

Urszula Cisko\*

## Standaryzacja zapisu geoinformacji 3D

### 1. Wprowadzenie

Powszechny dostęp do geoinformacji jest uznawany za jeden z elementów kształtujących rozwój społeczeństwa informacyjnego. Wobec dynamicznego rozwoju technologicznego w zakresie pozyskiwania geodanych oraz ich przetwarzania i zdalnego udostępniania, kluczowym zagadnieniem staje się interoperacyjność systemów informacji przestrzennej (określanych powszechnie jako GIS), czyli możliwość wykonania określonych zadań z zakresu przetwarzania lub przesyłania informacji pomiędzy niezależnymi organizacjami w ramach szeroko rozumianego współdziałania.

Warunkiem koniecznym powszechnego wprowadzenia interoperacyjności jest standaryzacja [13]. W systemach informacji przestrzennej standaryzacja dotyczy nie tylko udostępniania danych przestrzennych WMS (*Web Map Server*) czy WFS (*Web Feature Server*), ale i samych danych, czyli elementarnych obiektów, wyróżnień (*features*), z których zbudowany jest model rzeczywistości. Przyjmuje się, że standaryzacja musi dotyczyć atrybutów geoprzestrzennych (geometrycznych i topologicznych), natomiast może pomijać atrybuty tematyczne związane z określoną dziedziną danych geoprzestrzennych [13].

Postęp technologiczny w zakresie pozyskiwania, przetwarzania i przesyłania danych przestrzennych sprawił, że GIS rozwija się w wielu kierunkach, m.in. w kierunku trzeciego wymiaru (GIS 3D) [17]. Stosowana obecnie w GIS struktura danych jest określana jako 2.5D (dane 2D wraz z numerycznym modelem terenu, który uzupełnia trzeci wymiar). Taka struktura, odziedziczona po kartografii (sytuacja + warstwy), ma wiele ograniczeń, które utrudniają lub wręcz uniemożliwiają pełne i dokładne analizy przestrzenne, szczególnie w obszarach miejskich (np. rozchodzenie się zanieczyszczeń, hałasu, modele powodziowe itp.). W GIS 2.5D wysokość ( $H$ ) jest przedstawiona jako funkcja ciągła lub dyskretna współrzędnych płaskich:  $H = f(X, Y)$ , co uniemożliwia poprawne modelowanie obiektów 3D, które dla po-

---

\* Wydział Geodezji Górniczej i Inżynierii Środowiska, Akademia Górniczo-Hutnicza, Kraków

jedynczej pary współrzędnych płaskich ( $X$ ,  $Y$ ) mogą posiadać kilka wartości  $H$  (np. budynek z wystającym dachem) [1]. Dlatego też powstała konieczność opracowania systemu, który miałby możliwość modelowania, reprezentowania, zarządzania, manipulowania i analizowania danymi 3D, czyli złożonymi z punktów o znanych trzech współrzędnych ( $X$ ,  $Y$ ,  $H$ ). Szczególnie istotne stało się to w planowaniu przestrzennym, telefonii komórkowej, turystyce, zarządzaniu nieruchomościami, monitorowaniu środowiska, czy w zarządzaniu kryzysowym.

Celem publikacji jest przybliżenie koncepcji standardów służących do opisywania danych przestrzennych ze szczególnym uwzględnieniem obiektów trójwymiarowych. Kolejne rozdziały omawiają odpowiednio: rozwój standaryzacji informacji przestrzennej, podstawy języka XML, GML, VRML i X3D oraz schematu aplikacyjnego CityGML.

## 2. Standaryzacja informacji przestrzennej

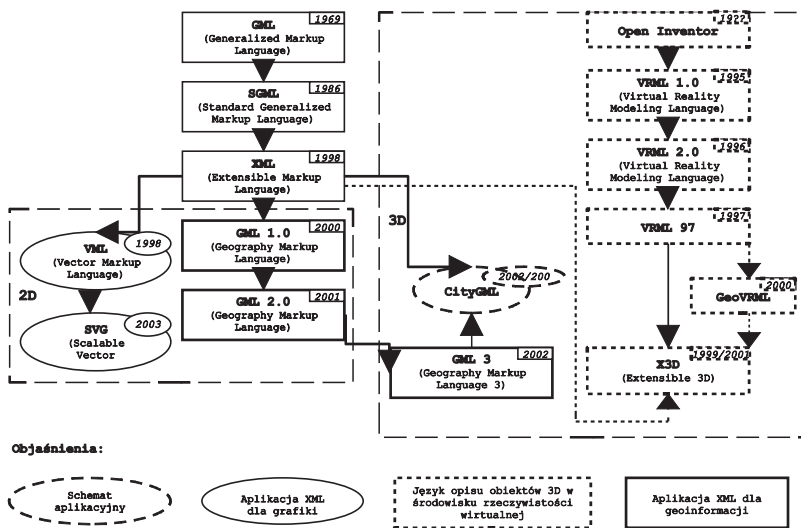
Obecnie opracowaniem standardów w zakresie geoinformacji zajmują się dwa międzynarodowe ośrodki: Open Geospatial Consortium (OGC) [3] i Komitet Techniczny 211 Międzynarodowej Organizacji Normalizacyjnej ISO (ISO/TC 211). Na potrzeby wymiany informacji geograficznej, w celu przesyłania jej pomiędzy różnymi systemami, konsorcjum OGS opracowało język GML (*Geography Markup Language*) (nie należy go utożsamiać z językiem GML, czyli *Generalized Markup Language*, który jest protoplastą późniejszych języków opartych na znacznikach – rys. 1). Najnowsza wersja GML 3.1.1 stanowi podstawę projektu normy ISO/DIS 19136 *Geographic Information – Geography Markup Language*, publikacja dokumentu normatywnego planowana jest na lipiec 2007 roku [6].

