Supporting User-defined Granularities and Indeterminacy in a Spatiotemporal Conceptual Model

Vijay Khatri[1], Sudha Ram[1], Richard T. Snodgrass[2] and Grady M. O'Brien[3]
[1] *Department of MIS, University of Arizona, Tucson, AZ, USA. Email:* {vkhatri, ram}@bpa.arizona.edu
[2] *Department of Computer Science, University of Arizona, Tucson, AZ, USA. Email:* rts@cs.arizona.edu
[3] *U.S. Geological Survey, 520 North Park Avenue, Suite 221*, Tucson, *AZ, USA. Email:* gmobrien@usgs.gov

Granularities are integral to spatial and temporal data. A large number of applications require storage of facts along with their temporal and spatial context, which needs to be expressed in terms of appropriate granularities. For many real-world applications, a single granularity in the database is insufficient. In order to support any type of spatial or temporal reasoning, the semantics related to granularities needs to be embedded in the database. Specifying granularities related to facts is an important part of conceptual database design because under-specifying the granularity can restrict an application, affect the relative ordering of events and impact the topological relationships. Closely related to granularities is indeterminacy, i.e., an occurrence time or location associated with a fact that is not known exactly. In this paper, we present an ontology for spatial granularities that is a natural analog of temporal granularities. We propose an upward-compatible, annotation-based spatiotemporal conceptual model that can comprehensively capture the semantics related to spatial and temporal granularities, and indeterminacy without requiring new spatiotemporal constructs. We specify the formal semantics of this spatiotemporal conceptual model via translation to a conventional conceptual model. To underscore the practical focus of our approach, we describe an on-going case study. We apply our approach to a hydrogeologic application at the United States Geologic Survey and demonstrate that our proposed granularity-based spatiotemporal conceptual model is straightforward to use and is comprehensive.

1. Introduction

Eighty percent of all human decisions contain a spatial component [1] and time is a component of almost all database applications [27, 64]. Geographic information is increasingly employed in a wide array of applications including social, environmental and economic studies. Granularities are intrinsic to spatial and temporal data. Many prior studies [15, 19, 20, 52, 62, 63] cite the need to support multiple spatial and temporal granularities in a database. For example, in a cadastral application [20], mortgages can be associated with a temporal granularity of day and the representation of long-term land-use changes may require a temporal granularity of year. Day and year, or more accurately Gregorian day and Gregorian year, are examples of *standard granularities*. On the other hand, *user-defined granularities* [6] like business week and irrigation year may have different definitions in different contexts; e.g., business week can imply 5, 5.5, 6 or even 7 days depending on an organization policy, industry norms, and even culture and traditions of a country or a region. Similarly, irrigation year for, say, North Dakota may be defined as a period from May 1 to September 1. Developing methodologies and tools to simultaneously support multiple granularities is a challenging and an active area of research [4, 5, 7, 11, 15, 16, 60, 61]. However, there does not exist a comprehensive mechanism to capture the users' granularity related requirements during conceptual database design. Additionally, indeterminacy, or "don't know *exactly* when or where", is related to granularity [15] and may be pertinent for a database application. In this paper, we integrate concepts related to granularities and indeterminacy in a conceptual model, thereby realizing the Spatiotemporal-Unifying Semantic Model (ST-USM).

Conceptual database design is widely recognized as an important step in the development of database applications [2, 17, 47]. During conceptual database design, a *conceptual model* provides the notation and formalism that can be used to construct a high level description of the real world referred to as a *conceptual schema*; in this paper, a conceptual schema is interchangeably referred to as a *schema*. Granularities provide a mechanism to hide details that are not known or not pertinent for an application [5]. In order to support multiple granularities in a schema, there is a need for a mechanism whereby the users can specify standard and user-defined spatial and temporal granularities during conceptual design of the database. Closely related to granularities is indeterminacy that recognizes our inability to capture precise information about the real world, and is important for many applications. Wang et al. [60] describe *logical design* for temporal databases with multiple temporal granularities. Conceptual database design is a precursor to logical design and takes the users' requirements as an input and transforms them into a conceptual schema. This implementation independent conceptual schema is the basis for communication between data analysts and users during database design; it is also used to identify potential inconsistencies in the users' requirements [17]. Since the conceptual schema is independent of the implementation environment, it is also useful in the event of technology upgrades and transfer.

Our work makes several contributions related to capturing granularities-related user requirements at the conceptual database design stage. According to Puppo and Dettori [39], existing Geographic Information Systems (GISs) do not provide much support in multi-resolution data handling. We have extended the concept of spatial *resolution* [62, 63] to define spatial granularity parallel to temporal granularity. These semantics related to multi-granularity representation can facilitate spatial reasoning [39] and generalization [62], i.e., transformation between finer and coarser levels of detail. We have embedded concepts related to spatial and temporal granularities into a granularity-based spatiotemporal conceptual model, ST-USM. ST-USM is annotation-based, does not introduce any special spatiotemporal constructs, and is upward compatible with a conventional conceptual model, the Unifying Semantic Model (USM) [40]. Using ST-USM, a data analyst can capture granularity-related requirements in the schema. Although we have used USM as the base model for ST-USM, our annotation-based approach is not specific to USM and can be applied to any conventional conceptual model, e.g., [17, 22, 37]. We have also defined a formalism to incorporate the semantics related to indeterminacy. We provide the formal semantics related to granularities and indeterminacy in the annotated ST-USM schema via translation to a conventional conceptual schema. These granularity- and indeterminacy-related semantics can help provide support to constraint reasoning and integration of distributed databases [7]. A contribution of this paper is to demonstrate that our approach is practical and useful. We describe an ongoing case study involving the design of a spatiotemporal database for a hydrogeologic application at the United States Geological Survey (USGS). The case study demonstrates that our approach to spatiotemporal modeling is straightforward to use and is comprehensive.

In summary, we have adapted an existing temporal granularity model [4, 5, 15, 16] and extended a spatial granularity model [62, 63] to propose a granularity-based spatiotemporal conceptual model. Our proposed approach is straightforward to implement, based on ontological concepts, provides a mechanism to capture the semantics related to granularities and indeterminacy during conceptual design, and dovetails with the existing conventional database design methodologies.

We outline the assumptions in this paper and describe the scope of this work. (i) Objects of interest in the real world are referred to as *entities*; we assume that geo-referenced entities are embedded in Euclidean space. Entities are grouped into an *entity class* based on some common characteristics and a set of entities in an entity class is referred to as an *entity set*. (ii) A database schema can evolve with time, and schema versioning [44, 45] is an important area of research; however, schema versioning is not the focus of this paper. (iii) While indeterminacy may be associated with many aspects in the representation of the real world, this paper focuses on indeterminacy related to time and space, i.e., *valid time indeterminacy* [16] and *spatial indeterminacy*. Informally, these types of indeterminacy may be characterized as "don't know exactly *when* or *where*" information. While we know that the phenomena occurred within the specified temporal/spatial bounds, we do not know the exact time or location of occurrence. (iv) In this paper, all sets are assumed to be finite. (v) Based on perception, space may be differentiated as *large-scale* and *small-scale* space [29]. While the former is defined as one that cannot be viewed from a single viewpoint, the latter is visible from a single vantage point. As with Mark and Frank [32], we construe geographic space to be equivalent to large-scale space; in this paper, the term *space* is used interchangeably to mean large-scale space or geographic space. We have defined spatial granularity for geographic database applications. In these applications, horizontal space is differentiated from vertical space; correspondingly, we define horizontal and vertical spatial granularities. In summary, the focus of this paper is on integrating spatial and temporal granularity- and indeterminacy-related semantics into a conceptual model thereby facilitating a semantically richer representation that is useful for design of temporal and geospatial database applications.

The rest of the paper is organized as follows. We motivate the need for capturing the semantics related to spatial and temporal granularities and indeterminacy in Section 2 using a case study at USGS. In Section 3, we provide an ontology related to granularity and indeterminacy. We describe how we have adapted the ontological concepts into ST-USM in Section4. In Section 5, we apply the proposed spatiotemporal model to develop a schema for the case described in Section 2. Finally we summarize our work and indicate future research directions. Throughout, we provide examples from an ongoing hydrogeologic application at USGS.

2. Motivation

Conceptual modeling [2, 8, 22, 37, 40] takes users' requirements as an input and transforms them into a high-level conceptual schema. This schema represents an aspect of the real world, often referred to as the *miniworld* [17]. The schema represents the structure of the data manipulated by the application and serves as the system metadata. A good conceptual schema connects users, database analysts, and the database implementation. We describe granularity- and indeterminacy-related requirements of the users for a ground-water application. We, thus, motivate the need for a spatiotemporal model that supports multiple standard and user-defined granularities, and indeterminacy using a case study; we highlight those aspects of this application that are pertinent to granularity and indeterminacy.

Sinton [48] defines geographic information as having (i) a *theme*—the phenomenon or object being observed, (ii) the *location* of the phenomenon, and (iii) the *time* related to the phenomenon. Frequently, spatial and temporal data in applications are associated with multiple granularities. From a database perspective, space and time are discrete and there is usually an arbitrary smallest unit that is managed by the database. However, all spatial and temporal information is generally not stored in terms of this smallest unit but in terms of other larger standard or user-defined units. For example, profit may be captured for quarter, rainfall in terms of day, permanent employee's salary in fortnight, a contract employee's salary in hour, and the *x*- and *y*-coordinates of a borehole's location in dms-degree (degree minute second). For many applications, we may also need to capture inherent indeterminacy associated with spatiotemporal data. For example, the *x*- and *y*-coordinates of a borehole location (in dms-degree) may have an associated indeterminacy, which is also expressed in dms-degree. The existing conceptual models do not provide a formalism to specify granularities and indeterminacy; as a result, the users' spatiotemporal data requirements are at best only partially captured. Many prior studies, e.g., [23, 42], attribute project failures to lack of identifying user requirements during conceptual design. Therefore, there is a need for a formal approach to capture the semantics related to granularities and indeterminacy at conceptual database design stage.

We are working with a group of researchers who are developing a ground-water flow model (GWFM) [13] for the Death Valley region. The Death Valley region includes approximately 80,000 km^2 in Nevada and California. Beneath the earth's surface there is a zone where all interstices are saturated; this is called ground water. Ground water is stored in voids, spaces and cracks between particles of soil, sand, gravel, rock or other earth materials. Saturated rocks that will yield adequate quantity of ground water to a well or spring are called aquifers. In arid regions like Death Valley, ground water provides a large percentage of water for domestic, industrial and agricultural uses.

The ground-water flow model objective is to characterize regional 3D ground-water flow paths so that policy makers can make decisions related to radio-nuclide contaminant transport, and ascertain the impact of ground water pumping on national parks and local communities in the region. To perform calculations related to the flow of ground water, the area being simulated is discretized using GWFM grid defined by *rows* and *columns* as illustrated in Figure 1. The flow model is subdivided into GWFM layer so that vertical ground-water flow variations can be simulated. The region defined by a horizontal grid (i.e., row and column) and vertical layers is referred to as a cell. Calculations are performed for each *active* cell and the results from each cell are used as an input for the surrounding cells; cells not included in the calculations are referred to as *inactive*. A model solution, which includes simulated water levels and ground-water discharge, is obtained by providing starting conditions and iterating until closure criteria have been met. However, the quality of the model output and the associated predictions are dependent on the input data.

There are several problems related to the design of the existing input database. Some of the tables have a large number of attributes, e.g., one of the tables has 57 columns. As a result, the DBMS does a lot of I/O to process even trivial queries. Many tables are not normalized and have excessive NULLs; as a result, 8% of the total disk space is unused. Additionally, many schema constraints are inconsistent and ambiguous. Because of the design problems associated with the input database, we are redesigning the input and the output database for the ground-water flow model.

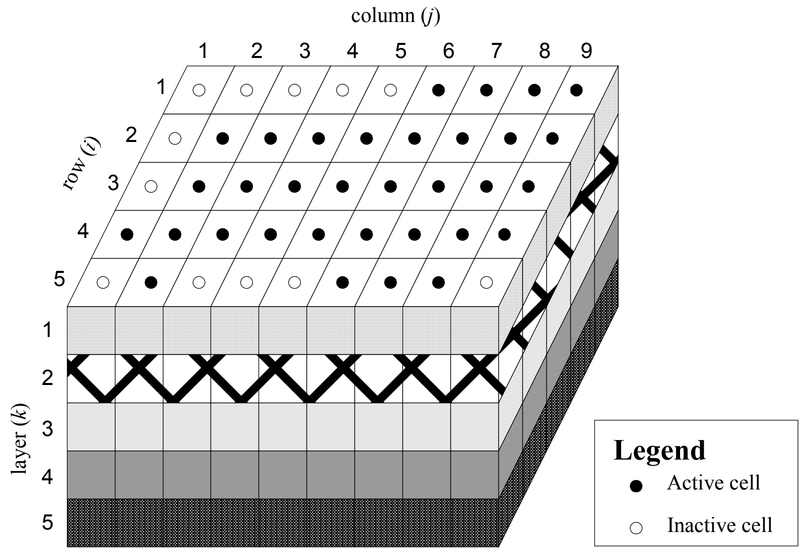


Figure 1: An example of an aquifer system represented using GWFM grid and layer

A large part of the input and output data for the ground-water flow model is spatial in nature. For example, two key objects of interest in the application are spring-water sites and borehole sites. Both of these are spatial in nature. A spring-water site represented as a point on the surface of the Earth whose location is given by *x*- and *y*-coordinates, in dms-degree. The borehole site, on the other hand, refers to a part of the well whose location is given by *x*- and *y*-coordinates on the Earth's surface in dms-degree, and depth below land surface, in foot. There can be different borehole sites at different depths at the same surface location. Additionally, borehole sites and spring-water sites have an inherent indeterminacy associated with their surface location. This indeterminacy needs be captured so that the associated model precision can be quantified. A borehole site may have a pump that removes water from the borehole site; this can affect other data collected at the borehole site. However, the location of wells used for ground water pumping are often defined with a township-range (TR) system. The TR coordinates provide a uniformly sized square-shaped grid that encompasses the well location. At the time when pumping data collection was initiated, quarter-quarter section provided an acceptable level of granularity and the convention has continued to this day. The output of the model is also spatial and is represented with GWFM grid on horizontal surface and GWFM layer on the vertical space. The grid definition and orientation are based on the flow system characteristics and anisotropy being simulated. Additionally, GWFM grid does not have to be coincident with or have the same orientation as any other pre-defined grids such as latitude/longitude used for surface location.

