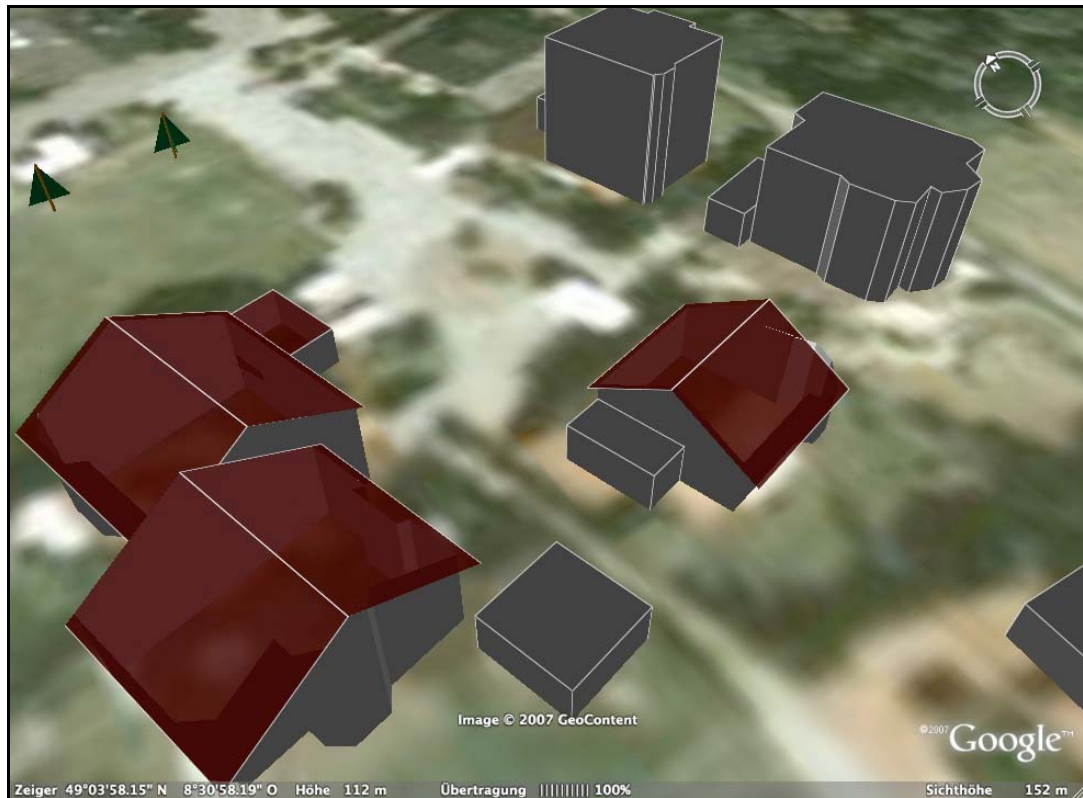


Dokumentation

Erweiterung (LoD2) des KML Converter für das WB3 Projekt



Autor  
Markus Hoffmann  
Im Rahmen des Karl-Steinbuch-Stipendiums

# 1. Inhaltsverzeichnis

<b>1. INHALTSVERZEICHNIS</b>	<b>2</b>
<b>2. EINLEITUNG</b>	<b>3</b>
2.1. Einführung	3
2.2. Aufgabe	3
<b>3. GRUNDLAGEN</b>	<b>4</b>
3.1. Das WB3-Projekt	4
3.2. 3D-Stadtmodelle	7
3.3. Level of Detail (LoD)	8
3.4. Keyhole Markup Language (KML)	8
3.5. XSL-T	9
<b>4. IMPLEMENTIERUNG</b>	<b>10</b>
4.1. Entwicklungsumgebung und Programmiersprache	10
4.2. Programmablauf	10
<b>5. ERGEBNISSE</b>	<b>14</b>
5.1. Generierte Beispieldatei	14
5.2. Google Earth	16
<b>6. MÖGLICHE VERBESSERUNGEN</b>	<b>17</b>
6.1. KML Einschränkungen	17

## 2. Einleitung

### 2.1. Einführung

Google Earth ist eine in ihrer Grundversion kostenlose Software. Die einfachen Navigations-, Mess- und Präsentationswerkzeuge machen sie zur beliebtesten 3D-Atlas-Software. Neben diesen Werkzeugen zur Betrachtung, können auch eigene Geodaten, Rundflüge und Modelle abgespeichert und wieder geladen werden.

Eine immer größer werdende Benutzeranzahl stellt eigene Daten zur Erweiterung der Basis-Daten von Google Earth<sup>1</sup> via Internet zur Verfügung. Neben einfachen Markierungen, erweiterten Orts-Informationen, Fotos, und Reiserouten werden auch 3D-Modell von einzelnen Gebäuden, Siedlungen oder ganzen Städten angeboten. Diese Modelle können beispielsweise durch Verwendung von Sketch-Up (einer einfachen CAD-Software) erstellt werden.

