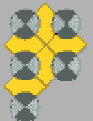


Publizieren von 3D-Modellen mit Google Earth

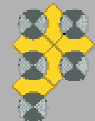
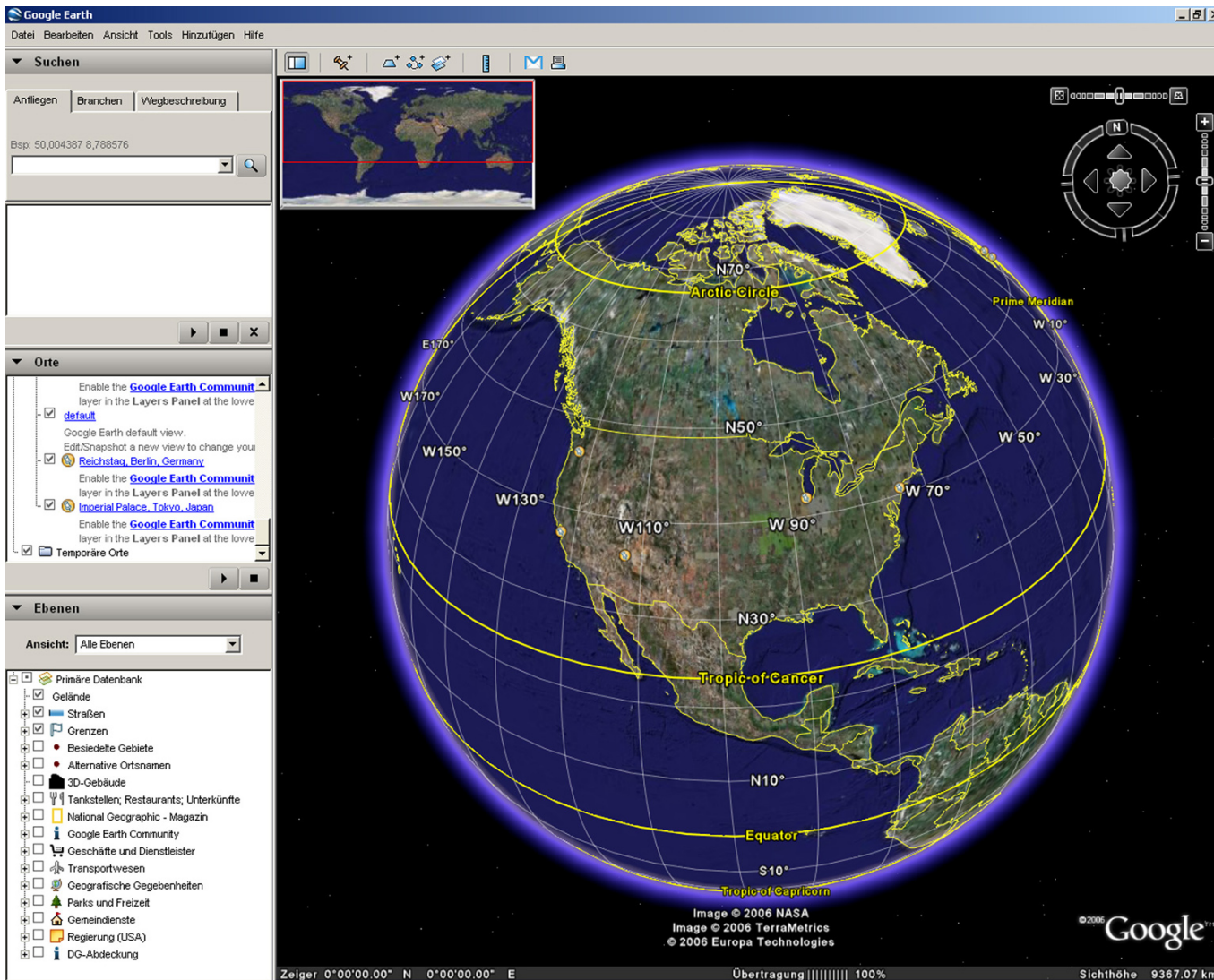
Günter Pomaska, FH Bielefeld



- 1 Virtueller Globus**
 - 1.1 Benutzeroberfläche
 - 1.2 Navigation
 - 1.3 Daten und Datenimport
- 2 KML Keyhole Markup Language**
 - 2.1 Datenstrukturierung
 - 2.2 Ortsmarken
- 3 3D-Modelle mit COLLADA**
 - 3.1 Modellexport nach COLLADA
 - 3.2 Modellimport
- 4 Applikationen**
 - 4.1 3D-Modelle, Style, Radio-Buttons
 - 4.2 Dynamische Applikation
- 5 Referenzen**



