(C_{E_Y} < \mathtt{V1Y}_{S_L}) \wedge (\mathtt{V1Y}_{S_U} < A_{S_Y})$
13. $(C_{S_X} < \mathtt{V1X}_{E_L}) \wedge (\mathtt{V1X}_{E_U} = C_{E_X})$
14. $(B_{E_Y} = \mathtt{V1Y}_{E_L}) \wedge (\mathtt{V1Y}_{E_U} < A_{E_Y})$

Constraints 1, 2, 11, and 13 are integrated into the X constraint set of the earlier map and similarly, constraints 6, 7, 12 and 14 are integrated into its Y constraint set. Constraints 3, 4, 5 and 8, 9, 10 are local to object $V1$. □

3 Modeling Space in Telos

This section describes the integration of the proposed features for representing spatial information with the conceptual modeling language Telos [MBJK90]. This integration endows the resulting spatial data model with abstraction mechanisms such as generalization, classification and attribution, inherited from Telos, as well as facilities for expressing meta-concepts and for asserting constraints and rules.

Integration of spatial modeling facilities into Telos is accomplished through a library of meta-classes and meta-attributes that capture the semantics of the features presented in the previous section. The central class of the model is the `Map` class. Spatial information is attached to physical objects through the attribute mechanism of Telos. A spatial object participates in one or more maps that may vary in scale, grain and implementation. This becomes possible due to the capacity of Telos to handle multi-valued attributes. Its practical importance is that the resulting spatial data model can facilitate multiple representations of space. Finally, scale and scale transitions are modeled using the Telos assertional language.

It is assumed that the world being modeled includes, among other things, physical objects, which might have a temporal and a spatial aspect [Hay85]. The class `PhysicalObjClass` is a metaclass whose instances include physical object classes such as the class of vehicles, and the class of buildings or parks. All these classes are also specializations of `PhysicalObject`, which is also an instance of `PhysicalObjClass`. In addition, we introduce the metaclass `SpatialObjClass`, whose instances are spatial object classes such as `Street`, `Lot`, `Parcel` etc. These classes are also specializations of `SpatialObject`, which is an instance of `SpatialObjClass`. Physical objects can have an associated spatial object, about which information is represented in terms of one or more maps.

```
CLASS PhysicalObjClass IN M1_CLASS
  WITH
    necessary,single
      when:TemporalClass
      where:SpatialClass
      what:OrdinaryClass
    attribute
      feature:AttributeClass
      time-feature:TemporalAttrClass
      space-feature:SpatialAttrClass
END PhysicalObject

CLASS PhysicalObject IN PhysicalObjClass
  WITH
    where
      place:SpatialObject ...
END PhysicalObject
```

```
CLASS SpatialObject IN SpatialObjClass
  WITH
    attribute
       in-space:SpatialObject
       in-map:Map
END SpatialObject

CLASS UofT-Lot IN SpatialObject
  WITH
    in-space
       s1:division-City-Toronto-Parcel
    in-map
       m1:map1;
       m2:map2;
       m3:map3
END UofT-Lot
```

According to these definitions, the `place` attribute of `PhysicalObject` is declared to be an instance of the `where` meta-attribute of `PhysicalObjClass`. A spatial object has an `in-space` attribute, which provides a spatial context, and zero or more associated maps that give information about the object. The next two definitions introduce different classes of spatial relationships. [1]

```
CLASS In-Map IN SpatialAttrClass
  WITH
    components
       from:SpatialObject
       label:in-map
       to:Map
    attribute
       rtype:RepresentationType
END In-Map
```

```
CLASS In-Space IN SpatialAttrClass
  WITH
    components
       from:SpatialObject
       label:in-space
       to:SpatialObject
    attribute
       stype:SpatialType
END In-Space
```

According to these definitions, `in-map` attributes have an associated attribute which specifies the representation type (landmark or indeterminate) of an object in a map. [2] Note that the same spatial object may have different types in different maps, i.e., be a landmark in one map and an indeterminate in another. Likewise, `in-space` associates a spatial type (region, point,...) to every spatial object / spatial context association. Again, the definition implies that a spatial object may have different types in different contexts.

