DISTIL: A Design Support Environment for Conceptual Modeling of Spatio-temporal Requirements

Sudha Ram, Richard T. Snodgrass, Vijay Khatri and Yousub Hwang

University of Arizona, USA
ram@bpa.arizona.edu, rts@cs.arizona.edu,
vkhatri@bpa.arizona.edu, yhwang@bpa.arizona.edu

Abstract. We describe DISTIL (DIstributed design of SpaTIo-temporaL data), a web-based conceptual modeling prototype system that can help capture the semantics of spatio-temporal data. Via DISTIL, we describe an annotation-based approach that divides spatio-temporal conceptual design into two steps: first capture the *current reality* of an application using a conventional conceptual model *without considering the spatial aspects*, and only then annotate the schema with the spatio-temporal semantics of the application. A database development team can use DISTIL to capture and validate their spatio-temporal data requirements. Using DISTIL we demonstrate that the annotation-based approach for capturing spatio-temporal requirements is straightforward to implement, satisfies ontology-based and cognition-based requirements, and integrates seamlessly into the existing database design methodologies.

1 Introduction

A design support environment automates many of the analysts' tasks in developing software; it helps reduce time and money spent on a project and can improve the quality of the end product. Lately, geographic information is increasingly employed for a wide array of applications—land information systems, environmental modeling, resource management, transportation planning, geo-marketing, geology and archaeology. As a result, there is a need for a mechanism to capture requirements for time-varying geographically referenced data, referred to as spatio-temporal requirements. In this paper, we describe a web-based spatio-temporal conceptual modeling environment called DISTIL (DIstributed design of SpaTIo-temporaL data).

Underlying the geographic applications described above is spatio-temporal data having a *theme*—the phenomenon or object being observed, and *location* and *time* associated with the phenomenon. Via DISTIL, we describe an annotation-based approach that divides spatio-temporal conceptual design into two steps: (i) capture the *current reality* of an application using a conventional conceptual model *without considering the spatial aspects*, and only then (ii) annotate the schema with spatial and temporal semantics of the application. Rather than creating any new constructs, DISTIL uses annotations to capture temporal and spatial aspects of the application.

Our work makes several contributions. First, DISTIL is based on an annotation-based approach, which integrates and extends prior research related to spatio-temporal conceptual modeling. We describe an intuitive ontology-based grammar (Appendix)

H.S. Kunii, S. Jajodia, and A. Sølvberg (Eds.): ER 2001, LNCS 2224, pp. 70-83, 2001.

for annotation, which comprehensively captures semantics related to space and time. Our approach to spatio-temporal conceptual modeling includes a mechanism to capture spatio-temporal semantics comprehensively, e.g., *event* and *state*, *valid time*, *existence time* and *transaction time*, various types of geometries associated with spatiality, user-defined *granularities*, *indeterminacy*, and *topological relationships*. Second, DISTIL demonstrates a comprehensive methodology for spatio-temporal conceptual modeling that can be applied to any traditional conceptual model, thereby translating it to a spatio-temporal conceptual model. We show how our proposed approach integrates with conventional conceptual design methodologies. Third, via a spatio-temporal application we demonstrate that our approach to spatio-temporal database design using DISTIL is practical and is straightforward to use. Fourth, DISTIL provides an integrated modeling environment that can be used by a distributed team of users and data analysts; it takes advantage of interconnectivity made possible by the Internet. Fifth, our design tool environment parallels human perception for spatio-temporal data. Mennis et al. [10] cite that humans cognitively store *what*, *where* and *when* data in separate knowledge structures. DISTIL with its two-step approach that first focuses on facts (*what*) and then on the context (*where* and *when*) related to the facts corresponds to the human perception of spatio-temporal data.

In summary, we present a web-based spatio-temporal database design environment that is intuitive from users' point of view, is straightforward to implement from CASE (Computer-Aided Software/System Engineering) tool vendors' perspective, comprehensively captures semantics related to space and time in a way that parallels human cognition, and integrates into existing database design methodologies.

The rest of the paper is organized as follows. In section 2, we motivate the need for an automated spatio-temporal database design tool using a hydrogeologic application at the United States Geological Survey (USGS). We outline the spatio-temporal ontology in section 3, which is the basis for annotations in DISTIL. In section 4, we describe how DISTIL supports spatio-temporal modeling using the example of section 2. In section 5, we present the architecture of DISTIL that supports spatio-temporal design described in the previous section. Finally, comparison with related work and future directions of our research round out the paper.