Durch die steigende Verbreitung und das immer größere Interesse an Google Earth wird es besonders interessant auch bereits bestehende 3D-Daten in anderen Formaten für Google Earth zu transformieren.

### 2.2. Aufgabe

Diese Dokumentation beschreibt die Entwicklung eines Exports von KML konformen Daten für das WB3-Projekt des media::lab der Hochschule Karlsruhe unter Leitung von Herrn Prof. Dr. Henning.

Das WB3-Projekt wurde bereits durch verschiedene Exportmöglichkeiten erweitert, darunter auch ein Export für das „Keyhole Markup Language“-Datenformat (KML) für Google Earth. Dieser Export beherrscht bisher nur die Darstellung des einfachsten Detaillierungsgrades.

Der Export soll dahingehend erweitert werden, dass neben einfachen extrudierten Grundflächen und Grundstücksbeschreibungen auch Dachformen und Vegetation in Google Earth angezeigt werden können. Das heißt, dass per Knopfdruck auch ein LoD2 – Stadtmodell für Google Earth generiert werden kann.

---

<sup>1</sup> Quelle: <http://earth.google.com/>

## 3. Grundlagen

### 3.1. Das WB3-Projekt

WB3 ist ein Projekt des media::lab<sup>2</sup> der Hochschule Karlsruhe unter Leitung von Herr Prof. Dr. Henning. Es stellt die konkrete Realisierung eines 3D-Stadtmodells auf Basis von Grundstücks- und Gebäudedaten der öffentlichen Verwaltung.

Die Liegenschaftsämter der deutschen Gemeinden und Städte verfügen mit diesen Daten über bisher weitestgehend ungenutzte Information. Ziel des WB3-Projekts ist es diesen Daten einen Mehrwert zu verleihen: Über ein Webinterface sollen interessierte Personen (fachliche, politische Planer, Eigentümer) Zugriff auf die WB3-aufgearbeiteten Daten und damit weit mehr als nur Zahlen erhalten.

The screenshot shows a web browser window with the address [http://212.227.99.140:8688/waldbruecke/wb3\\_id.html?id=187xx1](http://212.227.99.140:8688/waldbruecke/wb3_id.html?id=187xx1). The page title is 'WB3 Flurstück 187xx1'. Below the title, there are two main sections: 'Flurstück 187xx1' and 'Gebäude 187xx1'. Each section contains a table of data. The 'Flurstück' table includes columns for 'Eigentümer', 'Adresse', 'Nachbarn', 'Koord. System', 'Eckpunkte', and 'Grundfläche'. The 'Gebäude' table includes columns for 'Eigentümer', 'Adresse', 'Drehwinkel', 'Referenzpunkt', 'Eckpunkte', 'Grundfläche', 'Bounding Box', 'Geschosse unterirdisch', 'Geschosse oberirdisch', 'Gemessene Höhe', 'Baujahr', 'Firstlinie', 'Dachtyp', and 'Funktion'. At the bottom of the page, there is a 3D rendering of a green plot with a grey building footprint. The rendering is titled 'Render as' and includes options for 'SVG PNG JPEG' and '3D CityGML, Mathematica'. There is also an 'Edit' link and a footer with navigation links like 'Hauptseite', 'Gesamtansicht', 'Datenschutz', 'Beiträge', 'Impressum', 'Glossar', and 'Logout'.

*Das Web-Interface des WB3-Projekts*

Neben den rohen Daten aus der öffentlichen Verwaltung, werden weitere aufbereitete Daten angeboten. Konkret handelt es sich um grafische zwei- und dreidimensionale Daten. Diese werden nicht durch aufwendige händische Modellierung erarbeitet, sondern vielmehr im Moment des Abrufs aus den zweidimensionalen Katasterdaten der Ämter erzeugt.

Zur Generierung von beispielsweise 3D-Gebäudemodellen in niedriger Detailstufe genügen die einfachen Eckpunkte und eine Höhenangabe. Definiert man zusätzlich eine Firstlinie z.B. durch die Analyse von Luftfoto, kann auch eine realistische Dachkonstruktion erzeugt wer-

<sup>2</sup> <http://www.medialab-karlsruhe.de/>

den. Existiert ein noch detailliertes Modell des Gebäudes, so kann dieses beispielsweise durch VRML eingebunden werden. Die generierte dreidimensionale Darstellung wird primär im X3D-Format abgelegt, und kann damit in einer Vielzahl von Viewer-Software betrachtet werden.

