A CONCEPT FOR INTEGRATING TIME-DEPENDENT FEATURES IN 3D BUILDING MODELS

Hongchao Fan *, Liqiu Meng, Uwe Stilla

Institute of Photogrammetry and Cartography,
International Graduate School of Science and Engineering (IGSSE),
Technische Universität München
Arcisstr. 21, D-80333 Munich, Germany
{fan, meng, stilla}@bv.tum.de

KEY WORDS: 3D City Model, Spatiotemporal Modelling, CityGML, Query, Visualization

ABSTRACT:

The 3D buildings change over the time not only geometrically, but also semantically. However, the current existing 3D city models are not able to represent the change information, because they are reconstructed on the basis of information frozen at one certain time point. Although lots of available spatio-temporal data models have been reported in the literature, they can hardly manage complex 3D geometries along with their semantic changes at different levels of details. More critically, they do not support continuous spatiotemporal changes. This paper presents an approach for integrating and representing time-dependent features in 3D building models. The proposed spatiotemporal data model is hierarchical entity-based. It is intended to circumvent the above mentioned problems by adding timestamps to entities, categorizing the spatiotemporal changes, and mathematically describing the processes of changes. This data model could be realized either by extending the widespread OpenGIS standard CityGML or by establishing an own relational database. The data model in its current version supports four basic types of spatiotemporal queries. The results of queries can be visualized in two different modes: statically and dynamically.

1. INTRODUCTION

During the recent decade, a growing number of municipalities have decided to build up 3D city models. Many of the existing models serve the main purpose of supporting urban planning processes by means of visualization of virtual scenes. Being reconstructed on the basis of information frozen at one certain time point, however, these city models do not contain any clues about the temporal alterations of individual buildings. Therefore, their applications remain rather limited. In our approach, we attempt to integrate and visualize the time-dependent information with a 3D city model so as to create a so-called 4D virtual city environment.

Our approach is concerned with the following research questions of (a) how to contemplate a data model that is able to handle temporal information associated with 3D city objects at fine granularity levels, (b) how to define a suitable data structure that supports not only spatial queries, but also queries of events, processes as well as temporal topological relationships, (c) how to develop mechanisms of compressing spatio-temporal information for fast rendering, (d) how to design user interactions that go beyond virtual walks or flights through the static 3D models and allow an immersive access to and modification of dynamic behaviour of complex objects (e.g. complex 3D buildings), simple objects (e.g. single roofed garages) and object components (e.g. roofs, walls and grounds), and (e) how to augment video streams with computer-generated imagery of spatio-temporal information.

The first step of our approach is to conceptualize a data model. While the geometries of 3D buildings change over the time, their semantics e.g. the owner of a building could also be changed within a certain time span. Therefore, the anticipated data model should be able to represent and operate the spatiotemporal features on the one hand, and manage the semantics as well as their changes over the time on the other hand. In the past two decades many spatiotemporal data models have been developed by adding timestamps to the spatial and semantic attributes of spatial objects, e.g. the spatiotemporal snapshot model (Langran 1992), entity-based model (Tryfona & Jensen 1999, Erwig et al. 1998) and event-based spatiotemporal model (Peuquet & Duan 1995). However, most of the existing models are restricted to modelling two dimensional spatial data. Or they cover only partly the requirements by addressing either spatial or temporal modelling. Many of them handle at the logical level (Parent et al. 1999). Moreover, the existing spatiotemporal data models do not support the management of both discrete and continuous spatiotemporal changes (Jin et al. 2007). In this paper we propose a spatiotemporal data model specified for the 3D building model. The idea is rooted on the extension and integration of entity-based model, snapshot model and event-based model. In this model the key objects are 3D buildings, each with an ID number that holds all of the events happened to the corresponding building at different time points. As an event is defined as the change of an entity (3D building or part of it) from a state to another if the life of 3D building is viewed as a series of states or "snapshots" (Fan & Meng, 2008), every event would be described by a function or a model which describes how the entity changes from one state to another and could be derived from statistics of building construction in case of an event "*a new building is constructed*". At the same time every state (snapshot) and event would be tagged with a timestamp for representing their valid life. Thus the states and events of an entity might be well connected and the whole life of this entity could be represented continuously

* Corresponding author. fan@bv.tum.de; phone +49 89 289-22586; fax +49 89 280-9573; www.carto-tum.de.

and seamlessly in the data model. An important task involved in this first step is to define what an event of the 3D building is and categorize the events. This issue has been studied and presented by Fan and Meng (Fan & Meng, 2008).