GML, bazujący na XML, to język tekstowy oparty na znacznikach (*tags*), przystosowany do przechowywania i transportu danych przestrzennych. Jego właściwości pozwalają na jednoczesne zawarcie w nim zarówno części geometrycznej, jak i opisowej. W standardzie GML określono zarówno reguły semantyczne (znaczenie przypisywane obiektom oraz określanie związków znaczeniowych pomiędzy klasami obiektów), jak i topologiczne (zapis geodanych umożliwiający analizę relacji przestrzennych między obiektami). Standard ten zawiera jedynie informację o treści mapy, a nie jej formie. Aby zwizualizować dane zawarte w GML, należy im dodatkowo przypisać zarówno rodzaj, jak i parametry zmiennych graficznych (np. grubość czy kolor linii) [10]. Często spotykaną formą prezentacji danych GML jest standard SVG (*Scalable Vector Graphics*), wywodzący się z VML (*Vector Markup Language*), jako uniwersalny format dwuwymiarowej grafiki wektorowej.

Z kolei do wizualizacji danych trójwymiarowych często wykorzystywany jest VRML (*Virtual Reality Markup Language*), jego rozszerzenie dla danych geoprze-

strzennych GeoVRML, czy następcą VRML – X3D (*Extensible 3D*) będący standardem ISO dla grafiki 3D generowanej w czasie rzeczywistym (rys. 1). Jednak języki te modelują jedynie geometrię obiektów pomijając semantykę oraz topologię. Dzięki nim można jedynie zwizualizować dane, ale zapytania tematyczne czy analizy przestrzenne są już niedostępne.

Wspomniana wcześniej najnowsza wersja języka GML3 jest dedykowana trójwymiarowemu modelowaniu i dzięki temu umożliwia analizy przestrzenne, których dziedziną są obiekty 3D. Jednakże GML3, jak większość języków znacznikowych wymaga opisu znaczenia i kontekstu danych, czyli tzw. schematu aplikacyjnego. Od 2002 roku pracowali nad tym członkowie Special Interest Group 3D (SIG 3D). Owocem ich prac jest CityGML umożliwiający reprezentację, gromadzenie i wymianę typowych – dla wirtualnych modeli miast – obiektów 3D (rys. 1). Obecnie CityGML posiada status kandydata na otwarty standard OpenGIS (*Candidate OpenGIS® Implementation Specification*) [20].



Rys. 1. Schemat rozwoju standardów

### 3. Język XML

XML to uproszczony międzynarodowy standard GML (*Generalized Markup Language*) (rys. 1), którego zadaniem jest opisanie metody utworzenia języka znacznikowego, definiujący sposób nadawania znaczenia danym poprzez określenie składni i definicji elementów atrybutów tego języka. Zatem XML to nie język do tworzenia dokumentów, ale metajęzyk, czyli język definiujący inne języki [11].

Oto krótka lista cech XML [16]:

- XML może przechowywać i organizować dowolny rodzaj informacji w postaci przystosowanej do potrzeb użytkownika;
- jako standard otwarty, XML nie jest związany z rozwojem żadnej konkretnej firmy ani ograniczony do jakiegos szczególnego oprogramowania czy zastosowania;
- XML umożliwia sprawdzanie jakości dokumentu, ponieważ zawiera reguły poprawności składniowej, wewnętrznego sprawdzania łączy, porównywania z modelami (schematami) dokumentu i kontroli typów danych;
- dzięki jasnej, prostej składni i jednoznacznej strukturze znaczników umieszczonych w zwykłym tekście XML jest łatwy do czytania i analizy zarówno dla ludzi, jak i dla programów.

Najważniejsze jest jednak to, że został tak stworzony, aby każdy mógł go na własny użytek łatwo rozszerzać. Dzięki temu XML może zdefiniować dowolną liczbę innych podrzędnych języków – aplikacji języka XML do zapisu niemal wszystkiego. Trzeba mieć na uwadze, że opracowanie takie powinno zyskać ogólną akceptację środowiska zajmującego się informacją z danej dziedziny – tak, aby to było rozwiązanie standardowe, jak w przypadku języka dla geoinformacji – GML.

Język oparty na XML można stworzyć na dwa sposoby [16]:

- 1) XML o swobodnym formacie (*free-form*),
- 2) modelowanie dokumentu.

Pierwszy sposób to tzw. XML o swobodnym formacie, który zawiera minimalny zestaw reguł dotyczących kształtowania i użytkowania znaczników (można używać dowolnych nazw znaczników, występujących w dowolnej kolejności). Metoda ta niesie ze sobą pewne niebezpieczeństwo opracowania niepoprawnego lub bezużytecznego dokumentu, ponieważ nie ma ograniczeń co do stosowanych znaczników ani żadnych specyfikacji, które byłyby instrukcją użycia danego języka.

