On Augmenting Database Design-Support Environments to Capture the Geo-Spatio-Temporal Data Semantics

Vijay Khatri*, Sudha Ram** and Richard T. Snodgrass**
*Indiana University, Bloomington IN 47405; **University of Arizona, Tucson AZ 85721

Abstract: A database design-support environment supports a data analyst in eliciting, articulating, specifying and validating data-related requirements. Extant design-support environments—based on conventional conceptual models—do not adequately support applications that need to organize data based on time (e.g., accounting, portfolio management, personnel management) and/or space (e.g., facility management, transportation, logistics). For geo-spatio-temporal applications, it is left to database designers to discover, design and implement—on an ad-hoc basis—the temporal and geospatial concepts that they need to represent the miniworld. To elicit the geo-spatio-temporal data semantics, we characterize guiding principles for augmenting the conventional conceptual database design approach, present our annotation-based approach, and illustrate how our proposed approach can be instantiated via a proof-of-concept prototype. Via a proof-of-concept database design-support environment, we exemplify our annotation-based approach, and show how segregating "what" from "when/where" via annotations satisfies ontologic- and cognition-based requirements, dovetails with existing database design methodologies, results in upward-compatible conceptual as well as XML schemas, and provides a straightforward mechanism to extend extant design-support environments.

Keywords: Semantic Model, Data Semantics, Database Design, Geo-Spatio-Temporal Database, CASE Tool

1. Introduction

A design support environment is a combination of a software tool *and* structured development methodology; the former automates the software process, while the latter defines the process to be automated [72]. Prior research shows that design-support environments help reduce time and money spent on a project, and can help improve the quality of the end-product [32] and the effectiveness of software development [18]. However, extant database design-support environments (e.g., ERWin[1], ERStudio[2])—based on conventional conceptual models—do not adequately support applications that need to organize data based on time (e.g., accounting, portfolio management, personnel management) and/or space (e.g., facility management, transportation, logistics). *Conventional conceptual models*, e.g., [21, 30], provide a mechanism to elicit data semantics related to "what" is important for an application rather than the "when" and "where" semantics. Underlying the temporal and geospatial (or geographic) applications outlined above are temporal and geospatial data, collectively referred to as *geo-spatio-temporal data*. For geo-spatio-temporal applications, it is left to the database designers to discover, design and implement—on an ad-hoc basis—the temporal and geospatial concepts that they need to represent the "miniworld," or an aspect of the real world [30]. To augment a database design-support environment that would help elicit the data semantics related to space and/or time—at an abstract-level independent of implementation—we

[1] http://www3.ca.com/Solutions/Product.asp?ID=260
[2] http://www.embarcadero.com/products/erstudio/

characterize guiding principles, propose a geo-spatio-temporal conceptual design approach, and illustrate how the proposed approach can be instantiated via a proof-of-concept prototype.

Via a proof-of-concept design-support environment, we exemplify our geo-spatio-temporal conceptual design approach that advocates: (i) first eliciting requirements using a conventional conceptual model *without considering geospatial or temporal aspects* (i.e., "what"); and only then (ii) annotating a conventional schema to capture the geo-spatio-temporal requirements (i.e., "when/where"). An abstraction provides a mechanism to focus on selective details while deliberately omitting others [107]. Our proposed approach, based on orthogonality of "what" from "when/where," enables a supplementary level of abstraction via annotations. This approach is embodied in a proof-of-concept prototype, a DesIgn-Support environment for geo-SpaTIotemporaL data (DISTIL). Via DISTIL, we illustrate how guiding principles associated with geo-spatio-temporal conceptual modeling can be the basis for the development of a geo-spatio-temporal design-support environment. While our annotation-based approach is consonant with prior research [36, 85, 120, 141], we explicate how annotations along with syntactic orthogonality (of "what" from "when/where") can be the basis for: (i) enabling separation of concerns (i.e., "what" from "when/where") for data analysts; (ii) defining an upward compatible geo-spatio-temporal design approach; (iii) development of upward compatible geo-spatio-temporal schemas (both conceptual and XML); and (iv) augmenting extant design-support environments in a straightforward manner. Prior research posits that requirements specification is difficult because of human problem-solving limitations [26] and that problem solvers can effectively handle 7±2 chunks of information [76]. By defining a methodology that segregates "what" from "when/where," we provide an approach that will help ameliorate information overload during elicitation and validation of data-related requirements.

The research reported in this paper integrates and extends previous work by the authors. Previously [56, 57], we have abstractly shown the semantics of geo-spatio-temporal annotation grammar. A preliminary version of DISTIL was presented at a conference [95]. In this paper, we discuss guiding principles underlying an annotation-based approach, and illustrate how the proposed approach can be the basis for augmenting an existing database design-support environment; we, thus, demonstrate practical implications of our annotation-based approach. We show how segregating "what" from "when/where" via annotations satisfies cognition-based requirements, dovetails with existing database design methodologies, results in upward-compatible schemas, and provides a straightforward mechanism to extend an extant design-support environment. Via DISTIL, we illustrate how orthogonality (i.e., segregation of "what" from "when/where") and upward compatibility *consistently* applies to the proposed approach *and* design-support environment implementation *and* geo-spatio-temporal schemas, both conceptual and XML.

We outline the assumptions in this paper to delineate the scope of this work. (i) While *upper-CASE tools* focus on analysis and design, *lower-CASE tools* aid application development [73]. In this paper, we primarily focus on database design-support environment that enables analysis and design of geo-spatio-temporal applications. (ii) Previously [110], we have differentiated between *sequenced* and *non-sequenced* data semantics. While temporal (or geospatial) sequenced data semantics refer to each point of time (or space), non-sequenced data semantics do not treat time (or space) specially or reference all points of time (or space). In this paper, we focus on sequenced data semantics. (iii) A database schema can evolve with time, and *schema versioning* [99, 100] is an important area of research; however, schema versioning is not the focus of this paper. (iv) Based on perception, space may be differentiated into *large-scale* and *small-scale* space [64]. 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 [70], 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. In summary, the focus of this paper is to explicate augmentation criteria, propose an overall approach for augmenting extant design-support environments with geo-spatio-temporal data semantics, and illustrate how the proposed approach can be instantiated into a proof-of-concept prototype.

The rest of the paper is organized as follows. We first motivate the problem and present underlying guidelines for incorporating geo-spatio-temporal data semantics into a conventional conceptual model, and, subsequently, present our overall methodology in Section 2. According to Wand et al. [127], the power of a modeling language is driven by the semantics of its constructs, and that ontology can be the basis for defining concepts in a modeling language. Using a study at the United States Geological Survey (USGS), we provide a motivating example that highlights salient ontological concepts underlying geo-spatio-temporal applications. The basis for annotations (and the dialog panels of the proof-of-concept design-support environment) is time and space ontology, which is summarized in Section 3. In Section 4, we employ a data analyst's interaction with DISTIL (using the USGS example of Section 3) to present our proposed approach, and to illustrate how our proposed approach can be embedded in a proof-of-concept prototype. Via implementation architecture of the proof-of-concept design-support prototype (in Section 5), we show how the segregation of "what" from "when/where" is consistently maintained in the development of the design-support environment. We round out the paper with evaluation (Section 6), an overview of related research (Section 7), and a summary along with future directions of this research (Section 8).

2. Guidelines for Inducing the Geo-Spatio-Temporal Data Semantics

A systems development methodology, a systematic procedure for developing a system or a part of a system [47], provides a strategy for subdivision of the development process [79]. One of the important components underlying over a thousand systems development methodologies [48] is graphical representation – some [8] posit that a good representation constitutes half the solution. Hahn and Kim [43] argue how there is a dearth of guidelines for designing and upgrading new versions of representations. In the following, we motivate the problem, and then present guidelines for geo-spatio-temporal conceptual modeling from two perspectives: users (cognition and representation) and (extant) design methodologies. We conclude this section with a description of our overall geo-spatio-temporal conceptual design methodology.

2.1 Motivation

Lately, geographic information is increasingly employed in a wide array of applications including social, environmental and economic studies [126], e.g., land information systems, environmental modeling, resource management, transportation planning, geo-marketing, geology and archaeology [67]. For example, retailers like Ace Hardware Corporation have used geographic information to identify underserved customers, and make decisions related to store relocation based on where their customers come from [44]. From a practitioner's perspective, recent advances in technologies like high-resolution satellite-borne imaging systems, mobile systems, global positioning systems, and the overall decrease in hardware costs is resulting in temporal and geospatial data finding their way into many traditional applications. As a result, research interest in geo-spatio-temporal data management has increased dramatically over the past decade [5, 6, 29, 41, 59, 63, 74, 115, 117, 123, 135].

Accurate formal approach to elicitation, articulation, specification and validation of information requirements is critical to the development of an organization's information systems [129] including the geo-spatio-temporal applications described above. Conceptual models provide a formalism to develop precise and unambiguous representation (referred to as a *conceptual schema*) of the real world, which serve many divergent purposes [66]: (i) facilitate communication between users and developers; (ii) help analysts understand the domain; (iii) provide input for design and subsequent development; and (iv) document requirements for archival. Similarly, Bédard [13] argues that conceptual models are a combination of thinking, communication, development as well as documentation tools. Prior studies [45, 97] attribute project failure to lack of identifying real needs during conceptual design. Consequently, conceptual modeling for geo-spatio-temporal applications [39, 40, 42, 57, 61, 83-85, 104, 118, 120-122, 131, 134] has developed into an important area of research.

While geo-spatio-temporal conceptual models cited above provide a formalism, a design tool that embeds an approach (and a formalism) can help support elicitation of data-related requirements. According to a Standish Report [1], tools for requirements analysis have "the biggest impact on the success of projects." Computer-Aided Software/System Engineering (CASE) tools, which subsume requirements analysis tools, are defined as the automation of part of, or the entire, system development process [119]. Various prior studies [10, 31, 46, 77, 82] have concluded that CASE tools improve productivity and quality of the design and development process. CASE tools are now commonly used in systems development – a Software Engineering Institute survey [88] of "high" maturity organizations found that 25 out of 35 firms used CASE tools as standard practice or common practice. Another recent survey [103] found that the adoption and infusion of these CASE tools is primarily for analysis as well as representation of objects, relationships and processes. In this paper, we focus on tools—referred to as design-support environments—that embed a geo-spatio-temporal conceptual design approach. We highlight below the issues that need to be addressed for augmenting extant design-support environments to support elicitation of the geo-spatio-temporal data semantics.

A recent survey [1] found that one of the "recipe" for project success is the use of formal methodology: compared with 30% of challenged/failed projects, 46% of successful projects were found to use formal methodologies. A methodology needs to "fit" the activities it is intended to support [71], and a design-support environment helps enforce a defined methodology [125]. With geographic data finding their way into traditional applications (e.g., insurance, retail and distribution), there is a need for an overall geo-spatio-temporal conceptual database design methodology that can be integrated with conventional conceptual design approach, e.g., [21, 30]. It would be expedient to embed an approach in the geo-spatio-temporal design-support environment that is compatible with existing general-purpose conceptual design approach. Thus, the requirements of geo-spatio-temporal applications need to be met without resorting to a fundamentally different approach.

Juhn and Naumann [53] posit that representations like conceptual schemas "drive discovery" and should be precisely and rigorously defined; on the other hand, discovery needs to be "validated" implying that the schemas should be clear and comprehensible. A well-designed methodology (and interface to a design-support environment) should create a metaphor that bridges the conceptual gap between human thinking and computer system [68]. The challenge in adding space and time dimensions is balancing simplicity and understandability with preciseness and completeness. Thus, the proposed support for eliciting the geo-spatio-temporal data semantics in a database design-support environment should be—from data analysts' point of view—cognitively straightforward to use.

A recent survey by CIO Insight [2] found that—of the surveyed organizations—45.4% of the a company's systems were referred to as legacy systems and that a large number of organizations (44.2%) use legacy systems for data management. Another survey by WRQ [3] found that 86% of UK companies have critical business data residing on legacy systems. It seems that legacy systems are an integral part of the overall information systems – organizations consider their information systems assets as investments that grow with time rather than liabilities whose value depreciates over time [130]. Böhlen et al. [16] present requirements for protecting investments in extant geo-spatio-temporal legacy systems. Upward compatibility [17, 114] refers to the ability to render conventional conceptual schemas geo-spatio-temporal without affecting the legacy schemas. Upward compatibility is not only a requirement for conceptual schemas but also for XML schemas[3]. With the growing popularity of XML (eXtensible Markup Language) [4], XML schemas are envisioned to be used as the "external schemas" for data [69]. Conversion of conceptual schemas to XML schemas [69, 89, 136] is provided by extant database design-support environments like ERWin [34]. In the context of geo-spatio-temporal conceptual modeling, conversion of the conceptual schemas to XML schemas and the need for interoperability with legacy (existing) XML schemas, we need to ensure that the geo-spatio-temporal aspects that are rendered on the XML schema are upward compatible, i.e., they do not invalidate the legacy XML schemas.

In summary, the augmentation of extant design-support environment to support elicitation of the geo-spatio-temporal data semantics needs to address many issues: What are the underlying guiding principles for augmenting extant design-support environments? To dove-tail with extant conventional conceptual models, how should geo-spatio-temporal conceptual modeling approach be embedded in a design-support environment? From a user's (e.g., data analyst) perspective, how should geo-spatio-temporal aspects be elicited via the design-support environment so that it does not lead to cognitive overload? How should geo-spatio-temporal aspects be rendered on schemas (both conceptual and XML) so that legacy and geo-spatio-temporal schemas can co-exist?

2.2 Cognition and Representation

Kulpa [65] characterizes *propositional representation* as one where the representation *describes* the thing represented. Conceptual schemas are examples of visual propositional representation, and the effectiveness of visual languages—e.g., conceptual models—is "determined by how easy it is to state the facts in the language and how easy it is to perceive the facts from representation" [65]. Most work in cognition assumes that the mind has mental representations analogous to computer data structures – the former is referred to as *internal representation* and the latter as *external representation* [138, 139]. Zhang and Norman [140] propose distributed cognition that asserts that tasks require processing of information distributed across internal minds and external representation. Internal representations refer to knowledge structures in people's minds while external representations are knowledge structures in the environment,

[3] Like Mani et al [69], we refer to *XML schema* as a general term for a schema in XML as opposed to *XML-Schema* that refers to a kind of schema language proposed by W3C.

e.g., written symbols like conceptual schemas. In addition to providing inputs and stimulus to the internal mind (via extending working memory, forming permanent archives and allowing memory to be shared), external representations are intrinsic to problem-solving as they determine cognitive behavior itself. The effectiveness of external representation depends on how well it supports cognitive reasoning [43]. Prior research [20, 33, 75] posits that conceptual models for geographic data representation—i.e., formalisms for developing external representation—do not explicitly incorporate how humans cognitively store and use geographic knowledge. Thus, it is imperative to understand how the structure of knowledge—derived from linguistic/perceptual inputs—is stored in our long-term memory as an internal representation.

2.2.1 Structure of Knowledge

Some researchers [8, 101] posit that all human knowledge is stored as abstract conceptual propositions. As shown in Figure 1, Anderson and Bower's [8] Human Associative Model (HAM), based on propositions, is used to represent information in the long-term memory. A *proposition*—an assertion about the real world that provides abstract representation of both verbal and visual information [93]—is composed of a *fact* and *context* associated with the fact. The *subject* and *predicate* correspond with a topic, and a comment about the topic; this corresponds with information representation as object-property or property-value pairs. For some applications, the context in which the fact is true can be the key to reasoning about the mini-world. This context in turn is composed of *time* and *location* associated with the fact.

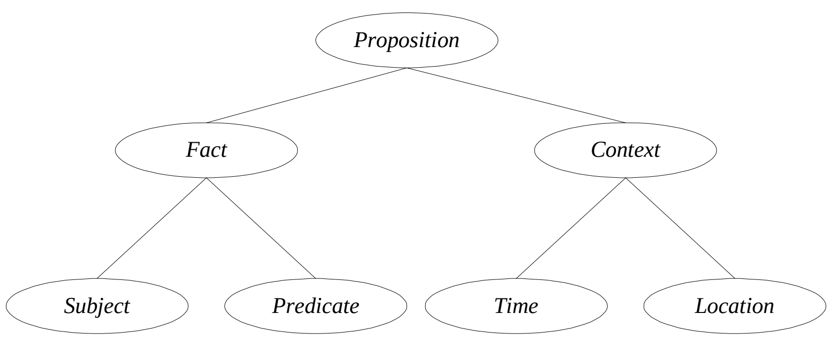

Figure 1: Adapted from Human Associative Memory Model [8]

The context element is orthogonal to the fact element, and specifies the geo-spatio-temporal reality for which the fact is true. Other researchers [60] similarly argue that what, when and where are distinct, and encoded separately in our brain.

Within the "context" element, we need to make an assumption whether there is symmetry in the treatment of space and time – for that we turn to linguistics.

2.2.2 Space vs. Time

While time and space are the basis for all our experiences, cognition and coordinated collective actions, there seems to be asymmetry in the treatment by languages: "whereas the speaker is free to talk about space or not, this is not so for time; each finite verb obligatorily includes temporal information—it expresses tense, aspect, or both…the expression of time is a consequence of the way in which languages is structured" [58]. Like Klein [58], we take objects, their properties and relationships to be predominantly embedded in time; one may or may not choose to capture the associated temporality in an application. As with Abraham and Roddick [2], we construe the time-varying geospatial aspect as a lineage of the geospatial aspects – this is consistent with other geo-spatio-temporal conceptual models.

2.3 Criteria for Augmenting Conventional Conceptual Models

Böhlen et al. [16] propose requirements to ensure that legacy DBMS application code (along with the data) remain operational when migrated to geo-spatio-temporal DBMS. While the requirements—upward compatibility and snapshot reducibility—described in their paper refer to logical data model and query languages, the requirements are equally applicable to conceptual modeling.

