A Conceptual Modeling Language for Spatiotemporal Applications

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

[1] Monash University, Australia, {rosanne.price}@csse.monash.edu.au

[2] Aalborg University, Denmark, {tryfona, csj}@cs.auc.dk

Abstract. This paper presents a conceptual modeling language for spatiotemporal applications that offers built-in support for capturing geo-referenced, time-varying information. More specifically, the well-known object-oriented Unified Modeling Language (UML) is extended to capture the semantics of space and time as they appear in spatiotemporal applications. Language clarity and simplicity is maintained in the new language, the Extended Spatiotemporal UML, which introduces a small base set of modeling constructs, namely, the spatial, temporal and thematic constructs, which can then be combined and applied at different levels (i.e., attribute, association, object class) in the object-oriented model. An example is used to illustrate the simplicity and flexibility of this approach, and a formal functional specification of the semantic constructs and their symbolic combinations is given.

Keywords: Spatiotemporal, Conceptual Data Modeling, Unified Modeling Language

1. Introduction

Spatiotemporal applications, such as navigational systems or cadastral systems, have been the focus of considerable attention recently. The need for a historical dimension in traditional spatial information systems and for high-level models useful for the conceptual design of the resulting spatiotemporal systems have become clear. Providing design support for applications dealing with space and time requires both an understanding of the underlying semantics and a conceptual data modeling language that can be easily used to express these semantics. Although having in common a need to manage *spatial data* and their *changes over time*, various spatiotemporal applications may manage different types of spatiotemporal data and may be based on very different models of space, time, and change. For example, the term *spatiotemporal* data is used to refer both to *temporal changes in spatial extents*, such as redrawing the boundaries of a voting precinct or land deed, and to *changes in the value of thematic* (i.e., *alphanumeric*) *data across time and space*, such as variation in soil acidity measurements depending on the measurement location and date. A spatiotemporal application may be concerned with either or both data types; this, in turn, is likely to influence the underlying model of space employed, e.g., the two types of spatiotemporal data generally correspond to an object- versus a field-based spatial model. Another type of spatiotemporal data is composite data whose components vary depending on time or location. An example is the minimum combination of equipment and wards required in a certain type of hospital (e.g., general, maternity), where the relevant regulations determining the applicable base standards vary by locality and time period. The conceptual data modeling language should provide a clear, simple, and consistent notation to capture alternative time, space, and change semantics, such as object- and field-based spatial models; continuous, instantaneous, and discrete views of change (i.e., support alternative time models, time units, and interpolation); and multiple dimensions for space and time.

Although there has been considerable work in conceptual data models for time and space separately, interest in providing an integrated spatiotemporal model is much more recent. Requirements are discussed in more detail in [Pfo98] and spatiotemporal data models are surveyed elsewhere [Abra99, Pav98, Price99a], including lower level logical models [Clar95, Lang93, Pequ95, Worb95]. Several conceptual frameworks have been designed to integrate spatial, temporal, and thematic data based on Entity-Relationship (ER) or Object-Oriented (OO) data models that include a high-level query language capable of specifying spatiotemporal entity types. The data definition component of these query languages thus has some potential for use in modeling spatiotemporal applications.

[Beck96] and [Fari98] have OO models based on extensions of ObjectStore and O2, respectively. [Beck96] considers both object- and field-based spatial models, defining a hierarchy of elementary object-based and field-based 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 [Fari98], 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 [Beck96] and [Fari98] have text-based query languages; the non-graphical query languages of these models reduce their suitability as conceptual modeling languages. [Dion98] proposed an extended ER model with multimedia and stream types representing sequences of timestamped entities. The only spatial data considered is that contained in image raster data, e.g., spatial relationships between objects in an image. While [Dion98] has the advantage of having a graphical query language based on PICQUERY, the model does not consider general spatial data or spatial variation of thematic data. Of the languages proposed, only [Beck96] supports multiple spatial models or temporal dimensions.

The MADS model [Par99] 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 complex constructs, rather than exploiting existing features of the ER or OO models, render this model somewhat complex.

In two of the few papers to specifically address conceptual modeling languages for spatiotemporal applications, [Tryf99a, Tryf99b] propose the SpatioTemporal ER model (STER) that adds temporal and spatial icons to entities, attributes, and relationships, to support timestamped spatial objects and layers. Composite data, whose components vary over space, are not considered: instead, spatial relationships are used to represent explicit geometric or topological relationships between associated spatial objects, which could otherwise be derived on demand.

An earlier spatiotemporal extension to the Unified Modeling Language (UML) is presented in [Price99b]. However, neither the syntax nor the semantics of the symbols introduced there are presented formally. Furthermore, the semantics adopted appear difficult to formalize, as different rules for combining the new symbols are used for the three different types of spatiotemporal data discussed earlier.

In this paper, we propose an extension of UML intended to address the goals outlined earlier, i.e., to support a range of spatiotemporal models and data types using a clear, simple, and consistent notation and overcome the limitations of the models described above. More specifically, we introduce a small base set of modeling constructs, namely, the spatial, temporal, and thematic constructs. These can then be combined and applied to different levels of the object-oriented model in a consistent manner guided by the same simple principles. The result is the *Extended Spatiotemporal UML*. A formal functional specification of the semantic constructs and symbolic combinations is given.

Extending the OMG standard for object-oriented modeling, UML was selected as the best approach to providing a language that has a high level of acceptance, understandability, and the flexibility needed for extensions. Adopting a widely-used industry standard as the basis of a spatiotemporal modeling language reduces the learning curve, facilitates later stages of development, since well-established tools are already available which automatically translate UML designs to database schemas, and facilitates compatibility and thus integration of spatiotemporal data with an organization's conventional applications. Finally, the graphical modeling language of UML has the advantage of being easier to understand than text- or logic-based equivalents and thus can be used as a basis of communication between database designers and domain experts. It should be noted that among the many advanced features in UML, we consider only those that help us to better capture the spatiotemporal semantics; other UML features can still be used in the conventional way for conventional data.

The rest of the paper is organized as follows. Section 2 presents an overview of UML and describes a spatiotemporal example, which will be used throughout the paper. Section 3 describes the syntax and semantics of the new language's fundamental modeling constructs—the spatial, temporal, and thematic symbols. Section 4 discusses two other symbols—the group symbol and the existence-dependent symbol—and the construct used to specify the details of the spatiotemporal semantics, the specification box. Section 5 offers conclusions and future directions.

2. UML and Spatiotemporal Data Modeling

UML [Booc99] consists of (a) nine types of diagrams specifying structure or behavior of a system and its data elements, (b) notational conventions based on the OO paradigm, and (c) built-in extension mechanisms for supplementing UML's core meta-model. Class diagrams and their specialization, Object diagrams, describe the structure of classes and object instances, respectively. Component diagrams describe implemented system software component structure and dependencies. Deployment diagrams describe the hardware and software configurations of delivered systems. Behaviors within the business process model are described by Use Case (system functionality), Sequence, and Collaboration diagrams. The latter two diagrams give a chronological versus procedural view of object interactions within a single system function. Class behavior is specified by State and Activity (also used for business systems) diagrams for external and internal events, respectively.

