

**OPRACOWANIE SCHEMATU APLIKACYJNEGO GML DLA POTRZEB  
TRÓJWYMIAROWEJ WIELOROZDZIELCZEJ BAZY TOPOGRAFICZNEJ**

**GML APPLICATION SCHEMA FOR THREE-DIMENSIONAL  
MULTIRESOLUTION TOPOGRAPHIC DATABASE**

**Urszula Cisło-Lesicka**

AGH Akademia Górniczo-Hutnicza, Wydział Geodezji Górniczej i Inżynierii Środowiska  
Katedra Geoinformacji, Fotogrametrii i Teledetekcji Środowiska

**SŁOWA KLUCZOWE:** baza danych topograficznych, baza wielorozdzielcza, dane 3D, GML, schemat aplikacyjny GML

**STRESZCZENIE:** W ramach badań wykonywanych w projekcie „Konwersja obiektów Bazy Danych Topograficznych do postaci trójwymiarowej dla potrzeb dynamicznej geowizualizacji” (N N526 192537) opracowywana jest koncepcja Bazy Danych Topograficznych w postaci 3D (TBD3D), która może stanowić dodatkowy moduł TBD umożliwiający trójwymiarową wizualizację danych topograficznych. Tworzona koncepcja zakłada istnienie obiektów w trzech wymiarach na trzech poziomach szczegółowości. Obiekty w TBD3D będą przedstawiane w swej zgeneralizowanej postaci 3D lub za pomocą symboli 3D. Trójwymiarowe obiekty TBD3D będą pozyskiwane z danych ewidencyjnych, stereodigitalizacji lub z danych lidarowych albo jako dane pozyskiwane w wyniku superpozycji danych dwuwymiarowych z numerycznym modelem rzeźby terenu. Jednym z etapów niniejszych badań było opracowanie schematu aplikacyjnego GML dla potrzeb TBD3D (TBD3DGML) w celu przesyłania danych pomiędzy różnymi systemami geoinformacyjnymi oraz udostępniania ich przez Internet. Schemat ten pod kątem struktury plików i katalogów oparto na dotychczasowym schemacie dla dwuwymiarowych danych TBDGML. Natomiast geometria i budowa poszczególnych obiektów została dostosowana do wymogów trzeciego wymiaru i standardu GML3.1 oraz GML3.2. Dodatkowo ze standardu CityGML zaczerpnięto możliwość opisywania obiektów prototypowych (*ImplicitGeometry*) oraz wykorzystano moduł *Appearance* i *TexturedSurface* umożliwiający definiowanie wyglądu powierzchni. Opracowany schemat aplikacyjny TBD3DGML został przetestowany na wybranych obiektach TBD3D przedstawiających wszystkie wykorzystane w schemacie typy geometrii. Do zapisu danych w schemacie TBD3DGML wykorzystano oprogramowanie FME firmy Safe Software. Wygenerowane dane zostały następnie zwalidowane pod kątem zgodności z opracowanym schematem TBD3DGML i wczytane do przeglądarki GML AristotelesViewer v.1.3.

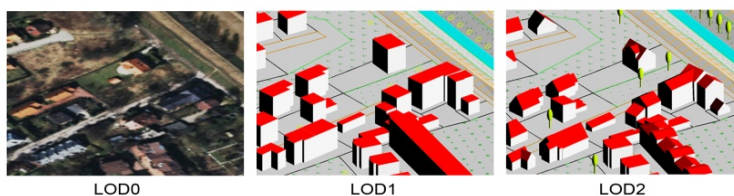
## **1. WPROWADZENIE**

Koncepcja Bazy Danych Topograficznych w postaci 3D (TBD3D) opracowywana w ramach projektu „Konwersja obiektów Bazy Danych Topograficznych do postaci trójwymiarowej dla potrzeb dynamicznej geowizualizacji” (N N526 192537), zakłada istnienie obiektów w trzech wymiarach na trzech poziomach szczegółowości (uogólnienia). W planowanym systemie TBD3D przewiduje się wstępny podział obiektów na dwie główne grupy. Pierwsza składa się z obiektów, które będą przedstawiane w swej rzeczywi-

stej postaci lub za pomocą symboli 3D. Dla obiektów w postaci rzeczywistej (np. budynki) informacja o trzecim wymiarze będzie pozyskana z danych ewidencyjnych, stereodigitalizacji lub poprzez wykorzystanie danych z lotniczego skaningu laserowego. Drugą grupę utworzą obiekty, które uzyskają trzeci wymiar w wyniku superpozycji danych dwuwymiarowych z numerycznym modelem rzeźby terenu (NMT), np. sieć dróg, kolei i cieków wodnych, kompleksy użytkowania terenu (Cisło, 2008).