Upward compatibility [17, 114] implies the ability to render conventional conceptual schemas geo-spatio-temporal without affecting the legacy schemas. The objective of upward compatibility is to be able to develop geo-spatio-temporal schemas without invalidating the extant legacy schemas, thus, helping protect investments in existing schemas. It also implies that both the legacy schemas and the geo-spatio-temporal schemas can co-exist. Upward compatibility requires that the syntax and semantics of the traditional conceptual model, e.g., [21, 30], remain unaltered. If the geo-spatio-temporal extension is a strict superset provided by adding non-mandatory semantics, it would ensure that the geo-spatio-temporal extension is upward compatible with conventional conceptual models.

Snapshot reducibility [17, 108] refers to "natural" generalization of the semantics of extant conventional conceptual models, e.g., [21, 30], for incorporating the geo-spatio-temporal extension. Snapshot reducibility ensures that the semantics of a geo-spatio-temporal model are understandable *in terms of* the semantics of the conventional conceptual model. Here, the overall objective is to help ensure minimum additional investment in training (or retraining) a data analyst.

2.4 Geo-Spatio-Temporal Design Approach

The requirements described in the previous sections induce a three-layered architecture (Figure 2) that incorporates the space and time semantics in two stages (Layer 2 and Layer 3). Layer 1 provides a mechanism for eliciting the semantics of "what" is important for an application; conventional conceptual models help realize this layer. Unifying Semantic Model (USM) [94]—a conventional conceptual model that helps elicit the semantics related to entities, relationships and attributes—instantiates layer 1. Temporal and geospatial support for sequenced semantics (both temporal and geospatial) provided by layer 2 does not add any new syntax (i.e., constructs), and is based on layer 1. Annotations encode geo-spatio-temporal data semantics, and are used to instantiate layer 2. Upward compatibility and snapshot reducibility are applicable to an annotated schema. Finally, the geospatial and temporal semantics (i.e., non-sequenced) that have no counterpart in a traditional conceptual model (e.g., USM) are added via new constructs in the layer 3. We advocate explicating the non-sequenced semantics in a data dictionary.

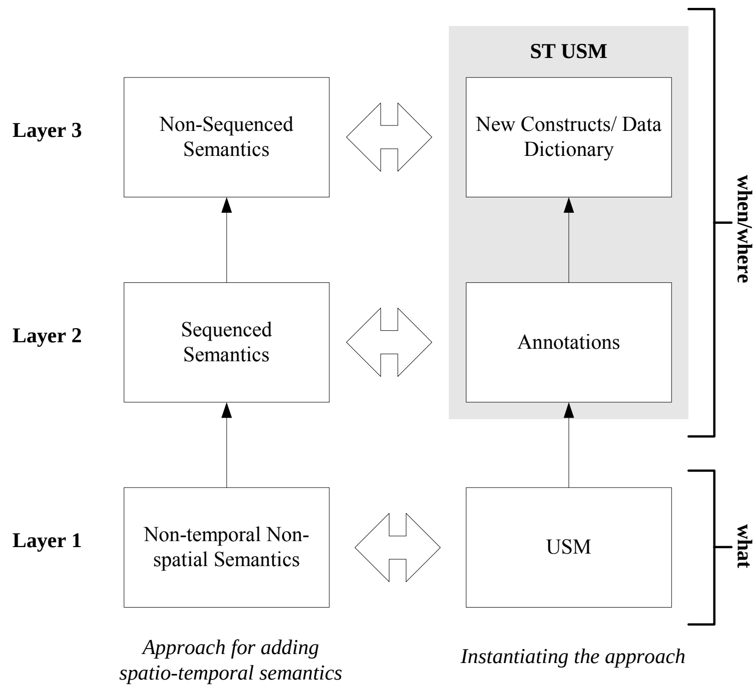

Figure 2: Inducing Space and Time Semantics with Three Layers

We next describe salient aspects of our proposed approach: representation of sequenced data semantics via annotations, orthogonality, and representation of non-sequenced data semantics.

2.4.1 Annotations

Our geo-spatio-temporal design methodology uses annotations to capture the semantics of (temporal and geospatial) sequenced statements. Via annotations, we enable a supplementary level of abstraction that succinctly encapsulates the geo-spatio-temporal data semantics, and naturally extends the semantics of a conventional conceptual model.

Annotated schemas do not invalidate legacy conceptual schemas as the syntax and semantics of conventional conceptual models is unaltered, i.e., annotated schemas are upward compatible. Upward compatibility implies that annotating the schema would induce the sequenced geo-spatio-temporal semantics; on the other hand, removing the annotations would render the schema with the traditional (snapshot) semantics. For example, in a conventional conceptual model a *key attribute* [30] uniquely identifies an entity (at a point in time). A *temporal key* [110] implies uniqueness *at each point in time*. As may be evident, the semantics of a temporal key here are implied by the semantics of a key in a conventional conceptual model. An annotated schema is snapshot reducible – as we assume that the data analysts will be conversant with conventional conceptual models, an extension using annotations should require minimum additional training costs, fewer errors and no significant drop in productivity.

2.4.2 Orthogonality

In our proposed methodology, orthogonality refers to two aspects: *syntactic* and *structural*. Syntactic orthogonality implies that geo-spatio-temporal semantics (i.e., "when" and "where") are syntactically orthogonal (cf. Appendix) to those of the conventional conceptual model (i.e., "what"). Syntactic orthogonality is consonant with how geo-spatio-temporal facts are represented in our memory (cf. 2.1.1). On the other hand, structural orthogonality implies that annotations are generic, and applicable to all types of conceptual modeling constructs, e.g., entity class, attribute, relationship. For example, an annotation phrase "S(day)/-//" can apply to an entity class, an attribute (including key, composite and multi-valued) or a relationship. We outline the syntax and semantics of annotations in Section 4.

2.4.3 Non-Sequenced Constraints

Previously [110], we have differentiated between *sequenced* and *non-sequenced* data semantics. For example, a *lifetime key constraint*—a type of temporal non-sequenced constraint—might require uniqueness over the entire lifespan of an entity; note that this constraint is orthogonal to the temporal sequenced key constraint.

In proposing the geo-spatio-temporal conceptual design methodology, we had to make choices related to what to explicate on the schema via annotations. This is consistent with conventional conceptual models, which do not explicate all the data semantics on the schema – e.g., a *uniqueness constraint* on an attribute is typically not shown on the schema, and may be specified in the *data dictionary*, an organized listing of data elements [92, 137]. We advocate documenting the non-sequenced semantics in a data dictionary for pragmatic reasons (i.e., help prevent overcrowding of the schema), and to maintain consistency (i.e., keep all the non-sequenced data semantics out of the schema). In the following, we focus on the sequenced data semantics represented via annotations.

In summary, our overall approach advocates first eliciting "what" (layer 1 in Figure 2) is important for the application, and subsequently eliciting the sequenced (layer 2 in Figure 2) and finally the non-sequenced data semantics (layer 3 in Figure 2). Having presented our approach and the rationale for our overall proposed approach, we next outline the ontological concepts that underlie annotations.

3. Needed Ontological Concepts

Design-support environments employ a formalism, i.e., a conceptual model, to help elicit the data-related requirements. Shoval and Frumermann [105] assert that a conceptual model should be powerful in "semantic expressiveness." Batini et al. [11] define expressiveness as the availability of a large variety of concepts for a more comprehensive representation of the real world. Wand et al. [127] propose that conceptual modeling be anchored in the models of human knowledge, and that ontology can be the basis for defining the semantics of the language. Ontology is the specification of the representational vocabulary for a shared domain of discourse [38], and space and time ontology is the basis for the dialog panel in the design-support environment as well as the associated annotations. We first present a motivating example, and then summarize the ontological concepts related to temporal [7, 14, 15, 27, 28, 49, 110-113] and geospatial [25, 33, 98, 132, 133] data that are embedded in DISTIL.

3.1 Motivating Example

Using an example of an application at USGS, we highlight geo-spatio-temporal data modeling requirements. We are working with a group of researchers who are developing a *ground-water flow model* [24] for the Death Valley region. Beneath the earth's surface, the zone where all interstices are filled with water is referred to as *ground water*. In arid regions like Death Valley, which encompasses approximately 80,000 km^2 in Nevada and California, ground water provides a large percentage of water for domestic, industrial and agricultural uses. The objective of the ground-water flow model is to characterize regional 3D ground-water flow paths so that policy makers can make decisions related to radio-nuclide contaminant transport, and the impact of ground-water pumping on national parks and local communities in the region. However, the quality of model outputs, and predictions based on the model are dependent on the data that forms an input to the model. We describe a subset of the input data required for the ground-water flow model.

Two key objects of interest for the ground-water flow model are *spring sites* and *borehole sites*. Both of these need to be geospatially referenced to the Earth, and are uniquely identified by a *site id*. A spring site is a point on the surface of the Earth given by geographic *x*- and *y*- coordinates, with a geospatial granularity of dms-second, i.e., seconds in the degree-minute-second specification of a geographical location. Geographically spring sites exist within a *spring* (represented as a region), and there can be many spring sites within a spring. A spring usually has a *name* by which it is known locally. An important characteristic of a spring is the *permanence* of discharge at the spring, e.g., `perennial springs` discharge continuously and `intermittent springs` are periodically dry. A borehole site refers to a part of the borehole whose 3D location is given by *x*- and *y*- coordinates on the Earth's surface, with a geospatial granularity of dms-second, and depth below the Earth's surface with a geospatial granularity of foot. While there can be one or more borehole sites at different depths within a drilled hole at the same surface location, each borehole site is associated with exactly one borehole. A borehole site is characterized by tests like *horizontal* (*hydraulic*) *conductivity* and *diffusivity*, and the values for these tests are valid for the minute at which the test was conducted. Spring sites also have an associated *description* that characterizes and defines the site. The measurements at the borehole site and spring site are taken by a *source agency*, and these measurements are central to the ground-water flow model.

A borehole site may have a pumplift that removes water from the borehole site, and this can affect other data collected at the borehole site. Some of the characteristics of a pumplift are *type* (e.g., `air lift`, `rotary pump`, `jet pump`), *manufacturer* and *serial number*. A pumplift has a lifespan that specifies the time periods when the pumplift has been operational. The time periods of pumplift existence denote the times when data collected at a borehole site can be influenced by a given pumplift.

The data semantics related to "what" is pertinent for this application (e.g., spring site, borehole site, borehole, spring, ground water station and source agency) can be elicited employing any conventional conceptual model [21, 30] – cf. Figure 4 in Section 4.1. The requisite geo-spatio-temporal data semantics (i.e., related to "when" and/or "where") described in the application above—based on geo-spatio-temporal ontology—is summarized next.

3.2 Temporal Ontology

The basis of time ontology is the definition of the time domain. We review extant definitions associated with how facts can interact with time [111-113]. Intrinsic to temporal data is temporal granularity. We summarize existing definitions associated with temporal granularity [14, 15, 27].

A *time domain* is denoted by the pair (T, ≤), where T is a nonempty set of *time instants* and "≤" is the total order on T. We can assume the time domain is either *discrete* or *dense*. While there is no general agreement if time domain is dense or discrete, the temporal database community agrees that a discrete model of time is generally adequate for representing reality [51]. Additionally, time is assumed to be bounded at both ends, i.e., the past and the future [109]. An *instant* is a time point on the time line. For example, ($\mathbf{Z}$, ≤) represents a discrete time domain where instants are isomorphic to integers, implying that every instant has a unique successor.

Facts can interact with time in two orthogonal ways resulting in *transaction time* and *valid time* [111]. Valid time denotes when the fact is true in the real world, and implies the storage of histories related to facts. On the other hand, transaction time links an object to the time it is current in the database. While the temporal granularity can be specified for valid time, that for transaction time is system-defined. The transaction time has duration from insertion to (logical) deletion [52] and can include granules only up to the current time granule in the real world. Time-varying data may be modeled as an *event* or a *state* [50].

An event occurs at a point of time, i.e., an event has no duration. A state has duration, e.g., a storm occurred from 5:06 PM to 5:42 PM.

Temporal granularity is a measure of the time datum. 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 with 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 29, 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 a 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.

3.3 Geospatial Ontology

Any data that can be associated with location on the Earth are referred to as geographic data [25]. Geographic space based on Euclidean geometry is the basis for most GISs [70] – location can be expressed by a set of coordinates, e.g., latitude and longitude. We briefly review concepts related to space and geospatial granularity.

We can view geospace 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 [67]. A geospatial object is associated with *geometry* and *position*. Geometry represents the shape and size of an object [25]. The position in space is based on coordinates in a mathematically-defined reference system, e.g., latitude and longitude. Geometry of the geospatial object may be 0-, 1- or 2-dimensional corresponding to a *point*, a *line* or a *region*. A point is a zero-dimensional geospatial object with coordinates, a line is a sequence of ordered points with a start node and an end node, and a region or polygon consists of one outer and zero or more inner rings [124].

For geographic applications, horizontal space is segregated from vertical space. We have adapted Worboys' formalism of geospatial resolution [132, 133] to define a notion of *horizontal* and *vertical geospatial granularities* [56] that parallels temporal granularity. Intuitively, the horizontal space domain corresponds to the Earth's surface while vertical space domain corresponds to the depth/height below/above the sea level. Horizontal geospatial granularity refers to the resolution of points in a system of *x*-*y* coordinates, and is a mapping from integers to any partition of horizontal space, where the partition may arise from pixellation of space – and may be a regular square or any other shape such as a triangular irregular network (TIN) or even an irregular shape (e.g., county). Formally, a horizontal geospatial granularity may be defined as a mapping SG_{xy} from index i to a subset of space domain such that: (i) granules from a geospatial granularity do not overlap; (ii) the index set of a geospatial granularity provides a contiguous encoding, though the granules in the space domain are not constrained to be contiguous in the underlying geospatial domain; and (iii) origin granule $SG_{xy}(0)$ is nonempty. Examples of horizontal geospatial 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°13′E/27°45′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 geospatial granularity.

3.4 Time-Varying Geospatial Ontology

In geography, space is indivisibly coupled with time [87]. Lately, there has been much interest in adding a temporal aspect to geographic databases [35] since time integrates human activity, orders events and separates cause from effect [126].

Peuquet [90] posits two types of geospatial change: change in location by change of position and change in geospatial extent by changing shape. Prior research [91, 120] has further refined the interaction between an object and space-time into: (i) Moving objects, i.e., objects whose position changes but whose shape does not. For example, a car moving on a road network changes its position over time while its shape does not. (ii) Objects whose geospatial shape changes *discretely*. For example, change of the shape of land parcels over time in a cadastral application. (iii) Change in shape along with continuous movement over time. For example, in modeling a storm, the shape and position changes over time continuously.

Having briefly outlined the geo-spatio-temporal ontology, we employ interaction with DISTIL to present our proposed approach.

4. Geo-Spatio-Temporal Conceptual Modeling

As shown in Figure 3, our geo-spatio-temporal conceptual modeling approach involves first developing a non-geo-spatio-temporal conceptual schema referred to as a *core conceptual schema* (cf. Section 4.1), augmenting the core schema with geo-spatio-temporal annotations (grayed portion of Figure 3) leading to an *annotated conceptual schema* (cf. 4.2), and explicating the semantics of the annotated schema resulting in the *translated conceptual schema* (cf. 4.3). Using an example, we illustrate the translation of a conceptual schema to a *logical schema* (cf. 4.4).

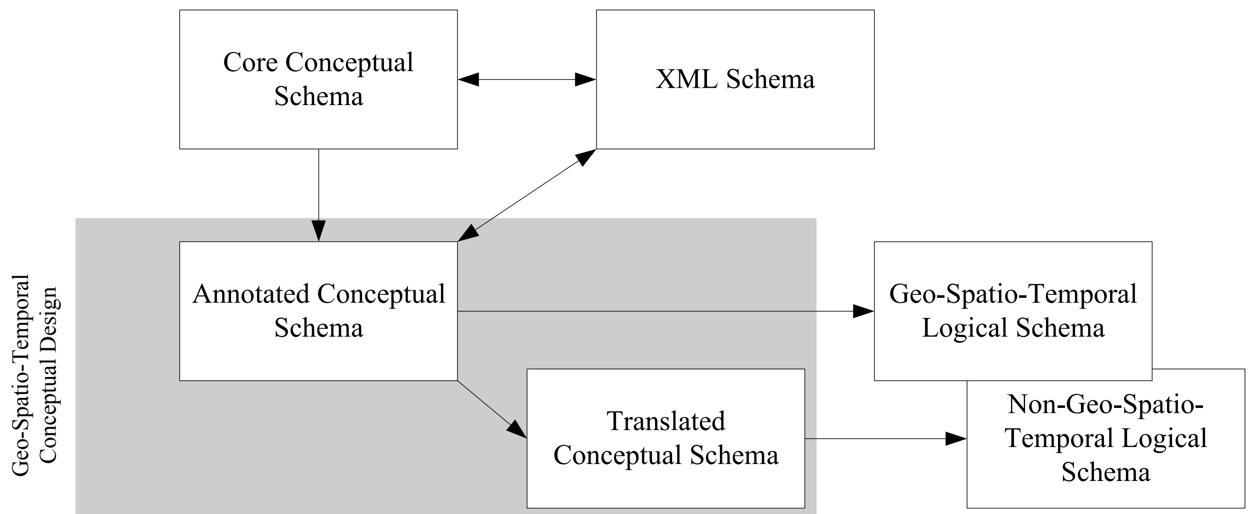

Figure 3: Overview of our Geo-Spatio-Temporal Conceptual Design

A geo-spatio-temporal logical schema innately "understands" the geo-spatio-temporal concepts that are encapsulated via annotations during geo-spatio-temporal conceptual design. This implies that the meaning of various geo-spatio-temporal aspects is built-in the schema – and the data analyst does not need to manually manage the geo-spatio-temporal data semantics, e.g., semantics of temporal granularity. As a result, the encoded semantics that is "wrapped" in the annotated conceptual schema does not need to be explicated—or "unwrapped"—in the geo-spatio-temporal logical schema. On the other hand, a non-geo-spatio-temporal logical schema needs translation from a *translated conceptual schema*, where the onus of defining and managing the geo-spatio-temporal semantics is on the data analyst. Note that the inherent constraints in the annotated schema are rendered as explicit constraints in the translated conceptual schema – these explicit constraints need to be managed by the data analyst. The translation of the core conceptual schema (i.e., non-temporal and non-spatial) or annotated schemas (i.e., geo-spatio-temporal) to an XML Schema helps generate "external schemas" (cf. 4.5).

