FEATURE INFORMATION SUPPORT AND THE SDTS CONCEPTUAL DATA MODEL: CLARIFICATION AND EXTENSION

Leone Barnett & John V. Carlis
Department of Computer Science, University of Minnesota
Minneapolis, Minnesota 55455

ABSTRACT

The Spatial Data Transfer Standard (SDTS) includes a definition of spatial objects and spatial features presented as part of what the standard calls a "conceptual model." Using a precise data modeling notation, we developed a Logical Data Model (LDS) of the SDTS spatial object. Unfortunately, the SDTS description of features, and their relationship to spatial objects, is too ambiguous to model precisely. However, once an interpretation of key concepts is assumed, that information can be modeled. We offer our interpretation, and use that as the basis for an extended model of real world features and their relation to spatial objects. The analysis and the resulting model are useful for designing a system that more effectively manages the "meaning" of its spatial objects.

INTRODUCTION

The Spatial Data Transfer Standard (National Institute of Standards and Technology 1992) presents a conceptual model of spatial objects, real world phenomena, and their relationship. We attempted to produce a data model of this with Logical Data Structure (LDS) notation, which is a precise specification that can be mapped directly to system implementations. The spatial object description was clear enough to model directly. However, the sections describing features and real world phenomena require some interpretation before a model can be specified. Therefore we present two LDS models: one reflects the SDTS spatial object description, and the other offers an extended view of the relationship between spatial objects and the real world features they represent. Our extended view makes certain types of information explicit, which is important for developing systems that will be capable of more sophisticated spatial data management. We use LDS notation for our models, so a brief introduction to it is provided next.

A Logical Data Structure (Carlis 1993) is a graphical diagram that names types of data and relationships to be remembered. An LDS is formed from basic structures called entities, attribute descriptors, relationships, relationship descriptors, and identifiers. Data modeling with LDS diagrams is a technique that supports precise data definition and clear communication of what data types are of interest. An LDS entity is displayed as a box, with its name in the top section. An LDS entity names a type of data. An LDS entity is described by a relationship descriptor when there is a relationship between the entity and another LDS entity. An attribute of an LDS entity appears as text inside the box for the entity. A relationship, depicted as a line connecting two boxes, may be one-to-one, one-to-many, or many-to-many. This indicates, for example in the one-to-many case, that one instance of entity-1 has relates to zero or more instances of entity-2.

Because the SDTS uses the term "entity" in a sense that is not defined the same as the "LDS entity," we must distinguish our usage. For this paper, when we mean to refer to "one of the boxes in an LDS model" we say "LDS entity." When we leave off "LDS" as a qualifier, we are using the term in the conceptual sense of the SDTS, or in our interpretation of same.

THE SDTS

Overview of SDTS

The Spatial Data Transfer Standard (SDTS) is a collective effort to standardize the transfer of spatial data. It has been designed for the transfer of data that are structured and used by scientists who use spatial data. According to USGS Fact sheet, Aug. 1992, it "provides an exchange mechanism for the transfer of spatial data between dissimilar computer systems." The SDTS was approved as a Federal Information Processing Standard (FIPS) 173, in July 1992.

The SDTS was developed to facilitate the sharing and transfer of spatial data across systems and users. As such, the concepts, ideas, and structures in the standard strongly reflect current systems and commonly used models for spatial data. To be generally relevant, it attempts to relate to GIS in general. However it is not a trivial problem to develop a standard that has general applicability and acceptance when use of terms and agreement on the best ways to conceptualize spatial issues is lacking. Spatial data models have long been an important topic in the GIS community, but terminology associated with the various models is not always consistent.

For example, Peuquet (Peuquet 1990) discusses two types of spatial data models most commonly used in computer systems. She calls them vector and tessellation models. Vector models use points, lines, and polygons to represent real world features or concepts (such as rivers, contour lines, political boundaries, etc.). Tessellation models involve methods of breaking up space into pieces, which may have regular, irregular, or nested divisions. Egenhofer and Herring (Egenhofer & Herring 1991) also describe different types of spatial data models, but their two most important groups of models are called regular tessellations and irregular tessellations of space. They describe vector structures as a means of implementing an irregular tessellation, and grid cell raster structures as a means of implementing a regular tessellation. This illustrates that a common term, i.e. "vector," is used to refer to a model in one case, and to a mechanism for implementing a model in another case.

The lack of consistency in how people think about spatial data models is due, in part, to the tendency to mix "what" and "how" together. That is, what is being modeled is mixed with terminology that describes how different structures are used to do the modeling. With spatial models, sometimes space itself is modeled, and sometimes things in space are modeled. Because they both have spatial structure, it is possible, if not always convenient, to use any of the methods for either purpose. For example, vector models are commonly, if not always, used when the primary task is the representation of real world entities (especially geographic ones). The real world entities have a position in space, but the space itself is assumed rather than modeled. This necessitates explicit modeling of spatial characteristics of the real world entity, often referred to as its topology. In addition, the location of the real world entity modeled might be viewed as an attribute. Regular tessellation models, such as the raster grid, are used when a primary task is the modeling space. Space is modeled by breaking it up into contiguous pieces. Qualities associated with the space, or things contained in the space, are attached secondarily as attributes of the space.