Das WB3 Projekt schreitet mit Innovation auf inhaltlicher und technischer Ebene voran<sup>3</sup>. Wie bereits angedeutet arbeiten die meisten Verfahren zur Erstellungen virtueller Abbilder realer Städte auf Basis von Neumodellierung. Diese ist zum einen sehr aufwändig, zum anderen flächendeckend kaum realisierbar, da nicht finanzierbar.

Das WB3 Projekt arbeitet stattdessen auf Basis von Transformationen bereits existierender Daten und schlägt so eine Brücke von der „informationellen Gegenwart“ zur Zukunft, dem „flächendeckenden Einsatz der 3D-Visualisierung.“ Anwendungsbeispiele findet man in en Bereichen

- 1.Siedlungsplanung (Gebäude, Vegetation, Gelände)
- 2.Simulationen im Bereich Emissionsschutz (Lärm, Luftbewegung) und Verkehr
- 3.Orientierung (3D-Stadtplan, Virtuelle Städtetouren, Virtuelle Geschäftsstraße)

Eine Anwendung bei der die inhaltliche Innovation des WB3 Projekts deutlich wird, könnte beispielsweise die 3D-Ansicht einer Stadt umfassen. Nähert man sich mit den Anwendungswerkzeugen (z.B.: Zoomen) einer bestimmten Stadtszene, werden Verkehrswege eingeblendet. Ähnlich könnte beim Betrachten bestimmter Objekte verfahren werden. Fokussiert man eine Einrichtung wie ein Kino oder eine Gaststätte werden das Kinoprogramm bzw. die Speisekarte in einem separatem Fenster geöffnet. Nähert man sich bestimmten Objekten noch weiter, kann das durch WB3 generierte Modell durch ein detailliertes 3D-Modell erweitert werden und je nach Aufwand „virtuell“ besucht werden.



WB3-Ausgabe der Siedlung Waldbrücke in zweidimensionaler und dreidimensionaler Form

Die zuvor angesprochene technische Innovation äußert sich in der Tatsache, dass WB3 unter Einsatz neuester XML-Technologien entwickelt wurde. Eingesetzt wurden XSL-Transformationen (siehe Seite **Fehler! Textmarke nicht definiert.**) gesteuert durch Stan-

<sup>3</sup> Quelle: Dokument WB3 – Projektbeschreibung und Benutzungsanleitung

## Erweiterung (LoD2) des KML Converter für das WB3 Projekt

Standardsoftware, einem Cocoon Server. Dieser ist ein Open Source-Produkt, der Software Cocoon und Bestandteil des Apache<sup>4</sup> Webserver-Projektes.

Die Zuführung dieses Mehrnutzen, verleiht den Katasterdaten wie beschrieben eine neue Bedeutung und wurde bereits honoriert: Das WB3-Projekt wurde im Juli 2006 mit dem doIT-Forschungspreis ausgezeichnet.

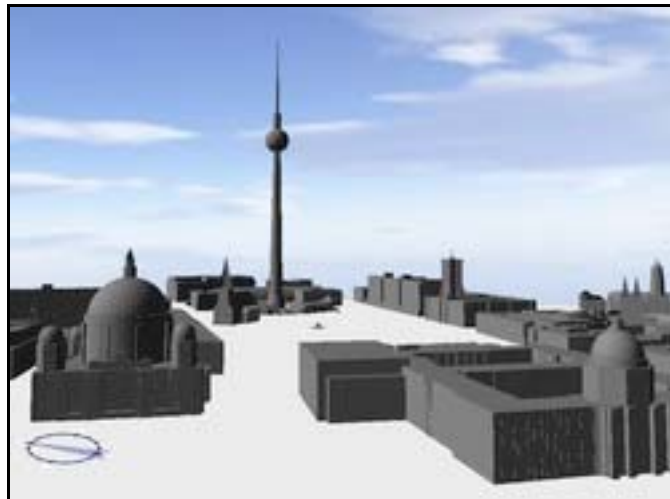
---

<sup>4</sup> Quelle: <http://cocoon.apache.org/>

### 3.2. 3D-Stadtmodelle

Digitale 3D-Stadtmodelle ermöglichen eine virtuelle Darstellung von bereits existierenden oder erst geplanten Gebäuden und Komplexen bis hin zu ganzen Stadtszenen. Dabei soll mit Hilfe des Computers ein idealisiertes 3D-Modell als Abbildung des realen Stadtbilds geschaffen werden.

