LNCS 2520, pp79-116, October 2003.
(URL: http://www.springerlink.com/link.asp?id=l3g1etcemxenpa2a)

3 Conceptual Models for Spatio-temporal Applications

Nectaria Tryfona[1], Rosanne Price[2], and Christian S. Jensen[1]

[1] Aalborg University, Denmark
[2] Monash University, Caulfield East, Australia

3.1 Motivation

Improved support for modeling information systems involving time-varying, georeferenced information, termed spatio-temporal information, has been a long-term user requirement in a variety of areas, such as cadastral systems that capture the histories of landparcels, routing systems computing possible routes of vehicles, and weather forecasting systems. This chapter concerns the conceptual database design phase for such spatio-temporal information systems and presents two models, namely the spatio-temporal Entity Relationship (ER) Model and the Extended spatio-temporal Unified Modeling Language (UML) as proposed in [33,34] and [26], respectively.

The conceptual design phase focuses on expressing application requirements without the use of computer metaphors. The design should be understandable to the user and complete, so that it can be translated into the logical phase that follows without any further user input. Popular conceptual models include the ER model [6], IFO [2], OMT [30], and UML [17].

For conventional administrative systems, exemplified by the "supplier-supplies-parts" paradigm, the available modeling notations and techniques that support the conceptual and logical modeling phases are mature and adequate. However, this is not the case in non-standard systems managing spatio-temporal, multimedia, VLSI, image, and voice data. Rather, these lead to new and unmet requirements for modeling techniques.

The basic rationale behind the work presented here is to introduce new modeling techniques, based on minimal extensions of existing models, developed to accommodate the peculiarities of the combined spatial and temporal information. The ER Model and the UML have been extended as prototypical examples for this purpose.

First, we present the fundamental aspects of the spatio-temporal domain, covering concepts such as objects, properties, and relationships. Based on these, in the ER approach, we present a small set of constructs aimed at improving the ability to conveniently model spatio-temporal information at the conceptual level. These constructs may be included in a wide range of existing conceptual data models, improving their modeling capabilities without fundamentally changing the models. We incorporate the proposed modeling constructs into the ER model ([5]), resulting in the semantically richer Spatio-Temporal ER (STER) model [33].

T. Sellis et al. (Eds.): Spatio-temporal Databases, LNCS 2520, pp. 79–116, 2003.

In the UML approach, an extension of UML, Extended spatio-temporal UML [26], is presented that addresses spatio-temporal modeling requirements. Extending the Object Management Group standard for object-oriented (OO) modeling was selected as the best approach, given UML's high level of acceptance, tool support, understandability, and extensibility. In order to satisfy the requirement for a clear, simple, and consistent notation, the extension introduces a small base set of modeling constructs for spatio-temporal data. These can be combined and applied to attributes, objects and associations in a consistent manner, guided by the same simple rules.

3.2 Spatio-temporal Foundations

In this section we discuss fundamental spatial and temporal aspects that should be considered for the conceptual modeling of spatio-temporal applications. Later on, we show the enhancement of the ER model and UML with constructs to support these concepts.

Objects, Properties, and Relationships. We perceive reality as a collection of objects, characterized by a set of properties. Objects are interrelated via relationships.

Spatial Aspects. Most real-world objects have a *position* or a *spatial extent*. In a spatio-temporal application, the positions of some objects *matter* and should be recorded, while the positions of other objects are not to be recorded. The former objects, we term *spatial*, or *geographic objects* (GO).

The function p (position) takes spatial objects as arguments and returns the positions of the objects. Positions are parts of space and may be points, lines, regions, or combinations thereof, and are called *geometric figures*. So, function p is defined as follows.

$$p : GO \to G$$

where G is the domain of geometric figures.

The embedding *space* must also be modeled in order to locate the objects in it. Capturing space is a rather complicated issue. Many researchers see space as what human beings experience [32] or see it as the description of reality based on geometric factors [21].

Philosophical discussions aside, we model space as a set, and we term the elements of space (*points*). Many different sets will do for space, but for practical reasons, space is modeled as a subset of R^2 or R^3 in current spatial applications and we use R^2 as our space; this choice does not affect the generality of the proposed approach.

It is also an inherent property of spatial objects that their positions may be viewed at different granularities and that the granularity affects the concrete data type of the position. For example, a "landparcel" may be seen as a point, a region, or both, depending on the granularity requirements of the application at hand. Such different *object views* have to be integrated into one conceptual description.

Spatial objects have *descriptive properties*, such as the "owner's name" or the "cadastral-id" of a landparcel, and *spatial properties*, such as the "soil type" of a landparcel. Spatial properties are properties of the embedding space that indirectly become properties of the spatial objects via their position in space, i.e., the spatial objects inherit them from space. For example, although one application may view the "soil type" of a landparcel as a property of the landparcel, it is clear that: (a) the "soil type" is defined whether or not the landparcel exists at that position in space, and (b) when the landparcel moves (or changes shape), the landparcel's "soil type" will not remain unchanged; rather the "soil type" attribute inherits (or, obtains) new values from the new position.

The spatial properties of objects may be captured independently of the objects using so-called fields (the term *layer* is also used). Formally speaking, a field can be seen as a function from geometric figures to a domain of descriptive attribute values [8].

$$f_1 : G \rightarrow D_1 \times D_2 \times \ldots \times D_k$$

where G is the set of geometric figures and the D_i are (not necessarily distinct) domains. In other words, a field is a set of geometric figures with associated values.

There are two basic types of fields:

(a) those that are continuous functions, e.g., "temperature," or "erosion," and
(b) those that are discrete functions, e.g., "county divisions" represented as regions.

In case (a), we visualize a field as a homogeneous (or continuous) area consisting of points, while in case (b), a field represents a set of areas with different values of the same attribute or positions of objects in space.

Finally, geographic objects may be related to each other in *space* via spatial (or *geographic*) *relationships*. For example, "the fjord Limfjorden *traverses* the city of Aalborg." Spatial relationships among geographic objects are actually relations on the objects' positions.

The set of spatial relationships is subdivided into three subsets: *topological* (e.g., "inside," "outside," etc.), *directional* (e.g., "North of," "North-East of," etc.), and *metric* (e.g., "5 km away from") relationships [10,14,15]. Spatial relationships are further translated into spatial integrity constraints on the database.

Temporal Aspects. Information about objects' properties and relationships among objects can be considered as *statements*, or *facts*, about the objects.

For example, an application involving countries may include a "capital" property for the countries. The "Copenhagen" value of the property "capital" associated with "country" Denmark denotes the fact that "Copenhagen is the capital of Denmark." Precisely everything that can be assigned a truth value is a fact. For example, "Denmark is south of Greece" is a fact; it can be assigned the truth value "false." The sentence "Denmark and Greece are south" is not a fact.

Three temporal aspects have been the focus of attention in the research literature; they are universal, and applications frequently require that these be captured in the database:

1. The *valid time* aspect applies to facts: the valid time of a fact is the time when the fact is true in the modeled reality. For example, the valid time of "Copenhagen is the capital of Denmark" is the time from year 1445 until the present day.
2. The *transaction time* aspect applies not only to facts, but to any "element" that may be stored in a database: the transaction time of a database element is the time when the element is part of the current state of the database. Put differently, the transaction time of element e is the valid time of "e is current in the database." Transaction time is important in applications that demand traceability and accountability.
3. The *existence time* aspect applies to objects: the existence time of an object is the time when the object exists. Again, this aspect can be formulated in terms of valid time. The existence time of object o is thus the valid time of "o exists."

We will assume that it only makes sense for an object to have properties and participate in relationships *when the object exists*. This implies that the valid times of facts associated with objects must be contained in the existence times for those objects.

Time values are drawn from a domain of time values, with the individual values being termed *chronons*. All three temporal aspects have duration, and they may be captured using time intervals, where a time interval $[t_s, t_e]$ is defined to be a set of consecutive chronons. We call t_s and t_e the start and the end chronon of the interval, respectively.

The following section extends the ER model in accordance with the foundations outlined here.

3.3 Spatio-temporal Entity-Relationship Model

In this section we present an extension of the ER model, namely the Spatio-temporal ER model (STER), to accommodate spatio-temporal peculiarities [33,34]. STER combines spatial and temporal aspects in a meaningful way. The following section serves to explicitly state which aspects are independent and, by implication, which are not. It thus provides a guide for applying the general design criteria.

3.3.1 Extending the ER with Spatio-temporal Constructs

The Entity Relationship (ER) model [6] is arguably the first conceptual model that appeared in the literature. This easy-to-use model, consisting of very few modeling constructs, has gained an unparalleled, widespread popularity in industry. The model's basic constructs include the following: (a) entity sets that represent objects are depicted by rectangles; (b) relationship sets that represent associations among entity sets are illustrated as diamonds; and (c) attributes of entity sets and relationship sets that capture properties of the objects and associations and are represented graphically as ovals. Relationships can be 1:1 (one to one), 1:M (one to many), and N:M (many to many). Furthermore, in some extensions of ER (reference [12] is both recent and accessible) a very useful, special kind of relationship, the ISA relationship, is proposed to model an entity set as a subset of another.

The Spatio-Temporal Entity Relationship Model (STER) [33] includes constructs with built-in spatial, temporal, and spatio-temporal functionality. A construct that captures a temporal aspect is called *temporal*; if it has built-in support only for a spatial aspect, it is termed *spatial*; and if it has both, it is *spatio-temporal*. The upper-right corner of each extended construct indicates its temporal support. The bottom-right corner indicates the spatial support. For each STER construct, we give its corresponding representation in the ER model.

While all basic constructs of the ER model can have spatial and temporal extents, not all temporal aspects are semantically meaningful for each construct. The aspect termed existence time when "something" is considered to exist. So existence time is applicable precisely to the entities in entity sets, which are the only elements in the ER model with independent existence.

An entity set may be given attributes that describe the properties of the set's entities. Earlier, we stated that valid time is meaningful only for *facts*. When assigning valid time to an attribute of an entity set, we indicate that the valid times of the facts—that specific entities in the set are associated with specific values for this attribute—are to be captured in the database. The same applies to attributes of relationship sets in place of entity sets. Finally, valid time may be assigned to a relationship set, indicating that the time when each relation in the set is true in the miniworld that is to be captured in the database.

Transaction time applies to any "element" stored in the database, regardless of whether or not it may be assigned a *truth* value. So unlike valid time, transaction time applies to entities in entity sets. Table 3.1 shows the meaningful combinations of temporal aspects and modeling constructs.

Table 3.1. Assigning temporal aspects to ER constructs

	entity set	attributes	relationship
existence time	Yes	No	No
valid time	No	Yes	Yes
transaction time	Yes	Yes	Yes

As the next step, we illustrate in more detail how to assign existence and valid time to the ER constructs. Transaction time is covered separately, in the next section. The abbreviations "et," "vt," "tt," and "bt" are used for "existence time," "valid time," "transaction time," and "bitemporal time," respectively. Abbreviation "bt" is a shorthand for the combination of "vt" and "tt" that occurs often in a spatio-temporal database.

Entity Sets. Entity sets represent objects.

(i) Temporal Entity Sets. Entities in an entity set can be assigned existence and transaction time. We term the former support for existence time, and this is indicated by placing an "et" in a circle in the upper-right corner of the entity set's rectangle as indicated in Figure 3.1. Figure 3.1(b) shows that for "car" entities, we keep track of their existence time. This notation is in effect shorthand for a

Fig. 3.1. (a) Capturing existence time, and (b) representing car entities and their existence times

larger ER diagram. This shorthand is convenient because it concisely states that the existence times of the entities in the entity set should be captured in the database. The more verbose ER diagram corresponding to the STER diagram in Figure 3.1(a) is given in Figure 3.2. Attributes connected to each other denote composite attributes, i.e., attributes resulting of the combination of other attributes [12]. Thus "existence time/id" values consist of pairs of "existence time" and "id" (i.e., identification number) values.