The SDTS does not resolve or even address these differences. Rather, it describes a spatial object that is based on the most commonly used and understood spatial models - the geometric or vector model, and the raster type of regular tessellation model. The spatial object description they offer for the vector model is quite detailed, and it was relatively straight-forward to develop an LDS that illustrates their description. The level of detail in their spatial object description reflects the historical and still prevalent, strong focus on geometric representations of real world phenomena, rather than on the phenomena themselves. This is due to the obvious utility of spatial descriptions for map-like displays and for storage of spatial data.

The detail of the SDTS spatial object description stands in contrast to the level of detail used to describe another concept, which they refer to as the "entity object." This concept focuses on the use of the spatial object for representation of real world features, or on the connection between these two, and is a concept that is separate from the spatial object descriptions per se. Their interest in the real world features, as well as their lack of a detailed description, reflects the current and growing interest, but lack of consensus, in how to develop "feature-based" GIS. Such systems promise to provide more of what may be called "semantic information" support. This refers to the system's capability of a) knowing more about data (what they mean, how they are represented, etc.); b) being able to use the data (specialized operators are available); and c) being capable of making decisions as part of processing (interpretive rules are available). We contend that understanding and developing support for the "entity object" is central to enabling a system to capture important semantic information. Because of this, the SDTS spatial object description and entity object description are both examined closely in the sections that follow.

The Spatial Object: Geometry and Topology

The SDTS provides a detailed description of a spatial object, and the development of an LDS graphical data model for this part of the SDTS description was relatively unproblematic. The concept of the spatial object, as described in the SDTS, is precisely modeled with the LDS shown in Figure 1. The next few paragraphs describe the model depicted in this figure. As with the SDTS spatial object description, the most important aspects are the geometric and topological structures.

The description of a spatial object containing the most detail is one that is based on vector spatial data models. The most basic type of spatial object is the point. Figure 1 shows that the LDS entity "Point" is a kind of 0-dimensional object that is uniquely identified by the attribute "NP," which is a point id. It has "x," "y", and optionally "z" attributes that are coordinates used to specify location. The "point-type" attribute is used to indicate the role of the point, which may be as a point feature location, a text display location, or an area feature attribute handle.

Other types of spatial objects are made up of points. A "Line-segment" is a pair of unordered points. A "String" is a sequence of points. An "Arc" is a locus of points that form a curve defined by a math expression. These are all 1-dimensional objects. A "Polygon" is a 2-dimensional object with an "Interior area" and a boundary. A polygon can be a "GT-Poly" or a "G-Poly" which differ with respect to what kind of boundary they have. A "GT-Poly" has a "GT-Ring" for a boundary, which is a closed "Chain" sequence. A "G-Poly" has a "G-Ring" for a boundary, which is a closed "String" or "Arc" sequence. The substantive difference in the two polygon types is whether or not they carry topological information. The "GT-Poly" does and the "G-Poly" does not.

Two kinds of topological data are modeled. One, an adjacency relationship, is modeled via the entity called "Chain-Poly pair elt" (or element). This entity captures the idea that a "Chain" has a relationship with one "GT-Poly" on its right, and another "GT-Poly" on its left. A "GT-Poly" is either on the left or right of each of the chains comprising it. This left or right position is called "face." Since face (or left-ness and right-ness) depend on orientation, a second kind of topological data, direction, must be captured, which is modeled in the LDS entity called "Chain-node ordered elt." A "Chain," comprising a sequence of line segments or arcs, also has two nodes. These nodes are ordered as the first and last, or beginning and end. Direction of a chain moves from the first node to the last node, with the chain's sequence falling in between. Since one node may be associated with more than one chain, the many-to-many relationship between chains and nodes is modeled with "Chain-node ordered elt."

These structures all relate closely to what Peuquet calls the vector spatial data model, which uses geometry to describe the shape of something, and possibly topology to record some spatial relationships. The shapes occur in space, but the space containing them is not specified directly. Rather, it is implicit, and at best derivable. At worst, there is no explicit mapping to, for example, a location on earth. In contrast, a regular tessellation spatial data model makes the description of the containing space explicit. The SDTS offers a limited description of spatial objects which are based on the regular tessellation model. The standard specifies the "Pixel" and the "Grid cell" as 2-dimensional spatial object primitives that can be aggregated into contiguous sets called either a "Digital image" or "Grid" respectively. Both of these are types of "Raster" data. With these spatial objects, the containing space is explicitly defined, but shapes, objects, or other descriptions of the things in space must be derived.

Despite the lack of equal attention given to all types of spatial models (others are

