Archiwum Fotogrametrii, Kartografii i Teledetekcji, Vol. 20, 2009, s. 161–171
ISBN 978-83-61-576-10-5

CITYGML W ŚWIETLE INTEROPERACYJNOŚCI TRÓJWYMIAROWYCH DANYCH GEOPRZESTRZENNYCH*

CITYGML IN THE INTEROPERABILITY OF 3D GEODATA

Renata Jędryczka

Katedra Fotogrametrii i Teledetekcji, Uniwersytet Warmińsko-Mazurski w Olsztynie

SŁOWA KLUCZOWE: CityGML, KML, inteoperacyjność geodanych, wolne oprogramowanie

STRESZCZENIE: Znaczny wzrost liczby różnych technik pozyskiwania i przetwarzania danych geoprzestrzennych dotyczący zarówno sprzętu jak i metod cyfrowych, a także form ich udostępniania, wymusza stworzenie standardów międzynarodowych do zapisu, wymiany i wizualizacji tych danych. W odpowiedzi na to zapotrzebowanie powstał między innymi język CityGML, ogłoszony przez OGC (*The Open Geospatial Consortium*), jako standard do reprezentacji, magazynowania i wymiany trójwymiarowych modeli wirtualnych miast, a także modeli terenu. Natomiast język KML konsorcjum OGC uznało za standard, nie tylko do tworzenia dwuwymiarowych internetowych map, ale także dla trójwymiarowych geo-przeglądarek (ang. *earth-browsers*). W artykule pokazano CityGML na tle innych formatów dotyczących trójwymiarowych obiektów budowlanych oraz porównano języki CityGML oraz KML. Zawarto również przegląd wolnego oprogramowania do pracy z CityGML, które wspiera OGC. Przedstawiono ponadto aplikację, napisaną w języku Java, do automatycznej konwersji obiektów geometrycznych zapisanych w CityGML do obiektów, które można zamieszczać w języku KML.

1. CITYGML A INNE JĘZYKI I FORMATY

Powszechna globalizacja to jeden z czynników wymuszający wymianę informacji oraz współdziałanie we wszystkich dziedzinach nauki i techniki współczesnego świata. Rozwój zaś Internetu sprawia, że jest to dzisiaj powszechna forma komunikacji, a usługa WWW, jest tą z której w sieci korzystają dzisiaj wszyscy. Pomyślana została jako sposób sięgania do zasobów zgromadzonych na komputerach o różnej konfiguracji zarówno sprzętowej jak i programowej. Dane, na początku tylko tekstowe, dzisiaj również obrazowe, czy dźwiękowe, mogą być rozproszone po całym świecie i udostępniane również w czasie rzeczywistym.

W odpowiedzi na tak zmieniający się rynek informatyczny, oraz by sprostać zadaniom interoperacyjności, powstał meta język XML, a na jego bazie wiele specjalistycznych języków. W grupie pochodnych XML znajduje się także język GML (ang. *Geography Markup Language*), którego wersja 3 stała się bazą dla języka CityGML. Oba te języki są rekomendowane przez OGC (*The Open Geospatial Consortium*) jako standardy dla danych geoprzestrzennych, przy czym GML jest językiem wzorcowym dla systemów geograficznych i otwartym formatem wymiany danych geograficznych poprzez Internet, natomiast CityGML standardem przedstawiania, magazynowania i wymiany trójwymiarowych modeli wirtualnych miast i terenu.