Zur Modellierung der Elemente der realen Welt werden in 3D-Stadtmodellen Geoobjekte eingesetzt. Je nach Anwendung, Detaillierungsgrad werden mehr oder weniger bzw. detailliertere oder weniger detailgetreue Geoobjekte abgebildet. Das wichtigsten Geoobjekte von 3D-Stadtmodellen sind Gebäudeobjekte. Die weiteren möglichen Objektarten dienen beispielsweise der Modellierung von Gelände oder Vegetation. Es können aber auch beliebig viele weitere reale Objekte modelliert werden. Darunter auch Infrastrukturelle Einrichtungen wie Verkehrsmittel, natürliche Begebenheiten wie Gewässer oder ähnliches.



*CityGML-Model der Stadt Berlin (Quelle: <http://www.citygml.org>)*

Immer mehr Kommunen entdecken das Thema 3D-Stadtmodell für sich. Neben dem zunehmendem Interesse steigert auch die neue Umgebungslärmrichtlinie (Juni 2005) der EU den Bedarf an Geodaten, die detaillierter sind als bisher erfasste Daten. Die deutsche Initiative CityGML<sup>5</sup> entwickelt sich zunehmend schneller. Die Hauptstadt Berlin hat begonnen große Bezirke des Ballungsraums zu erfassen und mit CityGML zu abbilden.

Im Rahmen des WB3 Projektes entstehen aus Katasterdaten 3D-Stadtmodelle. Je nach Informationsgehalt können detailliertere Stadtmodelle generiert werden. Den Detaillierungsgrad unterscheidet man anhand von Detaillierungsebenen (Level of Detail – LoD).

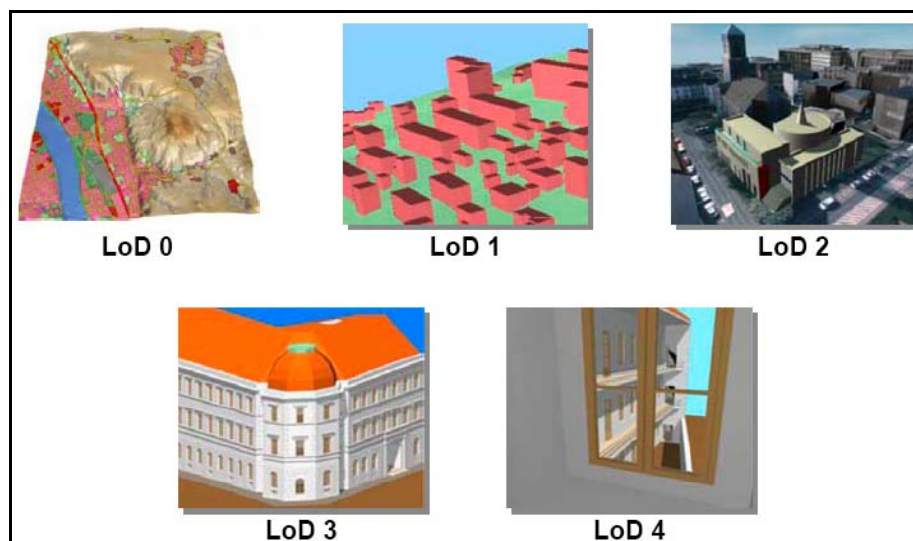
---

<sup>5</sup> Quelle: <http://www.citygml.org>

### 3.3. Level of Detail (LoD)

Mit dem „Level of Detail“ beschreibt man die verschiedenen Detailstufen bei der Darstellung virtueller Welten, so auch bei Stadtmodellen.

Um die Geschwindigkeit von Anwendungen der zu steigern, aber auch die Datenmenge zu reduzieren setzt man meist auf gleichzeitiges Nutzen von hoher Detailgenauigkeit im Nahbereich und niedriger Detailstufe im Fernbereich. Dies betrifft sowohl die Komplexität als auch die Anzahl dargestellter Geobjekte. Man kann z.B. ein 3D-Stadtmodell (nach CityGML) in mehreren Detailstufen darstellen:



Beispiele für die fünf verschiedenen Detaillierungsgrade (Quelle: <http://www.citygml.org>)

LOD 0 – Geländemodell

LOD 1 - Gebäudeblock mit Textur

LOD 2 - 3D-Modell der Außenhülle (Dachformen)

LOD 3 - 3D-Modell der Außenhülle mit Textur

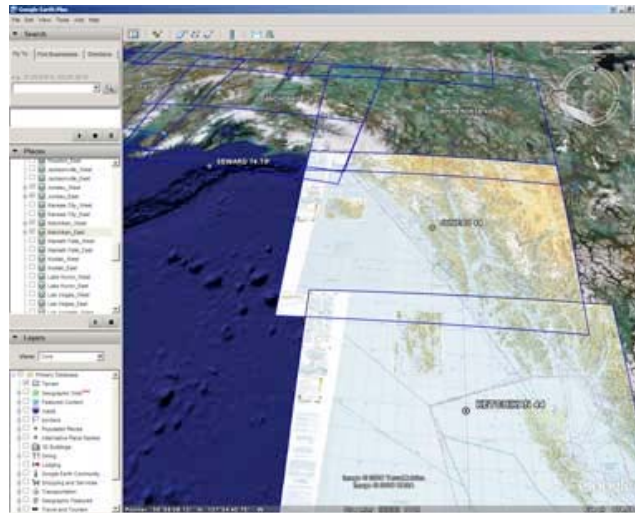
LOD 4 - 3D-Modell des Gebäudes mit Etagen, Innenräumen, etc. und Texturen