Częściej wykorzystuje się drugi sposób, który polega na tworzeniu specyfikacji wyznaczającej reguły opisujące wygląd dokumentu. Dzięki takiemu zapisowi otrzymujemy model logiczny, do którego możemy porównać konkretne dokumenty w celu przetestowania ich pod kątem zgodności ze specyfikacją języka.

Specyfikacje poszczególnych języków mogą być zapisywane na dwa sposoby:

- 1) starszy sposób posługuje się specjalnym językiem definiowania dokumentów DTD (*Document Type Definition*) będących zbiorem reguł albo deklaracji, które określają, jakich znaczników można używać i co mogą zawierać te znaczniki;
- 2) nowszy sposób XML Schema wykorzystuje fragmenty XML nazywane szablonami, aby zademonstrować, jak powinien wyglądać dokument.

Zaletą XML Schema jest to, że sam jest pewną postacią XML, więc można go edytować za pomocą tych samych narzędzi, które używane są do redagowania dokumentów. Ponadto pozwala on także na ściślejszą kontrolę typu danych, co umożliwia wykrywanie błędów nie tylko w znacznikach, ale także w treści dokumentu.

Poniżej przedstawiono podstawowe reguły syntaktyczne języka XML [14]:

- podstawowym fragmentem informacji zapisanej w XML jest „element” ograniczony znacznikiem początku i końca; reprezentuje on porcję informacji: definicje lub dane o obiektach świata rzeczywistego, elementach geometrycznych opisujących właściwości przestrzenne obiektów itd.:

```

<jakiśElement>Wnętrze_jakiegośElementu</jakiśElement>
  ↑                               ↑
znacznik początku                znacznik końca

```

- znacznik początku może zawierać atrybuty służące do zapisu danych atomowych takich jak np. identyfikator czy nazwa:

```
<jakiśElement nazwa="NazwaJakiegośElementu">
```

- element złożony zawiera w sobie inne elementy, proste lub również złożone, co stanowi pewną hierarchię danych, gdzie element główny, który jest elementem najwyższym w hierarchii zawiera dalsze elementy niższego szczebla, zagnieżdżone w ramach elementu głównego:

```

<elementZłożony>
  <elementProsty>
    "coś"
  </elementProsty>
  <innyElementProsty>
    "coś innego"
  </innyElementProsty>
</elementZłożony>

```

## 4. Język GML

Język GML (*Geography Markup Language*) jest aplikacją języka XML przeznaczoną do zapisu geoinformacji w celu przesyłania jej pomiędzy różnymi systemami – *on-line*, niezależnie od platformy sprzętowo-systemowej i niezależnie od charakteru i technologii systemu geoinformacyjnego [3, 21].

GML daje możliwość sformalizowanego opisu aspektów geograficznych występujących w świecie rzeczywistym, takich jak obiekty geograficzne wraz z ich geometrią i zależnościami topologicznymi, systemy odniesienia przestrzennego, czas czy jednostki miary. Podstawowe pojęcia zamodelowane w języku GML zo-

stały pierwotnie zaczerpnięte ze *Specyfikacji abstrakcyjnej OGC* [6], a następnie dostosowane do postanowień norm serii ISO 19100. W tym czasie GML ewoluował koncepcyjnie i metodycznie.

GML nie jest w dosłownym sensie językiem aplikacyjnym, dlatego nie można go użyć wprost do celów praktycznych. GML definiuje jedynie elementy dotyczące aspektu przestrzennego (geometrycznego i topologicznego) geoinformacji. Zatem, aby zastosować go w konkretnych dziedzinach, konieczne jest opracowanie schematów XML zawierających elementy specyficzne dla danej dziedziny.

Najistotniejszym atutem stosowania schematów XML jest sformalizowany zapis informacji, co daje możliwość weryfikacji poprawności nie tylko składni, ale i samych danych. Język GML posiada również inne zalety wynikające z faktu, iż bazuje on na XML.

Należą do nich takie cechy, jak:

- czytelność – do przeglądu czy edycji danych wystarczy zwykły edytor tekstu;
- uniwersalność – można zapisywać dowolne informacje, również nieprzestrzenne;
- elastyczność – łatwość rozszerzania struktury informacji z możliwością wykorzystania wcześniej zdefiniowanych struktur;
- niezależność od platformy sprzętowo-systemowej i od konkretnego systemu GIS;
- niskie koszty w związku z możliwością wykorzystania standardowych narzędzi do obsługi XML (np. sprawdzania jakości dokumentu).

Zasadniczo język GML realizuje ogólny model obiektu *General Feature Model* (ISO, 20C przyjęty w rodzinie norm ISO 19100), który zakłada, że informacja geograficzna stanowi kolekcję obiektów [6]. Każdy obiekt (*feature*) reprezentuje pewną abstrakcję zjawisk występujących w świecie rzeczywistym, a jeżeli jest on dodatkowo powiązany położeniem względem Ziemi, to jest to obiekt geograficzny. Stan każdego obiektu jest opisany przez zbiór właściwości, które mogą być proste lub złożone, mogą mieć charakter nieprzestrzenny lub przestrzenny. Liczbę właściwości danego obiektu, ich nazwy i typy definiuje się w typie tego obiektu. Ponadto obiekty mogą być ze sobą powiązane zależnością silnego typu, tzn. jeden obiekt jest częścią składową drugiego obiektu (zagnieżdżenie obiektów) lub słabą zależnością, w której jeden obiekt odwołuje się do drugiego obiektu. Powiązania te muszą jawnie i jednoznacznie określać relację między typami danego obiektu oraz rolę, jakie każdy z obiektów pełni w związku.