2 Motivation

Conceptual database design is widely recognized as an important step in the development of database applications. As a result of conceptual design, a conceptual schema is developed that captures database requirements of the users. This schema is very useful in the event of technology upgrades and transfer, and acts as a communication vehicle between different groups of end-users and the data analyst.

We are designing a database for a group of researchers at USGS who are developing a ground-water flow model for Death Valley [4]. 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 predictions based on the model are dependent on the data that form an input to this model.

A large part of the input data for this model is geo-referenced and temporal in nature. For example, two key objects of interest for the application are spring-water sites and borehole sites, which are spatial in nature. A spring-water site is a spring represented as a point on the surface of the earth whose location is given by geographic *x*- and *y*-coordinates, with spatial granularity of degree. A borehole site refers to a part of the well whose 3D location is given by the *x*- and *y*-coordinates on the earth's surface with spatial granularity of degree, and depth below land surface with spatial granularity of foot; there can be different borehole sites at different depths at the same surface location. Other input data for the model includes discharge at the spring site and water level at the borehole site, both of which are collected by source agencies. Additionally, there are various hydraulic tests conducted at the borehole site. Borehole sites may include a pump that removes water from the borehole site; this can impact other data collected at the borehole site.

Capturing requirements related to spring-water sites, borehole sites, source agencies and pump lifts has many associated issues and a design tool developed to capture these user requirements should: (i) be based on a comprehensive methodology that adequately captures the semantics related to space and time; (ii) be easy to use from the data analysts' point of view, i.e., include a formalism that is intuitive and that provides spatio-temporal modeling support as an extension from their existing database design paradigm; (iii) incorporate a metaphor that bridges the conceptual gap between a computer system and how human beings perceive the real world; (iv) be straightforward to implement from the perspective of a CASE tool vendor; and (v) support geographically distributed users and data analysts. Having elucidated the spatio-temporal requirements, we briefly describe the spatio-temporal ontology that is the basis for spatio-temporal semantic concepts embedded in DISTIL.

3 Ontology

A *time domain* is denoted by the pair $(T, \leq)$, where T is a nonempty set of *time instants* and "≤" is the total order on T; e.g., $(\mathbf{Z}, \leq)$ represents a discrete time domain, where the instants are isomorphic to integers. There are two kinds of facts associated with time, *events* and *states*. An event occurs at a point of time, i.e., an event has no duration. A state, on the other hand, has duration, e.g., a storm occurred from 5:07 PM to 5:46 PM. Facts can interact with time in two orthogonal ways [14] resulting in *transaction time* and *valid time*. Valid time denotes when the fact is true in the real world. On the other hand, transaction time links an object to the time it is current in the database. *Existence time*, which applies to objects, is the valid time when an object exists; it is also referred to as a *lifespan* of an entity. *Temporal granularity* is an integral part of temporal data [2] and refers to the measure of temporal datum. Temporal granularities in the same time domain can be related by *coarser than* and *finer than* relationships. For many applications, it is known only approximately when the event occurred and this is referred to as *valid-time indeterminacy* [6]. While the temporal granularity can be specified for existence time and valid time, that for transaction time is system-defined.

We can view the Earth as a spheroid in three-dimensional space; let us assume that $\mathbf{Z}^3$ represents the space domain. A spatial object is associated with *geometry* and *position*. Geometry represents the shape and size of an object [5]. The position in space is based on co-ordinates in a mathematically defined reference system, e.g., latitude and longitude. Geometry of the spatial object may be 0-, 1- or 2- dimensional corresponding to a point, a line and a region. Like temporal granularity, spatial granularity can be defined for horizontal and vertical dimension [8]; e.g., the granularity of the *x*- and *y*-dimension can be in degree and that of the *z*-dimension may be foot.

Three types of interaction between an object and space-time are possible [12, 17]: (i) moving objects, i.e., position changes but the geometry does not (e.g., car moving on a road network); (ii) objects whose characteristics and position change with time, i.e., change of position by changing geometry (e.g., change of the position of land parcels in a cadastral application); (iii) integration of the above two behaviors, i.e., moving and changing phenomena (e.g., modeling a storm).

In the following sections, we describe how we have incorporated these ontological concepts into DISTIL.