A spatial object that serves as context for another object is itself described in terms of one or more maps. For example, the spatial token `UofT-Lot` is part of a division of the city of Toronto parcel, another spatial object, and participates in maps `map1, map2` and `map3` through relationships `m1`, `m2` and `m3`. The two types of spatial object types supported in our model are declared as instances of `RepresentationType`. Analogously, the geometric types of objects (i.e., point vs. region) are defined as instances of `SpatialType`.

[1] Telos does not have a built-in distinction between attributes and relationships.

[2] In Telos, all relationships are represented by a three-tuple, ⟨`from`, `label`, `to`⟩, which is called *proposition*. Intuitively, a proposition can be thought of as a link.

```
CLASS Map IN SpatialObjClass
  WITH
    single,necessary
      grain: Grain
      scale: Scale
    attribute
      X-constraint: ConstraintSet
      Y-constraint: ConstraintSet
    deductiveRule
      R1:indeterminate(ThisClassInstance,X):-
             instance(X,SpatialObject), instance(Y,In-map),
             proposition(X,Y,ThisClassInstance), rtype(Y,Indeterminate).
      R2:landmark(ThisClassInstance,X):-
             instance(X,SpatialObject), instance(Y,In-map),
             proposition(X,Y,ThisClassInstance), rtype(Y,Landmark).
    integrityConstraint
      :indeterminate(ThisClassInstance,X) and X.haze>ThisClassInstance.grain
END Map
```

The `Map` class models the principal data structure of our spatial data model. A map is characterized by its `grain` and `scale` attributes and the set of objects it inter-relates. `X-` and `Y-constraint` attributes take as values object inter-relationships, where the objects included in a map are retrieved by deductive rules. [3] Additional integrity constraints specify properties that any `Map` instance needs to satisfy. Granularity and scale are modeled in a similar fashion.

4 Spatial Constraints

This section presents a formalization of the proposed spatial constraint language and derives some of its formal properties. The section ends with an enumeration of four spatial reasoning tasks and a discussion of algorithms for solving each one of them.

4.1 A Language for Spatial Constraints

$\mathcal{L}_{\mathcal{H}}$ is intended as a language for specifying qualitative and quantitative constraints in one-dimensional space. $\mathcal{L}_{\mathcal{H}}$ is defined as a sorted predicate calculus with two sorts: hazy points, denoted by H, and exact points, denoted by E; $x, y, z, ..$ are variables of sort H, $c_1, c_2, c_3, ..$ are constants of sort E, and g is a designated constant also of sort E. The predicate symbols of $\mathcal{L}_{\mathcal{H}}$ include $\approx$, $\prec$ relating objects of sorts $H \times H$ and $H \times E$, and predicate symbols $=, <$ relating objects of sorts $E \times E$. $\mathcal{L}_{\mathcal{H}}$ also has a the function symbol "$-$" of sort $H \times H \rightarrow E$. An atomic formula of $\mathcal{L}_{\mathcal{H}}$ has one of the following forms: $x \approx y$, $x \prec y$, $c_1 = c_2$, $c_1 < c_2$, $x \approx c_i$, $x \prec c_i$, $y - x = c_i$, and $y - x < c_i$.

The well-formed formulas of $\mathcal{L}_{\mathcal{H}}$ are build up from the atomic formulas using logical connectives ($\neg$, $\vee$ and $\wedge$) and quantifiers ($\exists, \forall$) whose variables have to be of sort H.

The terms of $\mathcal{L}_{\mathcal{H}}$ are interpreted over the domain of integers. g is interpreted as the integer constant `g`. Each constant c_i of the sort E is interpreted as the integer

[3] For simplicity, deductive rules R1 and R2 are specified in Prolog notation.

$\mathtt{i}$. Symbols $<, =$ and $-$ receive the standard interpretation of order, equality and subtraction over integers. Each variable x of sort H is interpreted as an integer $\mathtt{x}$ that ranges in the interval $[\mathtt{x}^* - \mathtt{g}, \mathtt{x}^* + \mathtt{g}]$, where $\mathtt{x}^*$ is some integer. Intuitively, $\mathtt{x}^*$ denotes the median of x's interpretation. Predicate symbols $\approx$ and $\prec$ are interpreted by the relations $R(\mathtt{x}, \mathtt{y}) = \{(\mathtt{x}, \mathtt{y}) : |\mathtt{x} - \mathtt{y}| \leq 2\mathtt{g}\}$ and $S(\mathtt{x}, \mathtt{y}) = \{(\mathtt{x}, \mathtt{y}) : \mathtt{y} - \mathtt{x} > \mathtt{g}\}$, respectively. The notions of truth and satisfaction are defined in the usual way [End72].