Właściwości przestrzenne obiektów geograficznych opisywane są z wykorzystaniem elementów geometrycznych i topologicznych, które również posiadają typ, właściwości i powiązania.

Poniżej przedstawiono przykład obrysu prostokątnego budynku zapisanego w języku GML wraz z definicją i typem jego atrybutów opisowych [12]:

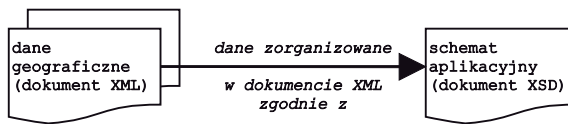
```
<?xml version = "1.0"encoding = "UTF-8"?>
<Vectort_ayer xmlns:gml = "http://www.opengis.net/gml" xmlns:xsi = "http://www.
w3.org/2000/10/XMLSchema-instance" id = "budyunki">
  <gml:name>budyunki – test</gml:name>
  <SchemaDef>
    <AttributeDef name = "ID" type = "integer"/>
    <AttributeDef name = "FUNKCJA" type = "string"/>
    <AttributeDef name = "KONSTRUKCJA" type = "string"/>
    <AttributeDef name = "KUBATURA" type = "double"/>
    <AttributeDef name = "LICZBA_LOKALI" type = "integer"/>
  </SchemaDef>
  <Features>
    <Feature gml:fid = "12000">
      <Attributes>
        <Attribute name = "ID" value = "12000"/>
        <Attribute name = "FUNKCJA" value = "MIESZKALNY"/>
        <Attribute name = "KONSTRUKCJA" value = "MUR"/>
        <Attribute name = "KUBATURA" value = "200"/>
        <Attribute name = "LICZBA_LOKALI" value = "3"/>
      </Attributes>
      <gml::geometryProperty type = "simple">
        <gml:Polygon gml:gId = "12000" gml:srsName = "UTM.U.34">
          <gml:outerBoundaryIs>
            <gml:L1nearRing>
              <gml:coordinates>427547.18,5454456.67 427414.06,5454448.37
                427424.43,5454412.53 427558.43,5454420.37 427547.18,5454456.
                67</gml:coordInates>
            </gml:LinearRing>
          </gml:outerBoundaryIs>
        </gml:Polygon>
      </gml:: geometryProperty>
      <gml:boundedBy>
        <gml:Box>
          <gml:coordinates>427413.10,5454408.92 427559.41 5454460.26</gml:coordinates>
        </gml:Box>
      </gml:boundedBy>
    </Feature>
```

Podobnie jak w XML zapis danych w języku GML jest dwuczęściowy:

- 1) „surowe” dane są zapisywane w dokumentach GML (plik z rozszerzeniem XML, przedstawiony w powyższym przykładzie);
- 2) deklaracyjny opis danych, opisujący ich logiczną organizację w pliku oraz ich znaczenie i kontekst, jest podany w schemacie aplikacyjnym GML (plik z rozszerzeniem XSD, takim schematem jest CityGML).

Zawartość obu plików stanowią tzw. elementy XML (rys. 2) [6].

Schemat aplikacyjny GML określa zakres wymiany, definicje typów (lista atrybutów i zawartości), deklaracje (nazewnictwo elementów) oraz kolejność zagnieżdżeń elementów XML wykorzystywanych w dokumencie GML. Przy czym trzeba pamiętać, iż każdy schemat aplikacyjny zgodny ze specyfikacją języka GML powinien importować tzw. predefiniowane schematy języka GML, które definiują podstawowe struktury danych do zapisu informacji geograficznej oraz spełniać reguły budowy schematów aplikacyjnych podane w specyfikacji języka.



Rys. 2. Dwuczęściowy zapis danych w języku GML  
Źródło: [6]

## 5. Języki do wizualizacji 3D: VRML i X3D

VRML (*Virtual Reality Modeling Language*) – język opisu rzeczywistości wirtualnej [2] – rozwijał się od 1994 r. W 1997 został uznany przez Międzynarodową Organizację Standaryzacji ISO (ISO/IEC 14772-1:1997) jako międzynarodowy standard pod nazwą VRML97 (*Virtual Reality Modeling Language*) (rys. 1) [22].

Składnia VRML oparta jest na obiektach nazywanych węzłami (*nodes*), które składają się z nazwy, pól i zdarzeń (określane odpowiednimi typami) oraz wartości domyślnych dla pól. W specyfikacji języka zdefiniowano szereg typów węzłów, które tworzą struktury hierarchiczne (węzły grupujące), opisując geometrię, własności (oświetlenie, materiały, tekstury) oraz możliwe modyfikacje obiektów i tworząc w ten sposób trójwymiarową scenę (zapisaną w pliku tekstowym ASCII). W języku tym zastosowano czujniki interakcyjne wykrywające akcję użytkownika (np. poruszenie, kliknięcie myszką), co pozwala na detekcje interakcji obiektów geometrycznych, dzięki którym użytkownik może mieć wpływ na wygląd obiektu [18]. Do wizualizacji sceny konieczna jest przeglądarka VR, która jest udostępniana zwykle w formie dodatku (*plug-in*) do przeglądarki internetowej (obecnie wszystkie prze-



glądarki internetowe posiadają taki właśnie *plug-in*) lub samodzielny program, pakiet narzędziowy czy aplet Java dostępne dla każdego systemu operacyjnego.