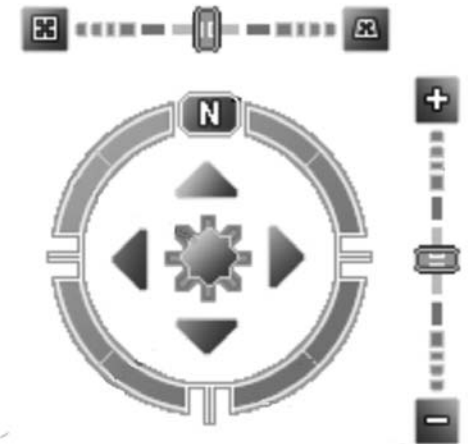
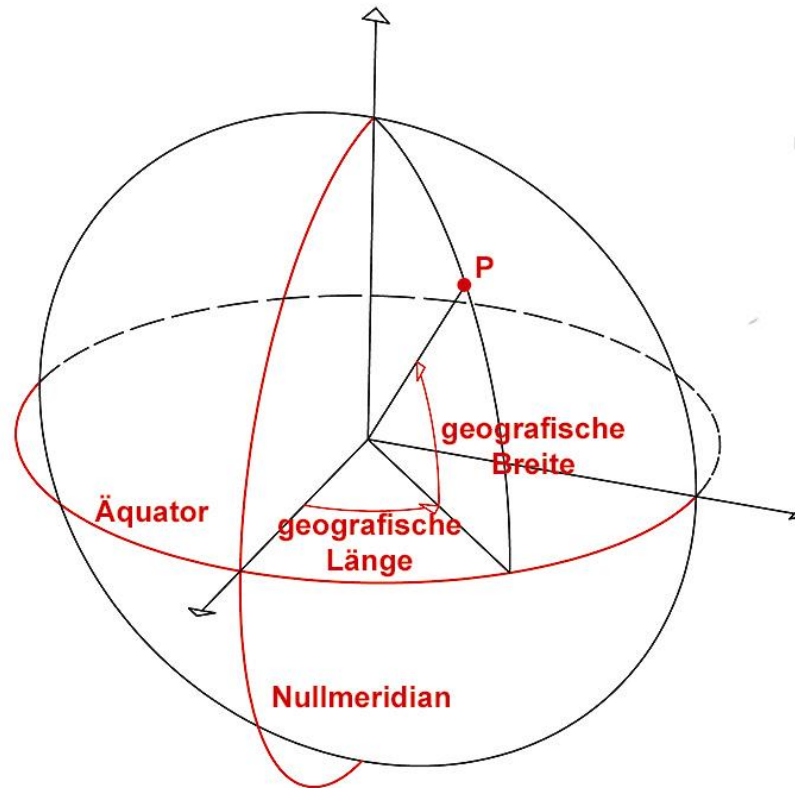
1 Virtueller Globus: Benutzeroberfläche



1.1 Virtueller Globus: Navigation

- Ansicht verschieben
- Gleichmäßig über die Erde driften
- Zoomen
- Ansicht neigen
- Ansicht drehen
- Navigationsmodi