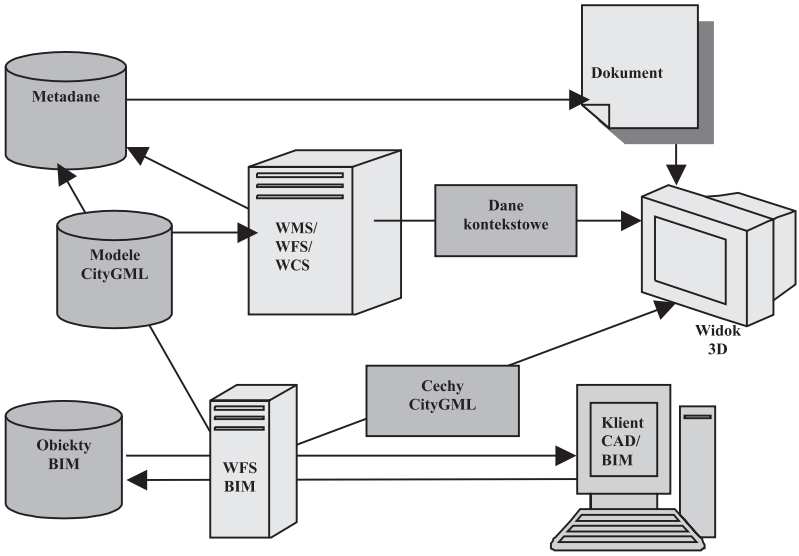
W grupie języków tzw. xml-owych znajduje się również język KML, stworzony przez Google, który w wersji 2.2 uznany został przez OGC jako internetowy standard wizualizacji map 2D i tzw. geo-przeglądarek 3D powierzchni Ziemi (ang. *earth browsers*).

CityGML powstał w odpowiedzi na potrzebę stworzenia standardu, który dla modeli 3D zawiera ich cechy geometryczne, topologiczne, semantyczne, ale także dotyczące ich wyglądu. Nie jest on jednak jedynym formatem semantycznym modeli 3D znajdującym się w użyciu.

Już w końcu latach 80-tych ubiegłego wieku pojawiło się określenie *Building Information Modelling* (BIM), które odnosiło się do koncepcji budowy modeli wirtualnych, jaką przedstawiono w programie ArchiCAD węgierskiej firmy Graphisoft. Z czasem przyjęto BIM jako wspólną nazwę procesu budowy semantycznych modeli cyfrowych i ich przekształcania w obszarze AEC/FM (ang. *Architecture, Engineering and Construction/Facilities Management*) oraz CAD (ang. *Computer Aided Design*). Obiekty w BIM opisywane są w trzech wymiarach, wraz ze swoimi cechami takimi jak np. rodzaj materiału budowlanego, ponadto uwzględniane są ich wzajemne powiązania. Do wymiany danych używa się natomiast klas standardu IFC (ang. *Industry Foundation Classes*).

IFC, posiadający certyfikat ISO, wprowadzony został w 1995 roku przez IAI (ang. *International Alliance for Interoperability*), (IFC, 2009; IAI, 2009). Dopiero dwa lata później pojawiła się pierwsza wersja języka XML, stąd pierwsze modele BIM/IFC były zapisywane w formie zwykłych plików tekstowe, gdyż tylko w ten sposób była możliwa ich dystrybucja i wymiana między różnymi systemami czy programami. Modele IFC zostały zaadaptowane przez większość programów typu CAD, jak np. firmy Autodesk. Obecnie istnieje również specyfikacja IFC nawiązująca do notacji XML pod nazwą ifcXML. IFC posiada klasy odpowiadające budynkom, brak natomiast osobnych klas dla elementów topografii takich jak np. teren, wody itp. (IFC, 2009). Można je opisać tylko poprzez ogólną klasę IFC_Proxy. Podobnie jak w GML, budynki to obiekty posiadające swoją geometrię (własności przestrzenne) jak i własności semantyczne. Modele geometryczne budowane są w IFC korzystając z różnych metod np. z: CSG (ang. *Constructive Solid Geometry*), czy bogatszej w operacje na zbiorach metody B-rep (ang. *boundary representation*). Nie jest jednak możliwa lokalizacja geograficzna obiektów, gdyż w IFC brak nawiązania do powszechnie obowiązujących układów odniesienia CRS (ang. *Coordinate Reference Systems*).