The key input data for the model includes discharge (in cubic feet per second) at the spring-water site and water depth (in feet below land surface) at the borehole site, which are collected by source agencies. Discharge and water depth need to be associated with the time of measurement. Additionally, the time associated with discharge and water depth data needs to be captured to the granularity of second and minute, respectively. There are various hydraulic tests conducted at borehole sites and the results of these tests need to be coupled with time (in minute). Ground water pumping from wells is often provided in acre-feet per year for a specific irrigation season. An irrigation season is defined based on the climate in the region. Cold, northern climates may have an irrigation season from April to September, whereas warm, southern climates may have a year round irrigation season. Another temporal granularity is the stream-flow water year, which is defined as the period from October 1 to September 30 and is a typical reporting period for stream-flow and water-quality data.

Thus, this hydrogeologic application needs a mechanism to capture users' spatial and temporal requirements associated with spring-water sites, borehole sites, discharge, water level, source agencies, and pump lifts. The ground-water flow model uses temporal and spatial data expressed in various standard granularities, e.g., minute, second, foot, dms-degree and quarter-quarter section; it also includes user-defined granularities, e.g., GWFM grid. However, the extant conceptual models provide limited modeling support to capture spatiotemporal data requirements. Additionally, there is no mechanism to capture granularity- and indeterminacy-related requirements. If the granularity-related requirements are not captured, the database analyst must make some assumptions during subsequent logical design. If the data analyst assumes a finer granularity than is required (i.e., over-specifying the granularity) by the user, it may not capture the reality. On the other hand, if the data analyst assumes a coarser granularity (i.e., under-specifying the granularity) than is required by the user, the relative ordering of events and topological relationships may not be correctly captured.

Thus, a proposed spatiotemporal conceptual model should: (i) provide a framework for expression of the structure of data which is easily understood and communicated to the users; (ii) contain a minimal number of constructs that comprehensively capture the semantics associated with spatiality and temporality; (iii) be easily translated into implementation dependent logical models; and (iv) be upward-compatible with the existing non-spatial and non-temporal conceptual models so that it does not invalidate the existing conventional conceptual schemas. Additionally, the granularity- and indeterminacy-related requirements include the ability to (i) define a methodology for multiple granularity representation in a conceptual schema, thereby (ii) allowing the users to choose the level of detail associated with facts, (iii) permitting transition from one level of detail to another, and (iv) supporting indeterminacy related to spatial and temporal data.

3. Ontology

Shared understanding of a domain of interest is referred to as an *ontology* [58]. We describe key semantic aspects related to spatiotemporal modeling based on ontological concepts related to granularities and indeterminacy, a companion to granularities.

3.1 Associating Facts with Time and Space

There are two kinds of facts associated with time: *events* and *states* [25]. On the other hand, geographic facts may be looked at in two different ways: view spatial data as attributes of position (i.e., *position-based*) or treat geometry as an attribute (i.e., *feature-based*). These two approaches correspond to the two geographic data models, *raster* and *vector*. Position-based view recognizes continuity and that any position has an associated value. On the other hand, feature-based view represents phenomena with respect to geographic entities having positional information. Logically one can define morphism between these two models. Linguistically, the feature-based model parallels our object-centric thought process [21]. We describe events and states related to temporal facts, and features related to geographic facts.

An event occurs at a point of time, i.e., an event has no duration. A state, on the other hand, has duration, e.g., a storm occurred from 5:07 PM to 5:46 PM. Facts can interact with time in two orthogonal ways [50] resulting in *transaction time* and *valid time*. Valid time denotes when the fact is true in the real world. Transaction time links an object to the time it is current in the database. *Existence time*, which applies to objects, is the valid time when an object exists [19]; it is also referred to as the *lifespan* [24] of an entity. While temporal granularity can be specified for existence time and valid time that for transaction time is system-defined.

The geometries of a geospatial object include a point, a line and a region. A point is a zero-dimensional spatial object with co-ordinates; a line is a sequence of ordered points, where the beginning of the line may have a special start node and the end a special end node; and, a region or polygon consists of one outer and zero or more inner rings [59]. David et al. [14] differentiate between a line and a region—the line itself is “the carrier” of the information while in a region, the area is of primary importance and the “boundary is secondary…to limit the area”.

Having briefly described the interaction of facts with time and space, we next describe granularities and indeterminacy associated with temporal and spatial data.

3.2 Temporal Granularity

Temporal granularity is a measure of the time datum. In this sub-section, the definitions of a time domain, temporal granularity and granularity relationships are based on those of the time glossary [4, 5].

A *time domain* is denoted by the pair $(T, \leq)$, where T is a nonempty set of *time instants* and “$\leq$” is a total order on T. We can assume the time domain to be *discrete* or *dense*. For example, $(\mathbf{Z}, \leq)$ represents a discrete time domain, which is a set of linearly ordered time points. A *temporal granularity* is defined as a mapping TG from *index* i to subsets of the time domain such that: (i) granules $TG(i)$ in a temporal granularity do not overlap; (ii) the index order of a temporal granularity corresponds to the time domain order; (iii) the index set of a temporal granularity provides a contiguous granule encoding; and (iv) a special granule called the *origin*, $TG(0)$

is non-empty. Although the index of a temporal granularity is constrained to be contiguous, the granules are not constrained to be contiguous on the time domain. Thus, a temporal granularity defines countable set of non-decomposable granules that can be composed of a set of contiguous instants or non-contiguous instants. Some examples of temporal granularities are Gregorian day, business day, and business week. While Gregorian day is a temporal granularity with contiguous granules of hour, business day is not. Each non-empty granule may have a textual representation termed a *label* (e.g., "November 25, 2000"), which can be mapped to the index integer with a mapping called the *label mapping*. The earliest time domain element in the origin is referred to as an *anchor* with respect to the time domain. The union of time granules is called an *image* of a temporal granularity. The smallest interval of the time domain that contains the image of the temporal granularity is called the *extent* of that granularity. The image of a temporal granularity can be contiguous or have holes in it. Gregorian day and business day are granularities with discrete image of days. However, Gregorian day has contiguous granules of hour while business day includes non-contiguous granules of hour.

Earlier we have advocated specifying granularities via mappings between pairs of granularities [15]. This is a more pragmatic way of defining granularities than the formal model of granularity as a mapping to subsets of time domain that was described above. The user need only specify conversion functions between various pairs of granularities to create a *granularity graph*. The granularity graph must contain a finest granularity, referred to as the *bottom* granularity. A set of functions must exist that defines a path from any granularity *TG* to the bottom granularity via successively finer granularities. Additionally, a set of functions must also exist that defines a reverse path from the bottom granularity to *TG* via successively coarser granularities. To specify the anchor of a temporal granularity in a granularity graph, it is sufficient to specify its origin with respect to any strictly finer granularity. For example, to anchor business week on business day, we only need to specify the first index of business day corresponding to the origin index of the business week. Granularities in a granularity graph form a *calendar*. A calendar may be considered to be a specification file that enumerates the names of the granularities and describes the mappings between them. It is unrealistic to assume that the granularity graph will always be predefined. While some of the granularities may be pre-specified, others may need to be added by the user resulting in a larger granularity graph involving multiple calendars. This can be done by defining mappings between any two granularities in different calendars. The finest bottom then becomes the bottom for a multicalendar system.

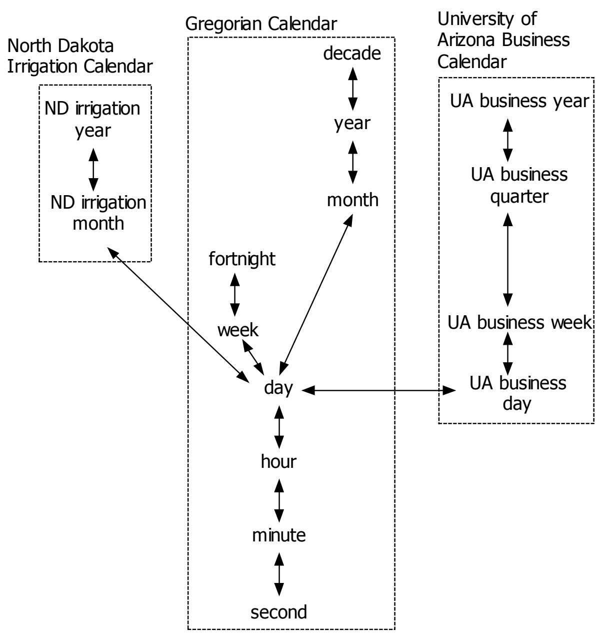


Figure 2: A multicalendar granularity graph

Figure 2 above shows a multicalendar graph with three calendars: (i) Gregorian calendar with a bottom granularity of second (ii) University of Arizona (UA) Business Calendar with a bottom granularity of UA business day (iii) North Dakota (ND) Irrigation Calendar with bottom granularity of ND irrigation month. ND irrigation month and UA business week are examples of user-defined granularities. In Figure 2, each node represents a granularity and each double-edged arrow between any two granularities represents two conversion functions between them. For example, a conversion function on Gregorian calendar from day (t) to week is $\lfloor t/7 \rfloor$, and week (t) to day is $7 \cdot t$ assuming that day and week have the same anchor; these mappings are straightforward and referred to as *regular mappings*. Other mappings like that between month and day do not involve simple multiply or divide and are referred to as *irregular mappings*. The bottom granularity of second in the Gregorian calendar is the bottom granularity of this multicalendar graph. The mapping between three calendars is achieved by defining the function between day and UA business day, and day and ND irrigation month. Additionally, mappings between temporal granularities TG_i require the granularities to share extent, i.e., any given granule is considered valid only if it lies within $\cap_i$ *extent*(TG_i); otherwise it is considered *invalid*. Given these mappings, it is possible to deduce a mapping from a granule in one granularity to any other granularity in the granularity graph [15].

We describe relationships between granularities, which are based on that of Bettini et al. [5]. A temporal granularity TG is said to *group-into* a temporal granularity TH if each granule in TH is a union of some set of granules in TG. Formally, TG groups-into TH if for every index j, there exists a subset S of index set such that $TH(j) = \cup_{i \in S} TG(i)$. For example, day groups into year and day also groups into month. TG is *finer-than* ($\leqslant$) TH or TH is *coarser-than* ($\geqslant$) TG if for each index i, there exists an index j such that $TG(i) \subseteq TG(j)$. Two granularities are *incomparable* if they do not have a finer-than or a coarser-than relationship. For example, week and month are incomparable.

We now define bitemporal granularity, which is pertinent to facts with which we want to associate both valid time (VT) and transaction time (TT). A bitemporal granule of granularity $TG_{TT,VT}$ is represented by a pair (i,j) such that $TG_{TT,VT}(i, j) = (TG_{TT}(i), TG_{VT}(j))$ where $TG_{TT}(i)$ and $TG_{VT}(j)$ are corresponding temporal granules for transaction time and valid time, respectively. Since by our definition of temporal granularity origins $TG_{TT}(0)$ and $TG_{VT}(0)$ exist, $TG_{TT,VT}(0, 0)$ also exists. A bitemporal granularity $TG^{(1)}_{TT,VT}$ is finer-than/coarser-than $TG^{(2)}_{TT,VT}$ only if both $TG^{(1)}_{TT} \leqslant$ (or, correspondingly $\geqslant$)$TG^{(2)}_{TT}$ *and* $TG^{(1)}_{VT} \leqslant$ (or, correspondingly $\geqslant$) $TG^{(2)}_{VT}$; otherwise, these two granularities are incomparable. An example of bitemporal granularity is sec/min, where sec and min represent the granularities for transaction time and valid time, respectively.

3.3 Temporal Indeterminacy

For many applications, it is known only approximately when a phenomenon occurred. For example, water depth of five feet at a borehole measured on $2001\text{-}04\text{-}01_{\text{day}}$ implies that the water depth was five feet sometime during the specified day but the precise hour is unknown. Thus, a determinate time at a given granularity is indeterminate at all finer granularities.

Earlier, we extended the SQL data model and query language to support *valid-time indeterminacy* [16]. An *indeterminate* instant includes an *upper* and *lower support* and an optional probability function called the *probability mass function* (*PMF*). The upper (u_{TG}) and lower (l_{TG}) support are indexes that refer to the minimum and maximum granule of a granularity TG within which an instant is located. The event *might* have occurred after l_{TG}, and *definitely* occurred by u_{TG}; l_{TG} and u_{TG} thus correspond to the Lipski bounds [31]. The upper and lower supports represent a *period of indeterminacy*, which is a contiguous set of granules. The probability mass function gives the probability that the instant is located within a given granule between the period of indeterminacy.

A *determinate* instant is indeterminate with respect to all finer granularities and an indeterminate instant is determinate with respect to some coarser granularities. So, for a determinate instant $g \in TG$ and a finer granularity TH, there exists an indeterminate instant $l_{TH} \sim u_{TH}$ such that $g = l_{TH} \sim u_{TH}$. If the upper and the lower support are the same, the instant is referred to as a determinate instant, otherwise it is an indeterminate instant. In the example above, $2001\text{-}04\text{-}01_{\text{day}}$ is a determinate instant to the granularity of day as we know the exact day when the water depth was measured. However, we do not know the hour at which it was recorded as it was sometime during 00 and 23 hours. Thus, the recording time is indeterminate at the granularity of hour.

3.4 Spatial Granularity

Montello [34] defines *geographic space* as one that cannot be experienced directly; rather, it is experienced from symbolic representations, e.g., maps. There is no comprehensive and widely accepted conceptual model of geographic space [35]: it depends on context and application. Based on the modes of thought, Sack [46] attributes different meanings to space implying that geographic space can be perceived in different ways in different cultures at different times. Philosophical discussions apart, geographic space based on Euclidean geometry is the basis for most GISs [32]. The level of abstraction of a geographic space is referred to as *resolution* [33], and multi-resolution representation is an active area of research [3, 18, 39, 43, 53-56, 62, 63]. We extend Worboys' formalism of spatial resolution [62, 63] to define a notion of horizontal and vertical spatial granularity that parallels temporal granularity.

We can view the Earth as a spheroid $\mathbf{Z}^3$ in three-dimensional Euclidean space where *position* is denoted by latitude and longitude and height/depth defined as the elevation/depth above/below sea level. Thus, position delimits an object in the geographic space and is defined with respect to a pre-specified origin [30]. Map-making requires conversion from curved to plane surfaces and *projections* [51] transform a two-dimensional surface over spheroid to another over a Cartesian plane while controlling the resulting distortions.