4 Spatio-temporal Conceptual Design Using DISTIL

In this section, we illustrate how a typical data analyst interacts with DISTIL, using the example scenario described in section 2. We highlight how DISTIL applies the two-step annotation-based approach to capture the users' spatio-temporal requirements.

4.1 Developing a Core Schema

DISTIL uses USM [13], an extended ER Model [3], as the base model for conventional conceptual design; however, the spatio-temporal modeling methodology exemplified with DISTIL is not specific to USM and can be applied to any conventional conceptual model. As a first step to spatio-temporal design using DISTIL, a data analyst develops a conceptual schema without considering the spatial and temporal aspects of the application; we refer to this as the *core USM schema*. We summarize key terms and terminology related to conceptual modeling, specifically for USM. Next, we show how a core USM schema can be developed using DISTIL.

A real world object is referred to as an *entity*. A collection of entities for which common characteristics are to be modeled is called an *entity class* (or simply *class*). Characteristics or properties of an entity are referred to as *attributes* of the entity class. An *interaction relationship* refers members of one entity class to members of one or more entity classes. The reader is referred to [13] for details related to *weak entity class*, *superclass/subclass*, *composite class* and *grouping class*.

Fig. 1 shows an example of a core USM Schema developed by a data analyst. This schema includes entity classes like SPRING_SITE, BORE_HOLE_SITE, SOURCE_AGENCY, and PUMPLIFT and their various attributes.

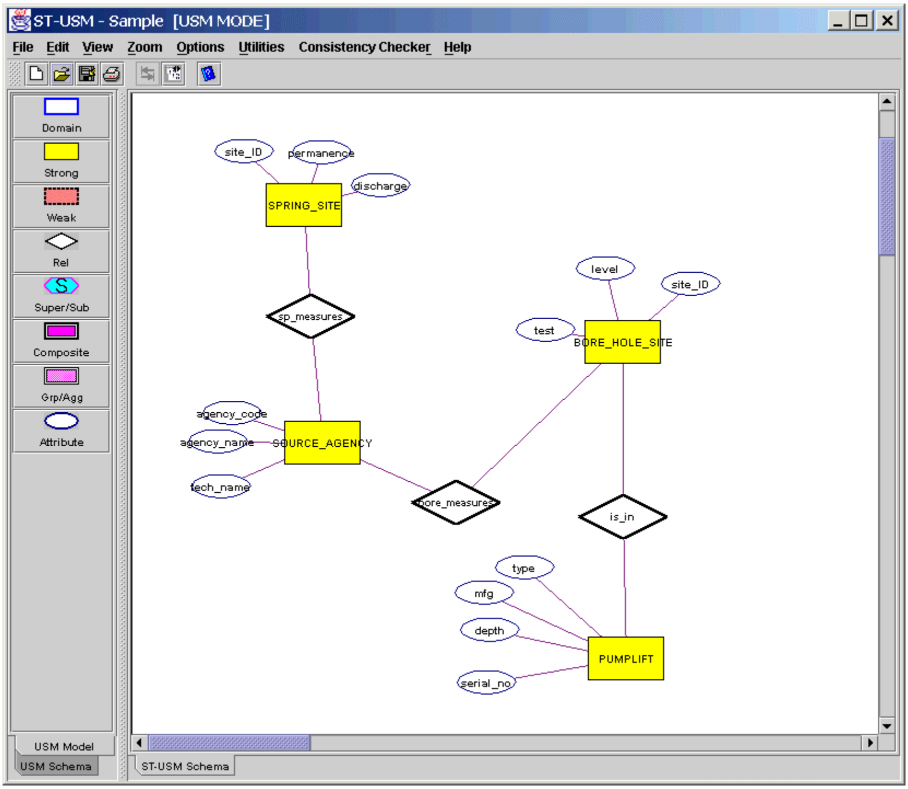


Fig. 1. Developing a schema using USM constructs

These entity classes are created by using the constructs on the "USM Model" panel on the left side. For example, the relationship sp_measures between SPRING_SITE and SOURCE_AGENCY relates an entity of SPRING_SITE with that of SOURCE_AGENCY. Each entity class has associated properties; e.g., PUMPLIFT has properties like type and manufacturer (mfg).