Obiekty IFC mogą być transformowane do obiektów CityGML (Isikdag, Zlatanova, 2009). Przykładowo możliwość taką posiada program IfcExplorer, który powstał w *Institute for Applied Computer Science* w Karlsruhe (IfcExplorer, 2009). Jednak jak na razie zadowalające testy przeprowadzono jedynie dla poziomów LOD1 i LOD2.



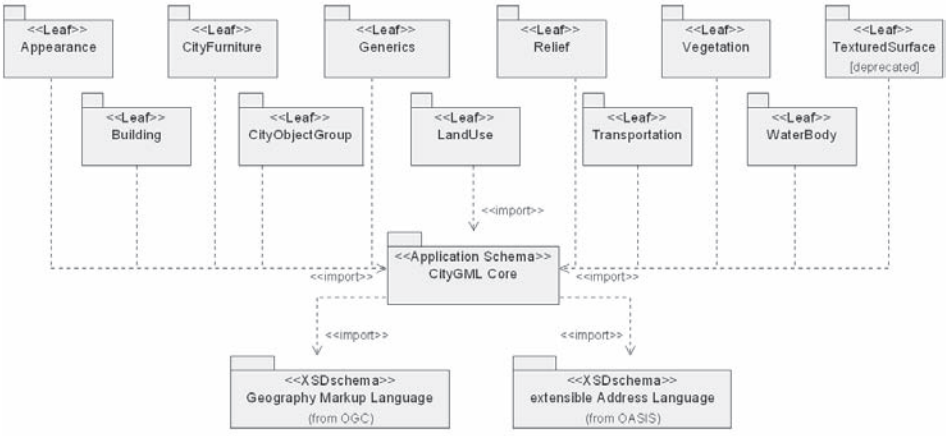
Rys. 1. Metadane, modele i wizualizacja w świetle OGC.
(Źródło: Döllner, 2008)

Na rysunku 1 przedstawiono wzajemne powiązania między BIM/CAD oraz CityGML w nawiązaniu do serwisów OGC (Döllner, 2008).

Obiekty zapisane zarówno w CityGML, jak i w IFC, można wizualizować posługując się standardami grafiki takimi jak X3D/VRML czy COLLADA (Kolbe *et al.,* 2009). Do grupy języków umożliwiających prezentację graficzną geodanych można także zaliczyć język KML, który posiada zarówno odpowiednie znaczniki umożliwiające umieszczanie obiektów geometrycznych na ekranie jak i mechanizmy pozwalające wstawiać w swój kod pliki w formacie COLLADA. W obu językach, X3D i KML, można stosować orientację geograficzną obiektów.

1.1. Modułowa budowa języka CityGML

W literaturze można znaleźć wiele opracowań dotyczących języka CityGML, ale bazą dla nich wszystkich jest obecnie specyfikacja jego oficjalnej wersji 1.0.0 z roku 2008 (Standard CityGML, 2008). Warto przy tym zwrócić uwagę na to, że język ten posiada budowę modułową. Oprócz modułów podstawowych (ang. *core module*) zawiera on 11 modułów tematycznych (ang. *extension modules*): *Appearance, Building*, *CityFurniture*, *CityObjectGroup*, *Generics, LandUse*, *Relief*, *Transportation*, *Vegetation*, *Water-Body, TexturedSurface [deprecated]*. Ostatni z nich nawiązuje do standardu X3D



Rys. 2. Diagram pakietów UML ilustrujący moduły CityGML.
(Źródło: Standard CityGML, 2008)

Tab. 1. Poziomy szczegółowości LOD w CityGML.