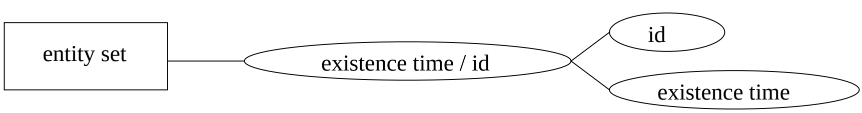

Fig. 3.2. ER diagram corresponding to Figure 3.1(a)

(ii) Spatial Entity Sets. Spatial objects have a position in space, and it is frequently necessary to capture this position in the database. The first step to support this is to provide means for representing the space in which the objects are embedded. The next step is to provide means for indicating that the objects' positions in this space are to be captured. For these purposes, we introduce the following special entity and relationship sets.

1. The special entity sets SPACE, GEOMETRY, POINT (or "P"), LINE (or "L"), REGION (or "R"). Entity set GEOMETRY is used for capturing

the shapes of entities and can be specialized as (i.e., is-a) POINT, LINE, REGION, or any other geometric type (or geometry). For simplicity we use only POINT, LINE, REGION, and their combinations.

2. The special relationship set "is_located_at" that associates a spatial entity set with its geometry. The cardinality of this set is 1:M, meaning that a spatial entity may have more than one geometry when multiple granularities are employed. Assuming that in each application we deal only with one space, then the relationship set "belongs_to" between GEOMETRY and SPACE with cardinality M:1 is also included. When GEOMETRY is connected to SPACE then it captures also the *locations* of objects. In this case, GEOMETRY describes objects positions, i.e., shapes and locations [9].

The letters "s," "P," "L," or "R" in a circle in the lower-right corner of an entity set rectangle specify the spatial support. Letter "s" stands for SPATIAL and is used to indicate a spatial entity set whose exact geometric type is unknown. Letters "P," "L," "R," and their combinations specify geometric types as indicated above. These annotations may occur simultaneously and represent then different views of the same object. A spatial entity set is depicted as shown in Figure 3.3(a), and its meaning in terms of the ER model is given in Figure 3.4. Figure 3.3(b) illustrates the spatial entity set "landparcel" with simultaneous geometries point and region; in this case, the representation in the ER model will have only REGION and POINT as geometries.

Fig. 3.3. (a) Spatial entity sets, (b) a landparcel as POINT or REGION

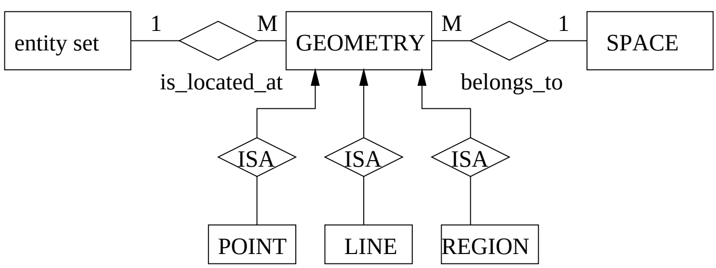

Fig. 3.4. ER diagram corresponding to Figure 3.3(a)

(iii) Spatio-temporal Entity Sets. For a geo-referenced object, GEOMETRY captures both its shape and location. When an object changes its position over time,

i.e., the object moves (change of location) or the object changes shape, it is the GEOMETRY aspect of the object that changes rather than the object itself. For example, a car moving on a road changes its location, but this is not considered to change the car's identity. To capture a temporal aspect of the positions of the objects in an entity set, an "svt," an "stt," or an "sbt" is placed in a circle in the lower-right corner of the entity set's rectangle. The first annotation indicates valid-time support: the objects' current positions as well as their past and future positions are to be captured. This is illustrated in Figure 3.5(a). The second annotation (i.e., "stt") indicates transaction-time support: the current positions as well as all positions previously recorded as current in the database are to be captured. The third annotation (i.e., "sbt") indicates support for both valid and transaction time. Figure 3.5(b) shows that, when a "car" changes position we record both the car's position in time (i.e., "Pvt") and the time this is recorded in the database (i.e., "Ptt"); we indicate this by "Pbt". If the geometric type of the entity set is known, the "s"-part is replaced by "P," "L," "R," or a combination of these. The meaning of the spatio-temporal entity in Figure 3.5(a) is

(a) (b)

Fig. 3.5. (a) A spatio-temporal entity set with valid-time support, (b) recording car position with valid-time and transaction-time support

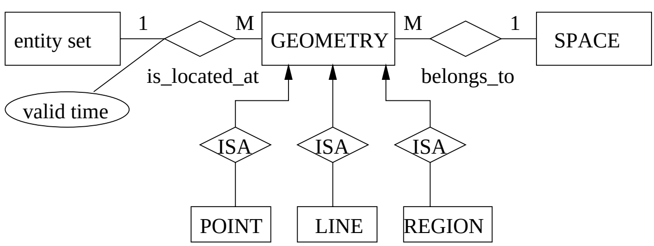

Fig. 3.6. ER diagram corresponding to Figure 3.5(a)

given in Figure 3.6 in terms of the ER model.

It is important to point out the difference between keeping track of (a) a spatial entity set in time, and (b) the position of a spatial entity set in time. In case (a) the temporal support refers to an entity's existence and recording in time. Figure 3.6 models case (a). Figures 3.7(a) and (b) show the entity set "landparcel" as a spatial entity set (indicated by an "s"-part). Figure 3.7(a) illustrates that a cadastral database captures the existence time (i.e., when it first existed-indicated by "et") as well as the transaction time (i.e., when it was recorded in the database-indicated by "tt"). In contrast, Figure 3.7(b) shows that for "landparcel," the database captures transaction time as before (indicated by "tt"), as well as its geometry over time (indicated by "svt"). This means that if a landparcel changes shape in time, the (current) shape and the time this shape is true is recorded (valid time).

Fig. 3.7. (a) A landparcel in space and time and (b) a landparcel in time, with position in time

Attributes of Entity and Relationship Sets. In a spatial environment, entity sets have two types of attributes: (a) *descriptive* attributes, such as the "cadastral-id" of a landparcel, and (b) *spatial* attributes, such as the "soil type" of a landparcel. The values of descriptive attributes for an entity (or a relationship) often change over time, and it is often necessary to capture this in the database. Spatial attributes for which a temporal aspect is captured are termed *spatio-temporal attributes.*

(i) Temporal Descriptive Attributes. Values of attributes of entities denote facts about the entities and thus have both valid- and transaction-time aspects. A circle with a "vt" or a "tt" in the upper-right corner of an oval denoting an attribute indicates that valid or transaction time, respectively, is to be captured. A circle with "bt" (bitemporal) indicates that both temporal aspects are to be captured. The sample STER diagram in Figure 3.8(a) contains an attribute with valid-time support and Figure 3.9 gives the equivalent ER diagram. Figure 3.8(b) shows an example keeping track of cars' colors and their valid time periods.

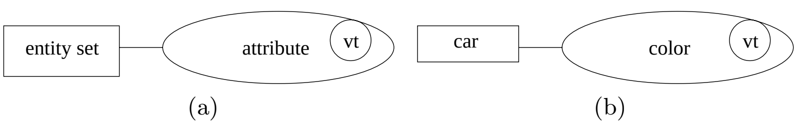

Fig. 3.8. (a) An attribute with valid-time support and (b) car "color" with valid-time support

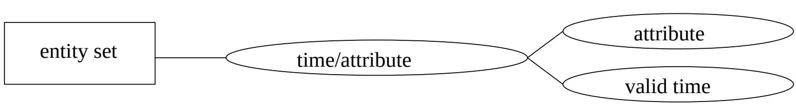

Fig. 3.9. ER diagram corresponding to Figure 3.8(a)

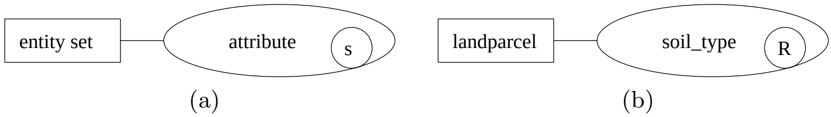

Fig. 3.10. (a) A spatial attribute in STER and (b) "soil type" as a spatial attribute

(ii) Spatial Attributes. Facts captured by attributes may also have associated locations in space, which are described as sets of geometric figures. To capture this spatial aspect of an attribute, a circle with an "s" is used, as shown in Figure 3.10. Figure 3.10(a) depicts the general representation of a spatial attribute, while Figure 3.10(b) shows that the "soil type" value of a landparcel is associated with a set of spatial regions ("R"). In terms of the ER model, a spatial attribute (Figure 3.11) is modeled as an entity set with a composite attribute "attribute/spatial_unit." This consists of the "attribute value" and the "spatial-unit," where the unit represents the geometry in which the "attribute value" is constant. So, the spatial unit can be "POINT" (or "P"), "LINE" (or "L") and "REGION" (or "R") geometric type. The spatial attribute is further connected to SPACE via the relationship set "has_spatial_attribute." In this way, each part of space is assigned a specific value of the attribute. By connecting a spatial entity set to GEOMETRY (via the special relationship "is_located_at," see previous figures) and GEOMETRY to SPACE (via "belongs_to"), an object inherits spatial attributes. So, spatial attributes of entities are *derived properties* ([11] uses *propagated* instead of *derived*) from space (indicated as shaded).

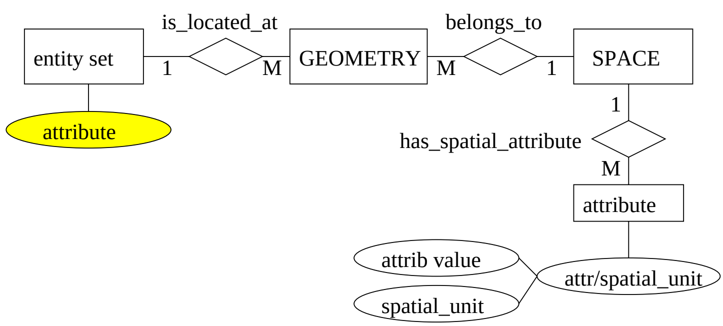

Fig. 3.11. ER diagram corresponding to Figure 3.10(a)

(iii) Spatio-temporal Attributes. Two cases are distinguished here. The first case concerns spatial attributes with temporal support that refers to the attributes' valid- and transaction-time periods (i.e., the spatial attribute is treated as a normal attribute in time). This is illustrated in Figure 3.12(a) and in Figure 3.13, which gives the equivalent ER diagram. Figure 3.12(b) gives an example. In the second case, the temporal aspects (valid and transaction time) of spatial attributes are recorded by placing "svt," "stt," or "sbt" (and replacing the "s"

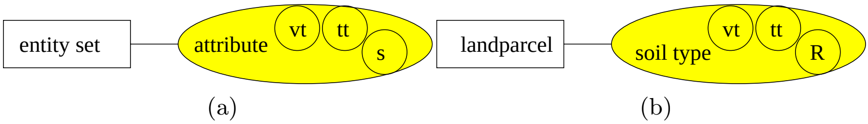

Fig. 3.12. (a) A spatial attribute with temporal support and (b) "soil type" with temporal support

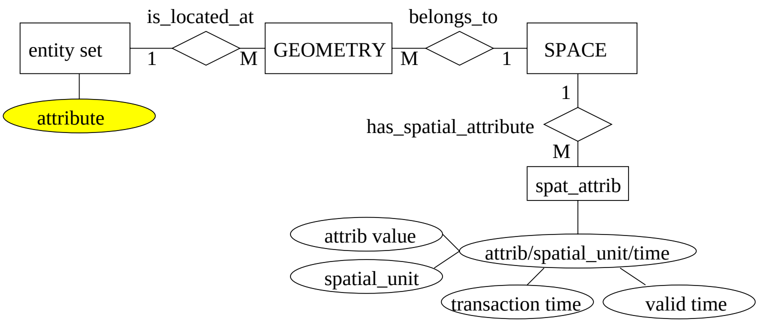

Fig. 3.13. ER diagram corresponding to Figure 3.12(a)

with "P," "L," or "R," or a combination of these if the geometric types of the geometric figures of the attributes are known) in same way as for entity sets. This is illustrated in Figure 3.14(a), and Figure 3.15 gives the equivalent ER diagram. Figure 3.14(b) gives an example. The difference between the two cases may be seen by comparing Figure 3.13 and Figure 3.15.

