# Spatial Databases with Partial Information: Representation and Reasoning

**Thodoros Topaloglou** and **John Mylopoulos**
{thodoros,jm}@cs.toronto.edu
Department of Computer Science
University of Toronto
Toronto, Ontario, M5S 1A4, Canada

June 8, 1995

**Abstract**

The paper proposes a novel 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. The paper also identifies four basic reasoning tasks required for query processing operations and presents an algorithmic complexity assessment for each task.

**Keywords:** Spatial Databases, Knowledge Base Systems, Constraint-Based Data Modeling.

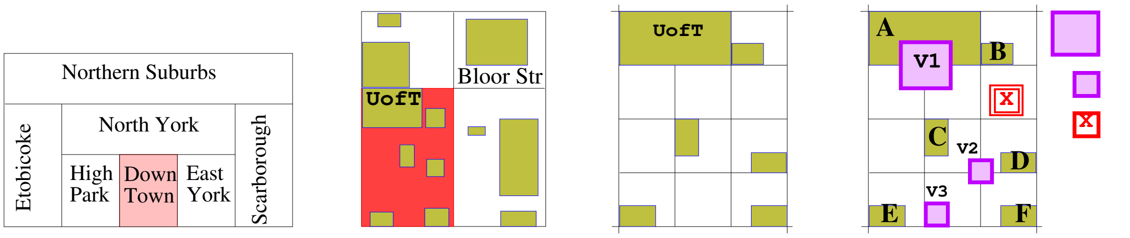
# 1 Introduction

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 [GFA92], 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 efficient representations of *complete* spatial information (see for example, [OM88], [RFS88], [Gut88]). The second category includes models that can accommodate and reason with *partial* spatial information (e.g., [GP92], [Her92]) but, by-and-large, ignore quantitative spatial information and performance concerns. [1]

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

[1] Of course, there are notable exceptions to this broad-brush classification, such as [Dav86].




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

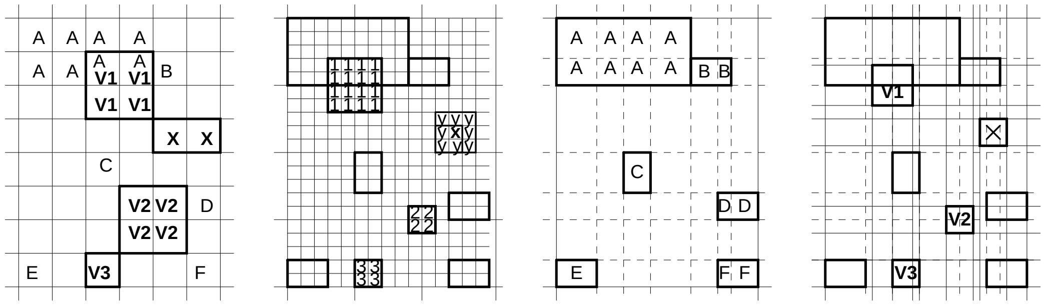
Figure 1: An example map at different scales

region. Moreover, indeterminate spatial objects are specified in varying degrees of precision, while for some of them only qualitative relationships ($<, \leq$ or $=$) may be known.

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. The different drawing styles distinguish between landmark objects and indeterminate objects. Figure 1(d) shows three vehicles and their scopes, along with a trouble spot (denoted by 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 $V_1$ and $V_2$".

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 a conceptual modeling language, while section 4 presents a formalization of the data model. Section 5, discusses related work, the expressiveness of the data model with examples of spatial queries, and implementation considerations. Finally, section 6, summarizes the contributions of the paper and briefly discusses future research.




(a) (b) (c) (d)

grid representations object-dependent representations

Figure 2: Representation of landmark and indeterminate spatial objects

# 2 Granularity and Haze 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 $\alpha$ at scale $s_i$ there exists a container object $container(\alpha)$ at scale $s_{i-1}$ that contains $\alpha$, 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 contains an indeterminate spatial object (point or region) and specifies 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 in given 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 develops 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 $\mathcal{L}$. $\mathcal{L}$ can be any of the two languages defined in section 4, namely, the language of qualitative and quantitative constraints on 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. Here are some examples:

- $close(V2,$ "University Ave. and Queen St.")
- $south\_side(V1,$ "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-occuring 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


one−dimensional envelope

two−dimensional envelope

Figure 3: Spatial Envelopes

two one-dimensional envelopes. A one-dimensional envelope constrains the exact position of an indeterminate spatial interval.