Kategorie obszarów	Wagi
Parki krajobrazowe z otuliną	
Tereny parków krajobrazowych poza dopuszczaną strefą zabudowy wg planu ochrony	100
Tereny parków krajobrazowych z dopuszczoną zabudową wg planu ochrony	5
Otulina parków krajobrazowych oraz obszar parków krajobrazowych z dopuszczeniem zabudowy	1
Obszary ochronne uzdrowiska Swoszowice	
strefa A	10
strefa B	5
strefa C	1
Strefy ochronne ujęć wód	
Teren ochrony bezpośredniej ujęcia wód podziemnych i powierzchniowych	100
Teren ochrony pośredniej ujęcia wód podziemnych	1
Teren ochrony pośredniej ujęcia wód powierzchniowych	10
Tereny ograniczonego użytkowania dla lotniska Kraków–Balice	
Strefa A	10
Strefa B	5
Strefa C	1
Tereny występowania zagrożenia powodzią	
Tereny bezpośredniego zagrożenia powodzią (tereny między wałami ochronnymi i leżące w strefie 50 m od napowietrznej podstawy wału przeciwpowodziowego)	100
Tereny potencjalnego zagrożenia powodzią	1

(Źródło: Standard CityGML, 2008)

i włączono go ze względu na modele CityGML z wcześniejszej wersji języka. Takie podejście pozwala tworzyć dowolne kombinacje modułów tematycznych, w powiązaniu z modułami głównymi, zwane profilami CityGML.

Na diagramie zamieszczonym na rysunku 2 zilustrowano wzajemne powiązania miedzy modułami, przy czym wszystkie one bazują na schemacie języka GML 3.1.1.

Podstawową cechą CityGML jest także to, iż każdy obiekt może być reprezentowany jednocześnie na różnych stopniach szczegółowości tzw. LOD (ang. *Levels of Detail*). Język ten zawiera 5 poziomów, które wymagają danych w trzech wymiarach o różnym stopniu dokładności, których zakresy zawiera tabela 1.

Jedną z idei powstania CityGML było także pokazanie, że budując modele wirtualnych miast można łączyć dane pozyskane z różnych źródeł (Stadler *et al*., 2009). Analizując parametry zawarte w tabeli 1 widać, że metody fotogrametryczne pozyskiwania danych w pełni zaspokajają zamieszczone tam wymagania, a zatem mogą one być wykorzystane przy zasilaniu bazy modeli opisanych językiem CityGML. Obecnie dane z lotniczych skanerów laserowych czy zdjęć, po niezbędnych przekształceniach stanowią podstawowe źródło danych 2D i 3D przy określaniu cech geometrycznych modeli.

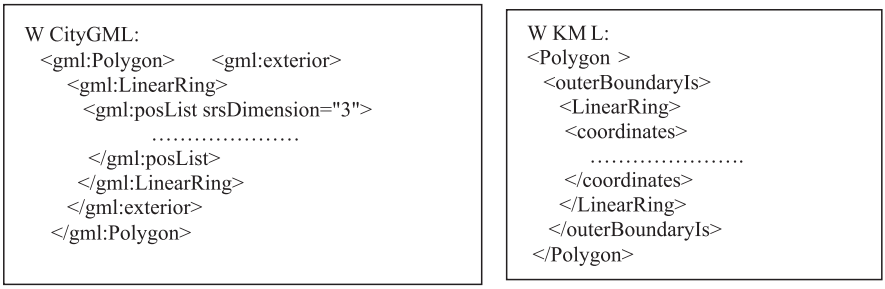
1.2. KML i CityGML

Oba języki, zarówno CityGML jak i KML, zostały rekomendowane przez OGC do pracy z geodanymi, ale ich przeznaczenie jest różne. Posiadają jednak wiele zbieżnych cech jak przede wszystkim obiekty geometryczne, takich jak np. punkty (*points*) czy wieloboki (*polygons*), które pozwalają opisać te same cechy geometryczne obiektów. Przykładowo, na rysunku 3, zamieszczono fragmenty plików w obu tych językach dotyczące zapisu wieloboku, gdzie odpowiadające sobie znaczniki wytłuszczono. Wybrano znacznik *Poligon*, gdyż służy on do zapisu np. ścian czy połaci dachów modeli budynków.