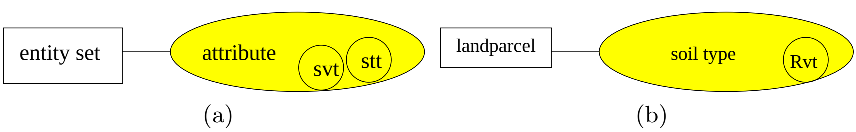

Fig. 3.14. (a) A spatial attribute with temporal support for its spatial part and (b) "soil type" as a spatio-temporal attribute

Relationship Sets

(i) Temporal Relationship Sets. By annotating a relationship set with a temporal aspect (valid time, transaction time, or both), we capture the changes of the set's relationships with respect to that aspect.

(ii) Spatial Relationship Sets. Spatial relationship sets are special kinds of relationship sets. In particular, they are associations among the geometries of the spatial entities they relate. For reasons of simplicity and ease of understanding, spatial relationship sets are given as relationships among the spatial entity sets

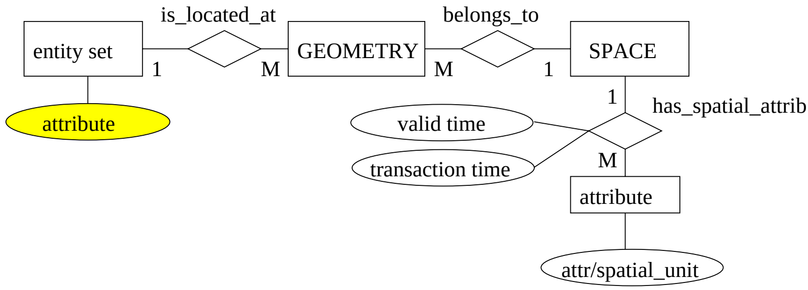

Fig. 3.15. ER diagram corresponding to Figure 3.14(a)

themselves. For example, the relationship "traverses" between cities and rivers relates the geometries of entities of these two spatial entity sets.

(iii) Spatio-temporal Relationship Sets. A spatio-temporal relationship set is a spatial relationship set with time support. In particular, by annotating a spatial relationship set with a temporal aspect, we capture the changes of the spatial relationship with respect to that aspect. Figure 3.16(a) shows the general representation of a spatio-temporal relationship set, while Figure 3.16(b) depicts changes of the relationship "traverses" between cities and rivers are recorded in time. Figure 3.17 gives the equivalent of Figure 3.16(a) in terms of an ER diagram. Finally, the previous discussion about temporal, spatial, and spatio-

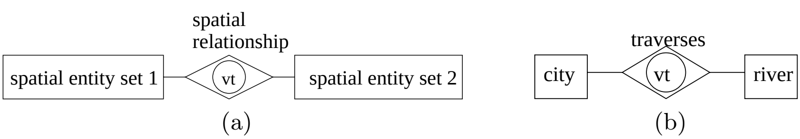

Fig. 3.16. (a) A spatio-temporal relationship set in STER, and "traverses" as a spatio-temporal relationship set

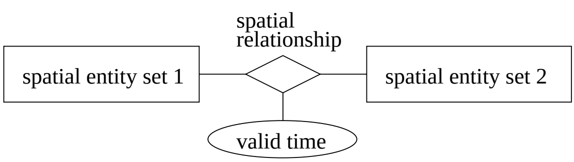

Fig. 3.17. ER diagram corresponding to Figure 3.16(a)

temporal attributes applies also to regular attributes of relationship sets.

In the description of STER so far, we have primarily focused on adding existence and valid time to the ER model constructs. The temporal aspect of transaction time is not strictly part of capturing the modeled reality. However, capturing transaction time, as reflected in the systems requirements, is important when designing a real-world information system, and an ER model should also provide built-in support for the capture of this aspect. Transaction time is applicable in STER exactly where valid time or existence time is applicable. For example, to capture the transaction time of a relationship set, in Figure 3.16(a), all that is needed is to add “tt” to the relationship set construct.

3.3.2 A Textual Notation for STER

In order to fully support the STER model, its graphical notation as well as its textual notation has to be provided. In this way all the constructs of the diagrammatic STER can be presented and/or transformed into the textual notation.

The definitions use the following meta syntax. A `typewriter-like` font indicates elements of the data definition language; syntactic definitions are given with the normal font; *italics* are used for semantic explanations. Upper case words are reserved words, and lower case words are variables that represent arbitrary names. For example, `attr_name_i` stands for any alphanumeric string that is not a reserved word. Capitalized words in lower case denote variables with restricted range. For example, “`Domain`” is one of INTEGER, STRING, DATE, etc. Optional elements are enclosed in “<>”’s, and “(, ...)” denotes repeatable elements; the notation “{...|...}” denotes selection (one of), and “( )” simply indicates grouping of arguments. A longhand notation (e.g., DEFINE instead of DEF) is used to facilitate a first reading; in practice one uses the shorthand.

Definition of Entity Sets

```
DEFINE ENTITY SET entity_set_name
<TYPE Entity_construct (entity_set_name_i, ...)>
<ATTRIBUTES
   ((attribute_name_j
      <VALID_TIME> <TRANSACTION_TIME>
      <GEOMETRY Geometric_type
         <VALID_TIME> <TRANSACTION_TIME>>), ...)>
<GEOMETRY Geometric_type
         <VALID_TIME> <TRANSACTION_TIME>>
<EXISTENCE_TIME> <TRANSACTION_TIME>
<AS ISA OF (entity_set_name_k, ...)>>
```

- `entity_set_name` is an identifying alphanumeric string, i.e., different from any other used in the same syntactic position. `entity_set_name` *is used as the name of the entity set being defined.*
- `Entity_construct` is one of PART_OF, GROUP_OF, SPATIAL_PART_OF, and SPATIAL_MEMBER_OF. `Entity_construct` *is optional and is used to define an entity's complex type.*
- `entity_set_name_i` is an identifying alphanumeric string. `i` is an integer. `entity_set_name_i` *is used to define the constructs of the complex entity set.*

- attribute_name_j is an identifying alphanumeric string. j is an integer. attribute_name_j *is used to define the name of an attribute of an entity set.*
- Geometric_type is one of P, L, or R or combination thereof. *This optional clause is used to define the geometric type of the entity set or the attribute (spatial) defined.*
- entity_set_name_k is an identifying alphanumeric string and k is an integer. entity_set_name_k *defines pre-existing classes used to construct the superset.*

Definition of Composite Attributes

```
DEFINE ATTRIBUTE attribute_name_m
   <AGGREGATION_OF ((attribute_name_n <VALID_TIME Valid_time>
      <TRANSACTION_TIME Transaction_time>
      <GEOMETRY Geometric_type
      <VALID_TIME> <TRANSACTION_TIME>>), ...)>
```

- attribute_name_m is an identifying alphanumeric string. m is an integer. *It is used to define the name of the composite attribute.*
- attribute_name_n is an identifying alphanumeric string. n is an integer. *It is used to define the names of the attributes that compose the complex one.*
- Geometric_type is one of P, L, or R or combination thereof. *This optional clause is used to define the geometric type of the entity set or the attribute (spatial) being defined.*

Definition of Relationship Sets

```
DEFINE <SPATIAL> RELATIONSHIP SET
   relationship_set_name (entity_set_name_i, ...)
   TYPE Relationship_type
   <ATTRIBUTES ((attribute_name_k
      <VALID_TIME> <TRANSACTION_TIME>), ...)>
   <VALID_TIME> <TRANSACTION_TIME>
```

- relationship_set_name is an identifying alphanumeric string. *It defines the functional relationship between entity sets.*
- entity_set_name_i is an identifying alphanumeric string and i is an integer. *It defines the entity sets which are related through the relationship.*
- Relationship_type is one of ONE_TO_ONE, ONE_TO_MANY, MANY_TO_ONE, MANY_TO_MANY. *It is used to define the relationship type.*
- attribute_name_k is an identifying alphanumeric string and k is an integer. *It is used to indicate attributes of the defined relationship set.*

3.3.3 Example of Usage of STER

The following example presents an excerpt of the conceptual schema for a cadastral application based on user requirements [22]. This excerpt is a substantial simplification of a real-world situation. We used STER in an intermediate phase to translate user requirements (which were expressed in natural language) to a formal logical schema consisting of relations and maps.

Figure 3.18 states that for "landparcels," existence time is recorded; the positions of landparcels can be either points or regions. Moreover, "landparcels" have two spatio-temporal attributes: (a) "soil type," which is of type REGIONS ("R"), and for which both valid and transaction time are captured, and (b) "elevation," which is recorded in terms of points. Additionally, "landparcels" may be traversed by "rivers." For the spatial relationship "traverses," we capture transaction time. For "rivers," transaction and existence time is captured. Finally, "rivers" are represented as lines ("L") and are distinguished by their "cadastral number." "landparcels" may have different "land use": (a) "agricultural" use is recorded in regions, based on its "vegetation" (which is also captured in regions), and (b) "industrial" use which is defined in regions and is characterized by different "types," such as heavy industry. For the type of industry we keep track of the time this is valid. For the relationship "land use," we record valid and transaction time (this could also be indicated by the symbol "bt"). Finally, "buildings" may be inside "landparcels," at different time periods, which are recorded in the database. A "building" occupies a region and belongs to "owners." Valid time is captured for "ownership"; for owners, Social Security Number ("SSN") and "name" are known. In the following, we informally explain an excerpt (the first five constructs) of the description line by line, showing the one-to-one translation of the constructs of the diagrammatic representation.