Worboys [62, 63] describes a formal theory for multiple representations of spatial objects. He contends that observation of a phenomenon takes place in a context, where context is represented by a schema. The *extent* of a schema specifies the size of the window on the observation. A finite collection of elements is *indiscernible* with respect to an observation if any pair of elements in the collection is indistinguishable from each other by the observation. Formally, ρ is defined as the *indiscernibility binary relation* on a collection S of elements (not necessarily a connected region on a Euclidean plane) where $u\rho v$ (read as "u is indiscernible from v") means that u and v belong to the same partition. Thus, a resolution R of S is a finite partition of S and an element $x \in R$ is referred to as a *resel*. We extend this concept of resolution to define spatial granularity, where a higher resolution corresponds to a finer granularity.

A space domain may be represented as a set (e.g., $\mathbf{R}^3$, $\mathbf{R}^2$, $\mathbf{N}^3$, $\mathbf{N}^2$) with elements referred to as *points*. However, for geographic applications, horizontal space is segregated from vertical space; correspondingly, we define *horizontal* and *vertical spatial granularities*. Intuitively, the horizontal space domain corresponds to the Earth's surface while vertical space domain corresponds to the depth/height below/above sea level. We define horizontal spatial granularity as a mapping from integers to any partition of horizontal space; the partition may arise from pixellation of space, and may be a regular square or any other shape like triangular irregular network (TIN) or even irregular shapes (e.g., county). Formally, a horizontal spatial granularity may be defined as a mapping SG_{xy} from index i to a subset of space domain such that: (i) granules from a spatial granularity do not overlap; (ii) the index set of a spatial granularity provides a contiguous encoding, though the granules in the space domain are not constrained to be contiguous in the underlying spatial domain; and (iii) origin granule $SG_{xy}(0)$ is nonempty. Examples of horizontal spatial granularities are dms-deg, dms-min and county. Each non-empty granule can have a textual representation called *label*, which can be mapped to the index integer by a mapping function called *label mapping*. For example, "45°23′E/24°35′N" is an example of a label that represents a point in space whose granularity is dms-min for both latitude/longitude. For granularities like dms-deg, space is partitioned along two perpendicular directions and the granularity is construed to be dms-deg along the two dimensions. On the other hand, county is an example of an irregular horizontal spatial granularity.

The vertical spatial domain may be important for some applications, such as geology, petroleum refining and ground water studies. Formally, a vertical spatial granularity may be defined as a mapping SG_z from index i to a subset of vertical space domain such that: (i) granules from a spatial granularity do not overlap; (ii) index order of vertical spatial granularity corresponds to vertical space domain order; (iii) the index set of a vertical spatial granularity provides a contiguous encoding, though the granules in the space domain are not constrained to be contiguous in the underlying spatial domain; and (iv) origin granule $SG_z(0)$ is nonempty. An example of vertical spatial granularity is foot. For an application, an object may need to be represented in a three-dimensional space. A three dimensional granularity is a cross product of horizontal and vertical spatial granularity. For example, a borehole in three-dimensional space is associated with a horizontal granularity of dms-degree and vertical spatial granularity of foot.

The definitions associated with vertical spatial granularity are similar to temporal granularity. We next define key terms related to horizontal spatial granularity. According to Worboys [62, 63], a partial order may be imposed on the set of all resolutions of a set S. We refer to these relationships between horizontal spatial granularities as *finer-than*/*coarser-than* relationships. In terms of discernibility relations ρ_1 and ρ_2, ρ_1 is *finer-than* ρ_2 iff $\forall\ u, v \in S$, $u\rho_1 v$ implies $u\rho_2 v$. Conversely, ρ_2 is *coarser-than* ρ_1. The set of all resolutions of S is a lattice with *top* and *bottom elements* [63]. The top element ($\top$) consists of a single resel S and the bottom element ($\bot$) has resolution where resels are singleton sets $\{s\}$ where $s \in S$. Our definition of bottom granularity in terms of ρ is defined as a partition such that $u\rho_1 v$ implies that $\forall\ i$, $u\rho_i v$. Two granularities are *incomparable* if they do not have a finer-than or coarser-than relationship. A horizontal spatial granularity SG_{xy} *groups-into* another spatial granularity SH_{xy} if for every index *j*, there exists a subset *S* of index set such that $SH_{xy}(j) = \cup_{i\in S} SG_{xy}(i)$. An *anchor* is any partition in a finer granularity corresponding to the origin of the coarser granularity. An *image* of a horizontal spatial granularity is the union of partitions in the granularity.

A *coordinate system* enumerates horizontal spatial granularities and specifies the mappings between them. A collection of horizontal spatial granularities denote a geographic coordinate system and projection mapping rules can affect conversions between projected coordinate systems for any specific granularity. Figure 3 shows a multi-coordinate system graph with three coordinate systems. (i) Geodetic Coordinate includes spatial granularities like dms-degree, dms-minute and dms-second and has a bottom granularity of dms-second. (ii) Township-range (TR) coordinate provides a square-shaped grid and includes granularities like section, quarter section and quarter-quarter section, which is equivalent to 40 acres. The bottom granularity of this coordinate system is quarter-quarter section. (iii) Political Coordinate, with irregular mappings, includes country, state and county. A double-edged arrow between any two granularities in Figure 3 represents two conversion functions, which may be regular or irregular. For example, the conversion function on Geodetic Coordinates from radian (*r*) to dms-degree is *r*/180 and is an example of regular mapping function. On the other hand, the conversion function between county and dms-degree is irregular.

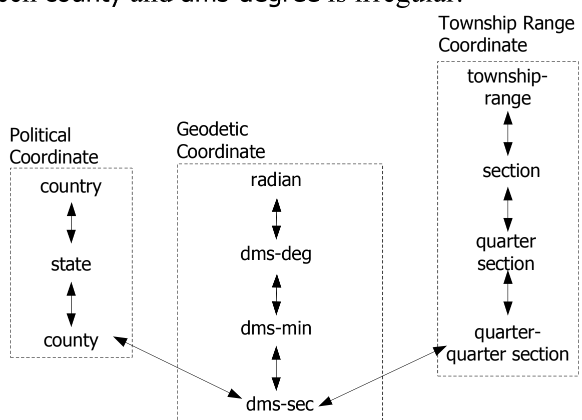


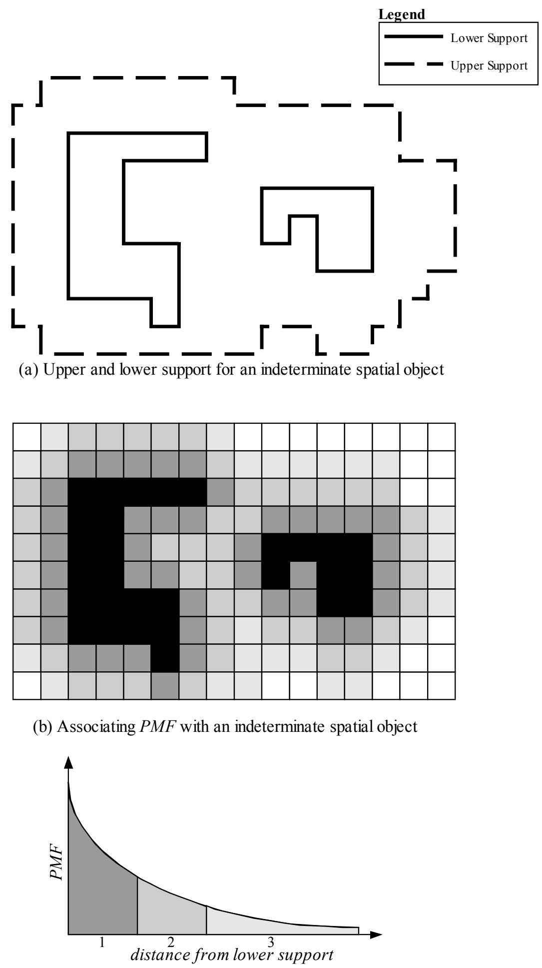
Figure 3: A multi-coordinate system graph for horizontal spatial space

POSC (Petrotechnical Open Software Corporation) [38] differentiates between *geographic* and *projected coordinate system*. The former refers to latitude/longitude coordinate systems while the latter represents the projection of geographic coordinate system on a plane. In many real world applications, coordinate transformation may involve projection transformations, which are outside the scope of this paper. We assume that such projection mapping rules across horizontal coordinate system are available; e.g., Oracle Spatial [36] maintains a table MDSYS.CS_SYS which defines valid coordinate systems and the associated conversion rules.

3.5 Spatial Indeterminacy

According to Worboys [62, 63], *imprecision* results from the limitation placed on an observation in relation with its context, represented by the schema. Thus, imprecision arises due to limitations on the granularity of the schema under which the observation is made. Analogous to temporal indeterminacy, we refer to this imprecision as spatial indeterminacy. We describe horizontal spatial indeterminacy in this sub-section. Vertical spatial indeterminacy is similar to temporal indeterminacy.

Worboys [63] defines a *resolution object* (*R*-object) with respect to a particular resolution *R* as a two stage set $\langle L, U\rangle$, where $L \subseteq R \subseteq U$. Each resel in *L* is definitely a part of the *R*-object; a resel in *U* may or may not be a part of the *R*-object and each resel not in *U* is definitely not in the *R*-object. This is similar to the upper and lower support in the definition of temporal indeterminacy and to Lipski's lower and upper bound [31]. Our definition of spatial indeterminacy is parallel to that of temporal indeterminacy with an upper and a lower support and a probability mass function.



(a) Upper and lower support for an indeterminate spatial object

(b) Associating *PMF* with an indeterminate spatial object

(c) Probability corresponding to distance from lower support

Figure 4: Spatial Indeterminacy

We describe spatial indeterminacy with an example in Figure 4 and map it to Worboys' [62, 63] model of imprecision. In Figure 4(b), black grids denote deterministic regions of a spatial object and lighter gray areas show those locations that *may* be part of the spatial object. As shown in Figure 4(a), Worboys' model [63] of imprecision would correspond to specifying only the upper and lower support. As illustrated by Figure 4(b), we also include a sample probability mass function (Figure 4(c)) that allows gradation in indeterminacy to be specified, as shown by three shades of gray grids; however, any number of levels of indeterminacy may be specified. As with temporal indeterminacy [16], the continuous *PMF* is discretized when associating with spatial granularities. The *PMF* in Figure 4(c) corresponds to the intuitive notion of "probably smaller"; spatial granules closest to the lower support are associated with a probability to the left of Figure 4(c) and those farthest from the lower support and, thus, closest to the upper support are associated with a probability to the right of Figure 4(c).

We next show how the ontological concepts described in this section have been incorporated into a conceptual model that can capture the semantics related to granularities and indeterminacy.

4. A Granularity-based Spatiotemporal Conceptual Model

Conceptual database design takes the user requirements as an input and transforms them into a high-level conceptual schema using a conceptual model. We integrate conceptual modeling and spatiotemporal concepts defined in the previous section to propose a spatiotemporal conceptual model called ST-USM. ST-USM uses an annotation-based approach that divides spatiotemporal conceptual design into two steps: (i) capture the *current reality* of the application using conventional conceptual model *without considering the spatial aspects*, and only then (ii) annotate the schema with spatiotemporal semantics of the application. USM [40] is the base model for ST-USM. ST-USM allows the database designer to focus first on the non-temporal and non-spatial aspects of the application; we refer to this schema as a *Core USM Schema*. The data analyst then augments the core USM schema with annotations to capture spatial and temporal aspects of the application; we refer to this as an *ST-USM Schema* or an *Annotated Schema*. We have defined the semantics of the annotated ST-USM schema in terms of USM constructs and the resulting schema is referred to as a *Translated USM Schema*; this translated USM schema can be used for subsequent mapping to a logical schema. In this section, we first outline key concepts of USM. Next, we describe how granularity- and indeterminacy-related concepts have been embedded into USM resulting in ST-USM.

USM is an extended version of the Entity Relationship model [10]. The various levels of abstractions supported by typical conceptual models, e.g., [2, 10, 17, 40, 47], include *entity class*, *attribute* and *relationship*. In USM, an entity class, an attribute and a relationship are graphically represented by a rectangle, an ellipse and a diamond, respectively. An object in the real world is referred to as an *entity*. Objects have properties called *attributes* A_i that describe the entity. Each attribute A_i has an associated value set referred to as the *domain*, $dom(A_i)$. Entities are grouped into *entity classes* based on some common semantic characteristics and may be formally defined as $E = \cup_i (A_i, dom(A_i))$. An entity e in an entity class E may be designated as $e(E)$ and a set of entities from an entity class, i.e., an *entity set*, is represented as $S(E)$, where $e(E) \in S(E)$.

Entity classes may be considered alternatively as *simple*, *interaction*, *superclass*, *composite* and *grouping classes*. USM defines several types of relationships, which includes *interaction*, *generalization/specialization*, *composite* and *grouping* relationships. Each of these abstractions has a well-defined semantics, which help clarify the meaning of the data in a database. An interaction relationship refers members of one entity class to one of more entity classes. Attributes may be created to describe an interaction relationship resulting in an interaction class. Generalization is a form of abstraction in which similar objects are related to higher-level generic objects referred to as superclass; the constituent objects may be considered as a specialization of the generic object. For example, one may define GROUND_WATER_STATION as a superclass with SPRING_SITE and BORE_HOLE_SITE as its subclasses. Such an abstraction would imply that there are certain common attributes in GROUND_WATER_STATION that apply to both SPRING_SITE and BORE_HOLE_SITE. A composite relationship defines a new class called a composite class that has a subset of the other class, referred to as a *base class*, as its members. The USM definition of the composite class requires that each member of it must also be a subtype of the base class. A grouping relationship defines a new class called a grouping class whose members are physically or logically made up of members or sets of members from some other entity classes, called *component classes*. The grouping establishes a part-of or a property-of relationship. For example, BORE_HOLE may be defined as a grouping class with CASING and OPENING as the component classes. This implies that a set of casing and opening are "grouped" to form a borehole. Note that BORE_HOLE is not of the same type as CASING or OPENING.

4.1 Syntax for Granularity Support

In this section, we describe the syntax for annotating spatial and temporal aspects of the application. The formal syntax related to ST-USM annotations is given in the Appendix. Our annotation syntax was influenced by that of spatiotemporal queries [57], which in turn is based on notations for queuing systems [28].

As shown in the Appendix, the overall structure of temporal and spatial annotation is:

⟨temporal annotation⟩ // ⟨spatial annotation⟩.

The temporal and spatial annotations are separated by a double forward slash (//).

