

Protokoll der Sitzung der AG Modellierung der SIG 3D



Datum: 18.8.06, 14:30 Uhr – 16:35 Uhr

Ort: LVerma NRW, Bonn

Teilnehmer: Joachim Benner, Dave Capstick, Angela Czerwinski, Rüdiger Drees, Heinz Elfers, Hanno Euteneuer, Heinrich Geerling, Gerhard Gröger, Ulrich Gruber, Haik Lorenz, Jörn Loviscach, Thomas H. Kolbe, Mark Pendlington, Martin Rechner, Andreas Spors, Jörg Schmittwilken (Protokoll), Michael Ziegert

- **Nächster Termin der AG Modellierung: 29.9.06, 10 Uhr, IKG, Uni Bonn**
- **Einzelgebäude – Gruppierung**
 - LVerma: Modellierungsproblem beim Beispieldatensatz für Lärmberechnung
 - Anwendungsfall: Mehrere freistehende Gebäude bilden eine Einheit mit gleicher Adresse – z.B. LVerma. Werden diese in CityGML als ein Hauptgebäude (Building) mit BuildingParts modelliert, oder als viele einzelne Buildings mit der gleichen Adresse?
 - Konsens: Ein Building mit BuildingParts liegt nur dann vor, wenn diese beiden baulich miteinander verbunden sind. Sind sie nicht baulich miteinander verbunden, so sind mehrere Buildings zu bilden.
- **Texturierung und Texturkoordinaten in CityGML**
 - **Problem:** Zuordnung der Texturkoordinaten zu Geometrien
 - GML sieht keine Texturkoordinaten vor – CityGML hält sich an GML
 - Grund in GML: GML ist für Geometrie und nicht für das Aussehen von Objekten konzipiert
 - CityGML definiert GML-Erweiterung zur Texturierung
 - Es kann zur Visualisierung nur gerendert werden, wenn zu jedem Umringspunkt eine Texturkoordinate existiert
 - Konkrete Probleme:
 - Teilproblem 1: „Abstand“ zwischen Texturkoordinaten und Geometrien im XML-Dokument zu groß, für Visualisierungs-Werkzeuge schwierig zu lesen
 - Teilproblem 2: Eine Textur für mehrere Polygone: mehrfache Nennung des Dateinamens und jeweils Texturkoordinaten.
 - Bei Exporten werden Objekte i.d.R. trianguliert → sehr viele Flächen → häufige Wiederholung der allg. Texturinfos → Explodieren der Dateigröße
 - Teilproblem 3: Zusammengesetzte Flächen (CompositeSurfaces, MultiSurfaces oder Surfaces, die aus Patches bestehen) können nicht texturiert werden, da diese keinen expliziten Umring haben
 - Teilproblem 4: Dreiecke können nicht einzeln texturiert werden, da Dreiecke (=Patches) keine Surfaces sind
 - Teilproblem 5: Dreiecksnetze können in GML nicht effizient repräsentiert werden (unabhängig von der Texturierung). Klasse *gml:Tin* ist auf Delaunay-Triangulation eingeschränkt.
 - **Vorschläge:**
 - Jeden Punkt in 5D überführen (die letzten beiden Dimensionen für Texturkoordinaten)