VRML jest bardzo popularnym narzędziem do tworzenia i udostępniania wirtualnych światów przez Internet ze względu na swoją uniwersalność, dostępność, skalowalność i przede wszystkim dobre zestandaryzowanie. Innymi istotnymi właściwościami języka VRML jest możliwość modyfikacji węzła LOD (*Level of Detail*), decydującego o poziomach szczegółowości umieszczanych obiektów na scenie oraz możliwość zawierania dodatkowego węzła do opisywania obiektów geograficznych. W ten sposób w 2000 roku VRML pozwolił GeoVRML Working Group zrzeszonej w Web3D Consortium stworzyć nowy język – GeoVRML. Jego celem jest umożliwienie prezentacji w Internecie, z wykorzystaniem standardowych dodatków VRML do przeglądarek, danych georeferencyjnych, takich jak mapy czy numeryczne modele terenu.

Innym otwartym standardem dla grafiki 3D generowanej w czasie rzeczywistym jest X3D (*Extensible 3D*) będący rozszerzeniem VRML97 w połączeniu z XML i zawierający w także elementy GeoVRML i NURBS [5]. X3D jest rezultatem kilkuletniej pracy rozwojowej X3D Task Group i Browser Working Group, zrzeszonych w Web 3D Consortium. W 2004 roku został przyjęty jako międzynarodowy standard ISO (ISO/IEC 19775:2004) 3D (X3D) (*Information Technology – Computer Graphics and Image Processing*). Język ten również bazuje na strukturze węzłowej, ale rozwinął nowe możliwości: dodatkowe formaty kodowania danych, ma lepszą zgodność wewnętrzną oraz bardziej rozwiniętą architekturę modułową.

Jego podstawowe zalety to [5]:

- możliwość łatwego sposobu zintegrowania z przeglądarkami Web poprzez XML lub inne aplikacje,
- łączenie możliwości statycznej grafiki dwu- i trójwymiarowej z nawigacją i animacją,
- możliwość wykorzystywania techniki audio oraz wideo,
- umożliwienie połączeń sieciowych i interakcji,
- możliwość rozbudowywania.

Równie ważna jest też jego uniwersalność, dzięki której można w nim tworzyć bannery reklamowe, prezentacje handlowe i biznesowe, dokumentację techniczną i naukową oraz gry sieciowe czy wirtualne światy dostępne dla wielu użytkowników jednocześnie.

## 6. Schemat aplikacyjny – CityGML

Przedstawione w poprzednim rozdziale standardy w większości przypadków, zapewniają jedynie możliwość wizualizacji, pomijając aspekt zapytań tematycznych czy analiz przestrzennych. Dlatego też w 2002 roku członkowie Special Interest Group 3D (SIG 3D) powołanej z inicjatywy GDI NRW (*Geodata Infrastructure*

*North-Rhine Westphalia*) w Niemczech rozpoczęli prace nad stworzeniem otwartego modelu informacji znaczeniowej do reprezentacji trójwymiarowych obiektów miejskich tak, aby mogły one być współużytkowane przez różne aplikacje, pod nazwą CityGML (*City Geography Markup Language*) [7]. Obecnie CityGML w wersji 1.0 ma status kandydata na specyfikację OpenGIS do reprezentacji, przechowywania i wymiany danych wirtualnych miast 3D oraz modeli regionalnych.

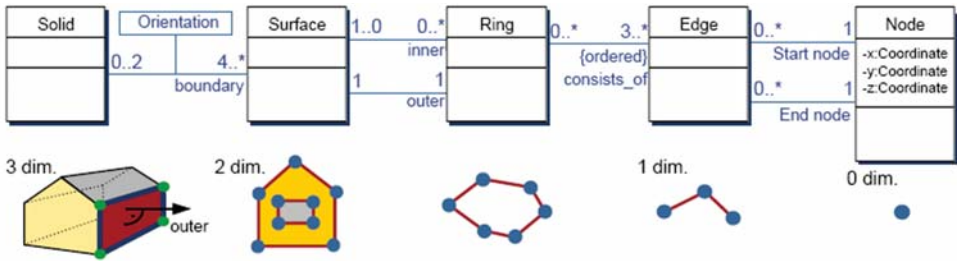
CityGML jest otwartym modelem danych opartym na języku XML. Został on wdrożony jako schemat aplikacyjny języka *Geography Markup Language 3* (GML3) (rys. 1), międzynarodowego standardu wymiany danych przestrzennych. Jak już wspomniano w rozdziale 3, GML definiuje jedynie elementy dotyczące aspektu przestrzennego (geometrycznego i topologicznego) geoinformacji i z tego powodu, aby zastosować go w konkretnych dziedzinach, konieczne jest opracowanie schematów XML zawierających elementy specyficzne dla danej dziedziny. W tym przypadku dziedziną zastosowania jest modelowanie miasta 3D, a schematem aplikacyjnym jest właśnie CityGML. CityGML został oparty na innych standardach (np. ISO 191xx) i był rozwijany przy starannej współpracy z innymi grupami (OGC, W3C Consortium, Web 3D Consortium czy OASIS).