ST-USM annotations have positional significance. The temporal annotation first specifies existence time (or valid time) followed by transaction time. The temporal annotation for existence time and transaction time is segregated by a forward slash (/). Any of these aspects can be specified as not being relevant to the associated conceptual construct by using a "-". The valid time or existence time can be modeled as an event (E) or a state (S) and has an associated temporal granularity. Similarly, transaction time is modeled with annotation T. ST-USM supports the expression of multiple granularities in the schema, e.g., sec (second), min (minute), hr (hour) and day. Though users are free to specify their own granularities (standard or user-defined) for valid time/existence time that for transaction time is system-defined. Below we give some example of temporal annotations:

Example 1: "S (day) / - //" associated with an entity class denotes that entities in the entity class have existence time with a temporal granularity of day represented as a set of states (S).

Example 2: "E (min) / T //" associated with an entity class denotes that entities in the entity class are bitemporal. The temporal granularity of the event (E) is minute. Additionally, we also need to capture transaction time associated with the entities. The granularity associated with transaction time is not specified in example 2, as it is system-defined.

Example 3: "S (sec) / - //" associated with amount denotes that amount is associated with time period (S) expressed in second.

Example 4: "E (min) / - //" associated with water_depth denotes that water depth is associated with instants (E) expressed in minute.

The annotation also includes a formalism to model indeterminacy; e.g., an indeterminate state with a probability distribution function [16] is designated as S~. Many times the probability distribution may not be known and a user may make a simplified assumption of a uniform distribution and in that case, indeterminate state is represented as S+-.

The spatial annotations follow a double forward slash (//) and includes the geometry in *x*-, *y*- and *z*-dimension; each dimension is segregated by a forward slash (/). Below we give some examples of spatial annotations:

Example 5: "// P(dms-deg) / P(dms-deg) / -" for an entity class describes a spatial entity that has a geometry of points on an *x*-*y* plane. The associated granularity is dms-degree. This is the annotation for SPRING_SITE.

Example 6: "// P(dms-deg) / P(dms-deg) / L(ft)" defines the geometry of an entity class that is a point (P) in the *x*-*y* plane (expressed in dms-deg) and a line (L) in the *z*-dimension expressed in foot. This is the annotation for BORE_HOLE_SITE.

Example 7: "// P(qqs) / P(qqs) / -" defines the geometry of a PUMPLIFT that is point (P) on quarter-quarter section (qqs).

Example 8: The annotation "// R(GWFM grid) / R(GWFM grid) / L(GWFM layer)" for SIMULATED_OUTPUT denotes a region (R) on horizontal space and line (L) in the vertical space. The associated granularities are GWFM grid and GWFM layer, respectively.

Note that the geometries associated with geospatial objects are restricted to only seven possibilities; other combinations are not applicable to geospatial applications. A point, a line or a region on the surface of the Earth are denoted by "//P($\langle g_{sxy} \rangle$)/P($\langle g_{sxy} \rangle$)/-", "//L($\langle g_{sxy} \rangle$)/L($\langle g_{sxy} \rangle$)/-" or "//R($\langle g_{sxy} \rangle$)/R($\langle g_{sxy} \rangle$)/-", respectively; $\langle g_{sxy} \rangle$ is the horizontal spatial granularity. Additionally, geospatial objects in three-dimensional space can have the following geometries: "//P($\langle g_{sxy} \rangle$)/P($\langle g_{sxy} \rangle$)/P($\langle g_{sz} \rangle$)", "//P($\langle g_{sxy} \rangle$)/P($\langle g_{sxy} \rangle$)/L($\langle g_{sz} \rangle$)", "//L($\langle g_{sxy} \rangle$)/L($\langle g_{sxy} \rangle$)/L($\langle g_{sz} \rangle$)" and "//R($\langle g_{sxy} \rangle$)/R($\langle g_{sxy} \rangle$)/L($\langle g_{sz} \rangle$)", where $\langle g_{sz} \rangle$ is the vertical spatial granularity. "//L($\langle g_{sxy} \rangle$)/L($\langle g_{sxy} \rangle$)/L($\langle g_{sz} \rangle$)" implies a geometry of line on the horizontal surface with an associated depth/height; similarly, "//R($\langle g_{sxy} \rangle$)/R($\langle g_{sxy} \rangle$)/L($\langle g_{sz} \rangle$)" implies a solid with a geometry of region on the horizontal surface with an associated depth/height.

Having described the syntax for temporal and spatial granularity support using our annotation-based approach, we next define the related semantics.

4.2 Temporal Granularity Support

We formally define a temporal entity class in terms of temporal granularity and then define the associated semantics with respect to ST-USM.

A *temporal entity class* refers to entities with associated existence time and/or transaction time. A temporal entity class implies that the membership of an entity in the entity set is temporal. We assume that a temporal entity class exists during the entire modeled time. The existence time represents the lifespan of an entity and defines the time when facts associated with an entity can be true in the miniworld. Similarly, we can capture the transaction time associated with an entity, which may be important for applications requiring traceability. In the real world, all objects are temporal. However, an entity class may not be modeled as temporal if the user is not interested in the lifespan of an entity or if the lifespan is not known.

A temporal entity with existence time is associated with an *existence predicate* $\varphi_{E,\mathrm{et}}$ that defines the lifespan of entities in terms of an existence time granularity $TG_{E,\mathrm{et}}$. $\varphi_{E,\mathrm{et}}$ takes an entity from the entity set $S(E)$ and an integer index i associated with the image of a granularity and returns a Boolean **B** which is true if the entity e exists at the time granule $TG_{E,\mathrm{et}}(i)$.

$$\varphi_{E,\mathrm{et}}: S(E) \times \mathbf{Z} \rightarrow \mathbf{B}$$

There are two constraints on the existence predicate $\varphi_{E,\text{et}}$:

(i) $\forall\, e \in S(E),\ \varphi_{E,\text{et}}(e, i) \Rightarrow (TG_{E,\text{et}}(i) \subseteq Image(TG_{E,\text{et}}))$

(ii) $\forall\, e \in S(E),\ \exists\, i \in \mathbf{Z},\ \varphi_{E,\text{et}}(e, i)$

The first constraint states that temporal entities can exist only within the defined image of the granularity. Second, every entity exists at some granule within the image of the granularity. Intuitively, if a temporal entity does not exist during any granule within the image, it is meaningless to store it in the database.

Similarly, a temporal entity with transaction time is associated with a *transaction time predicate* $\varphi_{E,\text{tt}}$ that defines the transaction time of an entity in terms of a transaction time granularity TG_{tt}. The transaction time granularity is only defined for all points less than now. If a transaction timestamp includes *Until Changed* (*UC*), a special transaction time marker, it denotes that the associated fact is current in the database. Unlike the existence time granularity, which can be specified by users, the transaction time granularity is system-defined. The transaction time predicate takes an entity from the entity set and an integer index associated with the image of a transaction time granularity and evaluates to true if the entity is current in the database.

$$\varphi_{E,\text{tt}}: S(E) \times \{\mathbf{Z} \cup UC\} \rightarrow \mathbf{B}$$

The constraints on the transaction time predicate are similar to those on the existence time predicate.

A bitemporal entity class is associated with a *bitemporal predicate* $\varphi_{E,\text{tt,et}}$ that defines a bitemporal lifespan for an entity in terms of an existence time granularity $TG_{E,\text{et}}$ and a transaction time granularity TG_{tt}. $\varphi_{E,\text{tt,et}}$ takes an entity from the entity set and a pair of granularity indexes from existence and transaction time image and evaluates to true if the entity exists in the bitemporal space.

$$\varphi_{E,\text{tt,et}}: S(E) \times \{\mathbf{Z} \cup UC\} \times \mathbf{Z} \rightarrow \mathbf{B}$$

The constraints on a bitemporal existence predicate are similar to those on the existence predicate.

An indeterminate entity is associated with an *indeterminate existence predicate* $\varphi_{E,\sim\text{et}}$ defined in terms of an existence granularity $TG_{E,\text{et}}$. $\varphi_{E,\sim\text{et}}$ takes an entity from the entity set, an index integer associated with the existence image, a *pmf* and the *plausibility level* (*pl*) [16], which is a positive rational number not greater than one, and evaluates to true if the entity exists at that plausibility. An entity exists only for those granules that lie within the existence granularity image where there is some plausibility of occurrence.

$$\varphi_{E,\sim\text{et}}: S(E) \times \mathbf{Z} \times PMF \times \mathbf{R} \rightarrow \mathbf{B}$$

There are two constraints on the indeterminate existence predicate $\varphi_{E,\sim\text{et}}$

(i) $\forall\, e \in S(E),\ \varphi_{E,\sim\text{et}}(e, i, pmf, pl) \Rightarrow (TG_{E,\sim\text{et}}(i) \subseteq Image(TG_{E,\sim\text{et}}))$
where $pmf \in PMF$ and $0 < pl \leq 1$

(ii) $\forall\, e \in S(E),\ \exists\, i \in \mathbf{Z},\ \varphi_{E,\sim\text{et}}(e, i, pmf, 1.0)$

The uniform indeterminate existence predicate is defined as:

$$\varphi_{E,+\text{-et}}: S(E) \times \mathbf{Z} \times \text{Uniform Distribution} \times \mathbf{R} \rightarrow \mathbf{B}$$

The constraints on a uniform indeterminate existence predicate are similar to those on an indeterminate existence predicate.

4.2.1 Temporal Entity Class

Having defined temporal entity class abstractly in the previous subsection, we describe the semantics of a simple temporal entity class in ST-USM using USM constructs.

Figure 5 shows a temporal entity class for which we want to capture the existence time expressed as state (S) with ⟨g_{et}⟩ as the granularity name (e.g., day). Based on the users' requirements, the database analyst simply annotates the ⟨ENTITY_CLASS⟩ as "S(⟨g_{et}⟩)/-//" and does not need to contend with the complexity of the underlying semantics or the associated constraints.

In order to express the semantics of a temporal entity class, we need to specify a TEMPORAL_GRANULARITY in which the evolution of a temporal object is embedded. The relationship ⟨ENTITY_CLASS⟩_has_ET associates an entity with a corresponding TEMPORAL_GRANULARITY. Each TEMPORAL_GRANULARITY is uniquely identified by a granularity_name, shown by the underlined attribute. An extent is the smallest time interval that includes the image of a granularity and is expressed by two indexes, minimum and maximum. Each anchor_gran is a recursive relationship (i.e., a relationship where an entity from the same entity set can play different roles) such that each participating granularity optionally has an anchor (0:1) and each granularity is an anchor for 0 to many (0:M) other granularities. The anchor of a granularity *TG* is the first index of a strictly finer granularity that corresponds to the origin of this granularity, i.e., *TG*(0). All granularities except the bottom granularity have an associated anchor. A finer-than and a coarser-than relationship between granularities are denoted by a recursive relationship groups_into, where one entity plays the role of finer-than and the other the role of coarser-than. The relationships anchor_gran together with groups-into helps create a granularity graph, which can help a user choose the level of detail associated with facts.

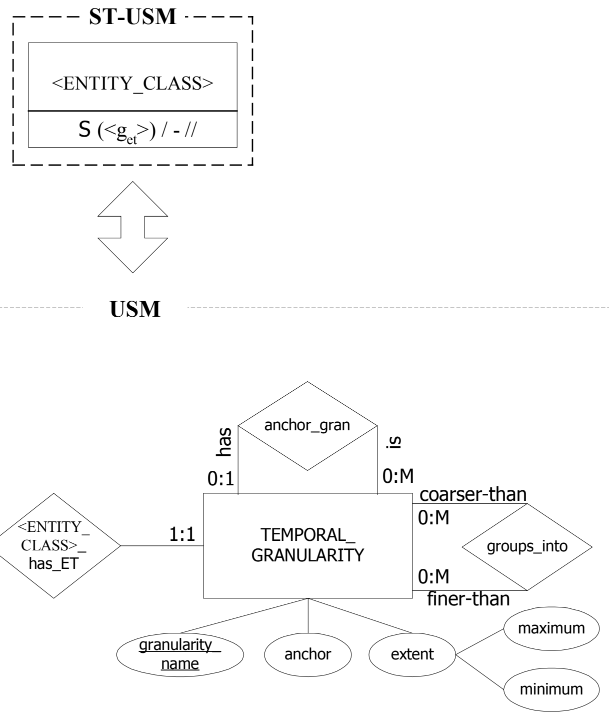


Figure 5: Temporal Entity Class in ST-USM and its semantics in USM

As described in a previous section, entities in an ⟨ENTITY_CLASS⟩ can interact with time in two ways, resulting in existence time and transaction time. A temporal entity with existence time may have a set of event_instants or state_periods associated with it. A time period is represented with indexes begin and end. A double-lined ellipse in USM denotes a multi-valued attribute. For example, state_periods is represented as multi-valued attribute and represents a set of state periods (i.e., a temporal element) associated with an entity. Figure 5 illustrates the complete semantics of a temporal entity class. In the subsequent examples, we will not show the complete semantics associated with a temporal granularity. This is done primarily to help the reader focus on the additional semantics not explicated in the previous sections.

We now describe the constraints on temporal entities. These constraints are implicit in the ST-USM schema but are explicit in the translated USM schema.

Constraint 4.2.1: The existence time for all the entities of a temporal entity class have the same associated granularity. In ST-USM, the granularity associated with the existence time of an entity class is denoted by $\langle g_{et} \rangle$, which corresponds to the granularity_name in the corresponding translated USM schema. For example, some of the valid values of the granularity_name are day, hour and minute.

$$\forall e \in S(\langle \text{ENTITY\_CLASS} \rangle),\ e.\langle \text{ENTITY\_CLASS} \rangle\text{\_has\_ET.TEMPORAL\_GRANULARITY}(\text{granularity\_name}) = \langle g_{et} \rangle$$

Constraint 4.2.2: Every entity has an associated temporal element containing correctly specified periods.

$$\forall e \in S(\langle \text{ENTITY\_CLASS} \rangle),\ \exists p \in e.\text{state\_periods},\ p.\text{begin} \le p.\text{end}$$

Constraint 4.2.3: Each TEMPORAL_GRANULARITY has a lower and an upper bound referred to as minimum and maximum; these bounds are well formed.

$$\forall e \in S(\text{TEMPORAL\_GRANULARITY}),\ e(\text{extent.minimum}) \le e(\text{extent.maximum})$$

Constraint 4.2.4: All the granularities, except one, have an anchor. The bottom granularity is allowed not to have an anchor.

$$\forall e \in S(\text{TEMPORAL\_GRANULARITY}),\ \neg\ \text{has}(e.\text{anchor\_gran}) \Rightarrow$$
$$\neg\ (\exists\ e_2 \in \text{TEMPORAL\_GRANULARITY} \wedge e \ne e_2 \wedge \neg\ \text{has}(e_2.\text{anchor\_gran}))$$

Constraint 4.2.5: For a temporal granularity, if an anchor does not exist then that is the bottom granularity that does not have any granularity finer than it; in other words, it cannot take the role of coarser-than in the relationship groups-into.