Ponadto TBD3D planowana jest jako wielorozdzielcza trójwymiarowa baza dla wybranych obiektów 3D tak, by możliwe było, w razie takiego życzenia użytkownika, pokazanie terenu i obiektów 3D, a w szczególności budynków, w różnym stopniu szczegółowości. Obiekty te będą zorganizowane w 3 poziomach szczegółowości – LOD (*Level of Detail*) (Rys. 1):

- LOD0 (model regionalny): ogólna prezentacja modelu regionu, NMT z nałożoną ortofotomapą,
- LOD1 (model blokowy): budynki przedstawione są jako proste bloki ze zgeneralizowaną geometrią i płaskim dachem,
- LOD2 (model geometryczny): budynki mają zaznaczony kształt dachów, wybrane obiekty przedstawiane są za pomocą symboli 3D.



Rys. 1. Trzy poziomy szczegółowości w TBD3D (opracowanie własne)

W ramach ww. projektu po opracowaniu koncepcji TBD3D stworzono bibliotekę symboli 3D, zbadano możliwości pozyskiwania 3D obiektów, w szczególności budynków, oraz sprawdzono możliwości wykorzystania w TBD3D ortofotomapy. Następnie przygotowano wariantowe opracowania TBD3D dla wybranych obszarów testowych. Ostatnim etapem niniejszych badań było opracowanie schematu aplikacyjnego GML dla potrzeb TBD3D (TBD3DGML) w celu przesyłania danych pomiędzy różnymi systemami geoinformacyjnymi oraz udostępniania ich przez Internet.

### 1.1. Geography Markup Language (GML)

Język Geography Markup Language (GML) jest aplikacją języka Extensible Markup Language (XML) opracowaną przez Open Geospatial Consortium (OGC) (OGC, 2008), do zapisu geoinformacji w celu przesyłania jej pomiędzy różnymi systemami geoinformacyjnymi, niezależnie od platformy sprzętowo-systemowej i niezależnie od charakteru i technologii systemu geoinformacyjnego (Gaździcki, 2007) oraz w celu udostępniania danych geograficznych przez Internet. Najnowsza wersja GML 3.2 pozwala na stosowanie nieliniowego modelu geometrii, danych rastrowych, topologii czy geometrii trójwymiarowej. Od 2007 roku GML w wersji 3.2 stanowi międzynarodowy standard de facto do wymiany danych geoinformacyjnych.

Główną zaletą języka GML jest możliwość sformalizowanego opisu obiektów geograficznych wraz z ich geometrią i zależnościami topologicznymi, systemów odniesienia

przestrzennego, czasu czy jednostki miary, dzięki czemu umożliwia on wykonywanie zapytań tematycznych, czy analiz przestrzennych na obiektach. Ponadto, ponieważ GML jest aplikacją XML, można łatwo rozszerzyć strukturę informacji zawartej w GML poprzez możliwość wykorzystania wcześniej zdefiniowanych struktur. Należy jednak pamiętać, że GML nie jest w dosłownym sensie językiem aplikacyjnym, przez co nie można go użyć wprost do celów praktycznych. GML definiuje jedynie elementy dotyczące aspektu przestrzennego geoinformacji i z tego powodu, aby zastosować go w konkretnych dziedzinach konieczne jest opracowanie schematów GML zawierających elementy specyficzne dla danej dziedziny. Dlatego też zapis danych w języku GML jest dwuczęściowy: „surowe” dane zapisywane są w dokumentach GML (pliki \*.gml lub \*.xml), a deklaracyjny opis danych, opisujący ich logiczną organizację w pliku oraz ich znaczenie i kontekst, podany jest w schemacie aplikacyjnym GML (pliki \*.xsd) (Kmieciak, 2005). Dzięki takiemu zapisowi otrzymujemy model, do którego możemy porównać konkretne dokumenty w celu przetestowania ich pod kątem zgodności ze specyfikacją języka. Przykładem takiego schematu aplikacyjnego GML2 (w wersji 2) jest schemat TBDGML przeznaczony do modelowania dwuwymiarowych obiektów Bazy Danych Topograficznych (GUGiK, 2008), czy schemat aplikacyjny GML3 (w wersji 3) CityGML będący standardem przeznaczonym do modelowania miast 3D (OGC, 2008). Szczegółowy opis wymienionych schematów GML przedstawiono w poniższych podrozdziałach.

## 1.2. Schemat aplikacyjny TBDGML

Schemat aplikacyjny TBDGML został określony przez Główny Urząd Geodezji i Kartografii (GUGiK) w „Wytocznych technicznych. Baza Danych Topograficznych (TBD) – wersja 1.0, uzupełniona” (GUGiK, 2008) w celu przekazywania danych pozyskiwanych dla TBD do zasobu geodezyjno-kartograficznego. Format GML przeznaczony jest do wymiany danych wektorowych i opisowych oraz danych zasobu kartograficznego, co ma szczególne znaczenie w procesie pozyskiwania danych od wykonawców prac.