It should be noted that $\mathcal{L}_{\mathcal{H}}$ does not use constants to name a specific element of sort E; instead, it uses unbound variables which receive as interpretation the specific element of E in any particular model of its theory.

Moreover, if in an interpretation g is assigned to the value 0, then $\approx$ and $\prec$ have "exact" meaning (i.e., same as $=$ and $<$), and $\mathcal{L}_{\mathcal{H}}$ becomes the language of discrete point order constraints. Discrete point order constraints have been employed in Section 2 to express spatial envelope constraints. Note that envelopes have already "compiled away" spatial indeterminacy.

Example 4.1.1 This example elucidates the meaning of the terms of $\mathcal{L}_{\mathcal{H}}$. Assume that $\mathtt{g}$=2 throughout the example. Let v be an interpretation which assigns y to $\mathtt{y}^*$=6, then $\mathtt{y}$ can be one of {4,5,6,7,8}. If $x \approx y$, then the image of x has to be within at most 2 points from a $\mathtt{y}$, i.e., $\mathtt{x}$ can be one of {2,3,4,5,6} if $\mathtt{y} = \mathtt{4}$ and so on.

In the same model, $x \prec y$ means that $\mathtt{x}^* < 4$ if $\mathtt{y}^*$=6. The meaning of the terms $c_1 = c_2$ and $c_2 < c_1$ is the obvious one. The (hazy) equality between a hazy point and an exact constant, $x \approx c_i$, means that $\mathtt{i} - \mathtt{g} \leq \mathtt{x} \leq \mathtt{i} + \mathtt{g}$, i.e., if $\mathtt{i}$=8 then $\mathtt{x}$ can be one of {6,7,8,9,10}. Analogously, in the same model, x for $x \prec c_8$ is interpreted by an integer which is less than 6.

The subtraction operator "$-$" is used to define a notion of distance between hazy points; "$-$" returns an exact quantity. [4] Let, for instance, $\mathtt{x}$ be interpreted in {1,2,3,4,5} and $\mathtt{y}$ in {7,8,9,10,11}, the term $y - x = c_6$ means that $\mathtt{x}$ and $\mathtt{y}$ are now restricted to be exactly 6 units apart, i.e., the following five pairs {(1,7),(2,8),(3,9),(4,10),(5,11)} are the only allowable interpretations for the pair x, y. □

The language of one-dimensional hazy-point space is extended to a language where one can state relationships between two-dimensional hazy points and rectangles. To do so we use the technique of independent combination introduced in [Top94]. Independent combination views each dimension in a multi dimensional space independently. For the two-dimensional case, for instance, it requires that each two-dimensional point is a pair of two one-dimensional points, and each two-dimensional rectangle is a pair of two two-dimensional points. The intuition behind independent combination is that any operator between two spatial objects in k-dimensions is written as a conjunction of k expressions involving only primitive one-dimensional operators over the projected coordinates. In [Top96] we proved that the satisfiability of a formula with the independent combination property is reduced to the satisfiability of its one-dimensional counterparts. Hence,

[4] A subtraction operator "$\sim$" which returns a hazy quantity is also defined but not included in this presentation.

we are able to generate a calculus on two-dimensional spatial objects which has the computational properties of its one-dimensional coordinate calculi. Moreover, we are able to compose a solution for a two-dimensional constraint satisfaction problem by combining solutions of its coordinate problems.

Independent combination imposes some limitations to the expressiveness of the constraint language for multi-dimensional space. One limitation is that it assumes orthogonality of space. However, this is a well-founded assumption in reality. Moreover, the resulting language is expressive enough to cover an interesting set of spatial relationships encountered in geographic information systems [Pap94a] and picture retrieval systems [SYH94]. Another less restrictive limitation is that we consider two-dimensional spatial objects to be either points or rectangles. This limitation is opted for presentation purposes as, in Section 5, we show that the language can be extended to model objects of arbitrary shape. In the next section, we discuss sets of spatial relations that are expressible in our formalism.