Currently existing 3D city models are predominantly reconstructed either with OpenGIS standards CityGML or with the newest release of the dominating GoogleEarth KML. In order to compare these two languages, thus gain an insight into their strengths and weaknesses, we created for one part of the building block of the Technical University of Munich, Germany, a 3D model with both CityGML and GoogleEarth KML. Our experiences so far have led to the conclusion that CityGML is better suitable for our approach of 4D Virtual City Environment. However, we have to extend the currently available CityGML as its newest vision is not yet able to model the temporal information describing the individual objects at various LODs (Levels of Detail).

At the same time, meaningful user interactions and queries of temporal information will be anticipated for the selected application scenarios of our 4D Virtual City Environment. The visualization of the query results could be static or dynamic. In the static mode, the changes of an object might be presented with different colors or different forms of lines and polygons. In the dynamic mode, individual 3D objects could be constructed or destructed at different time points within a given time span. Thus, the chronological sequence of events can be visualized.

Generally, in order to reach our goals, a suitable database should be constructed for different queries associated with different applications. Meanwhile, the algorithms for fast rendering are necessary which should guarantee the instant visualization of query results.

The remainder of this paper is structured as follows. In the second section we will explain our proposal for a hierarchical entity-based data model for spatio-temporal data. In the third section we will present our current work and experiences of creating 3D buildings with both CityGML and Google Earth KML. In the fourth section queries on the spatio-temporal data model will be discussed and two modes for visualization of the query results will be introduced. Finally, conclusions are presented in section 5.

2. DATA MODEL

Our aim is to integrate time-dependent features in 3D building models. Since not only the geometries but also the semantics of the 3D buildings change over the time, the data model to be designed should present and operate both the spatiotemporal and the semantic-temporal information. In our approach we add timestamps to every state of an individual entity. Besides, different changes of states in case of 3D building are described and represented by different types and mathematical models which can be derived from statistics of building construction. If e.g. "*a new building is constructed or created*" as a change is regarded, then every state during the change could be obtained through interpolation according to the mathematical model. Consequently, the whole life of an entity could be represented continuously in the data model.

As mentioned above, the data model is an entity-based data model. The entity here refers all of the single and complete units of 3D buildings, in both geometrical and semantic sense. For instance, it might be a building, a part of building, a roof, a wall, a door or a window in the case of geometrical entity. If a semantic entity is regarded, it could be the name, owner or function of a building or a part of building; it could also be color of the façade or the material of a door or a window. Every sort of entity will be stored in a table in which a record represents a state of an entity. And the change of the entity from one state to another will be indicated by a timestamp, the type and pattern of the change. Entities that are related with each other could be connected by pointers. All entities which belong to the same building and exist at the same time point compose or represent the building at that time point.

In the following, examples of some entities will be given. Table 1 shows the data structure of the entity of "Building". A building is defined by the thematic extension module *Building* which is the most detailed thematic concept of CityGML, and it allows for the representation of thematic and spatial aspects of buildings, building parts, building installations, and interior building structures in four levels of detail, LOD1 to LOD4 (OGC, 2008). This table is located on the top level in the whole data model. In this table all of the entities which could change during the life of the building will be stored in other tables respectively at the second level (see table 2 and 3).