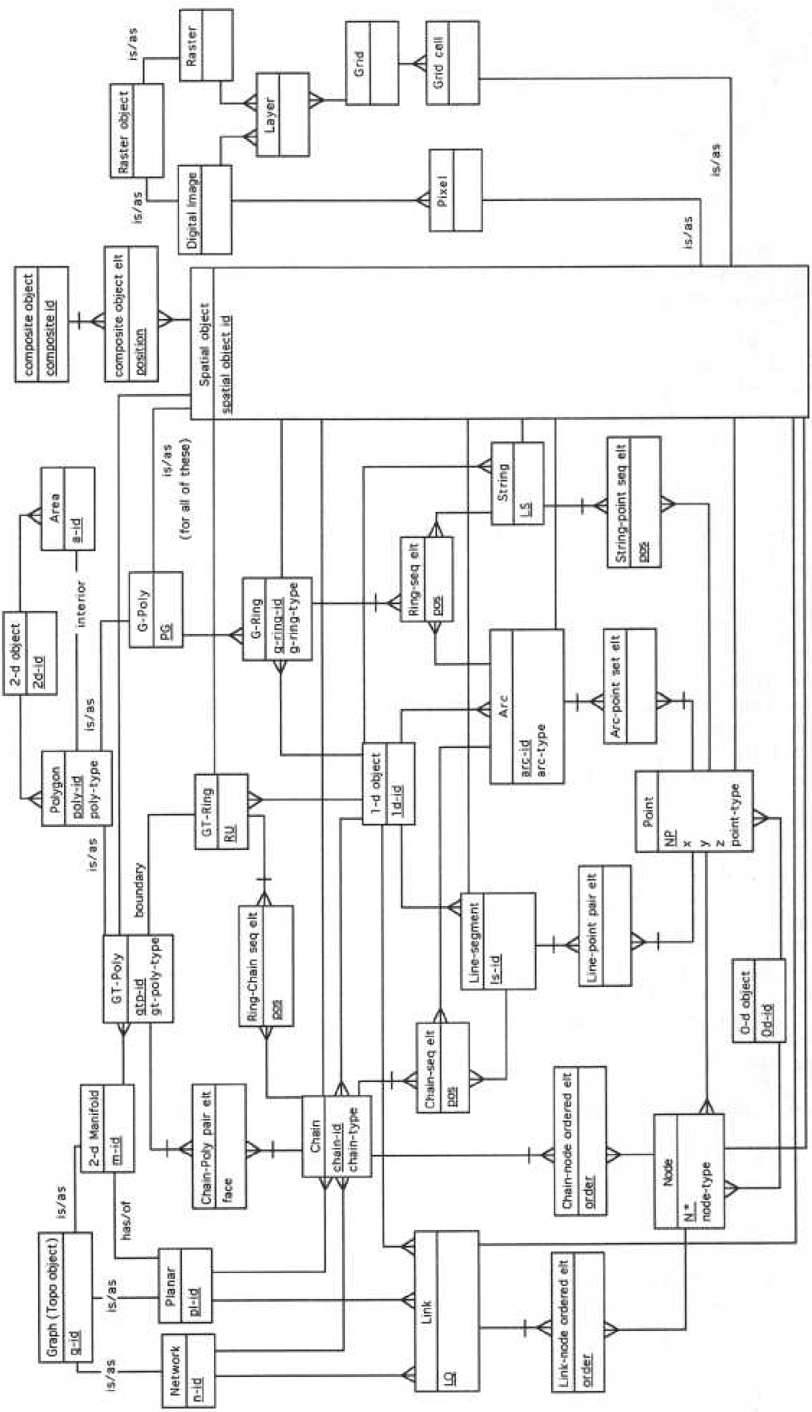
composite object
composite id
composite object elt
position
Raster object
Raster
Digital image
Layer
Grid
Grid cell
Pixel
is/as
is/as
is/as
is/as
Spatial object
spatial object id
2-d object
2d-id
Area
a-id
interior
Polygon
poly-id
poly-type
is/as
G-Poly
PG
is/as
(for all of these)
G-Ring
g-ring-id
g-ring-type
Ring-seq elt
pos
String
LS
String-point seq elt
pos
Arc
arc-id
arc-type
Arc-point set elt
1-d object
1d-id
GT-Ring
RJ
Point
NP
x
y
z
point-type
is/as
boundary
GT-Poly
gtp-id
gt-poly-type
Ring-Chain seq elt
pos
Line-segment
ls-id
Line-point pair elt
0-d object
0d-id
2-d Manifold
m-id
Chain-Poly pair elt
face
Chain
chain-id
chain-type
Chain-seq elt
pos
Chain-node ordered elt
order
Node
N*
node-type
is/as
has/of
Graph (Topo object)
g-id
Planar
pl-id
is/as
Network
n-id
is/as
Link
LQ
Link-node ordered elt
order

Figure 1

described by Egenhofer and Herring), the SDTS made an important statement. The standard treats vectors and rasters (and potentially other structures) as conceptually similar things in the sense that they are all spatial objects of one type or another. This emphasizes that there is a common ground in that they all deal with the structure of either space or things in it. Also it helps build recognition of the difference between structural spatial concepts and other concepts (such as attributes) that are not inherently spatial at all. That is to say that the structure of space, or things within a space, has different modeling requirements than the properties of the things that occupy space, or the abstract descriptions of features of the space.

Given the spatial models that it covers, the SDTS spatial object description is considerably detailed, and relatively straight-forward; and it stands in contrast to the description they offer of real world phenomena that spatial objects presumably represent. This is the topic of the following section.

Real World Phenomena: Confusion and Ambiguity

The SDTS is primarily concerned with spatial object data transfer, but recognizes that the context of use of spatial data involves real world features, or phenomena. The standard addresses this aspect of spatial data in two ways. First, it presents what it calls a "conceptual model" that serves as a "framework for defining spatial features and a context for the definition of a set of spatial objects." Second, it provides a set of common definitions for terms which are types of real world phenomena. In this section, we examine the "conceptual model" of the SDTS which offers a description of real world phenomena and spatial objects.