$\forall e \in S(\text{TEMPORAL\_GRANULARITY}),\ \neg\ \text{has}(e.\text{anchor\_gran}) \Rightarrow \neg\ \text{coarser-than}(e.\text{groups\_into})$

Constraint 4.2.6: State periods of an entity are well formed. We assume *closed-open representation* [49], i.e., the begin index is contained in the period while the index corresponding to the end is not.

$\forall e \in S(\langle\text{ENTITY\_CLASS}\rangle),\ \forall\, p \in e.\text{state\_periods},\ p.\text{begin} < p.\text{end}$

Constraint 4.2.7: Temporal elements are well formed. A temporal element is defined as a union of non-overlapping time intervals.

$\forall e \in S(\langle\text{ENTITY\_CLASS}\rangle),\ \forall\, p_1, p_2 \in e.\text{state\_periods},\ p_1.\text{begin} < p_2.\text{begin} \Rightarrow p_1.\text{end} \le p_2.\text{begin}$

Constraint 4.2.8: The extent of a temporal granularity defines the upper and lower bounds for any temporal element. In other words, a temporal element cannot include an index that is larger than the corresponding extent.maximum or smaller than the corresponding extent.minimum.

$\forall e \in S(\langle\text{ENTITY\_CLASS}\rangle),\ \forall\, p \in e.\text{state\_periods},$
$e.\langle\text{ENTITY\_CLASS}\rangle\text{\_has\_ET.TEMPORAL\_GRANULARITY (extent.minimum)} \le p.\text{begin} <$
$p.\text{end} \le e.\langle\text{ENTITY\_CLASS}\rangle\text{\_has\_ET.TEMPORAL\_GRANULARITY (extent.maximum)}$

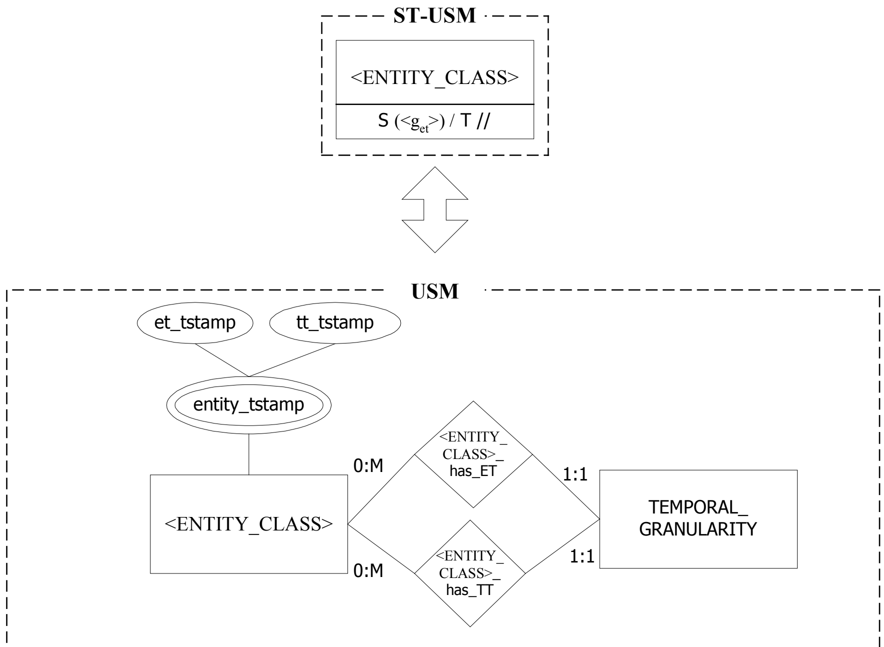


Figure 6: Bitemporal entity class in ST-USM and its semantics in USM

We next describe the semantics associated with a bitemporal entity class. As shown in Figure 6, a bitemporal entity class in ST-USM needs to include both existence time and transaction time annotation. In Figure 6, there are two relationships between the ⟨ENTITY_CLASS⟩ and the TEMPORAL_GRANULARITY, ⟨ENTITY_CLASS⟩_has_ET and ⟨ENTITY_CLASS⟩_has_TT. While the former defines an association between an entity *e* and its existence granularity (i.e., granularity_name = $\langle g_{et}\rangle$), the latter defines the association between the entity *e* and its transaction time granularity (i.e., granularity_name = g_{tt} where g_{tt} is a system-defined granularity).

Besides the constraints 4.2.1, 4.2.3, 4.2.4 and 4.2.5, there are additional constraints related to a bitemporal entity class.

Constraint 4.2.9: Every entity has an associated bitemporal granule within specified extent.

$\forall e \in S(\langle\text{ENTITY\_CLASS}\rangle),\ \exists p \in e.\text{entity\_tstamp},\ e.\langle\text{ENTITY\_CLASS}\rangle\text{\_has\_ET.TEMPORAL\_GRANULARITY (extent.minimum)} \le$
$p.\text{et\_tstamp} \le e.\langle\text{ENTITY\_CLASS}\rangle\text{\_has\_ET.TEMPORAL.GRANULARITY (extent.maximum)} \wedge$
$e.\langle\text{ENTITY\_CLASS}\rangle\text{\_has\_TT.TEMPORAL\_GRANULARITY (extent.minimum)} \le p.\text{tt\_tstamp}) \le$
$e.\langle\text{ENTITY\_CLASS}\rangle\text{\_has\_TT.TEMPORAL\_GRANULARITY (extent.maximum)}$

Constraint 4.2.10: For transaction time, we do not need to specify the granularity since it is system-defined. For the entire schema, there is a single transaction time granularity, g_{tt}.

∀*e* ∈ *S*(⟨ENTITY_CLASS⟩), *e*.⟨ENTITY_CLASS⟩_has_TT.TEMPORAL_GRANULARITY (granularity_name) = g_{tt}

Constraint 4.2.11: The bitemporal timestamp is such that the existence and transaction timestamps are within the extent of the respective granularities.

∀ *e* ∈ *S*(⟨ENTITY_CLASS⟩), ∀*p* ∈ *e*.entity_tstamp,
e.⟨ENTITY_CLASS⟩_has_ET.TEMPORAL_GRANULARITY (extent.minimum) ≤ *p*.et_tstamp ≤
e.⟨ENTITY_CLASS⟩_has_ET.TEMPORAL.GRANULARITY (extent.maximum) ∧
e.⟨ENTITY_CLASS⟩_has_TT.TEMPORAL_GRANULARITY (extent.minimum) ≤ *p*.tt_tstamp ≤
e.⟨ENTITY_CLASS⟩_has_TT.TEMPORAL_GRANULARITY (extent.maximum)

4.2.2 Indeterminate Temporal Entity Class

For many applications, the occurrence time may not be known precisely. An indeterminate state is designated as S~ and an indeterminate event is designated as E~. We give here the semantics for an indeterminate state; that for an indeterminate event can be similarly defined. An indeterminate state is composed of an upper and a lower support for both begin and end for each period, and a pmf (probability mass function) that gives the probability that a given begin/end instant is located within a given granule within the bounds (begin.lower and begin.upper, and end.lower and end.upper, respectively).

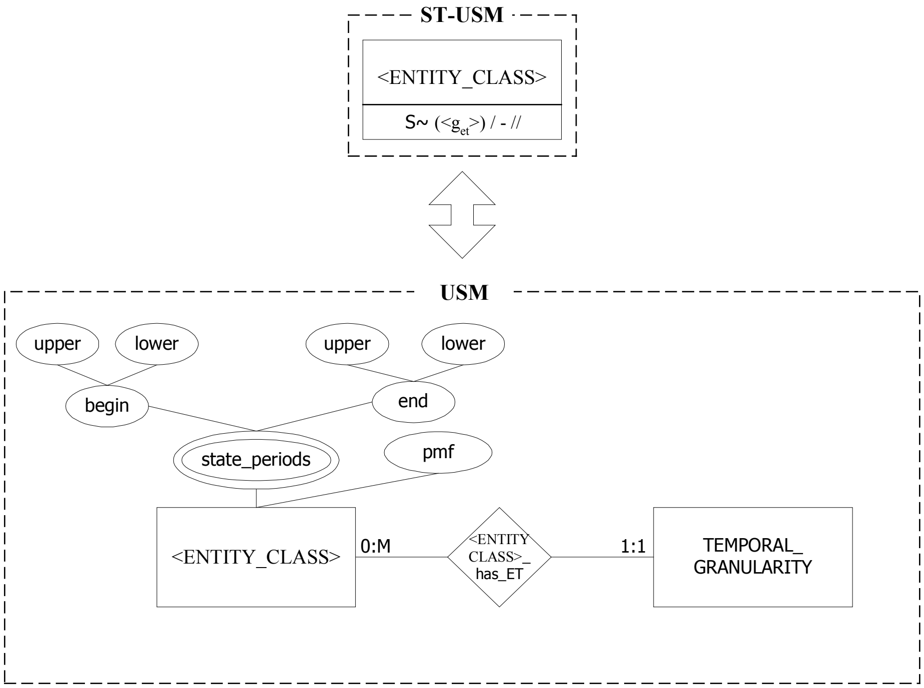


Figure 7: Temporal indeterminate entity class with state periods in ST-USM and its semantics in USM

Besides constraints 4.2.1, 4.2.3, 4.2.4 and 4.2.5, there are two additional constraints associated with an indeterminate entity class.

Constraint 4.2.12: The supports for begin and end granule lie within the extent of the corresponding granularity. Additionally the period of indeterminacy for each period within a temporal element cannot overlap. If an upper is equal to a lower, it denotes a determinate instant.

∀*e* ∈ *S*(⟨ENTITY_CLASS⟩), ∀*p* ∈ *e*.state_periods, *e*.⟨ENTITY_CLASS⟩_has_ET.TEMPORAL_GRANULARITY (extent.minimum) ≤
p.begin.lower ≤ *p*.begin.upper < *p*.end.lower ≤ *p*.end.upper ≤
e.⟨ENTITY_CLASS⟩_has_ET.TEMPORAL.GRANULARITY (extent.maximum)

Constraint 4.2.13: Indeterminate state periods are well formed.

∀*e* ∈ *S*(⟨ENTITY_CLASS⟩), ∀p_1, p_2 ∈ *e*.state_periods, p_1.begin.lower < p_2.begin.lower ⇒ p_1.end.upper < p_2.begin.lower

Many times the probability distribution may not be known and a user may make a simplified assumption of a uniform distribution. As a practical matter, the assumption of a uniform distribution is often made; that is the reason why we have included this special case into our annotation syntax. We give an example of an indeterminate state with a uniform distribution *pmf*; an indeterminate event can be similarly defined. As shown in Figure 8, a simplified version of an indeterminate state includes an index corresponding to begin and end, and precision for each temporal element. A precision has the same granularity as the temporal element. The annotation syntax associated with a simplified indeterminate state is S+- and that with an event is E+-.

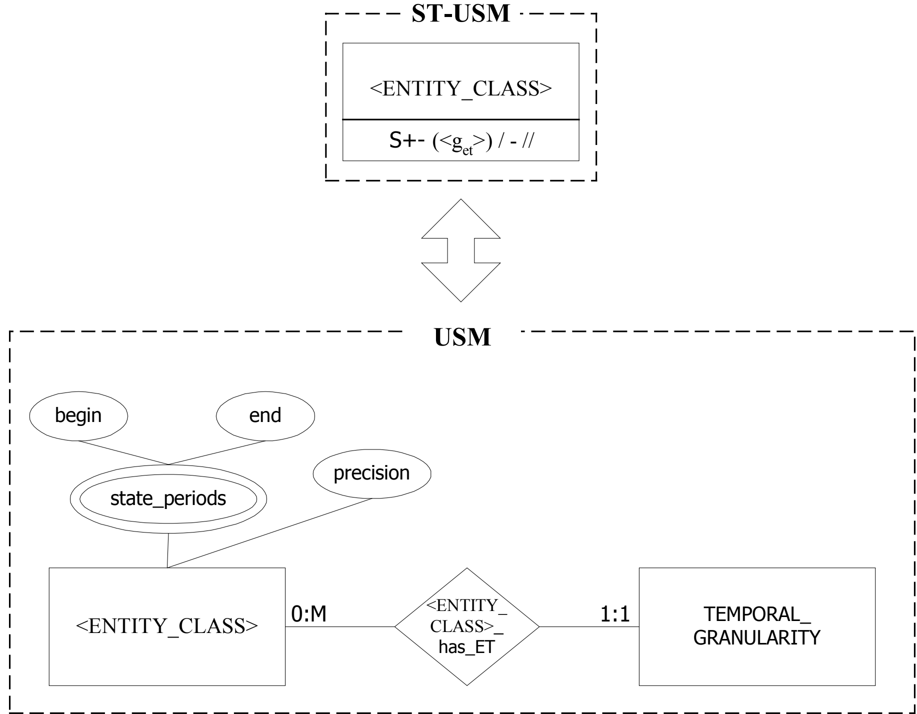


Figure 8: Simplified version of temporal indeterminate entity class

Constraint 4.2.14: For any period within a temporal element, the periods of indeterminacy cannot overlap.

$\forall\, e \in S(\langle \text{ENTITY\_CLASS} \rangle), \forall p \in e.\text{state\_periods}, p.\text{begin} + e(\text{precision}) < p.\text{end} - e(\text{precision})$

Constraint 4.2.15: Additionally, indeterminate state periods are well formed

$\forall\, e \in S(\langle \text{ENTITY\_CLASS} \rangle), \forall p_1, p_2 \in e.\text{state\_periods}, p_1.\text{begin} < p_2.\text{begin} \Rightarrow p_1.\text{end} + e(\text{precision}) < p_2.\text{begin} - e(\text{precision})$

4.3 Spatial Granularity Support

A *spatial entity class* refers to entities with an associated shape and position, which can be used to locate them in a two- or three-dimensional space. In this subsection, we first define a spatial entity in terms of spatial granularity and then describe the associated semantics of a spatial entity class in ST-USM.