Obecnie obowiązujący schemat aplikacyjny TBDGML 2.0.4 został oparty na specyfikacji World Wide Web z roku 2001, a elementy schematu opierają się na schemacie GML w wersji 2.1.2, który stanowi integralną część schematu aplikacyjnego TBDGML. W ramach schematu aplikacyjnego TBDGML zostały wydzielone dwa niezależne schematy, które funkcjonalnie odpowiadają definicjom zbiorów danych komponentów TOPO (zasób podstawowy) i KARTO (zasób kartograficzny). Schematy aplikacyjne TBDTOPO i TBDKARTO skonstruowane zostały na zasadzie łączenia schematów cząstkowych odzwierciedlających sposób wyodrębnienia poszczególnych plików GML na poziomie definicji zbioru danych (GUGiK, 2008). Dla schematu TBDKARTO wyodrębniono 132 schematy cząstkowe opisujące obiekty w komponencie KARTO oraz 2 schematy definiujące elementy podstawowe. Natomiast w TBDTOPO mamy 76 schematów cząstkowych opisujących obiekty w komponencie TOPO, 2 schematy podstawowe i 3 schematy definiujące dopuszczalne kody atrybutów. Każdy plik z danymi TBD powinien odnosić się bezpośrednio do odpowiedniego schematu cząstkowego definiującego jego elementarną strukturę danych. Każdy schemat cząstkowy, prócz definicji właściwej klasy obiektów, zawiera również odwołanie do zdefiniowanego globalnie typu *Metadane* umożliwiając w ten sposób dołączanie do plików z danymi metadanych na poziomie pojedynczej klasy obiektów.

W schemacie aplikacyjnym TBDGML, ze względu na podobny sposób konstrukcji wybranych klas obiektów, wyodrębnione zostały typy bazowe określające np. atrybuty specjalne zdefiniowane dla każdego obiektu TBD oraz typy geometryczne (punkt, linia i obszar).

### 1.3. Schemat aplikacyjny CityGML

W 2002 r. Special Interest Group 3D (SIG 3D) zaczęła wdrażać schemat aplikacyjny GML w wersji 3.1 pod nazwą CityGML, który przeznaczony jest do modelowania miast 3D. Zasadniczym zadaniem tego schematu jest uzyskanie sformalizowanego opisu podstawowych elementów, klas, atrybutów i relacji w modelu miasta 3D, w odniesieniu do ich właściwości geometrycznych, topologicznych i semantycznych (Kolbe, Bacharach, 2006). CityGML, obecnie w wersji 1.0, ma status międzynarodowego standardu OpenGIS do reprezentacji, przechowywania i wymiany danych wirtualnych miast 3D oraz modeli regionalnych.

Ogólnie obiekt *Miasto (City)* zdefiniowany jest tak, aby obejmował nie tylko konstrukcję budynków, ale również elewacje, roślinność, zbiorniki wodne, elementy infrastruktury drogowej, numeryczny model terenu i inne elementy. Oprócz opisu wyglądu miasta 3D w CityGML zawarte są również właściwości semantyczne i tematyczne, typologia (podział według typów obiektów), hierarchia generalizacji pomiędzy klasami tematycznymi, agregacja, relacje między obiektami a właściwościami przestrzennymi. W CityGML zostało wyodrębnionych 11 grup tematycznych (modułów), których zadaniem jest semantyczne przyporządkowanie wszystkich zapisywanych elementów do odpowiednich obszarów tematycznych: *CityGML Core* (podstawowe obiekty, o charakterze ogólnym, które mogą być wykorzystywane przez inne obiekty np. adres), *Building* (elementy dotyczące budynku), *CityFurniture* (niewielkie obiekty miejskie, takie jak hydranty, ławki, przystanki autobusowe itp.), *CityObjectGroup* (agregacja różnych obiektów występujących w mieście 3D), *GenericCityObject* (rozszerzanie modelu CityGML), *LandUse* (sposoby użytkowania terenu), *Transportation* (ruch drogowy), *Vegetation* (roślinność), *WaterBody* (zbiorniki wodne i wody płynące), *Relief* (numeryczny model terenu), *Appearance* (definiowanie wyglądu powierzchni). Ponieważ CityGML może prezentować teren i obiekty 3D odpowiednio w różnych stopniach dokładności od prostych, nieskalowanych modeli bez topologii i z niewielką semantyką, po bardzo złożone wieloskalowe modele z pełną topologią i szczegółową semantyką, jest on odpowiedni do zastosowania zarówno dla małych obszarów jak i dużych regionów. Podstawowy model rozróżnia 5 poziomów szczegółowości – LOD (Kolbe i inni, 2005).