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 3 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. For example, figure 2 shows several maps at various grains of scale *division* representing the same block of space. Specifically, $L = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}\}$, $I = \{\mathtt{V}_1, \mathtt{V}_2, \mathtt{V}_3, \mathtt{X}\}$ and $m_1 = \langle L, I, C, division, g_{m_1}\rangle$, $m_2 = \langle L, I, C, division, g_{m_2}\rangle$, $m_3 = \langle L, I, C, division, g_{m_3}\rangle$ are maps corresponding to figures 2(a), (b) and (d), respectively. The set of constraints, $C$, is explained below.

Every 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 map corresponding to figure 2(b) satisfies this condition since $haze(\mathtt{V}_1) = h_1$, $haze(\mathtt{V}_2) = haze(\mathtt{V}_3) = h_2$, $haze(\mathtt{X}) = h_x$, $g_{m_2} < h_2 < h_1 = h_x$. The map shown on figure 2(a) is an example of a map which violates this condition.

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 2.2.1** 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, $A_{S_X}, A_{E_X}, A_{S_Y}$ and $A_{E_Y}$. Let $g_m$ be the grain of the map. Then, with the help of an appropriate "cutting" function, we construct the array representation, say this is the array of figure 2(b). The following two sets of equality constraints define $X$ and $Y$ axis projections of this array:

$\{E_{S_X}=0, A_{S_X}=0, E_{E_X}=3, C_{S_X}=5, C_{E_X}=7, A_{E_X}=9, B_{S_X}=9, F_{S_X}=11, D_{S_X}=11, B_{E_X}=12, D_{E_X}=15, F_{E_X}=15\}$

$\{E_{S_Y}=0, F_{S_Y}=0, E_{E_Y}=2, F_{E_Y}=2, D_{S_Y}=5, D_{E_Y}=7, C_{S_Y}=7, C_{E_Y}=10, A_{S_Y}=15, B_{S_Y}=15, B_{E_Y}=17, A_{E_Y}=20\}$

$\Box$

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 2.2.2** 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. $0 \le \mathtt{V1X}_{\mathtt{S_U}} - \mathtt{V1X}_{\mathtt{S_L}} \le 2g_1$
2. $0 \le \mathtt{V1X}_{\mathtt{E_U}} - \mathtt{V1X}_{\mathtt{E_L}} \le 2g_1$
3. $\mathtt{V1X}_{\mathtt{S_L}} \le V1X_S \le \mathtt{V1X}_{\mathtt{S_U}}$
4. $\mathtt{V1X}_{\mathtt{E_L}} \le V1X_E \le \mathtt{V1X}_{\mathtt{E_U}}$
5. $V1X_E - V1X_S = c_1$
6. $0 \le \mathtt{V1Y}_{\mathtt{S_U}} - \mathtt{V1Y}_{\mathtt{S_L}} \le 2g_1$
7. $0 \le \mathtt{V1Y}_{\mathtt{E_U}} - \mathtt{V1Y}_{\mathtt{E_L}} \le 2g_1$
8. $\mathtt{V1Y}_{\mathtt{S_L}} \le V1Y_S \le \mathtt{V1X}_{\mathtt{S_U}}$
9. $\mathtt{V1Y}_{\mathtt{E_L}} \le V1Y_E \le \mathtt{V1X}_{\mathtt{E_U}}$
10. $V1Y_E - V1Y_S = c_1$
11. $(E_{E_X} = \mathtt{V1X}_{\mathtt{S_L}}) \wedge (\mathtt{V1X}_{\mathtt{S_U}} < C_{S_X})$,
12. $(C_{E_Y} < \mathtt{V1Y}_{\mathtt{S_L}}) \wedge (\mathtt{V1Y}_{\mathtt{S_U}} < A_{S_Y})$,
13. $(C_{S_X} < \mathtt{V1X}_{\mathtt{E_L}}) \wedge (\mathtt{V1X}_{\mathtt{E_U}} = C_{E_X})$
14. $(B_{E_Y} = \mathtt{V1Y}_{\mathtt{E_L}}) \wedge (\mathtt{V1Y}_{\mathtt{E_U}} < A_{E_Y})$

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 a spatial object which participates in one or more maps.