We express that the entity set "landparcel" has "soil type" and "elevation" as attributes. For "soil type" we record the valid and the transaction time of its geometry, which is REGION. For "elevation" we record only its geometry, which is POINT. "landparcel" is of type REGION or POINT and we keep track of its existence time. Additionally, we have the entity set "river" keeping track of its existence and transaction time in the database. A "river" is of type LINE and has its identification ("cadastral number") as attribute. Landparcels and rivers are related via the spatial relationship "traverses," for which we record the transaction time ("tt") of the changes.

```
DEFINE ENTITY SET landparcel
ATTRIBUTES
 ((soil type (GEOMETRY REGION VALID_TIME TRANSACTION_TIME))
 (elevation GEOMETRY POINT))
GEOMETRY POINT REGION
EXISTENCE_TIME

DEFINE ENTITY SET river
ATTRIBUTES (cadastral number)
```

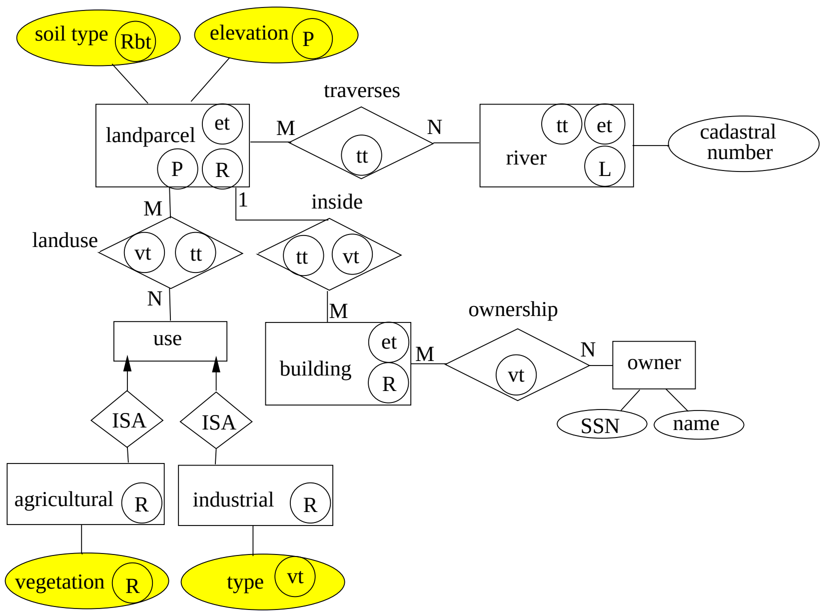

Fig. 3.18. Using STER to model an excerpt of a cadastral application

```
GEOMETRY LINE
EXISTENCE_TIME TRANSACTION_TIME

DEFINE RELATIONSHIP SET traverses (landparcel, river)
TYPE MANY_TO_MANY
TRANSACTION_TIME

DEFINE ENTITY SET agricultural
ATTRIBUTES(vegetation GEOMETRY REGION)
GEOMETRY REGION

DEFINE ENTITY SET industrial
ATTRIBUTES(type  VALID_TIME)
GEOMETRY REGION

DEFINE ENTITY SET use
AS ISA OF agricultural, industrial

DEFINE RELATIONSHIP SET landuse (landparcel, use)
TYPE MANY_TO_MANY
VALID_TIME TRANSACTION_TIME

DEFINE ENTITY SET building
```

```
GEOMETRY POINT REGION
EXISTENCE_TIME

DEFINE RELATIONSHIP SET inside (landparcel, building)
TYPE ONE_TO_MANY
VALID_TIME TRANSACTION_TIME

DEFINE ENTITY SET owner
ATTRIBUTES(SSN, name)

DEFINE RELATIONSHIP SET ownership (building, owner)
TYPE ONE_TO_MANY
VALID_TIME
```

3.4 Spatio-temporal Unified Modeling Language

In this section, an extension of UML, Extended spatio-temporal UML [26], is presented to meet spatio-temporal requirements by using a clear, simple and consistent notation to capture alternative semantics for time, space and change processes (i.e., discrete vs. continuous). The focus is on providing support for different categories of spatio-temporal data. These include temporal changes in spatial extents, changes in the value of thematic (i.e., alphanumeric) data across time or space, and composite data whose components vary depending on time or location.

3.4.1 Using UML Core Constructs for Spatio-temporal Data

In order to motivate the need for a spatio-temporal extension to UML, we first evaluate the core constructs of UML [4,17,31] in terms of their suitability for modeling spatio-temporal data. UML consists of a set of nine types of diagrams and notational conventions based on the OO paradigm. Diagrams can be categorized as specifying either structure or behavior of a system (business or software system) or its data elements (classes or object instances). The Class diagram, exemplified in Figure 3.19, is the most relevant in the current context since it describes the structure of object classes.

The fundamental element of the Class diagram is a class description, consisting of the class name, attribute descriptions, and operation signatures graphically represented in separate compartments of the class box. Attribute descriptions include the name and type (regarded as equivalent to domain for the purposes of this chapter) for each attribute. The classes can be connected by different types of standard OO links including generalization (sub-classes based on a super-class) and association (structural relationships between different object classes). These are represented graphically by a line between classes, terminated with a triangle next to the super-class in the case of generalization. Associations that have their own properties are represented in UML by promoting the association to a class, i.e., an association class, with attributes. UML also defines the specialized associations aggregation (with whole-part semantics) and composition (aggregation with the additional constraint that parts cannot be shared); however, their semantics are not standardized [16,18]. These relationships are represented by an association line terminating in a diamond next to the "whole" class, which is shaded in the case of composition. Any element of a class diagram can also be annotated by ad hoc notes (in a rectangle with a "folded" corner) or constraints (in curly braces) giving further details of that element's semantics.

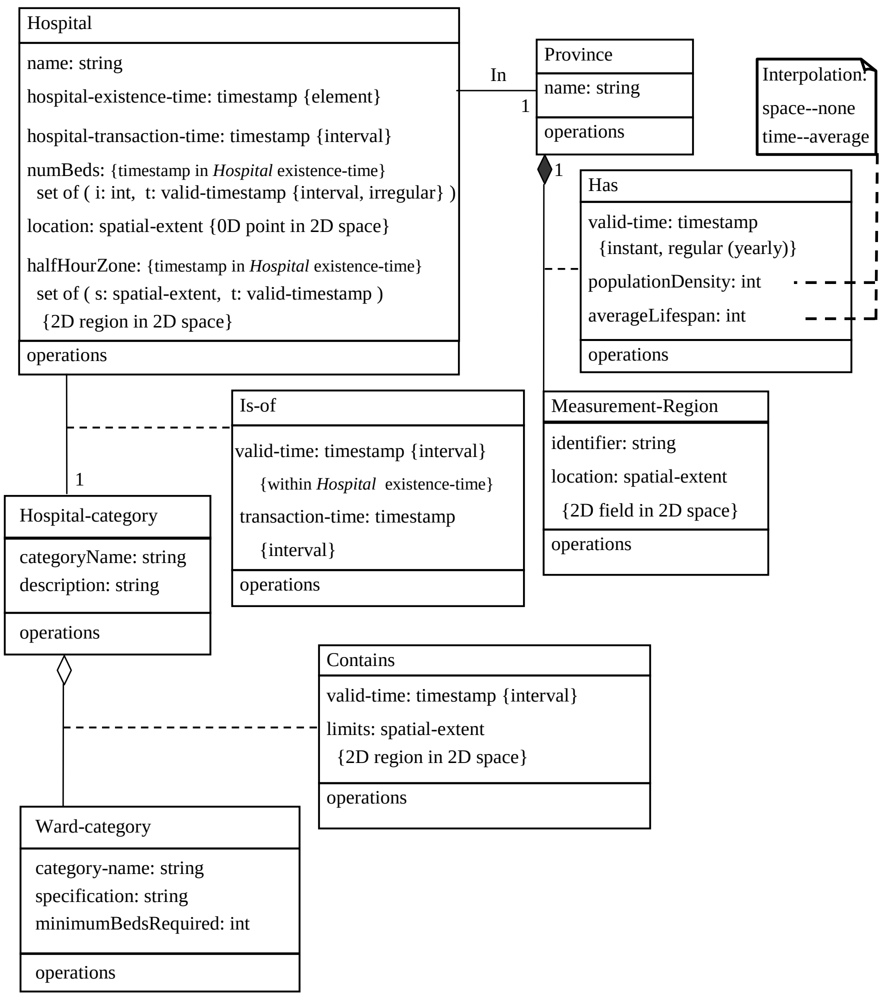

Fig. 3.19. Regional health application in UML

A regional health application will be used to illustrate the use of the standard UML class diagram for spatio-temporal semantics, as follows. Assume an application measuring health statistics of different provinces, in terms of average lifespan, as related to the location (i.e., a point in 2D space), number of beds, accessibility (i.e., a half-hour travel zone around the hospital), and surrounding population densities of a province's hospitals. A hospital is classified by category, where a given category is required to have a minimum number of beds in specific kinds of wards. However, category definitions may differ between regions due to local regulations.

For properties dependent on time, location, or both, we want to record information about when (using time intervals unless otherwise specified) or where a given value is valid in the miniworld (i.e., valid time) or is current in the database (i.e., transaction time). For example, a province's population densities and average lifespans can vary and are recorded yearly at the same time instants (values are averaged between yearly measurements) and for the same regions. The number of beds, the half-hour travel zone, a hospital's category, and the regional definition of hospital categories may change over time as well. We want to record existence and transaction time for hospitals, valid time and transaction time for a hospital's category, and valid time for all of the other time dependent properties. The time unit for the half-hour travel zone is not yet specified, demonstrating incremental design specification. Time elements (the union of a set of time intervals) are used to model hospital existence time since hospitals may sometimes be closed and later re-opened based on changes in local population density. Note that the number of beds, half-hour travel zone, and hospital category are only defined when the hospital is open.

This example demonstrates three categories of spatio-temporal data. For example, changes to the hospital's half-hour travel zone illustrate temporal changes in spatial extents, changes in population density demonstrate changes in the value of thematic data across time and space, and changing definitions of hospital categories show composite data whose components vary depending on time or location.

Representation of spatio-temporal concepts using the core constructs of UML is not straightforward. This is illustrated using the regional health example in Figure 3.19, which assumes that spatial extents can be a combination of points, lines, regions, and volumes, and that timestamps can be instants, intervals, and elements.

Attributes with spatial, temporal, or spatio-temporal properties (e.g., the half-hour travel zone) can be modeled (e.g., with the halfHourZone attribute) using composite attribute domains consisting of a set of tuples, where each tuple consists of a thematic value, possibly a spatial extent, and possibly one or more timestamps. Alternatively, an attribute with spatial and temporal properties (e.g., population density or average lifespan) could be promoted to a separate but associated class with the same information added to the new class. Although not required by the semantics of the example application, we must also create an artificial identifier attribute for this class because its instances must be uniquely identified [31]. Of more concern, this approach will lead to redundancy whenever the same attribute value is repeated for different object instances, times, or spatial extents. This is especially significant for spatial data because of their size. The extra classes also complicate the schema.

A more correct approach, in general, would be to promote the association to an association class (e.g., Has) with spatial data in the associated class (e.g., Measurement-Region) and thematic data, timestamp data (e.g., populationDensity, averageLifespan, and valid-time), or both in the association class. Classes and associations with temporal, spatial, or spatio-temporal properties (e.g., Hospital and hospital Is-of category respectively) can be treated similarly, by adding the timestamp and spatial attributes after promoting the association to an association class in the latter case. However, this still does not solve the problem of the artificial identifier or the extra complexity introduced.

Constraints are used to indicate the time units for timestamps, the time model, the dimensions of spatial extents, and the existence-dependencies described for the application example. Notes are used to show interpolation semantics. Association, rather than generalization, is used to represent the hospital category, since its definition varies regionally and does not affect the properties defined for the hospital class.

Figure 3.19 indicates that it is necessary to create a new association class for each association with spatial or temporal properties. As it can be seen, this leads to the creation of a host of artificial constructs that significantly complicate the schema diagram. Furthermore, there is no single, easily visible notation to represent spatio-temporal properties. This violates the requirement that the notation be simple, clear, and consistent. A better approach is to extend the fundamental characteristics of the existing UML elements to directly support spatio-temporal requirements ([26] compares different approaches to extending a data model). This would involve changes to the structure of instantiated UML elements (i.e., object, association, and attribute instances) to provide for associated time periods or spatial extents. Stereotypes (indicating a variation in usage or meaning), tagged values (adding properties), and constraints (adding semantics) are extension mechanisms defined within UML to extend any core element in the UML metamodel. However, none of the mechanisms support the structural changes in element instances necessary to model spatio-temporal semantics. Other limitations, inconsistencies, and ambiguities in the definitions of these extension mechanisms ([26,18] have a detailed discussion of the problems) further reinforce the argument that spatio-temporal extension constructs must necessarily go beyond the extension mechanisms defined for UML. Although these mechanisms may be used as a guide, a strict adherence to their UML definitions is not desirable. We then proceed to describe the proposed extension, based on a small set of orthogonal constructs and consistent rules for combining them.