Ponadto obiekty o typowym wyglądzie mogą posiadać odnośniki do zewnętrznych prototypowych obiektów (*ImplicitGeometries*), dzięki czemu można w znaczny sposób oszczędzić pamięć potrzebną do zapisu zbioru danych. CityGML przewiduje również, że niektóre obiekty są powiązane z zewnętrzną bazą lub plikiem danych, na przykład budynek ma swoją reprezentację w trójwymiarowym modelu oraz w dwuwymiarowej mapie ewidencyjnej. Wówczas cenne jest połączenie modelu z bazą ewidencyjną, szczególnie ze względu na aktualizację danych. Dlatego w CityGML każdy obiekt może być przyporządkowany do obiektu z zewnętrznego zbioru danych za pomocą zewnętrznego odnośnika (*Uniform Resource Identifier; URI*). Dodatkowo CityGML posiada predefiniowane, zewnętrzne listy kodów i słowniki dopuszczalnych wartości atrybutów oraz możliwość rozszerzania już istniejących obiektów i klas lub definicję zupełnie nowych typów obiektów (Iwaszczuk, 2008).

## 2. SCHEMAT APLIKACYJNY GML DLA POTRZEB TBD3D

### 2.1. Budowa i główne założenia schematu aplikacyjnego TBD3DGML

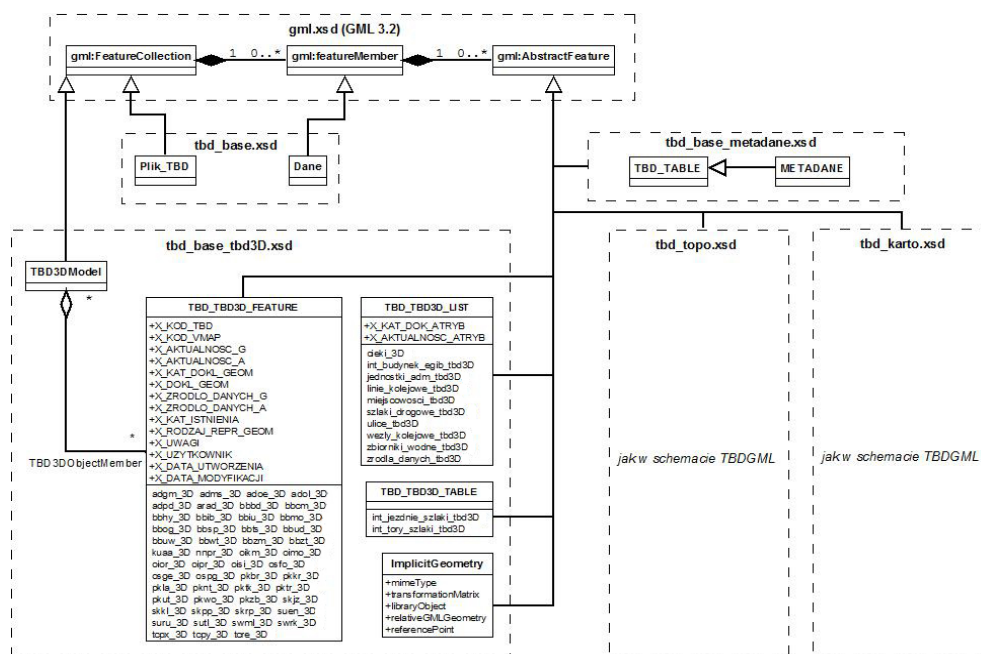
Jednym z etapów badań nad koncepcją TBD3D było opracowanie schematu aplikacyjnego GML dla potrzeb TBD3D (TBD3DGML). Schemat ten pod kątem struktury plików i katalogów oparto na dotychczasowym schemacie dla dwuwymiarowych danych TBDGML. TBD3D ma stanowić dodatkowy komponent TBD do trójwymiarowej wizualizacji danych, dlatego do dotychczasowego schematu TBDGML dołożono dodatkowy schemat *tbd\_tbd3D.xsd* określający dopuszczalne obiekty w TBD3D. Przy czym, by była możliwość definiowania obiektów 3D schemat TBD3DGML oparto na standardzie GML w wersji 3.2 (<http://www.opengis.net/gml/3.2>) oraz 3.1 (<http://www.opengis.net/gml>). Wykorzystanie starszej wersji standardu GML 3.1 było konieczne ze względu na zastosowaną w testach przeglądarkę GML AristotelesViewer v1.3, która nie obsługuje jeszcze plików GML zgodnych z wersją 3.2. Schemat aplikacyjny TBD3DGML, podobnie jak TBDGML, został oparty na specyfikacji World Wide Web z roku 2001 (<http://www.w3.org/2001/XMLSchema>), a przestrzeń nazw schematu została zadeklarowana jako: <http://www.tbd3d.pl/TBD>. Logiczną organizację schematu aplikacyjnego TBD3DGML przedstawiono na diagramie (Rys. 2).

Schemat aplikacyjny TBD3DGML funkcjonalnie odpowiada definicji zbioru danych komponentu TBD3D określonego w koncepcji Bazy Danych Topograficznych w postaci 3D. Opracowany schemat, podobnie jak schematy TBDTOPO i TBDKARTO, został skonstruowany na zasadzie łączenia schematów cząstkowych odzwierciedlających sposób wyodrębnienia poszczególnych plików GML na poziomie definicji zbioru danych.