The diagram most relevant to the support of spatiotemporal data modeling is the Class diagram, illustrated in Figure 1, since it captures the static structure of a system.

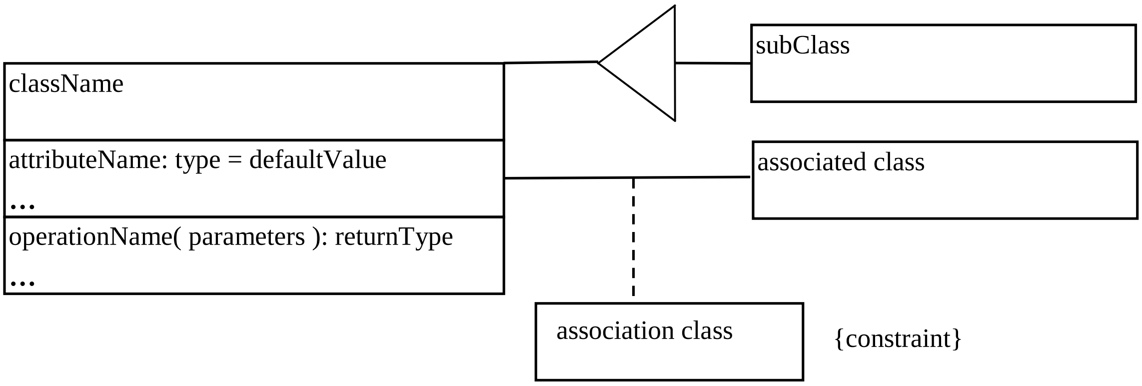


Figure 1: Class Diagram

The fundamental element of this diagram is the object class description, consisting of the class name, attribute descriptions, and operation signatures. Attribute descriptions consist of the name, types[1], and default values for class attributes. Operation signatures consist of the name, parameters, and return values for class operations. Visibility (scope) can be indicated for attributes or operations. Classes can be connected by different types of standard OO links, including inheritance (sub-classes defined based on a super-class) and association (structural relationships between different object classes). Constraints may be enclosed in braces and placed next to the model elements to which they relate.

Associations are similar to attributes, except that the domain is a separately defined class and typically uses reference (i.e., identifiable and shared instances) rather than value (i.e., owned and copied instances) data semantics. An association is usually represented by a line between classes. However, an association that has properties of its own is represented by *promoting* the association to an association class that is then connected by a dotted line to the original association line (see Figure 1). Specializations of associations with specific properties exist; however, there is no need to distinguish between them here.

Representation of spatiotemporal concepts using the core constructs of UML is not straightforward. One option would be to promote each spatiotemporal attribute of a class to a separate, but associated class with the appropriate timestamps and spatial extents as attributes. Temporal classes and associations could be treated similarly, by adding timestamp attributes to the class or to the association class respectively, after promoting the association to an association class in the latter case.

The following example will be used to illustrate the use of UML to model spatiotemporal data. Assume an application measuring health statistics of different states, in terms of average lifespan, as related to the location, size (i.e., number of beds), accessibility, and surrounding population densities of state hospitals of different types. The hospital type is defined as having a minimum number of beds in specific types of wards: however, hospital type definitions may differ between regions due to local regulations (e.g., district regulations). Accessibility is assessed by defining a zone around a hospital that represents travel times to the hospital of no more than half an hour.

Some data values may change over time: we want to record information about the time periods when a given value is valid (i.e. valid time) or current (i.e. transaction time). For example, population densities and average lifespans across states vary with respect to time and space, where both values are recorded yearly at the same time and for the same regions. The number of beds, the half-hour travel zone, and the hospital type for a hospital can change over time. Hospital type definitions 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 type, and valid time for everything else. We assume that hospital name and location are static, as are hospital type name and description and ward details.

Hospitals may sometimes be closed and later re-opened depending on changes in local population density. The number of beds, half-hour travel zone, and associated hospital type are only defined when the hospital is open; however, the name and location of a hospital are defined even when the hospital is closed.

Two alternative UML representation styles are possible for time or space dependent properties. This is illustrated in Figure 2 using a time-dependent property, a hospital's half-hour travel zone, where the definitions of timestamp and spatial extent types are given below in Backus-Naur Form (BNF).

timestamp := { instant | interval | element }   spatial-extent := { point | line | region | volume }^{n}

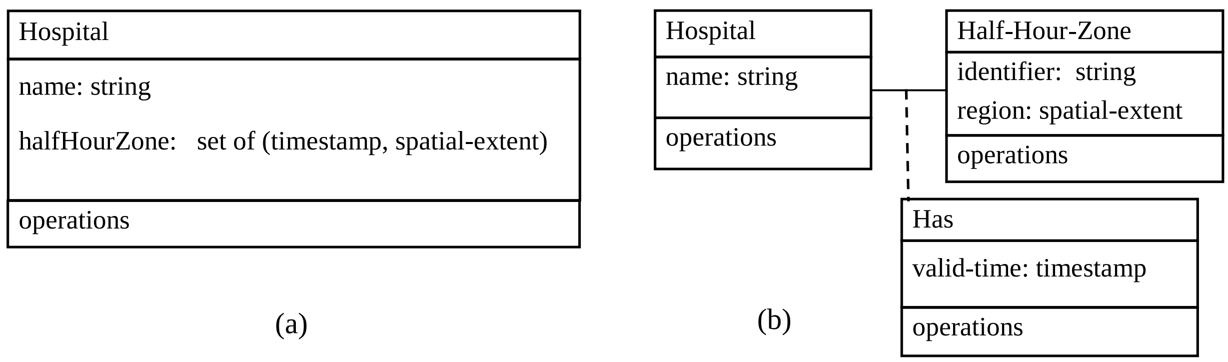


Figure 2: Alternative UML Representations of the Time-Varying Property, Half-Hour Zone

Since attributes are not limited to having atomic domains in the object-oriented model, one option for representing a time-dependent property, shown in Figure 2 (a), is to define an attribute with a composite domain consisting of a *set of* timestamp and property value pairs. The other approach is to define a separate class, whose association with the original class has been promoted to an association class with timestamps. As illustrated in Figure 2 (b), this complicates the diagram and requires the creation of an artificial identifier for the associated object class instances, e.g., for the *halfHourZone* class. However, this approach is preferred when several attributes are associated with the same timestamps or spatial extents, e.g., population density and average lifespan.

[1] The term *domain* is used as an equivalent to *type* in this paper.

The complete UML class diagram for the regional health example is shown in Figure 3. As illustrated in Figure 2 (a), composite attribute domains are used for the number of beds and half-hour travel zone. However, extra associated and association classes (specifically, the *Measurement-Region* class and *Has* association class, respectively) are used for population density and average lifespan. Figure 3 also illustrates that a new association class must be created for each association with spatial or temporal properties. Additionally, constraints are used to indicate which attributes of hospital are defined only during the periods when it is open.