Building				
Fieldname		**Data type**	**Length**	**IsNull**
Building ID		String	50	No
Timestamp		Time	50	No
BuildingPart		Pointer	50	No
Thematic Attributes	Owner	Pointer	50	Yes
	Address	Pointer	50	Yes
	Name	Pointer	50	Yes
	Function	Pointer	50	Yes
	Class	Pointer	50	Yes
	Market price	Pointer	50	Yes

Table 1. The data structure at the level of Building

Table 1 denotes that every building is a record in the data model. And it is represented by its geometric attributes (in BuildingPart) and thematic attributes. The timestamp records its whole life which counts from the time point when the first part of this building is created to the time point when the last part of it is destroyed. The entities of the building which could change during this lifespan would be stored in other tables and connected by pointers, e.g. the change of BuildingPart and owner will be stored in table 2 and table 3 respectively.

<table>
<tr><td colspan="2">BuildingPart</td><td colspan="3"></td></tr>
<tr><th colspan="2">Fieldname</th><th>Data type</th><th>Length</th><th>IsNull</th></tr>
<tr><td colspan="2">BuildingPart ID</td><td>string</td><td>50</td><td>No</td></tr>
<tr><td colspan="2">BuildingID</td><td>Pointer</td><td>50</td><td>No</td></tr>
<tr><td colspan="2">Wall</td><td>Pointer</td><td>50</td><td>No</td></tr>
<tr><td colspan="2">Roof</td><td>Pointer</td><td>50</td><td>No</td></tr>
<tr><td colspan="2">Previous</td><td>Pointer</td><td>50</td><td>Yes</td></tr>
<tr><td colspan="2">Next</td><td>Pointer</td><td>50</td><td>Yes</td></tr>
<tr><td colspan="2">Start of constructing</td><td>Time</td><td>50</td><td>No</td></tr>
<tr><td colspan="2">End of constructing</td><td>Time</td><td>50</td><td>No</td></tr>
<tr><td colspan="2">Type of constructing</td><td>String</td><td>50</td><td>No</td></tr>
<tr><td colspan="2">Start of destroying</td><td>Time</td><td>50</td><td>Yes</td></tr>
<tr><td colspan="2">End of destroying</td><td>Time</td><td>50</td><td>Yes</td></tr>
<tr><td colspan="2">Type of destroying</td><td>String</td><td>50</td><td>Yes</td></tr>
<tr><td rowspan="4">Thematic Attributes</td><td>Owner</td><td>Pointer</td><td>50</td><td>Yes</td></tr>
<tr><td>Rent</td><td>Pointer</td><td>50</td><td>Yes</td></tr>
<tr><td>Tenant or Resident</td><td>Pointer</td><td>50</td><td>Yes</td></tr>
<tr><td>Function</td><td>Pointer</td><td>50</td><td>Yes</td></tr>
</table>

Table 2. The data structure of BuildingPart

Owner			
Fieldname	**Data type**	**Length**	**IsNull**
OwnerID	String	50	No
BuildingID	pointer	50	No
First name	String	50	Yes
Surname or name of institute	String	50	No
Birth data	Time	50	Yes
Sex	String	20	Yes
Former Onwer	String	100	Yes
Date to Own	Time	50	Yes
Type to Own	String	50	Yes
Date to Transfer	String	50	Yes
Type to Transfer	String	50	Yes
Next Owner	String	50	Yes

Table 3. The data structure of owner

Table 2 and 3 are located at the second level of the data model. There can be many further tables to represent the changes of entities at the second level, e.g. table of address, name, function etc. Analog to the relationship between the table at the first level and those at the second level, the entities of a record e.g. "BuildingPart" will be pointed to the tables at the third level, if they could change during the life of this "BuildingPart". In this way a building including its geometrical and semantic attributes would be split at different geometrical scale and temporal scale, as be shown in the diagram below (figure 1).