4.1 Developing a Core Schema

We first summarize key terms and terminology related to USM [94], which is an extended version of the ER Model [21]. Note that our annotation-based geo-spatio-temporal conceptual design methodology is not specific to USM, and can be applied to any conventional conceptual model [21, 30].

USM is consistent with conventional conceptual models that include constructs to capture the data semantics related to *classification*, *aggregation*, *generalization*/*specialization* and *association*. Additionally, USM provides subtle semantics that segregates groupings from composites [96]. We chose USM as the base model for capturing "what" are the data semantics that are important for an application as it provides an overall framework that carefully defines entity class (i.e., classification) as well as various types of (class) relationships like interaction (i.e., association), generalization/specialization, and *composite* and *grouping* relationships (i.e., aggregation).

All real world objects are represented in the database as *entities*. Characteristics or properties of entities are referred to as *attributes*. A collection of entities for which common characteristics are to be modeled is called an *entity class* (or an *entity type*). The set of instances of an entity class is referred to as an *entity set*. In other words, an entity e of an entity class E may be designated as $e(E)$, and a set of entities of an entity class is represented by $S(E)$ where $e(E) \in S(E)$. Using the example of the USGS application described in Section 3, Figure 4 shows a core USM schema. This schema includes entity classes like SPRING_SITE, BORE_HOLE_SITE, SOURCE_AGENCY, PUMPLIFT, SPRING, BORE_HOLE, GROUND_WATER_STATION, and their various attributes; e.g., PUMPLIFT has attributes like type and manufacturer (mfg), and an identifier (or key) attribute, serial_no, which is represented by a shaded oval. Entity classes are created by using the constructs on the "USM Model" panel on the left side, specifically, the "Strong" rectangles.

An association between entities is referred to as a *class relationship*. We briefly describe various class relationships: interaction, generalization/specialization, composite and grouping. An *interaction relationship* (also referred to as a *relationship*)—among entity classes E_1, E_2,…, E_n—defines a set of associations among entities of the entity classes. For example, the relationship sp_measures—using the diamond "Rel" on the "USM Model" side panel—between SPRING_SITE and SOURCE_AGENCY relates an entity of SPRING_SITE with that of SOURCE_AGENCY.

Generalization implies that similar objects are related to a higher-level generic object – the constituent objects may be considered as the specialization of the generic object [8]. A generalization proceeds from the need for multiple classification of the same object. The crucial property of higher- and lower-level entities created by specialization and generalization is *attribute inheritance*, i.e., the attributes of higher-level entity classes (i.e., *superclass*) are said to be *inherited* by the lower-level entity classes (i.e., *subclass*) [106]. SPRING and BORE_HOLE have certain common properties that can be represented as

GROUND_WATER_STATION. A hexagon with "S" represents a generalization/specialization relationship, and the arrow points to the superclass, i.e., GROUND_WATER_STATION. Properties such as station_name and site_type are common to both SPRING and BORE_HOLE. On the other hand, the attributes like permanence and name are specific to SPRING, and source and method are specific to BORE_HOLE.

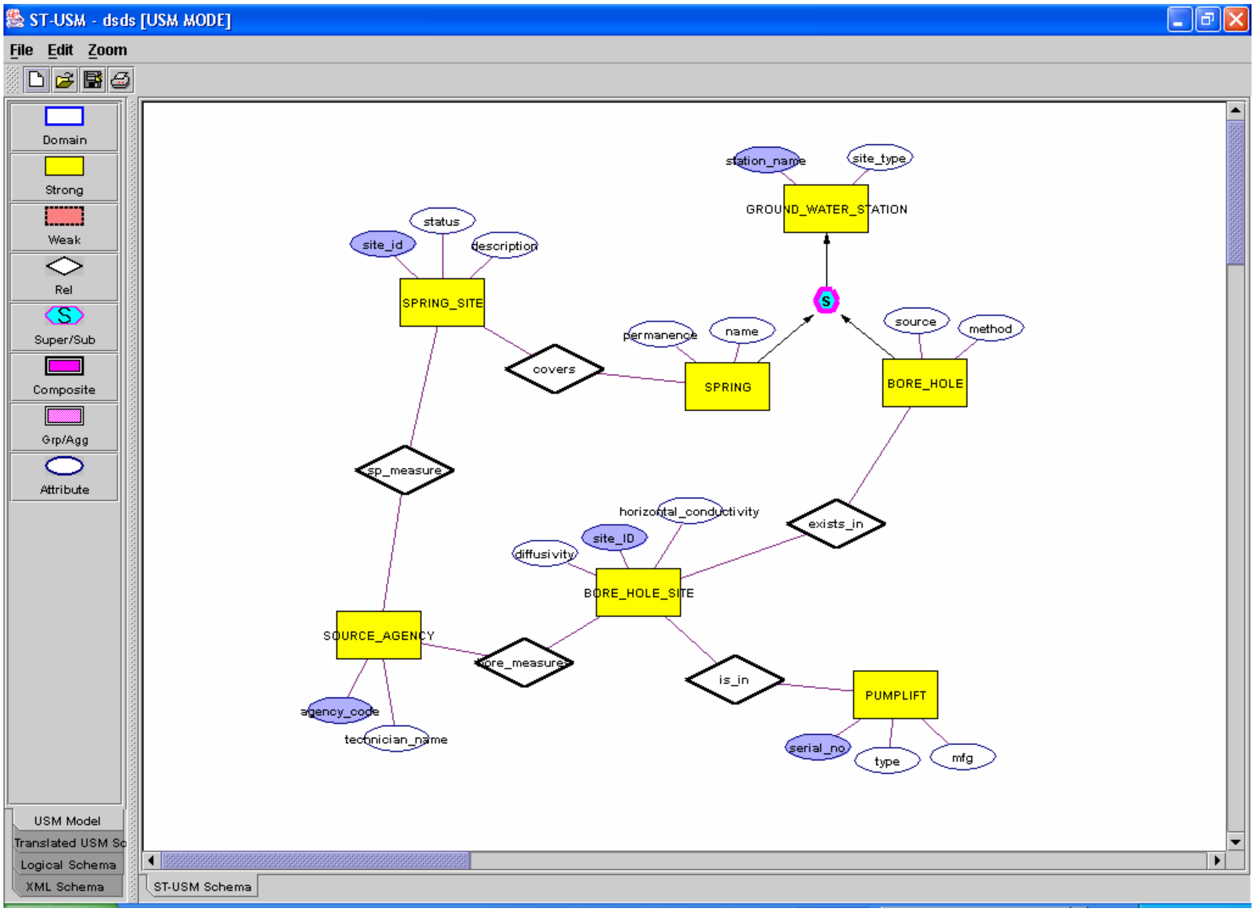

Figure 4: USM Schema

A composite relationship defines a new class called the *composite class* that has another entity set (or subsets of an entity set) as its members. A composite relationship is similar to the "power set grouping" in [102], in that they both represent a set whose members are subsets of the base entity set, $S(E)$. Each member from a composite class—referred to as a *composite*—is a subclass from some other class called the *base class*. If the requirements needed to model PUMPLIFT_TYPE, whose members were subsets of the base class, PUMPLIFT, PUMPLIFT_TYPE would be referred to as a composite class.

A grouping relationship—that establishes a "part-of" relationship—defines a new class, called a *grouping class* (also referred to as an aggregate [19]), whose members are physically or logically made up of members or sets of members from some other entity class(es) referred to as *component classes*.

Developing a schema like the one shown in Figure 4, which includes constructs like entity class, attributes, interaction relationships and generalization/specialization relationship, is supported by typical extant design-support environments. Note that developing such a schema—even without space and time aspects—is an involved task, and the resulting non-temporal and non-spatial schema can contain tens of entity types and hundreds of attributes. For example, a small fragment of the (non-temporal and non-spatial) schema for the USGS application includes 18 entity classes and 92 attributes [54].

We next show how a design-support environment that supports conventional conceptual modeling can be augmented with geo-spatio-temporal annotations.

4.2 Annotating the Core Schema

Via annotations, we exemplify a supplementary level of abstraction that "naturally" extends the semantics of a conventional conceptual model to capture the geo-spatio-temporal data semantics. The conceptual model that captures the geo-spatio-temporal semantics via annotations is referred to as ST-USM [57]. We describe the syntax related to annotations, and then show how DISTIL—based on data analysts' inputs—automatically creates annotation phrases within the schema. Note that these annotation phrases succinctly encapsulate the geo-spatio-temporal semantics of the application, and can be associated with an entity class, an attribute or a relationship.

As shown in the Appendix, the overall structure of an *annotation phrase* has the following components:

⟨temporal annotation⟩ // ⟨geospatial annotation⟩ // ⟨time-varying geospatial annotation⟩.

The temporal annotations, geospatial annotations and time-varying geospatial annotations are each separated by a double forward slash (//). In the Appendix, the terminals are shown in Tahoma font (e.g., foot) and non-terminals are shown within angular brackets (e.g., ⟨foot⟩). "ϵ" represents a null value and "|" refers to "or." Details related to temporal and geospatial indeterminacy ("~" and "+-" in the Appendix) are outside the scope of this paper, and described elsewhere [56].

4.2.1 Specifying Temporal Annotations

The temporal annotation first specifies the existence time (or valid time) followed by the 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 "-". The valid time or existence time can be modeled as an event (E) or a state (S), and has an associated temporal granularity. For example, "S(min)/T//" associated with PUMPLIFT would denote that PUMPLIFT is bitemporal, i.e., both valid time and transaction time (T) need to be recorded; the temporal granularity of the states (S) is minute (min). The granularity associated with transaction time is not specified as it is system-defined.

The temporal annotation described above can be specified using a dialog panel (e.g., Figure 5). In the pop-up box, the data analyst can specify if the application needs to organize data based on time.

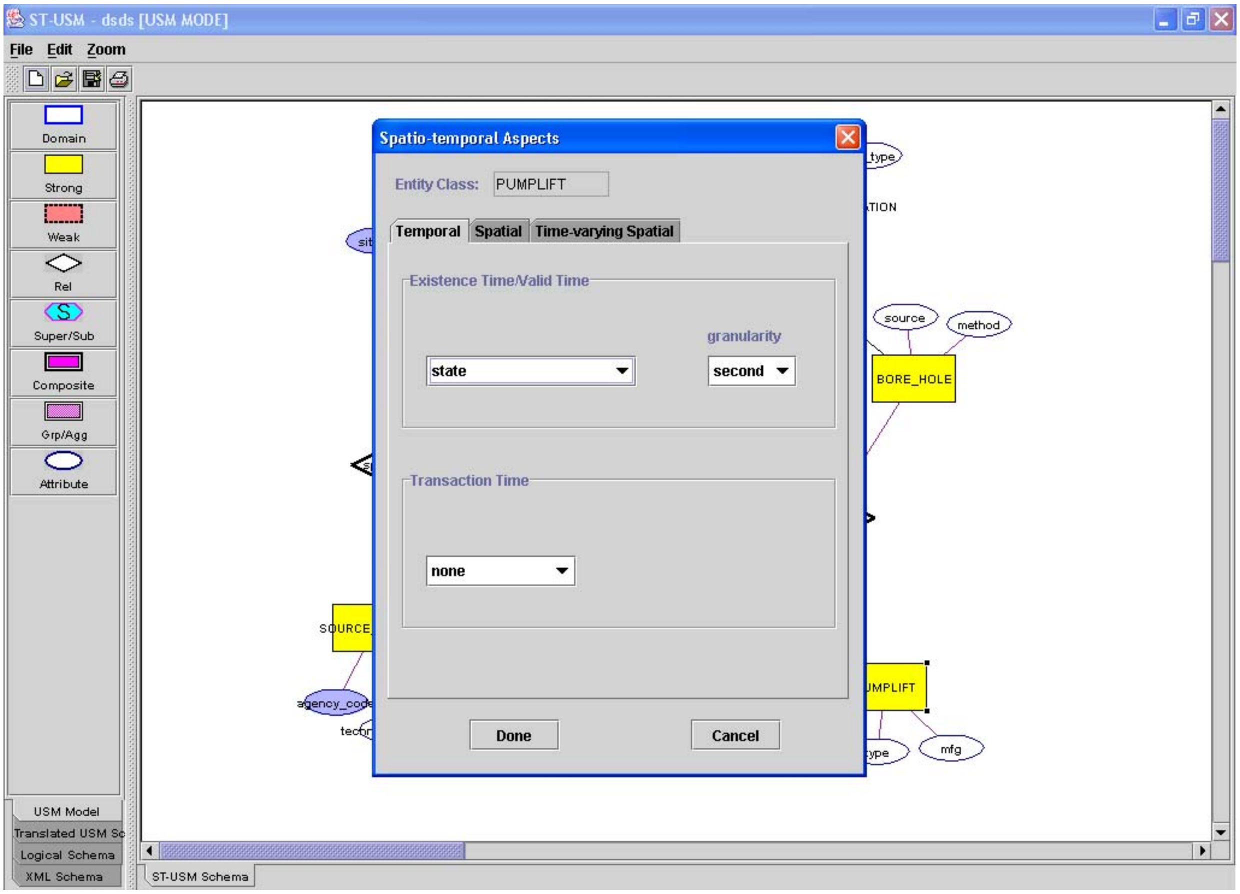

Figure 5: Specifying Temporal Aspects

The valid time may be represented as an event or state, and has an associated temporal granularity. On the other hand, granularity associated with transaction time does not need to be specified as it is system-defined. For example, Figure 5 shows how the data analyst can enter temporal details which would result in an annotation phrase “S(day)/-//” for PUMPLIFT. This annotation succinctly describes that the lifespan of a pumplift need to be represented as a state (S) with temporal granularity of day, and that transaction time is not pertinent (“-”).

4.2.2 Specifying Geospatial Annotations

Figure 6 shows how the data analyst can enter geospatial details that result in an annotation phrase “//P(dms-sec)/P(dms-sec)/-” for SPRING_SITE. The geospatial annotation includes geometry, and position in *x*-, *y*- and *z*-dimension; each dimension is segregated by a forward slash (/). For example, "// P(dms-sec) / P(dms-sec) / -" implies that the SPRING_SITE needs to be represented as a point ("P") on the *x*-*y* plane. The associated horizontal geospatial granularity is dms-sec.

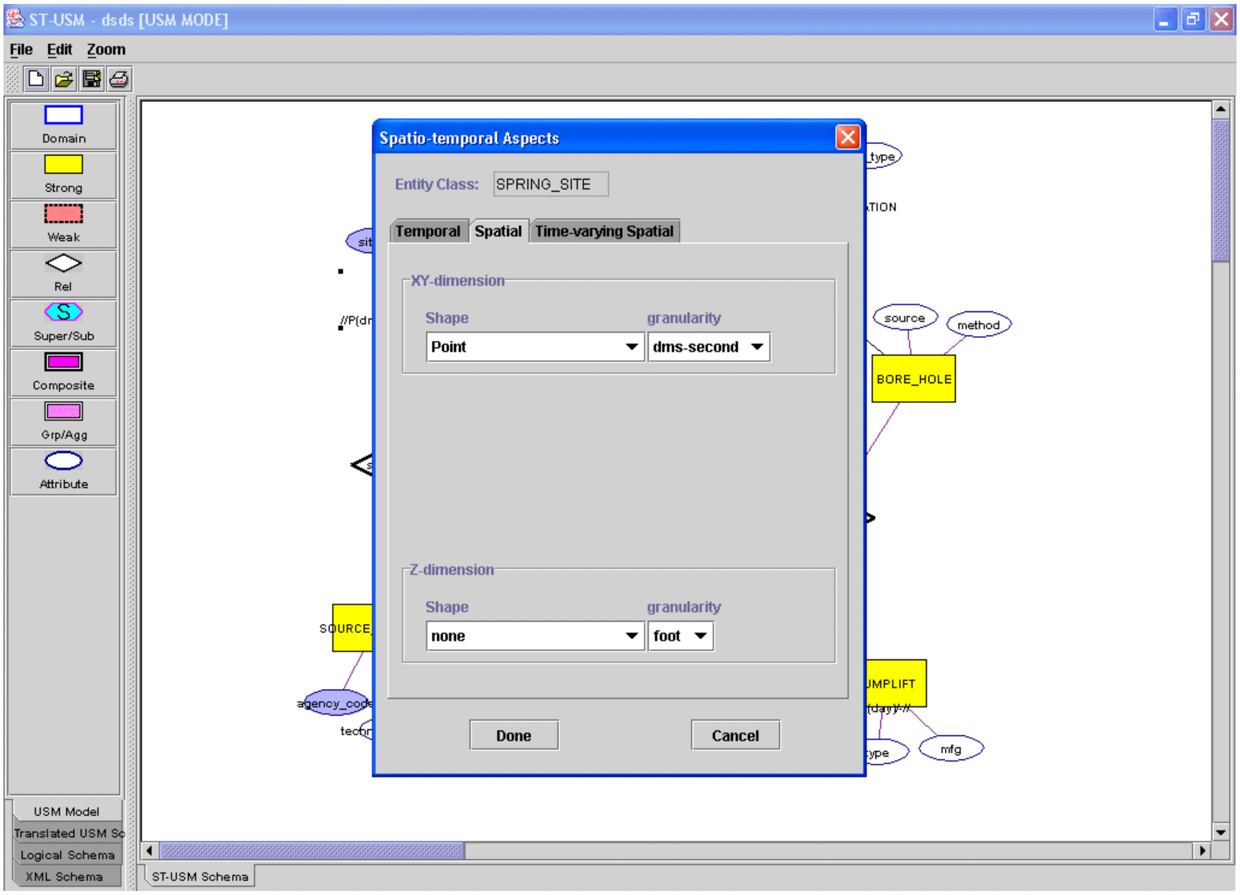

Figure 6: Specifying geospatial Aspects

4.2.3 Specifying Time-Varying Geospatial Annotations

The interaction between an object and space-time can result in change in the shape and/or change in the position of an object. A time-varying geospatial annotation can be specified only if geospatial and temporal annotation have already been specified. For example, a class of moving car tracked by satellite may be represented by an annotation phrase "E(sec)/-// P(deg)/ P(deg)/-//Pos@xy," which denotes a time-varying position while the shape is time-invariant. The geometry is a point (P) in an *x*-*y* plane with a geospatial granularity of degree. The position changes in the *x*-*y* plane (Pos@xy) over time and each geometry is valid for time granules (E, i.e., event) measured in second.