Je nach Datenformat bzw. eingesetzter Sprache zur Modellierung kann sich die LoD-Definition unterscheiden.

### 3.4. Keyhole Markup Language (KML)

Die Keyhole Markup Language (KML<sup>6</sup>) ist ein Dateiformat, das von der Firma Keyhole bzw. heute Google für die Anwendungen Google Earth und Google Maps entwickelt wurde. Mittels KML können geographische Daten abgebildet und in einem sogenannten Earth-Browser dargestellt werden.

<sup>6</sup> <http://code.google.com/apis/kml/documentation/>



*Beispiele für KML-Erweiterungen in Google Earth*

KML ist eine XML Anwendung und beschreibt als Auszeichnungssprache geführte Daten mit Tags, das heißt Namen und Attribute. Die aktuelle Version 2.1 kann verwendet werden um:

- Markierungen und Bezeichnung zur Beschreibung von Orten zu spezifizieren,
- Kamerapositionen zu definieren,
- Bilder zu platzieren und zu überlagern,
- Styles zu definieren, d.h. Farben und Schriftarten für die Präsentation hier genannter Objekte,
- HTML Beschreibungen zu integrieren,
- Punkte, Linien, Flächen zu definieren,
- einfache 3D-Objekte durch Polygon-Ringe bzw. Flächen zu zeichnen.

### **3.5. XSL-T**

XSL-T steht für XSL Transformations und ist Teil der Extensible Stylesheet Language (XSL). XSLT ist eine Programmiersprache zur Transformation von XML-Dokumenten. Es nutzt die logische Baumstruktur eines XML-Dokuments und erlaubt die Definition von Transformationsregeln auf diesen Baum. XSL-T-Programme werden als XSL-T-Stylesheets gespeichert, die selbst wieder dem Aufbau einer XML-Datei folgen.

Die Sprache XSLT beschreibt die Umwandlung eines XML-Dialekts in ein anderes Dokument. Diesen Vorgang nennt man Transformation. Das resultierende Dokument entspricht häufig der XML-Syntax, es können aber durchaus auch andere Dateiformate erstellt werden.

Eine Transformation besteht aus Transformationsregeln, diese werden Templates genannt. Ein Template besitzt ein auf XPath Muster, das Knoten definiert für die es gilt. Trifft man während des Programmablaufs auf einen solchen Knoten im Quelldokument, werden die entsprechenden Transformationsregeln des Templates aufgerufen und im Zieldokument ausgegeben.

## 4. Implementierung

### 4.1. Entwicklungsumgebung und Programmiersprache

Implementiert wurde eine XSL-Transformation mit Hilfe der XML Entwicklungsumgebung Oxygen<sup>7</sup> unter Mac OSX. Die entwickelten Transformationsregeln sind alle in der Datei „wb\_ex\_kml2.xsl“ gespeichert. Da XSL keine klassischen Variablen kennt, sondern nur das einmalige Beschreiben von Variablen bei ihrer Definition erlaubt, sind die meisten Probleme rekursiv zu lösen.

### 4.2. Programmablauf

Grundsätzlich wird die KML-Transformation beim Auftreten des `city`-Tags gestartet. Daraufhin werden alle Gebäude (Tag: `building`) und die durch den Tag `plant` ausgezeichneten Vegetationsobjekte abgearbeitet. Besteht ein Gebäude aus mehreren Teilen (`buildingparts`), so wird jeder Gebäudeteil für sich bearbeitet.

Zur Konstruktion der Flächen und eines LoD2-Modells in KML werden die bereits berechneten und anhand des Tags `sql:sides` (bzw. für Dachflächen speziell anhand des Tags `sql:roof`) markierten Koordinaten extrahiert und in das KML Format geschrieben. Sind diese Flächen bzw. die beiden Tags nicht vorhanden, wird als „Fallback“ auf den Tag `sql:points` zurückgegriffen und ein LoD1-Modell aus dem Umriss des Gebäudes durch Extrusion generiert.

1. Da KML auf Basis absoluter Längen- und Breitengradangaben arbeitet, müssen die Koordinaten de-referenziert und umgerechnet werden. Dazu wird ein bereits implementiertes Template aufgerufen.