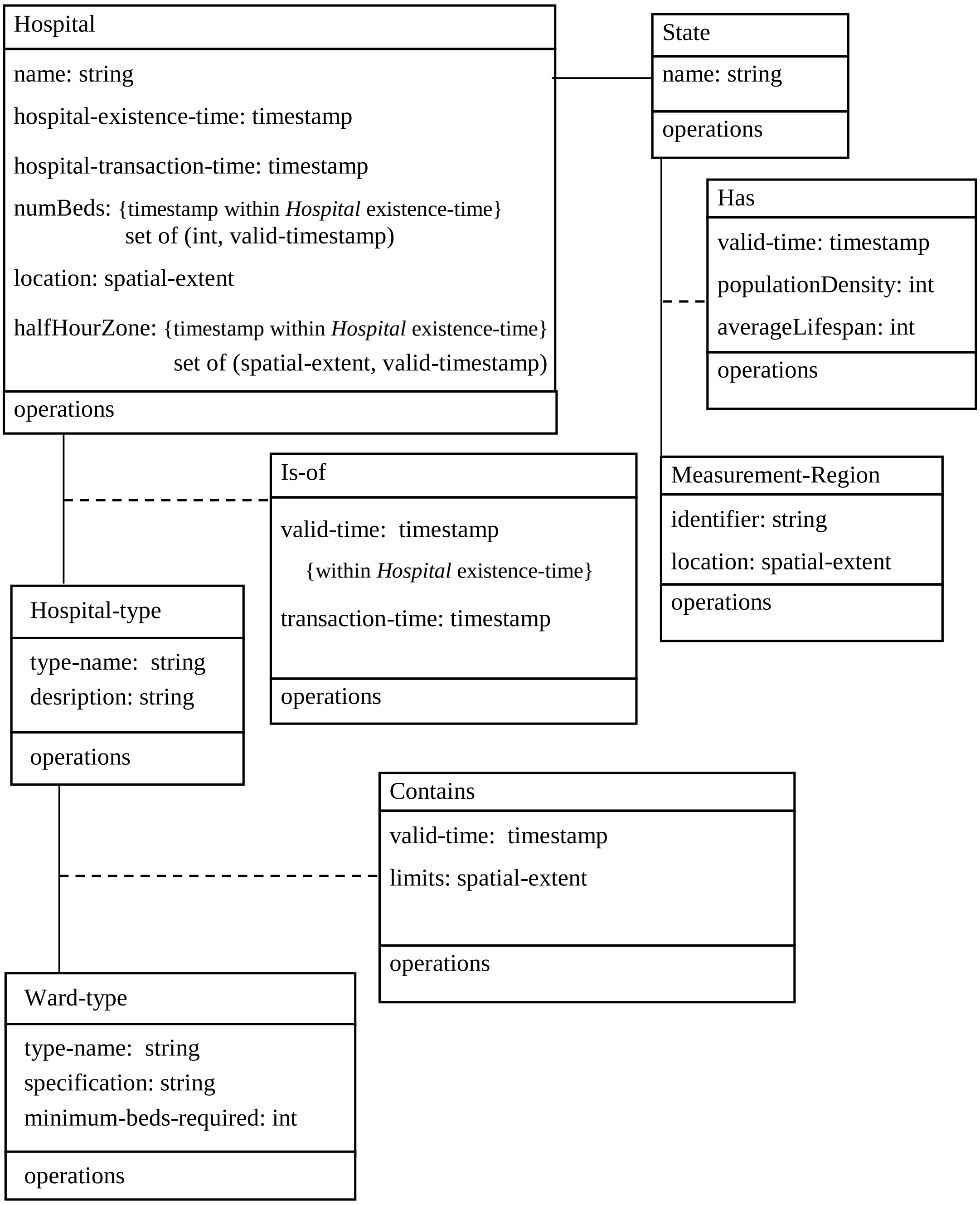


Figure 3: A Spatiotemporal Health Application Example in UML

As 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 indicate temporal or spatial variation. This violates the requirement that the notation be simple, clear, and consistent.

An advanced feature of UML is stereotypes. These are generally used to provide domain-specific extensions to UML core constructs, i.e., to add new constructs, which can be used as were they part of UML's original meta-model or definition [Booc99]. A set of standard stereotypes has been defined for UML, but none is defined as applying both to attributes and composite elements (i.e., classes and associations) or used for spatial and temporal properties. [Fowl97] suggests using a *history* stereotype to model historical associations between classes by adding a temporal subtype to one of the classes. But then this would seem to imply that a new stereotype should be added for each different level of granularity, and this not account for spatial and spatiotemporal attribute variation. A better approach is to extend the fundamental characteristics of the existing building blocks, such as classes, attributes, and associations, to meet the spatiotemporal requirements discussed previously. New "stereotypes" that consider the spatial and temporal properties at different levels of granularity are required: this motivates the Extended Spatiotemporal UML.

3. Extended Spatiotemporal UML: Spatial, Temporal, and Thematic Symbols

The five modeling constructs used in the Extended Spatiotemporal UML are illustrated in Figure 4. The basic approach is to extend UML by adding a minimal set of constructs for spatial, temporal, and thematic (i.e., alphanumeric) data, represented respectively by *spatial, temporal,* and *thematic* symbols[2]. These constructs can then be applied at different levels of the UML class diagram (i.e., object classes, attributes, and associations) and in different combinations to add spatiotemporal semantics to a UML model element. The *group* symbol, used to group attributes with common spatiotemporal properties or inter-attribute constraints, and the *existence-dependent* symbol, used to describe attributes and associations dependent on object existence, are discussed in section 3.2. The *spatial, temporal,* and *thematic* symbols are described in section 3.1.

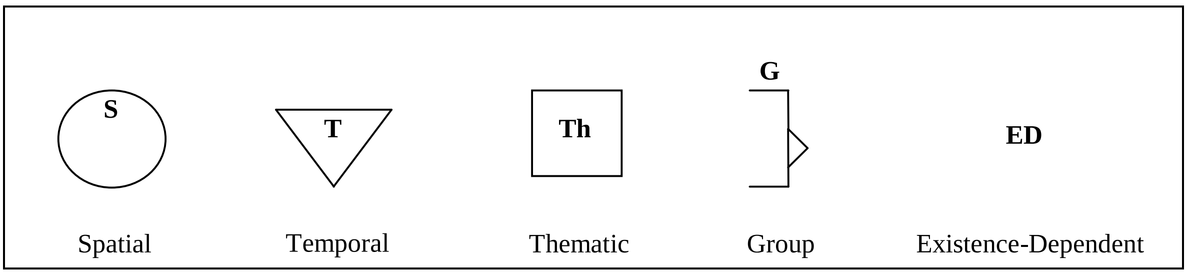


Figure 4: Extended Spatiotemporal UML Symbols

[2] The *S, T,* and *Th* letters inside the spatial, temporal, and thematic symbols are optional.

3.1 Spatial, Temporal, and Thematic Constructs

To understand the use and semantics of the *spatial*, *temporal*, and *thematic* constructs, we first discuss the interpretation of each individual corresponding symbol separately. The *spatial* symbol represents a spatial extent, which consists of an arbitrary set of points, lines, regions, or volumes. The *temporal* symbol represents timestamped data, i.e., the temporal dimension of 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 specific alphanumeric domain can be optionally indicated in the symbol, e.g., *Th: integer*.