Figure 7 shows the dialog panel for specifying the time-varying geospatial aspects of an application. The four options are: neither shape or position is changing, shape is changing, position is changing, and both shape and position are changing over time. Additionally the dimension over which the change is happening can be specified.

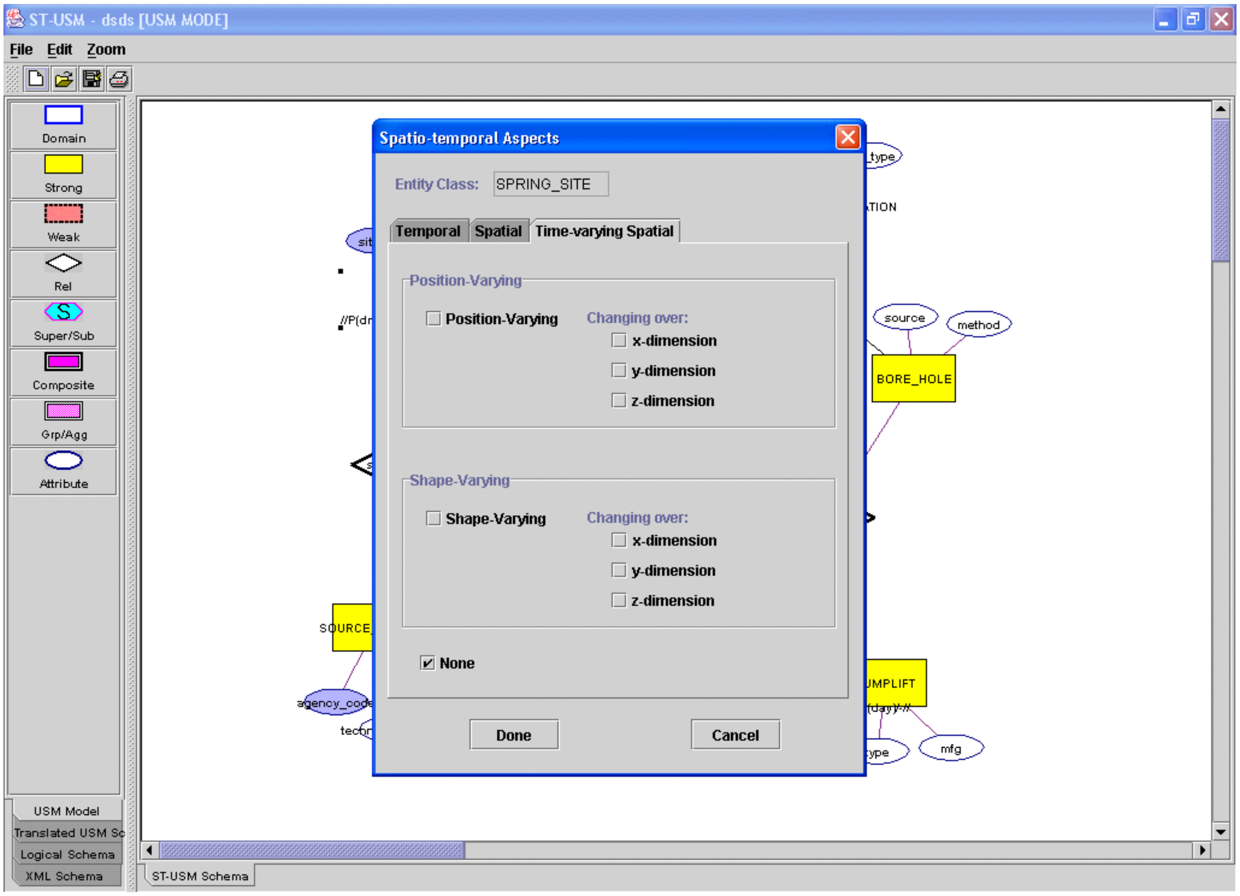

Figure 7: Specifying Time-Varying geospatial Aspects

In summary, for each construct in the core USM schema the data analyst, in consultation with the users, considers whether temporality and geospatiality are important for the application. The data analyst asks questions like: Do you want to store the history or only the current value of 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 objects, their properties or associations between objects? What is the geographical shape of objects, their properties or associations between objects? What is the associated geospatial granularity? Can the geospatial shape/position for these objects change over time? Accordingly, the data analyst enters the details using the dialog panel as shown in Figure 5, Figure 6 and Figure 7.

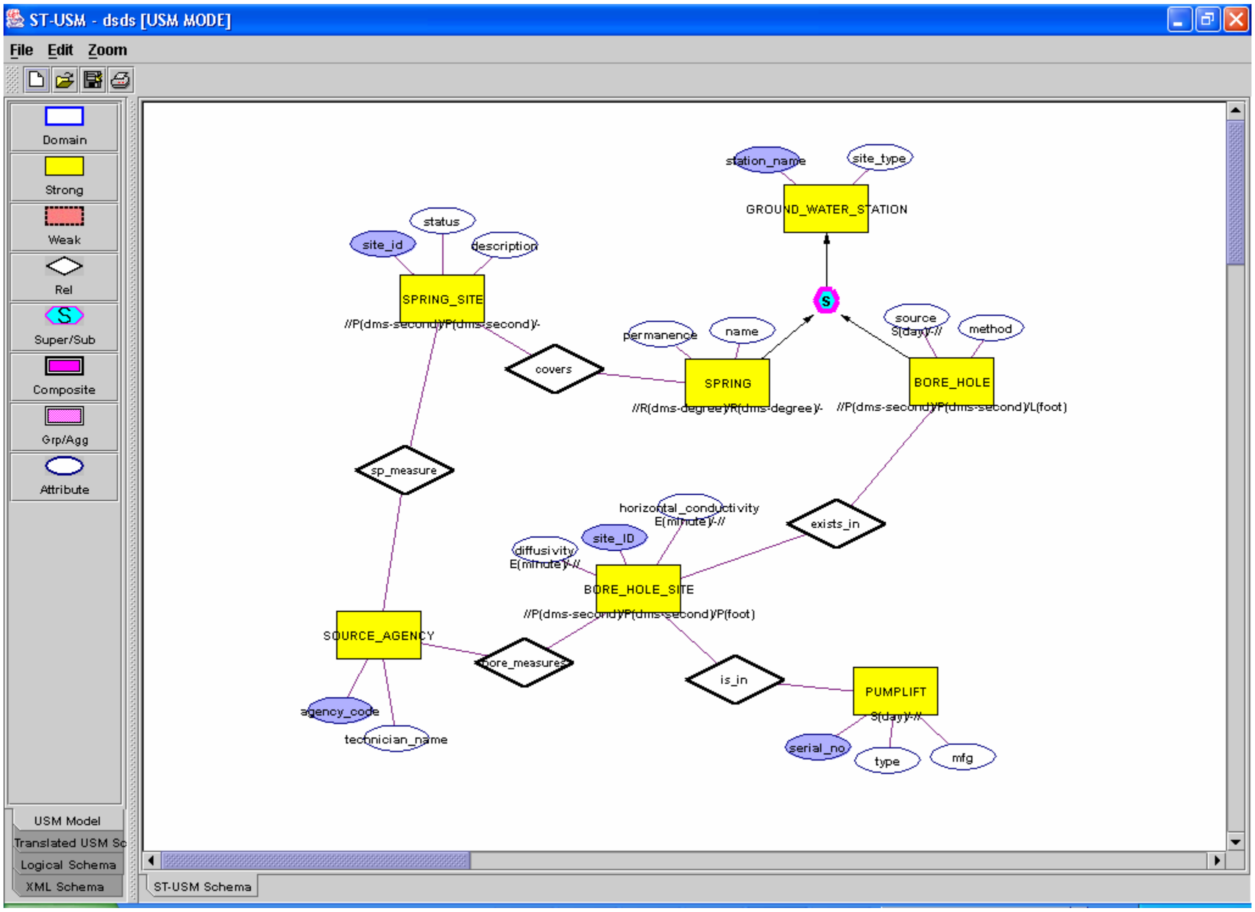

Figure 8: Annotated Schema

The schema shown in Figure 8 is automatically annotated according to the information filled into the pop-up boxes. The data analyst can annotate each entity class (e.g., SPRING_SITE, BOREHOLE_SITE, SPRING, BORE_HOLE, PUMPLIFT), relationship and attribute (e.g., diffusivity, horizontal_conductivity, source). Once the data analyst has made the annotated schema, the requirements so collected can be established with other users. Note how the annotated schema encapsulates the geo-spatio-temporal semantics for an application, and can be employed to verify (with the user) if the requisite geo-spatio-temporal semantics have been elicited on the schema.

Our annotation-based approach is upward compatible [17, 112], i.e., our approach renders conventional conceptual schemas geo-spatio-temporal without affecting the legacy schemas. The schema shown in Figure 8 includes both "un"annotated constructs (e.g., SOURCE_AGENCY) with conventional semantics, where space and time may not pertinent for the application, and annotated constructs (e.g., PUMPLIFT), where an annotation phrase represents the temporal and/or geospatial data semantics.

4.3 Semantics of the Annotated Schema

In this section, we describe how the encapsulated geo-spatio-temporal semantics can be explicated. Note that explication of the temporal and geospatial data semantics—resulting in what is referred to as a translated schema—is based on time and space ontology described in Section 3. Using examples of temporal and geospatial entity class, we show detailed semantics of an annotated abstraction. Details related to other constructs like attribute, interaction relationship, subclass, composite class and grouping class are described elsewhere [54].

A temporal entity class implies that the membership of an entity in the entity set is time-varying. We assume that a temporal entity class (as contrasted with entities of that class) exists during the entire modeled time. Thus, 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. When an entity class is defined as temporal, it implies that the application would have queries like "What is the average monthly power consumption by all pumplifts over their installed existence?" and "What are the pumplifts that were installed before 1995 and are operational now?"

Figure 9 illustrates the representation of existence time expressed as state (S) with day as the temporal granularity name. Based on users' requirements, the data analyst simply annotates PUMPLIFT with "S(day)/-//" and does not need to contend with the complexity of the underlying semantics or the associated temporal constraints. Figure 9 also shows the semantics of a temporal entity class in ST USM via a mapping using the concepts of a conventional conceptual model, which we refer to as a *translated USM schema*. This mapping from ST USM to (translated) USM is *snapshot equivalent*; i.e., the two schemas (ST USM and translated USM) represent the same information content over snapshots taken at all times. 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 PUMPLIFT_has_ET associates an entity with exactly one TEMPORAL_GRANULARITY (1:1). 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 class 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 (i.e., 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* [27], which can help a user choose the level of detail associated with facts. Details related to granularities and indeterminacy is presented elsewhere [56].

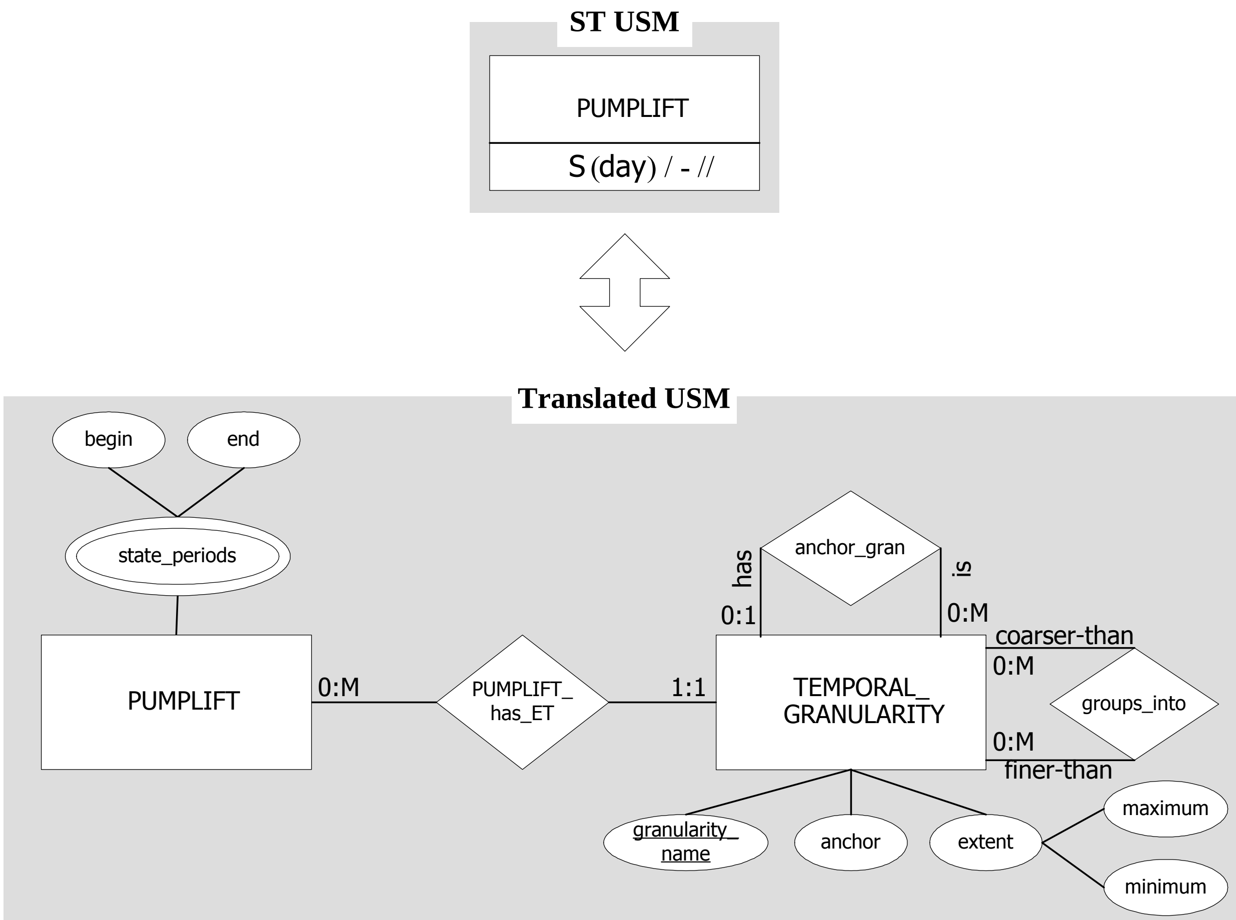

Figure 9: Temporal Entity Class in ST USM and its semantics in USM [57]

A temporal entity with existence time may have a set of event_instants or state_periods associated with it depending on whether a temporal entity is represented as an event or a state. A time period of PUMPLIFT is represented with indexes begin and end of state_periods. A double-lined ellipse in USM denotes a multi-valued attribute. For example, state_periods is represented as a multi-valued attribute and represents a set of state periods (i.e., a temporal element) associated with an entity. Additionally, eight constraints will be generated in the translated USM schema for PUMPLIFT [57]. These constraints are implicit in the ST USM schema but are explicit in the translated USM schema. As may be evident, a straightforward annotation phrase (i.e., "S(day)/- ") represents, or encapsulates, relatively involved semantics.

A *geospatial entity class* refers to geo-referenced entities with an associated shape and position, which is used to locate them in a two- or three-dimensional space. When an entity class (e.g., a spring site) is defined as geospatial, it implies that the application would have queries like "What are the three closest spring sites to Mesquite spring?" and "What is the surface area of Mesquite spring?"