4.2 Annotating the Core Schema

After having developed a core USM schema without considering spatial and temporal aspects, the data analyst next annotates the spatial and temporal aspects of the application; we refer to the schema with annotations as an *Annotated Schema*, or an *ST-USM Schema*. Our annotation-based approach integrates the semantics related to space and time into a traditional conceptual model without adding any special constructs. Since various types of conceptual modeling abstractions (i.e., *entity class*, *attribute*, *relationship*, *subclass*, *composite* and *grouping*) are orthogonal to space and time, the annotations are generic and applicable to all types of conceptual modeling abstractions. We first describe the syntax for annotating spatial and temporal aspects of the application. Next, we show how DISTIL provides a convenient way of automatically producing these short annotation phrases, which are displayed in the ST-USM schema.

As shown in the Appendix, the overall structure of spatio-temporal annotation phrase is:

⟨temporal annotation⟩ // ⟨spatial annotation⟩ // ⟨spatiotemporal annotation⟩

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

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 a "-". 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(day)/-//" succinctly denotes that entities in the entity class PUMPLIFT have an associated existence time with a temporal granularity of day (day) and the lifespan of entities is modeled as states (S); no transaction time is recorded for such entities. Similarly, "E (min)/T//" denotes that entities in the entity class exist in a bitemporal space. The temporal granularity of the event (E) is minute (min). Additionally, we also need to capture transaction time (T) associated with the entities. In this second example, the granularity associated with transaction time is not specified, as it is system-defined.

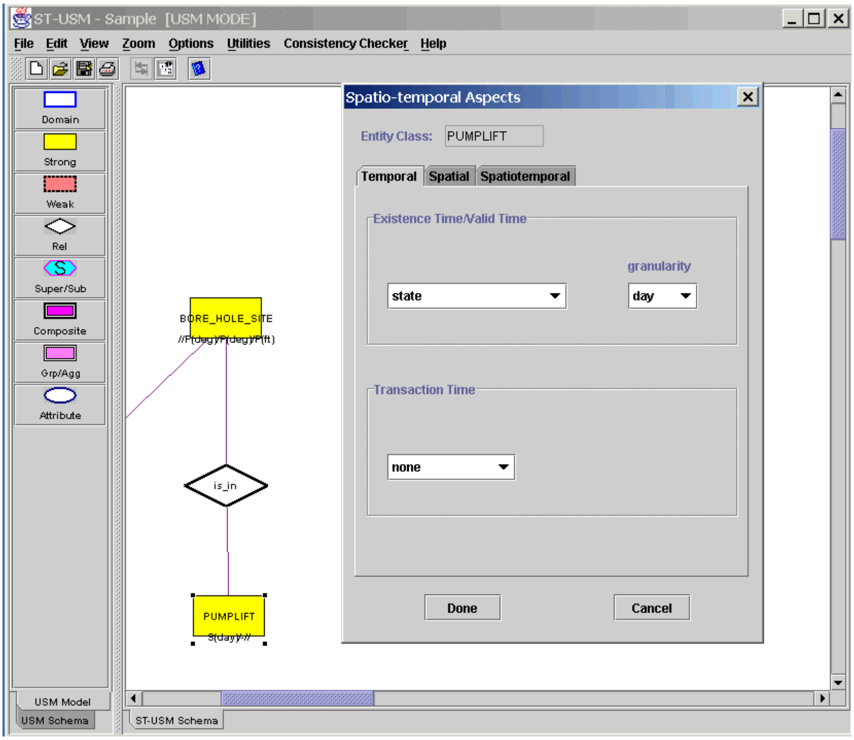
The spatial annotation includes geometry and position in *x*-, *y*- and *z*-dimension, each dimension segregated by a forward slash (/). For example, "//P(deg)/P(deg)/-" for SPRING_SITE describes a spatial entity with a geometry of points in the *x*-*y* plane. The associated spatial granularity is degree.

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 spatiotemporal annotation can be specified only if spatial and temporal annotations have already been specified. For example, "E(sec)/-//P(deg)/P(deg)/-//Pos@xy" denotes that entities in the entity class have only time-varying position while the shape is time-invariant. The geometry of the entities is a point (P) in an *x*-*y* plane with a spatial granularity of degree. The position of the entity changes in the *x*-*y* plane (Pos@xy) over time and each geometry is valid for time granules (E) measured in second. Our annotation also includes a formalism to model indeterminacy [8]; e.g., an indeterminate state with a probability distribution function [6] is designated as S~. Many times the probability distribution may not be known and a user may make a simplified assumption of a uniform distribution and in that case an indeterminate state is represented as S+-.