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., the attribute name followed by a colon and the name of the attribute's thematic domain. For instance, Figure 5 illustrates the four possible cases for a thematic attribute: an attribute with a thematic domain and (a) no spatial or temporal properties, (b) temporal properties, (c) spatial properties, or (d) spatiotemporal properties.[3] We term such attributes (a) thematic, (b) temporally-dependent, thematic, (c) spatially-dependent thematic, or (d) spatiotemporally-dependent thematic.

The semantics of Extended Spatiotemporal 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. Nesting one symbol inside another represents mathematically a function from the inner to the outer domain. Placing one symbol next to another symbol represents mathematically two separate functions, one for each symbol. In fact, the expressiveness of Extended Spatiotemporal UML is based on the application of the symbols in a consistent and intuitive manner to different levels of the model and in different combinations.

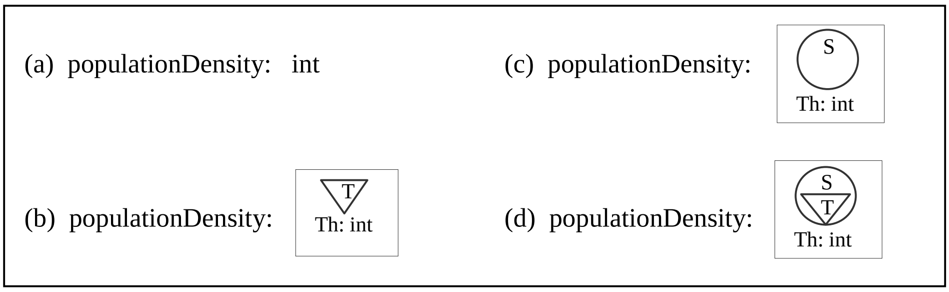


Figure 5: Thematic Attribute Examples

We now give the symbolic combinations legal at each model level, the semantic constructs defined at each level, and a mapping between the two using BNF notation. Textual and mathematical definitions are given for each semantic construct in the new language. Illustrative examples are given for each level.

[3] In fact, this can be considered a composite attribute domain consisting of a tuple with the thematic value, timestamp, and spatial-extent.

Some of the semantic constructs correspond to more than one symbol combination and thus corresponding formal definition, as variations in the order of nesting symbols does not change the database design semantics described, although they may have implications for query semantics and use in a query language. For example, if Figure 5(d) is modified by nesting the spatial symbol inside the temporal symbol for *populationDensity*, each thematic attribute integer value is still associated with a timestamped spatial extent; however, the perspectives differ. Where several different symbol combinations correspond to a single semantic construct, the first symbol combination listed is the one used as a standard for the purposes of database design in this paper. The mathematical definitions are given in terms of curried functions. We first summarize the primitive BNF symbols for time, space, and model elements used in the definitions.

<T> ::= domain of time instants
<2^T> ::= arbitrary set of time instants
<S> ::= domain of points in space
<2^S> ::= arbitrary set of points in space
<oid> ::= domain of object-identifiers
<aid> ::= domain of association-instance identifiers, essentially { <oid> }^{n}
<D> ::= thematic, i.e. alphanumeric, domain (e.g. integer, string)
<d> ::= thematic attribute symbol
<t> ::= temporal symbol
<s> ::= spatial symbol
<st> ::= any nested combination of a spatial and a temporal symbol
<sd> ::= any nested combination of a spatial and a thematic symbol
<td> ::= any nested combination of a temporal and a thematic symbol
<std> ::= any nested combination of a spatial, a temporal, and a thematic symbol
<ED> ::= existence-dependent symbol

3.2 The Attribute Level

A thematic attribute with no spatial or temporal properties is represented using standard UML, whereas spatial attributes and any thematic attributes with spatial or temporal properties use symbols of the Extended Spatiotemporal UML. Legal combinations of these symbols at the attribute level are any nested combination of a spatial symbol, a temporal symbol, and/or a thematic symbol. The only exception to this rule is that the temporal symbol cannot be used alone, i.e., an attribute with a temporal domain is treated as thematic data. The attribute domain can optionally be followed by an *existence-dependent* symbol (discussed in Section 4). The rule for the Extended Spatiotemporal UML notation at the attribute level can be defined in BNF as:

attribName: [<D> | <sd> | <td> | <std> | <s> | <st>] [<ED>]

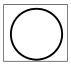
There are six different attribute domains possible in the Extended Spatiotemporal UML. These correspond to the six semantic categories of attributes (i.e., six semantic modeling constructs). Reading the domain symbols left to right, we have: *thematic attributes, spatially-dependent thematic attributes, temporally-dependent thematic attributes, spatiotemporally-dependent thematic attributes, spatial attributes,* and *temporally-dependent spatial attributes*. Adjectives are used to describe the attribute domain (e.g., *spatial* attribute) and adverbs with the word *dependent* to describe additional attribute properties (e.g., *temporally-dependent* spatial attribute). For each of the categories, a textual description, symbolic representation(s), and mathematical definition(s) are given below. Since a single category may have more than one possible symbolic representation and mathematical definition, textual descriptions of each variation are also given.

- *Thematic Attribute:* Thematic attribute values.

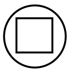
  <D> f: <oid> ➔ <D>

  Returns thematic attribute value for a given object instance.

- *Spatially-dependent Thematic Attribute:* Thematic attribute values associated with a set of spatial extents, each representing the location where a given attribute value is valid for an object or association. This implies that the attribute values may change over space and their changed values may be retained.

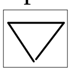
   f: <oid> ➔ (<S> ➔ <D>)

  Returns a set of spatial points with their associated thematic attribute values for a given object instance.

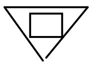
   f: <oid> ➔ (<D> ➔ <2^S>)

  Returns a set of spatial extents for each thematic attribute value for a given object instance.

- *Temporally-dependent Thematic Attribute:* A thematic attribute value is associated with one or more timestamps, representing the time when an attribute value is valid or part of the current database state. This implies that the attribute values may change over time and their changed values may be retained.

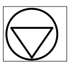
   f: <oid> ➔ (<T> ➔ <D>)

  Returns a set of timestamped thematic attribute values for a given object instance.

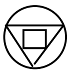
   f: <oid> ➔ (<D> ➔ <2^T>)

  Returns a set of timestamps for each thematic attribute value for a given object instance.

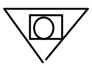
- *Spatiotemporally-dependent Thematic Attribute:* A combination of spatially- and temporally-dependent thematic attributes as defined above, i.e., each thematic attribute value is associated with a spatial extent and one or more timestamps.

   f: <oid> ➔ (<T> ➔ (<S> ➔ <D>))

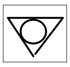
  Returns a set of time points each with its associated set of spatial points each with its associated thematic attribute value for a given object instance, i.e., returns the thematic attribute value associated with a given time and spatial point for a given object instance. The simpler explanation will be used henceforth.

   f: <oid> ➔ (<D> ➔ (<T> ➔ <2^S>))

  Returns a set of spatial points associated with a given thematic attribute value and timepoint for a given object instance.

   f: <oid> ➔ (<S> ➔ (<D> ➔ <2^T>))

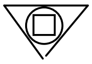
  Returns a set of timestamps associated with a given spatial point and thematic attribute value for a given object instance.