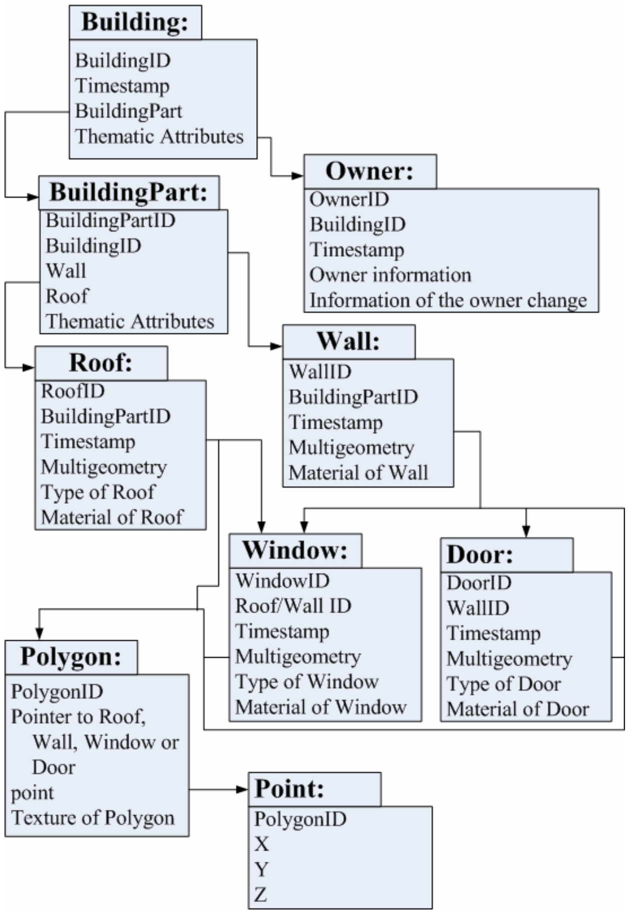

Figure.1. The hierarchical entity-based data model

Figure 1 illustrates the hierarchical structure of the entity-based data model. This ideal originates from the inherent characteristics of the relationships among the entities at different levels, not only geometrically but also semantically. Such an approach could benefit for the subsequent queries and visualization processes in a 4D virtual city environment.

Another highlight of our approach is that the data model could represent the life of an entity continuously by means of a mathematical model which can describe the change of the entity from one state to another. The mathematical model could be derived from the statistics of the construction process of the buildings or their parts. At the same time it is also meaningful to investigate the mathematical models themselves, because they could be treated as one of semantic attributes of building and their changes could reflect the development of the technology and the increase of the social wealth.

3. CITYGML AND GOOGLE EARTH KML

Currently existing 3D city models are predominantly reconstructed either with OpenGIS standards CityGML or with the newest release of the dominating GoogleEarth KML. In this section these two languages will be introduced briefly first of all. Then they would be compared so that one of them that is more appropriate for our project will be chosen.

3.1 Brief introduction

CityGML is the abbreviation of the City Geography Markup Language. It has been developed by the members of the Special Interest Group 3D (SIG3D) of the initiative Geodata Infrastructure North Rhine-Westphalia (GDI NRW) in Germany since 2002 and realised as an open data model and XML-based format to store and exchange virtual 3D city model. In fact, it could be implemented as an application schema for the Geography Markup Language 3 (GML3), the extendible international standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and ISO TC211 (CityGML, 2007).

In CityGML the classes and relations for the most relevant topographic objects in cities and regions are defined with respect to their geometrical, topological, semantic and appearance properties. Included are generalisation hierarchies between thematic classes, aggregations, relations between objects, and spatial properties. In other words, both simple, single scale models without topology and few semantics and very complex multi-scale models with full topology and fine-grained semantical differentiations can be represented in CityGML.

Another most used language for modelling 3D object is Keyhole Markup Language (KML). Similar to CityGML KML is also an XML-based language schema. It would be used to display geographic data in an Earth browser such as Google Earth, Google Maps, and Google Maps for mobile. The newest version of KML at the moment is the KML 2.2. Its specification has been submitted to the OGC to assure its status as an open standard for all geobrowsers (KML Reference, 2007).

3.2 Comparison and conclusion

In order to compare these two languages, thus gain an insight into their strengths and weaknesses; we created for one part of the building block of the Technical University of Munich, Germany, a 3D model with both CityGML and GoogleEarth KML. Figure 2 and 3 show the building in 3D environment presented by CityGML and Google Earth KML respectively.