Having described the annotations, we next describe how the data analyst can specify annotations in DISTIL.

For each construct in the core USM schema (e.g., Fig. 2) the data analyst together with the users consider whether temporality and spatiality is 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 the objects? What is the geographical shape of the objects? What is the associated spatial granularity? Can the spatial shape/position for these objects change over time? Accordingly, the data analyst enters the details using a pop-up box as shown in Fig. 2.



Entity Class: PUMPLIFT

Existence Time/Valid Time: state

granularity: day

Transaction Time: none

Fig. 2. Specifying the temporal aspects on the core USM schema

Fig. 2 shows that the data analyst enters temporal details which would result in an annotation “S(day)/-//” for PUMPLIFT. Similarly, the data analyst enters spatial details which would result in an annotation “//P(deg)/P(deg)/L(ft)” for BORE_HOLE_SITE. This implies that the BORE_HOLE_SITE needs to be represented as a point (P) on the *x*-*y* plane and a line (L) on the *z*-dimension. Additionally, the associated granularity in the *x*-*y* plane is degree (deg) and that in the *z*-dimension is foot (ft).

The data analyst also annotates SPRING_SITE resulting in the annotated schema shown in Fig. 3. The schema is automatically annotated depending on the information filled in the pop-up box. The data analyst can annotate each class, interaction relationship and attribute. Once the data analyst has made the annotated schema, the requirements so collected can be validated with other users. The distributed users may open the same project and make the necessary changes to the schema.

At this point, the data analyst can check if the Annotated Schema is consistent using “Consistency Checker”. For example, if a spatial interaction relationship were specified between non-spatial entity classes, the “Consistency Checker” would give an error message.

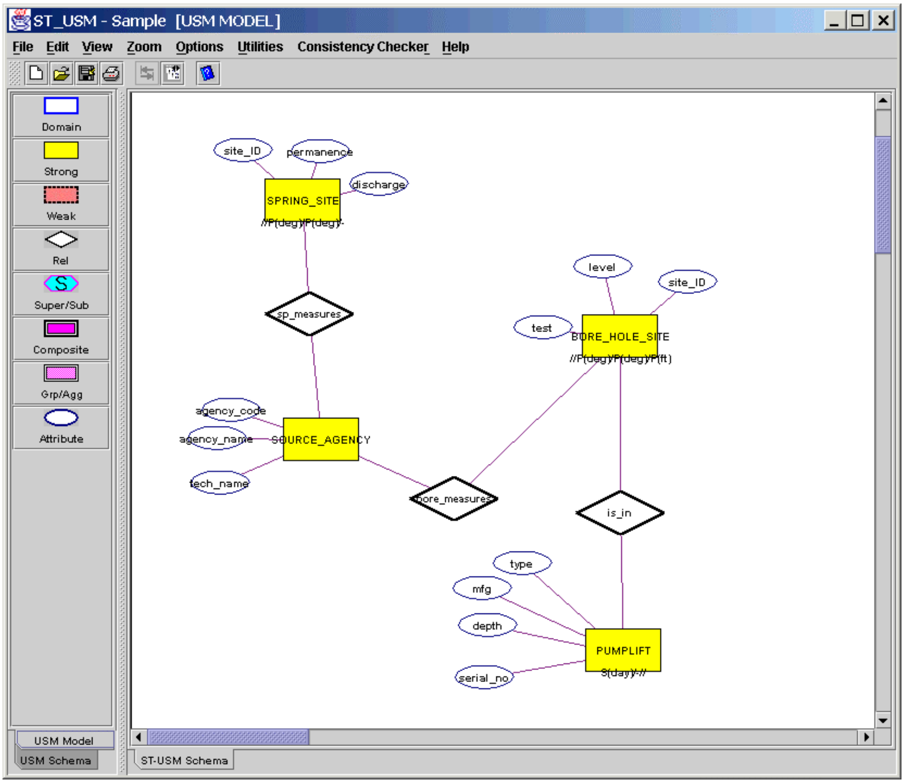


Fig. 3. Annotated schema (ST-USM Schema)

4.3 Semantics of the Annotated Schema

The Annotated Schema described in the previous sub-section can be used as communication vehicle between users and data analysts. We now explicate how data analysts can obtain explicit spatial and temporal semantics from the annotated schema; we refer to the schema so obtained as the *Translated USM Schema*. We use an example of a temporal entity class to briefly describe the semantics of an annotated ST-USM construct via a formal mapping to the USM schema. As USM constructs have been carefully defined, this mapping provides a comprehensive semantics for the richer ST-USM schema. We then show how these mapping rules have been embedded in DISTIL.