A spatial entity in a horizontal space domain is associated with a *horizontal geometry predicate* $\psi_{E,xy}$ that defines the location of an entity in terms of horizontal spatial granularity $SG_{E,xy}$. $\psi_{E,xy}$ takes an entity from the entity set and an integer i from the image of a horizontal spatial granularity and evaluates to true if the entity exists at that granule $SG_{E,xy}(i)$.

$$\psi_{E,xy}: S(E) \times \mathbf{Z} \rightarrow \mathbf{B}$$

There are two constraints on the horizontal geometry predicate

(i) $\forall\, e \in S(E), \psi_{E,xy}(e, i) \Rightarrow (SG_{E,xy}(i) \subseteq Image(SG_{E,xy}))$

(ii) $\forall\, e \in S(E), \exists\, i \in \mathbf{Z}, \psi_{E,xy}(e, i)$

The first constraint implies that any partition of the horizontal space domain denoted by an index i lies within the image of the granularity. The second constraint states that a spatial entity must exist somewhere within the defined image; i.e., each spatial entity has an associated geometry.

An indeterminate spatial entity is associated with an *indeterminate geometry predicate* $\psi_{E,\sim xy}$ defined in terms of spatial granularity $SG_{E,xy}$. $\psi_{E,\sim xy}$ takes an entity from an entity set, an index integer associated with the spatial granularity, a horizontal pmf_{xy} and a plausibility level, and evaluates to true if the entity exists at that plausibility.

$$\psi_{E,\sim xy}: S(E) \times \mathbf{Z} \times PMF_{xy} \times \mathbf{R} \rightarrow \mathbf{B}$$

The constraints on the indeterminate geometry predicate are similar to those of indeterminate existence predicate.

A spatial entity in 3-dimensional space is associated with 3-D geometry predicate $\psi_{E,xy,z}$ that defines the location of an entity in terms of horizontal and vertical spatial granularities, i.e., $SG_{E,xy}$ and $SG_{E,z}$. $\psi_{E,xy,z}$ takes an entity from an entity set and a pair (i, j) from the image of horizontal and spatial granularities and evaluates to true if the entity exists at the granules $SG_{E,xy}(i)$ and $SG_{E,z}(j)$, respectively. The associated constraints are similar to those described above.

$$\psi_{E,xy,z}: S(E) \times \mathbf{Z} \times \mathbf{Z} \rightarrow \mathbf{B}$$

4.3.1 Spatial Entity Class

We now describe the semantics of a spatial entity class in ST-USM using USM and the constraints.

Figure 9 shows a spatial entity class, which includes a geometry of points with a horizontal spatial granularity as $\langle g_{xy} \rangle$.

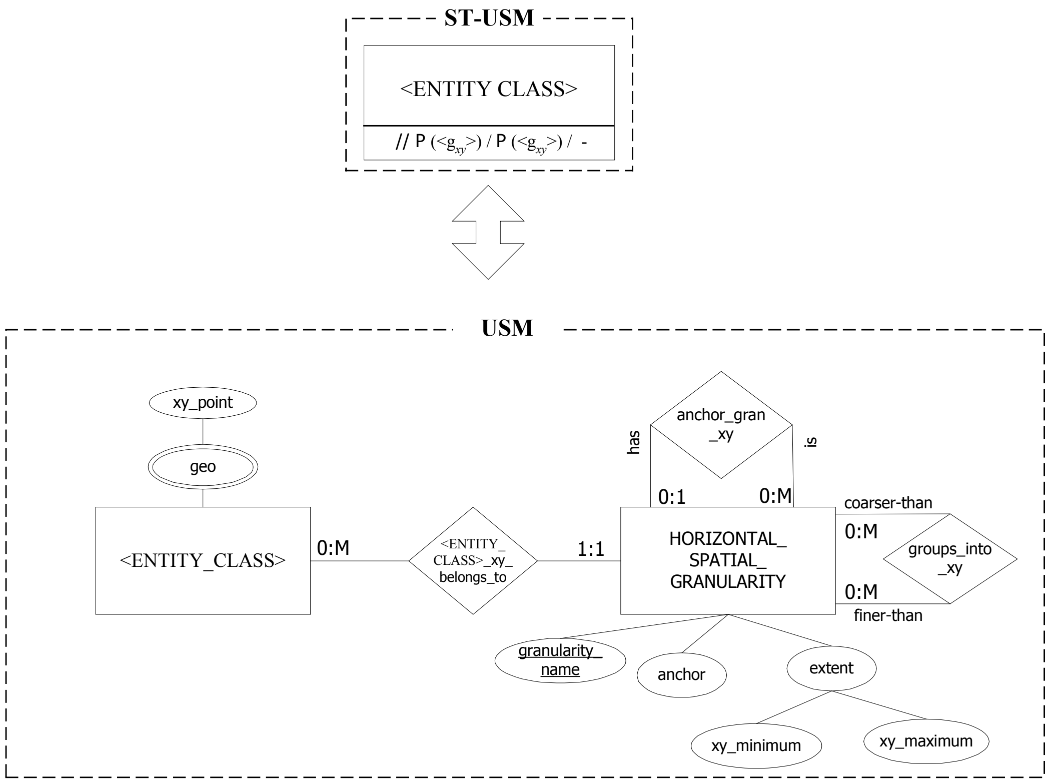


Figure 9: A spatial entity class in ST-USM in horizontal space and its semantics in USM

In order to specify the semantics of a spatial entity class, we need to define the HORIZONTAL_SPATIAL_GRANULARITY in which the geometry of a spatial entity is embedded. A HORIZONTAL_SPATIAL_GRANULARITY is uniquely specified by granularity_name. The extent is the minimum-bounding rectangle that includes the image of the granularity. The recursive relationships groups_into_xy and anchor_gran_xy are similar to those in the temporal entity class. The relationship ⟨ENTITY_CLASS⟩_xy_belongs_to relates a spatial entity with a corresponding horizontal spatial granularity. We next describe the associated constraints.

Constraint 4.3.1: All entities in a spatial entity class must have the same horizontal spatial granularity.

$\forall e \in S(\langle$ENTITY_CLASS$\rangle)$,
e.⟨ENTITY_CLASS⟩_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY (granularity_name) = $\langle g_{xy} \rangle$

Constraint 4.3.2: Every entity has an associated geometry (e.g., a point in horizontal space) within the specified extent.

$\forall e \in S(\langle$ENTITY_CLASS$\rangle)$, $\exists g \in e$.geo,
e.⟨ENTITY_CLASS⟩_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY (extent.xy_minimum) ≤
g.xy_point ≤ *e*.⟨ENTITY_CLASS⟩_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY (extent.xy_maximum)

Constraint 4.3.3: Extent is well formed.

$\forall e \in S$(HORIZONTAL_SPATIAL_GRANULARITY), e(extent.xy_minimum) < e(extent.xy_maximum)

Constraint 4.3.4: The indexes corresponding to the geometry of a spatial object lie within xy_minimum and xy_maximum.

$\forall e \in S(\langle$ENTITY_CLASS$\rangle)$, $\forall g \in e$.geo,
$e.\langle$ENTITY_CLASS$\rangle$_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY (extent.xy_minimum)
$\leq g$.xy_point $\leq e.\langle$ENTITY_CLASS$\rangle$_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY (extent.xy_maximum)

Constraint 4.3.5: All granularities except one (i.e., the bottom granularity) have an anchor.

$\forall e \in S$(SPATIAL_GRANULARITY), $\neg$ has(e.anchor_gran_xy) $\Rightarrow \neg$ ($\exists$ $e_2 \in$ SPATIAL_GRANULARITY $\wedge$ $e_2 \neq e_2$ $\wedge$
$\neg$ has(e_2.anchor_gran_xy)

Constraint 4.3.6: The bottom granularity does not have any granularity that is finer than it; in other words it cannot take the role of coarser-than in the relationship groups-into.

$\forall e \in S$(SPATIAL_GRANULARITY), $\neg$ has(e.anchor_gran_xy) $\Rightarrow \neg$ coarser-than(e.groups_into_xy)

Similarly, a spatial entity embedded in a three-dimensional space is associated with HORIZONTAL_SPATIAL_GRANULARITY and VERTICAL_SPATIAL_GRANULARITY (Figure 10). A spatial object in a three-dimensional space has two associated relationships, ⟨ENTITY_CLASS⟩_xy_belongs_to and ⟨ENTITY_CLASS⟩_z_belongs_to corresponding to its horizontal and vertical spatial granularities.

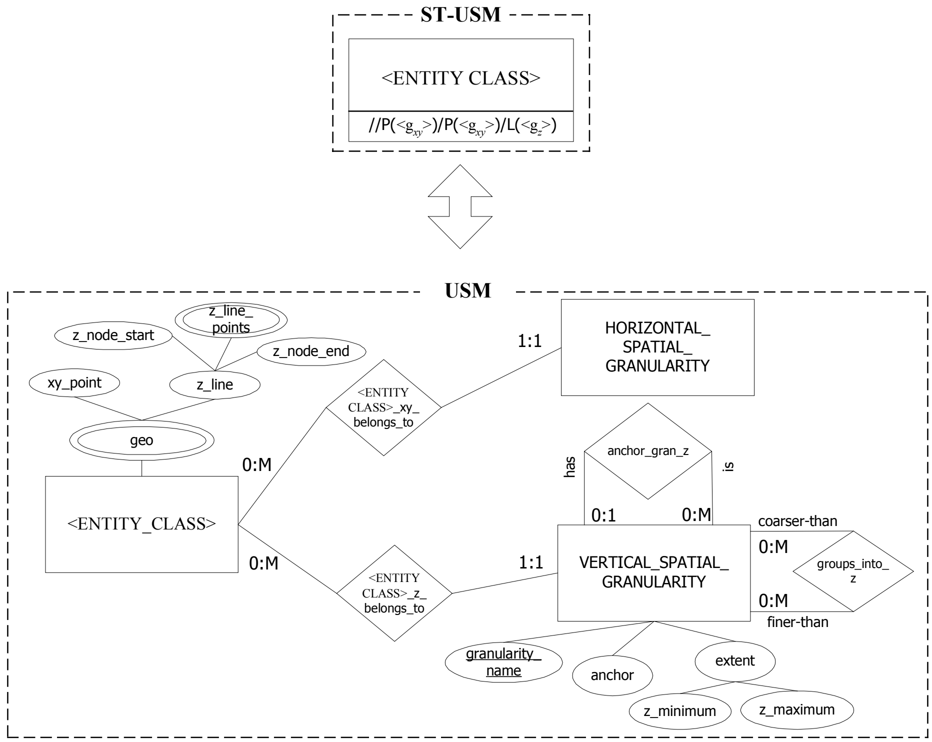


Figure 10: A spatial entity in three-dimensional space and its semantics in USM

We have not shown the details associated with HORIZONTAL_SPATIAL_GRANULARITY as they have already been described in Figure 9. The associated constraints are similar to ones described above. Constraints related to vertical spatial granularity are similar to those of temporal granularity.

4.3.2 Indeterminate Spatial Entity Class

Many applications need to capture indeterminacy associated with the location of an object. An indeterminate point is denoted by P~. We give the semantics of indeterminate point in this section.

An indeterminate point is composed of a region bounded by upper and lower along with pmf_xy that gives the probability that an object is located within the given partition in bounds as specified by upper and lower. Moreover, the upper and lower supports lie within the extent.

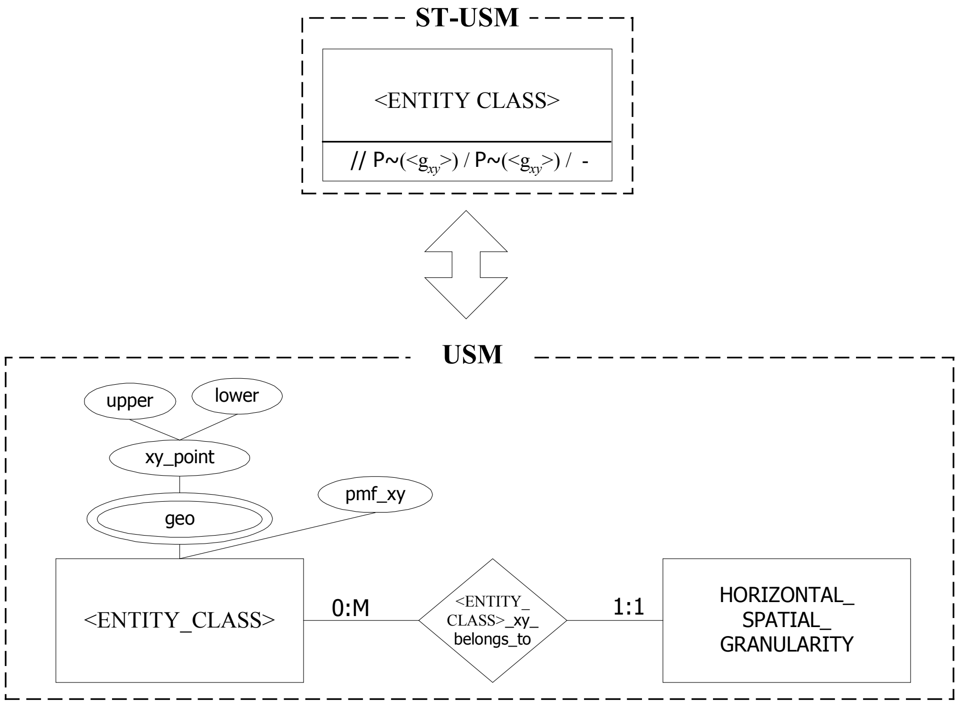


Figure 11: An indeterminate spatial entity class in ST-USM and its semantics in USM

Besides constraints 4.3.1, 4.3.4, 4.3.5 and 4.3.6, there is one additional constraint for an indeterminate spatial entity class.

Constraint 4.3.7: The indexes corresponding to the geometry of an indeterminate spatial object lie within the specified extent, xy_minimum and xy_maximum.

∀ *e* ∈ *S*(⟨ENTITY_CLASS⟩), ∀*g* ∈ *e*.geo,
e.⟨ENTITY_CLASS⟩_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY (extent.xy_minimum) ≤ *g*.xy_point.lower ≤ *g*.xy_point.upper ≤ *e*.⟨ENTITY_CLASS⟩_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY (extent.xy_maximum)

5. Case Study

We now apply our proposed approach for capturing granularity- and indeterminacy-related semantics to develop a schema for the case study described in section 2.