The SDTS does not provide a description of a model of the real world phenomena at the same level of detail and completeness as is provided for the spatial object description. This is not surprising since the development of spatial representations has been a dominant focus of research, whereas attempts to codify aspects of the represented phenomena, spatial and non-spatial, is more recent. This newer interest is due, in part, to requirements for spatial analysis support that involve richer models of the phenomena represented. The lack of detail and ambiguous description of concepts concerning the real world phenomena makes it difficult to clearly interpret the standard with respect to this topic. We examine two problem areas and, for each, justify an interpretation of the standard that is reasonable, although not the only one possible.

Explicitly Distinguish Real World from Model The first problem is that the SDTS does not explicitly state a distinction between real world phenomena and a model of what a system must remember about same. It is important to make this distinction because there is a difference in discussing phenomena themselves vs. the things about phenomena that must be remembered or recorded as data. In defining concepts of real world phenomena, where a "river" would be a class of phenomena, and the "Mississippi River" would be an example member of that class, the SDTS says, "... classes of phenomena are called entity types and the individual phenomena are called entity instances." Although this implies a discussion of modeling things to be recorded or remembered about phenomena, it could be interpreted in two different ways. One interpretation is that an "entity type" is an abstract class of phenomena and an "entity instance" is actually the phenomenon itself, which means the Mississippi River itself is an instance of the entity type River. A different interpretation matches the following re-phrasing of the SDTS statement: "In a data model, those named classes of phenomena that are to be remembered (and presumably recorded in a computer system) are called entity types and the named individual phenomena of a given type are called entity instances." This asserts that an "entity type" names a class of phenomena. Likewise, an "entity instance" names a particular member of the class. "River" is a named type, and "Mississippi River" is a named instance. The instance refers to the real world river, but clearly is not the river itself.

The latter interpretation is actually more useful for a discussion of data representation and standardization, and it is reasonable to assume that the standard can be interpreted this way. This is supported by the standard's use of concepts and terminology that derive from data modeling and knowledge representation literature. For example, the SDTS defines classification, generalization, aggregation, and association in terms of entities and entity instances. These are the same names of specific concepts that are often discussed in data modeling (in database and other computer science literature), and they are discussed in a very similar manner (Brodie & Mylopoulos 1986, Peckham & Maryanski 1988, Hull & King 1987, Sowa 1984).

However, there is a possibility that the former interpretation is the one that was intended. Deciding which interpretation is intended depends on how the issue of an "entity instance's" dependence on spatial objects is resolved. This topic is addressed next.

Entity Instance Dependency on Spatial Objects It is difficult to determine what level of dependency an "entity instance" has on a spatial object. The SDTS says that "entity instances have a digital representation (that) consists of one or more spatial objects." This raises the question of whether or not entity instances can be modeled and digitally represented independent of spatial objects. Despite this potential "entity instance" dependence, the independence of spatial objects is implied fairly strongly. The SDTS says that spatial objects do not have to represent an entity instance, but if they do they are called "entity objects."

The standard goes on to say that an "entity object (is) a digital representation of an entity instance," which, in the context of the discussion, may lead to the conclusion that the only kind of digital representation for an entity instance is in the form of a spatial object. It is easy to find that this constraint, if enforced, is overly limiting. For example, text may be used to represent entity instances, or their attributes, and GIS currently handle at least some such data as text. However, if one does assume that entity instances depend on spatial objects for existence, one might also assume that "entity instance" means the real world phenomenon itself, and an "entity object" is the representation of it using some spatial object as the means of description.

Whether or not it is the intention of the SDTS to assume the independent existence of entity instances (by which we mean the independent modeling of real world features), this independence is necessary for designing systems that can support semantic information relating to real world phenomena. This idea is elaborated in the next section.

The Entity Object Concept

In this section, we review the "entity object" concept introduced in the SDTS. To understand and model this concept precisely, it is first necessary to determine what the SDTS means by "entity instance." Once an interpretation is provided, a meaning of the "entity object" can be proposed.

We assume an interpretation wherein the SDTS "entity type" and "entity instance" are used to model (and name) real world phenomena, and that "entity instances" are not, strictly speaking, the phenomena themselves. We assume that models of real world phenomena may involve both spatial and non-spatial data, but they may exist as models independent of spatial models that can be related to them. Given these assumptions, the "entity object" described by the SDTS models the relationship between an instance of an entity (which models a type of real world phenomenon) and a spatial object used to represent it. Justification for the assumption of entity instance independence, and the subsequent interpretation of the entity object as a relationship follows.

Assuming the Independence of Entity Instances Viewing the entity object as a relationship necessitates viewing entity instances as having separate existence from spatial object representations, since, in general, a relationship between two things implies the separate existence of the things to be related. Below we argue for the existence of an entity instance independent of a spatial object.

An entity instance must be either a real world phenomenon or a modeled instance. As a real world phenomenon it clearly has a separate existence. As a particular modeled instance, it may or may not exist separate from a spatial object representation. However, it is clear that it is possible to model things about real world phenomena that are not spatially oriented, and it may indeed be useful to do so. For example, rivers are wet, they may be muddy, they have current and direction, they support eco-systems, etc. In addition, it is useful to recognize when two different spatial objects actually refer to the same real world phenomenon. System support for recognition of such an occurrence is simplified if real world phenomena are modeled independently of their spatial representations. Thus it is useful, in the general case, to assume the option of independent entity instances, even if in many cases (i.e. actual system implementations) there is no separate existence or model.