3.4.2 Overview of Extended Spatio-temporal UML

The proposed extension to UML is based on the addition of five new symbols, illustrated in Figure 3.20, and a specification box describing the detailed semantics of the spatio-temporal data represented using the five symbols. The basic approach is to extend UML by adding a minimal set of constructs for spatial, temporal, and thematic data, represented respectively by spatial, temporal, and thematic symbols. These constructs can then be applied at different levels of the UML class diagram and in different combinations to add spatio-temporal semantics to a UML model element. In addition, the group symbol is used to group attributes with common spatio-temporal properties or inter-attribute constraints and the existence-dependent symbol is used to describe attributes and associations dependent on object existence.

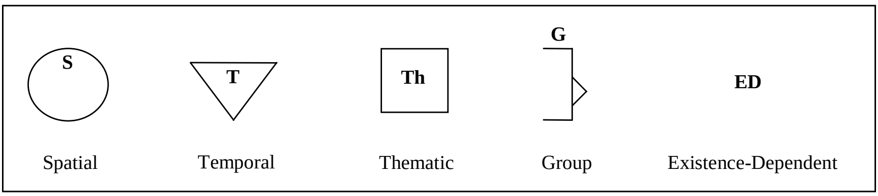

Fig. 3.20. Extended spatio-temporal UML symbols

As discussed previously, although these new symbols can be roughly described as stereotypes they do not adhere strictly to the UML definition. For improved readability, we use the alternative graphical notation for stereotypes described in [31]. These symbols can be annotated with a unique label used to reference the associated specification box. The first four symbols can optionally be used without the abbreviations shown in the figure (i.e., S, T, Th, and G respectively). The specific alphanumeric domain can be optionally indicated, e.g., Th: int, which denotes a thematic attribute of type integer. We first discuss the basic spatial, temporal, and thematic constructs.

3.4.3 Basic Constructs: Spatial, Temporal, Thematic

These constructs can be used to model spatial extents, object existence or transaction time, and the three different types of spatio-temporal data previously discussed (i.e., temporal changes in spatial extents; changes in the values of thematic data across time or space; and composite data whose components vary depending on time or location). To understand the use and semantics of the spatial, temporal, and thematic constructs, we first describe the notation used in this section, then discuss the interpretation of each individual symbol separately, and finally give rules for combining symbols.

The primitives used in this section to denote various time, space, and model elements are listed below. The subset of these that have graphic equivalents are illustrated in Figure 3.21.

- ⟨T⟩ ::= domain of time instants
- $\langle 2^{\mathrm{T}} \rangle$::= arbitrary set of time instants, i.e., a timestamp or set of timestamps
- ⟨S⟩ ::= domain of points in space
- $\langle 2^{\mathrm{S}} \rangle$ ::= arbitrary set of points in space, i.e., a spatial extent or set of spatial extents
- ⟨oid⟩ ::= domain of object-identifiers
- ⟨aid⟩ ::= domain of association-instance identifiers, essentially $\{\langle \text{oid} \rangle\}^n$
- ⟨id⟩ ::= domain of object and association identifiers, essentially {⟨oid⟩|⟨aid⟩}
- ⟨D⟩ ::= thematic, i.e., alphanumeric, domain (e.g., integer, string)
- ⟨t⟩ ::= temporal symbol, graphically illustrated by a triangle, see Figure 3.21
- ⟨s⟩ ::= spatial symbol, graphically illustrated by a circle, see Figure 3.21
- ⟨s⟨t⟩⟩ ::= temporal symbol inside a spatial symbol, see Figure 3.21
- ⟨d⟨s⟩⟩ ::= spatial symbol inside a thematic symbol (illustrated by a rectangle), see Figure 3.21
- ⟨d⟨t⟩⟩ ::= temporal symbol inside a thematic symbol, see Figure 3.21
- ⟨d⟨s⟨t⟩⟩⟩ ::= temporal symbol inside a spatial symbol; both inside a thematic symbol, see Figure 3.21
- ⟨s&t⟩ ::= any nested combination of a spatial and a temporal symbol
- ⟨s&d⟩ ::= any nested combination of a spatial and a thematic symbol
- ⟨t&d⟩ ::= any nested combination of a temporal and a thematic symbol
- ⟨s&t&d⟩ ::= any nested combination of a spatial, a temporal, and a thematic symbol
- ⟨ED⟩ ::= existence-dependent symbol

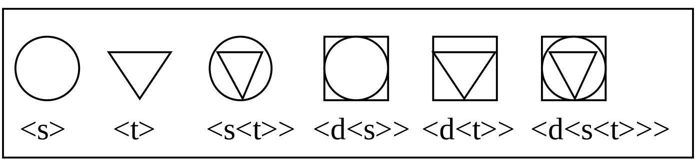

Fig. 3.21. Primitives with graphical equivalents

The spatial symbol represents a spatial extent, which consists of an arbitrary set of points, lines, regions, or volumes. The spatial extent may be associated with thematic or composite data, or may be used to define an attribute domain. The temporal symbol represents a temporal extent, or timestamp, which may be associated with thematic, spatial, or composite data. Timestamps may represent existence time for objects, valid time for associations and attributes, and transaction time for objects, associations, and attributes. The thematic symbol represents thematic data.

The thematic symbol can only be used at the attribute level and only in conjunction with one of the other two symbols to describe an attribute with temporal or spatial properties. A thematic attribute domain with no spatial or temporal properties uses standard UML notation, i.e., ⟨attribute-name⟩: ⟨domain⟩. When there are such properties, either this notation can be used for the thematic domain or the specific thematic domain can be indicated inside the thematic symbol. Figure 3.22 illustrates the four possible cases for a thematic attribute: attributes with a thematic domain and (a) no spatial or temporal properties, (b) temporal properties, (c) spatial properties, or (d) spatio-temporal properties. Adjectives are used to describe the attribute domain (e.g., thematic attribute) and adverbs with the word dependent to describe additional attribute properties for composite attribute domains (e.g., temporally-dependent thematic attribute). Therefore, the four possible cases for thematic attributes are called (a) thematic, (b) temporally-dependent thematic, (c) spatially-dependent thematic, or (d) spatio-temporally-dependent thematic attributes respectively.

The semantics of Extended spatio-temporal UML-depend on three factors: (a) the symbol used, (b) the model element described by the symbol (i.e., object, association, or attribute), and (c) whether the symbol is combined with other symbols. The general rules for combining symbols can be summarized as follows:

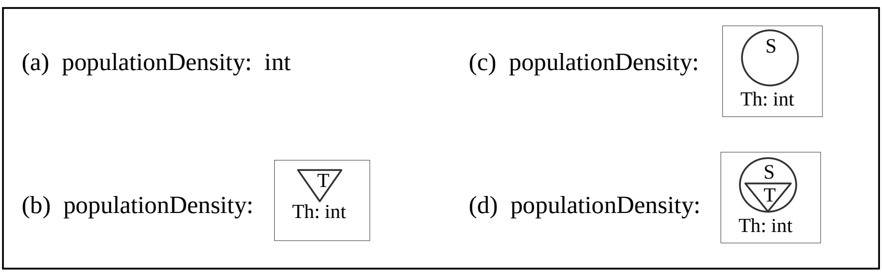

Fig. 3.22. Thematic attribute examples

- Nesting one symbol inside another represents mathematically a function from the domain represented by the inner symbol to the domain represented by the outer symbol. Therefore, different orders of nesting symbols correspond to different functional expressions and represent different perspectives of the data.
  For example, Figure 3.22(b) represents a function from the time to the integer domain for a given object or association instance as follows:

  $$f : \langle \text{id} \rangle \rightarrow (\langle \text{T} \rangle \rightarrow \langle \text{D} \rangle)$$

  If we reverse symbol nesting, this would represent the inverse function, from the integer domain to the powerset of the time domain (this follows directly from the standard definition of a function):

  $$f : \langle \text{id} \rangle \rightarrow (\langle \text{D} \rangle \rightarrow \langle 2^{\text{T}} \rangle)$$

  However, from the conceptual design perspective, both represent the same semantic modeling category and would result in the same conceptual and logical schema, i.e., a temporally-dependent, thematic attribute. Similarly, although both represent a spatio-temporally-dependent thematic attribute, Figure 3.22(d) and its inverse (rectangle inside a circle, both inside a triangle) correspond to different mathematical functions, respectively:

$$f : \langle \mathrm{id} \rangle \rightarrow (\langle \mathrm{T} \rangle \rightarrow (\langle \mathrm{S} \rangle \rightarrow \langle \mathrm{D} \rangle))$$
$$f : \langle \mathrm{id} \rangle \rightarrow (\langle \mathrm{D} \rangle \rightarrow (\langle \mathrm{S} \rangle \rightarrow \langle 2^{\mathrm{T}} \rangle)).$$

Allowing different orders of nesting allows users to select the one that best matches their perspective of the application data and could potentially be exploited for a graphical query language or to indicate preferred clustering patterns to the database management system when generating the physical schema.

Note also that in Figure 3.22(b), only one integer value is associated with each timestamp; however, several different timestamps may be associated with the same integer value. In Figure 3.22(d), several integer values will be associated with each timestamp, one for each spatial location.

- Placing one symbol next to another symbol represents mathematically two separate functions, one for each symbol. The order in which the two symbols are written is not significant.

We now give the rule for which symbolic combinations are legal at each model level, the semantic modeling constructs defined at each level, and a mapping between the two. For a given semantic modeling construct, a symbolic representation, mathematical definition in terms of function(s), and textual definition is given for the nesting used in this chapter's figures. However, as discussed previously, any other nesting order for the same semantic modeling construct is allowed and is described in detail in [26]. Note that any reference to a timestamp, timestamps, a time point, or time validity in the definitions for a given symbol nesting could be for any time dimension, i.e., one or both of transaction and valid time (for attributes and associations) or existence time (for objects).

The Attribute (and Attribute Group) Level. At the attribute level, we can model temporal changes in spatial extents, where the spatial extent represents a property of an object (i.e., spatial attribute), and changes in the value of thematic data across time, space, or both (i.e., temporally, spatially, or spatio-temporally-dependent thematic attributes).

Legal combinations of symbols at the attribute level are include any nested combination of spatial temporal and thematic symbols, except that neither the thematic nor the temporal symbol may be used alone. A thematic attribute uses standard UML notation as discussed earlier. The only exception is that the temporal symbol cannot be used alone. An attribute with a temporal domain is treated as thematic data since temporal data types are pre-defined for popular standard query languages such as SQL. The attribute domain can optionally be followed by an existence-dependent symbol. The rule for notation at this level can be defined using BNF notation and the primitives defined previously:

attribName: [$\langle$D$\rangle$ | $\langle$s&d$\rangle$ | $\langle$t&d$\rangle$ | $\langle$s&t&d$\rangle$ | $\langle$s$\rangle$ | $\langle$s&t$\rangle$] [$\langle$ED$\rangle$]

Six different attribute domains are possible, corresponding to the semantic categories of attributes (i.e., modeling constructs). Reading the domain symbols left to right, we have: thematic attributes; spatially, temporally, and spatio-temporally-dependent thematic attributes; spatial attributes; and temporally-dependent spatial attributes. Except for thematic attributes, these domains represent extensions for spatio-temporal data modeling. A general textual description and symbolic representation (with its corresponding mathematical and textual definition) is given below for each semantic attribute category (or for each semantic object or association category respectively in the following two sections). Note that each one of the definitions below applies to the identified object or association instance: therefore, we do not state this explicitly in the definitions.

- Thematic Attribute: This is an attribute with thematic values.