- Problem: kein richtiges GML mehr, 5D-Referenzsystem muss angegeben werden
- Spezialisierung jeder Geometrieklasse XYZ zu „TexturedXYZ“
 - Beispiel:


```

<TS>
  <bS>
    <SimpleTex>
      <Triangle>
        <TTriangle>
          ...
        </TTriangle>
      </Triangle>
    </SimpleTex>
  </bS>
</TS>
          
```
 - Nachteil: für jede Geometrieklasse, die in CityGML verwendet wird (Surface, CompositeSurface, ..) muss Unterklasse gebildet werden.
- MultiSurface – Zuordnung der Texturkoordinaten zu TexturedSurface
- Trennung der bisherigen TexturedSurface in 2 Klassen; eine mit Angaben zur Textur (Dateiname,..) und eine mit Texturkoordinaten. Die zweite Klasse referenziert die Erste; somit müssen Angaben zu Textur nicht jedes Mal wiederholt werden. Dies bringt Ersparnisse, wenn eine Textur sehr oft verwendet wird.
- Neue Klasse „TexturedTriangleMesh“ (explizite Angabe der Triangulation durch Reihenfolge der Punkte – 3 aufeinanderfolgende Punkte = 1 Dreieck).
 - Es muss nicht zwingend die Delauny-Triangulation verwendet werden, die für beliebige Rasterpunkte eine eindeutige Triangulation beschreibt.
- **Fragen:**
 - Ist das wirklich effizienter? Man spart doch kaum was...
 - Ist das alles noch GML-konform?
 - GML wird wohl kurzfristig nicht geändert werden können!
- Visualisierung als Fachschale???
 - Fachschale ist eher auf Features bezogen, weniger auf Geometrie...
- Wie wird ein Grid texturiert?
 - Ist die Textur für ein DGM überhaupt wichtig/sinnvoll? CityGML soll 3D darstellen. Müssen dann überhaupt texturierte DGMs her?
 - Ein Orthophoto trägt enorm zur Akzeptanz bei, weil in der entsprechenden Detaillierungsstufe nicht modelliert ist, und aus dem Orthophoto viele Informationen bezogen werden können.
- Das Thema Texturierung sollte auf der nächsten Sitzung behandelt werden
- **Wie können Fachschalen in CityGML integriert werden?**
 - **Vorschläge:**
 - Hinzufügen von generischen Attributen (GenericAttributes)
 - Generische Attribute können nicht gefiltert (=als Selektionskriterium in Anfragen verwendet) werden (dies ist ein

- generelles Problem des OGC Filters, ggf. sollte die bisherige Umsetzung der `GenericAttributes` überdacht werden)
- Eigenes XML-Schema von CityGML ableiten (durch Vererbung)
 - GML-Reader könnten ein solches neues Schema ggf. nicht richtig lesen. Werkzeuge interpretieren das XML-Schema nicht und erkennen nicht, dass z.B. `NoiseRoad` ist eine spezielle `Road` ist.
 - Weiteres Problem: Die Oberklasse `AbstractBuilding` kann keine neuen Ableitungen bekommen, weil sich die Änderungen nicht auf die Kinder `Building` und `BuildingPart` auswirkt. Es müssen folglich die identischen Ergänzungen an `Building` und `BuildingPart` gemacht werden.
 - Fachschale definiert neuen Namensraum für neue generische Attribute
 - Man könnte das CityGML-Schema dazu bringen, (an bestimmten Stellen) Elemente aus beliebigem Namensraum zuzulassen. Wie macht man das?
 - CityGML wird ergänzt um ein generisches Attributkonzept (ein anderes als die `GenericAttributes`)
 - Definition einer generischen *Property*, die die dann in dem Namensraum der Fachschale definiert werden kann. Diese *Property* sollte einfache Attribute, aber auch komplexe aufnehmen können
 - geklärt werden muss, ob dies GML bzw. XML-technisch geht. Es muss gültiges GML bleiben
 - Reader darf nur das Lesen, was das CityGML-Schema hergibt, und jedes andere Element übersehen.
 - **Fachschalen für CityGML, Beispiel: Umsetzung der Umgebungslärm-Richtlinie**
 - Spezielles Problem: `TransportationComplex`: Aufteilung einer `Road` durch Bilden von Knotenpunkten (zwischen Segmenten unterschiedlicher Beschaffenheit) und Abschnitten sind zur Lärmberechnung notwendig. CityGML sieht aber erst in höheren LOD flächenhafte Segmente vor
 - Problem der linearen Referenzierung evtl. in der Fachschale einführen. Lärmsoftware müsste Geometrie zerlegen. (Ggf. sollte CityGML um das Konzept der linearen Referenzierung ergänzt werden)
 - Aggregation von `NoiseRoadSegments` zu `Road` in CityGML-Schema zu ermöglichen ohne, dass danach das Schema verletzt wird (vgl. Ergänzung um *Properties*)