

Bedienungsanleitung

CityGML-Toolchain Editor 1.6

27.11.2011

CityGML-Toolchain im Web:

www.citygml-toolchain.de oder www.citygml.de

Inhalt

1	Allgemeine Informationen	3
1.1	Funktionsumfang	3
1.2	Problem „Planarität“	3
1.3	Problem fehlerhafte SketchUp Modellierung	4
2	Systemanforderungen	5
3	Installation	6
4	Importieren	8
5	Exportieren	12
5.1	Reiter Layer/LOD/textures	13
5.2	Reiter Coordinate system	14
5.3	Reiter CityGML options	15
6	CityGML Attribute	18
6.1	Attribute für Gebäude und Gebäudeflächen	18
6.2	Attribut Surfacetyp	21
6.3	Attribute Kopieren / Einfügen	22
7	Modellierungshinweise	23
7.1	Elemente richtig gruppieren	23
7.2	Selbstschneidenden Polygone vermeiden	23
7.3	Datenhaltung möglichst im SKP-Format	23
7.4	Kein Material auf gruppierte Flächen	23
7.5	Material nur auf der Vorderseite	24
8	Rückmeldungen	27
9	Auf Update prüfen	28

1 Allgemeine Informationen

Der Editor der CityGML-Toolchain ist kein eigenständiges Programm (mit Ausnahme des ab Version 1.6 hinzugekommenen neuen Importers), sondern ein CityGML-Plugin für Google SketchUp. Alle Dialoge dieses Plugins sind in Englisch realisiert. Da z.T. auf Windows-Standarddialoge zurückgegriffen wird, erscheinen auf einem Rechner mit deutschem Windows-Betriebssystem teilweise gemischtsprachige Dialoge.

1.1 Funktionsumfang

Mit dem CityGML-Plugin wird Google SketchUp um folgende Funktionen erweitert:

Importieren von CityGML-Modellen

Importiert werden einfache CityGML-Modelle, die aus *Building* und *ReliefFeature* Elementen bestehen. Attribute und IDs von *Polygon*, *LinearRing* und *Building* bleiben erhalten. Der Import erfolgt durch den CityGML-Toolchain SkpWriter, der eine SKP-Datei erstellt.

Exportieren von CityGML-Modellen

Die exportierten Modelle bestehen aus den CityGML Elementen *Building* mit *GroundSurface*, *RoofSurface* und *WallSurface*. Die eigentlichen Geometrien werden in GML-Elementen vom Typ *MultiSurface* gespeichert.

Bearbeiten von CityGML-Attributen

Anzeigen/Bearbeiten/Hinzufügen/Löschen von Standardattributen für Gebäude und Flächen sowie von generischen Attributen (nur *StringAttribute*) für Gebäude. Bei Flächen kann der SurfaceType festgelegt werden werden.

1.2 Problem „Planarität“

Beim Importieren von CityGML-Modellen kann es vorkommen, dass einzelne Wände (insbesondere Dächer) automatisch trianguliert werden. Dies liegt daran, dass die Toleranz bzgl. der Planarität von Flächen in SketchUp sehr gering ist. Uns liegen CityGML-Modelle vor, bei denen bis zu 5% aller Flächen von SketchUp als „unzureichend planar“ verworfen werden.

Das Problem wird noch verschärft, wenn das eingelesene Modell hochskaliert wird. Letzteres wäre für CityGML-Modelle eigentlich notwendig, da die Längeneinheit in CityGML Meter ist, SketchUp intern aber in Zoll arbeitet. Nach der Umrechnung importierter CityGML-Modelle in Zoll kann die Anzahl „nicht planarer“ Flächen aber deutlich zunehmen.

Aus diesem Grund wird beim Import auf die m → inch Umrechnung verzichtet, so dass in SketchUp alle Gebäude um den Faktor 39,37 zu klein sind. Da der Umrechnungsfaktor aber beim Export ebenfalls ignoriert wird, ist die Skalierung am Ende wieder korrekt. Als Workaround für die Arbeit in SketchUp wird empfohlen, alle Maße in inch anzeigen zu lassen und die Zahlen aber als Meter-Angaben zu interpretieren.

Im Unterschied dazu werden SketchUp-Modelle, die nicht über den CityGML-Importer gekommen sind, beim Exportieren in Meter umskaliert, so dass exportierte Modell immer richtig skaliert sein sollten, egal aus welcher Quelle sie stammen. Nur "mischen" kann man Modelle aus unterschiedlichen Quellen nicht.

1.3 Problem fehlerhafte SketchUp Modellierung

Um beim Export korrekte CityGML-Modelle zu bekommen, sind bei der Modellerstellung in SketchUp eine Reihe von Dingen zu beachten. Nicht alles, was in SketchUp auf den ersten Blick fehlerfrei aussieht, ist auch ein "gutes" Modell. Und beim CityGML-Export kann aus einem "nicht guten" Modell schnell ein semantisch fehlerhaftes Modell werden.

Häufigste Fehler sind falsche orientierte Flächen an Gebäuden, das sind Flächen mit texturierter Rückseite, die nach außen zeigen. Dies ist bereits als SketchUp Modell als fehlerhaft anzusehen, denn die besagte Fläche würde bei gerichtetem Licht nicht korrekt beleuchtet.

Hinsichtlich der korrekten Modellierung sei auf das Kapitel 7 verwiesen.

2 Systemanforderungen

Das Plugin wurde für Google SketchUp 7/8 und Microsoft Windows (XP, Vista, Windows 7) entwickelt. Die aktuelle SketchUp-Version finden Sie unter:

<http://SketchUp.google.com/intl/en/download>.

Sie können sowohl mit der kostenlosen Grundversion als auch mit der kostenpflichtigen Pro Version arbeiten. Ältere Versionen von SketchUp werden nicht unterstützt.

3 Installation

Für die Installation muss der Inhalt des heruntergeladenen Archivs (etwa 20Mbyte) entpackt und anschließend mit dem Windows Explorer in das Plugin-Verzeichnis von SketchUp kopiert werden. Der bei der Installation von SketchUp voreingestellte Standardpfad für Plugins ist

```
C:\Programme\Google\Google SketchUp 8\Plugins
```

bzw.

```
C:\Program Files\Google\Google SketchUp 8\Plugins
```

Wichtiger Hinweis für Nutzer von Packer-Programmen, Vista und/oder Windows 7: Nie direkt in das SketchUp Plugin-Verzeichnis entpacken, da Programme keine Dateien in das Programmverzeichnis schreiben dürfen und, falls sie es dennoch versuchen, "unbemerkt" in das "Kompatibilitätsverzeichnis" umgeleitet werden.

Sollte das versehentlich passiert sein, findet man im Windows Explorer einen Hinweis auf die Existenz von "Kompatibilitätsdateien", Abb. 1. Diese müssen unbedingt gelöscht werden:

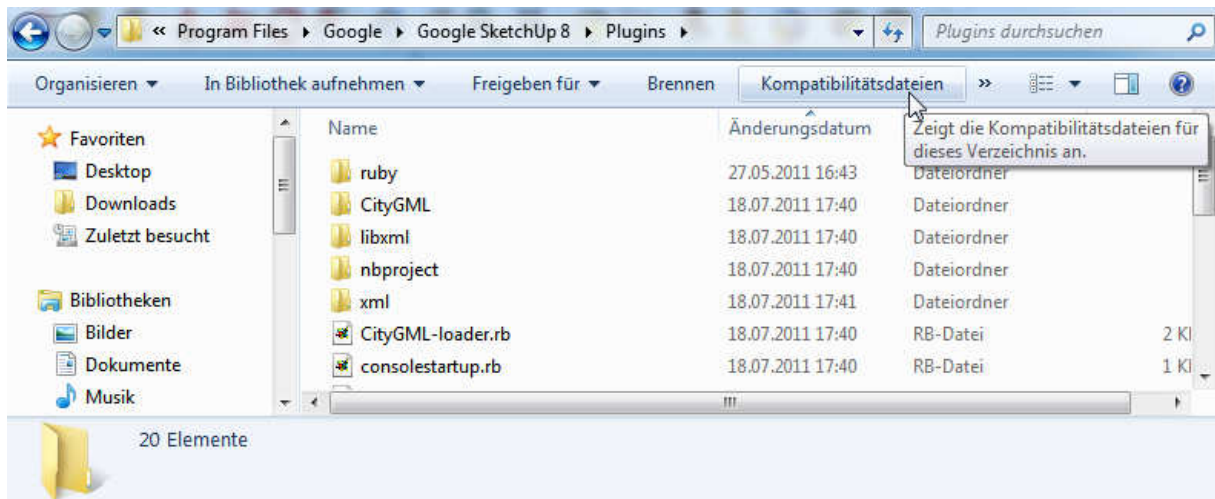


Abb. 1: SketchUp Plugin-Ordner mit "Kompatibilitätsdateien"

Beim nächsten SketchUp Start ist das Plugin verfügbar. Es kann über **Window→Preferences→Extensions** aktiviert/deaktiviert werden:

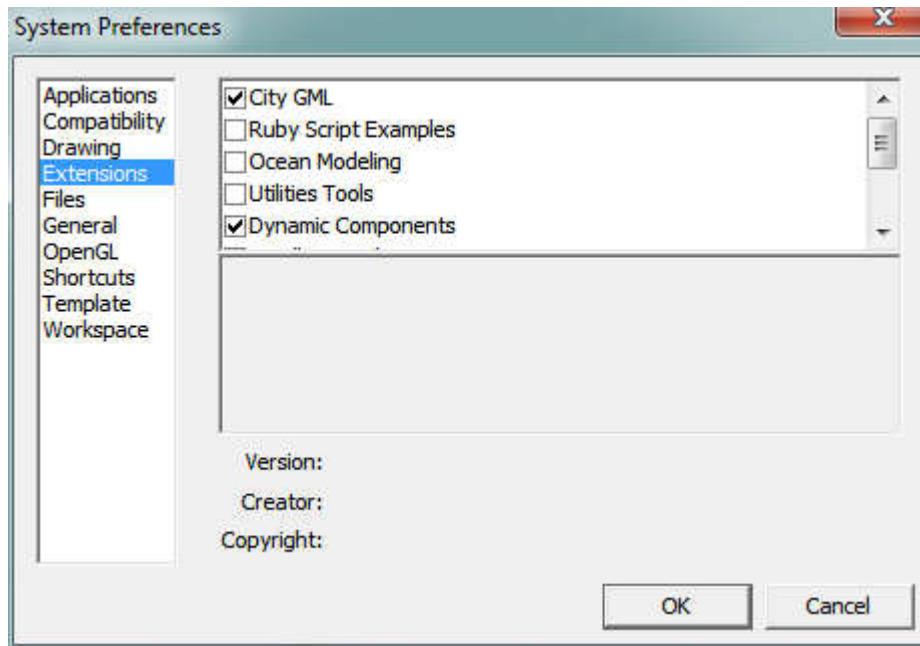


Abb. 2: SketchUp Plugin aktivieren.