4.2 Spatial Relations

Egenhofer's [Ege91] eight fundamental topological relations for two planar regions is the most popular set of topological spatial relations in spatial databases. One advantage of this proposal is its clean topological semantics. In our work, we offer an alternative semantics for these relations based on the ontology of hazy points. The novelty of our approach is that it considers spatial relationships between objects with vague boundaries. In addition, a measure of precision appears as a parameter in the relations' definition, thus making reasoning about imprecise spatial information possible.

Figure 3 shows a graphical presentation of topological relations. Their formal definition requires the definition of the helping relations in_0, in_1 and $close$. In the following, $P, P1$ and $P2$ are two-dimensional points and $R, R1, R2$ are rectangular regions. Subscript $_i$ denotes the projection coordinate: $_i$ is either $_X$ or $_Y$. R_S and R_E denote the bottom-left and the top-right points of R, respectively.

$$in_0(P,R) \equiv (R_{S_X} \prec_X P_X \wedge P_X \prec_X R_{E_X}) \wedge (R_{S_Y} \prec_Y P_Y \wedge P_Y \prec_Y R_{E_Y})$$
$$in_1(R1,R2) \equiv (R2_{S_X} \prec_X R1_{S_X} \wedge R1_{E_X} \prec_X R2_{E_X}) \wedge (R2_{S_Y} \prec_Y R1_{S_Y} \wedge R1_{E_Y} \prec_Y R2_{E_Y})$$
$$close(P1,P2) \equiv (P1_X \approx_X P2_X) \wedge (P1_Y \approx_Y P2_Y)$$

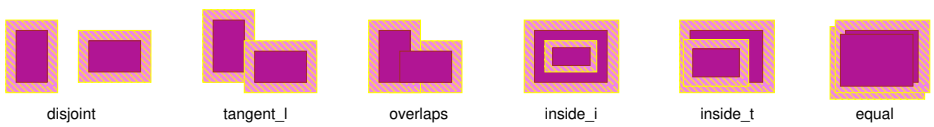


Fig. 3. Hazy topological relations

$$\texttt{disjoint}(R1,R2) \equiv \neg\exists P(in_0(P,R1) \wedge in_0(P,R2))$$
$$\texttt{tangent}_1(R1,R2) \equiv \exists P1,P2(in_0(P1,R1) \wedge in_0(P2,R2) \wedge close(P1,P2))$$
$$\texttt{overlap}(R1,R2) \equiv \exists P(in_0(P,R1) \wedge in_0(P,R2)0$$
$$\texttt{inside}_i(R1,R2) \equiv in_1(R1,R2)$$
$$\texttt{inside}_t(R1,R2) \equiv R2_{S_X} \preceq_X R1_{S_X} \wedge R1_{E_X} \preceq_X R2_{E_X} \wedge R2_{S_Y} \preceq_Y R1_{S_Y} \wedge R1_{E_Y} \preceq_Y R2_{E_Y}$$

$$\begin{aligned}
\texttt{contain}_{\texttt{t}}(R1,R2) &\equiv \texttt{inside}_{\texttt{t}}(R2,R1)\\
\texttt{contain}_{\texttt{i}}(R1,R2) &\equiv \texttt{inside}_{\texttt{i}}(R2,R1)\\
\texttt{equal}(R1,R2) &\equiv close(R1_S,R2_S) \wedge close(R1_E,R2_E)
\end{aligned}$$

[5] The notion of tangency as defined above, is "loose" tangency in the sense that two rectangles share a point of their haze. If the size of the haze decreases, a tangent relationship will change to disjointness.

Many researchers developed sets of directional relations exploring either characteristic points of the participant objects [PS94], [Her92] or the order relation of the underlying domain [SYH94]. Our directional relations definition is closely related to the approach of [SYH94]:

$$\begin{aligned}
\texttt{left_of}(R1,R2) &\equiv R1_{E_X} \prec_X R2_{S_X} \qquad \texttt{right_of}(R1,R2) \equiv \texttt{left_of}(R2,R1)\\
\texttt{above}(R1,R2) &\equiv R2_{E_Y} \prec_Y R1_{S_Y} \qquad \texttt{below}(R1,R2) \equiv \texttt{above}(R2,R1)
\end{aligned}$$

Finally, and as a result of the independent combination property, our model can handle relationships between one-dimensional and two-dimensional point or region objects. Space limitation do not allow us to list them here.