- Space-Mouse

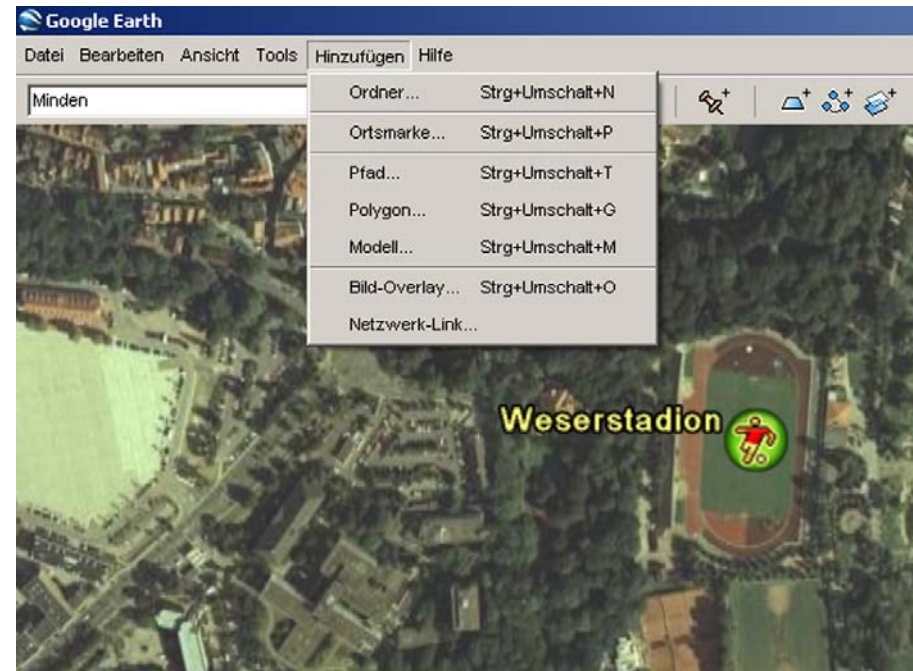


1.2 Virtueller Globus: Daten und Datenimport

- Satellitenbilder
- Luftbilder
- Geländeoberfläche
- Ebenen:
 - Datenbanken
 - Geografie im Web
 - Strassen,

- Ortsmarken
- GIS-Daten (Pfade, Polygone)
- Ground-Overlays
- Screen-Overlays
- 3D-Modelle

- Network Links
- 3D-Warehouse



Markup-Sprachen definieren die Darstellung von Objekten auf einem Ausgabegerät. Programmiersprachen definieren einen Prozess.

KML ist ein XML-basiertes Datenformat zur Modellierung und Speicherung geographischer Informationen zur Anzeige mit dem Google Earth Client.

Elemente: Linien, Polygone, Bilddateien, Icons, Stilvorlagen, Kamerapositionen, Container