Koordinaten einer Fläche in der Quelldatei:

```
<sql:sides>
(surface 1,4.8 3.16 -1.7,4.8 10.11 -1.7,4.8 10.11 6.38,4.8 3.16 6.38)
... (weitere Flächen)
</sql:sides>
```

(relative Koordinaten (X,Y), Trennzeichen xyz „“, Trennzeichen Koordinate „“)

Koordinaten einer Fläche in der Zieldatei (KML):

```
<Polygon>
...
  <outerBoundaryIs>
    <LinearRing>
      <coordinates>
        8.51485697099399,49.06643163843937,-2
        8.51499767144929,49.0664526275517,-2
```

---

<sup>7</sup> <http://www.oxygenxml.com/>

## Erweiterung (LoD2) des KML Converter für das WB3 Projekt

```
8.51499767144929,49.0664526275517,6.18
8.5149266368809,49.06644203090841,9.18
8.51485697099399,49.06643163843937,6.24
</coordinates>
</LinearRing>
</outerBoundaryIs>
</Polygon>
(absolute Koordinaten (WGS84), Trennzeichen xyz „,“, Trennzeichen Koordinate „,“)
```

2. Da Vegetationsobjekte wie z.B. Bäume nur durch Punkte und Radien beschrieben sind müssen hieraus zunächst Flächen bzw. 3D-Körper berechnet werden. Hierzu wird standardmäßig ein viereckiger Stamm und eine zweiflächige, dreieckige Baumkrone berechnet. Um Bäume der Realität etwas

Quelldaten zu einem Vegetationsobjekt:

```
<plant id="Ah1 ">
  <class>Baum</class>
  <species>Ahorn</species>
  <function>Garten 19029</function>
  <height>4.0</height>
  <trunkdiameter>0.3</trunkdiameter>
  <crowndiameter>2.0</crowndiameter>
  <points>5.0 -2.0</points>
</plant>
```

Zieldaten in für KML (Nur Stamm des Vegetationsobjekts!):

```
<Polygon>
...
  <outerBoundaryIs>
    <LinearRing>
      <coordinates>
        8.51576097780454,49.06647717557085,4
        8.51575816264377,49.06647892463291,4
        8.51575534748301,49.06647717557085,4
        8.51575816264377,49.06647542650878,4
        8.51576097780454,49.06647717557085,4
      </coordinates>
    </LinearRing>
  </outerBoundaryIs>
</Polygon>
```



*Beispiel für die Visualisierung von Vegetationsobjekten.*

3. Zusätzlich zu den hier beschriebenen Geometrischen Informationen werden auch Farbinformationen neu generiert. Diese sind beispielsweise für Dächer rot, Seitenwände weiß, Baumstämme braun, ...

KML Style-definitionen am Dateibeginn:

```
<Document>
  <Style id="roofStyle">
    <PolyStyle>
      <color>dd0000ff</color>
      <colorMode>normal</colorMode>
    </PolyStyle>
  </Style>
  <Style id="wallStyle">
    <PolyStyle>
      <color>ffffffff</color>
      <colorMode>normal</colorMode>
      <outline>0</outline>
    </PolyStyle>
  </Style>
  <Style id="truncStyle">
    <PolyStyle>
      <color>ff004080</color>
      <colorMode>normal</colorMode>
      <outline>0</outline>
    </PolyStyle>
  </Style>
```

## Erweiterung (LoD2) des KML Converter für das WB3 Projekt

Neben den Objektbeschreibungen müssen auch Szenenbeschreibungen in KML zur komfortablen Betrachtung definiert werden. D.h. die Kameraposition, die Objektbeschreibungen usw. müssen bestimmt und in das KML-Format transformiert werden, um in Google Earth nutzbar zu sein.

Definition der Kamera-Startposition vor der Objektdefinition

```
<LookAt>  
  <longitude>8.51485697099399</longitude>  
  <latitude>49.06643163843937</latitude>  
  <range>300</range>  
  <tilt>0</tilt>  
  <heading>0</heading>  
</LookAt>
```

## 5. Ergebnisse

### 5.1. Generierte Beispieldatei

Transformiert man bestehenden Daten des WB3-Projekts mit dem entwickelten KML-Export entsteht eine KML-Datei die für die Anzeige der exportierten Geodaten in Google Earth geeignet ist.