W schemacie aplikacyjnym TBD3DGML określono następujące schematy podstawowe definiujące dopuszczalne obiekty oraz wartości ich atrybutów (Rys. 2):

- *tbd\_tbd3D.xsd* → określenie dopuszczalnych obiektów w TBD3D (63 schematy cząstkowe),
- *tbd\_base\_tbd3D.xsd* → określenie typów bazowych i podstawowych elementów w TBD3D,
- *tbd\_base\_tbd3D\_texturedSurface.xsd*, *tbd\_base\_tbd3D\_Appearance.xsd* → określenie wyglądu powierzchni w TBD3D,
- *tbd\_base\_sl.xsd*, *tbd\_base\_slx.xsd*, *tbd\_base\_slp.xsd* → określenie dopuszczalnych wartości atrybutów dla obiektów TBD3D,
- *MimeTypeType.xml* → zewnętrzna lista kodów dla atrybutów typu MimeTypeType.

Schematy podstawowe: *tbd\_tbd3D.xsd*, *tbd\_base\_sl.xsd*, *tbd\_base\_slx.xsd* i *tbd\_base\_slp.xsd* zostały oparte na schematach podstawowych TBDGML. Natomiast schematy: *tbd\_base\_tbd3D\_texturedSurface.xsd*, *tbd\_base\_tbd3D\_Appearance.xsd* i *MimeTypeType.xml* zaczerpnięto z CityGML. W schemacie bazowym *tbd\_base\_tbd3D.xsd* ze względu na podobny sposób konstrukcji wybranych klas obiektów, zdefiniowano trzy typy bazowe: TBD\_TBD3D\_FEATURE, TBD\_TBD3D\_LIST, TBD\_TBD3D\_TABLE oraz typ podstawowy TBD3DModelType agregujący wszystkie obiekty w TBD3D. Dodatkowo w schemacie bazowym zdefiniowano typ ImplicitGeometryType za pomocą którego można opisywać symbole 3D. Ten sposób opisywania obiektów został zaczerpnięty ze schematu CityGML i umożliwia opisanie obiektów o typowym wyglądzie za pomocą odnośników do prototypowych obiektów zapisanych w zewnętrznych plikach (np. VRML, X3D). Dzięki temu można w znaczny sposób zaoszczędzić pamięć potrzebną do zapisu zbioru danych.

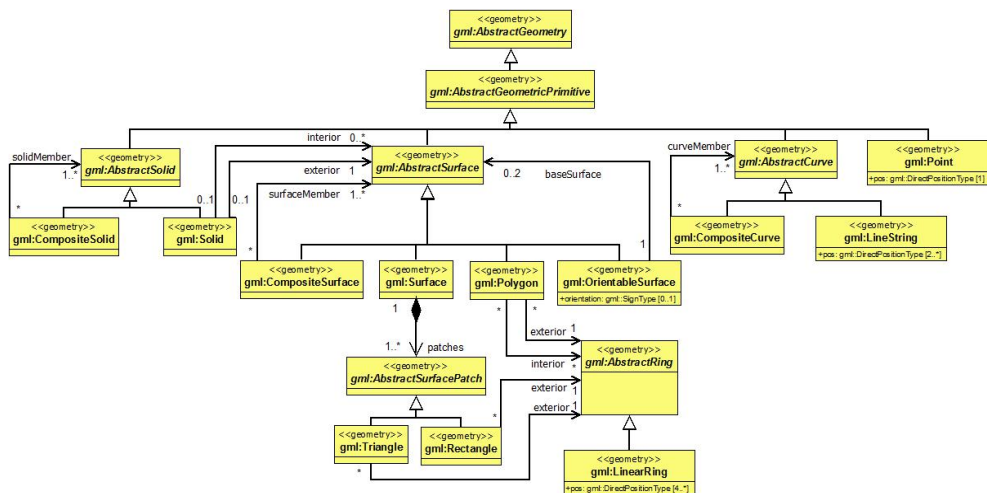


Rys. 2. Logiczna organizacja schematu aplikacyjnego TBD3DGML (opracowanie własne)