  $\langle \mathrm{D} \rangle \qquad f : \langle \mathrm{id} \rangle \rightarrow \langle \mathrm{D} \rangle$

  Returns the thematic attribute value.
- Spatially-dependent Thematic Attribute: This is a set of thematic attribute values, each associated with a spatial extent representing the location where that attribute value is valid. This implies that the attribute values may change over space and their changed values may be retained.

  $\langle \mathrm{d}\langle \mathrm{s} \rangle\rangle \qquad f : \langle \mathrm{id} \rangle \rightarrow (\langle \mathrm{S} \rangle \rightarrow \langle \mathrm{D} \rangle)$

  Returns a set of spatial points, each with its associated thematic attribute value (valid for that spatial point).
- Temporally-dependent Thematic Attribute: This is a set of thematic attribute values, each associated with one or more timestamps, representing the valid time, the transaction time, or both times of the attribute value. This implies that the attribute values may change over time and their changed values may be retained.

  $\langle \mathrm{d}\langle \mathrm{t} \rangle\rangle \qquad f : \langle \mathrm{id} \rangle \rightarrow (\langle \mathrm{T} \rangle \rightarrow \langle \mathrm{D} \rangle)$

  Returns a set of time points, each with its associated thematic attribute value (i.e., valid for that time point).
- Spatio-temporally-dependent Thematic Attribute: This is a combination of spatially and temporally-dependent thematic attributes as defined above, i.e., a set of thematic attribute values, each associated with a spatial extent and one or more timestamps.

  $\langle \mathrm{d}\langle \mathrm{s}\langle \mathrm{t} \rangle\rangle\rangle \qquad f : \langle \mathrm{id} \rangle \rightarrow (\langle \mathrm{T} \rangle \rightarrow (\langle \mathrm{S} \rangle \rightarrow \langle \mathrm{D} \rangle))$

  Returns a set of time points, each with its associated set of spatial points, and, for each spatial point, its associated thematic attribute value (i.e., valid for that time and spatial point).

- Spatial Attribute: This is an attribute with a spatial domain, i.e., the attribute value is a spatial extent.

  $\langle \mathrm{s} \rangle \qquad\qquad f : \langle \mathrm{id} \rangle \rightarrow \langle 2^{\mathrm{S}} \rangle$

  Returns the spatial attribute value.
- Temporally-dependent Spatial Attribute: A spatial attribute is associated with one or more timestamps, representing the valid time, transaction time, or both times of the spatial extent.

  $\langle \mathrm{s}\langle \mathrm{t} \rangle \rangle \qquad\qquad f : \langle \mathrm{id} \rangle \rightarrow (\langle \mathrm{T} \rangle \rightarrow \langle 2^{\mathrm{S}} \rangle)$

  Returns a set of time points, each with its associated spatial attribute value (i.e., spatial extent).

The use of these symbols at the attribute level is illustrated in Figure 3.23. The difference between (a) thematic attributes, (b) temporally-dependent thematic attributes, (c) spatio-temporally-dependent thematic attributes, (d) spatial attributes, and (e) temporally-dependent spatial attributes is illustrated by (a) name (for Hospital and Province), (b) numBeds, (c) populationDensity, (d) location, and (e) halfHourZone respectively.

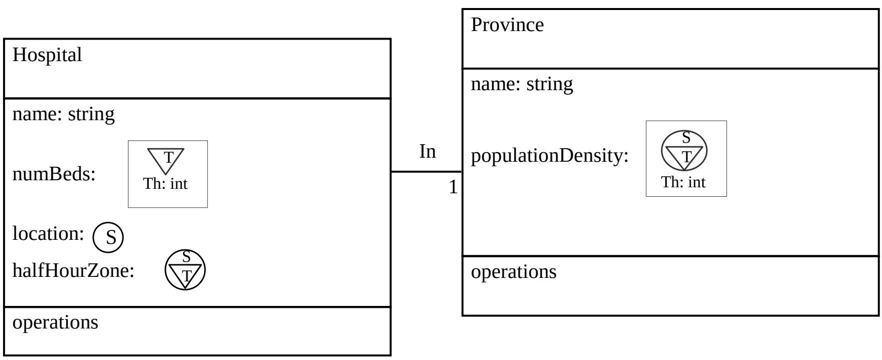

Fig. 3.23. Using extended spatio-temporal UML at the attribute level

A thematic attribute domain is indicated as a string after the attribute or-if that attribute also has temporal or spatial properties-by the use of a thematic symbol. If no domain is explicitly specified for an attribute, then the use of the spatial symbol indicates that the attribute has a spatial domain. Thus, the Hospital location and halfHourZone attributes represent spatial data. The nested temporal symbol used for halfHourZone indicates that the spatial extent associated with this attribute may change over time and thus should be timestamped. Therefore, an attribute marked by a spatio-temporal symbol (and no thematic domain) represents a spatial extent that changes over time. In this case, as transport networks change, the geometry of the half-hour travel zone must be updated.

In contrast, an attribute that has a thematic domain and a spatial and/or temporal symbol represents a spatially- and/or temporally-dependent thematic attribute. This is indicated graphically by using the thematic symbol; thus this symbol is used to differentiate two different types of spatio-temporal data: temporal changes in spatial extents and changes in the value of thematic data across time and space. Therefore, the fact that numBeds has an integer domain associated with a temporal symbol indicates that the integer value of numBeds may change over time and should be timestamped. Analogously, the integer value of populationDensity may change over time or space and thus each value is associated with a timestamp and spatial extent.

The Object Class Level. At the object class level, we can model temporal changes in spatial extents, where the spatial extent is associated with an object instance. We can also model the time an object exists in the real world (i.e., existence time) or is part of the current database state (i.e., transaction time).

An object class can be marked by a temporal symbol, a spatial symbol, or any nested combination of these. In addition, this is the only level where the symbols can be paired; i.e., a temporal symbol can be paired with either a spatial symbol or a nested combination of the two symbols. The separate temporal symbol represents the existence or transaction time of the object. The spatial symbol represents the spatial extent associated with that object. If the spatial symbol is combined with a nested temporal symbol, then the spatial extent is timestamped to show the valid or transaction time of the spatial extent. Since the object can exist or be current even when not actually associated with a spatial extent, separate timestamps are required for the object instance and for the object instance's spatial extent. The rule for object level notation can be given in BNF as follows:

className [⟨s⟩ | ⟨s&t⟩] [⟨t⟩]

Corresponding to the five possible instantiations of this rule, ⟨s⟩; ⟨s&t⟩; ⟨t⟩; ⟨s⟩⟨t⟩; and ⟨s&t⟩⟨t⟩, there are five different categories of object classes as defined below.

- Spatial Object (Class): An object is associated with a spatial extent. This is equivalent to an object having a single spatial attribute except that there is no separate identifier for the spatial extent.

  ⟨s⟩ $f : \langle \text{oid} \rangle \rightarrow \langle 2^{\text{S}} \rangle$

  Returns the spatial extent of the identified object.
- Temporally-dependent Spatial Object (Class): The spatial extent associated with a spatial object is also associated with one or more timestamps, representing the spatial extent's valid time or transaction time (or both).

$$\langle \text{s}\langle \text{t}\rangle\rangle \qquad f : \langle \text{oid}\rangle \rightarrow (\langle \text{T}\rangle \rightarrow \langle 2^{\text{S}}\rangle)$$

Returns a set of timepoints, each associated with the spatial extent of the identified object at that timepoint.

- Temporal Object (Class): An object is associated with one or more timestamps, representing the object's existence time or transaction time (or both).

$$\langle \text{t}\rangle \qquad f : \langle \text{oid}\rangle \rightarrow \langle 2^{\text{T}}\rangle$$

Returns the timestamp of the identified object.

- Spatio-temporal Object (Class): This is a combination of a spatial and temporal object as defined above, i.e., each object instance is associated with a spatial extent and one or more timestamps representing the object's existence time or transaction time (or both).

$$\langle \text{t}\rangle\ \langle \text{s}\rangle \qquad f : \langle \text{oid}\rangle \rightarrow \langle 2^{\text{T}}\rangle \text{ and } f : \langle \text{oid}\rangle \rightarrow \langle 2^{\text{S}}\rangle$$

Returns the timestamp and the spatial extent of the identified object.

- Temporally-dependent Spatio-temporal Object (Class): This is a combination of a temporally-dependent spatial object and a temporal object as defined above, i.e., an object is associated with a spatial extent, one or more timestamps representing the spatial extent's valid time or transaction time (or both), and one or more timestamps representing the object's existence time or transaction time (or both).

$$\langle \text{t}\rangle\ \langle \text{s}\langle \text{t}\rangle\rangle \qquad f : \langle \text{oid}\rangle \rightarrow \langle 2^{\text{T}}\rangle \text{ and } f : \langle \text{oid}\rangle \rightarrow (\langle \text{T}\rangle \rightarrow \langle 2^{\text{S}}\rangle)$$

Returns the timestamp of the identified object and a set of timepoints, each with its associated spatial extent (i.e., valid at that timepoint), for the identified object.

The use of symbols at the object class level is illustrated in Figure 3.24. In Figure 3.24(a), the temporal symbol at the Hospital object level represents a temporal object class with existence and transaction time (discussed in Section 3.4.4). In Figure 3.24(b), we give an example of a temporally-dependent spatial object. This example assumes that there is no need to represent hospital location separately from the half-hour travel zone. Instead, a hospital object is treated as a spatial object with a single associated spatial extent, showing the half-hour travel zone around that hospital. The temporal symbol indicates that the spatial extent should be timestamped, since the half-hour travel zone can change over time. Finally, Figure 3.24(c) combines (a) and (b), illustrating a temporally-dependent spatio-temporal object. The object is spatio-temporal because it has a timestamp and a spatial extent; and temporally-dependent because the spatial extent also has a timestamp.

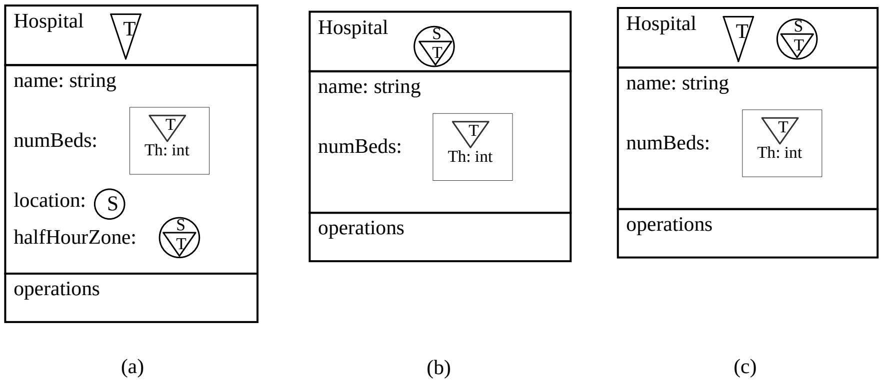

Fig. 3.24. Using extended spatio-temporal UML at the object class level

The Association Level. At the association level, we can model temporal changes in spatial extents, where the spatial extent is associated with a relationship between object instances, and composite data whose components vary depending on time or location. The following discussion applies to any type of association, including aggregation and composition.

At the association level, any nested combination of a spatial and a temporal symbol represents a legal combination describing spatio-temporal properties of the association. Except for the omission of the thematic symbol, the association level is similar to the attribute level. The association spatio-temporal properties can optionally be followed by an existence-dependent symbol (discussed in Section 3.4.4). The rule for the association level notation can be given in BNF as follows:

assoc-line [⟨s⟩ | ⟨t⟩ | ⟨s&t⟩] [⟨ED⟩]

Reading the BNF rule from left to right, three different categories of associations are possible, as defined below.