4 Importieren

Der Import erfolgt über ein eigenständige Programm SkpWriter, das aus dem Plugin heraus über **Plugins** → **CityGML** → **Import** gestartet wird.

Der SkpWriter meldet sich mit folgendem Fenster:

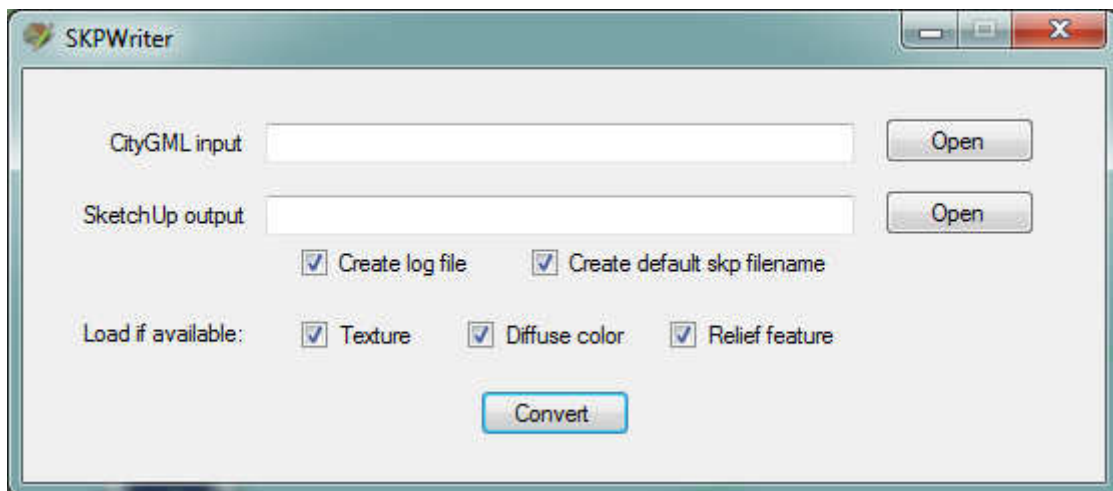


Abb. 3: SkpWriter Startfenster.

Neben den beiden Eingabefeldern für die Dateinamen von CityGML und SKP-Datei (voller Pfad) existieren folgende Optionen:

- | | |
|-----------------------------|---|
| Create log file | Es wird eine Log Datei im Zielverzeichnis der SKP-Datei erzeugt, in der wichtige Kennzahlen (z.B. Anzahl der erzeugten Flächen) sowie sämtliche bei der Konvertierung aufgetretenen Fehler eingetragen sind. Der Name dieser Logdatei ist:

[Name der Skp Datei]_log_[Tag].[Monat].[Jahr]_[Stunde]-[Minute].txt |
| Create default skp filename | Der SketchUp output Dateiname wird nach folgende Schema automatisch aus dem input Dateinamen erzeugt:

<i>PfadMeineStadt.xml</i> → <i>PfadMeineStadt.skp</i> |
| Texture | Es werden Texturen importiert, falls im Modell vorhanden. |
| Diffuse Color | Es wird Diffuse Color importiert, falls im Modell vorhanden. |
| Relief feature | Es wird Gelände (CityGML Element <i>ReliefFeature</i>) importiert, falls im Modell vorhanden. |

Nach Start der Konvertierung durch Drücken auf **Convert** ändert sich die Anzeige:

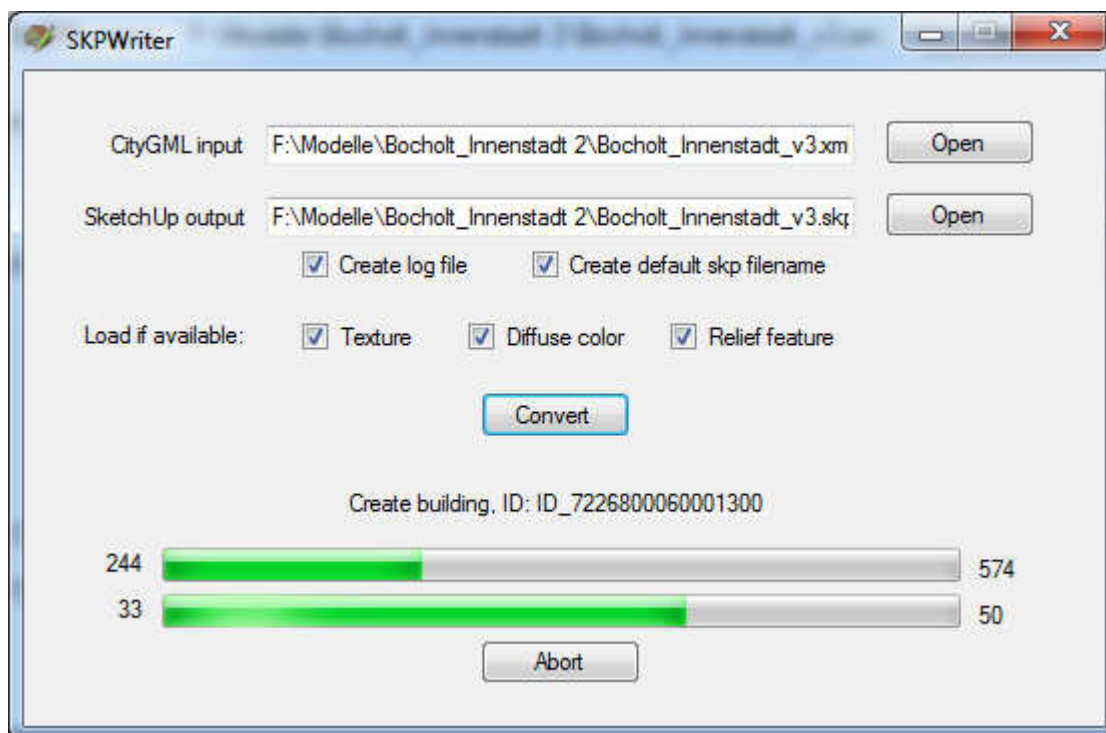


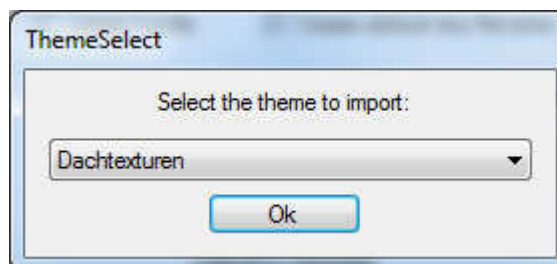
Abb. 4: SkpWriter Anzeigefenster während des Import Vorgangs

Der Text über den beiden Fortschrittsbalken gibt den aktuellen Bearbeitungsschritt an.

Der obere Fortschrittsbalken stellt den Bearbeitungsfortschritt im aktuellen Gebäude dar. Links bzw. rechts von diesem Balken wird Zahl der bereits importierten bzw die Gesamtzahl der Gebäudeflächen angezeigt.

Der untere Fortschrittsbalken stellt sinngemäß das Gleiche für die gesamte CityGML-Datei dar, gezählt werden hier Gebäude. Aufgrund programminterner Strukturen muss der Gesamtimport in Pakete zu 50 Gebäuden unterteilt werden. Der untere Fortschrittsbalken zeigt (leider) nur den Fortschritt im aktuellen Paket an.

Sofern das zu importierende CityGML Modell mehr als ein *Theme* enthält, müssen Sie über folgenden Dialog, der nach dem Start der Konvertierung erscheint, eines dieser Themes auswählen:



Nach erfolgreicher Konvertierung erscheint ein abschließendes Info-Fenster:

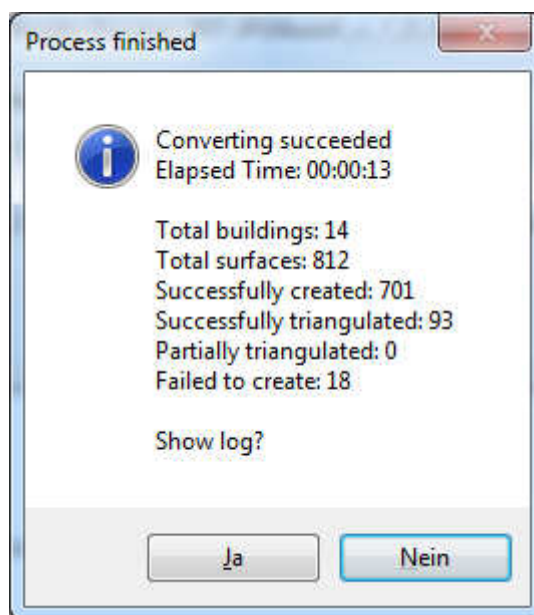


Abb. 5: SkpWriter Infofenster nach beendetem Importvorgang.