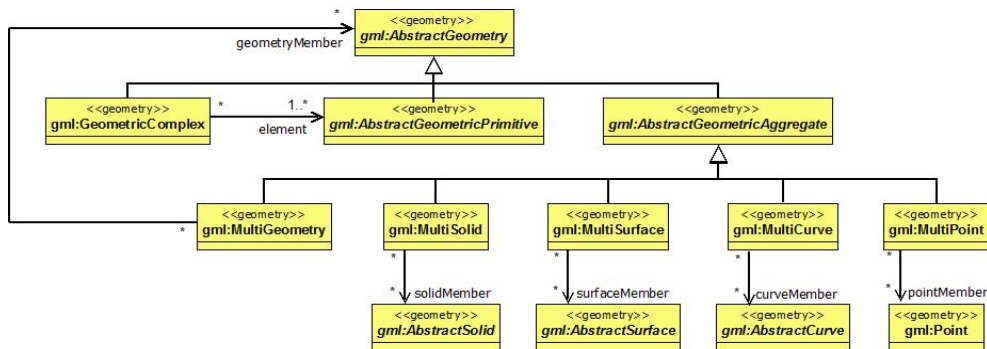
Jako, że standard GML nie definiuje sposobu w jaki należy reprezentować wygląd powierzchni, do tworzenia schematu aplikacyjnego, dla TBD3D wykorzystano z CityGML moduł *Appearance* oraz *TexturedSurface*, który umożliwia przypisanie do powierzchni właściwości fizycznych (kolor, odblaskowość, przezroczystość) oraz tekstur. Definicja właściwości fizycznych w tych modułach została zapożyczona ze specyfikacji standardu wykorzystywanego głównie w grafice komputerowej, eXtensible 3D (X3D), który przyporządkowuje kolor do powierzchni za pomocą trzech wartości zapisanych w systemie RGB wraz z ich odblaskowością oraz przezroczystością. Schemat *Appearance* umożliwia przyporządkowanie dwóch rodzajów tekstur: *ParameterizedTexture* lub *Georeferenced-Texture* oraz właściwości fizycznych powierzchni za pomocą typu *X3DMaterial*. Przyporządkowanie tekstur następuje nie bezpośrednio w obiekcie (*feature*), gdzie zapisana jest jego geometria, ale poprzez referencję elementu *Appearance* do docelowej powierzchni. Dzięki takiemu rozwiązaniu ograniczono redundancję danych w pliku. Inne podejście reprezentuje schemat *TexturedSurface*. Zgodnie z nim każda powierzchnia może zostać określona jako typ *TexturedSurface*, do której można przy użyciu typu *Materials* przypisać kolor, odblaskowość, czy przezroczystość lub jako typ *SimpleTextures* przypisując w ten sposób do powierzchni teksturę. Ponieważ rozwiązanie to powodowało zwiększenie ilości danych w pliku, w aktualnej wersji CityGML (1.0) wycofano się ze stosowania tego schematu, zastępując go modulem *Appearance*. Konieczność definicji dwóch schematów do określania wyglądu powierzchni w TBD3D, podobnie jak opracowanie dwóch wersji schematu aplikacyjnego TBD3DGML zgodnych z wersją GML 3.2 i 3.1, wynikała z niekompatybilności wykorzystywanej w testach przeglądarki GML AristotelesViewer

v.1.3 z modułem *Appearance*. Planowanym docelowym rozwiązaniem dla TBD3D jest schemat oparty na standardzie GML 3.2 i module *Appearance*.

W schemacie aplikacyjnym TBD3DGML geometria poszczególnych obiektów została dostosowana do wymogów trzeciego wymiaru, zgodnie ze standardem GML w wersji 3.2 i 3.1. W trakcie opracowania, wzorując się na modelu geometrycznym CityGML, zostały określone podstawowe elementy (*Primitives*) i kompleksy (*Composites*) geometryczne (Rys. 3) oraz zespoły i agregaty (*Complexes, Aggregates*) geometryczne (Rys. 4).



Rys. 3. Model geometryczny schematu aplikacyjnego TBD3DGML dla podstawowych elementów i kompleksów geometrycznych (opracowanie własne)



Rys. 4. Model geometryczny schematu aplikacyjnego TBD3DGML dla zespołów i agregatów geometrycznych (opracowanie własne)

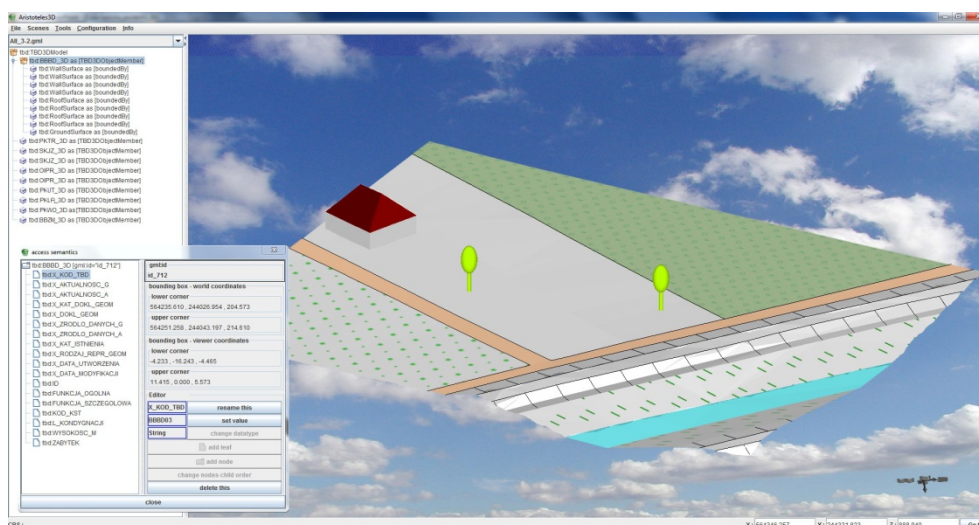
W schemacie aplikacyjnym TBD3DGML obiekty wykorzystują następujące typy geometrii trójwymiarowej określonej w GML: punkt (typ *gml:PointPropertyType*), linia (typ *gml:MultiCurvePropertyType*), powierzchnia (typ *gml:MultiSurfacePropertyType*), bryła (typ *gml:SolidPropertyType*) oraz ogólny typ geometryczny (*gml:GeometryPropertyType*).