The existence time of a temporal entity class refers to the lifespan of the entity and defines the time when the facts associated with the entity can be true. Fig. 4 shows a temporal entity class for which we want to capture the existence time expressed as state (S) with ⟨g_{et}⟩ as the granularity name (e.g., day). Based on the users' requirements, the data analyst simply annotates ⟨ENTITY_CLASS⟩ as "S(⟨g_{et}⟩)/-//"

and does not need to contend with the complexity of the underlying semantics or the associated constraints.

Fig. 4 shows the semantics of a simple temporal entity class using USM constructs. In order to express the semantics of a temporal entity class, we need to specify a TEMPORAL_GRANULARITY in which the evolution of the temporal object is embedded. The relationship 〈ENTITY_CLASS〉_has_ET associates an entity with a corresponding TEMPORAL_GRANULARITY. Each TEMPORAL_GRANULARITY is uniquely identified by a granularity_name. The relationship anchor_gran helps create a granularity lattice, which can help a user choose the level of detail associated with facts.

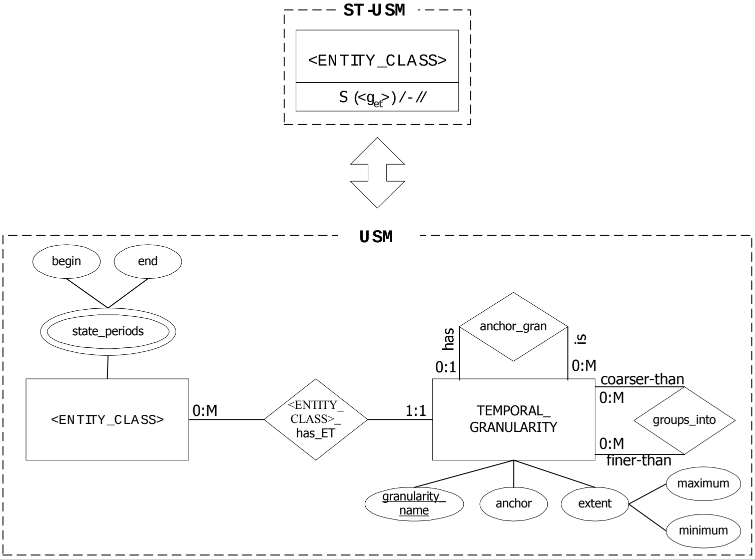


Fig. 4. Temporal Entity Class in ST-USM and its semantics in USM

Each anchor_gran is a recursive relationship (i.e., a relationship where an entity from the same entity set can play different roles) such that each participating granularity optionally has an anchor and each granularity is an anchor for 0-to-many (0:M) other granularities. A finer-than and a coarser-than relationship between granularities is 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 translation from the ST-USM schema to the USM schema also includes constraints, which are implicit in the ST-USM schema but are explicit in the USM schema. An example of such a constraint is that a temporal element formed by the state_periods is well formed, i.e., it is a union of non-overlapping intervals. Details of the semantics for annotated constructs and the associated constraints are outside the scope of this paper and are defined formally elsewhere [7, 8].

We now describe how these semantic mapping rules have been embedded in DISTIL. To view detailed explicit semantics associated with the annotated schema, the data analyst clicks on the "USM Schema" tab to obtain a translated USM schema corresponding to the ST-USM schema (Fig. 5). We have embedded translation rules (e.g., Fig. 4) into DISTIL, which help translate the annotated constructs to an equivalent USM schema with explicit representation of the associated spatiality and temporality. 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.