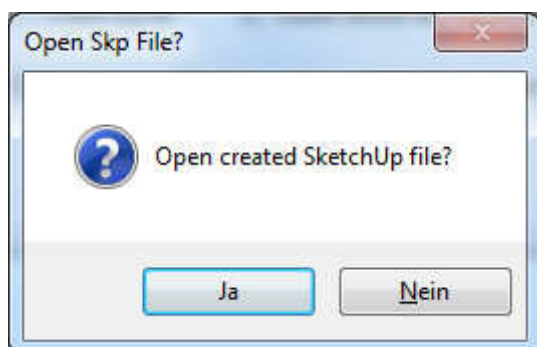
Dieses zeigt folgende Kennzahlen:

Total buildings / surfaces	Anzahl der gelesenen Gebäude und Flächen
Successfully created	Anzahl der Flächen, die ohne Triangulierung erstellt werden konnten
Successfully triangulated	Anzahl der Flächen, die aufgrund nicht ausreichender Planarität nur durch Triangulieren erstellt werden konnten. Welche Flächen nicht "hinreichend planar" sind, entscheidet SketchUp nach einem uns nicht bekannten Kriterium.
Partially triangulated	Anzahl der Flächen, die trotz Triangulierung nur zum Teil nachgebildet werden konnten.
Failed to create	Anzahl von der Flächen, die garnicht (auch nicht im Sinne von "Partially triangulated") erstellt werden konnten.

Sofern vor Beginn der Konvertierung auch "Create log file" mit ausgewählt wurde, kann dieser nach Abschluss der Konvertierung geöffnet werden. Er befindet sich im Zielverzeichnis der SketchUp Datei.

In der Log Datei finden Sie neben den Kennzahlen weitere Details, wie z.B. die Angabe, welche Optionen für die Konvertierung ausgewählt wurden, welche Flächen trianguliert wurden oder ggf. garnicht erstellt werden konnten und welche Texturen nicht gefunden wurden.

Nach erfolgreicher Konvertierung kann die erstellte SKP-Datei direkt geöffnet werden:

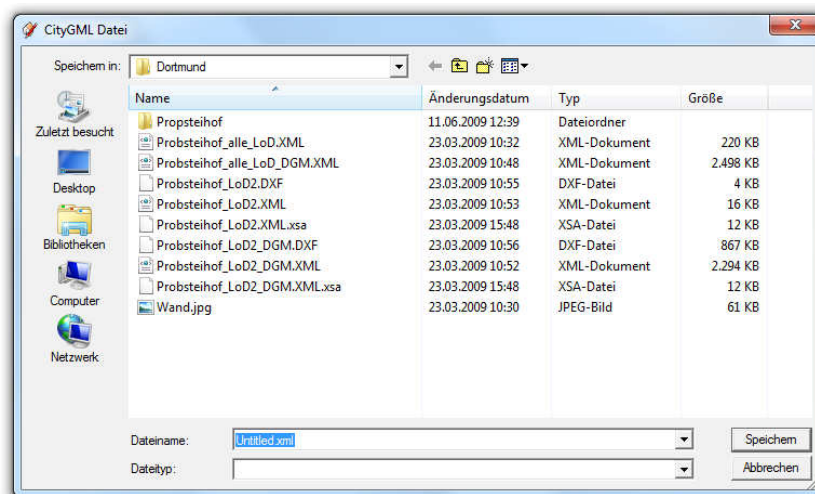


5 Exportieren

Beim Exportieren wird jede Gruppe in SketchUp als *Building* gespeichert. Sollten innerhalb der Gruppe weitere Gruppe vorkommen, werden diese bislang ignoriert, d.h. alle Flächen von Untergruppen werden dem *Building* zugerechnet, das der obersten Gruppe entspricht.

Beim Exportieren werden alle Flächen als *Lod1MultiSurface* .. *Lod4MultiSurface* geschrieben, sofern kein SurfaceType festgelegt ist. Sollte bei einer Fläche der SurfaceType gesetzt sein, wird die Fläche innerhalb von *boundedBy* und dem entsprechenden SurfaceType exportiert. Flächen, die aufgrund fehlender Gruppenzugehörigkeit keinem Building zugeordnet sind, werden einem automatisch generierten „Sammel“-*Building* zugeordnet.

Um das aktuell geöffnete SketchUp-Modell nach CityGML zu exportieren, wählen Sie aus dem Menü **Plugins** → **CityGML** → **Export** aus. Es erscheint folgender Dialog:



Wichtig: Der Dateiname (incl. Pfad) für die exportierte Datei darf keine deutschen Sonderzeichen wie ü oder ß enthalten!/
/

Nach Klicken auf **Speichern/Save** erscheint ein Dialogfenster mit drei Reitern, in dem das erzeugte CityGML-Modell durch eine Vielzahl von Parametern beeinflusst werden kann:

5.1 Reiter Layer/LOD/textures

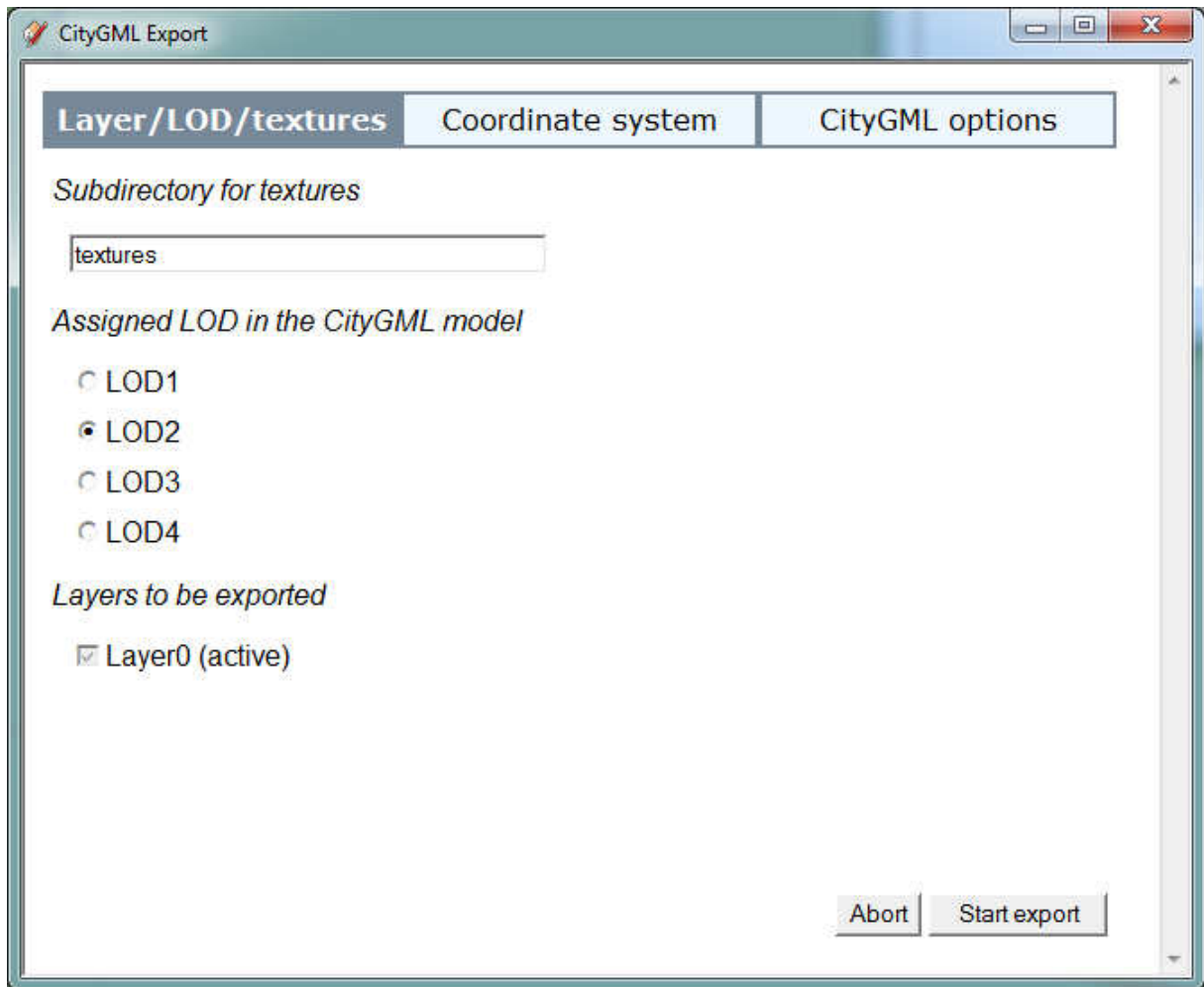


Abb. 6: CityGML Export Dialog: Reiter Layer/LOD/textures

Subdirectory for textures gibt den Unterordner für eventuell zu exportierende Texturen an. Bleibt dieses Eingabefeld leer, so werden die Texturen in dasselbe Verzeichnis exportiert, in dem auch die CityGML Datei erstellt wird.

Unter **Assigned LOD in the CityGML model** legen Sie den LOD fest, dem alle Flächen des Modells zugeordnet werden sollen. Eine individuelle Zuordnung einzelner Flächen ist nicht möglich.

Unter **Layers to be exported** werden alle Layer markiert, die exportiert werden sollen. Der aktuelle Layer kann nicht deaktiviert werden.