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns:sql="http://apache.org/cocoon/SQL/2.0"
xmlns:kml="http://earth.google.com/kml/2.0">
  <Document>
    <Style id="roofStyle">
      <PolyStyle>
        <color>dd0000ff</color>
        <colorMode>normal</colorMode>
      </PolyStyle>
    </Style>
    ... <!-- weitere Style Definitionen -->
  <Folder>
    <name>187xx1</name>
    <visibility>0</visibility>
    <open>1</open>
    <Placemark> <!-- Objekt Informationen -->
      <description>Wohnhaus mit Garage</description>
      <name> Gebäude: 187xx1187xx1 Adresse: Amselweg Hausnummer:
11 Eigentümer: Bensching</name>
      <LookAt> <!-- Betrachtungswinkel- und Kameraabstand -->
        <longitude>8.51426623310092</longitude>
        <latitude>49.06685815281128</latitude>
        <range>300</range>
        <tilt>0</tilt>
        <heading>0</heading>
      </LookAt>
      <Point>
        <extrude>0</extrude>
        <tessellate>0</tessellate>
        <altitudeMode>clampToGround</altitudeMode>

        <coordinates>8.51426623310092,49.06685815281128</coordina
tes>
      </Point>
    </Placemark>
    <Placemark>
      <styleUrl>#wallStyle</styleUrl>
      <Polygon> <!-- Wandflächen eines Gebäudes -->
        <extrude>0</extrude>
        <tessellate>0</tessellate>
        <altitudeMode>relativeToGround</altitudeMode>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
              8.51426623310092,49.06685815281128,-1.7
              8.51436144737806,49.066872105568,-1.7
```