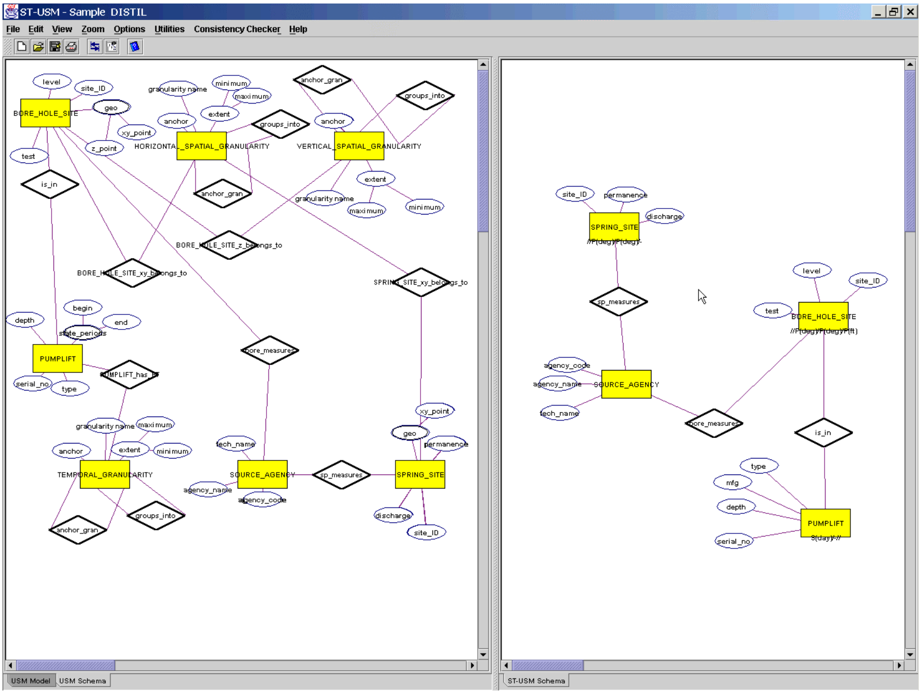


Fig. 5. Annotated Schema and its semantics using USM

In this section, we described a typical data analyst's interaction with DISTIL. The ST-USM schema can be used as communication vehicle. It can also be used to decide if all the spatio-temporal requirements of the user have been captured and whether the requirements are conflicting. On the other hand, the translated USM schema can be used by the data analyst to subsequently develop the logical schema.

5 Architecture

In this section, we describe the underlying architecture of DISTIL that enables the interaction described in section 4. DISTIL has been implemented using Java 2 (JDK 1.2) and Oracle 8.1.6. The prototype system is currently running on NT workstations. DISTIL can be accessed through a web-browser or be used as a stand-alone application.

As shown in Fig. 1, 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., Fig. 1) that captures current reality without considering spatial aspects. Our *Spatio-temporal Conceptual Design* includes annotating the Core USM Schema via *Annotation Designer* (e.g., Fig. 2) resulting in the *ST-USM Schema* (e.g., Fig. 3), mapping the ST-USM Schema to a *Translated USM Schema* (e.g., Fig. 5) through the *Semantic Mapper* and checking the consistency (i.e., *Semantic Error Log*) in the *ST-USM Schema* via *Consistency Checker*.

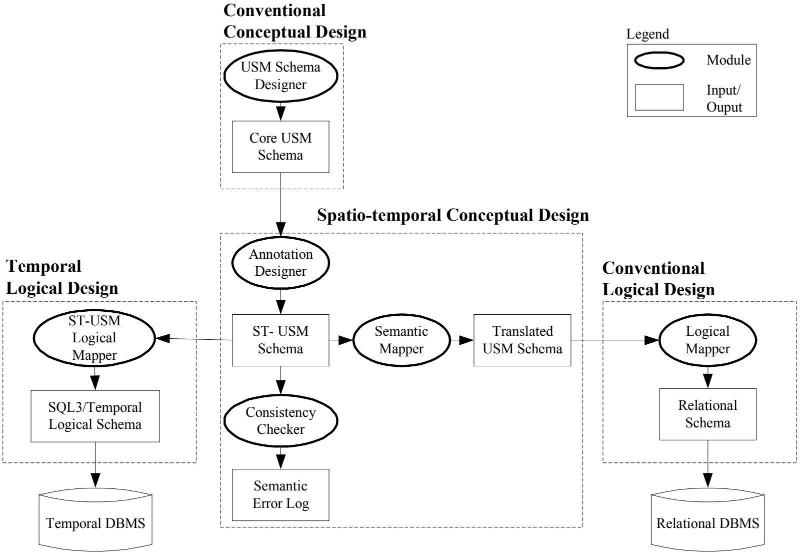


Fig. 6. DISTIL Architecture

As shown in Fig. 6, our proposed Spatio-temporal Conceptual Design implemented via DISTIL integrates with Conventional Conceptual Design and the translated USM Schema (including constraints) 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 spatial support that can be implemented in a relational DBMS (e.g., Oracle Spatial). We also envision incorporating *ST-USM Logical Mapper*, which would translate the ST-USM schema to an *SQL3/Temporal Logical Schema* [15, 16] that can be implemented in a temporal DBMS.