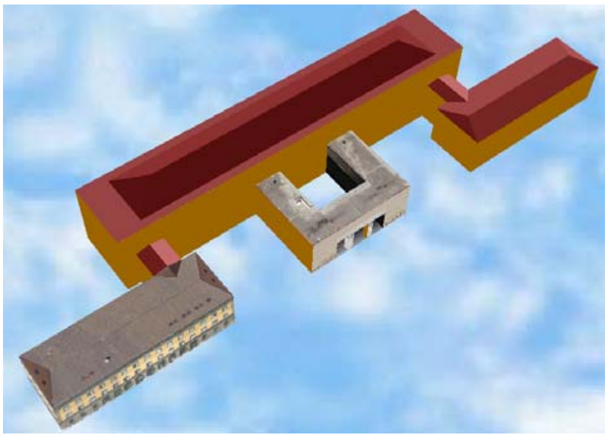

Figure 2. TUM building in CityGML

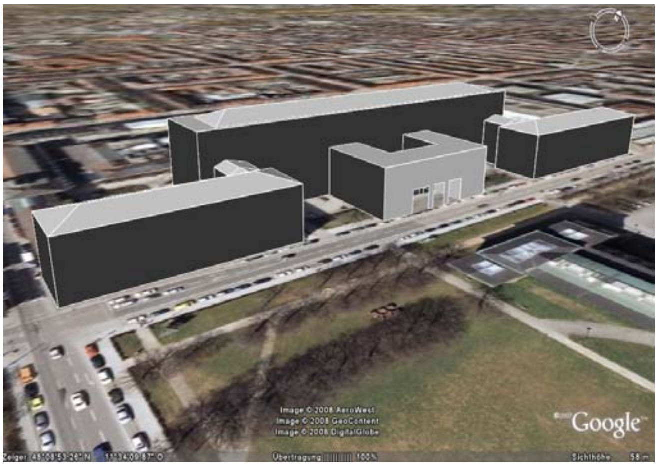

Figure 3. TUM building in Google Earth KML

Obviously, the two 3D models are different. The differences could be outlined as follows:

- Walls and roofs in CityGML are defined as different types of entity. In fact, CityGML defines lots of types of city objects, e.g. wall, roof, window, door, even the furniture in rooms of buildings, like tables and chairs, are defined as different types. And every visible object could be represented not only geometrically but also semantically. On the contrary objects are not distinguished in the KML. All of them are treated as "Placemark". And they could only be represented semantically.
- In CityGML every polygon can be connected with an image for its texture, while Google Earth KML does not have such function.
- In CityGML many different types of geometric primitives such as polygon, circle, cylinder, cubic spline etc. are defined. But KML objects are limited to point, line and polygon or their combinations.
- With regard to time component, in KML the life of an object could be represented by start, end point of its life or sometimes by their lifespan. Unfortunately, the time in CityGML is only defined as attributive information of an entity rather than an independent entity.
- In CityGML both local and global coordinate systems are allowed. But in KML only one global coordinate system WGS84 is supported.

Based on this comparisin, we choose the CityGML as our standard data format because our emphasis is laid on the representation of 3D buildings at various Levels of Detail (LoD) both geometrically and semantically.

However, we have to extend the currently available CityGML as its newest vision is not yet able to model the temporal information describing the individual objects at various LoDs.

Two different approaches for integrating temporal information in the 3D building model are considered. One of them is to extend the CityGML language in line with the proposed hierarchical entity-based data model in section 2. In this case the temporal elements including time, types and models of changes about 3D buildings would be defined in XML schema. Another approach is to establish additionally a relational database. All analyses and queries would be conducted on the database. And their results would be presented in CityGML.

4. QUERIES AND VISUALIZATION

Since our data model manages spatial and temporal information also events, processes as well as topological relationships, it allows the following four basic sorts of queries:

- find and display the status of an entity or group of entities within a certain space scope and a given time interval.
- find and display the status of an entity or group of entities within a certain space scope at a given time point.
- find and display a certain type of events within a certain space scope and a given time interval.
- find and display a certain type of events within a certain space scope at a given time point .