Tag-Struktur: Name, Attribute, Inhalt

**Google Earth Client ist ein Browser für KML-Dokumente
ähnlich einem Web-Browser für HTML-Dokumente**



2.1 Strukturierung der Daten

Daten werden in KML in geschachtelten Containern verwaltet, die in einer hierarchischen Eltern-Kind-Beziehung zueinander stehen.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.1">
  <!-- Wurzelement kml -->
  <Document>
    <!-- Dokumentenhierarchie:
      Document, Folder, Placemark
      Document: Features, Styles, Schemas-->
    <Folder>
      <!-- Anordnung von Features:
        Folders, Placemarks, NetworkLinks, Overlays -->
      <Placemark>
        <!-- Daten oder Referenz auf Daten -->
      </Placemark>
    </Folder>
  </Document>
</kml>
```

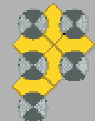
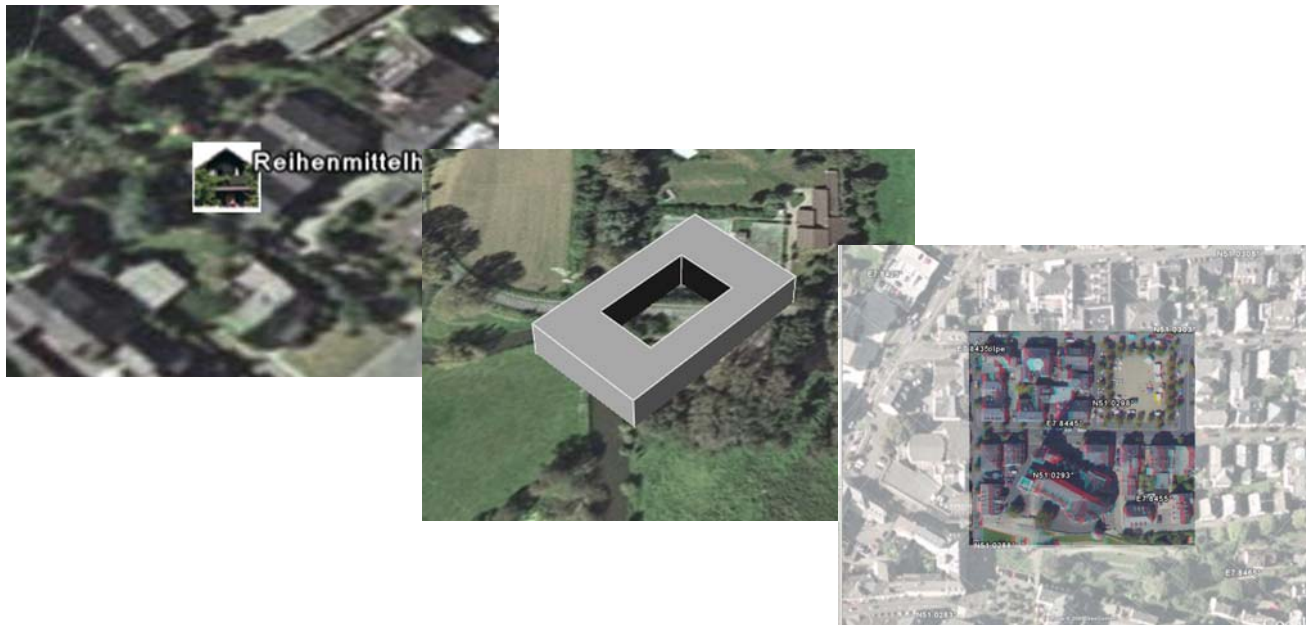


2.2 Ortsmarken (Placemarks)

Placemarks in GE sind vergleichbar mit Bookmarks in Web-Brausern. Sie dienen dem Wiederauffinden von Daten.

Placemarks definieren

- Ort <point>
- Betrachtungsstandpunkt (Kamera) <LookAt>
- Beschreibung (Text) <name><description>
- Geometrie oder Referenz <Model>



2.2.1 Beschreibung der Ortsmarke

<placemark id="ID">

<name>erscheint als Ortsbezeichnung</name>

<snippet maxLines="2">alternativ zu description</snippet>

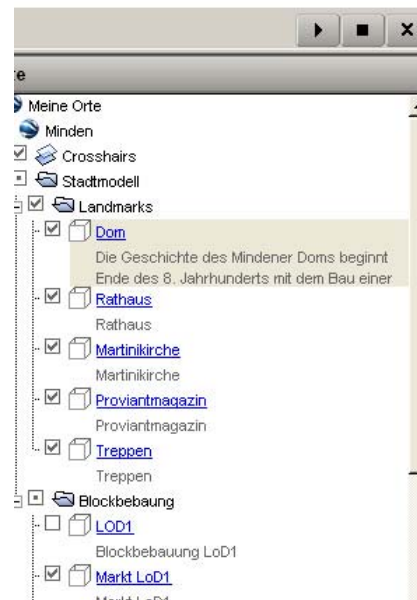
<description>

<![CDATA[Hypertext-Link

HTML-Document, eingebettet in CDATA, Text erscheint

in der Sprechblase]]>

</description>



2.2.2 Kameraeinstellungen – Look At

<LookAt id="ID">

<longitude>geografische *Laenge in Grad –180 bis +180***</longitude>**

<latitude>geografische *Breite in Grad –90 bis +90***</latitude>**

<altitude>*Höhe über der Bezugsflaeche in Meter (altitudeMode beachten)***</altitude>**

<range>*Abstand des Beobachters in Meter (altitudeMode beachten)***</range>**

<tilt>*Winkel zwischen Beobachtungsrichtung und Flächennormale in Grad 0 bis 90***</tilt>**

<heading>*Kompassrichtung in Grad –180 bis 180***</heading>**

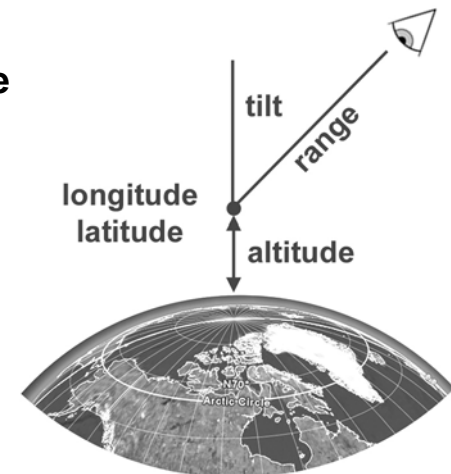
<altitudeMode>*clampToGround, relativeToGround, absolute***</altitudeMode>**

</LookAt>

<!--

Daten können aus der Google Earth Ansicht direkt in die Eigene Ortsmarke übernommen werden

-->



2.2.3 Position der Ortsmarke