5.2 Reiter Coordinate system

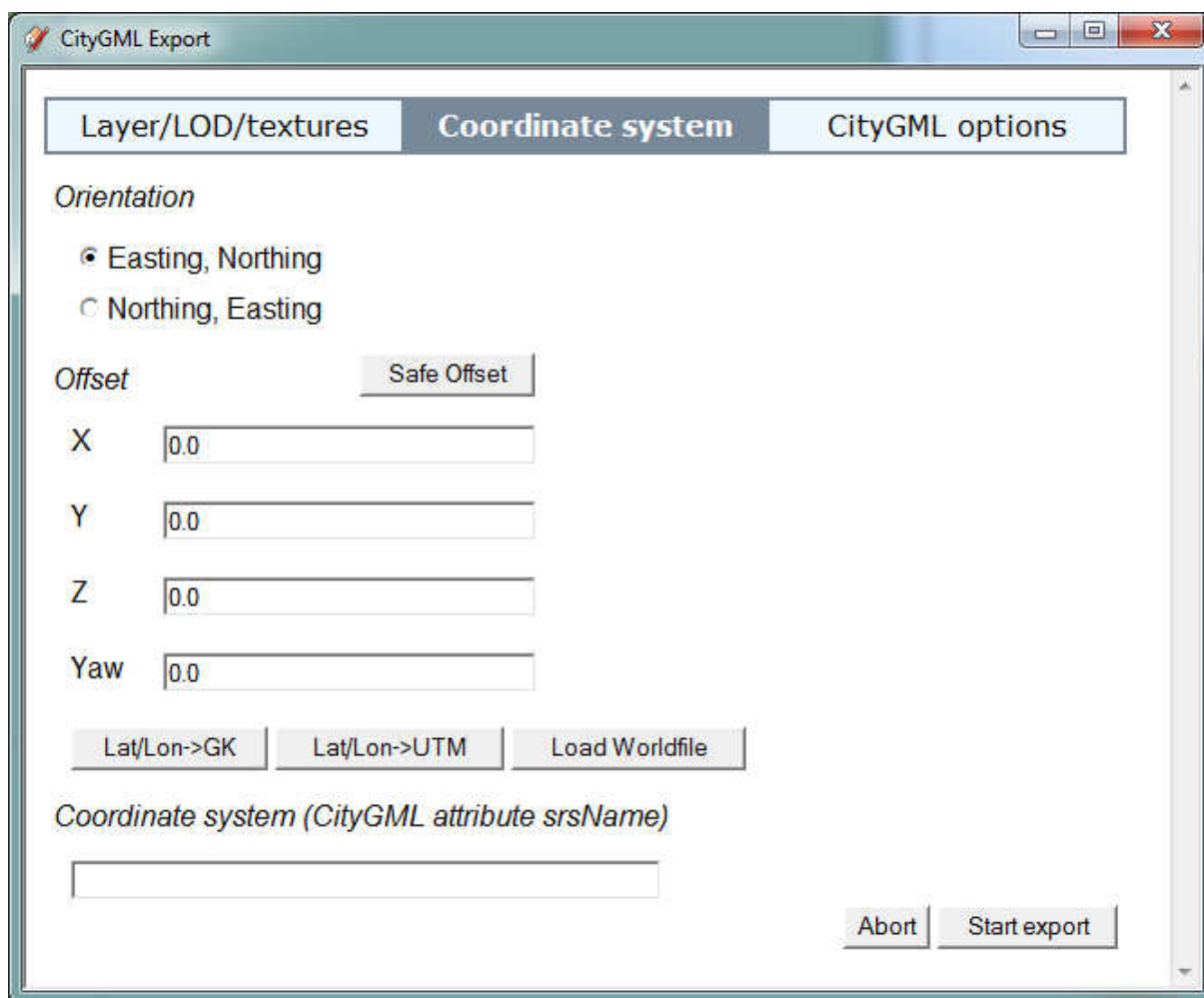


Abb. 7: CityGML Export Dialog: Reiter Layer/LOD/textures

Orientation legt die Reihenfolge von Rechtswert (Easting) und Hochwert (Northing) fest.

Mit der Option **Offset** kann das Modell in X, Y und Z-Richtung verschoben werden. Zusätzlich kann unter Yaw ein Winkel abgegeben werden, um den das Modell um die z-Achse gedreht werden kann (in SketchUp ist dies unter den Standardeinstellungen die blaue Achse). Diese Werte können durch den Benutzer verändert werden. Zusätzlich können sie aber auch über **Load Worldfile** aus einer DXT-Datei ausgelesen werden. Dies ist eine Textdatei mit je einer Zeile für X, Y, Z und Yaw.

Außerdem kann über den **Lat/Lon → GK** bzw **Lat/Lon → UTM** Button auch die Geo-Location, die in der Modell Info in SketchUp hinterlegt wurde, in das Gauss Krüger bzw UTM Koordinatensystem transformiert und als Modell-Offset für X und Y gesetzt werden.

Alle Einträge/Änderungen, die beim Offset (X, Y, Z, Yaw) vorgenommen werden, gelten zunächst nur für den aktuellen Exportvorgang. Sollen die geänderten Offset-Daten auch in das SketchUp-Modell übernommen werden, so muss der Button **Safe Offset** gedrückt werden.

Über **Coordinate system** kann man das zu benutzende Koordinatensystem in die CityGML Datei als Attribut des GML-Elements <gml:MultiSurface> eintragen. Dazu ein Beispiel:

Wenn das Feld leer, ist ändert sich am Element nichts. Ist in dem Feld etwas eingetragen, so erstellt der Exporter das Element <gml:MultiSurface srsName="*Ihre Eingabe*" srsDimension="3">.

5.3 Reiter CityGML options

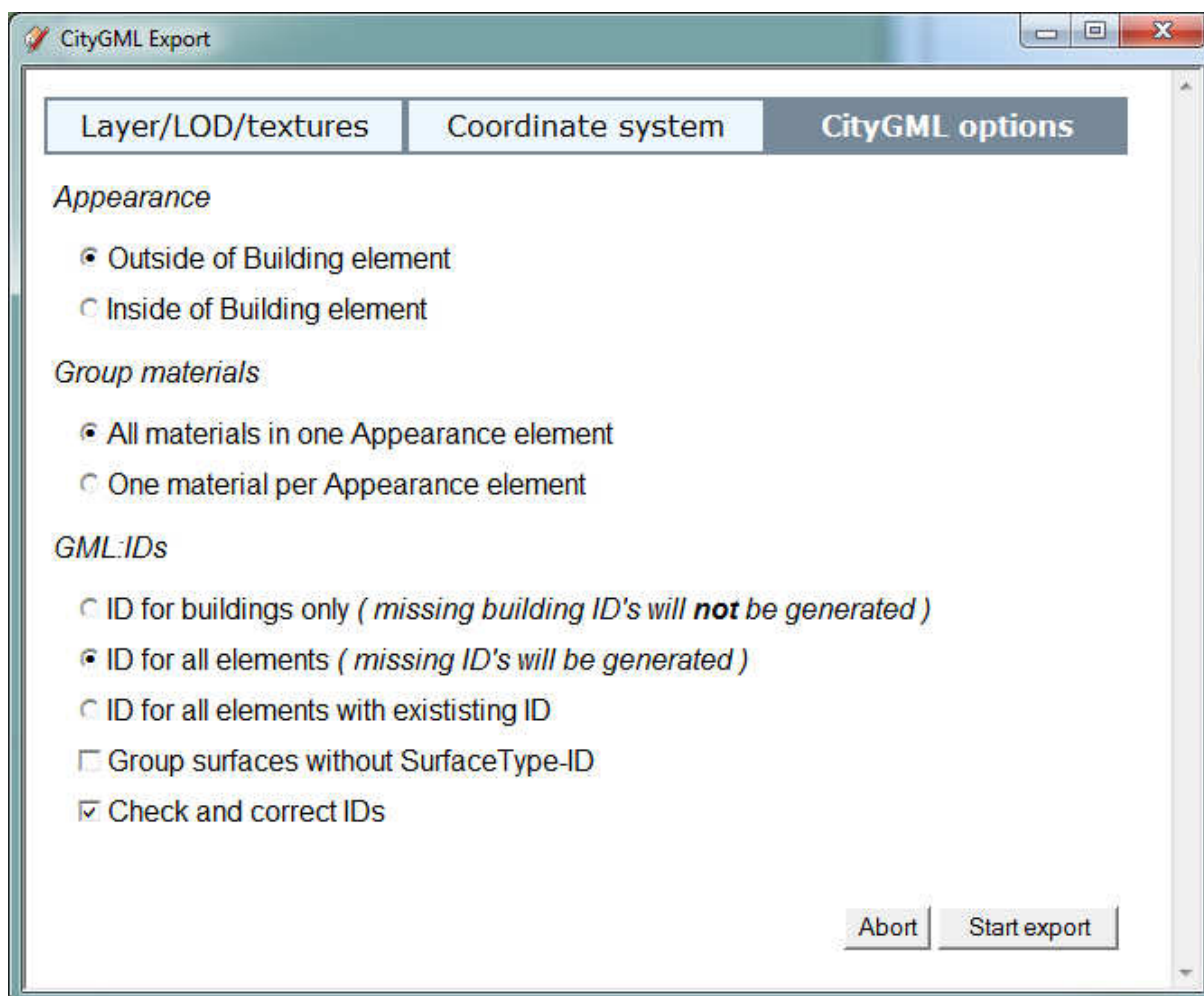


Abb. 8: CityGML Export Dialog: Reiter CityGML options

Dieser Reiter bietet allgemeine Modellierungs Optionen an.

Über die Option **Appearance** wird festgelegt, ob Appearance-Elemente Unterelemente des zugeordneten Buildings (**Inside of Building element**) sein sollen, oder ob sie hinter allen Buildings (**Outside of Building element** → Standard) erscheinen.

Über die Option **Group materials** wird festgelegt, wie Materialien und Texturen auf Appearance-Elemente abgebildet werden. **All materials in one Appearance element** bedeutet, dass alle Materialien / Texturen (als surfaceDataMember) des zugeordneten Buildings (**Appearance** → **Inside of Building element**) oder des gesamten Modells (**Appearance** → **Outside of Building element**) innerhalb eines einzigen Appearance-Elements geschrieben werden (Standard). Bei **One material per Appearance element**

bekommt jedes Material / jede Textur (d.h. jedes surfaceDataMember-Element) sein eigenes Appearance-Element.

Die Option **GML:ID** beeinflusst das Schreiben von IDs für die Elemente Building, Polygon, LinearRing, WallSurface, GroundSurface und RoofSurface. Hierzu gibt es drei Möglichkeiten:

Bei **ID for buildings only** werden IDs nur bei Buildings geschrieben. Für ein Building ohne ID wird beim Schreiben keine ID generiert. Alle übrigen Elemente werden ohne ID exportiert, auch wenn IDs vorhanden sind.