 $f: \langle oid \rangle \rightarrow (\langle S \rangle \rightarrow (\langle T \rangle \rightarrow \langle D \rangle))$

Returns the thematic attribute value associated with a given spatial and time point for a given object instance.

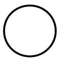
 $f: \langle oid \rangle \rightarrow (\langle T \rangle \rightarrow (\langle D \rangle \rightarrow \langle 2^S \rangle))$

Returns a set of spatial points associated with a given timepoint and thematic attribute value for a given object instance.

 $f: \langle oid \rangle \rightarrow (\langle D \rangle \rightarrow (\langle S \rangle \rightarrow \langle 2^T \rangle))$

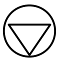
Returns a set of timestamps associated with a given thematic attribute value and spatial point for a given object instance.

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

 $f: \langle oid \rangle \rightarrow \langle 2^S \rangle$

Returns spatial extent (i.e., attribute value) for spatial attribute (i.e., attribute with spatial domain) for a given object instance.

- *Temporally-dependent Spatial Attribute:* A spatial attribute is associated with one or more timestamps, representing the time when the spatial attribute value is valid or part of the current database state.

 $f: \langle oid \rangle \rightarrow (\langle T \rangle \rightarrow \langle 2^S \rangle)$

Returns a set of timestamped spatial extent(s) for a spatial attribute for a given object instance.

 $f: \langle oid \rangle \rightarrow (\langle S \rangle \rightarrow \langle 2^T \rangle)$

Returns a set of timestamps when the spatial attribute's spatial extent intersects a given spatial point for a given object instance.

The use of these symbols at the attribute level is illustrated in Figure 6 using part of the Spatiotemporal Health application from section 2. The difference between (a) thematic attributes, (b) temporally-dependent thematic attributes, (c) spatiotemporally-dependent thematic attributes, (d) spatial attributes and (e) temporally-dependent spatial attributes is illustrated by (a) *name (for Hospital and State)*, (b) *numBeds*, (c) *populationDensity*, (d) *location*, and (e) *halfHourZone*, respectively.

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 other domain is explicitly specified for an attribute, then the use of the Extended Spatiotemporal UML 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 spatiotemporal symbol (and no thematic domain) represents spatial data whose geometry 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 spatiotemporal 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 is associated with a timestamp and spatial extent. If the temporal symbol were omitted from *populationDensity*, then this would be an example of a spatially-dependent, thematic attribute whose value may change over space and is thus associated with a spatial extent.

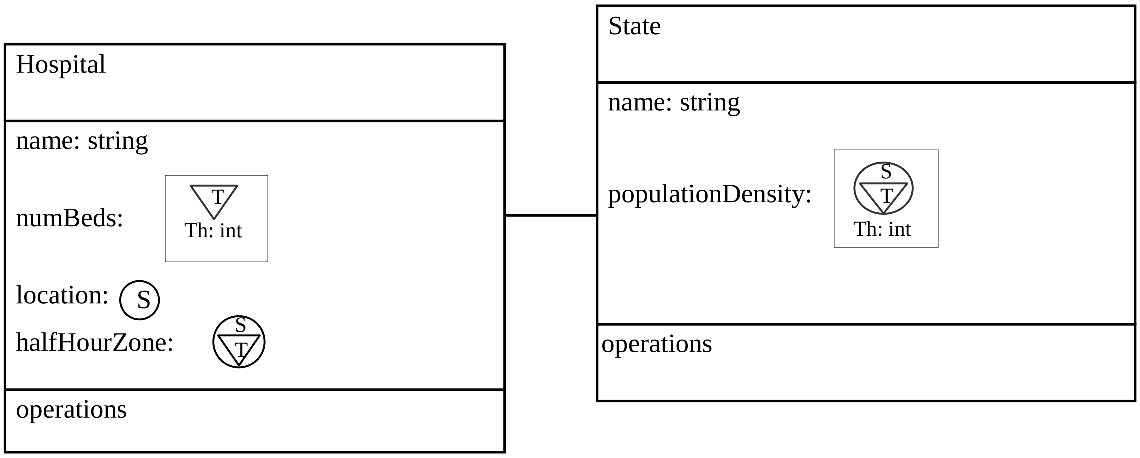


Figure 6: The Use of the Extended Spatiotemporal UML at the Attribute Level

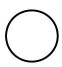
3.3 The Association Level

At the association level, any nested combination of a spatial and/or a temporal symbol represents a legal combination describing spatiotemporal properties of the association. Except for the omission of the *thematic* symbol, using the Extended Spatiotemporal UML *spatial* and *temporal* symbols at the association level is similar to their use at the attribute level. The association spatiotemporal properties can optionally be followed by an *existence-dependent* symbol (discussed in Section 4). The rule for the Extended Spatiotemporal UML notation at the association level can be given in BNF as follows.

assoc-line [<s> | <t> | <st>] [<ED>]

Three different categories of associations are possible: *spatially-*, *temporally-*, and *spatiotemporally-dependent associations* reading the BNF rule above from left to right. These are defined below.

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

   f: <aid> ➔ <2^S >

  Returns spatial extent of association instance.

- *Temporally-dependent Association:* An association instance is associated with one or more timestamps, representing the time when an association instance is valid or part of the current database state. This implies that association values may change over time and the changed values may be retained.

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

  Returns timestamp of association instance.

- *Spatiotemporally-dependent Association:* The spatial extent associated with a spatial association is also associated with one or more timestamps, representing the time when the spatial extent is valid in the real world or part of the current database state.

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

  Returns a set of timestamped spatial extent(s) of association instance.

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

  Returns a set of timestamps when association instance's spatial extent intersects a given spatial point.

The use of these symbols at the association level is illustrated in Figure 7 using part of the Spatiotemporal Health application. For instance, marking an association of *Hospital* and *Hospital-type* classes with a *temporal* symbol signifies that the type of a hospital (e.g. general, maternity) may change over time, as local health needs change and hospital wards are opened or closed. Therefore, the association instances should be timestamped.