4.3 Reasoning

A map structure is represented in terms of two constraint sets, $Xcon$ and $Ycon$, each of which is a conjunction of $x\ R\ c$ and $x - y\ R\ d$ atomic constraints, where x, y are variables representing the parameters of a spatial envelope, R is one of $\{=, <, \leq\}$, d is a grain parameter, and c is a constant corresponding to a landmark parameter. Any map with an envelope representation for its indeterminate objects can be placed in this simple normal form. Note that the set of variables in the two sets is disjoint (except for d's). The reader should also recognize that the deployed language in the map constraints is the language of linear order constraints on integers as resulted from the compilation of indeterminates into envelope constraints.

The fundamental reasoning problems addressed in a constraint representation of a map structure are as follows:

P1. *Given the Xcon and Ycon constraint sets, decide if the constraint sets are satisfiable, i.e., there is an assignment for variables that satisfies every atomic formula in the Xcon and Ycon set.*

P2. *Given the Xcon and Ycon constraint sets, compute an assignment for all variables that satisfies every atomic formula in the Xcon and Ycon set.*

The type of the d parameters plays a pivotal role in the determination of the complexity of the above problems. If d's are integer constants (fixed grains) then both problems are straight-forward to solve in polynomial time. For instance, problem **P1** could be solved using one of several path consistency algorithms proposed in [Mac77], [DMP91], [KL91], [Kou94b]. The complexity of path consistency algorithms for the type of constraints considered, is $O(n^3)$ where n is the number of variables in the constraint set. In the database literature, the classic results of [RH80] offer another alternative with the same complexity. The second reasoning problem, is closely related to the first one. In fact, a solution for

[5] Notation: $x_i \preceq_i y_i \equiv x_i \prec_i y_i \vee x_i \approx_i y_i$.

P2 implies a solution for **P1**. Plethora of solutions is available for **P2**, including the above-mentioned path consistency algorithms as well as the dual method involving variable elimination algorithms (see [LM88], [Kou92]).

The two problems change complexity if d's are taken to be integer variables. Then for a single d value efficient algorithms are still possible , since computing a solution involves solving a system of linear inequalities, a known polynomial complexity problem. For more than one d values, on the other hand, the problem of computing a minimal solution becomes intractable since it is equivalent to integer programming [Pap94b]. Our on-going work attempts to identify efficient special cases for the last problem.

In addition to **P1** and **P2**, there are two derivative spatial reasoning problems which require attention:

P3. *Given a consistent and minimal constraint set, $Xcon \cup Ycon$, of a map structure m, and i, an indeterminate of m, find the strongest possible bounds for the parameters of i.*

A solution to **P3** involves, first, projecting a solution of **P2** to the variables of i's envelope and, second, applying a path consistency algorithm on the selected set of constraints conjoined with i's local constraints. Both steps are realized in polynomial time. Note that, to determine the consistency of a map, m, we need to test **P3** for all of its indeterminates.

P4. *Given a consistent and minimal constraint set, $Xcon \cup Ycon$, of a map structure m, and g', a new grain value (resolution) for m, recompute problem* **P2** *with grain value g'.*

Problem **P4** involves recomputing the constants for the landmark parameters in the $Xcon$ and $Ycon$ constraint sets and then computing **P2**. The first step is solved in linear time. Assume that $g = r * g'$, $r > 1$, i.e., resolution is refined. Then each $P = v$ conjunct in the $Xcon$ and $Ycon$ constraint set is replaced by the constraint $(r * (v - 1) \leq P) \wedge (P \leq r * v)$. Similarly, if $g' = c * g$, $c > 1$, i.e., resolution coarsens, then each term $P = v$ is replaced by with $P = \frac{v}{s}$. [CCMSP93] presents an elaborate framework for scale-related granularity which is relevant to the above problem.

5 Discussion

This section completes the presentation of the proposed spatial data model by pointing at some of its expressive features but also discussing directions in which it can be extended.

The proposed spatial data model integrates ideas from object-oriented knowledge representation [MBJK90], constraint-based data models [KKR90], spatial knowledge representation [PS94], quantitative and qualitative temporal reasoning [KL91], and granularity modeling [CCMSP93]. In result, our spatial data model covers wide range of features, such as indeterminate spatial objects, multiple scales, granularity and spatial relations.