As mentioned above, events would be queried by using their types. In their work about analysis of events in 3D building models, Fan & Meng (2008) categorized events in terms of geometric and semantic changes (Fan & Meng, 2008). The resulted typology serves as the fundament for the queries in our approach.

The input of an explicit query triggers the corresponding operation on the spatio-temporal data model. The operations on spatio-temporal relations are similar to relational algebra. However, the time point or the time interval given in the query must be compared with the timestamps of the referred entity or entities. If the time point given in the query is viewed as a special time interval whose start and end is equal to each other, the comparison is then between two time durations and it outputs temporal relationships between them. In his paper on maintaining knowledge about temporal intervals, James Allen introduced 13 basic temporal relationships (Allen 1983). Fan & Meng modulated these 13 temporal relationships by changing the conditions of the observation (Fan & Meng, 2008). The temporal relationship determined by the comparison will be used to decide whether an entity, its whole life or part of its whole life satisfy the conditions defined in the query.

The visualization of the query results could be static or dynamic. In the static mode, the changes of an object might be presented with different colors or different forms of lines and polygons. Similar performance could also be found in our real world (see figure 4).

Figure 4 shows the "*Alte Pinakothek*" in Munich. The building was erected as gallery for painting collection in 1826. But it was considerably damaged in the World War II. Then the building was rebuilt by Hans Döllgast in 1957. However, rather than merely reconstructing the missing parts of the walls and windows, these areas were replaced with bare brickwork, in order to remain as visible "wounds" (website of Alte Pinakothek). If we observe this building from the time point of view, the building could be regarded as a visualization of different parts of buildings by using different colors and patterns to represent different time of construct. Figure 5 shows an example of static visualization for 3D buildings which were built in different years.