```
<Point id="ID">  
  <extrude>Verbindung zum Boden 0/1</extrude>  
  <tessellate>Gelaendeverfolgung der Geometrie</tessellate>  
  <altitudeMode>clampToGround, relativeToGround, absolute</altitudeMode>  
  <coordinates>longitude, latitude, [altitude]</coordinates>  
</Point>  
<!--  
  Geomtrie: Point, LineString, LinearRing, Polygon, MultiGeometry, Model  
-->
```



2.2.4 Modellreferenz

<Model id="ID">

<altitudeMode>*clampToGround, relativeToGround, absolute***</altitudeMode>**

<Location>

<longitude>*<longitude>***</longitude>****<latitude>***</latitude>***</latitude>****<altitude>***</altitude>***</altitude>**

</Location>

<Orientation>

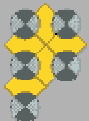
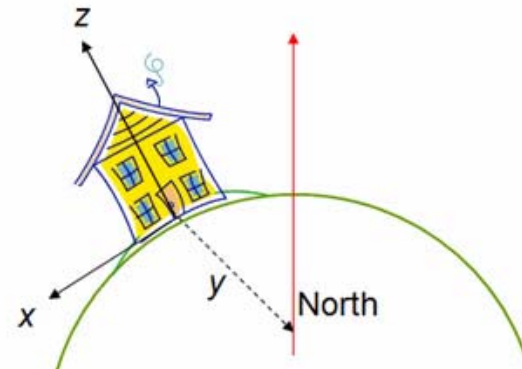
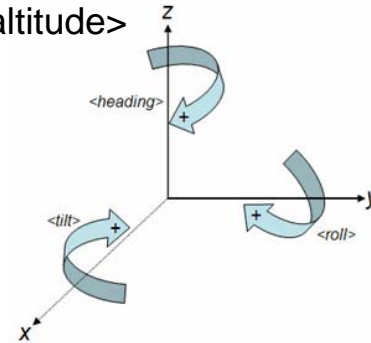
<heading>*</heading>***</heading>****<tilt>***</tilt>***</tilt>****<roll>***</roll>***</roll>**

</Orientation>

<Scale>

<x>*</x>***</x>****<y>***</y>***</y>****<z>***</z>***</z>**

</Scale>



2.2.4 Modellreferenz

<Link id="ID">

<href>*URL http oder lokale Datei*