Bei **ID for all elements** werden für alle Elemente IDs geschrieben. Ist keine ID vorhanden wird eine eindeutige ID generiert.

Bei **ID for all elements with exististing ID** werden nur für die Elemente IDs geschrieben, für die eine ID existiert. Für Elemente ohne ID wird auch keine ID generiert.

Für alle drei Varianten gilt zusätzlich folgende Einschränkung: Sofern ein Face in SketchUp (repräsentiert in CityGML ein SurfaceType-Element mit all seinen Unterelemente wie z.B. Polygon) ein Material und/oder eine Textur besitzt, wird unabhängig vom gewählten Modus für das daraus generierte Appearance-Element immer eine ID geschrieben (wenn keine ID existiert wird sie automatisch erzeugt), da diese in dem CityGML-Modell für die Referenzierung benötigt wird.

Die Option **Group surfaces without SurfaceType-ID** bewirkt, dass alle Faces, die den gleichen SurfaceType (z.B. *RoofSurface*) aber keine BoundarySurfaceTypeld (das würde beim Export die ID des zu erzeugenden SurfaceType-Elements) haben, zusammen innerhalb eines einzigen SurfaceType-Elements geschrieben werden. Ist diese Option deaktiviert, wird jedes Face in Form eines eigenen SurfaceType-Elements exportiert. SurfaceType Elemente sind *RoofSurface*, *Wallsurface* oder *RoofSurface*.

Ist **Check and correct IDs** aktiviert, werden alle IDs auf Eindeutigkeit überprüft. Tritt beim Schreiben eine bereits einmal geschriebene ID erneut auf, dann wird eine neue, eindeutige ID nach dem Muster *AlteID_Laufindex* generiert. Ist diese Option nicht aktiviert, führt eine doppelte ID zu einer Fehlermeldung und der Exportvorgang wird abgebrochen.

Automatisch generierte IDs werden wie folgt generiert:

Building: „_[Filebasename]_BD.[BD Laufindex]“

Polygon: „_[Building_id]_PG.[PG Laufindex]“

LinearRing: „_[Polygon_id]_LR.[LR Laufindex]“

Beispiele:

- automatisch generierte ID eines Linear Rings in File house.xml, bei der auch die darüberliegenden Elemente keine ID besitzen: `_house_BD.3_PG.5_LR.2.`
- wie oben, aber das Gebäude hat die bereits die ID 4711: `4711_PG.5_LR.2.`

Durch Klicken auf **Start export** starten Sie den Exportvorgang. Es öffnet sich ein Konsolenfenster, in dem Statusinformationen beim Exportieren angezeigt werden:

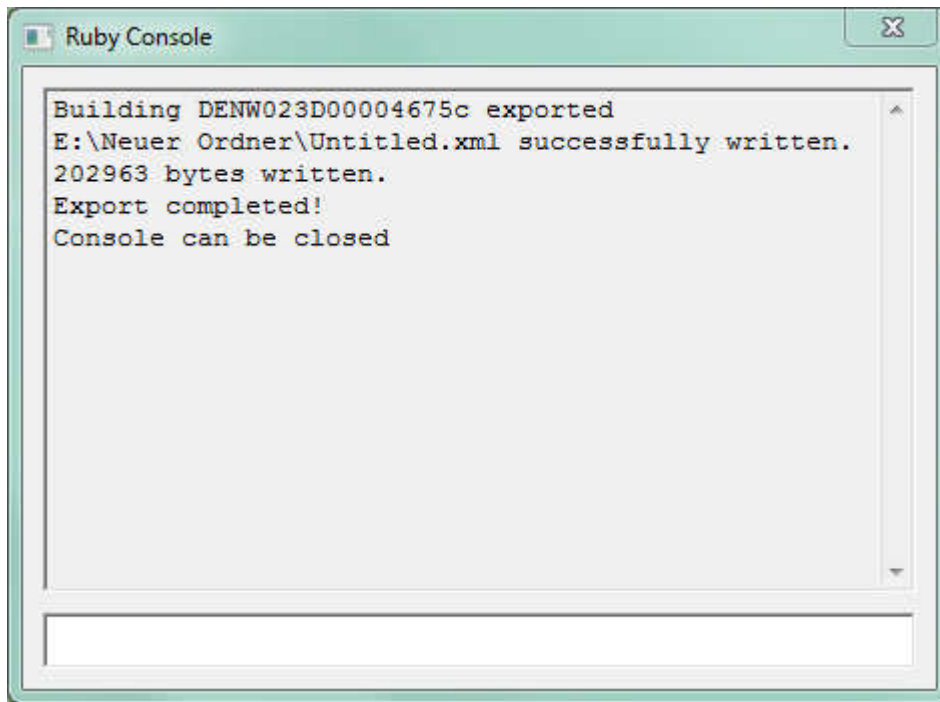


Abb. 9: (Ruby-)Konsolenfenster mit Statusinformationen beim CityGML Export

6 CityGML Attribute

6.1 Attribute für Gebäude und Gebäudeflächen

Mit dem CityGML SketchUp-Plugin ist es möglich, Attribute für Gebäude (*Building*) und Gebäudeflächen (*MultiSurface*) zu bearbeiten. Um die Attribute eines Gebäudes zu bearbeiten, klicken Sie mit der rechten Maustaste auf ein Element der Gruppe und wählen **Attribute** → **Edit** (Abb. 10).

Für das Bearbeiten von Attributen einer einzelnen Gebäudefläche aus einer Gruppe (d.h. einem *Building*) klicken Sie als erstes mit der rechten Maustaste auf die Fläche und wählen **Edit Group**. Anschließend klicken Sie erneut mit der rechten Maustaste auf diese Fläche und wählen **Attribute** → **Edit** (Abb. 10).

Es erscheint ein Dialog, in dem alle vorhandenen Attribute angezeigt werden (Abb. 11). Die Attribute werden in die Kategorie Standard Attribute (CityGML: *StandardAttribute*) und Generische Attribute (CityGML: *StringAttribute*) eingeteilt, die über den entsprechenden Reiter ausgewählt werden können. Nun können Sie Attribute ändern, hinzufügen oder löschen.

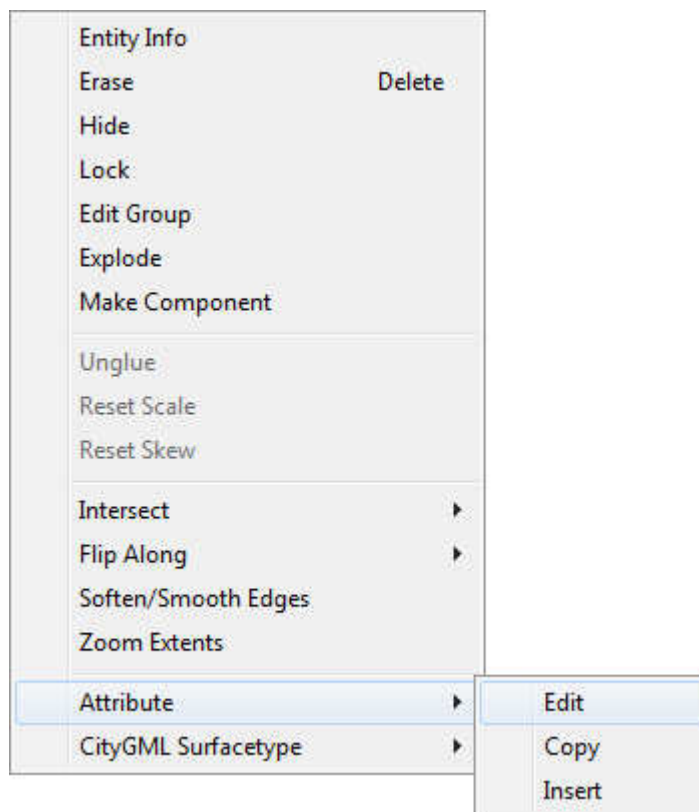


Abb. 10: Kontextmenü für Attribute

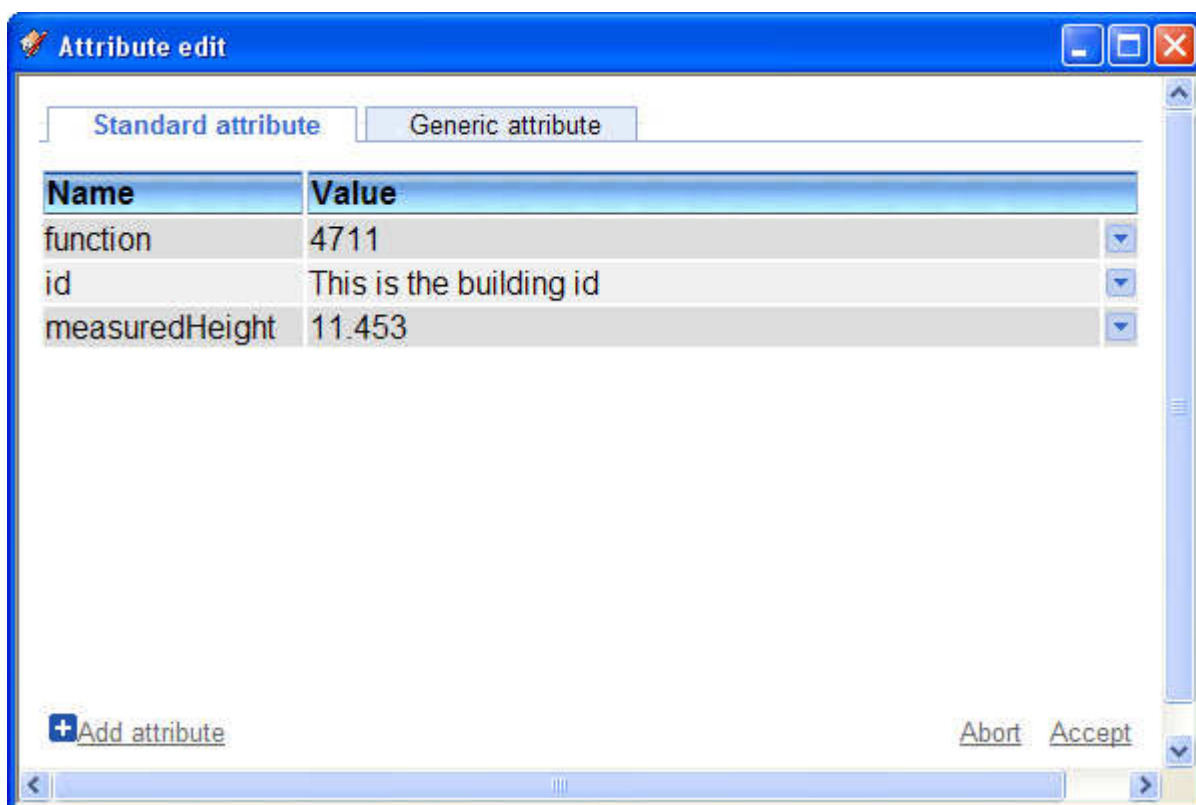



Abb. 11: Attribute bearbeiten (hier für ein Gebäude)

Alle in der Dialogmaske vorgenommenen Änderungen können rückgängig gemacht werden, indem der Dialog über  geschlossen oder auf **Abort** geklickt wird. Durch Klicken auf **Accept** werden alle Veränderungen gespeichert.