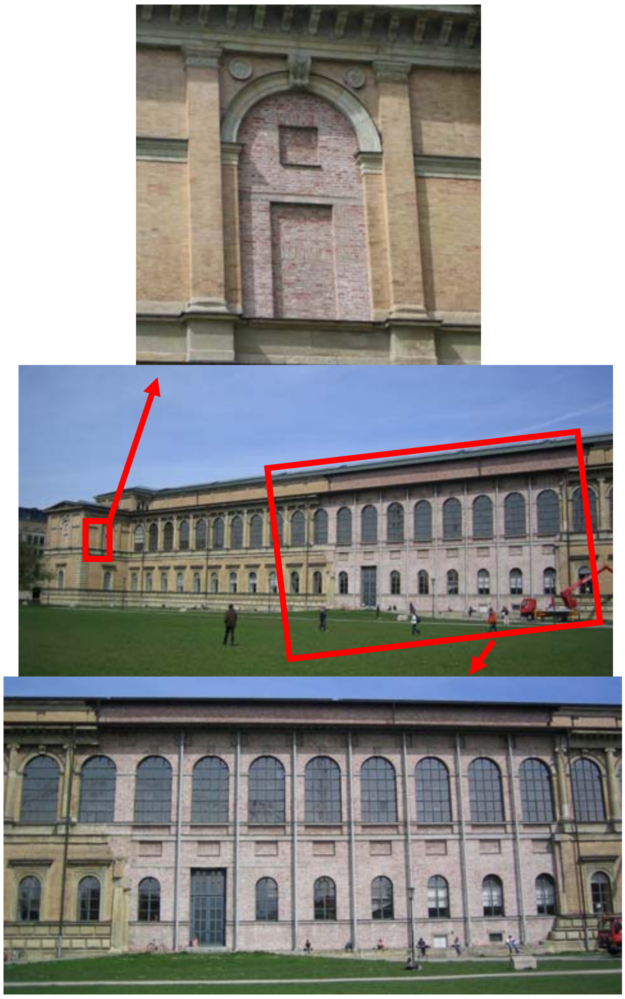

Figure 4. "*Alte Pinakothek*" in Munich

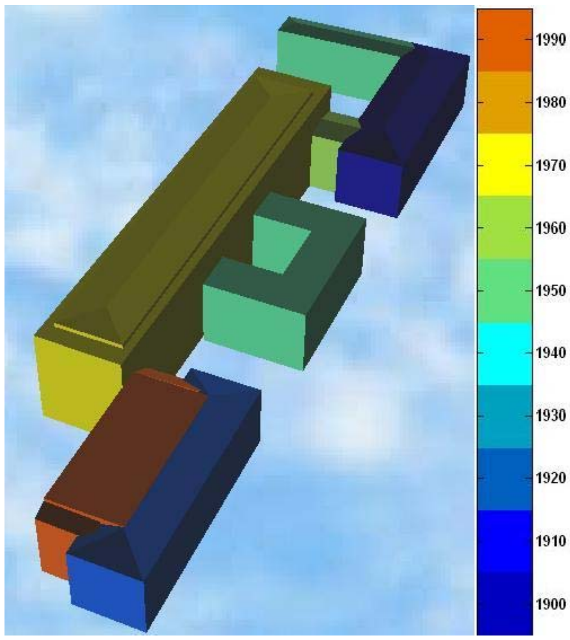

Figure 5. An example of static visualization for 3D buildings

At the same time, the results could be also visualized dynamically, if an entity or group of entities is/are queried with a time interval. In this case individual 3D objects could be constructed or destructed at different time points within a given time span. Thus, the chronological sequence of events can be visualized.

Additionally to the visual presentation, the results of the query could be saved in table, text file or even CityGML file for later use.

5. CONCLUSION AND OUTLOOK

This paper proposes a concept for integrating time-dependent features in 3D building models. The spatio-temporal data model is hierarchical entity-based. In this model every state and change of the state of an entity are tagged with timestamps. Besides, different changes of 3D buildings could be described and represented by different types and mathematical models which could be derived from statistics of building construction. Then every state during the dynamic change process could be interpolated depending on the nature of the change. In the way,, the whole life of an entity could be represented continuously in the data model.
On the other hand the hierarchical entity-based data model considers the inherent characteristics of the relationships among the entities at different levels, not only geometrically but also semantically.

The spatio-temporal data model allows four basic sorts of queries, which address not only spatial, but also events, processes as well as temporal topological relationships. The results of a query could be visualized statically by using different colors or different forms of lines and polygons for objects with different temporal attributes. Alternatively, the dynamic visualization mode could be activated, in which individual 3D objects are constructed or destructed at different time points within a given time span. Thus, the chronological sequence of events can be rendered.

In the future the statistical data about building construction in city Munich will be analysed in order to extract mathematical models for different types of construction. Secondly, the CityGML language will be extended according to the proposed data model. Correspondingly, a viewer for visualization and interacting with spatio-temporal data will be developed.

ACKNOWLEDGEMENT

The research work presented in this paper is financed by the International Graduate School of Science and Engineering (IGSSE) of the Technische Universität München, Germany.

REFERNCES

Allen, J.C. 1983. Maintaining knowledge about temporal intervals. CACM 26(11) 832-843.

Architecture of Alte Pinakothek.
http://www.pinakothek.de/alte-pinakothek/

Fan, H. & Meng, L. 2008. Analysis of events in 3D building models. Proceedings of SPIE, 16th Geoinformatics 2008 – Cartography, Guangzhou, June 2008

Gröger, G., Kolbe, T.H., Czerwinski & A., Nagel, C., 2008. OpenGIS® City Geography Markup Language (CityGML) Implementation Specification, Open Geospatial Consortium, http://www.opengeospatial.org/legal/.

Jin, P., Yue, L. & Gong, Y. 2007. Design and implementation of a unified spatio-temporal data model, Taylor & Francis Ltd.

KML 2.1 Reference. 2007.
http://code.google.com/apis/kml/documentation/kml_tags_21.html

Kolbe, T.H. 2007. What is CityGML.
http://www.citygml.org/1523/

Langran, G. 1992. Time in Geographic Information Systems. Taylor & Francis Ltd.

Peuquet, D.J. & Duan, N. 1995. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International Journal of Geographical Information Science*, 9:7, 7-24.

Worboys, M. 1994. A unified model for spatial and temporal information. *The Computer Journal* 37(1): 26-34.