Głównym celem tworzenia CityGML było uzyskanie definicji podstawowych elementów, klas, atrybutów i relacji w modelu miasta 3D i w modelach regionalnych, w odniesieniu do ich właściwości geometrycznych, topologicznych, semantycznych i innych.

Ogólnie „miasto” (*City*) zdefiniowane jest tak, aby zawierało nie tylko konstrukcję budynków, ale również elewacje, roślinność, zbiorniki wodne, elementy infrastruktury (tunele, przejścia podziemne, przystanki autobusowe, latarnie itp.), numeryczny model terenu i inne. Oprócz opisu „wyglądu” miasta 3D w CityGML zawarte są również właściwości semantyczne i tematyczne, typologia (grupowanie, podział według typów), hierarchia generalizacji pomiędzy klasami tematycznymi, agregacja, relacje między obiektami a właściwościami przestrzennymi. CityGML jest odpowiedni do zastosowania zarówno dla dużych obszarów, jak i małych regionów. 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. Podstawowy model rozróżnia 5 (0–4) kolejnych poziomów szczegółowości – LoD (*Level of Detail*) [9].

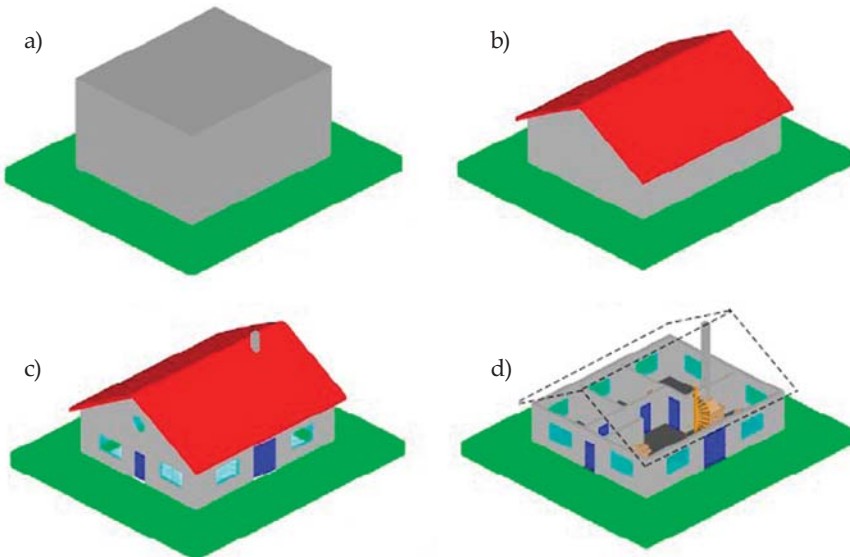
Właściwości przestrzenne obiektów CityGML określone są przez model geometryczno-topologiczny zgodny z powszechnie znanym modelem *Boundary Representation* (BRep) [4]. Dla każdego wymiaru istnieje odpowiedni geometryczno-topologiczny prymityw (*primitive*): zerowymiarowy obiekt to węzeł (*node*), jednowymiarowa jest krawędź (*edge*), dwuwymiarowa ściana (*face*) i trójwymiarowa bryła (*solid*). Bryła ograniczona jest przez ściany, ściany – przez krawędzie, a kra-

węździe – przez węzły, przy czym krawędź musi być linią prostą, a ściana musi być płaska (rys. 3). Ważne jest również to, by wnętrza prymitywów były rozłączne i jeśli dwa prymitywy się dotykają, to wspólna granica musi być prymitywem niższego wymiaru. Dzięki temu ograniczeniu została zapewniona przejrzysta topologia bez żadnej nadmiarowości.



Rys. 3. Model geometryczno-topologiczny CityGML  
Źródło: [8]

Rdzeniem CityGML jest model budynku, który pozwala na reprezentację tematyczną i przestrzenną budynków, części budynków oraz ich elementów w czterech poziomach szczegółowości (LoD1 – LoD4) (rys. 4).



Rys. 4. Model budynku: a) LoD1; b) LoD2; c) LoD3; d) LoD4  
Źródło: [15]

Na poziomie LoD1 budynek określony jest poprzez prostą bryłę i może stanowić część kompleksu budynków. W poziomie LoD2 jest możliwe już wyróżnienie powierzchni ograniczających bryłę budynku jako obiektów tematycznych, które mogą być zaklasyfikowane jako dach, ściana czy podłoga. Na tym poziomie budynek może zawierać dodatkowe elementy typu: komin, balkon czy schody zewnętrzne.

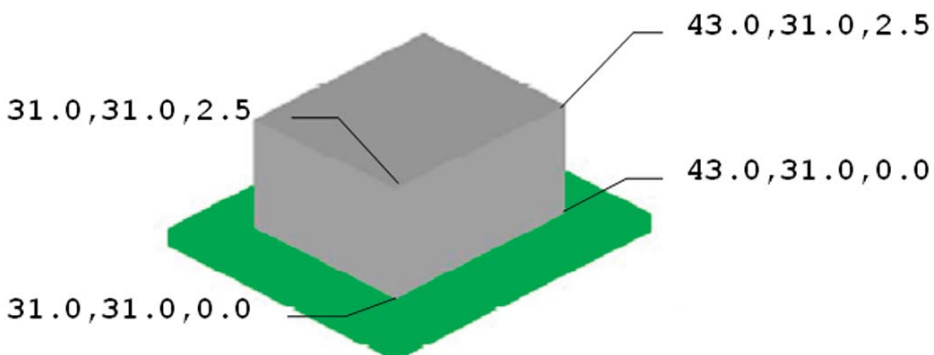
W LoD3 budynki są znacznie bardziej szczegółowe niż w poziomie LoD2 i mogą również posiadać „otwory”, takie jak okna czy drzwi. Natomiast LoD4 jest rozwinięciem LoD3 o wewnętrzne struktury takie jak pokoje (ograniczone przez podłogi i ściany wewnętrzne), filary, schody czy wyposażenie (meble).