Ändern

Nach Doppelklick auf ein Attribut erscheint ein Fenster, in dem Sie den Wert des Attributs ändern können. Die Änderungen können entweder gespeichert oder verworfen werden. Sie sollten darauf achten, dass einige Attribute, wie z.B. die ID, nur mit Vorsicht zu verändern sind.


Hinzufügen

Wollen Sie ein neues Attribut erzeugen, wählen Sie im Fenster links unten **+ Add attribute** aus. Es erscheint ein kleines Fenster (Abb. 12, Abb. 13), in dem Sie den Namen und den Wert festlegen können.

Beim Hinzufügen eines „Standard attribute“ können Sie den Namen aus einer Liste von Attributen auswählen, die im CityGML-Standard festgelegt sind, Abb. 12. Als „Generic attribute“ sind nur String Attribute möglich. Deren Name kann, bis auf Namen von Standardattributen (unabhängig von Groß- und Kleinschreibung) frei gewählt werden. Abb. 13. Generische Attribute existieren ausschließlich für Gebäude nicht aber für Flächen.

Das bei Flächen (Faces) angebotene Attribut PolygonID wird beim Export dem GML-Element *Polygon*, das Attribut LinearRingID dem GML-Element *LinearRing* zugeordnet.

Löschen

Wollen Sie ein Attribut löschen, klicken Sie auf den Pfeil  rechts neben dem Attribut und wählen **Delete** aus.

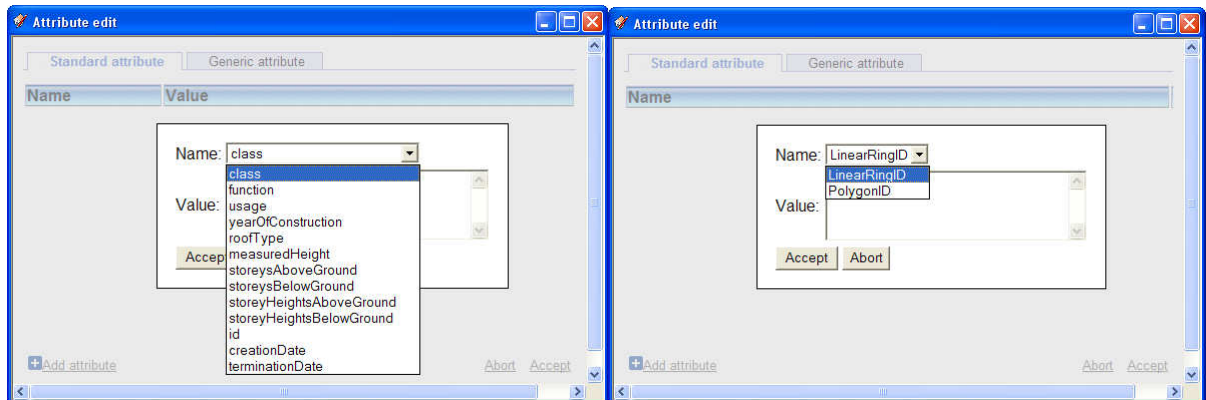


Abb. 12: Standard Attribut hinzufügen für Gebäude (links) bzw. Flächen (rechts)

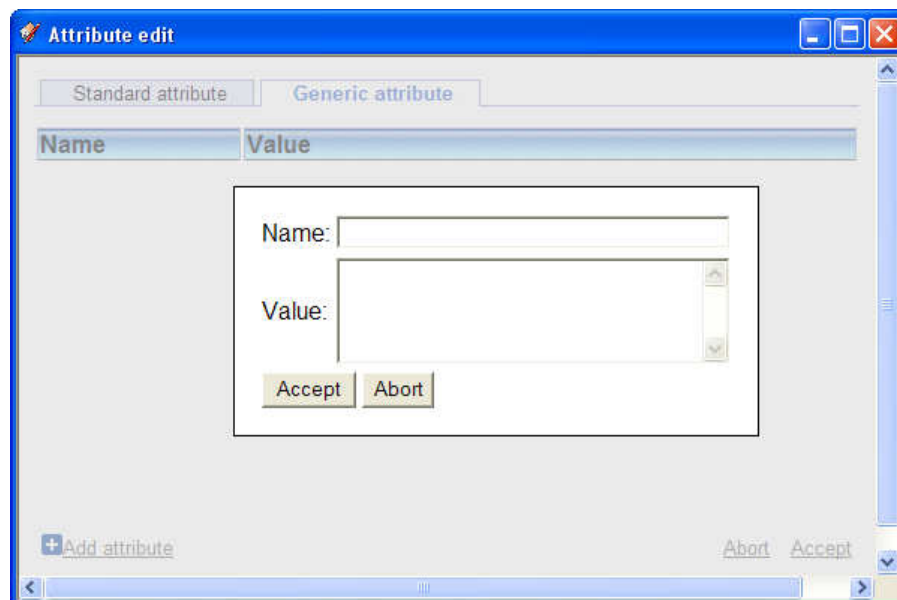


Abb. 13: Generisches Attribut (nur CityGML *StringAttribute*) hinzufügen

6.2 Attribut Surfacetyp

Für jede Fläche kann festgelegt werden, welchen SurfaceType diese Fläche beim Export erhalten soll. Um den SurfaceType einer Fläche zu verändern, klicken Sie mit der rechten Maustaste auf diese Fläche und wählen in dem erscheinenden Kontextmenü den Punkt **CityGML SurfaceType** (Abb. 14). Es erscheint eine Liste, aus der Sie **GroundSurface**, **RoofSurface** oder **WallSurface** auswählen können. Der aktuell zugeordnete SurfaceType wird durch ein Häkchen angezeigt. Fehlt das Häkchen, so ist kein SurfaceType zugeordnet. Mit **None** entfernen Sie SurfaceType einer Wand.

Beim Exportieren werden Flächen als *lod1Multisurface* .. *lod4MultiSurface* unterhalb des entsprechenden SurfaceType-Elements (z.B. *WallSurface*) abgelegt. Die BoundarySurfaceType ID legt die ID des SurfaceType-Elements fest. Flächen vom gleichen SurfaceType und gleicher BoundarySurfaceType ID werden als *lod1Multisurface* .. *lod4MultiSurface* im selben SurfaceType-Element gespeichert. Flächen mit unterschiedlicher BoundarySurfaceType IDs werden dagegen immer getrennten SurfaceType-Elementen zugeordnet.

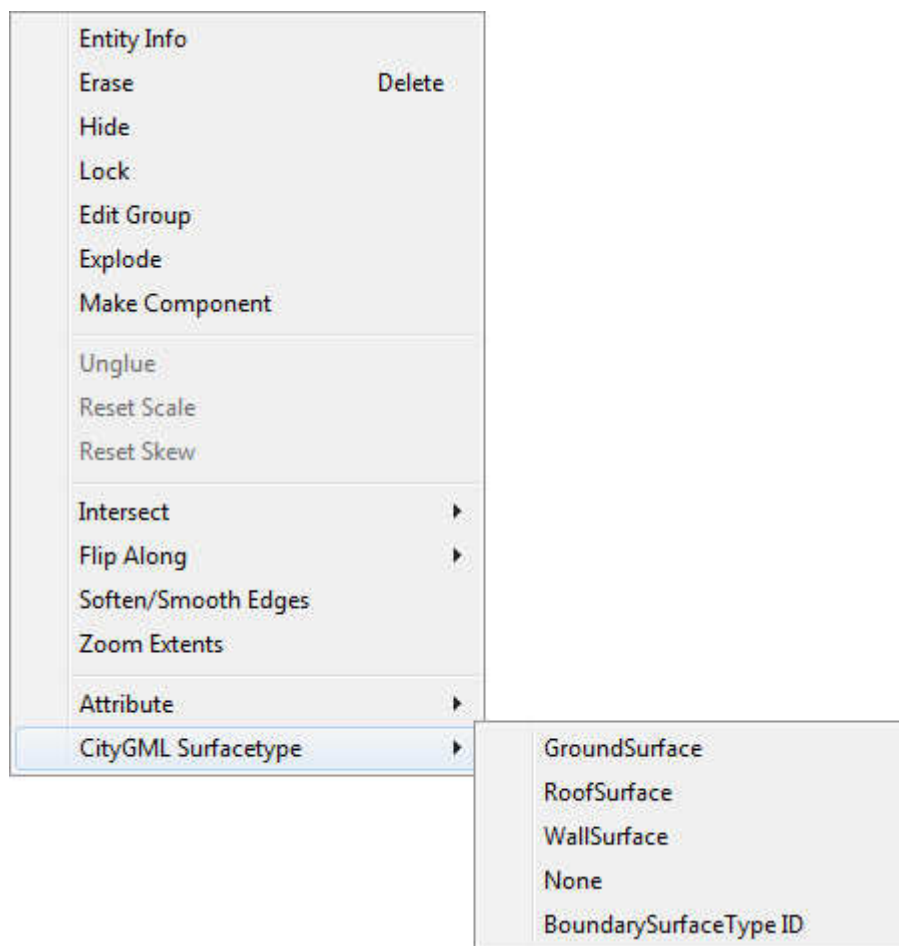


Abb. 14: Kontextmenü CityGML SurfaceType

6.3 Attribute Kopieren / Einfügen

Beim Bearbeiten von Modellen muss gelegentlich eine Fläche unter Beibehaltung ihrer Attribute durch eine andere Fläche ersetzt werden. Damit die Attribute nicht wieder neu eingetragen werden müssen, können alle Attribute einer Fläche oder eines Gebäudes über die Zwischenablage auf eine andere Fläche oder ein anderes Gebäude kopiert werden.