- Spatially-dependent association: An association instance is associated with a spatial extent representing the location where the association instance is valid. This implies that the association instances may change over space and their changed instances may be retained.

  $\langle \mathrm{s} \rangle \qquad f : \langle \mathrm{aid} \rangle \rightarrow \langle 2^{\mathrm{S}} \rangle$

  Returns the spatial extent of the identified association.
- Temporally-dependent association: An association instance is associated with one or more timestamps, representing the association's valid time or transaction time (or both). This implies that association instances may change over time and the changed instances may be retained.

  $\langle \mathrm{t} \rangle \qquad f : \langle \mathrm{aid} \rangle \rightarrow \langle 2^{\mathrm{T}} \rangle$

Returns the timestamp of the identified association.

- Spatio-temporally-dependent association: This is a combination of spatially and temporally-dependent associations as defined above, i.e., an association is associated with a spatial extent and one or more timestamps.

  $\langle \mathrm{s}\langle \mathrm{t}\rangle\rangle \qquad f : \langle \mathrm{aid}\rangle \rightarrow (\langle \mathrm{T}\rangle \rightarrow \langle 2^{\mathrm{S}}\rangle)$

  Returns a set of time points, each with the associated spatial extent for the identified association at that time point.

The use of these symbols at the association level is shown in Figure 3.25. Marking the Is-of association with a temporal symbol signifies that the category of a hospital may change over time, as local health needs change and wards are opened or closed. Therefore, association instances should be timestamped.

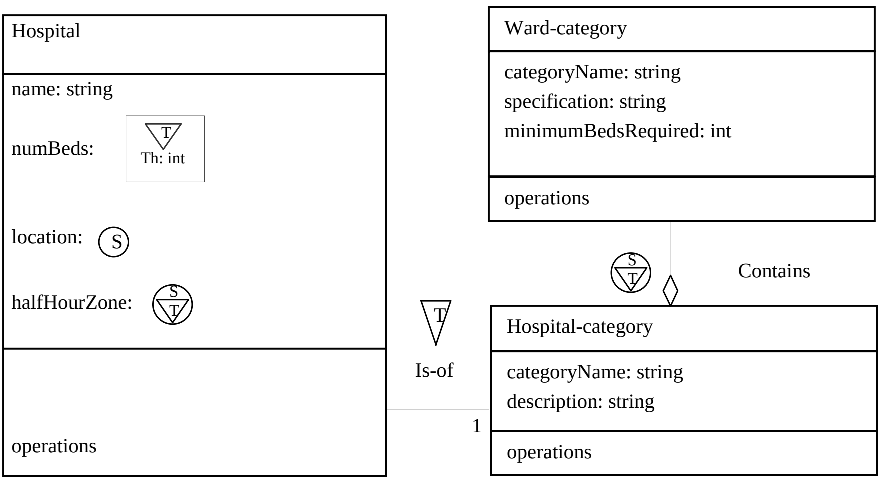

Fig. 3.25. Using extended spatio-temporal UML at the association level

A spatially-dependent association is one where an association instance is associated with a spatial extent to show where that instance is valid. For example, the same category of hospital may require different categories of wards in different areas depending on local regulations. Therefore, the Contains aggregation association must be spatially-dependent. In fact, since the local categories may also change over time, the Contains aggregation association is actually spatio-temporally-dependent. In this case, both of the associated object classes are purely conceptual. An association between two physical object classes can also be spatio-temporally-dependent; e.g., a consultation of a ward doctor with a specialist is scheduled for a specific location and period of time in the hospital.

Since constraints are often application-dependent, no constraints are assumed between the timestamps or spatial extents of participating objects with those of a temporally, spatially or spatio-temporally-dependent association. They can be specified explicitly for individual applications either (1) on an ad hoc basis as required using UML constraints or (2) by defining explicit modeling constructs for commonly used constraint patterns. The latter approach is illustrated by the introduction of the existence-dependent symbol, described in the next section, to support the semantics of temporal-dependency in composite model elements. Explicit modeling constructs for common spatial constraints have been defined in [27].

3.4.4 Additional Constructs: Specification Box, Existence Time, and Groups

The previous section described the different types of timestamps that can be associated with an attribute, association, or object class: but where do we specify which types are required for a given application? Detailed spatio-temporal semantics are specified in a specification box, which can be associated with any of the icons or combinations using a unique naming convention (used here) or label. The specification box includes information on the time units and the time and space dimensions, models, and interpolation. Users can specify regular (recorded at regular intervals) or irregular time models and object- or field-based space models. Interpolation functions can be specified to derive values in-between those explicitly recorded for spatially, temporally, and spatio-temporally-dependent thematic attributes. The time dimensions and units (i.e., instant, interval, element) used are defined in [19]. Specification boxes can be inherited from parent classes as with any other class property. The specification box syntax is illustrated in Figure 3.26.

```
SPECIFICATION BOX <Identifier>:
TimeDimen. ::= [existence | valid ] [ transaction ]
TimeInterpolation ::= discrete | step | min | max | avg | linear | spline | <user-defined>
TimeModel ::= irregular | ( regular {<frequency> [,<beginning>,<end> ]} )
TimeUnit [ (<TimeDimen>) ] ::= instant | interval | element
SpaceInterpolation ::= <same as TimeInterpolation>
SpaceModel ::= '('<max object/field dim>,<max search space dim> ')': object | field
Group ::= independent | (dependent (formula)* )]
```

Fig. 3.26. Specification box syntax in extended spatio-temporal UML

Time dimensions include existence time (for objects), valid time (for attributes and associations), and transaction time (for objects, attributes, or associations), as defined in [19]. However, object existence time is more precisely defined as the time during which existence-dependent attributes and associations can be defined (i.e., have legal values) for that object. In other words, existence-dependent attributes (e.g., person's home phone number) and associations are those that are defined only when the related object(s) (e.g., the person) exist. This implies that attributes and associations that are not existence-dependent (e.g., a person's social-security number) may be defined even when the related object(s) no longer exist. Object identifiers can never be existence-dependent, as they can be used to refer to historical objects. The superscript ED is added to those attributes and associations defined only when the corresponding object (or objects, in the case of an association) exists, which means that existence time must be recorded for the object (or objects). This further implies that any valid timestamps recorded for the existence-dependent attributes and associations must be within the existence time of the corresponding object(s).

The time model applies only to valid time and time interpolation only to the valid time of temporally-dependent thematic attributes. Space dimensions include the dimensions of the spatial extent(s) being specified, followed by the dimensions of the underlying search space. The object-based spatial model is used for a spatial attribute, where an attribute instance consists of a single spatial extent. The field-based spatial model is used for a spatially-dependent thematic attribute; where an attribute instance is a set of thematic values, each associated with a different spatial extent. Space interpolation applies only to spatially-dependent thematic attributes.

The specification box can also be used to specify spatio-temporal constraints, including those within an attribute group. The group symbol is used to group attributes sharing the same timestamps or spatial extents, thus requiring only one specification for the group. Thus, this symbol graphically illustrates associated sets of attributes and ensures that spatial extents and timestamps are not specified redundantly. Note that a group's attributes never share thematic values, even if the thematic symbol is used in the group specification. If the group's attributes have different thematic domains, then these can be indicated next to each attribute using standard UML text notation.

Following UML convention, another compartment is added to the object class to accommodate its specification boxes, i.e., the specification compartment. The specification compartment can be used to specify spatio-temporal semantics for the object, its attributes, and any associations in which the object class participates. Alternatively, a specification compartment can be added to an association class to specify spatio-temporal semantics for that association and its attributes (see [26,23] for further discussion of the specification box).

3.4.5 Example of Usage

Figure 3.27 shows the same full regional health application described previously as it would be represented using the proposed extension and illustrates the use of the specification box, group symbol, and existence-dependent symbol.

For example, Hospital location is specified as a single point in the 2D space. Hospital halfHourZone and Contains are specified as a region in 2D space. In contrast, the Province populationDensity and averageLifespan group is associated with a 2D field in 2D space. This means that, for a single object instance, the two attributes in the group are associated with a set of regions and have a separate attribute value for each region for a given point in time. Since these two attributes share common timestamps and spatial extents, they are grouped. Since both attributes are integers, we can specify the thematic domain in the group symbol. If the attributes had different thematic domains, then we would specify them for each attribute rather than for the group.

The group is then associated with a single symbol and specification box. Here we specify that any attribute in the group uses average interpolation in time and no interpolation in space, has a valid time dimension using instant as the time unit, and is measured yearly (i.e., a new set of values is recorded for the attribute each year). This means that the population density and average lifespan between recorded time instants is assumed to be the average of the values at the two nearest time instants and undefined outside of recorded spatial regions. No inter-attribute constraints are defined for the group, as shown by the keyword independent.

The temporal symbol at the Hospital object level is used to indicate existence and transaction time. Existence time is used to model the periods when the hospital is open, i.e., when the existence-dependent numBeds, halfHourZone and Is-of are defined. Since these model elements are temporally-dependent, the valid timestamps of all their instances must be included within the Hospital existence time. Attribute numBeds is specified as irregular because this attribute is not recorded periodically: whenever it changes the new value is recorded.

The specification box for an association (e.g., Is-of) can be placed in the specification compartment of either of its participating object classes (e.g., Hospital or Hospital-category). Note that since Hospital-category is not temporal and therefore does not have existence time defined, the only constraint on the valid-time timestamps of the Is-of association comes from the Hospital class existence time.

3.5 Related Work

A range of aspects of spatial and temporal databases have been studied in isolation for more than a decade. Only more recently has the combination of spatial and temporal data, i.e., spatio-temporal data, been subject to scrutiny.

Story et al. [28] propose a design support environment for spatio-temporal databases, focusing on the integration of time with application data. Temporal classes, events, and states are emphasized as components of an ideal environment that supports the developer's better understanding of spatio-temporal data and applications. Claramunt and Theriault [7] integrate time in GIS's by presenting a systematic typology of spatio-temporal processes, leading to an event-oriented model. Allen et al. in [1] present a generic model consisting of objects, states, events, and conditions for explicitly representing links within a spatio-temporal GIS. This work focuses on the interaction among the com-