Istotną sprawą jest również integracja budynku z terenem. CityGML stosuje przy budynku krzywą łamaną, która – przedstawiona za pomocą zamkniętej obryczy (*ring*, rys. 3) otaczającej budynek – wskazuje dokładne miejsce kontaktu terenu z budynkiem. Zastosowanie tej krzywej łamanej zapewnia również prawidłowe położenie tekstury.

Natomiast model terenu może być określany za pomocą siatki prostokątnej, nieregularnej siatki TIN (*Triangulated Irregular Network*), linii nieciągłości terenu (*breaklines*) i linii szkieletowych (*skeleton lines*) czy chmury punktów 3D (*mass point*). Ponadto przewidziano modele hybrydowe złożone z dwóch i więcej wymienionych typów, np. możliwe jest modelowanie terenu w części obszaru zgrubną siatką prostokątną, a w bardziej interesującym nas obszarze – szczegółowym siatką TIN.

CityGML umożliwia przypisanie do obiektów wybranych tekstur czy materiału. Ponadto obiekty mogą posiadać odnośniki do odpowiednich obiektów w zewnętrznych zbiorach danych. Dodatkowo CityGML posiada predefiniowane, zewnętrzne listy kodów i słowniki dopuszczalnych wartości atrybutów.

Na rysunku 5 oraz na s. 101 przedstawiono przykład zapisu budynku w poziomie LoD1 [15].



Rys. 5. Wizualizacja przykładowego zbioru danych w poziomie LoD1

Źródło: [15]

```

<?xml version="1.0" encoding="UTF-8"?>
<CityModel xmlns=http://www.citygml.org/citygml/1/0/0 xmlns:gml=http://www.opengis.
net/gml
  xmlns:xAL="urn:oasis:names:tc:ciq:xdschema:xAL:2.0" xmlns:xlink="http://
www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.citygml.org/citygml/1/0/0 ../CityGML.xsd">
  <gml:description> Prosty przykład danych XML zgodnych z CityGML, schematem
aplikacyjnym GML opracowanym przez SIG 3D. Źródło danych www.citygml.org
  </gml:description>
  <gml:name>3D city model of Samplecity</gml:name>
  <gml:boundedBy>
  <gml:Envelope srsName="urn:adv:crs: ETRS89_3GK2-h">
    <gml:pos srsDimension="3">>0.0 0.0 0.0 </gml:pos>
    <gml:pos srsDimension="3">>43.0 34.0 3.5</gml:pos>
  </gml:Envelope>
</gml:boundedBy>
<cityObjectMember>
<!-- Prosty budynek na poziomie szczegółowości LoD1 -->
  <Building gml:id="Build0816">
    <gml:name>Villa Kunterbunt</gml:name>
    <function>1000</function>
    <yearOfConstruction>1952</yearOfConstruction>
    <roofType>1030</roofType>
    <lod1Solid>
      <gml:Solid srsName="urn:adv:crs: ETRS89_3GK2-h">
        <!-- prosty model blokowy -->
        <gml:exterior>
          <gml:CompositeSurface>
            <gml:surfaceMember>
              <!-- frontowa ściana -->
              <gml:Polygon>
                <gml:exterior>
                  <gml:LinearRing>
                    <gml:pos srsDimension="3">31.0 31.0 0.0</gml:pos>
                    <gml:pos srsDimension="3">43.0 31.0 0.0</gml:pos>
                    <gml:pos srsDimension="3">43.0 31.0 2.5</gml:pos>
                    <gml:pos srsDimension="3">31.0 31.0 2.5</gml:pos>
                    <gml:pos srsDimension="3">31.0 31.0 0.0</gml:pos>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </gml:surfaceMember>
            .....
          </gml:CompositeSurface>
        </gml:exterior>
      </lod1Solid>
    </Building>
  </cityObjectMember>

```

} Dla uproszczenia współrzędne zosta-  
ły skrócone, w oryginale są podane  
w określonym układzie współrzęd-  
nych prostokątnych płaskich

## 7. Podsumowanie

W artykule uzasadniono konieczność stworzenia standardu zapisu danych geograficznych niezależnego sprzętowo i narzędziowo, jakim jest *Geography Markup Language*. Odniesiono się także do popularnych w zastosowaniach internetowych standardów grafiki trójwymiarowej VRML, GeoVRML i X3D, wskazując na ich ograniczenia. Pomijają one aspekt zapytań tematycznych i analiz przestrzennych. Dlatego też powstała idea rozwinięcia GML, a raczej stworzenia dla niego schematu aplikacyjnego – CityGML.