The advantage of modeling entity instances in a manner that supports their independence is that this allows the system to be able to deal with concepts that apply to real world phenomena that are in addition to, and independent of, a geometric view of them. The need for this kind of view is reflected by the increasing interest in GIS that handle additional data about real world phenomena besides the spatial data. Albeit that spatial objects are used to represent the real world phenomena, and often this is the only representation of them the system has access to, in order to be more informed about them, the system must have access to more complete or different models of them.

In sum, a spatial object representation should be one, but not the only, representation of real world phenomena available to the system. Richer representations of the meaning of spatial objects in the context of real world phenomena require greater flexibility in modeling these things. In addition, the system ought to be able to support multiple spatial object representations for the same entity instance.

Viewing the Entity Object as a Relationship The standard implies that an entity object involves some sort of relationship between a spatial object and an entity instance without specifying the details. There are indications to support that an entity object actually is a relationship, even if the standard does not state this expressly. (Remember that part of the trouble is that without establishing the clear independence of entity instances and their role in modeling, it is hard to discuss the entity object as a relationship.)

The SDTS does say that an entity object is a spatial object that represents an entity instance. Since an entity instance ought to have an existence that is independent of a spatial object representation, and since spatial objects apparently do have independent existence, this implies that an entity object exists when these two are paired together. That is, an entity object exists when a relationship exists between spatial object and entity instance.

The standard has taken a step toward clearly separating spatial objects from the phenomena they represent (i.e. the entity instances) in another way. They explicitly acknowledge the confusion resulting from overloading the term "feature", which can mean either the real world phenomena, or the geometric (spatial object) representations of same. Because of the different ways the term is used, the standard specifies that the term "feature" means a spatial object only when it is used to represent a real world phenomenon.

This distinction can be made more clearly and effectively, by defining an entity object as a relationship between two independently defined concepts. That is, the entity object is the relationship between a particular entity instance (which models some particular real world phenomenon) and a particular spatial object. Then, a feature may be defined as "the combination of the real world phenomena that is modeled as an entity instance, and the spatial object used for its geometric representation." With this definition, a "feature" is effectively an "entity object" which in turn is a relationship.

EXTENDING THE MODEL

In this section we address some limitations of commonly used models for GIS, by describing and extending a model based on the SDTS. The purpose of the extensions is to make more data available to the system, enabling it to make more decisions about data management that are based on a greater understanding of the meaning of the data stored, i.e. the semantics. Semantic information includes domain specific data concerning real world phenomena, as well as specialized information on the use of geometric data, its properties, conversions, and interpretation. Our extensions center around the concepts of the real world phenomena that spatial objects may represent, the spatial object itself, and the relationship between these things.

The Meta Model Concept

The SDTS presents a description of a context, which they call "a conceptual model," within which their description of spatial objects can be understood. We can view the SDTS "conceptual model" as a statement about a meta-model of GIS, wherein a model of the types of things you create models for in a GIS would be a meta-model.

Specifying that, in general, concepts about real world phenomena are of interest, without specifying exactly which classes are of concern, is a meta-modeling concept. In a meta-model, generality of the structures used to model phenomena is important because the various views of real world phenomena, and what users want to remember about them, can vary greatly from application to application. In contrast, a specific class of phenomena, such as roads, is an application level concept, which would involve a data modeling effort concerning use of roads data in a specific GIS application.

We use an LDS to describe a meta-model of GIS, based upon but extending the SDTS. The meta-model is shown in Figure 2. The model reflects general components of GIS but includes data that would allow a system to handle greater complexity and semantic interpretation. The motivation for decisions reflected in the model are the subject of the next three sections. We organize the discussion around three principle concepts: the modeling of real world phenomena themselves, the spatial object used to represent modeled instances, and the relationship between these two.

Real World Phenomena and Attributes

In this discussion of extensions to the GIS model, we focus on developing the model of real world phenomena and their attributes. The important points made are that clear, explicit separation of the real world phenomena model is necessary, and general capability for modeling real world complexity is important.

Spatial Object Model and Real World Phenomena Model Separation Clear separation of the model of spatial objects and the model of the types of real world phenomena that a GIS might support is illustrated in Figure 2. LDS entities that model real world phenomena are grouped so that connections to spatial objects occur at the entity object. Justification for this type of clean separation is that real world phenomena are complex and much of the complexity is irrespective of any spatial object representation. This separation is necessary for both handling multiple spatial object representations of the same phenomena, and for designing systems that can manage complexity and multiple views of the data.

Complexity of real world phenomena The SDTS asserts the existence and complexity of the real world phenomena and suggests that we can model them with various constructs. The LDS fragment in Figure 2a (a part of Figure 2 enclosed with a dotted line box) expresses our interpretation of what is required to adequately model real world phenomena, and to do so in a way that does not assume spatial object dependency. As a meta-model, this fragment is similar to meta-models previously developed, and found useful in general database design (Carlis & March 1984, and Held & Carlis 1989).The LDS entity called "Entity: Phenomenological category" is what the SDTS calls an "Entity." We added the phrase "Phenomenological category" to emphasize that instances of this are used for the modeling of real world phenomena. "Entity: Phenomenological category" has descriptors which are either of the type "attribute" or "relationship." Relationship descriptors allow many kinds of relationships (including class/subclass, etc.) to be created between phenomenological categories that are modeled in an application LDS. An instance of an "Entity: Phenomenological category" has many "phenomenon instances" which represent real world features or phenomena, such as the "Mississippi River." These "phenomenon instances" have associated with them many "attribute value" of "attribute" 's.