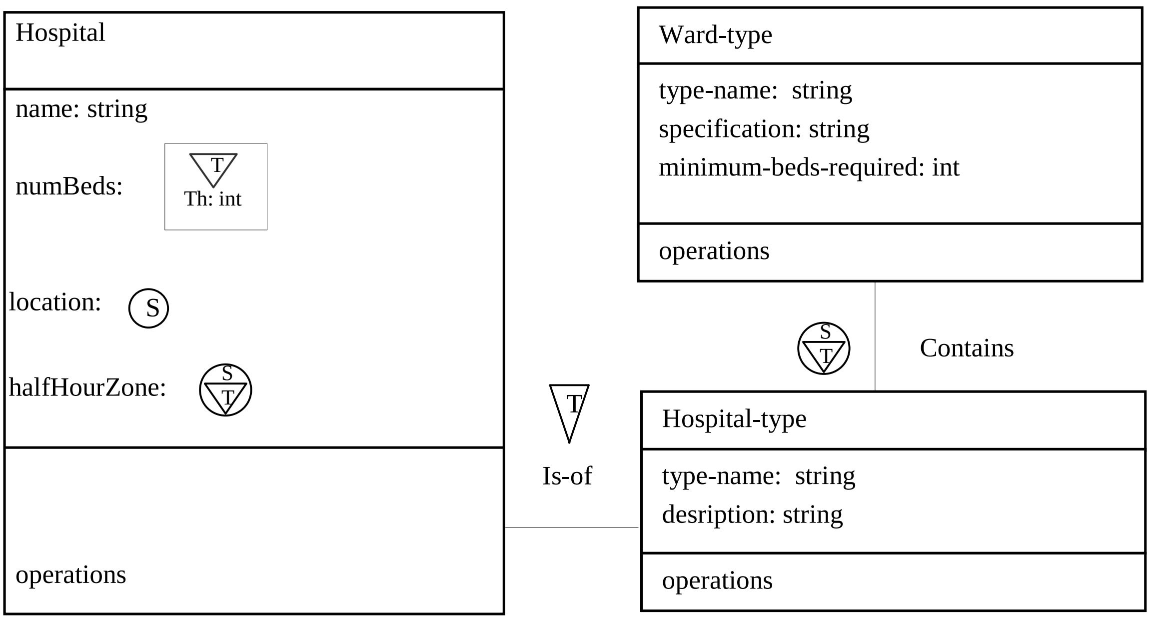


Figure 7: The Use of the Extended Spatiotemporal UML at the Association Level

A spatially-dependent association is one where an association instance may change over space and thus should be associated with a spatial extent to show where that instance is valid. Any physical object class (i.e., having a material form) can only be in one place at a time, therefore, changes over space correspond with changes over time and can be modeled simply as a temporal association. However, spatially-dependent (and spatiotemporally-dependent) associations are useful for abstract object classes (e.g., *Hospital-type*). For instance, the same type of hospital may require different types of wards in different areas depending on local regulations. The wards required for a given hospital type may also change over time as regulations change, thus the *Contains* association is spatiotemporally-dependent.

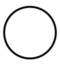
3.4 The Object Class Level

An object class can be marked by a temporal symbol, a spatial symbol, or any nested combination of these. In addtition, 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> | <st>] [<t>]

Corresponding to the five possible instantiations of this rule, <s>; <st>; <t>; <s><t>; and <st><t>,there are five different categories of object classes: *spatial object, temporally-dependent spatial object, temporal object, spatiotemporal object, and temporally-dependent spatiotemporal object respectively* 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.

   f: <oid> ➔ <2^S>

  Returns spatial extent of object instance.

- *Temporally-dependent Spatial Object (Class):* The spatial extent associated with a spatial object is also associated with one or more timestamps, representing the time when the spatial extent is valid in the real world or part of the current database state for that object.

   f: <oid> ➔ (<T> ➔ <2^S>)

  Returns a set of timestamped spatial extent(s) of object instance.

   f: <oid> ➔ (<S> ➔ <2^T>)

  Returns a set of timestamps when object instance's spatial extent intersects a given spatial point

- *Temporal Object (Class):* An object is associated with one or more timestamps, representing the existence time of the object or the time the object is part of the current database state.

   f: <oid> ➔ <2^T>

  Returns timestamp of object instance.

- *Spatiotemporal Object (Class):* 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 existence time of the object or the time the object is part of the current database state.

  $f: \langle oid \rangle \rightarrow \langle 2^T \rangle$ & $f: \langle id \rangle \rightarrow \langle 2^S \rangle$

  Returns timestamp of object instance and spatial extent of object instance.

- *Temporally-dependent Spatiotemporal Object (Class):* 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 valid or transaction time of the spatial extent, and one or more timestamps representing the existence or transaction time of the object itself (including all its attributes, etc.).

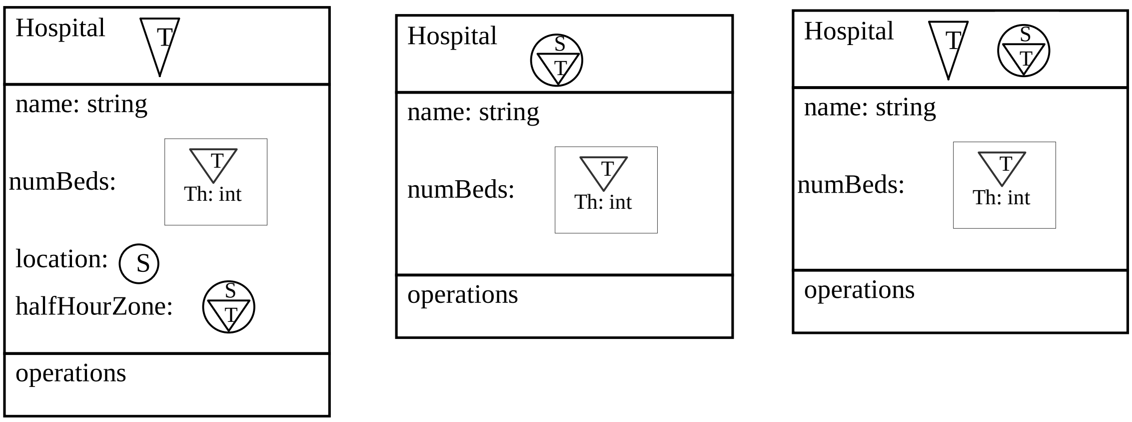
  $f: \langle oid \rangle \rightarrow \langle 2^T \rangle$ & $f: \langle id \rangle \rightarrow (\langle T \rangle \rightarrow \langle 2^S \rangle)$

  Returns timestamp of object instance and a set of timestamped spatial extent(s) of object instance.

  $f: \langle oid \rangle \rightarrow \langle 2^T \rangle$ & $f: \langle oid \rangle \rightarrow (\langle S \rangle \rightarrow \langle 2^T \rangle)$

  Returns timestamp of object instance and a set of timestamps when object instance's spatial extent intersects a given spatial point.

The use of symbols at the object class level is illustrated in Figure 8 using part of the Spatiotemporal Health application. In 8 (a), the temporal symbol at the *Hospital* object level represents a temporal object class with existence and transaction time (see Section 4). In Figure 8 (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 or even to have a separately named attribute for 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, 8(c) combines (a) and (b), illustrating a temporally-dependent spatiotemporal object. The object is *spatiotemporal* because it is timestamped and has a spatial extent; and it is *temporally-dependent* because the spatial extent is also timestamped.