Języki te różnią się natomiast przede wszystkim możliwością zapisu cech semantycznych obiektów - w odróżnieniu od CityGML język KML w ogóle ich nie uwzględnia. Stopień złożoności CityGML jest też większy i dodatkowo zawiera on moduł do szczegółowego opisu wyglądu obiektów, a także ich topologii, co jest niedostępne w KML. Wszystkie relacje wzajemne między topologią przestrzeni, a przestrzennymi obiektami geometrycznymi, oparte są w CityGML na koncepcji *XLink,* np. obiekt *Poligon:* <gml: Polygon gml:id=”wallSurface4711”> i nawiązanie do niego w obiekcie *surfaceMember:* <gml:surfaceMember xlink:href=”#wallSurface4711”/> przez *link:ref*.

Mimo, że dane w KML nie są zorganizowane zgodnie z GML, to jednak na poziomie geometrycznym nawiązuje on do koncepcji języka GML 2.1.2 z ustalonym z góry układem przestrzennym - KML używa globalnego układu odniesienia WGS84. Natomiast obiekty w CityGML mogą być zapisane, albo w układzie globalnym, albo we własnym układzie lokalnym, a następnie transformowane do układu globalnego (znana macierz transformacji oraz punkt nawiązania do układu przestrzennego CRS).

Ponadto, ponieważ CityGML bazuje na GML 3.0, można łączyć go bez problemu ze wszystkimi standardami OGC takimi jak np. Web Feature Service (WFS), Web Coordinate Transformation Service (WCTS), czy Web Processing Service (WPS) w celu pozyskania, przetworzenia czy zidentyfikowania jego zasobów (Altmaier, Kolbe, 2009).



Rys. 3. Odpowiadające sobie elementy w językach CityGML i KML - fragmenty kodów.

2. CITYGML I WOLNE OPROGRAMOWANIE

2.1. Projekty wspierane przez OGC

Język CityGML przeznaczony jest do użytku ogólnego i bezpłatnego. Powstało także różne oprogramowanie, które umożliwia, bądź ułatwia, pracę z modelami tworzonymi w CityGML, a z którego można korzystać na zasadach tzw. wolnego oprogramowania. Niektóre projekty wspierane są przez OGC i można z nich czerpać pisząc własne aplikacje. W świecie współczesnym zdominowanym przez Internet, większe znaczenie mają aplikacje działające w środowisku rozproszonym, a najczęściej spotykanym językiem programowania jest język Java. Właśnie w nim jest najwięcej materiałów takich jak np. biblioteki, pakiety narzędziowe itp.

Poniżej wskazano na projekty, które udostępniają oprogramowanie na zasadach licencji LPGL (*GNU Lesser Public License*). Licencja tego typu umożliwia łączenie kodu źródłowego z oprogramowaniem nie koniecznie wolnym. Ponadto istotą LPGL w wersji trzeciej jest możliwość przekształcania fragmentów kodu z LPGL na GPL, dzięki czemu można tworzyć nowe wersje kodu, który jednak nigdy nie będzie mógł być rozpowszechniany jako oprogramowanie własnościowe.

W Berlinie w Instytucie Technicznym na Wydziale Geodezji i Geoinformacji powstało oprogramowanie, które pozwala zarządzać modelami CityGML (Berlin, 2009). Jest to oprogramowanie na bazie licencji LGPL w wersji 3, a zawiera ono:

- schemat bazy 3D (*3DCityDBv2*) dla Oracle 10G R2 lub 11G,
- narzędzia do bazy (*3DCityDBv2 Import/Export Tool*),
- bibliotekę klas Javy oraz API (*citygml4j*) do pracy z CityGML.

Biblioteka klas Javy, o nazwie citygml4j (Berlin, 2009), zawiera między innymi kod ułatwiający pracę z obiektami CityGML przy tworzeniu własnego oprogramowania, gdyż wiąże on schemat *XML Schema,* definiujący CityGML, z modelem obiektowym Javy. Wykorzystano przy tym bibliotekę Javy o nazwie JAXB (ang. *Java Architecture for XML Binding*), która jest obecnie częścią Java SE 1.6. Biblioteka obsługuje dwie wersje CityGML 1.0.0 oraz 0.4.0, oraz wspiera konwersję miedzy nimi.