Um Attribute einer Fläche in die Zwischenablage zu kopieren, wählen Sie mit der rechten Maustaste die gewünschte Fläche aus und klicken im erscheinenden Menü auf **Attribute** → **Copy** (Abb. 10). Die Attribute werden hierdurch in die Zwischenablage als sogenanntes Attribute Set kopiert. Solange SketchUp nicht geschlossen wird, können alle Attribute Sets wieder in das Modell eingefügt werden.

Sollen Attribute aus der Zwischenablage auf eine Fläche übertragen werden, wählen Sie diese Fläche mit der rechten Maustaste aus und klicken auf **Attribute** → **Insert**. Es erscheint eine Auswahl aller in der Zwischenablage befindlichen Attribute-Sets, die zu dem angewählten Element (Fläche oder Gebäude) passen. Wählen Sie eines der Attribute Sets aus und übertragen Sie mit **Auswahl übernehmen** alle in diesem Set enthaltenen Attribute auf die ausgewählte Fläche.

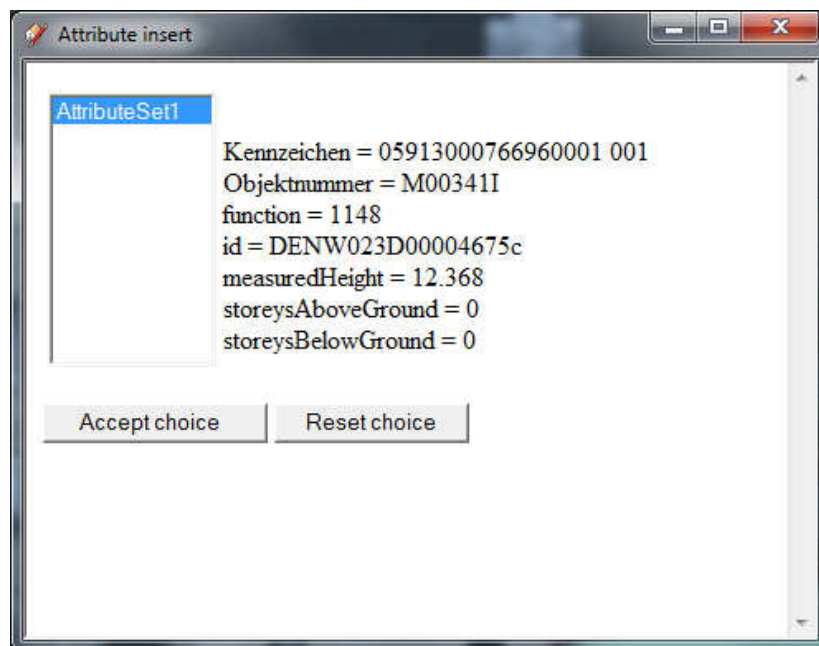


Abb. 15: Anzeige der Attribut Sets aus der Zwischenablage

7 Modellierungshinweise

Damit ein Modell aus SketchUp korrekt nach CityGML exportiert wird, müssen einige Details beim Modellieren beachtet werden.

7.1 Elemente richtig gruppieren

In SketchUp können Elemente gruppiert werden. Beim CityGML-Export werden gruppierte Flächen (Faces) aus der obersten Gruppierungshierarchie als *Building* interpretiert. Gruppierungen innerhalb von Gruppierungen werden ignoriert, sollen aber in späteren Versionen des CityGML-Plugins als *BuildingPart* interpretiert werden. Daraus ergibt sich folgender Modellierungshinweis. Gruppieren Sie in der obersten Ebene Elemente so, dass sie in Ihrem Stadtmodell einem Building entsprechen. Verzichten Sie unbedingt darauf, diese Gruppen weiter zu gruppieren, um die Zuordnung zu Buildings nicht zu zerstören. Verzichten Sie außerdem auf weitere Untergruppen in den "Buildings", es sei denn, sie verstehen diese als *BuildingPart*. Ein *BuildingPart* ist im übrigen ein Teilgebäude und kein Gebäudeteil!

7.2 Selbstschneidenden Polygone vermeiden

Unter Umständen können bei der Modellierung sich selbstschneidende Polygone entstehen. Solche sich selbstschneidenden Polygone werden beim Export nicht erkannt. Sie sind in CityGML nicht erlaubt und es ist daher nicht definiert, wie sie in CityGML-Viewern angezeigt werden.

7.3 Datenhaltung möglichst im SKP-Format

Wir empfehlen bei allen Modellen, die ihren Ursprung in SketchUp haben, die SKP-Dateien als "Originaldaten" zu behalten und Änderungen nach Möglichkeit aus den SKP-Originaldaten abzuleiten, in denen übrigens auch alle CityGML-bezogenen Modelldaten (versteckt) abgespeichert sind, wie Attribute, Klassifizierung von Flächen, Definition von Gebäuden usw., so dass nichts CityGML-Spezifisches bei einer Abspeicherung als SKP-Datei verloren geht.

7.4 Kein Material auf gruppierte Flächen

In SketchUp besteht die Möglichkeit, Material auf eine Gruppe von Faces zu "gießen". Dieses wird dann auf allen Faces der Gruppe sichtbar, ohne dass es datentechnisch in die entsprechenden Faces kopiert wird. Da beim CityGML-Export aber ausschließlich Materialien berücksichtigt werden, die unmittelbar den Faces zugeordnet sind, bleiben solche "indirekt" definierten Materialien unberücksichtigt.

7.5 Material nur auf der Vorderseite

In SketchUp kann, im Unterschied zu CityGML, eine Wand (Face) von beiden Seiten Material (Textur und/oder Diffuse Color) besitzen. Beim Export eines Face mit Material auf der Vorderseite wird vorhandenes Material auf der Rückseite ignoriert. Bei einem Face mit Material ausschließlich auf der Rückseite, wird die Rückseite samt Material durch Änderung der Orientierung zur Vorderseite erklärt und anschließend exportiert. Letztere berücksichtigt die Tatsache, dass viele fehlerhafte SketchUp-Modelle (Wände als texturierte Rückseiten, die nach außen zeigen) dennoch richtig nach CityGML exportiert werden.

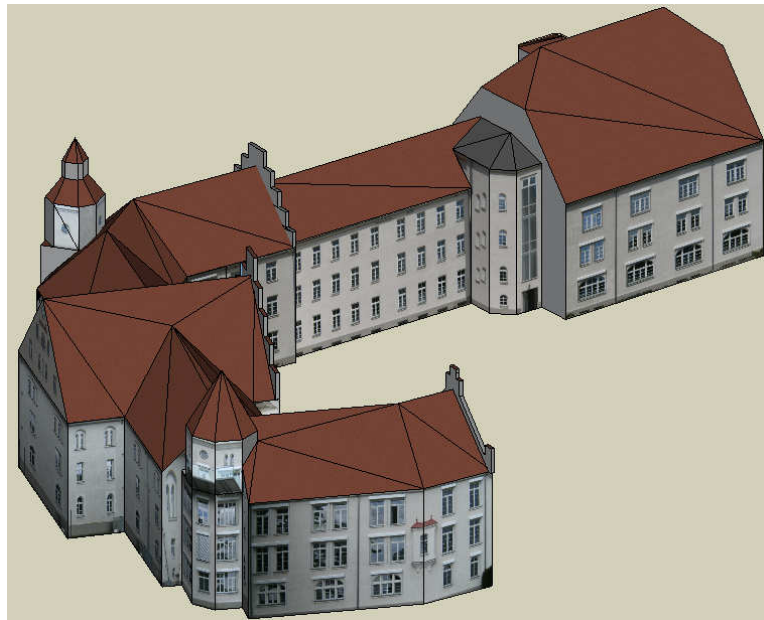
Zur Vermeidung von Missverständnissen hinsichtlich von Texturen auf richtigen Seiten empfehlen wir folgendes Vorgehen:

- Jede Fläche (Face) wird so modelliert, dass die sichtbare Seite, d.h. die Außenseite bei Gebäuden, die Vorderseite ist. Die Vorderseite ist in SketchUp daran zu erkennen, dass sie weiß ist. Rückseiten sind dagegen standardmäßig grau/blau. Eigentlich dürften in einem Modell ohne Material keine grau/blauen Flächen zu sehen sein, sofern man sich außerhalb von Gebäuden befindet.
- Texturen werden ausschließlich an der Vorderseite angebracht. Rückseiten werden, da nie sichtbar, überhaupt nicht mit Material belegt.

