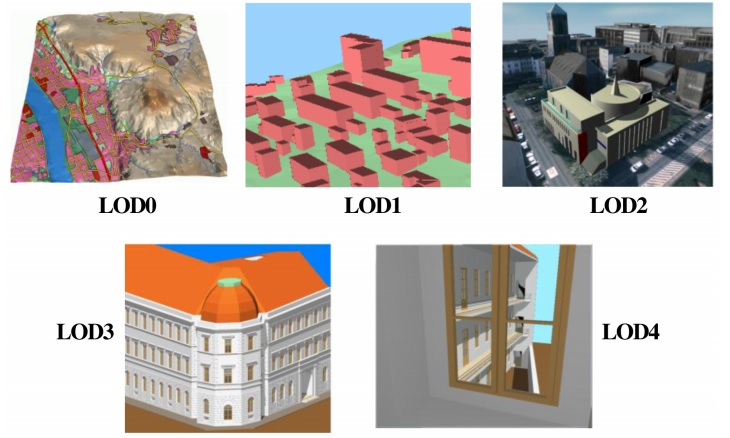
1 Benutzte Bibliotheken

1.1 CityGML:  
  
CityGML wurde 2002 in NRW in offener Arbeitsgemeinschaft unter Mitarbeit von über 70 Institutionen entwickelt. Es ist ein XML-basiertes Datenmodell für die Speicherung von 3D-Stadtmodellen. Es werden relevante Objekte des städtischen Raumes klassifiziert und ihre Eigenschaften beschrieben. Auch ohne Visualisierung lassen sich bereits viele Informationen aus den XML-Daten extrahieren, da verschiedene thematische Attribute wie zum Beispiel die Adresse oder das Baujahr vieler Gebäude mit angegeben sind. Seit 2004 wird CityGML als internationale Standardisierung vorangetrieben und auch zunehmend in der Forschung verwendet. Dies ist an der großen Bandbreite an Einrichtungen zu erkennen, die eigene Parser oder Visualisierungen für CityGML anbieten. Das KIT (Karlsruhe), die Universität Bonn, die TU Berlin und viele weitere vermessen Gebäude vieler vorwiegend größerer Städte (zum Beispiel Berlin, Hamburg, Frankfurt, Stuttgart, ...). Grundsäztlich finden einige Basisklassen in CityGML verwendung: Gebäude, Gelände, Vegetation, Wasserkörper, Straßenmöbel und einige weitere. Für diese Arbeit sind die Gebäude wichtig, andere geographische Merkmale finden bisher keine Verwendung. Jede der genannten Basisklassen besitzt wiederrum ein “Level Of Detail” (LOD), also eine Angabe wie detailliert die XML-Daten angegeben wurden. Die Stufen sind am Beispiel von Gebäuden wie folgt zu verstehen:



5   
  
  
  
  
  
  
  
  
  
  
Skalenbereiche, Levels of Detail, LOD:  
  
Die LOD-Stufen nach [GRÖGERet al.] am Beispiel von Gebäuden:

• LOD 0 bietet 3D-Landmarken, die gegebenenfalls mit Luftbildern versehen werden kann. Man nennt es “Regionalmodell”.

• LOD 1 beinhaltet Häuser quaderförmiger Struktur. Materialeigenschaften können angegeben werden.

• LOD 2 besitzen einzelne Oberflächen eine eigene Geometrie, Dächer und Wände werden vom restlichen Gebäude unterschieden. Einzelne Flächen können hier mit Eigenschaften belegt werden.

• LOD 3 ermöglicht detaillierte Hausfassaden mit “Löchern” in der Textur (zum Beispiel Fenster, Türen, ...).

• LOD 4 beinhaltet auch die Innenarchitektur der Gebäude samt Einrichtungsgegenständen  
  
Für die Volumenberechnung in diesem Projekt wurde die Detailstufe “LOD 2” gewählt, da sie die Ergebnisse bei hinnehmbarem Aufwand schon genau genug abbilden kann und zudem viele Testdaten in dieser Stufe vorhanden sind.

1.2 StAX

Die “Streaming API for XML” (StAX) wurde 2006 vom “Java Community Process” ist dafür gedacht, XML-Dateien mit Java zu verarbeiten. Es stellt einen Mittelweg zwischen den zuvor beliebten DOM- und SAX-Parsern dar. Statt einer Baumstruktur, die komplett in den Arbeitsspeicher geladen wird, wie es bei DOM-Parsern der Fall ist, verwendet es einen Mechanismus der die Daten genau dann abruft, wenn sie gebraucht werden. Statt wie beim SAX-Parser mit mehreren Durchläufen zu arbeiten, ist es mit StAX möglich, den Baum dynamisch zu verändern.