Godny polecenia jest także projekt o nazwie *deegree* (Deegree, 2009), organizacji OSGeo (*Open Source Geospatial Foundation*), która jest organizacją typu "non-profit" (dobrowolne, trwałe i samorządne zrzeszenie o celach niezarobkowych). Powołano ją, aby wspierać oraz tworzyć otwarte oprogramowanie GIS. W budowie tego projektu bierze udział wielu uczestników, przy czym za koordynację projektu odpowiedzialni są przedsiębiorstwo lat/lon (lat/lon, 2009) oraz *GIS Research Group* na Wydziale Geografii Uniwersytetu w Bonn.

Deegree jest to Java Framework (biblioteki kodu źródłowego) oferujący szereg bloków do budowy infrastruktury geodanych stworzonych zgodnie ze standardami OGC.

Oprogramowanie *deegree* dzieli się na dwie grupy:

1) deegree OGC Web Services – strona serwera:
   - deegree Web Map Service (WMS),
   - deegree Web Feature Service (WFS),
   - deegree Web Coverage Service (WCS),
   - deegree Catalogue Service (CSW),
   - deegree Web Terrain Service (WTS) / Web Perspective View Service (WPVS), deegree Web Processing Service (WPS),
2) degree - strona klienta:
   - deegree iGeoPortal (standard edition).

Z niektórych klas biblioteki *deegree Web Feature Service* skorzystano pisząc własną aplikację.

2.2. Aplikacja w języku Java

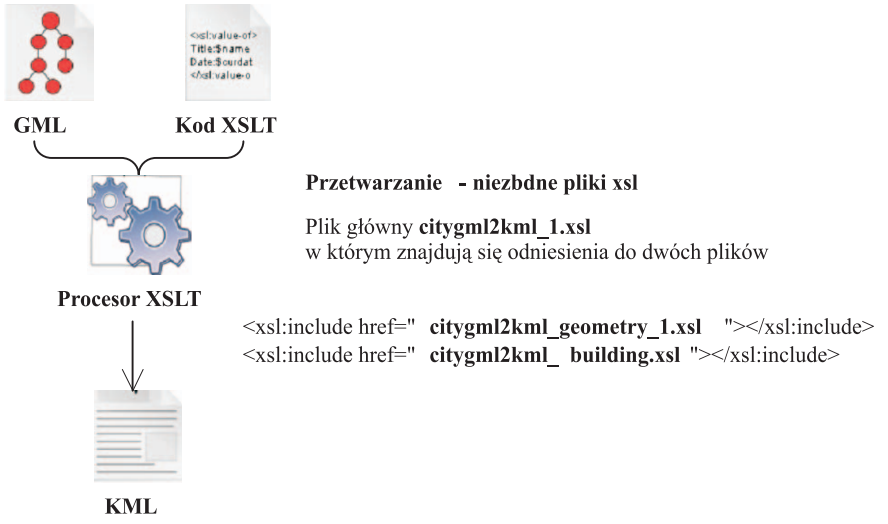
Modele zapisane w formacie języka CityGML mogą być wizualizowane jedynie za pomocą przeglądarek specjalnie do tego przystosowanych takich jak np. darmowy Aristoteles Viever, który powstał w Instytucie Geodezji i Geoinformacji w Bonn (Arystoteles, 2009) i wymaga dodatkowo zainstalowania oprogramowania Java 1.5 i Java 3D, czy komercyjny LandXplorer firmy Autodesk (LandeXplorer, 2009).

Natomiast w przypadku wizualizacji modeli, w przeglądarkach obsługujących format KML (głównie chodzi tu o Google Earth), a zapisanych w formacie CityGML, niestety trzeba przekształcać je do innych formatów przez nie akceptowanych.