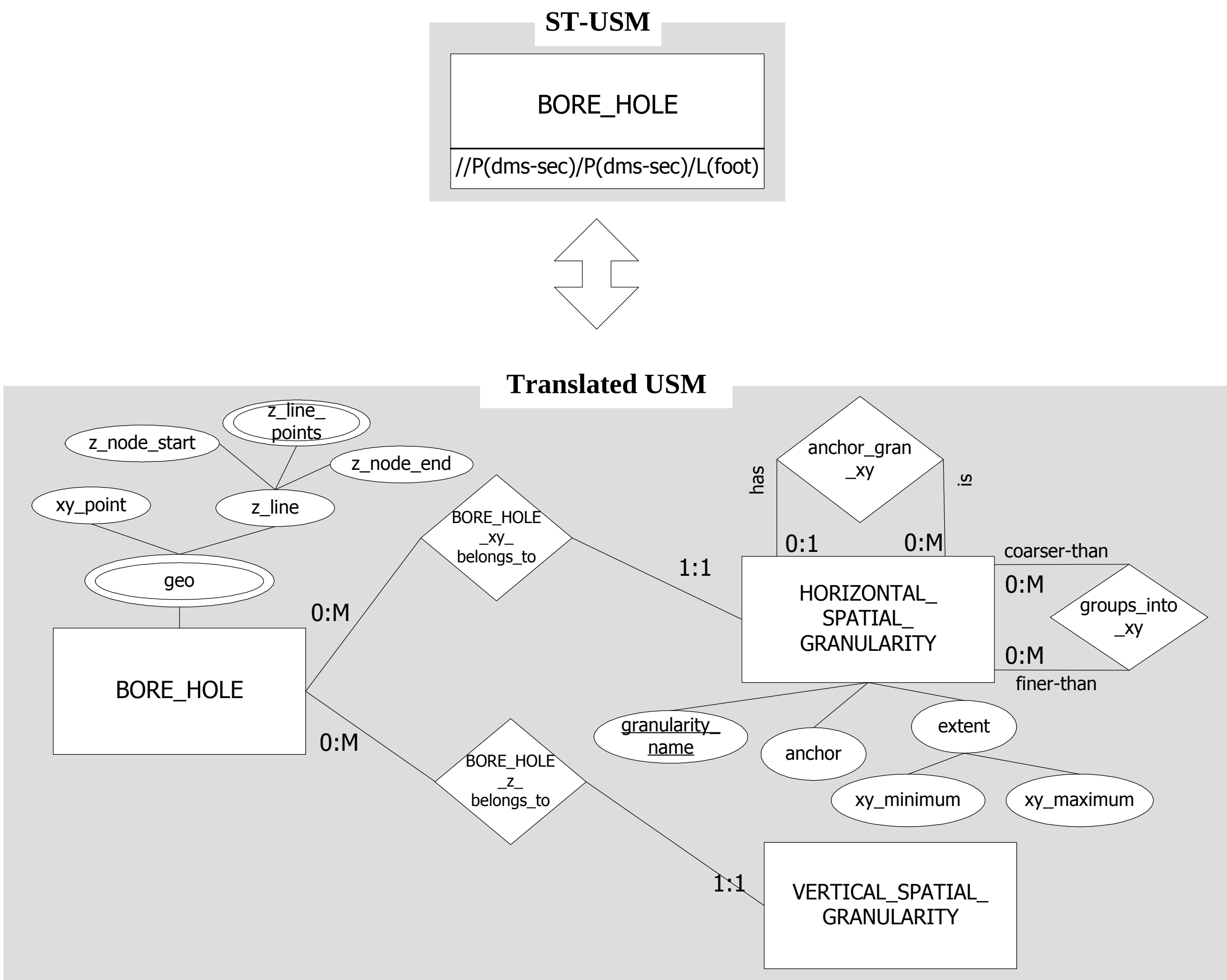

Figure 10: Geospatial Entity Class in ST USM and its semantics in USM

A geospatial entity class, BORE_HOLE, embedded in a three-dimensional space, is associated with HORIZONTAL_SPATIAL_GRANULARITY and VERTICAL_SPATIAL_GRANULARITY (Figure 10). A geospatial object in a three-dimensional space has two associated relationships, BORE_HOLE_xy_belongs_to and BORE_HOLE_z_belongs_to corresponding to its horizontal and vertical geospatial granularities. 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. We have not shown the details associated with VERTICAL_SPATIAL_GRANULARITY as they similar to that of HORIZONTAL_SPATIAL_GRANULARITY. In Figure 10, a line starts with a node (i.e., z_node_start) and ends with a node (i.e., z_node_end), and has multiple points (i.e., z_line_points) between the start and end nodes [181].

We next present the associated horizontal geometry related constraints implicit in an ST USM schema. Constraints 1.1 and 1.2 are based on our definition of a geospatial entity, constraints 1.3−1.6 are based on the definition of geospatial granularity, and constraints 1.7−1.8 are based on the geometry of geospatial entities.

Constraint 1.1: All entities of BORE_HOLE must have the same horizontal geospatial granularity.

$\forall\ e \in S$(BORE_HOLE),
e. BORE_HOLE_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY(granularity_name)= dms-sec

Constraint 1.2: Every entity of BORE_HOLE has an associated geometry within the specified extent.

$\forall\ e \in S$(BORE_HOLE), $\exists g \in e$(geo),
e. BORE_HOLE_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY (extent.xy_minimum) $\leq$
g.xy_point $\leq e$. BORE_HOLE_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY (extent.xy_maximum)

Constraint 1.3: The indexes corresponding to the geometry of a geospatial object lie within xy_minimum and xy_maximum.

$\forall\ e \in S$(BORE_HOLE), $\forall g \in e$(geo),
e. BORE_HOLE_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY(extent.xy_minimum)$\leq$
g.xy_point $\leq e$. BORE_HOLE_xy_belongs_to.HORIZONTAL_SPATIAL_GRANULARITY(extent.xy_maximum)

Constraint 1.4: Extent is well-formed.

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

Constraint 1.5: 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 1.6: 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)

Constraint 1.7: The horizontal and vertical geometry of BORE_HOLE are represented as a point and a line, respectively.

$\forall\ e \in S$(BORE_HOLE), $\forall\ g \in e$(geo), g.xy_point $\in$ *point* $\wedge\ g$.z_line $\in$ *line*

Constraint 1.8: The vertical geometry is associated with a horizontal geometry.

$\forall\ e \in S$(BORE_HOLE), $\forall g_1 \in e$(geo), $\exists g_2 \in e$(geo),
g_1.z_line.z_node_start = g_2.xy_point $\wedge\ g_1$.z_line.z_node_end = g_2.xy_point

The implicit constraints related to vertical geospatial granularity can be defined like constraints 1.1-1.6.

We next show how the encapsulated geo-spatio-temporal semantics can be explicated using DISTIL. To view detailed explicit semantics associated with the annotated schema, the data analyst clicks on the "Translated USM Schema" tab to obtain a translated USM schema corresponding to the ST-USM schema (Figure 11). We have embedded translation rules (e.g., Figure 9 and Figure 10) into DISTIL, to help translate the annotated constructs to an equivalent USM schema with explicit representation of the associated geospatiality and temporality. In Figure 11, the portion of the ST-USM schema on the right has been translated into the translated USM schema on the left.

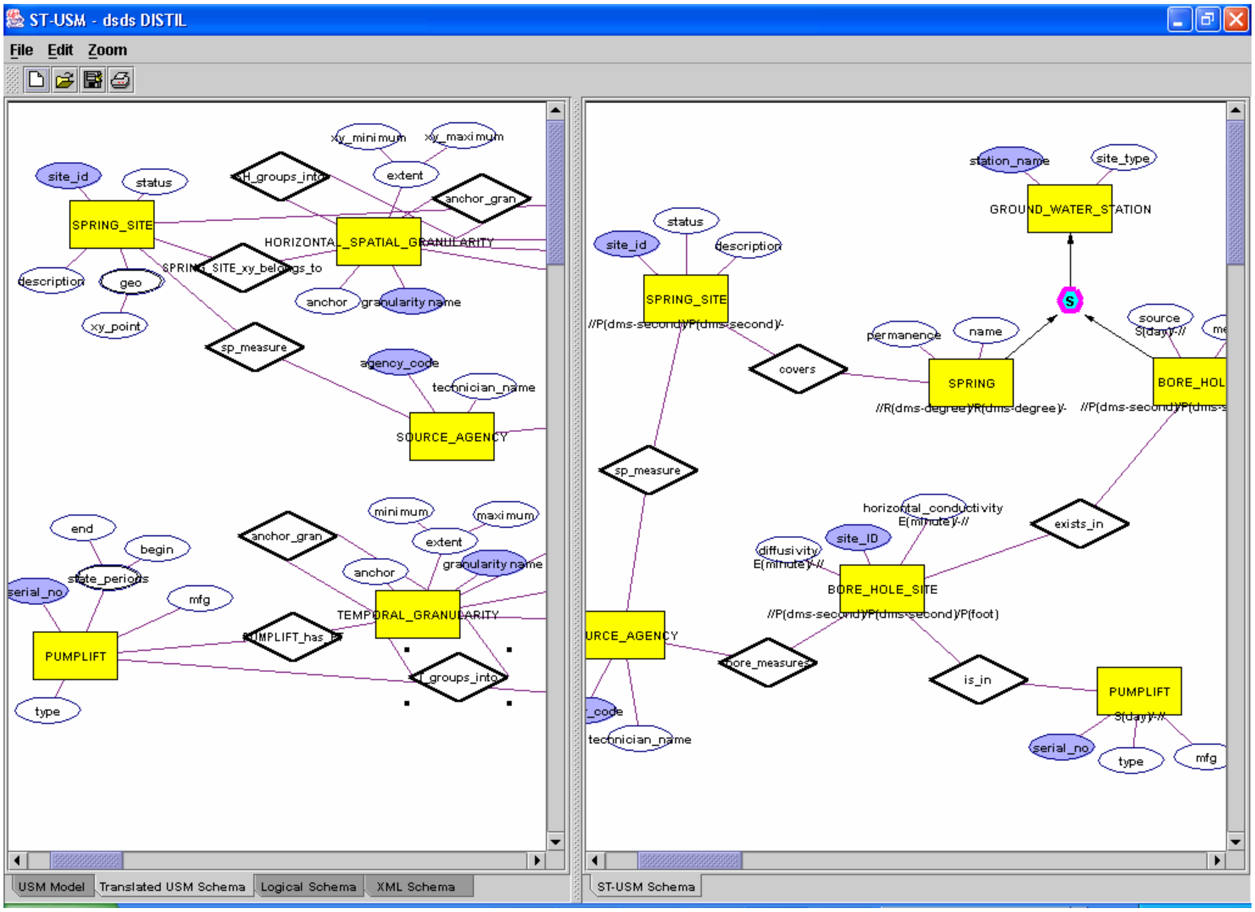

Figure 11: Semantics of the Annotated Schema

For example, the semantics associated with a temporal entity class PUMPLIFT includes an entity class TEMPORAL_GRANULARITY, which specifies the temporal granularity in which PUMPLIFT is embedded. The relationship PUMPLIFT_has_ET relates an entity from PUMPLIFT with a corresponding temporal granularity. A multi-valued attribute state_periods (with components begin and end) is added to the entity PUMPLIFT because PUMPLIFT lifespan was modeled as state. A multi-valued attribute implies that each PUMPLIFT can have many associated state_periods. Similarly, other constructs of the annotated ST-USM schema are converted to a translated USM schema using the embedded rules in DISTIL.

Note how annotation phrases succinctly encapsulate the geo-spatio-temporal data semantics on the conceptual schema, and how these semantics are "unpacked" in the translated USM schema. This rendition from an annotated schema to a (translated) USM schema is snapshot equivalent, that is, the two schemas (ST USM and translated USM) represent the same information content over snapshots taken at all times.

The translated schema that is shown in Figure 11 includes only a fragment of the entire schema, which includes 12 entity classes, 51 attributes, 22 relationships and 42 constraints. Thus, a relatively straightforward annotated schema (e.g., Figure 8) encapsulates rich-semantics explicated in the translated schema. In this section, we described how our formal ontology-based approach helps elicit—at a conceptual level—the geo-spatio-temporal data semantics, e.g., *event* and *state* [49], *valid time* and *transaction time* [111], *existence time* [36], *temporal granularities* [14, 15, 27], *shape* and *position* [25], *spatial resolution* [132, 133], and *change in position* and/or *shape* over time [91, 120]. These formalized semantics are the basis for the development to the representational schema, which we describe next.

4.4 Logical Schema

While conceptual models provide a mechanism that is close to the way users perceive the data, physical models provide concepts how data is stored in the computer. A *representational model*, e.g., relational model, provides concepts that may be understood by users, and is not too far removed from the way data is organized in the computer [30]. Mapping rules that provide correspondences between conceptual and representational model constructs are applied in logical design, and result in the development of a *logical schema*.

The mappings to a logical schema depend on the type of representational model used. Standard SQL (Structured Query Language), the basis for a relational schema, does not include time support except for user-defined time. As a result, over 50 temporal query languages have been proposed [49], most of which are a result of extending SQL for the temporal domain. While change proposals to SQL3 that includes temporal semantics is referred to as SQL/Temporal [112, 113], the Open GIS Consortium has proposed extensions to SQL that supports simple geospatial collections that is referred to as SQL/OGIS [78]. For each of these languages, mapping rules may be developed to convert an ST USM schema to a logical schema in that language. As illustrated in Figure 3, there are two mapping rules, which depend on the logical model used: a non-temporal logical model (e.g., SQL) or a temporal/ geo-spatio-temporal conceptual model (e.g., SQL/Temporal, SQL/OGIS). None of the existing DBMS products yet support a geo-spatio-temporal logical model. So, we do not discuss this latter case further, other than to note that such a mapping could be developed as an extension of a previously-developed mapping from ER to SQL/Temporal [110]. Instead, we briefly describe a mapping from a geo-spatio-temporal conceptual schema to a logical schema that does not include built-in support for temporal and geospatial semantics. This mapping is more complex than the one to a geo-spatio-temporal logical data model, because there are no special geo-spatio-temporal constructs, such as temporally-sequenced primary and foreign keys, provided by the logical model that can be exploited in the mapping.

The translation from conceptual schema to logical schema is similar to the one described in standard database textbooks [30]. Each constraint described in the previous section needs to be translated to an assertion, which, in turn, can be implemented using a trigger in Oracle [110].

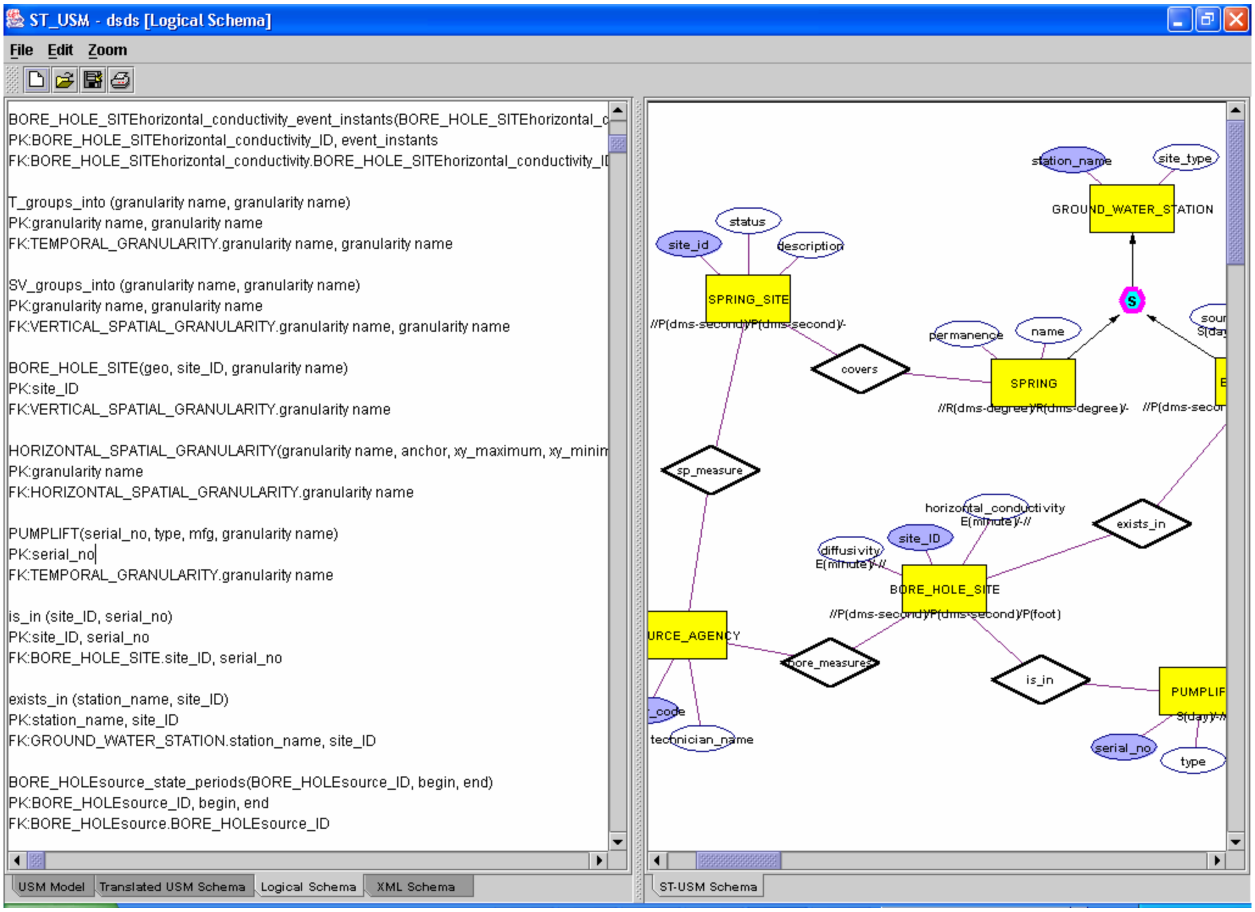

Figure 12: Logical Schema

Once the users' geo-spatio-temporal requirements have been captured and validated, the data analyst can click the "Logical Schema" tab to get the logical schema (Figure 12). This schema includes the name of the table, attributes in the table, primary key and foreign key (if any). For example, PUMPLIFT table includes four attributes: serial_no, type, mfg and granularity_name. The primary key (PK) is serial_no and the foreign key (FK) is TEMPORAL_GRANULARITY.granularity_name.