(a) (b) (c)

Figure 8: Use of the Extended Spatiotemporal UML at the Object Class Level

4. 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 spatiotemporal semantics are specified in a *specification box*, which can be associated with any of the icons or combinations. The specification box includes information on the time and space dimensions, model (i.e., regular or irregular time models and object- or field-based spatial models), time unit (i.e., instant, interval, time element), and–for spatially- and/or temporally-dependent thematic attributes–interpolation (e.g., step, average, maximum, minimum) can be specified to derive values between recorded spatial locations or timestamps. The *specification box* syntax is illustrated in Figure 9.

```
SPECIFICATION BOX  <Identifier>:
TimeDimen. ::=  [existence | valid ] [ transaction ]
TimeInterpolation ::= discrete | step | min | max | avg | linear | spline | <user-defined>
TimeModel [ (<TimeDimen.>) ] ::= 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 )* )]
```

Figure 9: Specification Box Syntax

Time dimensions include existence (i.e., lifespan) time (for objects), valid time (for attributes and associations), and transaction time (for objects, attributes, or associations), as defined in [Jensen98]. However, object existence time is more precisely defined asu the time during which existence-dependent attributes and associations can be defined (i.e., have legal values) for that object. This implies that some attributes, e.g., an employee's social-security number, may be defined even after the employee no longer "exists." Other attributes, e.g., work-phone number, are defined only while the employee works at the company. Note that existence time is not necessarily equivalent to biological lifespan. The exact meaning will be dependent on the application; therefore, individual applications define which attributes and associations are existence-dependent. Object identifiers are never existence-dependent, as they can be used to refer to historical objects.

If existence time is associated with a given object, the existence-dependent attributes and associations for that object class must be explicitly marked as such by adding the superscript *ED* to the attribute or association name. Conversely, existence-dependent attributes and associations can only be defined for objects having existence time specified. In the case of an existence-dependent association, existence time must be defined for at least one of the participating objects.

If an existence-dependent attribute is temporally-dependent, then every valid-time timestamp for the attribute's instance data must be included within the existence time of the corresponding object instance. If an existence-dependent association is temporally-dependent, then every valid-time timestamp for the association's instance data must be included within the existence time—if specified—of the participating object instances. If the existence-dependent attribute is not temporally-dependent, then the static instance values are assumed to be valid during the entire existence time of the corresponding object instance. If the existence-dependent association is not temporally-dependent, then the static instance values are assumed to be valid for the intersection of the existence times for those participating object instances which have existence time defined. The existence-dependent attribute or association is undefined at all other times.

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, i.e., the attribute instance for a single object instance consists of a single spatial extent. The field-based spatial model is used for a spatially-dependent, thematic attribute, where a single object instance has a set of thematic values with each value being associated with a different spatial extent.

The specification box can also be used to specify spatiotemporal constraints, including constraints within an attribute group. The group symbol is used to group attributes sharing the same timestamps or spatial extents, which then only need to be specified once for the group. Thus the group symbol graphically illustrates associated sets of attributes. 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 the *specification boxes* for that class, i.e., the *specification* compartment. The *specification* compartment can be used to specify spatiotemporal semantics for the attributes of the object class and any associations in which the object class participates. Alternatively, a *specification* compartment can be added to an association class to specify spatiotemporal semantics for that association and its attributes. A detailed discussion of the specification box can be found in [Price99b].

Figure 10 shows the full Spatiotemporal Health application described in section 2 as it would be represented using the Extended Spatiotemporal UML and illustrates the use of the specification box, the group symbol, and the existence-dependent symbol. For example, *Hospital location* is specified as a single point in 2D space. *Hospital halfHourZone* and *Contains* are specified as a region in 2D space. In contrast, the *State 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. Since these two attributes share common timestamps and spatial extents (and have the same thematic domain), they are grouped. The group is then associated with a single symbol and *specification box*. Here we specify that any attribute in the group uses step interpolation in space and average