```

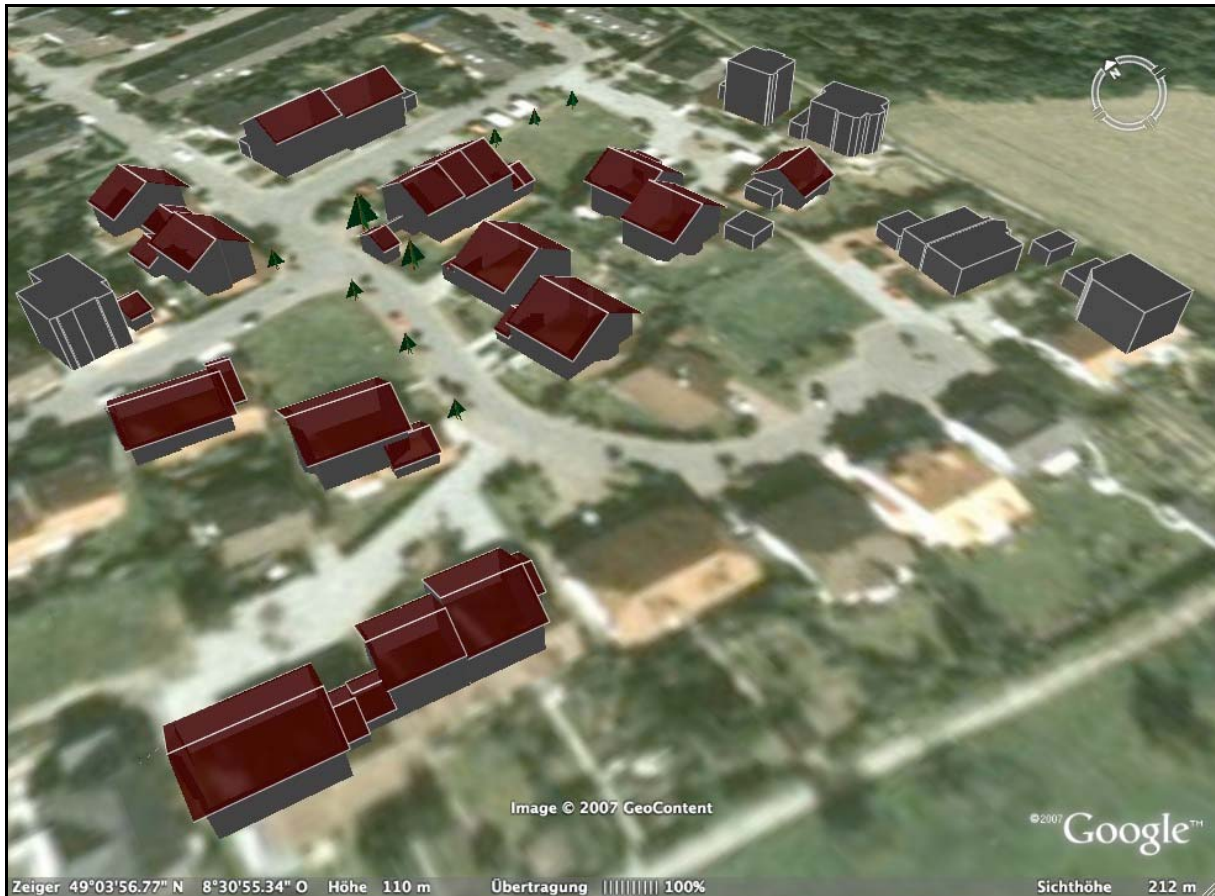
            8.51436144737806,49.066872105568,6.38
            8.51426623310092,49.06685815281128,6.38
        </coordinates>
    </LinearRing>
</outerBoundaryIs>
</Polygon>
</Placemark>
... <!-- weitere Wandflächen -->
    <Placemark>
        <styleUrl>#roofStyle</styleUrl>
        <Polygon> <!-- Dachflächen eines Gebäudes -->
            <extrude>0</extrude>
            <tessellate>0</tessellate>
            <altitudeMode>relativeToGround</altitudeMode>
            <outerBoundaryIs>
                <LinearRing>
                    <coordinates>
                        8.5142819047586,49.06681687078291,9.2
                        8.51437711903574,49.06683082353963,9.2
                        8.51439443864093,49.06678520051446,6.09
                        8.51429922436379,49.06677124775774,6.09
                    </coordinates>
                </LinearRing>
            </outerBoundaryIs>
        </Polygon>
    </Placemark>
    ... <!-- weitere Dachflächen -->
</Folder>
</Document>
</kml>
```

Die Struktur der KML-Datei ist dabei sehr einfach zu verstehen. Flächen werden als Polygone exportiert und werden unterhalb des Knotens Placemark gespeichert. Für jede Fläche ist mindestens ein Linearer Ring aus Punkten zu definieren, der die Berandung des Polygons darstellt. Ist ein Öffnung in der Fläche vorhanden muss diese durch einen zweiten, inneren Ring aus Punkten modelliert werden. Polygone können dabei.

Zusätzliche Informationen wie z.B. die Betrachtungsposition (Tag: LookAt) oder Farbe (Tag: styleUrl) des jeweiligen Polygons werden ebenfalls innerhalb der Placemark-Umgebung definiert.

## 5.2. Google Earth

Die Abbildung zeigt die gesamte Siedlung Waldbrücke in Google Earth. Der in Google Earth geladene KML-Export beinhaltet den vollständigen Datensatz. Dächer, Seitenwände, und Vegetationsobjekte sind farblich unterschiedlich gekennzeichnet und verdeutlichen die Szene. Gebäude zu denen keine Firstlinie definiert wurde, haben keine LoD2 Repräsentation und werden vereinfacht in LoD1 dargestellt.



*Die Siedlung Waldbrücke exportiert in KML und angezeigt in der Atlas-Software Google Earth (Quelle: Google Earth Screen shot)*

## 6. Mögliche Verbesserungen

### 6.1. KML Einschränkungen

Zur Darstellung von Sphären bzw. Kugelartigen Körpern eignet sich KML nicht. Will man beispielsweise Vegetationsobjekte modellieren so repräsentiert eine einfache Kugel die Baumkrone bzw. ein Zylinder den Baumstamm. Diese können in KML jedoch nicht ohne weiteres modelliert werden. Da nur ebene Polygone oder Linien definierbar sind, müsste man eine Kugel zunächst Triangulieren und aus einzelnen Dreiecken zusammensetzen, die jedes für sich als Polygon in KML gespeichert sind.

Nach ausführlicher Recherche ergab sich, dass KML als Dateiformat grundsätzlich nicht für das Abbilden von 3D-Modellen geeignet ist. KML wird viel mehr zur Steuerung und Aggregation verschiedener Dateien zu Präsentationszwecken in Google Earth eingesetzt. Fotos, Beschreibungen, Reiserouten und einfache Polygonartige Objekte können definiert werden, um Google Earth individuell zu erweitern.

Für dreidimensionale Modelle von Gebäudestrukturen ist das Format „KMZ“ geeigneter. Grundsätzlich ist KMZ eine komprimierte KML Datei, die als Archiv Erweiterungen und zusätzlich eingebundene Dateien zulässt. So besteht beispielsweise die Möglichkeit auch COLLADA-Dateien anzuhängen.

COLLADA etabliert sich momentan als XML-Anwendung zum Austausch von Daten für verschiedene Grafik-Software. Neben Modellen und Texturen werden auch Software-Einstellungen usw. in COLLADA gespeichert. Die COLLADA-Beschreibung eines Modells wird als Datei mit einer zugehörigen KML-Datei zur Beschreibung des Ortes und der Kamerablickwinkel im KMZ Archiv gespeichert. COLLADA kennt im Gegensatz zu KML komplexere Objekte und ermöglicht das Definieren eines Zylinders oder eine Sphäre anhand weniger Parameter. Es ist deshalb zu überlegen, ob als nächster Schritt ein KMZ-Export für WB3 sinnvoll wäre.