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 `PhysicalObjectClass` is a metaclass whose instances include physical object classes such as the class of vehicles, the class of buildings or parks. All these classes are also specializations of `PhysicalObject`, which is also an instance of `PhysicalObjectClass`. In addition, we introduce the metaclass `SpatialObjectClass`, whose instances are spatial object classes such as `Street`, `Lot`, `Parcel` etc. These classes are also specializations of `SpatialObject`, which is an instance of `SpatialObjectClass`. Figure 4 illustrates this class hierarchy. Physical objects can have an associated spatial object, about which information is represented in terms of one or more maps.

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

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

```
CLASS SpatialObject IN SpatialObjectClass
  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 `PhysicalObjectClass`. 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. [2]

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

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

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

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. [3] 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`.

```
CLASS Map IN SpatialObjectClass
  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. [4] Additional integrity constraints specify properties that any `Map`

[3] 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.

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

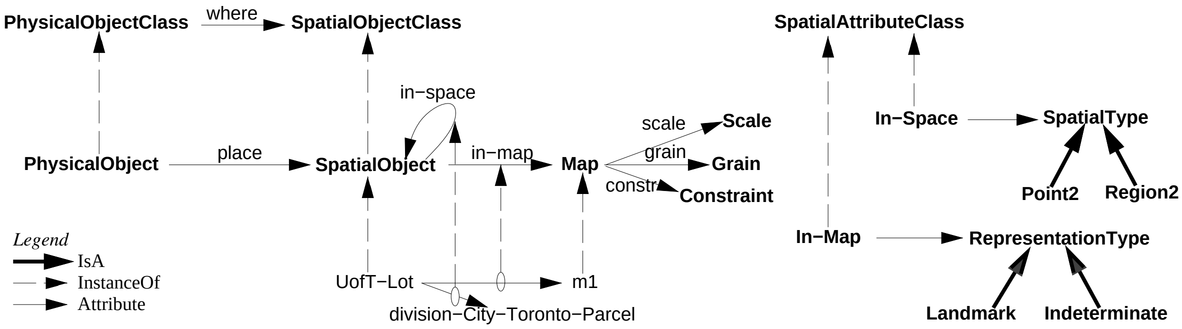



Figure 4: Modeling Spatial Information in Telos

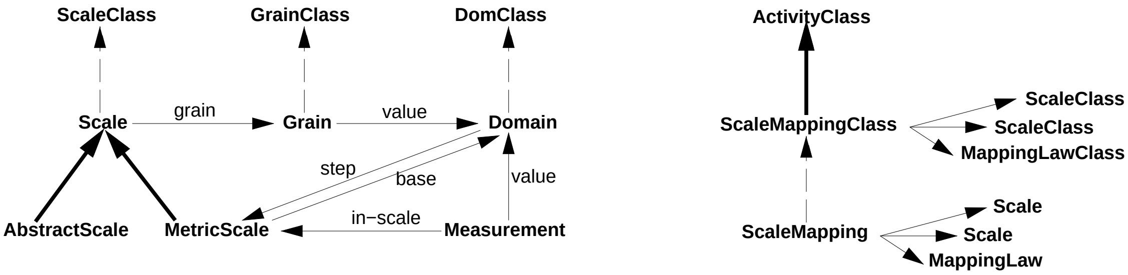



Figure 5: Modeling Scale in Telos

instance needs to satisfy.