*Parents: NetworkLiink *.kml/kmz*

*Model *.dae*

Overlay Image-File </href>

<refreshMode>*onChange, onInterval, onExpire</refreshMode>*

<refreshInterval>*default : 4 Sekunden</refreshInterval>*

<viewRefreshMode>*never, onStop, onRequest, onRegion</viewRefreshMode>*

<viewRefreshTime>*Wartezeit auf Refresh nach Kamerabewegung</viewRefreshTime>*

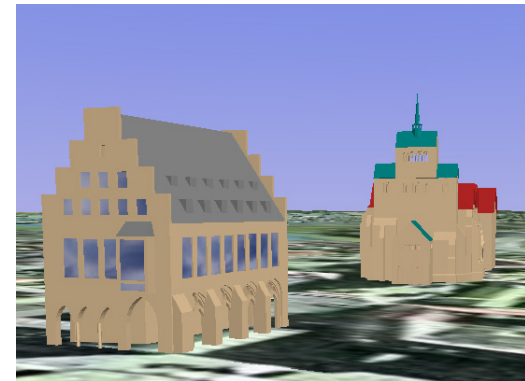
<viewBoundScale>*.. Uebermittlung an den Server </viewBoundScale>*

<viewFomat>*Anhang an Server Request</viewFormat>*

</Link>

</Model>

</Placemark>

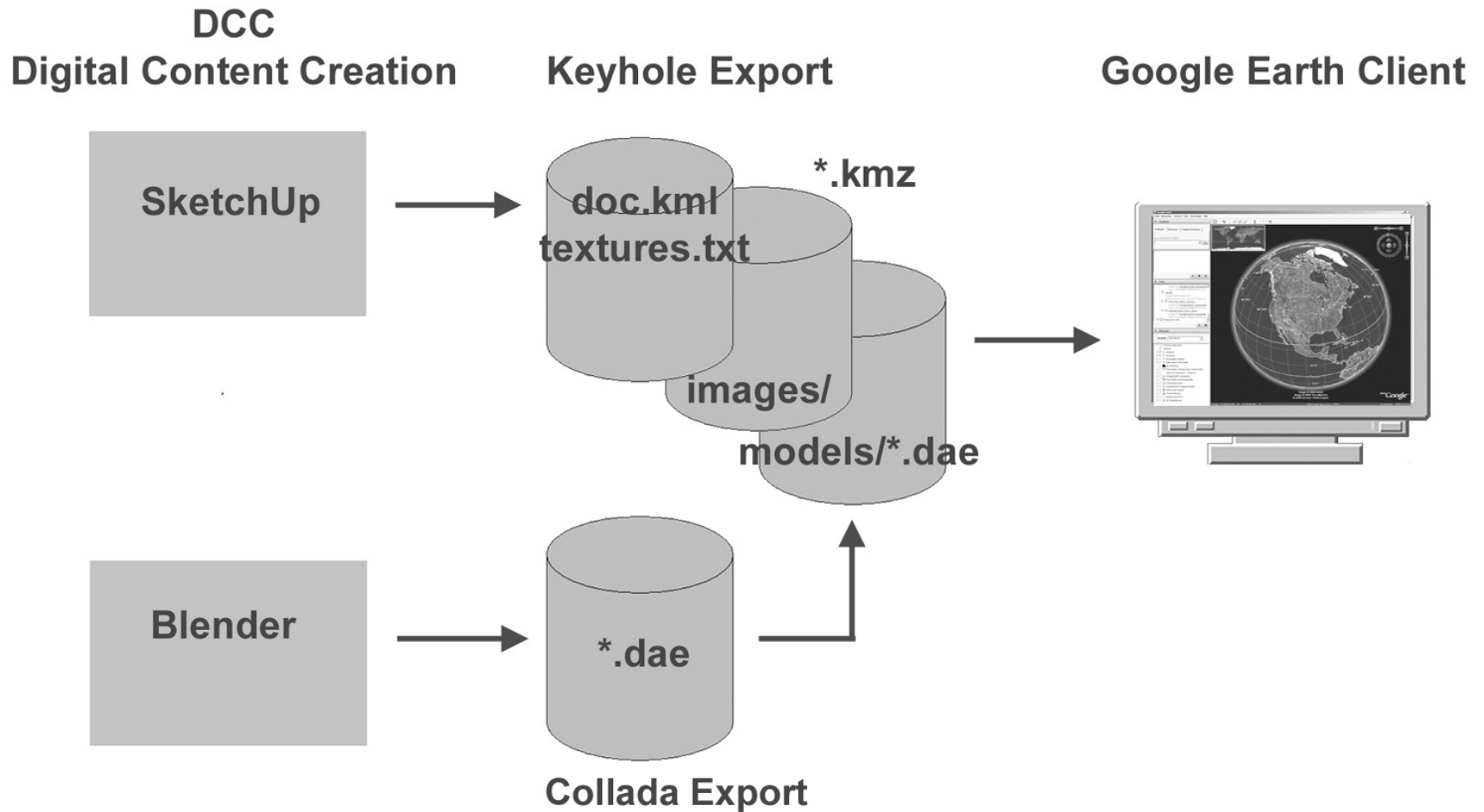


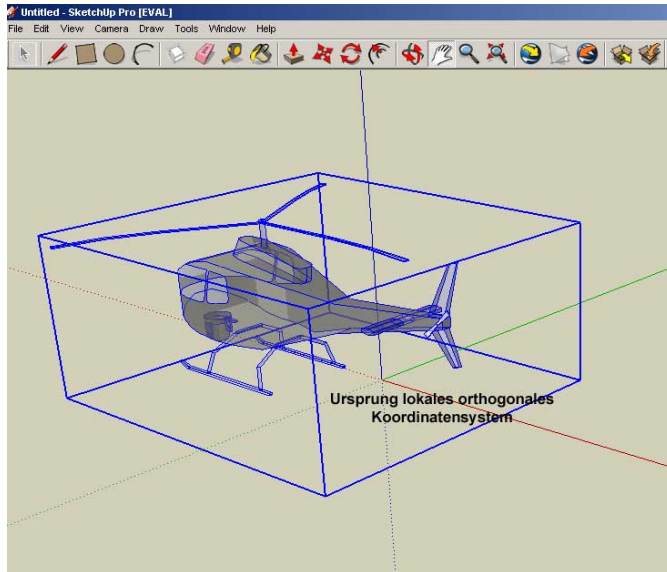
COLLADA (Collaborative Design Activity) ist ein XML-basiertes offenes Dateiformat zum Austausch von Daten zwischen modernen 3D-Autorenwerkzeugen und DCC (Digital Content Creation)-Programmen.

Ziel von Collada ist es , den Transport von Assets ohne Datenverlust zu unterstützen.

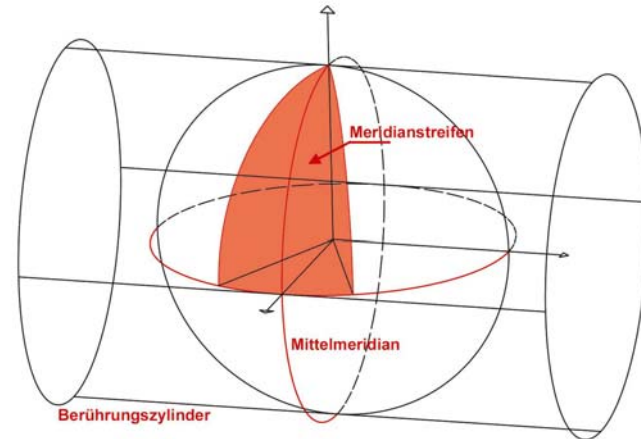