As shown in Figure 12, the data analyst first develops a schema using a conventional conceptual model based on requirements elicited from the users; in this example, she uses USM. Based on the user requirements described in section 2, this hydrogeologic application is primarily concerned with SPRING_SITE and BORE_HOLE_SITE. Often, a BORE_HOLE_SITE has a PUMPLIFT associated with it. There are multiple source agencies that measure DISCHARGE and WATER_LEVEL at SPRING_SITE and BORE_HOLE_SITE respectively. WATER_LEVEL and DISCHARGE are inputs used for the SIMULATED_OUTPUT. Some of the properties that the users are interested in are shown in ovals and associated with the appropriate entity class. At this stage, the data analyst is not concerned with spatial and temporal aspects associated with the facts. We refer to this schema as the *core USM Schema*.

Figure 12: The hydrogeologic schema using USM

For each construct in the core USM schema (e.g., Figure 12) the data analyst together with the users consider whether temporality and spatiality is important for the application. The data analyst next asks users questions like: Do you want to store the history or only the current value associated with this fact? Do you want to capture valid time or transaction time, or both? What is the associated temporal granularity? Is it important to store the geographical reference for the objects? What is the geographical shape of the objects? What is the associated spatial granularity? Is it important to capture indeterminacy? Is the probability associated with indeterminacy known? Is a uniform distribution assumption valid for the application? Accordingly, the database analyst now annotates the schema resulting in the *ST-USM Schema* (or the *Annotated Schema* in Figure 13). For example, SPRING_SITE is a spatial entity represented as an indeterminate point (P+-) in two-dimensional space where the location needs to be captured in dms-degree. Similarly, the data analyst annotates water_depth as temporal because the users are interested in the valid time associated with the measurement of the water_depth.

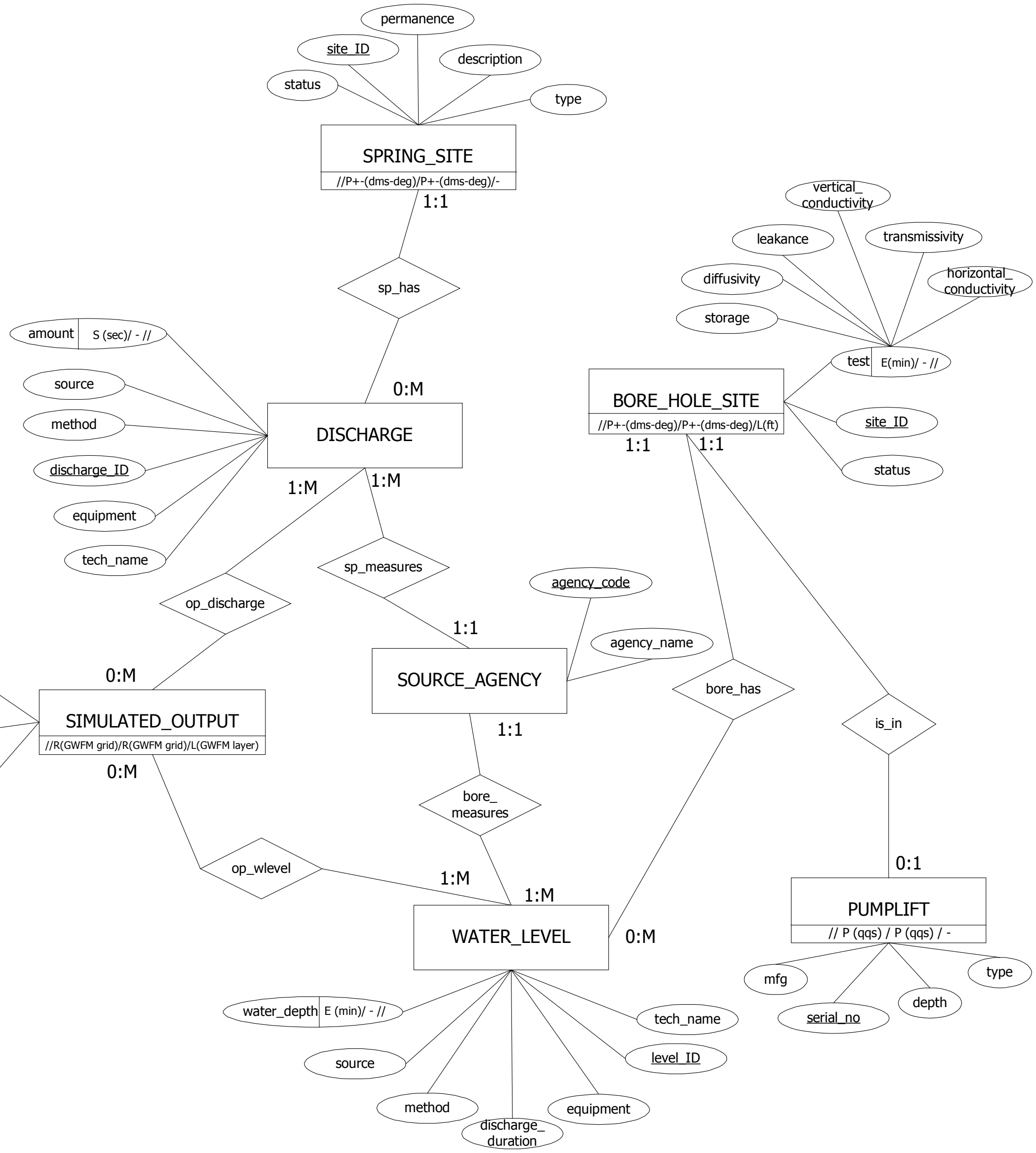


Figure 13: The ST-USM schema for the case study

The Annotated Schema described in Figure 13 can be used as a communication vehicle between the users and data analysts. As is evident, the ST-USM schema does not add clutter in the schema. It can also be used to decide if all the spatiotemporal requirements of the user have been captured and whether the requirements are conflicting. Figure 14 provides the semantics of this schema using conventional conceptual modeling constructs. This *Translated USM Schema*, including the constraints specified in the previous section, can be used by the data analyst to subsequently develop the logical schema.

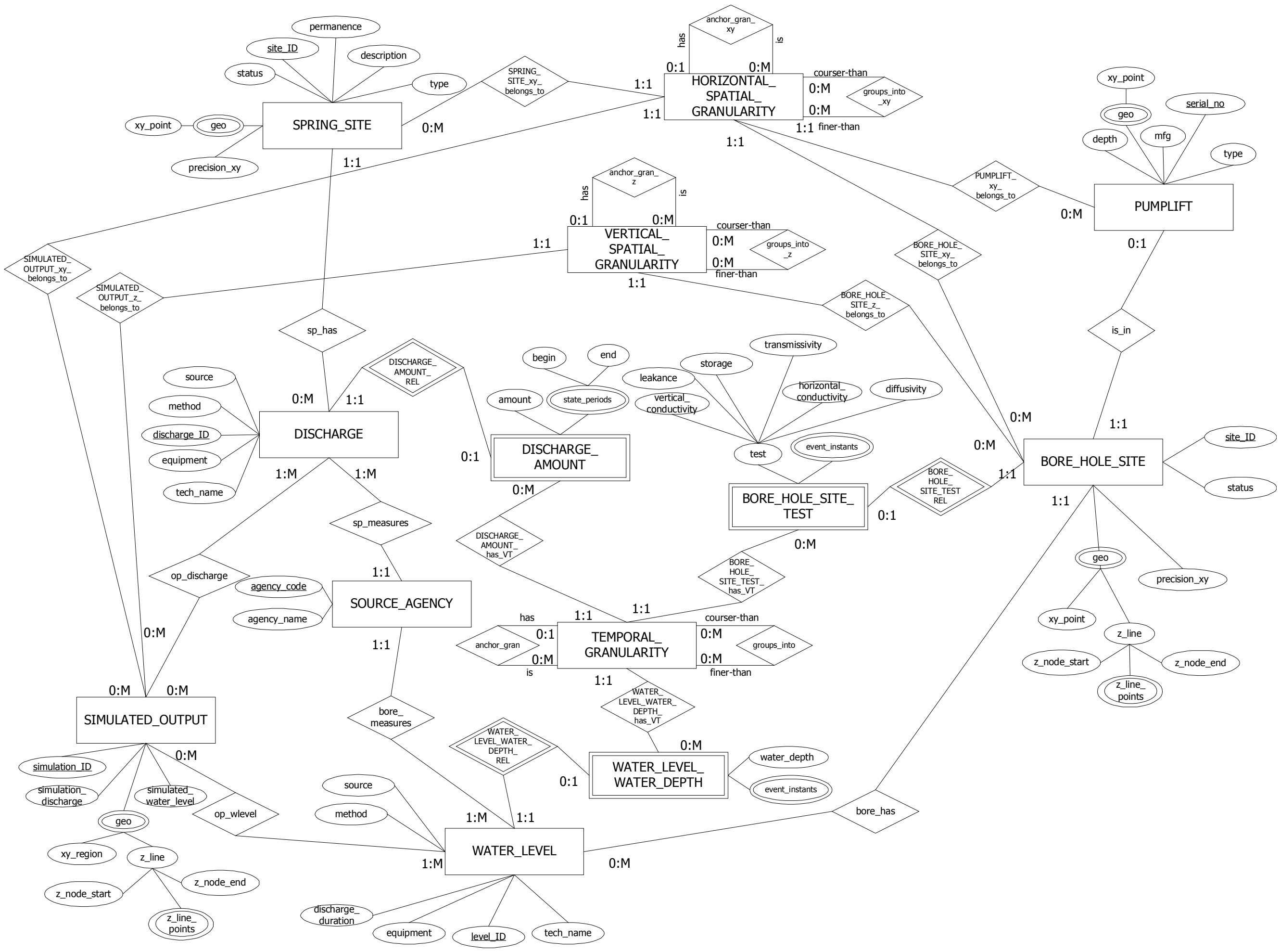


Figure 14: The semantics of the ST-USM schema using USM

We are developing a spatiotemporal database design environment called DISTIL (DIstributed design of SpaTIotemporaL data) [41] that takes the users' spatiotemporal requirements via pop-up boxes, where the data analyst can specify spatial and temporal aspects of the application. Based on these specifications, DISTIL automatically annotates the schema. DISTIL facilitates the development of a core USM, ST-USM and translated USM schema, thus, assisting in conceptual database design of spatiotemporal applications.

6. Conclusions

In this paper, we introduce annotations to capture the semantics related to granularities and indeterminacy in a spatiotemporal conceptual model. The case study related to a hydrogeologic application demonstrates that our approach is straightforward and comprehensive.

Burrough and Frank [9] classify geographic users based on procedure and purpose: *managers of defined objects* use GIS in areas like cadastral mapping and utility management and *planners and resource managers* deal with geographic entities whose behavior they try to control. Couclelis [12] extends the classification based on well boundedness of an empirical entity and its representation and the users' requirements of well-bounded entities. She further posits that managers of defined objects deal with well-bounded empirical objects, well-bounded representation and their manipulations require well-defined boundaries. On the other hand, planners and resource managers deal with objects that do not have and cannot be assigned well-defined boundaries, but need to control and manipulate geographic objects as some entities. We believe that ST-USM with its support for granularities and indeterminacy can be used by both of these classes of users to capture their spatial and temporal database requirements.

Our work has several practical implications related to capturing the spatiotemporal semantics during conceptual database design stage. From the users' point of view, the schema developed using ST-USM is easy to understand, intuitive and simple. Thus, the ST-USM schema can be effectively used as a communication tool during requirements analysis. From the database analysts' point of view, our annotation-based approach in ST-USM does not introduce any new spatial or temporal constructs. Thus, the data analysts do not need to know about any new constructs related to spatiality and temporality. Additionally, our approach does not change the semantics of existing conventional conceptual model. From a CASE tool vendors' point of view, our annotation-based approach is straightforward to incorporate into their existing design tools. The CASE tool could allow annotation specification via a pop-up box associated with any abstraction type.

In this paper, we have focused on describing the semantics related to granularities and indeterminacy in a conceptual model. We are developing a detailed framework [26] that captures the spatiotemporal semantics related to various types of abstractions, i.e., entity types (simple, generalization/ specialization, composite and grouping), relationships (interaction, class, composite and grouping), and attributes (simple, composite and multi-valued). We are working on details related to mapping our spatiotemporal conceptual model to a logical model accommodating granularities, e.g., [60]. The logical design will need intelligent mapping rules so that the semantics related to granularities are incorporated at the schema level. Incorporating the granularities semantics at the schema level is now possible in view of the opportunities presented by object-relational DBMS (e.g., Oracle 8i). In future, we envision extending spatial granularity for small-scale space applications [29], e.g., computer-aided design (CAD).

The annotation-based approach is applicable to a conceptual model, design tool or query language. The present paper is the first to our knowledge that explains how to support granularity and indeterminacy in a spatiotemporal conceptual model with annotations.

7. Acknowledgements

We thank two anonymous reviewers for their excellent feedback that resulted in many changes to the first draft of this paper. The first two authors were support in part by a NASA grant 314401. The third author was supported in part by an NSF grant EIA-0080123.

8. References

[1] G. Albaredes, "A New Approach: User Oriented GIS," in Proceedings of European Conference and Exhibition on Geographic Information Systems, EGIS '92, Munich, pp. 830-837, 1992.

[2] C. Batini, S. Ceri, and S. B. Navathe, *Conceptual Database Design: An Entity-Relationship Approach*: Benjamin/Cummings Publishing Company, 1992.

[3] M. Bertolotto, L. D. Floriani, and P. Marzano, "A Unifying Framework for Multilevel Description of Spatial Data," in Spatial Information Theory: A Theoretical Basis for GIS, International Conference (COSIT 1995), Semmering, Austria, pp. 1995.

[4] C. Bettini, C. E. Dyreson, W. S. Evans, R. T. Snodgrass, and X. S. Wang, "A Glossary of Time Granularity Concepts," in *Temporal Databases: Research and Practice*, O. Etzion, S. Jajodia, and S. Sripada, Eds.: Springer-Verlag, 1998, pp. 406-413.

[5] C. Bettini, S. Jajodia, and S. X. Wang, *Time Granularities in Databases, Data Mining and Temporal Reasoning*. Berlin: Springer-Verlag, 2000.

[6] C. Bettini and R. D. Sibi, "Symbolic Representation of User-Defined Time Granularities," in 6th International Workshop on Temporal Representation and Reasoning (TIME 1999), Orlando, Florida, pp. 17-28, 1999.

[7] C. Bettini, X. S. Wang, and S. Jajodia, "Mining Temporal Relationships with Multiple Granularities in Time Sequences," *Data Engineering Bulletin*, Vol. 21, No. 1, pp. 32-38, 1998.