Granularity and scale is modeled in a similar fashion. Figure 5 illustrates graphically the Telos formalization.

```
CLASS MetricScale ISA Scale WITH          CLASS Measurement IN S_CLASS WITH
   attribute                                 attribute, single
      grain:Domain                              value: Domain
      base :Domain                              inscale: MetricScale
      unit :Domain                          END Measurement
   integrityConstraint
      :in(X,MetricScale)==>(X.unit=X.grain.value)
END MetricScale
```

# 4 Formalization

This section presents a formalization of the proposed model of space 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 Formal 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, $H$ (for hazy points) and $E$ (for exact points); $x, y, z, ..$ are variables of sort $H$, $c_1, c_2, c_3, ..$ are constants (uninterpreted integers) of sort $E$, and $g$ is a designated constant also of sort $E$. The non-logical symbols of $\mathcal{L}_\mathcal{H}$ include the predicate symbols $\approx$, $\prec$ relating objects of sorts $H \times H$ and $H \times E$, the predicate symbols $=$, $<$ relating objects of sorts $E \times E$, and 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 defined in the standard way, i.e., they 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 $\mathtt{g}$. Each constant $c_i$ of the sort $E$ is interpreted as the integer $\mathtt{i}$. 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. 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.

Formally, the domain of integers, $\mathbb{Z}$, along with the relations $R, S, <, =$ and function $-$ define an $\mathcal{L}_\mathcal{H}$-structure. The notions of truth and satisfaction are defined in the usual way [End72], that is, if $\mathbf{M}$ is an $\mathcal{L}_\mathcal{H}$-structure, $\phi$ is a formula of $\mathcal{L}_\mathcal{H}$, and $v$ is an assignment of variables of $\phi$ into elements of the domain of $\mathbf{M}$, then $\mathbf{M} \models \phi[v]$ means that $\mathbf{M}$ *satisfies* $\phi$ under variable assignment $v$. The first order theory of the qualitative and quantitative constraints expressed by $\mathcal{L}_\mathcal{H}$, is the set of $\mathcal{L}_\mathcal{H}$ sentences which is closed under logical implication.

The reader should note 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 $g$ is assigned 0 in an interpretation, 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. In fact, this is the language used for the spatial envelope constraints, since the envelopes have already

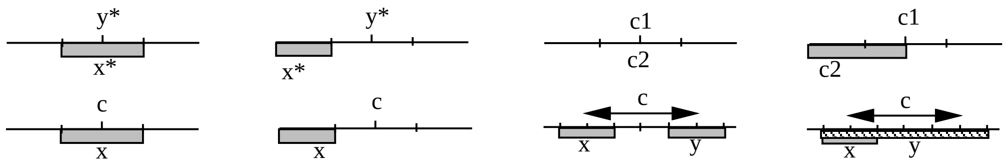



Figure 6: Graphical interpretation of $\mathcal{L}_{\mathcal{H}}$ terms

"compiled away" spatial indeterminacy.

**Example 4.1.1** Figure 6 illustrates graphically the meaning of the terms of $\mathcal{L}_{\mathcal{H}}$. Assume that g=2 throughout this example. If an interpretation $v$, assigns $y$ to $\mathtt{y}^*$=6, then 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 y, i.e., x can be one of {2,3,4,5,6} if $\mathtt{y} = \mathtt{4}$ and so on. The $\mathtt{x}^*, \mathtt{y}^*$ notation is used in the figure to show this relationship.

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 haze 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 i=8 then x can be one of {6,7,8,9,10}. Analogously, in the same model, $x$ for $x \prec c_8$ is interpreted as an integer which is less than 6.

The subtraction operator "$-$" is useful in defining a notion of distance between haze points; "$-$" returns an exact quantity. [5] Let, for instance, x be interpreted in {1,2,3,4,5} and y in {7,8,9,10,11}, the term $y - x = c_6$ means that x and 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 $x, y$. □

## 4.2 From Points to Regions

The language of one-dimensional hazy-point space is now extended to a language where one can state relationships between two-dimensional hazy points and rectangles. This is accomplished by using two "copies" of $\mathcal{L}_{\mathcal{H}}$, with disjoint sets of symbols, each one corresponding to one of the two coordinate axes of the Cartesian plane. These copies are called $\mathcal{X}$ and $\mathcal{Y}$, and their combination $PR2$. In addition, we introduce two pairing functions:

$\langle x_1, x_2 \rangle$ it is a pairing function which returns the two-dimensional point formed by its one-dimensional coordinates;

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

$[p, q]$ is the pairing function which returns the rectangle which is formed by the two-dimensional points $p, q$.