1.3 JAXB

Neben StAX wird in diesem Projekt noch die API “Java Architecture for XML Binding” (JAXB) als Alternative zu SAX und DOM verwendet, da sich herausstellte, dass die Möglichkeiten zum Bearbeiten der XML-Dokumente aus der Java-Anwendung wesentlich komfortabler möglich ist.

1.4 OpenGL

OpenGL wurde 1992 von der Khronos Group, einem Industriekonsortium mit namhaften Mitgliedern wie zum Beispiel AMD, Google und Oracle entwickelt und stetig weiterentwickelt. In diesem Projekt wird OpenGL zum Triangulieren genutzt. In Kapitel [???] wird genauer darauf eingegangen.

2 Prototyp

Zum Einarbeiten in CityGML und um dessen Funktionsweise zu verstehen, wurde ein Prototyp geschaffen. Er entstand in der ersten Iteration und wurde stetig weiterentwickelt, sobald Fortschritte erfolgt waren. In Kapitel [???] werden die Probleme Erläutert, die dabei enststanden.

3 Probleme und Lösungen  
3.1 Namenskonflikte mit Datenmodell

Nachdem das Datenmodell entworfen war, entstanden in der Parser-Gruppe Namenskonflikte, die wir nur durch direktes “Ansprechen” der gewünschten Klasse beheben konnten (zum Beispiel org.citygml4j.model.citygml.building.Building Building, damit nicht das Building unseres eigenen Datenmodells verwendet wird).

3.2 Laufzeit

Das Parsen der XML-Daten benötigt nach wie vor relativ lange Zeit. So hat die Beispieldatei eines kleinen Vorortes von Ludwigsburg weniger als 100MB Inhalt. Wählt man das Stadtmodell größer, werden definitiv nicht hinnehmbare Zeiten entstehen. Das Stadtmodell von Berlin ist zum Beispiel 21,1GB groß und lässt erahnen, dass die Laufzeit für unser Projekt in Ordnung ist, für größere Projekte an der Performanz aber noch gearbeitet werden muss.

3.3 Genauigkeit (ungelöst)   
  
Durch das “casten” von double-Werten und dem damit verbundenen Abschneiden von Nachkommastellen, verlieren die Daten an Genauigkeit. Da die Weiterverarbeitung jedoch mit double-Werten nicht funktioniert, muss dieser Schritt beim Parsen unternommen werden. Denkbar wäre eine andere Datenstruktur, die die Daten in zwei Abschnitten bearbeitet und das Ergebnis wieder zusammenführt.

3.3 CityGML-Dokumentation

Zum Zeitpunkt der Erstellung des Prototypens gab es nicht viele Tutorials oder Beispiele im Internet zu finden, mit denen man hätte arbeiten können. So musste an manchen Stellen “gecastet” werden, die wir ohne die Hilfe von Prof. Dr. Coors nur mit wesentlich mehr Zeitaufwand eventuell gefunden hätten. Um auf den Namen der Stadt zuzugreifen in dem sich ein Gebäude befindet, ist zum Beispiel folgende Verkettung von Methodenaufrufen notwendig, um den XML-Baum bis an die gewünschte Stelle hinab zu gehen:

String theCity = building.getAddress().get(0).getAddress().getXalAddress().getAddressDetails().getCountry().getLocality().getLocalityName().get(0).getContent();  
  
Dabei sind theoretisch auf jeder Ebene NullPointerExceptions möglich, die abgefangen werden sollten.

Abbildungsverzeichnis:

citygml2: CityGML Encoding Standard, Kolbe et al. 2012

Quellen:  
Fachreferate:

T.H. Kolbe, 2008, CityGML - Ein Standard für virtuelle 3D-Stadtmodelle, Skript  
T.H. Kolbe, 2005, CityGML, OGC TC Meeting Bonn (Open Geospatial Consortium Technical committee meeting)  
  
Veröffentlichungen:  
C.A.L.Sánchez, 2013, Estimation of Electric Energy Demand using 3D City Models, Master Thesis  
Löwner et al., 2013, CityGML 2.0 – Ein internationaler Standard für 3D-Stadtmodelle, Zeitschrift für Geodäsie, Geoinformation und Landmanagement (138 Jg., Heft 2)

G.Gröger, Lutz Plümer, 2012, CityGML - Interoperable semantic 3D city models, ISPRS Journal of Photogrammetry and Remote Sensing  
Felix Kunde, 2012, CityGML in PostGIS, Master Thesis, Universität Potsdam