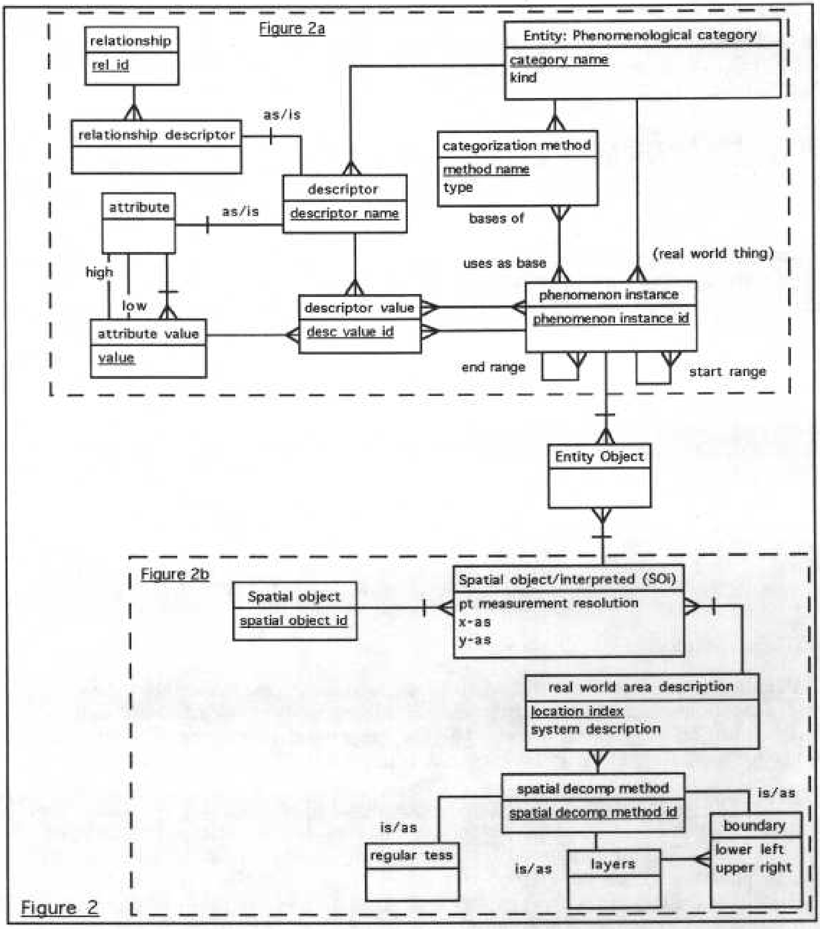

Figure 2

Kinds of real world phenomena Part of the complexity concerning real world phenomena is that things concerning or describing the real world are of different kinds, including at least physical, political, and abstract. The meta-model we present offers a unified concept for dealing with the fact that in general things occupy space, things have attributes, and attributes are used to describe different areas. An important aspect of this model is that it allows the system to manipulate an abstract phenomenological instance in a manner similar to a physical phenomenological instance. An abstract phenomenological instance of something like "an area of land use that is agricultural," can be treated in a way similar to a phenomenological instance of a river. Why are these similar? Because both occupy space in some sense, but both are separate from a specific location. Some instances may in fact shift over time, in terms of the location they occupy. In addition, both a river and a homogeneous, contiguous area (such as an area that has an agricultural land use), have in common the tendency to be described by geometry.

In order to provide a general model of the real world phenomena modeled in GIS, we need to allow an "phenomenological instance" to include abstract descriptions of an area, such as land use. Although a land use value, such as "agricultural" is actually an attribute associated with an area, we can model this as a phenomenological instance of an abstract kind of phenomenological category. The utility of this view is that an area with special properties attributed to it can be manipulated and given a geometric form in a manner conceptually similar to a physical concept. And they both are viewed by the system as occupying space. We can still maintain clear separation of location and things, even abstract things, that are in that location.

A separate issue is how classes get created, and have instances assigned to them which ultimately relate to spatial objects.

<u>Examples of Types and Instances</u> Since meta-models are abstract and difficult to understand, examples of how the meta-model relates to some specific data models of real world phenomena are presented in Figure 3. First, note that there is a difference in the meta LDS and the application level LDS. In a specific application, such as a database or GIS built for some purpose, an LDS contains LDS entities that name the types of data you want to remember and store instances of. In the meta-model (in our case, of GIS) the LDS contains LDS entities that name arch-types in a sense, or "types of types" of data. Thus, instances of "Entity: Phenomenological category" in the meta-model are names of LDS entities in a GIS application LDS.

Figure 3 shows some simple example data. The "mapping" between a meta level LDS and application level LDS's is illustrated with instance data. We show that an instance of "Entity: Phenomenological category" at the meta level becomes the name of a type (i.e. an LDS entity) in a application LDS. The instance of "Phenomenon instance" at the meta level is an instance of an application level LDS entity. The examples include classes that are physical, political, and abstract.