The following four operators define the inverse operations of the pairing functions:

$s(R)$ (alternatively $R_S$) returns the lower and leftmost point of a rectangle $R$;

$e(R)$ (alternatively $R_E$) returns the topmost and rightmost point of a rectangle $R$;

$x(P)$ (alternatively $P_X$) returns the horizontal projection of a two-dimensional point $P$;

$y(P)$ (alternatively $P_Y$) returns the vertical projection of a two-dimensional point $P$.

Consequently, $R = [R_S, R_E]$, $P = \langle P_X, P_Y \rangle$ and $R = [\langle R_{S_X}, R_{S_Y} \rangle,\ \langle R_{E_X}, R_{E_Y} \rangle]$, where $P$ and $R$ are a two-dimensional point and rectangle respectively.

A number of translation functions are also needed:

- $r2p$: translates rectangles to points in $PR2$ formulas.
- $p2x$: projects points to their $\mathcal{X}$ coordinates in $PR2$ formulas.
- $p2y$: projects points to their $\mathcal{Y}$ coordinates in $PR2$ formulas.

The technique applied for the points-to-regions extension, is called *independent combination*. The formal properties of independent combination as well as a syntactic procedure for obtaining $p2x$ and $p2y$, are presented in [Top94]. The application of the independent combination rule on $\mathcal{X}$ and $\mathcal{Y}$, guarantees that every sentence in the resulting $PR2$ language has the following syntactic form:

$$Qx_{1_X} \ldots x_{i_X}/H_X, Qx_{1_Y} \ldots x_{i_Y}/H_Y(x_1 = \langle x_{1_X}, x_{1_Y} \rangle \wedge \ldots \wedge x_i = \langle x_{i_X}, x_{i_Y} \rangle \wedge p2x(\Phi) \wedge p2y(\Phi))$$

where $Q$ is either $\exists$ or $\forall$, $H_X$ (resp. $H_Y$) denotes the sort of $\mathcal{X}$ (resp. $\mathcal{Y}$) coordinate variables, and $\Phi$ is a $PR2$ formula. $\bigwedge_i x_i = \langle x_{i_1}, x_{i_2} \rangle$ is called the pairs-part. The intuition behind independent combination is that any operator between two-dimensional spatial objects can be written as a conjunction of two one-dimensional formulas. The following theorem relates the satisfiability of a $PR2$ formula with the satisfiability of its $\mathcal{X}$ and $\mathcal{Y}$ translations.

**Theorem 1** *Let $\phi$ be a formula of $PR2$ and pair variables $x_i = \langle x_{i_1}, x_{i_2} \rangle, i = 1..n$. If $v_1$ and $v_2$ are variable assignments for $x_{i_1}$'s and $x_{i_2}$'s in some models $M_1$ and $M_2$, of $\mathcal{X}$ and $\mathcal{Y}$, respectively,* *and $M$ is the composite* [6] *model of $M_1$ and $M_2$, then*

$$M \models \phi[v] \text{ iff } M_1 \models p2x(\phi)[v_1] \text{ and } M_2 \models p2y(\phi)[v_2]$$

*with* $v = \{\langle x_1, x_2\rangle | x_1 \in v_1 \text{ and } x_2 \in v_2\}$

This theorem establishes that independent combination can generate a calculus on two-dimensional spatial objects which has the computational properties of its one-dimensional coordinate calculi. Moreover, we can compose a solution for a two-dimensional constraint satisfaction problem by combining solutions of its coordinate problems. Of course, such a restricted form of combination imposes limitations to the expressiveness of the constraint language for the two-dimensional space. An already mentioned, a major limitation is that the two-dimensional spatial objects are limited to points and rectangles only. Fortunately, 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]. In the next section, we explore sets of spatial relations that are expressible in our formalism.