While the tool described above gives a textual description of the logical schema, this can be tailored for specific DBMSs, such as Oracle. The geospatial data types are provided by geospatial abstract data types in the object-oriented approach [90]. For example, Oracle Spatial [80, 81] includes a pre-defined object type SDO_GEOMETRY and a table for SPRING can be defined as shown below.

```
CREATE TABLE SPRING (
        station_name VARCHAR2(30) PRIMARY KEY,
        name VARCHAR2(100),
        permanence NUMBER (5,2),
        geo MDSYS.SDO_GEOMETRY);
```

Oracle Spatial defines the object type SDO_GEOMETRY that includes SDO_GTYPE, which has code corresponding to various valid geometries, e.g., point, line – these codes are specified during data insertion. Note that these are all implementation-specific issues that are outside the scope of this paper.

We described how different mapping rules can be employed to translate a given conceptual schema to a logical schema. These mapping rules depend on the specific logical model, and on the DBMS under consideration. We illustrated how a methodical approach ensures that the geo-spatio-temporal semantics elicited during conceptual design are embedded in the subsequent logical schema. A design-support environment can automate this translation, thus, ensuring that the rich geo-spatio-temporal semantics elicited during conceptual design are not "lost" during translation to the logical schema.

4.5 Encoding a Conceptual Schema to an XML Schema

XML provides a standard format for exchanging conceptual schema among diverse design-support environments. If the schemas are generated by different design-support environments—with different syntax but same semantics—the XML schema can enable sharing of the conceptual schema across platforms. Using an example of PUMPLIFT, we show the translation rules to an XML schema.

```
  <?xml version="1.0" ?>
- <ST_USM xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://www.cs.arizona.edu/timecenter/STUSM/usm.xsd">
  - <simple_class name="PUMPLIFT">
      <normal_attribute name="serial_no" datatype="INTEGER" mandatory="yes" />
      <normal_attribute name="type" datatype="STRING" mandatory="no" />
      <normal_attribute name="mfg" datatype="STRING" mandatory="no" />
    - <primarykey_constraint>
        <normal_attribute name="serial_no" />
      </primarykey_constraint>
    </simple_class>
  - <ST_ANNOTATION>
    - <simple_class name="PUMPLIFT">
        <state granularity="day" />
      </simple_class>
    </ST_ANNOTATION>
  </ST_USM>
```

Core Schema: "What" Semantics

Annotations: "When"/ "Where" Semantics

Figure 13: XML Schema for a fragment of ST-USM Schema

While the first part of the XML schema shown in Figure 13 describes the semantics of conventional conceptual schema (e.g., USM), the second part corresponds to the geo-spatio-temporal annotations. The second part of the XML schema—between ⟨ST_ANNOTATION⟩ and ⟨/ST_ANNOTATION⟩—specifies the geo-spatio-temporal data semantics. In this case, the temporal entity class PUMPLIFT is represented as a state having a granularity of day. Note how the orthogonality of geo-spatio-temporal semantics in a conceptual schema (e.g., Figure 8) is consistently maintained in an XML schema, where the conventional semantics (i.e., “what”) are segregated from the “when/where” semantics.

Having described key design issue (i.e., orthogonality) in the translation from a conceptual schema to an XML schema, we next describe how a design-support environment supports such a translation. The data analyst can click on “XML Schema” tab to obtain the XML schema corresponding to the ST USM Schema. This XML file can be saved (“Save XML File”) and is useful for sharing schemas among data analysts using different design-support environments.

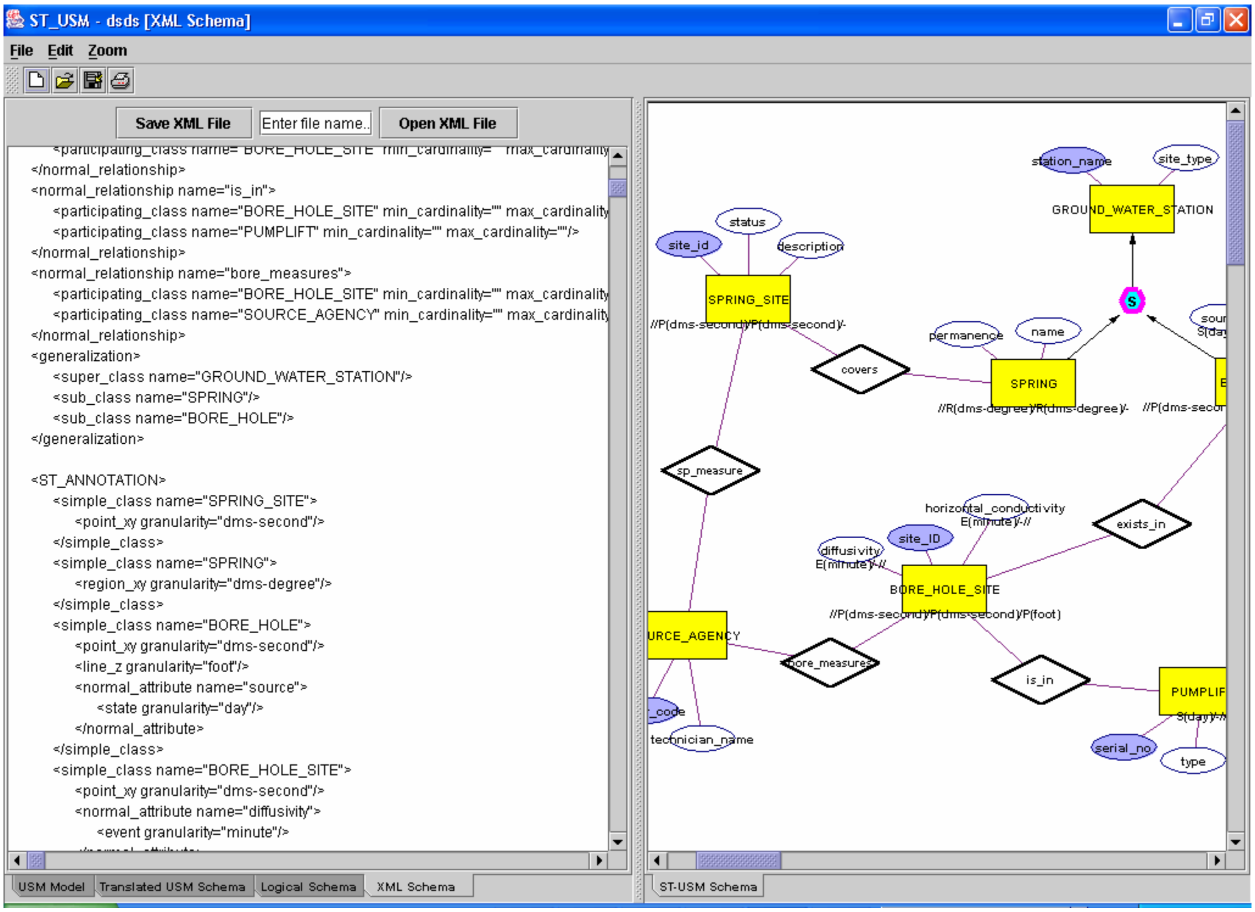

Figure 14: XML Schema

An XML schema can be useful in many ways. A geo-spatio-temporal conceptual schema can be “exported” to an XML schema. The XML schema can be “imported” into another design-support environment, with possibly different syntax, but same underlying semantics, i.e., one that supports representation of classification, association, generalization/specialization and aggregation. Thus, conversion of conceptual schemas to XML schemas can help support exchange and sharing of conceptual schemas across design-support environments.

In this section, we illustrated how the “when” and “where” semantics can be kept orthogonal to the “what” semantics even in the XML schema (cf. Figure 13). Such orthogonality ensures that the geo-spatio-temporal XML schemas are upward compatible with conventional XML schemas.

5. Architecture

We now describe the underlying architecture of DISTIL (Figure 15) that enables the development of geo-spatio-temporal conceptual schemas described in previous section. Figure 15 also explicates our geo-spatio-temporal conceptual design approach that segregates “what” (during *Conventional Conceptual Design*) from “when/where” (during *Geo-Spatio-Temporal Conceptual Design*).

As illustrated in Figure 15, the data analyst first develops a *Core USM Schema* during Conventional Conceptual Design using *USM Schema Designer*. The USM Schema Designer allows the data analyst to develop a core conceptual schema (e.g., Figure 4) that captures data requirements without considering *temporal* or *geospatial aspects*. We have adapted the USM Schema Designer from the data modeling module of CREAM [86].

Our Geo-Spatio-Temporal Conceptual Design includes annotating the Core USM Schema via *Annotation Designer* (e.g., Figure 5, Figure 6) resulting in the *ST-USM Schema* (e.g., Figure 8). This architecture also supports the translation of existing USM schemas that have geo-spatio-temporal semantics incorporated in an ad-hoc manner (perhaps because they were designed with CASE tools that did not have such support). The user loads such a schema in as a Core USM Schema. For each time-varying entity class, relationship and attribute, the user can annotate that semantic object, then manually remove the ad hoc modeling constructs. For example, if the original schema had used a ternary relationship to model a time-varying binary relationship (with one of the entity classes being a time value), the relationship could be designated as time-varying with an annotation, then the time value entity class and its connection to the relationship removed, leaving a simpler schema with the same abstract semantics. The user could then add more detail to the annotation, such as granularity, indeterminacy; it is doubtful that all of these details were in the original schema. Because the resulting schema is at a higher level of abstraction, different logical schemas could then be generated, under user control.

A consistent *ST-USM Schema* (e.g., Figure 8) can next be converted to a *Translated USM Schema* (e.g., Figure 11) through the *Semantic Mapper*. While the geo-spatio-temporal semantics are encapsulated in the ST-USM schema, these semantics are explicated in the translated USM schema. While the consistent ST-USM schema can be employed for eliciting and validating (with the user) the geo-spatio-temporal requirements for an application, the translated USM schema is useful for translation to a logical schema that is interpretable by a DBMS.

As shown in Figure 15, our proposed Geo-Spatio-Temporal Conceptual Design implemented via DISTIL integrates with Conventional Conceptual Design, and the translated USM Schema merges again with the *Conventional Logical Design*. The *Logical Mapper* in Conventional Logical Design includes rules to convert a Translated USM Schema to *Relational Schema* with geospatial support that can be implemented in a relational DBMS. The *XML Mapper* converts a USM or ST USM Schema to an *XML Schema*. The Schema in XML format can be shared by distributed teams which may be, possibly, using different design-support environment with different modeling formalism (i.e., syntax) but the same underlying semantics. In the future, it would be useful to incorporate the *ST-USM Logical Mapper*, which would translate the ST-USM schema to an *SQL3/Temporal-OGIS Schema* that can be implemented in a geo-spatio-temporal DBMS.

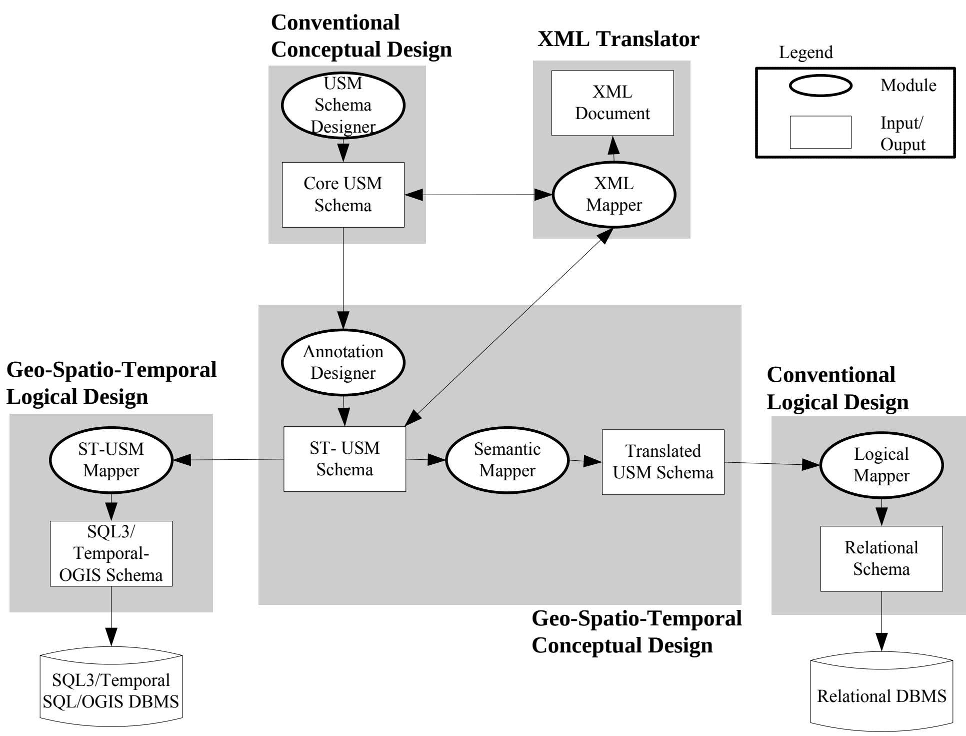

Figure 15: DISTIL Architecture

DISTIL has been implemented using Java 2 (JDK 1.2) and Oracle 8.1.6. A dialog panel in DISTIL (cf. Figure 5, Figure 6, Figure 7) is created to elicit the geospatial and temporal semantics associated with an entity class, a relationship or an attribute. When a user clicks an icon of a persistent object (e.g., an entity class), a dialog panel pops up and allows the user to input the geospatial and temporal information of that persistent object. The "annotation" class is a Java Bean implemented to capture the geospatial and temporal aspects associated with a persistent object that becomes a property of the persistent object. This annotation class also summarizes the geospatial and temporal information into a simple annotation string. This string is displayed on the drawing canvas and stored in the database. An application can keep track of the geospatial and temporal information of an object class simply by looking up the annotation string that accompanies the object. Having described the design-support environment, we next evaluate the proposed approach instantiated via DISTIL.

6. Evaluation

Batini et al. [11] posit that conceptual models should possess the following qualities: *expressiveness*, *simplicity*, *minimality* and *formality*. These qualities can also be the basis for evaluation of a design-support environment that supports elicitation of geo-spatio-temporal data semantics.

Expressiveness refers to the availability of a large variety of concepts for a more comprehensive representation of the real world. We propose intuitive ontology-based dialog panels in DISTIL, which comprehensively capture the semantics related to space and time. These pop-up boxes automatically annotate the schema, thus, helping represent the geo-spatio-temporal data semantics on the conceptual schema.

One of the conflicting goals related to expressiveness is simplicity, which requires that augmenting an existing design-support environment should be minimal. As shown in Table 1, the additional number of classes (and lines of code) required to augment an existing design-support environment (i.e., [86]) with the geo-spatio-temporal semantics is relatively modest.

Module	Classes	Lines of Code (kLOC)
USM Schema Designer	118	34.1
Annotation Designer	6	2.2
Semantic Mapper	3	1.4
Logical Mapper	2	1.0
XML Mapper	5	2.4

Table 1: Number of Classes and Lines of Code for Modules of DISTIL

The annotation designer and semantic mapper entailed a 10.6% increase in the number of lines of code. Additionally, adding the geo-spatio-temporal semantics via annotations did not entail any changes to the existing code. Thus, orthogonality of a conceptual framework (i.e., annotation-based approach) is mirrored with orthogonality of an implementation structure (i.e., DISTIL). Additionally, to incorporate annotations the changes to the (original) database repository were minimal. The XML schema encoding an annotated ST USM Schema captures geo-spatio-temporal semantics orthogonal to conventional semantics. All this implies that our approach is straightforward from the perspective of repository (database) design and application development.

Minimality ensures that no concept can be expressed through composition of other concepts. Because of orthogonality in the annotated ST-USM schema and the corresponding XML schema, minimality is also supported.

Formality specifies that the model must present a unique, precise and well-defined interpretation. Wand et al. [128] posit that effective use of conceptual modeling constructs requires that their meanings be defined "rigorously." The syntax and semantics of the underlying ST-USM is formally defined using BNF (cf. Appendix) and first-order logic [54], respectively.

Snapshot reducibility ensures that the semantics of a geo-spatio-temporal conceptual model are understandable in terms of the semantics of the conventional conceptual model, thus, helping ensure minimum additional investment in data analyst training.

Upward compatibility allows the legacy and geo-spatio-temporal schemas to co-exist. Upward compatibility requires that the syntax and semantics of the traditional conceptual model, e.g., [21, 30], remain unaltered. Our proposed approach has not altered the syntax or semantics of extant models.

While we posit that the geo-spatio-temporal annotation presented in this paper is comprehensive, it is impossible to assert completeness with conceptual modeling because any formalism is motivated in part by pragmatic rather than purely theoretical reasons [102]. It is possible that the formalism presented in this paper may need to be extended. Since geo-spatio-temporal annotation is orthogonal to the conceptual modeling constructs, our annotation-based approach is not only generic but also straightforward to extend.

7. Related Work

In presenting a framework for research in conceptual modeling, Wand and Weber [129] differentiate between *conceptual-modeling grammar* and *conceptual-modeling method*. While the former provides constructs and rules to model the miniworld, the latter provides procedures by which a grammar can be modeled. In the following, we discuss how extant work in geo-spatio-temporal conceptual modeling has focused rather exclusively on grammar.

We refer the reader to Gregersen and Jensen [37] for an excellent survey on temporal conceptual modeling grammars. Given the need to capture the geo-spatio-temporal data semantics, many divergent proposals have been made. Claramunt et al. [22] present design patterns for representation of spatio-temporal processes. Worboys [131] proposed a mathematically-oriented model that includes several dimensions: spatial, graphical, temporal and textual/numeric. Claramunt and Thériault [23] integrate time in GIS to propose TGIS, an event-oriented model. For modeling the geo-spatio-temporal data semantics, various conceptual modeling grammars have been proposed. While some of them are based on extant conceptual models [9, 42, 83-85, 120-122], others are GIS-application oriented formalisms [39, 40, 62, 104, 118]; we focus on the former.