Google bietet zur Modellierung von 3D-Modellen das Programm **SketchUp** für den privaten Gebrauch kostenfrei an. Beim Export von SektchUP-Daten nach Google Earth wird eine Archiv-Datei geschrieben (**KMZ-Format**), in der **KML**-Informationen, Modelldaten (**DAE-Formate**) und Texturen enthalten sind.







1. **Lokales System nach GK**
Geodätisches Bezugssystem für Deutschland
Gauss-Krüger-Abbildung

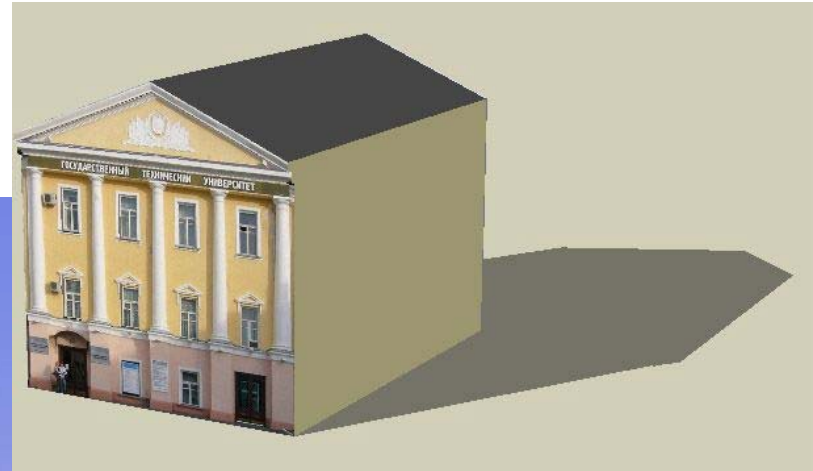


2. **Transformation in geografische Koordinaten (Bezugssystem Bessel)**
3. **Transformation in WGS84 (Postdam Datum)**

Berechnungsroutinen im Internet / grafische Lösung Google

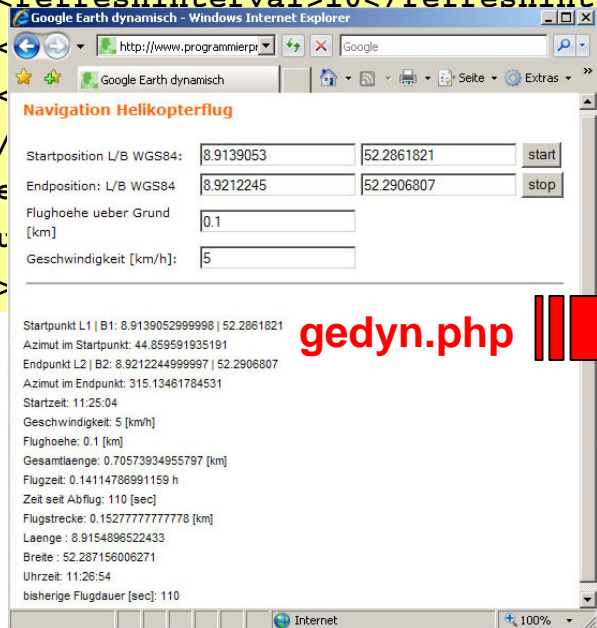


- Texturen
- Style
- Radio-Buttons



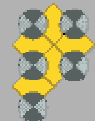
4.2 Google Earth dynamisch

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
<Document>
  <NetworkLink>
    <name>Hubschrauber dynamisch</name>
    <Url>
      <href>http://www.programmierpraktikum.de/gedyn/hubschrauber.kml</href>
      <refreshMode>onInterval</refreshMode>
      <refreshInterval>10</refreshInterval>
    </Url>
  </NetworkLink>
</Document>
</kml>
```