### 4.3 Spatial Relations

Egenhofer's proposal [Ege91] of eight fundamental topological relations for two planar regions is the most popular set of topological spatial relations. One advantage of his proposal is its clean topological semantics. In our earlier work [Top94], we have shown an alternative semantics for the above relations based on the ontology of hazy points. The difference in 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 7 shows a graphical presentation of topological relations. Their formal definition requires the definitions of the helping relations $in_0, in_1$ and *close*.

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

[6] The construction of $M$ in terms of $M_1$ and $M_2$ is described in [Top94]

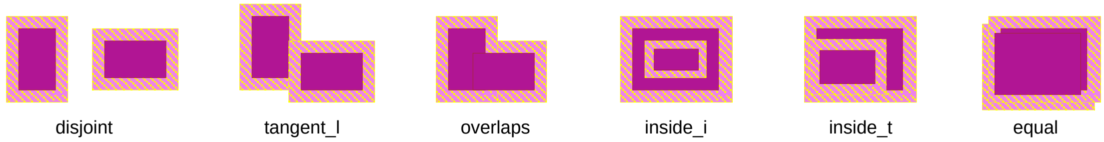



Figure 7: Hazy topological relations

$P, P1$ and $P2$ are points and $R1, R2$ rectangles. [7]

$$
\begin{array}{lcl}
\texttt{disjoint}(R1,R2) & \equiv & \neg\exists P(in_0(P,R1) \wedge in_0(P,R2)) \\
\texttt{tangent}_\texttt{l}(R1,R2) & \equiv & \exists P1,P2(in_0(P1,R1) \wedge in_0(P2,R2) \wedge close(P1,P2)) \\
\texttt{overlap}(R1,R2) & \equiv & \exists P(in_0(P,R1) \wedge in_0(P,R2)0 \\
\texttt{inside}_\texttt{i}(R1,R2) & \equiv & in_1(R1,R2) \\
\texttt{inside}_\texttt{t}(R1,R2) & \equiv & (R2_{S_X} \preceq_X R1_{S_X} \wedge R1_{E_X} \preceq_X R2_{E_X}) \wedge (R2_{S_Y} \preceq_Y R1_{S_Y} \wedge R1_{E_Y} \preceq_Y R2_{E_Y}) \\
\texttt{contain}_\texttt{t}(R1,R2) & \equiv & \texttt{inside}_\texttt{t}(R2,R1) \\
\texttt{contain}_\texttt{i}(R1,R2) & \equiv & \texttt{inside}_\texttt{i}(R2,R1) \\
\texttt{equal}(R1,R2) & \equiv & close(R1_S,R2_S) \wedge close(R1_E,R2_E)
\end{array}
$$

The notion of tangency as defined here 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]; this model comes with a deductive reasoner that is shown to be sound and complete for the three-dimensions and sound only for the two dimensions. We show that the axioms of this deductive reasoner (except one) are theorems of $PR2$. Its incompleteness result does not affect our case since we do not use this rule system for inferencing.

$$
\begin{array}{lcl@{\qquad}lcl}
\texttt{left\_of}(R1,R2) & \equiv & R1_{E_X} \prec_X R2_{S_X} & \texttt{right\_of}(R1,R2) & \equiv & \texttt{left\_of}(R2,R1) \\
\texttt{above}(R1,R2) & \equiv & R2_{E_Y} \prec_Y R1_{S_Y} & \texttt{below}(R1,R2) & \equiv & \texttt{above}(R2,R1)
\end{array}
$$

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 all of them. Instead, we list selective examples:

[7] Notation: $x_i \preceq_i y_i \equiv x_i \prec_i y_i \vee x_i \approx_i y_i$. Subscript $_i$ denotes the projection coordinate. $_i$ is either $_X$ or $_Y$.