To represent the geo-spatio-temporal data semantics, MADS (*M*odeling of *A*pplication *D*ata with *S*patiotemporal) [83, 85] and STER (*S*patio-*T*emporal *E*ntity *R*elation) [120] employ an annotation-based grammar. In MADS, geospatiality can be associated with object types, attributes, relationships and aggregation. Space and time features are supported via abstract data type (ADT). While spatial ADTs provide shape and location information, temporal ADTs support timestamping. Spatial entities are associated with a spatial ADT, *geo*, e.g., point, line. Each of these shapes has an icon associated with it, which is used to indicate spatiality. A spatiotemporal entity is one whose geospatial reference is recorded by defining the spatial ADT and by associating a temporal ADT to that fact. MADS defines topological constraints: disjunction, adjacency, crossing, overlapping, inclusion and equality. STER is a graphical extension to the ER Model that applies geo-spatio-temporal concepts to the modeling constructs of the ER model. Geographic objects are ones whose position needs to be recorded. Spatial attributes are properties of space that indirectly become properties of objects. Geographic objects may be related to each other via spatial relationships, and represent spatial constraints: topological, directional and metric.

With the growth of geographic applications, design tools to support modeling of geo-spatio-temporal data have been proposed. Perceptory [13] provides a Spatial PVL (Plug-in for Visual Languages) that adds graphical notations into the UML (Unified Modeling Language) Class Model. The Spatial PVL includes three basic constructs (0-, 1- and 2-dimensional shape) with a number of variations (e.g., complex shape composed of 1-D river and 2-D lakes), all of which are represented as pictograms on the schema. The spatial constraints can be captured using textual description. Perceptory primarily focuses on geometry associated with geospatial objects and attributes, while ignoring support for geo-spatio-temporal relationships. Additionally, it is a drawing tool developed as a Visio template. On the other hand, a visual schema editor based on MADS [83] aids in capturing geo-spatio-temporal data semantics. Design using the visual schema editor involves drag-and-drop operations along with interaction with windows-like forms. The visual schema editor includes modules to transform a MADS schema to an equivalent ER-like schema or a relational schema.

Prior research in geo-spatio-temporal conceptual modeling—including our prior work [56, 57]—has focused on conceptual-modeling grammar. To the best of our knowledge, this is the first paper that focuses on providing underlying principles, and defining an overall geo-spatio-temporal conceptual design approach (based on the set guidelines) that augments extant conceptual design methodology. We presented an annotation-based approach that is upward compatible with extant conceptual modeling approach. In our proposed approach, annotations are employed to induce sequenced geo-spatio-temporal semantics. It would be useful to extend design-support environment (like DISTIL) to include various non-sequenced data semantics proposed in prior literature [83, 85, 120], e.g., topological constraints, lifetime constraints.

8. Conclusion

Modeling—a process of abstraction in which important details are represented while unimportant ones are ignored—is often employed to manage complexity in problem-solving [116]. Conceptual modeling is one of the more involved parts of systems design: it involves an open-ended semantically rich problem space that requires both broader conceptual thinking as well as focused problem-solving activities [12].

Via a proof-of-concept prototype, we illustrated how orthogonality and upward-compatibility can be consistently applied to: (i) proposed approach; (ii) geo-spatio-temporal schemas, both conceptual and XML; and (iii) design-support environment implementation. The geo-spatio-temporal schema—developed using DISTIL—can be used as communication vehicle: it can also be used to decide if all the geo-spatio-temporal requirements of the user have been captured, and whether the requirements are conflicting. Schemas developed via DISTIL can be saved as XML schemas, which can be used for schema exchange among data analysts. Moreover, incorporating annotation—via pop-up boxes—into an existing CASE tool is also straightforward to implement. An empirical study that evaluated our geo-spatio-temporal conceptual modeling approach on comprehension and ease of use is outside the scope of this paper, and described in detail elsewhere [55].

In the future, it would be useful to embed various non-sequenced semantics from other modeling grammars—e.g., lifetime constraints [36, 141], topological constraints [120]—into our overall proposed approach. We plan to incorporate the *ST-USM Logical Mapper*, which would translate the ST-USM schema to an *SQL3/Temporal Logical Schema* [112, 113] and an *SQL/OGIS Logical Schema* [78] that can be implemented in a temporal/geospatial DBMS. It would be useful to consider the details of tailoring the mapping for specific DBMSs, e.g., Oracle. Additionally, we want to investigate generating triggers in the logical schema that are the temporal and geospatial equivalents of the non-temporal constructs in USM.

9. Acknowledgements

We are indebted to three anonymous reviewers and the associate editor for their insightful comments that substantially improved this paper. We are grateful to Arvind Gupta, Jun Liu, Mihir Shah and Xingtao Wang for their help in development of various parts of DISTIL. This research was supported in part by NASA grant 314401, and NSF grants IIS-0100436 and EIA-0080123.

10. References

[1] "Extreme CHAOS," Standish Group, 2001, http://standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf.

[2] "The CIO Insight Research Study Legacy Systems," CIO Insight, 2002, http://common.ziffdavisinternet.com/download/0/1805/0121_research.pdf.

[3] "WRQ survey finds legacy systems still used by the majority of UK companies," M2 Communications, 2003, http://articles.findarticles.com/p/articles/mi_m0ECZ/is_2003_Oct_7/ai_108610963.
[4] "Extensible Markup Language (XML)," 2004, http://www.w3.org/TR/2004/REC-xml11-20040204/.
[5] T. Abraham and J. F. Roddick, "Survey of Spatio-Temporal Databases," *GeoInformatica*, vol. 3, no. 1, pp. 61-99, 1999.
[6] K. K. Al-Taha, R. T. Snodgrass, and M. D. Soo, "Bibliography on Spatiotemporal Databases," *International Journal of Geographical Information Systems*, vol. 8, no. 1, pp. 95-103, 1994.
[7] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832-843, 1983.
[8] J. R. Anderson and G. H. Bower, *Human Associative Memory*. Washington, D.C.: V. H. Winston & Sons, 1973.
[9] S. Balley, C. Parent, and S. Spaccapietra, "Modeling geographic data with multiple representations," *International Journal of Geographical Information Science*, vol. 18, no. 4, pp. 327-352, 2004.
[10] R. D. Banker and R. J. Kauffman, "Reuse and Productivity in Integrated Computer-Aided Software Engineering," *MIS Quarterly*, vol. 15, no. 3, pp. 375-401, 1991.
[11] C. Batini, S. Ceri, and S. B. Navathe, *Conceptual Database Design: An Entity-Relationship Approach*: Benjamin/Cummings Publishing Company, 1992.
[12] D. Batra and J. G. Davis, "Conceptual Data Modeling in Database Design: Similarities and differences between Expert and Novice Designers," *International Journal of Man-Machine Studies*, vol. 37, no. 1, pp. 83-101, 1992.
[13] Y. Bedard, "Visual Modeling of Spatial Databases: Towards Spatial PVL and UML," *Geomatica*, vol. 53, no. 2, pp. 169-186, 1999.
[14] 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.
[15] C. Bettini, S. Jajodia, and S. X. Wang, *Time Granularities in Databases, Data Mining and Temporal Reasoning*. Berlin: Springer-Verlag, 2000.
[16] M. Böhlen, C. S. Jensen, and B. Skjellaug, "Spatio-temporal database support for legacy applications," Proceedings of the 1998 ACM symposium on Applied Computing, Atlanta, Georgia, United States, ed., Atlanta, Georgia, United States, pp. 226-234, February 27 - March 01, 1998.
[17] M. H. Böhlen, C. S. Jensen, and R. T. Snodgrass, "Temporal Statement Modifiers," *ACM Transactions on Database Systems*, vol. 25, no. 4, pp. 407-456, 2000.
[18] J. C. Brancheau, B. D. Janz, and J. C. Wetherbe, "Key Issues in Information Systems Management" 1994-1995 SIM Delphi Results," *Management Information Systems Quarterly*, vol. 20, no. 2, pp. 225-242, 1996.
[19] M. L. Brodie, "On the Development of Data Models," in *On Conceptual Modeling: Perspectives from Artificial Intelligence, Databases and Programming Languages*, M. L. Brodie, J. Mylopoulos, and J. W. Schmidt, Eds.: Springer-Verlag, 1984, pp. 19-47.
[20] 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.
[21] 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.
[22] C. Claramunt, C. Parent, and M. Thériault, "An Entity-relationship Model for Spatio-Temporal Processes. DS-7 1997: 455-475," Data Mining and Reverse Engineering: Searching for Semantics, IFIP TC2/WG2.6 Seventh Conference on Database Semantics (DS-7), Leysin, Switzerland, ed. S. Spaccapietra and F. J. Maryanski, Leysin, Switzerland, pp. 455-475, 1997.

[23] C. Claramunt and M. Thériault, "Managing Time in GIS: An Event-Oriented Approach," Proceedings of the International Workshop on Temporal Databases: Recent Advances in Temporal Databases, Zürich, Switzerland, ed. J. Clifford and A. Tuzhilin, Zürich, Switzerland, pp. 23-42, 17-18 September, 1995.
[24] 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.
[25] 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.
[26] G. B. Davis, "Strategies for Information Requirements Determination," *IBM System Journal*, vol. 21, no. 1, pp. 4-30, 1982.
[27] 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.
[28] C. E. Dyreson and R. T. Snodgrass, "Supporting Valid-Time Indeterminacy," *ACM Transactions on Database Systems*, vol. 23, no. 1, pp. 1-57, 1998.
[29] M. J. Egenhofer, "Spatial SQL: A Query and Presentation Language," *IEEE Transactions of Knowledge and Data Engineering*, vol. 47, no. 3, pp. 261-273, 1994.
[30] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, Fourth ed. Redwood City, CA: Benjamin/ Cummings Publishing Co., 2003.
[31] P. N. Finlay and A. C. Mitchell, "Perceptions of the Benefits from Introduction of CASE: An Empirical Study," *MIS Quarterly*, vol. 18, no. 4, pp. 353-370, 1994.
[32] D. Finnigan, E. A. Kemp, and D. Mehandjiska, "Towards an Ideal CASE Tool," International Conference on Software Methods and Tools, ed., pp. 189-197, 6-9 November, 2000.
[33] A. U. Frank, "Spatial Concepts, Geometric Data Models, and Geometric Data Structures," *Computers and Geosciences*, vol. 18, no. 4, pp. 409-417, 1992.
[34] B. Gervin, "Creating an XML Dictionary in ERwin 3.5," 2001, http://www.infoadvisors.com/articles/ERwin40/xml_dictionary_in_erwin.htm.
[35] M. F. Goodchild, "Geographic Information Systems," in *Ten Geographic Ideas that Changed the World*, S. Hanson, Ed. New Brunswick, New Jersey: Rutgers University Press, 1997.
[36] H. Gregersen and C. Jensen, "Conceptual Modeling of Time-Varying Information," TIMECENTER Technical Report TR-35, September 10 1998.
[37] H. Gregersen and C. S. Jensen, "Temporal Entity-Relationship Models-A Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 3, pp. 464-497, 1999.
[38] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199-220, 1993.
[39] O. Gunther and J. Lamberts, "Object-Oriented Techniques for the Management of Geographic and Environmental Data," *The Computer Journal*, vol. 37, no. 1, pp. 16-25, 1994.
[40] O. Gunther and W. F. Riekert, "The Design of GODOT: An Object-Oriented Geographical Information System," *IEEE Data Engineering Bulletin*, vol. 16, no. 3, pp. 4-9, 1993.
[41] R. H. Güting and M. Schneider, "Realm-Based Spatial Data Types: The ROSE Algebra," *VLDB Journal*, vol. 4, no. 2, pp. 243-286, 1995.
[42] T. Hadzilacos and N. Tryfona, "An Extended Entity-Relationship Model for Geographic Applications," *SIGMOD Record*, vol. 26, no. 3, pp. 24-29, 1997.
[43] J. Hahn and J. Kim, "Why are some representations (sometimes) more effective?" Proceedings of the Twentieth International Conference on Information Systems, Charlotte, North Carolina, ed. P. De and J. I. DeGross, Charlotte, North Carolina, pp. 245-259, December 13-15, 1999.
[44] C. Harder, *ArcView: GIS means Business*. Redlands, CA: Environmental Systems Research Institute, Inc., 1997.
[45] A. F. Hutchings and S. T. Knox, "Creating Products -- Customers Demand," *Communications of the ACM*, vol. 38, no. 5, pp. 72-80, 1995.

[46] J. Iivari, "Why Are Case Tools Not Used?" *Communications of the ACM*, vol. 39, no. 10, pp. 94-103, 1996.
[47] J. Iivari, R. Hirschheim, and H. K. Klein, "Beyond methodologies: Kepping up with information ystems development apporaches through dynamic classification," 32nd Annual Hawaii International Conference on System Sciences (HICSS-32), Maui, Hawaii, USA, ed., Maui, Hawaii, USA, pp. 1-10, 1999.
[48] N. Jayaratna, *Understanding and Evaluating Methodologies: Nimsad, a Systematic Framework*: McGraw-Hill Companies, 1994.
[49] 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.
[50] C. S. Jensen and R. T. Snodgrass, "Semantics of Time-Varying Information," *Information Systems*, vol. 21, no. 4, pp. 311-352, 1996.
[51] C. S. Jensen and R. T. Snodgrass, "Semantics of Time-Varying Attributes and their use for Temporal Database Design," TimeCenter Technical Report, Tucson, Arizona TR-1, January 29 1997.
[52] C. S. Jensen and R. T. Snodgrass, "Temporal Data Management," *Transactions of Knowledge and Data Engineering*, vol. 11, no. 1, pp. 36-44, 1999.
[53] S. Juhn and J. Naumann, "The Effectiveness of Data Representation Characteristics on User Validation," 6th International Conference on Information Systems, ed., pp. 212-226, 1985.
[54] V. Khatri, "Bridging the Spatio-Temporal Semantic Gap: A Theoretical Framework, Evaluation and A Prototype System," in *Doctoral Thesis: Management Information Systems Department*. Tucson, AZ: University of Arizona, 2002, 320.
[55] V. Khatri, S. Ram, V. Ramesh, and I. Vessey, "Can expressiveness *and* simplicity be simultaneously achieved in conceptual modeling? Exploring the role of supplementary level of abstraction for spatio-temporal conceptual modeling," Proceedings of the Third Symposium on Research in Systems Analysis and Design, St. John's, NL, Canada, ed. D. Batra, J. Parsons, and V. Ramesh, St. John's, NL, Canada, pp. 1-16, June 12-13, 2004.
[56] V. Khatri, S. Ram, and R. T. Snodgrass, "Supporting User-defined Granularities and Indeterminacy in a Spatiotemporal Conceptual Model," *Annals of Mathematics and Artificial Intelligence*, vol. 36, no. 1-2, pp. 195-232, 2002.
[57] V. Khatri, S. Ram, and R. T. Snodgrass, "Augmenting a Conceptual Model with Geospatiotemporal Annotations," *IEEE Transactions of Knowledge and Data Engineering*, no., forthcoming.
[58] W. Klein, *Time in Language*. London: Routledge, 1994.
[59] N. Kline, " An Update of the Temporal Database Bibliography," *SIGMOD Record*, vol. 22, no. 4, pp. 66-80, 1993.
[60] S. M. Kosslyn, C. F. Chabris, C. L. Marsolek, and O. Koenig, "Categorical versus coordinate spatial relations: Computational analyses and computer simulations," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 18, no. 2, pp. 562-577, 1992.
[61] G. Kösters and B.-U. Pagel, "The GeoOOA-Tool and Its Interface to Open Software Development Environments for GIS," 4th ACM workshop on Advances on Advances in Geographic Information Systems, Rockville, Maryland, ed. P. Bergougnoux and S. Shekhar, Rockville, Maryland, pp. 163-171, November 15-16, 1996.
[62] G. Kosters, B.-U. Pagel, and H.-W. Six, "Object Oriented Requirements Engineering for GIS-Applications," International Conference on ACM Geographical Information System, ed., pp. 61-68, 1995.
[63] M. Koubarakis, T. K. Sellis, A. U. Frank, S. Grumbach, R. H. Güting, C. S. Jensen, N. A. Lorentzos, Y. Manolopoulos, E. Nardelli, B. Pernici, H.-J. Schek, M. Scholl, B. Theodoulidis,