gedyn.php

```
<?xml version='1.0' encoding='UTF-8'?>
<kml xmlns='http://earth.google.com/kml/2.1'>
  <Placemark>
    <name>Hubschrauber</name>
    <Model>
      <altitudeMode>absolute</altitudeMode>
      <Location> ... </Location>
      <Orientation> ... Orientation</Orientation>
      <Scale> ... </Scale>
      <Link><href>models/helicopter.dae</href></Link>
    </Model>
  </Placemark>
</kml>
```



4.2 Dynamische Applikation: Backend Navigation

Google Earth dynamisch

Ein Hubschrauber fliegt auf der kürzesten Entfernung von P1(L1, B1) nach P2(L2, B2) mit einer Flughöhe h ueber Grund und einer konstanten Geschwindigkeit v . Die Dateneingabe erfolgt in dem Dokument http://www.programmierpraktikum.de/gedyn/ge_dynamisch.html. Aus diesem Dokument heraus wird kontinuierlich die Position des Hubschraubers nach Laenge und Breite bestimmt. Synchron wird eine KML-Datei mit der aktuellen Position geschrieben (<http://www.programmierpraktikum.de/gedyn/aircraft.kml>). In dieser KML-Datei wird der Hubschrauber referenziert (<http://www.programmierpraktikum.de/gedyn/helicopter.kml>), dessen geometrische Daten im COLLADA-Format in einer externen DAE-Datei vorliegen.

Sie muessen den Hubschrauber also zuenaechst anhalten, wird er sich auch ueber den Zielpunkt bewegen.

Lassen wir den Hubschrauber ueber ein 3D-Stadtmodell fliegen (http://www.x3dgrafik.de/stadtmodell_mindestdaten.html) ist 10 sec.

- [Google Earth installieren ?](#)
- [3D-Stadtmodell KMZ-Datei aufrufen](#)
- [Navigation starten](#)
- [KML-Datei Hubschrauber aufrufen](#)

Navigation Helikopterflug

Startposition L/B WGS84:	<input type="text" value="8.9139053"/>	<input type="text" value="52.2861821"/>	<input type="button" value="start"/>
Endposition: L/B WGS84	<input type="text" value="8.9212245"/>	<input type="text" value="52.2906807"/>	<input type="button" value="stop"/>
Flughoehe ueber Grund [km]	<input type="text" value="0.1"/>		
Geschwindigkeit [km/h]:	<input type="text" value="5"/>		

Aktuelle Position: Startpunkt L1 | B1: 8.9139052999998 | 52.2861821
Azimut im Startpunkt: 44.859591935191
Endpunkt L2 | B2: 8.9212244999997 | 52.2906807
Azimut im Endpunkt: 315.13461784531
Startzeit: 11:25:04
Geschwindigkeit: 5 [km/h]
Flughoehe: 0.1 [km]
Gesamtlaenge: 0.70573934955797 [km]
Flugzeit: 0.141147868991159 h
Zeit seit Abflug: 110 [sec]
Flugstrecke: 0.15277777777778 [km]
Laenge: 8.9154896522433
Breite: 52.287156006271
Uhrzeit: 11:26:54
bisherige Flugdauer [sec]: 110

Zeiger Lat 52|288107° Lon 8|914558° Höhe 58 m Übertragung 100% Sichthöhe 136 m



Günter Pomaska
Web-Visualisierung mit Open Source
Vom CAD-Modell zur Real-Time Animation
Wichmann-Verlag, Heidelberg, 2007

www.x3dgrafik.de
Dynamische Applikation:
www.programmierpraktikum.de/gedyn

David A. Crowder
Google Earth for Dummies
Wiley Publishing, 2007

Martin C. Brown
Hacking Google Maps and Google Earth
Wiley Publishing, 2006

Remi Arnau, Mark BarnesC.
Collada. Sailing the Gulf of 3D Digital Content Creation
Transatlantic Publishers, 2006