6 Related Work

With the growth of geographic applications, recently design tools [1, 9, 11] to support modeling of spatio-temporal databases have been proposed. Perceptory [1] and GeoOOA [9] focus on capturing the semantics associated with geometry of the spatial objects; spatiality of entities is defined by a relationship link with geometry, e.g., point, line and region. A visual schema editor based on MADS [11] helps capturing some spatio-temporal semantics. However, none of these tools provide a mechanism to capture semantics related to granularity and indeterminacy. Granularities related to facts needs to be captured during conceptual design because under-specifying granularities can restrict an application, affect relative ordering of events, and impact topological relationships. Indeterminacy, i.e., don't know *exactly* where/when information, is important for many geographic applications. Because these tools do not demonstrate a generic spatio-temporal database design methodology that integrates naturally with conventional conceptual design, methodologies proposed via these tools cannot be easily adopted by existing CASE tools that support conventional conceptual design.

Our approach to spatio-temporal conceptual modeling is comprehensive and captures various aspects related to temporality and spatiality, e.g., valid time, transaction time, events, states, position, geometry, shape, granularities and indeterminacy. Additionally, our proposed annotation-based approach used in DISTIL integrates into existing database design methodologies. Annotating the schema is intuitive from the perspective of data analysts and users as it corresponds to the way human beings perceive spatio-temporal objects. Moreover, incorporating annotation—via pop-up boxes—into an existing CASE tool is also straightforward to implement.

7 Summary and Future Directions

With advances in technologies like high-resolution satellite-borne imaging systems and global positioning systems, geographic data is finding its way into many traditional applications. In this paper, we described a design support environment for modeling the users' spatio-temporal requirements that integrates with conventional conceptual design.

We are working on completing various aspects of spatio-temporal conceptual design support provided by DISTIL. We are embedding translation rules associated with complex abstractions, e.g., grouping and composite, and constraints including topological relationships. In future, we plan to conduct more case studies in different application domains and evaluate the design support provided by DISTIL.

We believe that with our annotation-based approach implemented using DISTIL, we have achieved both comprehensiveness and simplicity in spatio-temporal conceptual modeling.

References

[1] Y. Bedard, "Visual Modeling of Spatial Databases: Towards Spatial PVL and UML," *Geomatica*, Vol. 53, No. 2, pp. 169-186, 1999.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

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

[8] 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* (Also available as TIMECENTER Technical Report TR-55), 36 pages, forthcoming.

[9] G. Kösters and B.-U. Pagel, "The GeoOOA-Tool and Its Interface to Open Software Development Environments for GIS," in Proceedings of the fourth ACM workshop on Advances on Advances in Geographic Information Systems, Rockville, Maryland, pp. 163-171, 1996.

[10] 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.

[11] C. Parent, S. Spaccapietra, and E. Zimanyi, "Spatio-temporal conceptual models: Data structures + space + time," in Proceedings of the 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, USA, pp. 1999.

[12] D. Pfoser and N. Tryfona, "Requirements, Definitions, and Notations for Spatiotemporal Application Environments," in Proceedings of the 6th International Symposium on Advances in Geographic Information Systems, Washington, United States, pp. 124-130, 1998.

[13] S. Ram, "Intelligent Database Design using the Unifying Semantic Model," *Information and Management*, Vol. 29, No. 4, pp. 191-206, 1995.

[14] R. T. Snodgrass and I. Ahn, "Temporal Databases," *IEEE Computer*, Vol. 19, No. 9, pp. 35-42, 1986.

[15] 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.

[16] 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.

[17] N. Tryfona and C. S. Jensen, "Conceptual Data Modeling for Spatiotemporal Applications," *Geoinformatica*, Vol. 3, No. 3, pp. 245-268, 1999.

Appendix: Annotation Syntax Using BNF

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