Our spatial data model may be atypical of other proposals, however it complies with general structure of spatial data models. Guting [Gut94] identifies four

desirable abstractions of space that need to be represented in a database. Our spatial model matches features from three of these abstractions, namely, organizes the underlying space on a geometric basis (represented by constraints), offers a spatial relation-based language, and, integrates geometric types into the data model. Until now, our model has limited spatial data types support, points and rectilinear regions, but as we show bellow, the integration of lines and polygons is straight forward.

Our work is related also to temporal databases where the issues of granularity, scale and partially specified information have received considerable attention (e.g., [Kou94a], [DS93], [WJS93]) in the last few years.

Work on "undetermined" boundaries [Had95], and multiple representations of space [PD95], [RS95] are also related to this work. In fact, our model agrees with the proposal of [Had95] which suggests that undetermined boundaries can be founded on stochastic space, given that a haze area that surrounds a point space essentially represents the uniform distribution of the probability that this point can be found inside the area. Consequently, our model sets up a formal and computational base to address issues emerging from this proposal.

The ability to reduce the resolution of a map is called in Geography the map generalization operation [PD95]. Reasoning operation **P4** is essentially an instance of the map generalization operation in the context of minimum boundary rectangle representations of spatial data such as the one studied in [PTSE94].

An obvious, and for many practical reasons useful, extension of our data model, is to provide facilities for the representation of arbitrary complexity objects. This can be accomplished by defining a line to be a pair of two points, called endpoints. A chain is then defined to be a sequence of connected lines, and a polygon to be a closed polygon. A polygon in a plane defines two regions the interior and the exterior. Since the points used in these definitions have a haze, the defined geometric types are inheritedly "haze". The representational model is now extended with construction operators such as $line(P_1, P_2)$ which constructs a line out of two points, etc. Relations between objects can now be defined either in terms of Egenhofer's 3-intersection model [Ege91] or by relating characteristic points of objects [Pap94a]. A necessary extension to the underlying computational model is the redefinition of the primitive *point_to_polygon* operation as a 3-valued operation (i.e, to return as answer yes/no/maybe). This change affects only reasoning with topological relations using the 3-intersection model. The proposed constraint-based reasoning model based on two one-dimensional projections and minimum bounding rectangle (mbr) approximations is still applicable. In addition, further non-mbr based computations using the one-dimensional projections are possible provided that a new mapping between two-dimensional relations and their one-dimensional projections is defined.

An implementation of the proposed data model is achieved by first expressing the non-spatial data concepts using an existing data model (e.g., an object-based or a relational system), then integrating the proposed spatial features and finally implementing a constraint-based inference engine for spatial constraints.

To-date, there is no generally accepted way for implementing large constraint databases. Logic programming is an obvious alternatives with serious limitations

regarding its scalability to large datasets. Constraint-based reasoners is another option. This work follows the last option. Our proposed solution to the scalability problem is to explore the scale and grain features in order to partition large chunks of space into many small maps.

The first part of our prototype implementation integrates the proposed features for representing spatial information using the conceptual modeling language Telos [MBJK90]. This is accomplished through a library of meta-classes and meta-attributes as has already been discussed in Section 3. The integrated system is then put together using the framework developed in the Telos knowledge base management project [MCP+96]. Figure 4 displays part of the knowledge base that implements the spatial data model using the Telos system browser.