[8] M. Boman, J. A. J. Bubenko, P. Johannesson, and B. Wangler, *Conceptual Modelling*: Prentice Hall, 1997.

[9] P. A. Burrough and A. U. Frank, "Concepts and Paradigms in Spatial Information: Are Current Geographical Information Systems Truly Generic?," *International Journal of Geographic Information Systems*, Vol. 9, No. 2, pp. 101-116, 1995.

[10] P. P. Chen, “The Entity-Relationship Model - Toward a Unified View of Data,” *ACM Transactions of Database Systems*, Vol. 1, No. 1, pp. 9-36, 1976.

[11] C. Combi, “Representing Absolute Time Expressions with Vagueness, Indeterminacy, and Different Granularities,” in Workshop on Spatial and Temporal Granularity, Austin, Texas, pp. 17-24, 2000.

[12] H. Couclelis, “Towards an Operation Typology of Geographic Entities with Ill-defined Boundaries,” in *Geographic Objects with Indeterminate Boundaries*, P. A. Burrough and A. U. Frank, Eds. Bristol, PA: Taylor and Francis, Inc., 1996, pp. 45-55.

[13] F. A. D'Agnese, C. C. Faunt, A. K. Turner, and M. C. Hill, “Hydrogeologic evaluation and numerical simulation of the Death Valley Regional ground-water flow system, Nevada and California,” U.S. Geological Survey Water Resources 96-4300, 1997.

[14] B. David, M. V. D. Herrewegen, and F. Salge, “Conceptual Models for Geometry and Quality of Geographic Information,” in *Geographic Objects With Indeterminate Boundaries*, P. A. Burrough and A. Frank, Eds.: Taylor & Francis, 1996, pp. 352.

[15] C. E. Dyreson, W. S. Evans, H. Lin, and R. T. Snodgrass, “Efficiently Supporting Temporal Granularities,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, No. 4, pp. 568-587, 2000.

[16] C. E. Dyreson and R. T. Snodgrass, “Supporting Valid-Time Indeterminacy,” *ACM Transactions on Database Systems*, Vol. 23, No. 1, pp. 1-57, 1998.

[17] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, Second ed. Redwood City, CA: Benjamin/ Cummings Publishing Co., 1994.

[18] L. D. Floriani, E. Puppo, and P. Magillo, “A Formal Approach to Multiresolution Modeling,” in *Geometric Modeling: Theory and Practice*, W. Strasser, R. Klein, and R. Rau, Eds.: Springer-Verlag, 1997, pp. 302-323.

[19] H. Gregersen and C. Jensen, “Conceptual Modeling of Time-Varying Information,” TIMECENTER Technical Report TR-35, September 10 1998.

[20] L. H. Hermosilla, “A Unified Approach for Developing a Temporal GIS with Database and Reasoning Capabilities,” in Proceedings of the Fifth European Conference and Exhibition on Geographic Information Systems, EGIS '94, pp. 122-131, 1994.

[21] J. R. Herring, “The Mathematical Modeling of Spatial and Non-Spatial Information in Geographic Information Systems,” in Proceedings of the NATO Advanced Study Institute on Cognitive and Linguistic Aspects of Geographic Space, Las Navas del Marques, Spain, pp. 1990.

[22] R. Hull and R. King, “Semantic Database Modeling: Survey, Applications, and Research Issues,” *ACM Computing Surveys*, 210-260, 1987.

[23] A. F. Hutchings and S. T. Knox, “Creating Products -- Customers Demand,” *Communications of the ACM*, Vol. 38, No. 5, pp. 72-80, 1995.

[24] C. S. Jensen, C. E. Dyreson, M. Bohlen, J. Clifford, R. Elmasri, S. K. Gadia, F. Grandi, P. Hayes, S. Jajodia, W. Kafer, N. Kline, N. Lorentzos, Y. Mitsopoulos, A. Montanari, D. Nonen, E. Peresi, B. Pernici, J. F. Roddick, N. L. Sarda, M. R. Scalas, A. Segev, R. T. Snodgrass, M. D. Soo, A. Tansel, R. Tiberio, and G. Wiederhold, “A Consensus Glossary of Temporal Database Concepts-February 1998 Version,” in *Temporal Databases: Research and Practice*, O. Etzion, S. Jajodia, and S. Sripada, Eds.: Springer-Verlag, 1998.

[25] C. S. Jensen and R. T. Snodgrass, “Semantics of Time-Varying Information,” *Information Systems*, Vol. 21, No. 4, pp. 311-352, 1996.

[26] V. Khatri, S. Ram, and R. T. Snodgrass, “ST-USM: Bridging the Semantic Gap with a Spatio-temporal Conceptual Model,” TimeCenter Technical Report TR-64, 2001.

[27] R. Kimball, *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*: John Wiley & Sons, 1996.

[28] L. Kleinrock, *Queueing SystemsTheory*. New York: John Wiley & Sons, 1975.

[29] B. Kuipers, “Modeling Spatial Knowledge,” *Cognitive Science*, Vol. 2, 129-153, 1978.

[30] R. Laurini and D. Thompson, *Fundamentals of Spatial Information Systems*. London: Academic Press, 1992.

[31] J. Lipski, “On Semantic Issues Connected with Incomplete Information Databases,” *ACM Transactions of Database Systems*, Vol. 4, No. 3, pp. 262-296, 1979.

[32] D. M. Mark and A. U. Frank, "Experiential and Formal Models of Geographic Space," *Environment and Planning*, Vol. 23, No. 1, pp. 3-24, 1996.
[33] C. B. Medeiros, M.-J. Bellosta, and G. Jomier, "Managing Multiple Representations of Georeferenced Elements," in Seventh International Workshop on Database and Expert Systems Applications, DEXA 1996, Zurich, Switzerland, pp. 364-370, 1996.
[34] D. Montello, "Scale and Multiple Psychologies of Space," in *Spatial Information Theory: A Theoretical Basis for GIS*, vol. 716, *Lecture Notes in Computer Science*, A. Frank and I. Campari, Eds. New York: Springer-Verlag, 1993, pp. 312-321.
[35] J. Nunes, "Geographic Space as a Set of Concrete Geographical Entities," in Proceedings of the NATO Advanced Study Institute on Cognitive and Linguistic Aspects of Geographic Space, Las Navas del Marques, Spain, pp. 1990.
[36] Oracle, "Coordinate Systems User's Guide," Oracle Inc., Release 8.1.6 April 28 2000.
[37] J. Peckham and F. Maryanski, "Semantic Data Models," *ACM Computing Surveys*, Vol. 20, No. 3, pp. 153-189, 1988.
[38] POSC, "Epicentre Usage Guide," POSC, http://www.posc.org/Epicentre.2_2/DataModel/ExamplesofUsage/eu_cs.html, 1998.
[39] E. Puppo and G. Dettori, "Towards a Formal Model for Multiresolution Spatial Maps," in Fourth International Symposium on Advances in Spatial Databases (SSD 1995), Portland, ME, pp. 154-169, 1995.
[40] S. Ram, "Intelligent Database Design using the Unifying Semantic Model," *Information and Management*, Vol. 29, No. 4, pp. 191-206, 1995.
[41] S. Ram, R. T. Snodgrass, V. Khatri, and Y. Hwang, "DISTIL: A Design Support Environment for Conceptual Modeling of Spatio-Temporal Requirements," in Proceedings of the 20th International Conference on Conceptual Modeling (ER2001), Yokohama, Japan, pp. 70-83, 2001.
[42] C. V. Ramamoorthy, A. Prakash, W. Tsai, and Y. Usuda, "Software Engineering: Problems and Perspectives," *Computer*, Vol. 17, No. 10, pp. 191-209, 1984.
[43] P. Rigaux and M. Scholl, "Multiple Representation Modelling and Querying," in International Workshop on Geographic Information Systems (IGIS 1994), Ascona, Switzerland, pp. 59-69, 1994.
[44] J. F. Roddick, "A Survey of Schema Versioning Issues for Database Systems," *Information and Software Technology*, Vol. 37, No. 7, pp. 383-393, 1995.
[45] J. F. Roddick and R. T. Snodgrass, "Schema Versioning," in *The TSQL2 Temporal Query Language*, R. T. Snodgrass, Ed. Boston: Kluwer Academic Publishers, 1995, pp. 427-449.
[46] R. D. Sack, *Conceptions of Space in Social Thought: A Geographic Perspective*. Minneapolis: University of Minnesota Press, 1980.
[47] A. Silbershatz, H. Korth, and S. Sudarshan, *Database System Concepts*, Third Edition ed: WCB/ McGraw Hill, 1997.
[48] D. F. Sinton, "The Inherent Structure of Information as a Constraint to Analysis: Mapped Thematic Data as a Case Study," *Harvard Papers on GIS*, Vol. 7, 1-17, 1978.
[49] R. T. Snodgrass, *Developing Time-Oriented Database Applications in SQL*. San Francisco: Morgan Kaufmann Series in Data Management Systems, 1999.
[50] R. T. Snodgrass and I. Ahn, "Temporal Databases," *IEEE Computer*, Vol. 19, No. 9, pp. 35-42, 1986.
[51] J. P. Snyder, *Map Projections used by the US Geological Survey*, Second ed, 1982.
[52] S. Spaccapietra, C. Parent, and E. Zimányi, "Modeling Time from a Conceptual Perspective," in Proceedings of the 1998 ACM CIKM International Conference on Information and Knowledge Management, Bethesda, Maryland, pp. 432-440, 1998.
[53] S. A. Starks, V. Kreinovich, and A. Meystel, "Multi-resolution data processing: it is necessary, it is possible, it is fundamental," in Proceedings of the International Conference on Intelligent Systems and Semiotics (ISAS'97), Gaithersburg, MD, pp. 145-150, 1997.
[54] J. G. Stell, "The Representation of Discrete Multi-Resolution Spatial Knowledge," in Proceedings of Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, pp. 38-49, 2000.

[55] J. G. Stell and M. F. Worboys, "Stratified Map Spaces: A Formal Basis for Multi-resolution Spatial Databases," in Proceedings 8th International Symposium on Spatial Data Handling (SDH 1998), pp. 180-189, 1998.

[56] J. G. Stell and M. F. Worboys, "Generalizing graphs using amalgamation and selection," in Sixth International Symposium on Advances in Spatial Databases (SSD '99), pp. 19-32, 1999.

[57] V. J. Tsotras, C. S. Jensen, and R. T. Snodgrass, "An Extensible Notation for Spatiotemporal Index Queries," *SIGMOD Record*, Vol. 27, No. 1, pp. 47-53, 1998.

[58] M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications," *Knowledge Engineering Review*, Vol. 11, No. 2, pp. 93-136, 1996.

[59] J. W. van Roessel, "Design of a Spatial Data Structure using the Relational Normal Form," *International Journal of Geographical Information Systems*, Vol. 1, No. 1, pp. 33-50, 1987.

[60] X. S. Wang, C. Bettini, A. Brodsky, and S. Jajodia, "Logical Design for Temporal Databases with Multiple Granularities," *ACM Transactions on Database Systems*, Vol. 22, No. 2, pp. 115-170, 1997.

[61] X. S. Wang, S. Jajodia, and V. S. Subrahmanian, "Temporal Modules: An Approach Toward Federated Temporal Databases," *Information Sciences*, Vol. 82, No. 1-2, pp. 103-128, 1995.

[62] M. F. Worboys, "Computation with Imprecise Geospatial Data," *Computer, Environment and Urban Systems*, Vol. 22, No. 2, pp. 85-106, 1998.

[63] M. F. Worboys, "Imprecision in Finite Resolution Spatial Data," *GeoInformatica*, Vol. 2, No. 3, pp. 257-279, 1998.

[64] C. Zaniolo, S. Ceri, C. Faloutsos, R. T. Snodgrass, R. Zicari, and V. S. Subrahmanian, *Advanced Database Systems*: Morgan Kaufmann Publishers, 1997.

Appendix: Annotation Syntax

⟨annotation⟩	::=	ϵ | ⟨temporal annotation⟩ **//** ⟨spatial annotation⟩
⟨temporal annotation⟩	::=	ϵ | ⟨valid time⟩ / ⟨transaction time⟩
⟨valid time⟩	::=	⟨state⟩ (⟨g_t⟩) | ⟨indeterminate state⟩ (⟨g_t⟩) | ⟨event⟩ (⟨g_t⟩) | ⟨indeterminate event⟩(⟨g_t⟩) | -
⟨transaction time⟩	::=	T | -
⟨state⟩	::=	S | State
⟨indeterminate state⟩	::=	⟨state⟩~ | ⟨state⟩+-
⟨event⟩	::=	E | Event
⟨indeterminate event⟩	::=	⟨event⟩~ | ⟨event⟩+-
⟨spatial annotation⟩	::=	ϵ | ⟨horizontal geometry⟩ / ⟨vertical geometry⟩
⟨horizontal geometry⟩	::=	⟨geometry⟩ (⟨g_{xy}⟩) / ⟨geometry⟩ (⟨g_{xy}⟩)
⟨vertical geometry⟩	::=	⟨geometry⟩ (⟨g_z⟩) | -
⟨geometry⟩	::=	⟨point⟩ | ⟨indeterminate point⟩ | ⟨line⟩ | ⟨indeterminate line⟩ | ⟨region⟩ | ⟨indeterminate region⟩ | -
⟨point⟩	::=	P | Point
⟨indeterminate point⟩	::=	⟨point⟩~ | ⟨point⟩+-
⟨line⟩	::=	L | Line
⟨indeterminate line⟩	::=	⟨line⟩~ | ⟨line⟩+-
⟨region⟩	::=	R | Region
⟨indeterminate region⟩	::=	⟨region⟩~ | ⟨region⟩+-
⟨g_t⟩	::=	⟨day⟩ | ⟨hour⟩ | ⟨minute⟩ | ⟨second⟩ | ⟨user defined⟩
⟨day⟩	::=	day
⟨hour⟩	::=	hr | hour
⟨minute⟩	::=	min | minute
⟨second⟩	::=	sec | second
⟨g_{xy}⟩	::=	⟨mile⟩ | ⟨dms-degree⟩ | ⟨dms-minute⟩ | ⟨foot⟩ | ⟨user defined⟩
⟨g_z⟩	::=	⟨mile⟩ | ⟨foot⟩ | ⟨user defined⟩
⟨mile⟩	::=	mile
⟨dms-degree⟩	::=	dms-deg | dms-degree
⟨dms-minute⟩	::=	dms-min | dms-minute
⟨foot⟩	::=	ft | foot