- one-dimensional, interval-to-interval relation

  $overlaps_i(I_i, J_i) \equiv I_{S_i} \prec_i J_{S_i} \wedge J_{S_i} \prec_i I_{E_i} \wedge I_{E_i} \prec_i J_{E_i}$

- two-dimensional, point-to-point relation

  $north\_east(P, Q) \equiv Q_Y \prec_Y P_Y \wedge P_X \prec_X Q_X$

- two-dimensional, point-to-region relation

  $on\_left\_side(P, A) \equiv P_X \approx_X A_{S_X} \wedge A_{S_Y} \prec_Y P_Y \wedge P_Y \prec_Y A_{E_Y}$

## 4.4 Reasoning

A map structure is represented now 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], [DMP89], [KL91], [Kou94b] (and the lists goes long). 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. Our guess is that even better performance can be achieved if one explores the structure and especially the sparsity of constraint sets. The second reasoning problem, is closely related to the first one. In fact, a solution for **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 is attempting to identify efficient special cases for the last problem.

**Theorem 2** *For a representation of the map structure using linear order constraints and constant grain, reasoning problems* **P1** *and* **P2** *are solved in* $O(n^3)$ *time.*

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* $i$*.* [8]

Algorithmically, **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**. For the first step, the following linear time procedure applies. There are two directions to which the representation's grain can change

[8] Recall that parameters of an indeterminate are constrained by its envelope and the size constraints. Problem **P3** calls for the determination of tight bounds, i.e., the smallest interval that a parameter can take values from, for the parameters of a specific indeterminate.

- *refinement* ($g' < g$): Let $r$ be the refinement factor, $r = \frac{g}{g'}$ (assume $g'$ divides $g$). Then each $P = v$ conjunct in the $Xcon$ and $Ycon$ constraint set is replaced by the constraint $(r * (v - 1) \leq P) \wedge (P \leq r * v)$.
- *coarsening* ($g' > g$): Let $s$ be the coarseness factor, $s = \frac{g'}{g}$ (assume $g$ divides $g'$). Replace each term $P = v$ with $P = \frac{v}{s}$.

There are various semantic issues that our quick coverage of the transition operations has overlooked. [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 discussing related work, the expressiveness of the data model and last, but not least, some implementation issues.

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]. To the best of our knowledge, there are no other proposals for spatial data model which cover as wide a range of features, such as indeterminate spatial objects, multiple scales, granularity and spatial relations. Related work for each separate feature has already been given in previous sections.

Although our spatial data model may be atypical of other proposals, it still complies with general principles of spatial data models. For example, a recent survey, [Gut94], identifies four different approaches in representing space in a database. Our spatial model matches features from three of these categories, 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.

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.

To illustrate the expressiveness of the proposed spatial data model, consider the following query examples.

*1. Querying Spatial Relations with Granularity.* Assume the query: *Find if the CS Dept. is inside the scope of* $V1$. This query may be initially evaluated with the highest possible precision (lowest grain), i.e., $g_q = 0$. Then the query is whether the CS Dept., an exact point with coordinates, say, $\langle 5, 16\rangle$, is inside the scope of $V1$. [9]

$$Q1_0 = \{true | P = \langle 5, 16\rangle \wedge g_q = 0 \wedge (inside(P, V1, g_q) \vee on\_*\_side(inside(P, V1, g_q))\}$$

This query is an instance of problem **P3** for $V1$ followed by the evaluation of the query predicates. As there are more than one database models describing a consistent set of parameters for $V1$, this query is ambiguous and can be answered with respect to one or all possible worlds (see [Kou94a] for possible world semantics in constraint-based query processing). The existence of a grain notion in our representation makes the answers to these two problems dependent on the grain size of the query or the representation. Below, we ask the same query assuming indeterminate position for the CS Dept.

$$Q1_g = \{true | P = \langle 5, 16\rangle \wedge g_q = 1 \wedge (inside(P, V1, g_q) \vee on\_*\_side(inside(P, V1, g_q))\}$$

The implication of using an imprecise search point is to extend the selectivity of the query predicates. This a done by compiling their expressions for $g_q = 1$. One can obtain another variant of the query by changing the precision of $V1$. This means that its envelope constraints have to be recompiled, leading to new instances of problems **P1** and **P2**, i.e., the consistency and minimality of the map must be verified for the new envelope constraints. Then the query is evaluated following the steps described above. Other spatial relation-based queries can be expressed, and evaluated, along the same lines.