W odróżnieniu od schematu aplikacyjnego TBDGML w schemacie TBD3DGML geometrię określono bezpośrednio w schematach cząstkowych definiujących poszczególne obiekty w TBD3D.

Dodatkowo ponieważ TBD3D przewidziana jest jako wielorozdzielcza w zależności od stopnia szczegółowości poszczególne obiekty mogą posiadać różną geometrię. Na przykład budynki w mniej szczegółowym LOD1 (model blokowy) przedstawiane są za pomocą bryły (element: *lod1Bryla*, typ: *gml:SolidPropertyType*) lub w bardziej szczegółowym LOD2 (model geometryczny ze zróżnicowaną strukturą dachów) za pomocą multi-powierzchni (element: *lod2MultiPowierzchnia*, typ: *gml:MultiSurfacePropertyType*) z rozróżnieniem na powierzchnię dachu (*RoofSurface*), ścian (*WallSurface*) i podłogi (*GroundSurface*). Z kolei wybrane obiekty reprezentowane w TBD3D za pomocą symboli 3D w zależności od stopnia szczegółowości będą zapisywane w LOD1 jako np. *lod1MultiPunkt*, a w LOD2 jako trójwymiarowy symbol zdefiniowany w zewnętrznym pliku VRML (element: *lod2Symbol3D*, typ: *ImplicitGeometryType*).

## 2.2. Wykonanie schematu aplikacyjnego TBD3DGML, przeprowadzone testy i kontrole

Do opracowania schematu aplikacyjnego TBD3D (zarówno zgodnego z wersją GML3.1, jak i GML3.2) wykorzystano oprogramowanie MS Visual Studio 2010 Express i Altova XMLSpy. W programie XMLSpy wykonano również walidację opracowanych schematów, czyli sprawdzono ich zgodność syntaktyczną i semantyczną z regułami zapisu XML i standardu GML.



Rys. 5. Wizualizacja obiektów TBD3D w przeglądarce GML (opracowanie własne)

Następnie przeprowadzono testy na wybranych obiektach TBD3D reprezentatywnych dla wszystkich wykorzystanych w schemacie typów geometrycznych: punkt, linia, powierzchnia, bryła, symbol 3D oraz ogólny typ geometryczny. Do testów wybrano obiekty:



budynki (BBBD\_3D), tereny roślinności trawiastej i upraw rolnych (PKTR\_3D), tereny upraw trwałych (PKUT\_3D), odcinki jezdní (SKJZ\_3D), obiekty przyrodnicze (OIPR\_3D), tereny leśne lub zadrzewione (PKLA\_3D), obszary wód (PKWO\_3D) oraz budowle ziemne (BBZM\_3D). Konwersję danych geometrycznych i semantycznych obiektów testowych do formatu GML zgodnego z opracowywanym schematem aplikacyjnym TBD3DGML wykonano w oprogramowaniu Safe Software FME Desktop 2012 (beta).

Kontrolę wygenerowanych danych GML pod kątem zgodności ze schematem aplikacyjnym TBD3DGML wykonano w programie Altova XMLSpy (walidacja plików GML). Wszystkie obiekty testowe zostały poprawnie przekonwertowane do formatu GML zgodnego ze schematem TBD3DGML.

W ostatnim etapie opisywanych badań wykonano wizualizację obiektów testowych zapisanych w pliku GML zgodnych ze schematem aplikacyjnym TBD3DGML w przeglądarce GML AristotelesViewer v.1.3 (Rys. 5).

### 3. WNIOSKI

Podczas opisywanego w niniejszym artykule etapu badań wykonywanych w ramach projektu „Konwersja obiektów Bazy Danych Topograficznych do postaci trójwymiarowej dla potrzeb dynamicznej geowizualizacji” (N N526 192537) stworzono schemat aplikacyjny GML dla potrzeb TBD3D w celu przesyłania danych pomiędzy różnymi systemami geoinformacyjnymi oraz udostępniania ich przez Internet. Schemat aplikacyjny TBD3DGML został sprawdzony pod kątem poprawności semantycznej i syntaktycznej oraz wykonano konwersję obiektów testowych TBD3D do formatu GML zgodnego z opracowanym schematem aplikacyjnym.