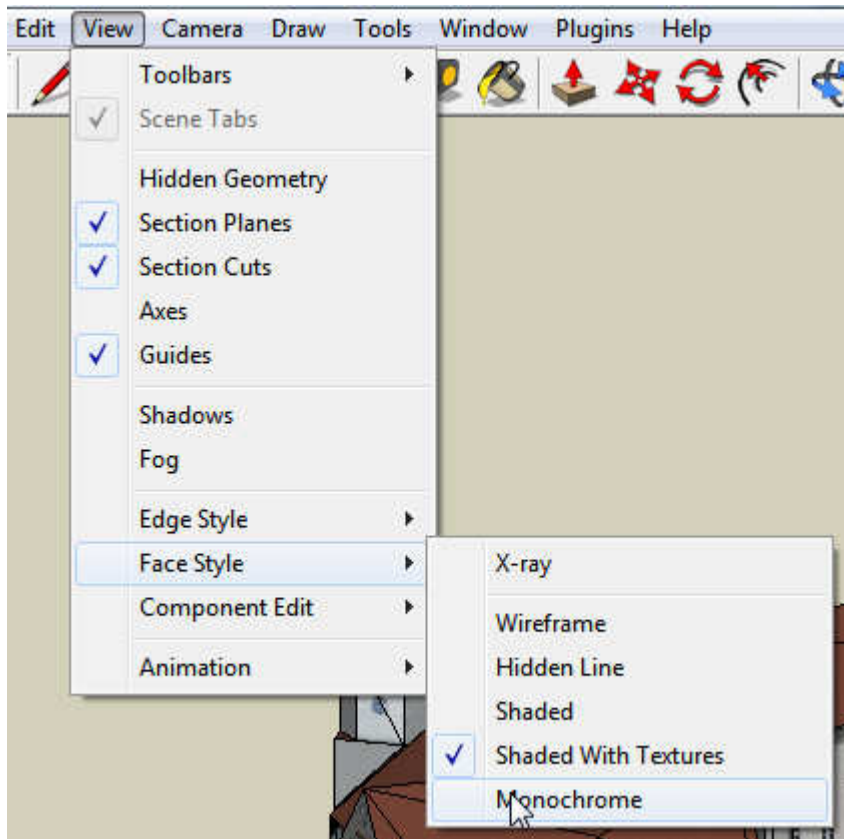
Anmerkung: Ein weiterer Grund dafür, dass die Außenseiten von Gebäuden tatsächlich als Vorderseiten von Flächen modelliert werden sollten, ist deren korrekte Darstellung in entsprechenden Viewern beim Vorhandensein von gerichtetem Licht.

Testen von Modellen

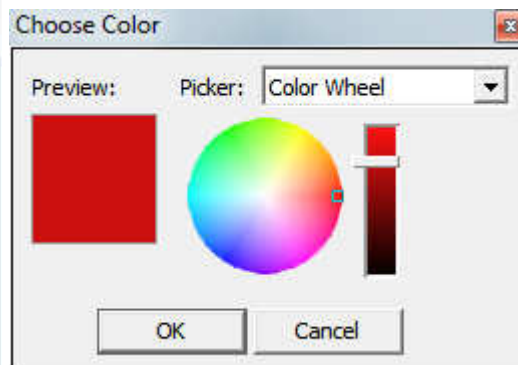
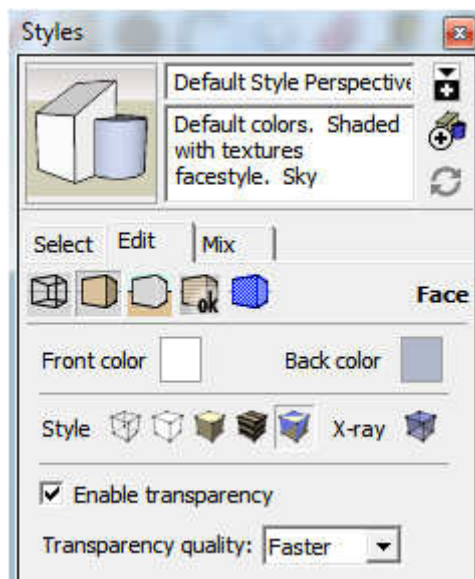
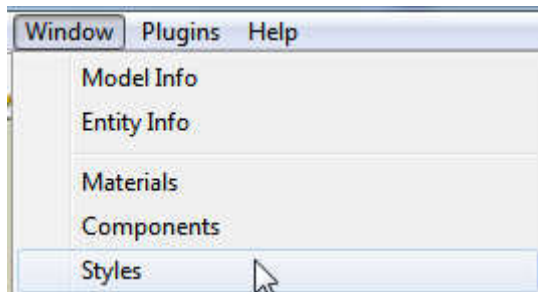
Der nachfolgende Screenshot zeigt ein auf den ersten Blick korrektes Modell:



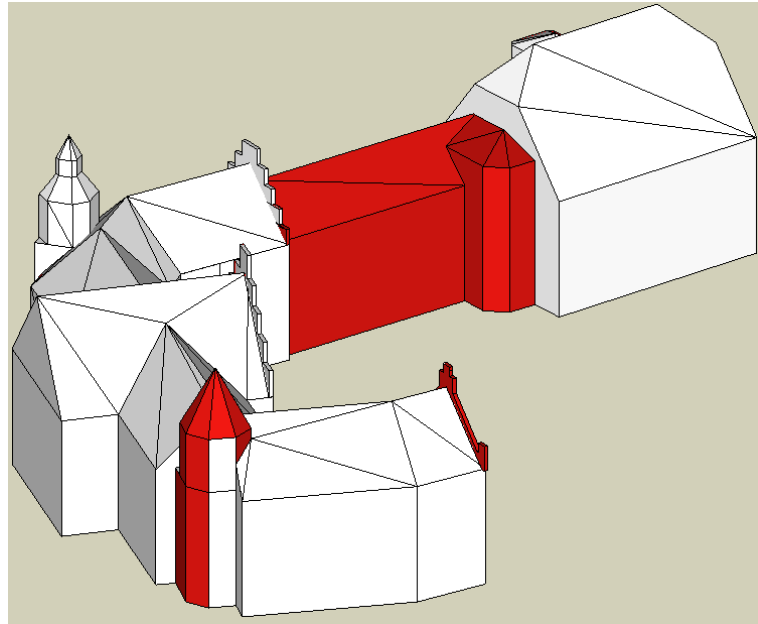
Tatsächlich enthält dieses Modell eine Reihe falsch orientierter Flächen. Die richtige Orientierung von Flächen können Sie einfach prüfen, indem Sie über **View** → **Face Style** → **Monochrome** die Darstellung von Material ausblenden:



Wenn Sie dann noch über **Windows** → **Styles** → **Edit** (Reiter) die Rückseiten rot einfärben



erhalten Sie eine Sicht auf das Modell, in der Sie die falsch orientierten Flächen (rot) sofort erkennen:



Diesen Test sollten Sie durchführen, bevor Sie für die erstellten Flächen Material definieren.

8 Rückmeldungen

Das CityGML-Editor Plugin befindet sich noch in der Entwicklungsphase. Daher ist nicht auszuschließen, dass es in einigen Fällen zu Fehlermeldungen oder fehlerhaftem Verhalten kommt. Sollte Ihnen ein solcher Fehler auffallen, übermitteln Sie uns bitte mit der integrierten Fehlerberichterstattung unter **Plugins → CityGML → Feedback** eine Fehlerbeschreibung, damit wir diesen Fehler beheben können. Sollten Sie darüber hinaus weitere Verbesserungsvorschläge haben, können Sie uns diese ebenfalls mit Hilfe der Fehlerberichterstattung übermitteln.

Produktinformationen		Systeminformationen	
Produkt	Editor	Betriebssystem	Microsoft Windows 7 Professional IC
Version	1.3	Prozessor	Intel(R) Core(TM)2 Duo CPU E74
Build	3304	Speicher	Total virtual: 7336460, Free virtual: ...
Typ	Fehlerbericht	Anzeige	ATI Radeon HD 4800 Series, Vertic: ...
Name		.NET Runtime	v2.0.50727
Email		HINWEIS: Alle gesammelten Informationen werden ausschließlich für die Fehleranalyse verwendet.	
Betreff			
Beschreibung			

Abb. 16: Feedback-Dialog

9 Auf Update prüfen

Innerhalb von SketchUp kann überprüft werden, ob eine neuere Version des CityGML-Editors verfügbar ist. Dafür wird eine aktive Internetverbindung benötigt. Die Überprüfung kann über den Menüpunkt **Plug-ins** → **CityGML** → **Search for updates...** gestartet werden. Durch einen Klick auf **Jetzt suchen** wird die Versionsnummer des momentan installierten Editors mit der aktuellen Version auf www.citygml.de überprüft. Sollte eine neuere Version verfügbar sein, erscheint ein entsprechender Hinweis.

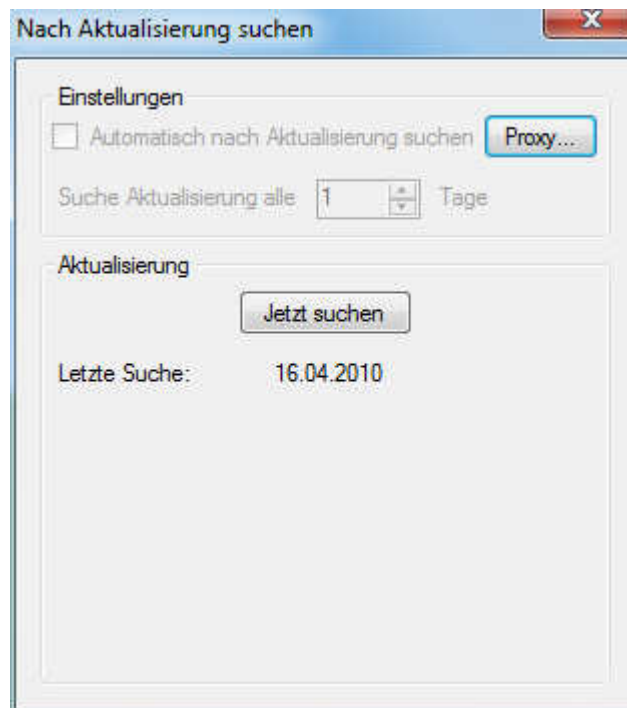


Abb. 17: Dialog zum Suchen nach Updates