and N. Tryfona, "Spatio-Temporal Databases: The CHOROCHRONOS Approach," in *Lecture Notes in Computer Science*, vol. 2520: Springer, 2003.
[64] B. Kuipers, "Modeling Spatial Knowledge," *Cognitive Science*, vol. 2, no., pp. 129-153, 1978.
[65] Z. Kulpa, "Diagrammatic representation and reasoning," *Machine Graphics and Vision*, vol. 3, no. 1/2, pp. 77-103, 1994.
[66] C. H. Kung and A. Solvberg, "Activity modeling and behavior modeling," in *Information Systems Design Methodologies: Improving the Practice*, T. W. Olle, H. G. Sol, and A. A. Verrijn-Stuart, Eds. Amsterdam, The Netherlands: North-Holland, 1986, pp. 145-171.
[67] R. Laurini and D. Thompson, *Fundamentals of Spatial Information Systems*. London: Academic Press, 1992.
[68] U. Leonhardt, J. Kramer, B. Nuseibeh, and A. Finkelstein, "Decentralised process enactment in a multi-perspective development environment," Proceedings of the 17th international conference on Software engineering, Seattle, Washington, United States, ed., Seattle, Washington, United States, pp. 255-264, 1995.
[69] M. Mani, D. Lee, and R. R. Muntz, "Semantic Data Modeling Using XML Schemas," 20th International Conference on Conceptual Modeling, Yokohama, Japan, ed. H. S. Kunii, S. Jajodia, and A. Sølvberg, Yokohama, Japan, pp. 149-163, November 27-30, 2001.
[70] 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.
[71] I. R. McChesney and D. Glass, "Post-implementation management of CASE methodology," *European Journal of Information Systems*, vol. 2, no. 3, pp. 201-209, 1993.
[72] C. McClure, "The CASE Experience -- CASE works. But do you need a tool, a toolkit, or a workbench? And how do you begin?" *Byte*, vol. 14, no. 4, pp. 235, 1989.
[73] F. R. McFadden and J. A. Hoffer, *Database Management*, 3rd ed. Redwood City, CA: Benjamin/Cummings Publishing Company, Inc., 1991.
[74] L. E. McKenzie, "Bibliography: Temporal Databases," *SIGMOD Record*, vol. 15, no. 4, pp. 40-52, 1986.
[75] J. L. Mennis, D. J. Peuquet, and L. Qian, "A Conceptual Framework for Incorporating Cognitive Principles into Geographical Database Representation," *International Journal of Geographic Information Science*, vol. 14, no. 6, pp. 501-520, 2000.
[76] G. A. Miller, "The Magical Number Seven, Plus Minus Two: Some Limits on Our Capacity for Processing Information," *The Psychological Review*, vol. 63, no. 2, pp. 81-97, 1956.
[77] R. J. Norman and J. F. Nunamaker, "CASE productivity perceptions of software engineering professionals," *Communications of the ACM*, vol. 32, no. 9, pp. 1102-1108, 1989.
[78] OGIS, "OpenGIS Simple Feature Specification for SQL," Open GIS Consortium, Inc. 99-049, May 5 1999.
[79] A. Olerup, "Design Approaches: A Comparative Study of Information System Design and Architecutral Design.," *The Computer Journal*, vol. 34, no. 3, pp. 215-224, 1991.
[80] Oracle, "Coordinate Systems User's Guide," Oracle Inc., Release 8.1.6, April 28 2000.
[81] Oracle, "Oracle8i Documentation, Release 8.1.6," Oracle, December 1999, http://technet.oracle.com/docs/products/oracle8i/ doc_index.htm.
[82] W. J. Orlikowski, "CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development," *MIS Quarterly*, vol. 17, no. 3, pp. 309-340, 1993.
[83] C. Parent, S. Spaccapietra, and E. Zimanyi, "Spatio-temporal conceptual models: Data structures + space + time," 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, USA, ed., Kansas City, USA, pp., 1999.
[84] C. Parent, S. Spaccapietra, and E. Zimányi, "Conceptual modeling for federated geographical information systems over the Web," International Symposium on Information Systems and Technology for Network Society, Fukuoka, Japan, ed., Fukuoka, Japan, pp. 173-182, 1997.
[85] C. Parent, S. Spaccapietra, E. Zimányi, P. Donini, C. Plazanet, and C. Vangenot, "Modeling spatial data in the MADS conceptual model," 8th International Symposium on Spatial Data Handling, SDH'98, Vancouver, Canada, ed., Vancouver, Canada, pp. 138-150, 1998.

[86] J. Park, "Facilitating Interoperability among Heterogeneous Geographic Database Systems: A Theoretical Framework, A Prototype System and Evaluation," in *Management Information Systems Department Dissertation*. Tucson: University of Arizona, 1999, 507.
[87] D. Parkes and N. Thrift, *Time, spaces and places*. New York: John Wiley & Sons, 1980.
[88] M. C. Paulk, D. Goldenson, and D. M. White, "The 1999 Survey of High Maturity Organizations," CMU/SEI-2000-SR-002, http://www.sei.cmu.edu/publications/documents/00.reports/00sr002.html, 1999.
[89] G. D. Penna, A. D. Marco, B. Intrigila, I. Melatti, and A. Pierantonio, "Xere: Towards a Natural Interoperability between XML and ER Diagrams.," Proceedings of the 6th International Conference on Fundamental Approaches to Software Engineering, Warsaw, Poland, ed. M. Pezzè, Warsaw, Poland, pp. 356-371, April 7-11, 2003.
[90] D. J. Peuquet, "Making space for time: Issues in space-time data representation," *GeoInformatica*, vol. 5, no. 1, pp. 11-32, 2001.
[91] D. Pfoser and N. Tryfona, "Requirements, Definitions, and Notations for Spatiotemporal Application Environments," 6th International Symposium on Advances in Geographic Information Systems, Washington, United States, ed., Washington, United States, pp. 124-130, November 3 - 7, 1998.
[92] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, Fourth ed. Singapore: Mc-Graw Hill, 1997.
[93] Z. W. Pylyshyn, "Mental Imagery: In search of a theory," *Behavioral and Brain Sciences*, vol. 25, no. 2, pp. 157-237, 2002.
[94] S. Ram, "Intelligent Database Design using the Unifying Semantic Model," *Information and Management*, vol. 29, no. 4, pp. 191-206, 1995.
[95] S. Ram, R. T. Snodgrass, V. Khatri, and Y. Hwang, "DISTIL: A Design Support Environment for Conceptual Modeling of Spatio-Temporal Requirements," 20th International Conference on Conceptual Modeling (ER2001), Yokohama, Japan, ed. A. S. a. S. J. Hideko S. Kunii, Yokohama, Japan, pp. 70-83, November 27-30, 2001.
[96] S. Ram and V. C. Storey, "Composite and grouping: extending the realm of semantic modeling," 26th Hawaii International Conference on System Sciences (HICSS), ed., pp. 80-89, 1993.
[97] C. V. Ramamoorthy, A. Prakash, W. Tsai, and Y. Usuda, "Software Engineering: Problems and Perspectives," *Computer*, vol. 17, no. 10, pp. 191-209, 1984.
[98] P. Rigaux, M. O. Scholl, and A. Voisard, *Spatial Databases: With Application to GIS*: Morgan Kaufmann Publishers, 2001.
[99] J. F. Roddick, "A Survey of Schema Versioning Issues for Database Systems," *Information and Software Technology*, vol. 37, no. 7, pp. 383-393, 1995.
[100] 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.
[101] D. Rumelhart, P. Lindsay, and D. Norman, "A process model for long-term memory," in *Organization of Memory*, E. Turving and W. Donaldson, Eds. New York: Academic Press, 1972.
[102] E. A. Rundensteiner, A. Bic, J. P. Gilbert, and M. Yin, "Set restrictions on semantic groupings," *IEEE Transactions on Knowledge and Data Engineering*, vol. 6, no. 2, pp. 193-204, 1994.
[103] S. Sharma and A. Rai, "CASE Deployment in IS Organizations," *Communications of the ACM*, vol. 43, no. 1, pp. 80-88, 2000.
[104] S. Shekhar, M. Coyle, B. Goyal, D.-R. Liu, and S. Sarkar, "Data Models in Geographic Information Systems," *Communications of the ACM*, vol. 40, no. 4, pp. 103-111, 1997.
[105] P. Shoval and I. Frumermann, "OO and EER Conceptual Schemas: A Comparison of User Comprehension," *Journal of Database Management*, vol. 5, no. 4, pp. 28-38, 1994.
[106] A. Silbershatz, H. Korth, and S. Sudarshan, *Database System Concepts*, Third Edition ed: WCB/ McGraw Hill, 1997.
[107] J. M. Smith and D. C. P. Smith, "Database Abstractions: Aggregation and Generalization," *ACM Transactions of Database Systems*, vol. 2, no. 2, pp. 105-133, 1977.

[108] R. T. Snodgrass, "The Temporal Query Language TQuel," *ACM Transactions of Database Systems*, vol. 12, no. 2, pp. 247-298, 1987.
[109] R. T. Snodgrass, *The TSQL2 temporal query language*. Boston: Kluwer Academic Publishers, 1995.
[110] R. T. Snodgrass, *Developing Time-Oriented Database Applications in SQL*. San Francisco: Morgan Kaufmann Series in Data Management Systems, 1999.
[111] R. T. Snodgrass and I. Ahn, "Temporal Databases," *IEEE Computer*, vol. 19, no. 9, pp. 35-42, 1986.
[112] R. T. Snodgrass, M. H. Böhlen, C. S. Jensen, and A. Steiner, "Adding Transaction Time to SQL/Temporal," ISO-ANSI SQL/Temporal Change Proposal, ANSI X3H2-96-152r ISO/IEC JTC1/SC21/WG3 DBL MCI-143, 1996.
[113] R. T. Snodgrass, M. H. Böhlen, C. S. Jensen, and A. Steiner, "Adding Valid Time to SQL/Temporal," ISO-ANSI SQL/Temporal Change Proposal, ANSI X3H2-96-151r ISO/IEC JTC1/SC21/WG3 DBL MCI-142, 1996.
[114] R. T. Snodgrass, M. H. Böhlen, C. S. Jensen, and A. Steiner, "Transitioning Temporal Support in TSQL2 to SQL3," in *Temporal Databases: Research and Practice*, O. Etzion, S. Jajodia, and S. M. Sripada, Eds. New York: Springer Verlag, 1997, pp. 150-194.
[115] M. D. Soo, "Bibliography on Temporal Databases," *SIGMOD Record*, vol. 20, no. 1, pp. 14-23, 1991.
[116] A. Srinivasan and D. Te'eni, "Modeling as Constrained Problem Solving: An Empirical Study of the Data Modeling Process," *Management Science*, vol. 41, no. 3, pp. 419-434, 1995.
[117] R. B. Stam and R. T. Snodgrass, "A Bibliography on Temporal Databases," *Data Engineering Bulletin*, vol. 11, no. 4, pp. 53-61, 1988.
[118] A. Y. Tang, T. M. Adams, and E. L. Usery, "A Spatial Data Model Design for Feature-based Geographical Information Systems," *International Journal of Geographical Information Systems*, vol. 10, no. 5, pp. 643-659, 1996.
[119] M. Todd, R. L. Coleman, and J. M. Shimonek, "A CASE Perspective of System-Development: Woodmen Accident and Centel," *Journal of Systems Management*, vol. 42, no. 9, pp. 13-16, 1991.
[120] N. Tryfona and C. S. Jensen, "Conceptual Data Modeling for Spatiotemporal Applications," *Geoinformatica*, vol. 3, no. 3, pp. 245-268, 1999.
[121] N. Tryfona and C. S. Jensen, "Using Abstractions for Spatio-Temporal Conceptual Modeling," 2000 ACM Symposium on Applied Computing Applied, Como, Italy, ed., Como, Italy, pp. 313-322, March 19-21, 2000.
[122] N. Tryfona, R. Price, and C. S. Jensen, "Conceptual Models for Spatio-temporal Applications. Spatio-Temporal Databases: The CHOROCHRONOS Approach," in *Spatio-Temporal Databases: The CHOROCHRONOS Approach*, vol. 2520, *Lecture Notes in Computer Science*, M. Koubarakis, T. K. Sellis, A. U. Frank, S. Grumbach, R. H. Güting, C. S. Jensen, N. A. Lorentzos, Y. Manolopoulos, E. Nardelli, B. Pernici, H.-J. Schek, M. Scholl, B. Theodoulidis, and N. Tryfona, Eds.: Springer, 2003, pp. 79-116.
[123] V. J. Tsotras and A. Kumar, "Temporal Database Bibliography Update," *SIGMOD Record*, vol. 25, no. 1, pp. 41-51, 1996.
[124] 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.
[125] I. Vessey, S. L. Jarvenpaa, and N. Tractinsky, "Evaluation of vendor products: CASE tools as methodology companions," *Communications of the ACM*, vol. 35, no. 4, pp. 91-104, 1992.
[126] M. Wachowicz, *Object-Oriented Design for Temporal GIS*: Taylor and Francis, 1999.
[127] Y. Wand, D. E. Monarchi, J. Parsons, and C. C. Woo, "Theoretical Foundations for Conceptual Modeling in Information Systems Development," *Decision Support Systems*, vol. 15, no. 4, pp. 285-304, 1995.

[128] Y. Wand, V. C. Storey, and R. Weber, "On Ontological Analysis of the Relationship Construct in Conceptual Modeling," *ACM Transactions of Database Systems*, vol. 24, no. 4, pp. 494-528, 1999.
[129] Y. Wand and R. Weber, "Research Commentary: Information Systems and Conceptual Modeling -- A Research Agenda," *Information Systems Research*, vol. 13, no. 4, pp. 363-376, 2002.
[130] N. H. Weiderman, J. K. Bergey, D. B. Smith, and S. R. Tilley, "Approaches to Legacy System Evolution," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213 CMU/SEI-97-TR-014, http://www.sei.cmu.edu/pub/documents/97.reports/pdf/97tr014.pdf, 1997.
[131] M. F. Worboys, "A Unified Model for Spatial and Temporal Information," *The Computer Journal*, vol. 37, no. 1, pp. 26-34, 1994.
[132] M. F. Worboys, "Computation with Imprecise Geospatial Data," *Computer, Environment and Urban Systems*, vol. 22, no. 2, pp. 85-106, 1998.
[133] M. F. Worboys, "Imprecision in Finite Resolution Spatial Data," *GeoInformatica*, vol. 2, no. 3, pp. 257-279, 1998.
[134] M. F. Worboys, H. M. Hearshaw, and D. J. Maguire, "Object-oriented data modeling for spatial databases," *International Journal of Geographical Information Systems*, vol. 4, no. 4, pp. 369-383, 1990.
[135] Y. Wu, S. Jajodia, and X. S. Wang, "Temporal Database Bibliography Update," in *Temporal Databases: Research and Practice*, O. Etzion, S. Jajodia, and S. M. Sripada, Eds.: Springer, 1997, pp. 338-366.
[136] R. Xiaou, T. S. Dillon, E. Chang, and L. Feng, "Modeling and Transformation of Object-Oriented Conceptual Models into XML Schema," Proceedings of 12th International Conference of Database and Expert Systems Applications (DEXA), Munich, Germany, ed. H. C. Mayr, J. Lazanský, G. Quirchmayr, and P. Vogel, Munich, Germany, pp. 795-804, 2001.
[137] E. N. Yourdon, *Modern Structured Analysis*: Prentice Hall, 1990.
[138] J. Zhang, "The nature of external representations in problem solving," *Cognitive Science*, vol. 21, no. 2, pp. 179-217, 1997.
[139] J. Zhang, "External Representations in Complex Information Processing Tasks," in *Encyclopedia of Microcomputers*, vol. 26, A. Kent, J. G. Williams, and C. M. Hall, Eds. New York: Marcel Dekker, Inc., 2001.
[140] J. Zhang and D. A. Norman, "Representations in distributed cognitive tasks," *Cognitive Science*, vol. 18, no. 1, pp. 87-122, 1994.
[141] E. Zimanyi, C. Parent, S. Spaccapietra, and A. Pirotte, "TERC+: A Temporal Conceptual Model," International Symposium Digital Media Information Base, ed., pp., 1997.

Appendix: Annotation Syntax in BNF

⟨annotation⟩	::=	⟨temporal annotation⟩ **//** ⟨spatial annotation⟩ | ⟨temporal annotation⟩ **//** ⟨geospatial annotation⟩ **//** ⟨time-varying geospatial 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⟩+-
⟨geospatial annotation⟩	::=	ϵ | ⟨horizontal geometry⟩ / ⟨vertical geometry⟩
⟨horizontal geometry⟩	::=	⟨geometry⟩ (⟨g_{sxy}⟩) / ⟨geometry⟩ (⟨g_{sxy}⟩)
⟨vertical geometry⟩	::=	⟨geometry⟩ (⟨g_{sz}⟩) | -
⟨geometry⟩	::=	⟨point⟩ | ⟨indeterminate point⟩ | ⟨line⟩ | ⟨indeterminate line⟩ | ⟨region⟩ | ⟨indeterminate region⟩ | ⟨user defined⟩ | -
⟨point⟩	::=	P | Point
⟨indeterminate point⟩	::=	⟨point⟩~ | ⟨point⟩+-
⟨line⟩	::=	L | Line
⟨indeterminate line⟩	::=	⟨line⟩~ | ⟨line⟩+-
⟨region⟩	::=	R | Region
⟨indeterminate region⟩	::=	⟨region⟩~ | ⟨region⟩+-
⟨time-varying geospatial annotation⟩	::=	ϵ | ⟨position varying⟩ | ⟨shape varying⟩ | ⟨position varying⟩ / ⟨shape varying⟩
⟨position varying⟩	::=	⟨position⟩@⟨varying in dimension⟩
⟨shape varying⟩	::=	⟨shape⟩@⟨varying in dimension⟩
⟨position⟩	::=	Pos | Position
⟨shape⟩	::=	Sh | Shape
⟨varying in dimension⟩	::=	x | y | z | xy | yz | xz | xyz
⟨g_t⟩	::=	⟨day⟩ | ⟨hour⟩ | ⟨minute⟩ | ⟨second⟩ | ⟨user defined⟩
⟨day⟩	::=	day
⟨hour⟩	::=	hr | hour
⟨minute⟩	::=	min | minute
⟨second⟩	::=	sec | second
⟨g_{sxy}⟩	::=	⟨dms-degree⟩ | ⟨dms-minute⟩ | ⟨dms-second⟩ | ⟨user defined⟩
⟨g_{sz}⟩	::=	⟨mile⟩ | ⟨foot⟩ | ⟨user defined⟩
⟨mile⟩	::=	mile
⟨dms-degree⟩	::=	degree | deg | dms-deg | dms-degree
⟨dms-minute⟩	::=	dms-min | dms-minute
⟨dms-second⟩	::=	dms-sec | dms-second
⟨foot⟩	::=	ft | foot