*2. Spatial Join as Constraint Merging Operation.* Assume that we are given a map of Toronto hospitals and wish to find which hospital is close to the trouble spot. This involves adjusting the scales, recomputing the constants (problem **P4**), merging the constraint sets of the two, and proceeding with problems **P1** and **P2** (i.e., verify the consistency of the merged map) and evaluation of the predicate $close(\mathtt{X}, H)$. A number of optimizations are possible in this case, especially if we exploit the inclusion/nesting property of the scale hierarchy.

An implementation of the proposed data model could be achieved by first defining the non-spatial concepts using an existing data model (e.g., the one presented in section 3 or as relations in an object-based or relational system) and then implementing a constraint-based inference engine

[9] $on\_*\_side$ abbreviates $on\_left\_side \vee on\_right\_side \vee on\_top\_side \vee on\_bottom\_side$.

for spatial constraints.

To-date, there is no generally-accepted way for implementing large constraint databases. Logic programming and constraint logic programming are two obvious alternatives with respective limitations regarding their scalability for large workloads. Constraint-based reasoners is another option. We are currently exploring 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 and small maps. The integrated system is then put together using the Telos knowledge base management system. It should be added that many implementations of Telos are currently available, including ConceptBase [Jar92], SIB [CJMV95] and the Telos Repository [Sta95].

# 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.

The proposed spatial data model is unique in the combination of features that it accommodates. Moreover, we believe that the model could be of use in non-spatial applications as well, particularly ones involving dimensional data, such as temporal databases, genome databases and financial databases.

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

- A thorough case study using an application domain with a wide range of granular 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 [CM93], 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.

[CJMV95] P. Constantopoulos, M. Jarke, J. Mylopoulos, and Y. Vassiliou. The Software Information Base: A Server for Reuse. *The VLDB Journal*, To appear, 1995.

[CM93] P. Consens and A. Mendelzon. Hy+: A hygraph-based query and visualization system. In *Proceedings of ACM SIGMOD Conference*, pages 511–516, 1993.

[Dav86] Ernest Davis. *Representing and Acquiring Geographic Knowledge*. Pitman and Morgan Kaufmann Publishers Inc., 1986.

[DMP89] R. Dechter, I. Meiri, and J. Pearl. Temporal Constraint Networks. In *Proceedings of 1st Knowledge Representation and Reasoning Conference*, pages 83–93, 1989.

[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 *Proceedings of ACM SIGMOD Conference*, pages 398–402, 1993.

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

[GFA92] J. Glasgow, S. Fortier, and F. Allen. Molecular Schene Analysis: Crystal Structure Determination through Imagery. In L. Hunter, editor, *Artificial Intelligence and Molecular Biology*. AAAI Press, 1992.

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

[Gut88] Ralf H. Guting. Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems. In *Proceedings of EDBT'88*, pages 506–527, 1988.

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

[Hay85] Pat Hayes. The Second Naive Physics Manifesto. In J. Hobbs and R. Moore, editors, *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.

[Jar92] M. Jarke. ConceptBase V3.1 User Manual. Technical Report 92-17, University of Aachen, 1992.

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

[KL91] Henry Kautz and Peter 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] Peter Ladkin and Roger Maddux. Representation and Reasoning with Convex Time Intervals. Technical Report KES.U.88.2, Kestrel Institute Technical Report, 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.

[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.

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

[RFS88] Nick Roussopoulos, Christos Faloutsos, and Timos Sellis. An Efficient Pictorial Database System for PSQL. *IEEE Transactions on Software Engineering*, 14(5):639–650, 1988.

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

[Sta95] Martin Stanley. The Telos Repository User's Manual. University of Toronto, Dept. of Computer Science, Forthcoming, 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.

[vB90] Peter van Beek. Exact and Approximate Reasoning About Qualitative Temporal Relations. Technical Report TR 90-29, Department of Computing Science, University of Alberta, August 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.