Sprawdzono również możliwość wizualizacji obiektów w przeglądarce GML. Podstawowym problemem wizualizacji jest przedstawianie linii, gdyż w schemacie nie da się określić wyglądu (kolor, grubość, styl) obiektów liniowych. Problem wizualizacji linii zapisanych w formacie GML należy rozwiązać po stronie przeglądarek GML, określając sposób wyświetlania obiektów liniowych. W trakcie wykonywanych badań stwierdzono, że obecnie istnieje tylko jedna przeglądarka umożliwiająca wizualizację trójwymiarowych danych zapisanych w standardzie GML3, AristotelesViewer v.1.3. Jest to niekomercyjne narzędzie stworzone głównie do celów naukowych, stąd przeglądarka ta ma kilka wad. Przede wszystkim w AristotelesViewer nawigacja jest wyjątkowo niewygodna, co spowodowane jest prawdopodobnie błędnie zdefiniowaną macierzą obrotu. Brakuje w niej również narzędzi pozwalających na umiejscowienie danego obiektu w pożądanej pozycji i widoku. Jednak podstawowa zaleta tej przeglądarki, czyli charakter oprogramowania otwartego (*Open Source*), pozwala na rozbudowę AristotelesViewer przez użytkownika za pomocą samodzielnie napisanych rozszerzeń (*plugins*). Dzięki temu można dostosować sposób przedstawiania graficznego obiektów TBD3D, tak by było możliwe wyświetlanie nie tylko tekstur, czy właściwości fizycznych przypisanych do powierzchni, ale i również wyświetlanie koloru, stylu i grubości linii.

### 4. LITERATURA

Cisło U., 2008. Zarys koncepcji trójwymiarowej wielorozdzielczej bazy topograficznej. *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, Vol. 18a, s. 49–57.

Gaździcki J., 2007: Internetowy leksykon geomatyczny, [www.ptip.org.pl](http://www.ptip.org.pl).

- Główny Urząd Geodezji i Kartografii, 2008. Wytyczne techniczne Baza Danych Topograficznych (TBD) część 4 „Standardy wymiany danych TBD”. Warszawa.
- Iwaszczuk D., 2008. Modelowanie zabudowy miasta w języku CityGML z uwzględnieniem kompleksowego dokumentowania budynków. Praca magisterska realizowana w Katedrze Geomatyki, Akademia Górniczo-Hutnicza w Krakowie.
- Kmieciak A., 2005: Problematyka modelowania informacji geograficznej w schematach GML, *Roczniki Geomatyki (Annals of Geomatics) 2005* – Tom III, Zeszyt 4, s. 91–98.
- Kolbe T.H., Bacharach S., 2006: CityGML: An Open Standard for 3D City Models, *Directions Magazine*, www.directionsmag.com
- Kolbe T.H., Gröger G., Plümer L., 2005. CityGML – Interoperable Access to 3D City Models, *Geo-information for Disaster Management*, Springer Berlin Heidelberg New York, s. 883–900.
- Open Geospatial Consortium, 2008. OpenGIS® City Geography Markup Language (CityGML) Encoding Standard.

### **GML APPLICATION SCHEMA FOR THREE-DIMENSIONAL MULTIRESOLUTION TOPOGRAPHIC DATABASE**

KEY WORDS: topographic database, multiresolution database, 3D data, GML, GML application schema

SUMMARY: In the research performed in the project "Conversion Topographic Data Base objects to three-dimension for dynamic geo-visualization" (N N526 192537) is developed a concept of 3D Topographic Data Base (TBD3D). TBD3D is intended to be an additional module of existing 2D civil Topographic Data Base (TBD) that allows three-dimensional visualization of topographic data. Conception of three-dimensional multiresolution topographic database (TBD3D) assume that all features in TBD3D exist in three dimensions on three different Levels of Detail (LOD). Features in TBD3D will be presented in two forms: as real generalized features (e.g. buildings) or by 3D symbols. Information about three dimension of real features will be gain from Land and Buildings Cadastre (number of storeys), stereodigitizing, LIDAR data or from superposition of 2D data with digital terrain model (e.g. roads, trains, watercourses, landuse). One of the stages of this research was to develop a GML application schema for TBD3D (TBD3DGML) to transfer data between different geo-information systems and share them over the Internet. The structure of files and directories of this schema is based on the existing schema for 2D topographic data TBDGML. While the geometry and definition of TBD3D objects has been adapted to the requirements and standards of the third dimension and GML3. In addition, in TBD3DGML schema it is possible (as in CityGML) to describe the prototype objects (*ImplicitGeometry*) and define the appearance of the surface by means of *Appearance* or *TexturedSurface* schemas. Developed application schema TBD3DGML was tested at selected TBD3D objects representing all geometry types used in the schema. FME Safe Software was used to write data in the TBD3DGML schema. Then generated data were validated for compliance with the developed schema and loaded into the GML browser AristotelesViewer v.1.3.

mgr inż. Urszula Cisło-Lesicka  
e-mail: cislo@agh.edu.pl  
telefon: +12 6173993