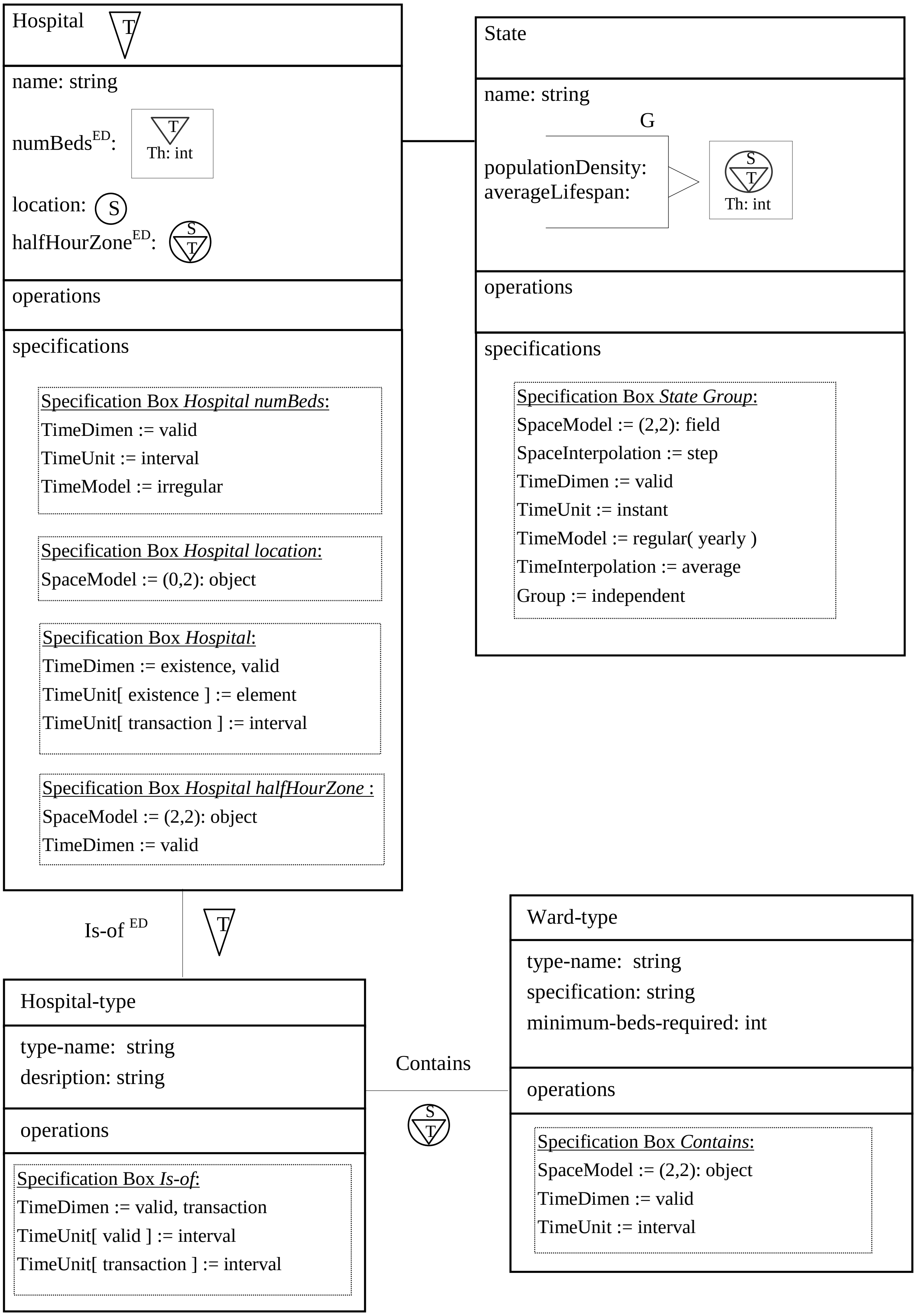


Figure 10: The Health Example Using the Extended Spatiotemporal UML

interpolation in time, 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; whereas, the values of the same attributes between recorded spatial locations is assumed to be the same as for the nearest recorded spatial location. 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 time and transaction time. Existence time is used to model the periods when the hospital is open, i.e., when the existence-dependent attributes *numBeds* and *halfHourZone* and the existence-dependent association *Is-of* are defined. Thus the valid timestamps of all instances for these attributes and association must be included within the *Hospital* existence time. Because the hospital may have intermittent closures, *time element* is selected as the time unit for the *Hospital* existence time. Transaction time captures when hospital objects are part of the database's current state and is represented using *interval*. 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 the *Contains* association could be placed in either the *Hospital-type* or the *Ward-type* specification compartment. Similarly, the *Is-of* association could be placed in either the *Hospital* or the *Hospital-type* specification compartment. Note that since the *Hospital-type* object class is not temporal and therefore does not have existence time defined, the only constraint on *Is-of* valid-time timestamps comes from the *Hospital class* existence timeThe time unit for *halfHourZone* is not yet specified. This illustrates that specifications can be added incrementally during the design process.

5. Conclusion

A comparison of the use of UML in Figure 3 and the Extended Spatiotemporal UML used in Figure 10 to represent the same Spatiotemporal Health application illustrates the advantages of the Extended Spatiotemporal UML in modeling spatiotemporal data. The use of UML in Figure 3 results in the creation of a host of artificial constructs that obscure the schema design and fails to provide a single, easily visible notation for spatiotemporal semantics. In Figure 10, Extended Spatiotemporal UML is used to hide the complexity of modeling spatiotemporal data. The level of detail in the schema is reduced without sacrificing understandability. This allows the application developer to concentrate on the characteristics of the specific application domain of interest. Furthermore, the Extended Spatiotemporal UML provides a clear and consistent framework for the detailed specification of spatiotemporal semantics using the specification box. These semantics, not shown in Figure 2, would have to be represented in UML using constraints and other annotations that are unlikely to be standardized among users. The specification box can thus provide a guideline for application developers, which highlights generally relevant semantics that should be considered when modeling spatiotemporal data, and it facilitates communication and design documentation. The Extended Spatiotemporal UML thus provides a simple and natural notation for modeling applications requiring support for a range of spatiotemporal models and data types.

The contributions represented by the Extended Spatiotemporal UML presented in this paper include:

- providing a more exact definition of existence time in terms of application-defined dependencies than is currently provided in the literature and introducing modeling constructs to reflect the new semantics,
- introducing a thematic symbol to provide a consistent notation for representing thematic attributes with spatial and/or temporal properties,
- defining a flexible and consistent method for combining spatial, temporal, and thematic symbols at different levels of the UML model with precisely defined semantics,
- providing a formal definition of spatiotemporal UML notation which can be used for data retrieval and for potential use in developing a graphical spatiotemporal query language.

We are currently investigating the specification of spatiotemporal constraints using UML's Object Constraint Language (OCL) and the use of the Extended Spatiotemporal UML for multimedia applications.

Acknowledgements

Thanks are due to Bala Srinivasan and Kotagiri Ramamohanarao for their continuing support. This research was conducted in part while the first author visited Aalborg University. The work was supported in part by the Danish Technical Research Council through grant 9700780, by the Chorochronos project, funded by the European Commission, contract no. FMRX-CT96-0056, and by a grant from the Nykredit Corporation.

References

[Abra99] Abraham, T. and J. F. Roddick. *Survey of Spatio-Temporal Databases*, GeoInformatica, 3(1), 1999, 61-99.

[Beck96] Becker, L., A. Voigtmann, and K. Hinrichs. *Temporal Support for Geo-Data in Object-Oriented Databases*, Proc. of DEXA, 1996, 79-93.

[Booc99] Booch, G., J. Rumbaugh, and I. Jacobson. The Unified Modeling Language User Guide, Addison-Wesley, Reading, MA, 1999.

[Clar95] Claramunt, C. *Managing Time in GIS An Event-Oriented Approach*, Proc. of Intl. Workshop on Temporal Databases, 1995, 23-42.

[Dion98] Dionisio, J.D.N. and A.F. Cardenas. *A Unified Data Model for Representing Multimedia, Timeline, and Simulation Data*, IEEE TKDE, 10(5), Sept/Oct 1998, 746-767.

[Fari98] Faria, G., C. B. Medeiros, and M.A. Nascimento. *An Extensible Framework for Spatio-Temporal Database Applications*, Time Center Technical Report TR-27, 1998, 1-15.

[Fowl97] Fowler, M. and K. Scott. UML Distilled Applying the Standard Object Modeling Language, Addison-Wesley, Reading, MA, 1997.

[Jensen98] Jensen, C. and C. E. Dyreson, eds. *The Consensus Glossary of Temporal Database Concepts*—Feb 1998 Version. In O. Etzion, S. Jajodia, and S. Sripada, eds., Temporal Databases: Research and Practice, LNCS 1399, Springer-Verlag, 1998, 367-405.

[Lang93] Langran, G. Time in Geographic Information Systems, Taylor & Francis, London, 1993.

[Par99] Parent, C., S. Spaccapietra, and E. Zimanyi. *Spatio-Temporal Conceptual Models: Data Structures + Space + Time*, Proc. 7th ACM Symposium on Advances in GIS, ACM GIS'99, Nov 5-6, 1999.

[Pav98] Pavlopoulos, A. and B. Theodoulidis. *Review of SpatioTemporal Data Models*. TimeLab Technical Report TR-98-3, 1998.

[Pequ95] Pequet, D. J. and Duan. N. *An Event-Based Spatiotemporal Data Model (ESTDM) for Temporal Analysis of Geographical Data*, Intl. Journal of GIS, Vol. 9, No. 1, 1995, 7-24.

[Pfo98] Pfoser, D. and N. Tryfona. *Requirements, Definitions, and Notations for Spatiotemporal Application Environments*, 6th ACM International Workshop on Advances in Geographic Information Systems, ACM Press Baltimore, MD, Nov 1998, 124-130.

[Price99a] Price, R., B. Srinivasan, and K. Ramamohanarao. *Recent Trends in Spatiotemporal Data Modelling*, Monash University Technical Report# 1999/43, 1-12.

[Price99b] Price, R., B. Srinivasan, and K. Ramamohanarao. *Extending the Unified Modeling Language to Support Spatiotemporal Applications*, Asia TOOLS'99, 1999, 163-174.

[Tryf99a] Tryfona, N. and C. Jensen . *Conceptual Data Modeling for Spatiotemporal Applications*, Geoinformatica 1999 (in press).

[Tryf99b] Tryfona, N. and C. Jensen. *Using Abstractions for Spatio-Temporal Conceptual Modeling*, Proc. of 14th Annual Symposium on Applied Computing (SAC'00), Como, Italy, March 2000 (in press)

[Worb95] Worboys, Michael. *Next-Generation Systems*, GIS A Computing Perspective, Ch. 8, Taylor & Francis, London, 1995, 331-332.