<u>The Spatial Object</u>

This section describes extensions that support an explicit understanding of the spatial object in the context of real world location. It introduces the notion of an interpreted spatial object, where the word "interpretation" is used to signify how a spatial object maps to a real world location, and how the basic building blocks of spatial objects (points or cells) actually map to an area. Note that in the meta-model in Figure 2, we show the spatial object as an LDS entity. However, for simplicity, we do not include the additional detail (i.e. all LDS entities that relate to spatial object) shown in Figure 1.

<u>Separate concept of location from geometry of something in the location</u> One purpose of a spatial object in a GIS is to use it as an indication or specification of a real world area or location. As a spatial object is a geometric description based on organization of points, or a raster like collection of cells, it can be mapped to many real world area descriptions. A "real world area description" is shown as an LDS entity in the LDS fragment identified as Figure 2b (a part of Figure 2 enclosed with a dotted line box). A real world area description may have many spatial objects that map to it. This creates a many-to-many relationship between a spatial object and a real world area description. When a spatial object is mapped to a real world area description, so that the x and y values of the points in the spatial object map to a particular area in the real world, we call it an "interpreted spatial object."

<u>Introduce Measurement Resolution</u> Because a geometric point is a pure abstraction, with no area, we introduce the concept of measurement resolution associated with a point to provide more explicit interpretation of how much real world area the point is used to represent. When a spatial object based either on geometric notation or raster cell models, is mapped to a location, there is an explicit or implicit assumption concerning the measurement resolution of a point or cell involved. This is because the spatial object is not used as a purely geometric object, but rather to describe some object set at some location.

We first discuss how measurement resolution relates to a raster cell. A raster cell is used to stand for an area rather than something in an area, but it is typically associated with data collected and tied to an area. That real world area has some dimension in the sense that there is a minimum or average distance between points at which the data were collected. This can be calculated as a length and stored as "pt measurement resolution." For raster cells, this value gives a measure of the real world area associated with a single cell.

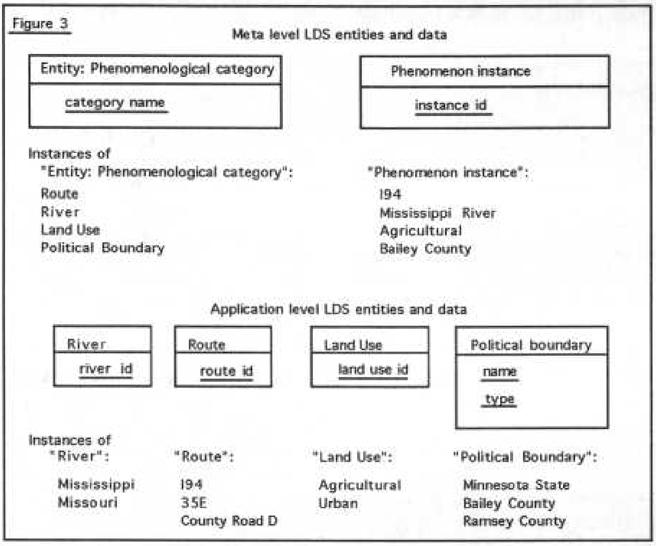

Figure 3

Less intuitive is the notion of measurement resolution when applied to vector or point based geometric data. However even in this case, a line, for example, is used to represent a real world feature such as a road. There is an assumption, although implicit, of width of the road. Thus the point ought to be interpreted in a manner that conveys the notion of width. "Pt measurement resolution" can be used with geometric points to indicate a real world area that the geometry supposedly relates to.

The result is an interpreted spatial object which has a geometric or simple cell description, as well as a means for specific mapping to real world location. It is the interpreted spatial object that is really the basis for describing and representing real world phenomena, and thus the meta LDS in Figure 2 models this. However, the interpretive aspects are often managed by users rather than a GIS. With a meta-model expressing this explicitly, we can explore more options for developing system support of this type of information. This clearer specification of what is really meant by a spatial object can enable a system to know about, and subsequently be able to use, information in more sophisticated data processing and information management.

Modeling The Entity Object Relationship

In this section we focus on the SDTS's entity object concept. The SDTS's description of the "entity object" expresses some sort of relationship between a "spatial object" and an "entity instance." We explore the nature of this relationship, with sufficient attention to detail to allow modeling it with the LDS notation. Different ways of viewing the relationship, which may be investigated as part of a modeling process, result in different ways of representing the entity object concept in an LDS. The important messages here are that the entity object, being essentially a relationship between two complex things, is actually quite complex itself. Other data, which must be modeled, relate directly to this union

Entity Object Complexity The entity object concept introduced in the SDTS should be examined in terms of the different ways in which relationships between spatial objects and phenomenon instances might exist. Figure 4 shows an example where a single phenomenon instance ("Mississippi River") has many possible spatial objects ("line-1" and "Poly-10"). The LDS models this possibility. Another situation, modeled in Figure 5, shows that a single spatial object ("Poly-12") may be associated with many phenomenon instances ("Bailey County" and "Agricultural land use"). The general case, in Figure 6, is that phenomenon instances may have many spatial objects, and a spatial object may represent many different phenomenological instances.