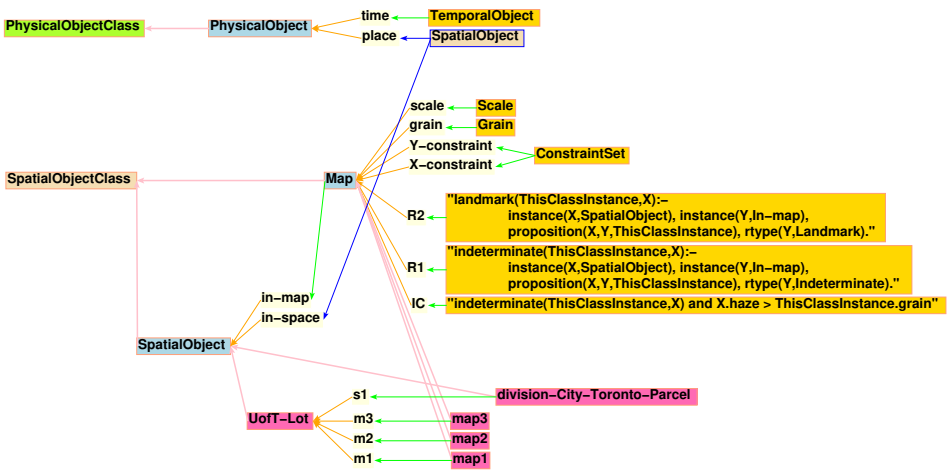


Fig. 4. Implementing Spatial Information in Telos

The second part of our implementation involves implementation of constraint reasoning in the context of haze orders and integrating it with the object model. Our accomplishment to date include the implementation of algorithms for supporting the reasoning tasks **P1** and **P2** presented in Section 4.3. A description of the algorithms and their implementation appears in [Top96].

The implementation will be completed with the development of a third component which will be responsible for answering queries with spatial and non-spatial qualifications. To date, this component is not available.

6 Conclusions and Future Research

We have presented a spatial data model which facilitates the representation of and reasoning with various forms of qualitatively and quantitatively incomplete spatial information, including indeterminate objects, multiple scales and granularity. Representation of incomplete spatial information is accomplished through a spatial constraint language with built-in notions for representing partial spatial information. Reasoning with such representations is addressed by identifying four classes of reasoning tasks and offering efficient processing algorithms for each

class. Our proposal accommodates object-orientation by embedding the proposed model within Telos and exploiting the meta-modeling facilities of the latter.

A number of research issues remain open and will be addressed in the future:

- A thorough case study using an application domain with varying granularity and imprecise data which can be a source for implementation ideas; Genome and crystallographic databases are two promising applications for this study.
- A query language facility that can accommodate spatial and non-spatial qualifications. Although a logic-based language is the most obvious alternative, its inefficiencies suggest that a better alternative, possibly graphical, must be sought.
- The extension of the model to accommodate Datalog [BCW93] type of rules. Beyond advancing the expressive power of the formalism, this would also make it more usable.

References

[BCW93] M. Baudinet, J. Chomicki, and P. Wolper. Temporal Deductive Databases. In *Temporal Databases: Theory, Design, and Implementation*, pages 294–320. Benjamin/Cummings, 1993.

[CCMSP93] E. Ciapessoni, E. Corsetti, A. Montanari, and P. San Pietro. Embedding Time Granularity in a Logical Specification Language for Synchronous Real-time Systems. *Science of Computer Programming*, 20:141–171, 1993.

[DMP91] R. Dechter, I. Meiri, and J. Pearl. Temporal Constraint Networks. *Artificial Intelligence*, 49:61–95, 1991.

[DS93] K. Dyreson and R. Snodgrass. Valid Time Indeterminacy. In *Proceedings of the 9th International Conference on Data Engineering*, 1993.

[Ege91] Max Egenhofer. Reasoning about Binary Topological Relations. In *Second Symposium on Large Spatial Databases (SSD'91)*, pages 143–160, 1991.

[Ege93] Max Egenhofer. What's Special about Spatial: Database Requirements for Vehicle Navigation in Geographic Space. In *Proc. of ACM/SIGMOD Conf.*, pp. 398–402, 1993.

[End72] H. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 1972.

[GP92] J. Glasgow and D. Papadias. Computational Imagery. *Cognitive Science*, 16:355–394, 1992.

[GS95] R-H. Guting and M. Schneider. Realm-Based Spatial Data Types: The ROSE Algebra. *The VLDB Journal*, 4:243–286, 1995.

[Gut94] Ralf H. Guting. An Introduction to Spatial Database Systems. *The VLDB Journal*, 3(4):357–399, 1994.

[Had95] Thanasis Hadzilacos. On Layer-Based Systems for Undetermined Boundaries. In P.A. Burrough and A. Frank, editors, *Geographic Objects with Undetermined Boundaries*. Taylor & Francis, 1995.

[Hay85] Pat Hayes. The Second Naive Physics Manifesto. In J. Hobbs, R. Moore: *Formal Theories of the Commonsense World*. Ablex Publishing, 1985.