W tym celu napisano przykładową aplikację w języku Java do automatycznej transformacji pliku zapisanego w formacie CityGML (rozszerzenie .gml) do formatu języka KML (rozszerzenie .kml). Korzystano przy tym z darmowego środowiska platformy Eclipse. Nie zastosowano jednak formatu COLLADA, do zapisu modelu geometrycznego, ponieważ zdecydowano się na wykorzystanie odpowiadających sobie znaczników obu języków. Oba języki bazują na XML zatem zastosowano transformację XSLT, której schemat ideowy przedstawia rysunek 4.

W aplikacji użyto niektórych klasy pochodzących z dwóch pakietów projektu *deegree*, a odnoszących się do dokumentów XML i przekształceń XSLT, a mianowicie:

- org.deegree.framework.xml.XMLFragment oraz
- org.deegree.framework.xml.XSLTDocument.



Rys. 4. Przekształcenie XSLT: z GML do KML.

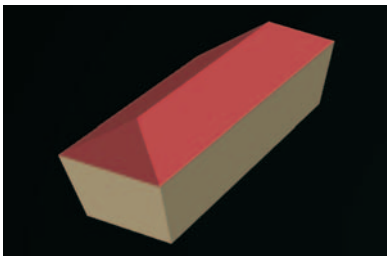
Plikiem głównym dla przekształcenia XSLT jest w aplikacji plik o nazwie citygml2kml_1.xsl, będący modernizacją pliku udostępnianego w projekcie *deegree*.

Do testów wykorzystano prosty model budynku w formacie CityGML na poziomie LOD2. Ponieważ modele mogą być zapisane w dowolnym lokalnym układzie współrzędnych zastosowano układ 2000. Aplikacja przekształca automatycznie współrzędne punktów modelu do układu WGS84, obowiązującego w Google Earth. Wykorzystuje się przy tym bazę danych stworzoną przez EPSG (ang. *European Petroleum Survey Group*) (EPSG, 2009). Poniżej, we fragmencie kodu źródłowego z pliku citygml2kml_geometry_1.xsl, pokazano znaczniki zawierające odwołania do odpowiednich zbiorów, przy czym kod dla zbioru wejściowego (SRCCRS) jest parametrem aplikacji.

```
<xsl:param name="SRCCRS"><xsl:value-of select="//@srsName" /></xsl:param>
<xsl:param name="TARGETCRS">EPSG:4979</xsl:param>
<LinearRing>
<coordinates xmlns:java="java" xmlns:geometryutil="org.deegree.framework.
    xml.GeometryUtils">
<xsl:value-of select="geometryutil:getPolygonInnerRing( .. , position(), $SRCCRS,
    $TARGETCRS )"></xsl:value-of>
</coordinates>
</LinearRing>
```

W aplikacji uwzględniono jedynie transformację cech geometrycznych modeli. Na rysunku 5 zamieszczono wizualizacje tego samego modelu przed i po transformacji, wyświetlonego odpowiednio w aplikacji LandXplorer firmy Autodesk i w GoogleEarth.

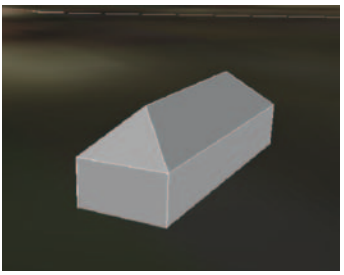
CityGML



Współrzędne układ 2000
EPSG:2178

XSLT

KML



Współrzędne WGS84
EPSG:4979

Rys. 5. Wizualizacje modelu budynku w formacie CityGML i po przekształceniu w KML.

Kolory dachu i ścian nie podlegały rozważaniom i zostały one przydzielone z domniemania przez aplikacje wizualizujące, a różnice w wyglądzie wynikają także z różnego położenia obserwatora względem obiektu (nie uwzględniono parametrów perspektywy widoku: współrzędnych położenia obserwatora, kąta widzenia, odległości od obiektu).