CityGML definiuje klasy i relacje dla większości istotnych obiektów topograficznych w miastach i modelach regionalnych w odniesieniu do ich geometrycznych, topologicznych, semantycznych i zewnętrznych właściwości. Daje to możliwość wnikliwych analiz tematycznych i przestrzennych. CityGML definiuje nie tylko konstrukcję budynków, ale również elewacje, roślinność, zbiorniki wodne, elementy infrastruktury (tunele, przejścia podziemne, przystanki autobusowe, latarnie itp.) czy numeryczny model terenu. Dzięki możliwości obsługi różnych poziomów szczegółowości (LoD) jest on odpowiedni do wykorzystania zarówno dla małych jak i dużych obszarów. Obecnie schemat ten jest w fazie testowania i opiniowania przez Open Geospatial Consortium.

W Polsce, od marca 2005 roku, trwają prace nad projektem polskiej normy EN-ISO 19136 *Informacja geograficzna – Geography Markup Language* – będącej odpowiednikiem międzynarodowej normy ISO/DIS 19136 *Geographic information – Geography Markup Language*. W założeniach koncepcyjnych bazy danych topograficznych (TBD) jako standard wymiany danych wskazano na GML. Stało się tak po okresie testów, w których poziom interoperacyjności sprawdzanych standardów, *de facto* preferowanych przez producentów oprogramowania GIS, okazał się niezadowolający.

CityGML, jako kandydat na otwarty standard modelowania wirtualnych miast 3D, również zasługuje na szczególne zainteresowanie szerokiej rzeszy użytkowników korzystających z geoprzestrzennych danych, a w szczególności zajmujących się planowaniem przestrzennym, turystyką, zarządzaniem kryzysowym, czy telefonią komórkową. W Polsce można zaobserwować wzrost zapotrzebowania na dane 3D. Przykładem może być trójwymiarowy model Wrocławia, opracowany na zlecenie Urzędu Miasta Wrocławia w 2006 r [19]. Obecnie opracowuje się podobny model dla miasta Krakowa.

## Literatura

- [1] Apel M.: *From 3d geomodelling systems towards 3d geoscience information systems: Data model, query functionality, and data management*. Computers & Geosciences, vol. 32, 2006.
- [2] Dąbkowski K.: *VRML97 Trzeci wymiar sieci*. ZNI „MIKOM”, Warszawa 1998.

- [3] ISO/TC 211/WG 4/PT 19136 *Geographic information – Geography Markup Language (GML)*. 2004.
- [4] Foley J.D.: *Wprowadzenie do grafiki komputerowej*. Wydawnictwo Naukowo-Techniczne, Warszawa 2001.
- [5] Jędrzycka R.: *Nowe techniki wymiany i gromadzenia informacji a dane fotogrametryczne*. Archiwum Fotogrametrii, Kartografii i Teledetekcji, vol. 13, 2003.
- [6] Kmiecik A.: *Problematyka modelowania informacji geograficznej w schematach GML*. PTIP Roczniki Geomatyki, t. 3, z. 4, 2005.
- [7] Kolbe T.H., Bacharach S.: *CityGML: An Open Standard for 3D City Models*. Directions Magazine, 2006.
- [8] Kolbe T.H., Gröger G.: *Towards unified 3D City Models*, ISPRS Comm. IV Workshop „Challenges in Geospatial Analysis, Integration and Visualization II”, Stuttgart 2003.
- [9] Kolbe T.H., Gröger G., Plümer L.: *CityGML – Interoperable Access to 3D City Models*. 1st International Symposium on Geo-information for Disaster Management, Delft 2005.
- [10] Lake R.: *Introduction to GML Geography Markup Language*. WAP Forum – W3C 2000.
- [11] Liberty J., Kralej M.: *XML od podstaw*. Translator, Warszawa 2000.
- [12] Litwin L., Myrda G.: *Systemy Informacji Geograficznej. Zarządzanie danymi przestrzennymi w GIS, SIP, SIT, LIS*. Helion, Gliwice 2005.
- [13] Michalak J.: *Interoperacyjność w zakresie informacji geoprzestrzennej*. XII Konferencja Naukowo-Techniczna „Systemy Informacji Przestrzennej”, Warszawa 2002.
- [14] Michalak J.: *Podstawy metodyczne i technologiczne infrastruktur geoinformacyjnych*, PTIP Roczniki Geomatyki, t. 1, z. 2, 2003.
- [15] Open Geospatial Consortium: *Candidate OpenGIS CityGML Implementation Specification (City Geography Markup Language)*. 2006.
- [16] Ray E.T.: *Nauka języka XML*. ReadMe, Warszawa 2001.
- [17] Stoter J., Zlatanova S.: *3D GIS where are we standing?* Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis, Quebec 2003.
- [18] Weiss Z., Konieczny R., Kasica M., Kowalski M.: *Interaktywne projektowanie z zastosowaniem narzędzi VRML*. III Forum ProCAX Mechanik Nr 11/2004.
- [19] *Wrocław to dachy!*, Geodeta, nr 1 (140), 2007 (wywiad z S. Dunin-Wąsowiczem i R. Rutkowskim).

- [20] Homepage of CityGML. Źródło internetowe: <http://www.citygml.org>.
- [21] Polskie Towarzystwo Informatyki Przemysłowej: [www.ptip.org.pl](http://www.ptip.org.pl).
- [22] World Wide Web Consortium. Źródło internetowe: <http://www.w3.org>.