[Her92] Daniel Hernandez. *Qualitative Representation of Spatial Knowledge*. PhD thesis, Faculty of Informatics, Technical University of Munich, 1992.

[KKR90] P. Kanellakis, G. Kuper, and P. Revesz. Constraint Query Languages. Long version. Also appearred in *Proc. of 9th ACM PODS*, 1990.

[KL91] H. Kautz and P. Ladkin. Integrating Metric and Qualitative Temporal Reasoning. In *Proceedings of AAAI-91*, pages 241–246, 1991.

[Kou92] Manolis Koubarakis. Dense Time and Temporal Constraints with $\neq$. In *Proceedings of KR-92*, pages 24–35, 1992.

[Kou94a] Manolis Koubarakis. Database Models for Infinite and Indefinite Temporal Information. *Information Systems*, 19(2):141–173, March 1994.

[Kou94b] Manolis Koubarakis. *Foundations of Temporal Databases.* PhD thesis, National Technical University of Athens, 1994.

[LM88] P. Ladkin and R. Maddux. Representation and Reasoning with Convex Time Intervals. Technical Report KES.U.88.2, Kestrel Institute, 1988.

[Mac77] A.K. Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8(1):99–118, 1977.

[MBJK90] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis. Telos: A Language for Representing Knowledge About Information Systems. *ACM Transactions on Office Information Systems*, 8(4):325–362, 1990.

[MCP+96] J. Mylopoulos, V. Chaudhri, D. Plexousakis, A. Shrufi, and T. Topaloglou. Implementing Knowledge Management Systems . *The VLDB Journal (to appear)*, 1996.

[MP94] C.B. Medeiros and F. Pires. Databases for GIS. *ACM SIGMOD Record*, 23:107–115, 1994.

[OM88] J. Orenstein and F. Manola. PROBE: Spatial Data Modelling and Query Processing in an Image Database Application. *IEEE Transactions on Software Engineering*, 14(5):611–629, 1988.

[Pap94a] Dimitris Papadias. *Relation-based Representations for Spatial Databases.* PhD thesis, National Technical University of Athens, 1994.

[Pap94b] Christos Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[PD95] E. Puppo and G. Dettori. Towards a formal model for multiresolution spatial maps. In *Proc. of SSD'95*, pages 153–169, 1995.

[PS94] D. Papadias and T. Sellis. Qualitative Representation of Spatial Knowledge in Two-Dimensional Space. *The VLDB Journal*, 3(4):479–517, 1994.

[PTSE94] D. Papadias, Y. Theodoridis, T. Sellis, and M. Egenhofer. Topological Relations in the World of Minimum Bounding Rectangles: A Study with R-trees. In *SIGMOD-95*, 1994.

[RFS88] N. Roussopoulos, C. Faloutsos, and T. SellisAn Efficient Pictorial Database System for PSQL. *IEEE Transactions on SE*, 14(5):639–650, 1988.

[RH80] D. Rozenkrantz and H. Hunt. Processing Conjunctive Predicate and Queries. In *VLDB-80*, pages 64–72, 1980.

[RS95] P. Rigaux and M. Scholl. Multi-Scale Partitions: Applications to Spatial and Statistical Databases. In *Proc. of SSD'95*, pages 171–183, 1995.

[SYH94] P. Sistla, C. Yu, and R. Haddad. Reasoning about Spatial Relationships in Picture Retrieval Systems. In *VLDB-94*, 1994.

[Top94] Thodoros Topaloglou. First Order Theories of Approximate Space. In *AAAI-94 Workshop on Spatial and Temporal Reasoning*, pages 47–53, 1994.

[Top96] Thodoros Topaloglou. *On the Representation of Partial Spatial Information in Knowledge Bases.* PhD thesis, Department of Computer Science, University of Toronto, 1996.

[vB90] Peter van Beek. Exact and Approximate Reasoning About Qualitative Temporal Relations. TR/CS 90-29, University of Alberta, 1990.

[WJS93] X.S. Wang, S. Jajodia, and V.S. Subrahmanian. Temporal Modules: An Approach Towards Federated Temporal Databases. In *Proceedings of ACM/SIGMOD Conference*, pages 227–236, 1993.

This article was processed using the LaTeX macro package with LLNCS style