3. PODSUMOWANIE

Obecnie CityGML jest już wspierany przez wiele aplikacji i systemów GIS. Złożone produkty geodanych jakimi są modele 3D miast wymagają odpowiedniego formatu, aby proces zapisu, udostępniania i wizualizacji był sprawny i wymagał jak najmniej pracy. City GML spełnia pod tym względem współczesne wymagania interoperacyjności. Ponieważ jest tzw. językiem xml-owym, może podlegać transformacjom typu XSLT. Zapewnia to potencjalne wykorzystanie go w każdym oprogramowaniu, które zawiera obsługę plików XML.

Na razie nie ma co liczyć na powstanie uniwersalnego formatu dla trójwymiarowej grafiki komputerowej. Najczęściej używane są obecnie standardy takie jak X3D/VRML, COLLADA czy służący do geowizualizacji język KML. Ze wszystkimi tymi formatami CityGML jest kompatybilny, tzn. da się w sposób stosunkowo prosty przenieść do nich modele CityGML. W niniejszym artykule wskazano jedynie na jedną z możliwości, mianowicie transformację cech geometrycznych obiektów do języka KML.

4. LITERATURA

Altmaier A., Kolbe, T., 2009. Applications and Solutions for Interoperable 3d Geo-Visualization. *3D Geo-Information Sciences*, Springer, Lecture Notes in Geoinformation and Cartography, s.253-267.

Deegree. 2009: http://www.deegree.org

Döllner J., 2008. 3D Geoinformation and 3D Geovirtual Worlds, *CEN/TC 287 Interoperability Workshop*, Berlin April 23.

Encoding Standard, ver.1.0.0, Gröger E., Kolbe T., Czerwinski A., Nagel C. (red.) http://www.opengeospatial.org/standards/citygml

EPSG. 2009: http://www.epsg.org

EPSG. 2009: http://www.epsg-registry.org/

IAI. 2009: www.iai-international.org

IFC. 2009: http://www.iai-tech.org/ifc/IFC2x3/TC1/html/index.htm

IfcExplorer. 2009: http://www.iai.fzk.de/www-extern/index.php?id=796&L=1

Isikdag U., Zlatanova S., 2009. Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models. 3D Geo-Information Sciences, Part II, Springer, Berlin Heidelberg, s. 79-96.

Kolbe T., Nagel K., Stadler A., 2009. CityGML – Standard in Photogrammetry? 52nd Photogrammetric Week in Stuttgart, September 7-11.

LandeXplorer. 2009: http://www.landxplorer.com/

lat/lon. 2009: http://www.lat-lon.de/

Stadler A., Nagel K., König G, Kolbe T., 2009. Making interoperability persistent: A 3D geo database based on CityGML. Springer, Lecture Notes in Geoinformation and Cartography, s.175-192.

Standard CityGML. 2008. OpenGIS® City Geography Markup Language (CityGML)

CITYGML IN THE INTEROPERABILITY OF 3D GEODATA

KEY WORDS: CityGML, KML, geodata interoperability, free software

SUMMARY: The substantial increase in the number of different techniques of obtaining and processing geospatial data (an increase that concerns equipment as well as digital methods and methods of rendering the data accessible) necessitates the creation of international standards for the recording, exchange and visualisation of such data. The CityGML language is a response to this need. It is presented by the OGC (Open Geospatial Consortium) as a standard for the representation, storage, and exchange of 3D models of virtual cities or terrain models. As for the KML language, the OGC considers it standard not only for the creation of 2D internet maps but also for 3D earth-browsers. This paper presents CityGML against the background of other 3D building

formats and compares CityGML to KML. It also provides an overview of free, OGC-supported software designed to accompany CityGML. Additionally, it presents a Java-based application that automatically converts CityGML-based geometrical objects to KML-based ones.

dr Renata Jędryczka
renata.jedryczka@uwm.edu.pl
telefon: +48 89 5234915
fax: +48 89 5233841

* wersja kolorowa artykułu jest dostępna na stronie http://www.sgp.geodezja.org.pl/ptfit