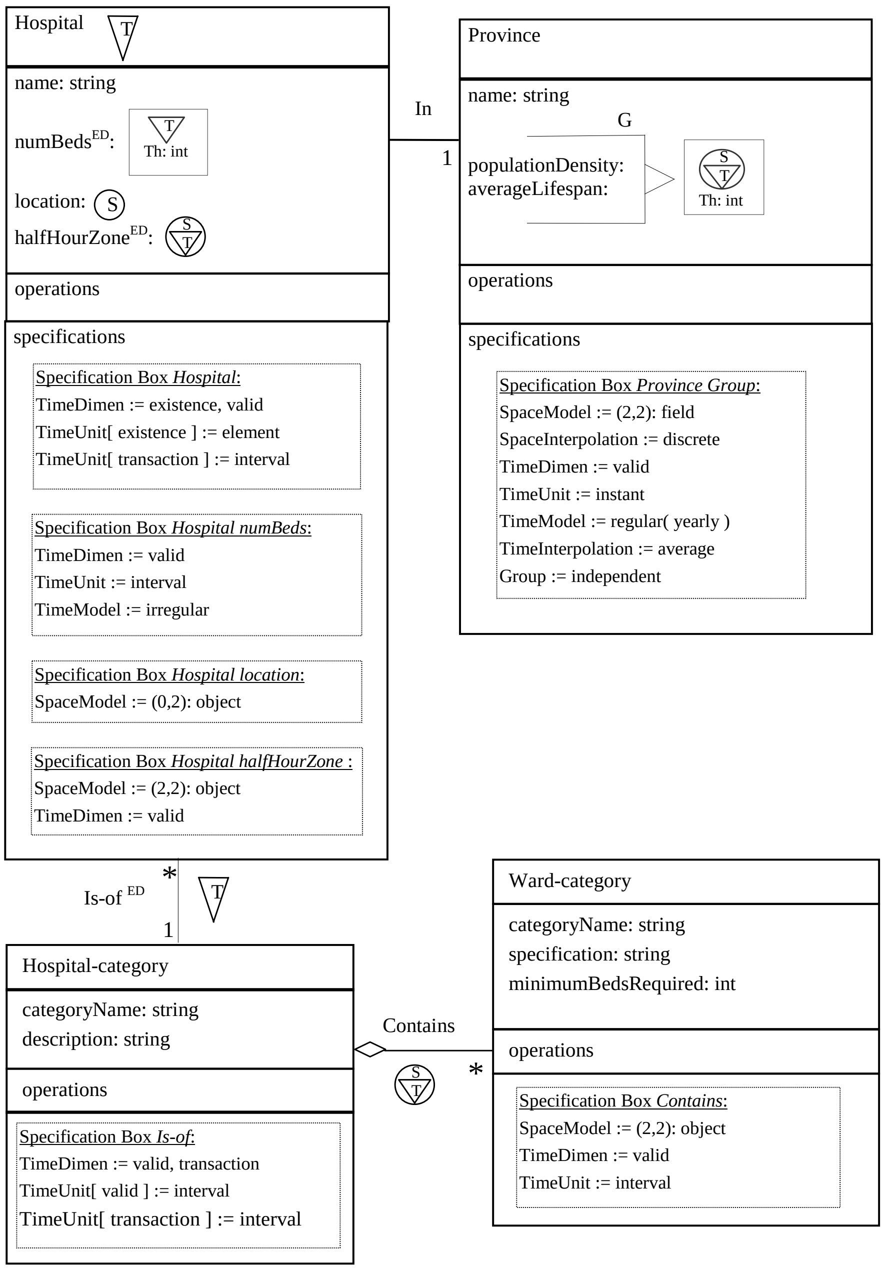

Fig. 3.27. Regional health application in extended spatio-temporal UML

ponents over time rather than on presenting them as the appropriate mechanisms for representing spatio-temporal data. Worboys in [37] proposes a unified model for spatio-temporal data with two spatial and two temporal dimensions (database and event times). This model is not aimed at conceptual design, but is a mathematically-oriented model. Claramunt et al. [25] present a set of design patterns for spatio-temporal processes expressed in an object-relationship data model. The focus is on the analysis of spatio-temporal processes and on the properties of object-oriented and entity-relationship data models. This work focuses on the design of processes rather than on the design of modeling constructs. Moreover, spatial issues such as the representation of time-varying fields (or spatial attributes) are not considered. Langran in [20] provides a broad survey of research on spatio-temporality, the philosophy of time, temporal databases, and spatial data structuring. Finally, mathematical models have been developed based on a systems perspective [29] with the purpose of understanding and, sometimes, predicting a wide variety of natural phenomena. Such models almost always incorporate notions of change over time.

Becker et al. and Faria et al. [5,13] propose OO models based on extensions of ObjectStore and O2 respectively. [5] considers both object- and field-based spatial models, defining a hierarchy of elementary spatial classes with both geometric and parameterized thematic attributes. Temporal properties are incorporated by adding instant and interval timestamp keywords to the query language. In [13], spatial and temporal properties are added to an object class definition by associating it with pre-defined temporal and spatial object classes. This solution is not suitable for representing temporal or spatial variation at the attribute level, as the timestamp and spatial locations are defined only at the object component level. In addition, both offer text-based query languages; the non-graphical query languages of these models reduce their suitability as conceptual modeling languages.

Perceptory [3] is a spatio-temporal model aligned with international geographic standards. It is a CASE tool that stores the conceptual content of object-oriented application schemas and dictionaries. The basic approach of the underlying model is to add spatial and temporal stereotypes to objects in UML. The emphasis is on supporting geographic objects rather than fields or associations between objects.

The MADS model [24] extends an object-based model with pre-defined hierarchies of spatial and temporal abstract data types and special complex data types to describe all of an attribute's properties, i.e., name, cardinality, domain, and temporal or spatial dimensions. The use of a non-standard, hybrid ER/OO model and the definition of new composite data structures for spatio-temporal properties, rather than exploiting existing features of the ER or OO models, increase the complexity of the model syntax. There is no provision for attributes having a spatial domain, therefore any data element associated directly with several different spatial extents must be modeled as an association of spatial objects.

3.6 Conclusions

In this chapter, we view the design of spatio-temporal applications from the perspective of information systems development. STER and Extended spatio-temporal UML focus on the special modeling needs of spatio-temporal applications at the conceptual phase. They both provide a small set of new constructs for spatio-temporal data and thus reduce the complexity of the resulting schemas. This allows the application developer to concentrate on the characteristics of the specific application domain of interest.

Both models facilitate the integration of two disjoint views [36] of spatial information, namely the (a) field-based view, suitable for the representation of properties like "temperature" or "vegetation" and (b) the object-based view, which captures spatial information over time in terms of identified objects, like "landparcel."

STER was applied in real, large-scale applications ([35] for the development of a utility network management system; [22] for the design of the Greek cadastral system), as well as in small prototypical examples with quite encouraging results: diagrams were more easily understood by the users, and the data modeling process proceeded more rapidly.

In STER, topological relationships between spatial objects can be explicitly represented in the schema diagram using spatial relationships. STER also addresses the problem of modeling multiple granularities.

In contrast, Extended STUML supports associations having their own spatial extent separate from those of participating objects, e.g., the location of an accident involving two cars, using spatially-dependent associations. It further provides formal mathematical definitions of the modeling constructs and offers a more detailed representation of the spatio-temporal semantics using the specification box, which provides a standard notation for their description, including interpolation and alternative time models (e.g., periodic versus aperiodic recording of data values). This facilitates effective communication and consistent design documentation. Analogously, the use of an explicit existence dependent construct, rather then UML constraints or notes, to represent application defined temporal dependencies between associations, attributes, and objects further contributes to standardization and consistency.

References

1. E. Allen, G. Edwards, and Y. Bedard. Qualitative casual modeling in temporal gis. In *International Conference on Spatial Information Theory (COSIT)*, volume 988 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995.
2. A. Abiteboul and R. Hull. Ifo: A formal semantic database model. *ACM TODS*, 12(4):525–565, 1987.
3. J. Brodeur, Y. Bedard, and M.J. Proulx. Modeling geospatial application databases using uml-based repositories aligned with international standards in geomatics. In *Eighth International Symposium of ACMGIS*, pages 39–46, Washington DC, November 2000.

4. G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.
5. L. Becker, A. Voigtmann, and K. Hinrichs. Temporal support for geo-data in object-oriented databases. In *International Conference on Database and Expert Systems (DEXA)*, volume 1134 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.
6. P.S. Chen. The entity-relationship model: Toward a unified view of data. *ACM TODS*, 1(1):9–36, 1976.
7. C. Claramunt and M. Theriault. Managing time in gis: An event-oriented approach. In *International Workshop on temporal Databases*, Zurich, September 1995.
8. V. Delis, T. Hadzilacos, and N. Tryfona. An introduction to layer algebra. In *6th International Conference on Spatial Data Handling (SDH)*. Taylor and Francis, 1995.
9. B. David, M. van den Herrewegen, and F. Salge. *Geographic Objects with Indeterminate Boundaries*, chapter 13. Taylor and Francis, 1996.
10. M.J. Egenhofer and R. Franzosa. Point-Set Topological Spatial Relationships. *International Journal of GIS*, 5(2):161–174, 1991.
11. M.J. Egenhofer and A.U. Frank. Connection between Local and Regional: Additional Intelligence Needed. In *18th International Congress of FIG*, Toronto, 1986.
12. R. Elmasri and S. Navathe. *Fundamentals of Database Systems.* Benjamin/Cummings, 1994.
13. G. Faria, C.B. Medeiros, and M.A. Nascimento. An extensible framework for spatio-temporal database applications. Technical report, TimeCenter, 1998.
14. A.U. Frank. Qualitative Spatial Reasoning: Cardinal Directions as an Example. *International Journal of GIS*, 10(3):269–290, 1996.
15. A.U. Frank, J. Raper, and J.P. Cheylan. Life and Motion of Socio-Economic Units. ESF Series, Taylor and Francis, 2000.
16. M. Fowler and K. Scott. *UML Distilled*. Addison-Wesley, 1997.
17. Object Management Group. Omg unified modeling language specifications, version 1.3. In `http://www.ftp.omg.org/pub/docs/ad/99-06-08.pdf`, pages 1–808, World Wide Web, Retrieved 19 May, 2000 1999.
18. B. Henderson-Sellers and F. Barbier. Black and white diamonds. In *Conference on the Unified Modeling Language*, pages 550–565, 1999.
19. C.S. Jensen and C.E. Dyreson (with multiple contributors). The consensus glossary of temporal database concepts. In *Temporal Databases: Research and Practice*, pages 367–405, 1998.
20. G. Langran. *Time in Geographic Information Systems.* Taylor and Francis, 1993.
21. D.R. Montello. The geometry of environmental knowledge. In *International Conference on Spatial Information Theory (COSIT'92)*, volume 639 of *Lecture Notes in Computer Science*, 1992.
22. National Technical University of Athens NTUA. Definition of a standard for the exchange of digital cadastral data. Technical report, National Technical University of Athens, 1997.
23. R. Price, B. Srinivasan, and K. Ramamohanarao. Extending the unified modeling language to support spatiotemporal applications. In *Conference on Technology of Object Oriented Languages and Systems, Asia*, pages 163–174, Melbourne, Australia, November 1999.
24. C. Parent, S. Spaccapietra, and E. Zimanyi. Spatio-temporal conceptual models: Data structure+space+time. In *Seventh International Symposium of ACMGIS*, Dallas, Texas, November 1999.
25. C. Claramunt C. Parent and M. Theriault. Design patterns for spatiotemporal processes. In *IFIP*. Chapman and Hall, 1997.

26. R. Price, N. Tryfona, and C.S. Jensen. Extended spatiotemporal UML: Motivations, requirements, and constructs. *Journal of Database Management*, 11(4):14–27, October/December 2000.
27. R. Price, N. Tryfona, and C.S. Jensen. Modeling part-whole relationships for spatial data. In *Eighth International Symposium of ACMGIS*, pages 1–8, Washington DC, November 2000.
28. P.A. Story and M. Worboys. A design support environment for spatio-temporal database applications. In *International Conference on Spatial Information Theory (COSIT)*, volume 988 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995.
29. N. Roberts, D. Andersen, M. Garet, and W. Shaffer. *Introduction to Computer Simulation: A System Dynamics Modeling Approach*. Addison-Wesley, 1983.
30. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object-Oriented Modeling and Design*. Prentice-Hall, 1991.
31. J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999.
32. M. Raubal and M. Worboys. A formal model of the process of wayfinding in built environments. In *International Conference on Spatial Information Theory (COSIT'99)*, volume 1626 of *Lecture Notes in Computer Science*, 1999.
33. N. Tryfona and C.S. Jensen. Conceptual data modeling for spatiotemporal applications. *Geoinformatica*, 3(3), 1999.
34. N. Tryfona and C.S. Jensen. Using abstractions for spatio-temporal conceptual modeling. In *14th Annual Symposium on Applied Computing (SAC 2000)*, 2000.
35. UTILNETS. A network unitlity management system. Technical report, European Commission, Brite-EURAM project 7120, 1994.
36. M. Worboys. Object-oriented approaches to geo-referenced information. *International Journal of GIS*, 8(4), 1994.
37. M. Worboys. A unified model for spatial and temporal information. *The Computer Journal*, 37(1), 1994.