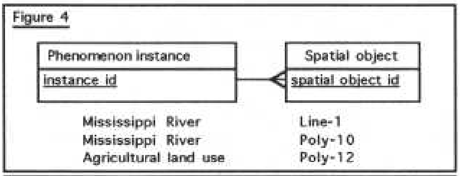

Figure 4

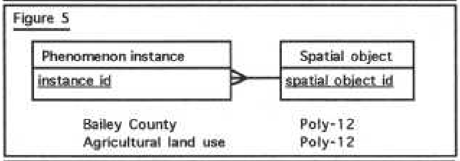

Figure 5

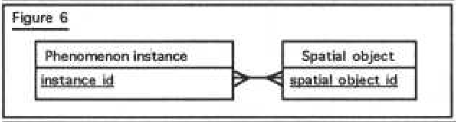

Figure 6

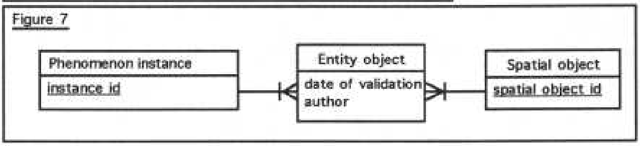

Figure 7

In addition to being a many-to-many relationship, the relationship itself may have descriptors, which, when using LDS notation, means the relationship is promoted to an LDS entity itself. Once this happens, specification of types of data that relate specifically to the relationship is possible.

In a meta-model, we are interested in modeling the most general case, thus Figure 7 models the entity object as an LDS entity allowing it to have its own descriptors. We presented this logic up to here using the "spatial object " instead of the "interpreted spatial object " for simplicity. As is the case with many real systems, this simpler model of a spatial object implies that locational interpretation, if desired, is provided by the user. The extension of Figure 7 that shows how the entity object relates to the interpreted spatial object is shown in Figure 2.

Entity Object as a Concept Occupying a Location. The entity object concept concerns the modeling of an object that occupies space at some real world location. It is not the real world phenomenon itself, nor the attributes accorded to a location, nor the location itself. But location of the object (including the abstract objects such as a land use coverage) is determinable along with the phenomenon.

UTILITY OF THE EXTENDED MODEL

As real world phenomenological information becomes a more important aspect of GIS, the standard may be expanded to include transfer of such data. Better models of these data (which ultimately would be the basis for a system's understanding of what real world features are, and how they are related to spatial objects) are needed for extending the standard in this direction. A step toward this is conceptual analysis that results in a clearer data model of the types of information involved.

The extended model highlights the role of the entity object. Seeing the real world phenomena and spatial object distinction clearly, facilitates analysis at the more conceptual level, removing the need to focus on data structure. The model is based on a separation of *what* the data are from *how* they are stored or represented in the system. This type of modeling activity helps us to identify new types of data to use in the processes that enable system decision making. Fuller, richer distinctions, modeled and made available to the system, are part of the process of semantic information analysis. If the system is to handle more of the data management responsibility, more complete modeling of the actual meaning of data must be made explicit, and not stored only in people's minds. The system cannot use what it has not been told about.

Clear separation of what types of data actually are used and how they ought to be interpreted will be increasingly important as more data are collected and must be managed by systems.

CONCLUSION

The SDTS contains ambiguous statements concerning their conceptual model of real world "features" and their relationship to spatial objects. Disambiguation is necessary for development of a clear model of these data, and is important if systems are to be designed that will effectively manage such data. Assuming one reasonable interpretation of the standard, an extended model of important types of GIS data is proposed, and presented in LDS notation. The model makes data explicit that typically are implicit, and potentially available to humans, but not to systems. The model also illustrates the relationship between real world entities and spatial objects in a manner that differentiates them clearly. This facilitates better modeling of real world phenomena and supports development of future GIS systems that will have a greater ability to manage the "meaning" of the geometric (spatial object) data. This will allow development of specialized processing in spatial analysis that is closely tied to, or dependent upon other information besides spatial object information. It will also allow development of systems than can handle more of the data integration burden that GIS commonly cannot handle, due to their insufficient "knowledge" of the meaning of spatial objects that are stored.

REFERENCES

Brodie, M.L., and Mylopoulos, J., 1986, *On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies*, Springer-Verlag, New York.

Carlis, J.V., 1993, *Logical Data Structures*, Book manuscript in press.

Carlis, J.V. and March, S.T., 1984, "A Descriptive Model of Physical Database Design Problems and Solutions." In *Proceedings of the IEEE COMPDEC Conference*.

Egenhofer, M.J., and Herring, J., 1991, "High Level Spatial Data Structures for GIS," from D.J. Maguire, M.F. Goodchild, and D.W. Rhind (eds.), *Geographic Information Systems*, Vol. 1, pp. 457-475, Longman Group UK, Harlow.

Held, J. and Carlis, J.V., 1989, ADM-An Applicative Data Model, *Information Sciences*, January, 1989.

Hull, R., and King, R., 1987, "Semantic Database Modeling: Survey, Applications, and Research Issues," *ACM Computing Surveys*, Vol. 19, No. 3.

Peckham, J., and Maryanski, F., 1988, "Semantic Data Models," *ACM Computing Surveys*, Vol. 20, No. 3.

Peuquet, D.J., 1990, "A Conceptual Framework and Comparison of Spatial Data Models," in *Introductory Readings in Geographic Information Systems*, Donna Peuquet and Duane Marble (eds.), Taylor & Francis, New York.

Sowa, J.F., 1984, *Conceptual Structures*, Addison-Wesley, Reading, Massachusetts.