

## Implementacja GML w praktyce

**Streszczenie:** Ostatnio w różnych dziedzinach życia mamy do czynienia z językiem XML, a właściwie z jego specjalistycznymi aplikacjami dedykowanymi do różnych zastosowań. W stosunku do danych przestrzennych taką aplikacją jest język GML. W pierwotnych zamierzeniach GML miał służyć do wymiany danych przestrzennych między różnymi systemami. Praktyka wybiega jednak poza pierwotne założenia i GML zaczyna być wszechobecny w naszym życiu. W ostatnim czasie jest używany również do opisywania struktur danych w systemach oraz jego duże wstawki pojawiają się w aktach prawnych, wprowadzając tam niemałe zamieszanie. Artykuł niniejszy prezentuje w krótki i przystępny sposób podstawowe informacje dotyczące GML oraz przedstawia praktyczne aspekty jego stosowania na bazie wiedzy ogólnej i doświadczeń autora w konkretnych realizacjach.

### 1. Podstawowe informacje o języku XML

XML (ang. *eXtensible Markup Language*) jak nazwa wskazuje jest rozszerzalnym językiem znaczników. Zaprojektowanym został przez World Wide Web Consortium (<http://www.w3.org>). XML jest narzędziem ogólnym, przeznaczonym do tworzenia innych specjalistycznych języków, które nazywamy aplikacjami XML. Powstało już wiele różnych aplikacji XML do różnych zastosowań np.:

<b>CML</b>	Chemical Markup Language (Język znaczników chemicznych)	format zapisu różnorodnych informacji o związkach chemicznych i ich reakcjach
<b>MathML</b>	Mathematical Markup Language (Język znaczników matematycznych)	język będący zastosowaniem XML-a, służący do opisywania wzorów i symboli matematycznych
...	...	...
<b>GML</b>	Geography Markup Language (Język Znaczników Geograficznych)	aplikacja XML do opisu danych przestrzennych

Język XML oparty jest na znacznikach, które umieszczane są między znakami < > np. <osoba>. Istotne jest, że każdy znacznik otwierający np. <osoba> musi posiadać znacznik zamykający </osoba>. Należy pamiętać, że w nazwie znacznika nie mogą występować spacje, wielkość liter jest rozróżniana, a znaczniki muszą być prawidłowo w sobie zagnieżdżone. Między znacznikiem otwierającym i zamykającym zawierają się poszczególne elementy opisywane przez XML. Elementy mogą dodatkowo posiadać atrybuty, których wartości umieszcza się w cudzysłowie.

```
<pracownicy>
  <pracownik>
    <imie>Waldemar</imie>
    <nazwisko>Izdebski</nazwisko>
    <uczelnia>Politechnika Warszawska</uczelnia>
  </pracownik>
  <pracownik>
    <imie>Jan</imie>
    <nazwisko>Kowalski</nazwisko>
    <uczelnia>Politechnika Warszawska</uczelnia>
  </pracownik>
</pracownicy>
```

Reguły dotyczące plików XML są ściśle i nienaruszalne. Źle zapisany znacznik, zgubiony nawias czy atrybut nie ujęty w cudzysłów czyni plik XML wadliwym. Oficjalna specyfikacja języka XML zabrania aplikacjom domyślać się co ma znaczyć dany fragment uszkodzonego pliku XML. Jeśli w pliku jest błąd, program powinien wstrzymać wykonywanie i zgłosić zaistniały problem.

Z powodu zastosowania postaci tekstowej i używania znaczników do rozgraniczenia danych, pliki XML są znacząco większe niż inne pliki, z takimi samymi danymi. Jest to jednak świadoma decyzja projektantów języka XML, gdyż postać tekstowa, w razie konieczności, daje możliwość bezproblemowego odczytania danych, a wady płynące z wielkości plików mogą być łatwo skompensowane przez ich kompresję.

## 2. Język GML jako aplikacja XML

GML (ang. *Geography Markup Language*) czyli język znaczników geograficznych, który definiuje zbiór elementów przestrzennych (geograficznych) zapisanych w formacie języka XML. Do opisu obiektów przestrzennych wykorzystuje się schematy XML, w których zdefiniowano podstawowe właściwości zapisywanych obiektów. Schematy przechowywane są w plikach z rozszerzeniem XSD (ang. *XML Schema Definition*). Opis ważniejszych schematów występujących w wersji 3.1 specyfikacji języka GML przedstawiono poniżej.

1	<b>basicTypes.xsd</b> <b>gmlBase.xsd</b>	Definicje podstawowych typów danych i atrybutów
2	<b>coordinateOperations.xsd</b> <b>coordinateSystems.xsd</b> <b>coordinateReferenceSystems.xsd</b> <b>datums.xsd</b> <b>referenceSystems.xsd</b> <b>temporalReferenceSystems.xsd</b>	Definicje układów odniesienia, definicje typów danych z tym związanych
3	<b>geometryBasic0d1d.xsd</b> <b>geometryBasic2d.xsd</b> <b>geometryPrimitives.xsd</b>	Definicje podstawowych typy obiektów geometrycznych: punkt, linia, poligon
4	<b>geometryComplexes.xsd</b> <b>geometryAggregates.xsd</b>	Definicje złożonych obiektów geometrycznych
5	<b>dictionary.xsd</b>	Definicja słownika
6	<b>defaultStyle.xsd</b>	Definicje obiektów pozwalających na określanie stylu graficznej prezentacji obiektów.
7	<b>gml.xsd</b>	Podstawowy schemat łączący pozostałe

Restrykcyjne przestrzeganie reguł w programach przetwarzających pliki oparte o XML, powodują, że należy zwracać szczególną uwagę na poprawność schematów, aby później nie powielać błędów w generowanych na ich podstawie plikach GML.

## 3. Doświadczenia w korzystaniu z języka GML w wymianie danych

Jak już wspomniano wcześniej, pierwotnie GML był pomyślany jako język do wymiany danych między różnymi systemami. Mówiąc o wymianie danych należy pamiętać, że nie ma dwóch systemów o identycznej strukturze danych. Zawsze więc jeśli dane z systemu **X** zapiszemy w jakimś formacie wymiany odzwierciedlającym jego strukturę danych, to do systemu **Y**, który będzie posiadał inną strukturę trafi tylko pewien podzbiór informacji z systemu **X**. Pewne wartości atrybutów, które nie będą występowały w transferze, ze względu na inny model mogą co najwyżej być uzupełnione wartościami domyślnymi lub pustymi. Tak więc żaden format transferu nie poprawi jakości danych i dotyczy to również standardu GML, chociaż czasami jest on przedstawiany jako narzędzie rozwiązujące wszystkie problemy związane z danymi przestrzennymi.

### 3.1. Doświadczenia z wykorzystaniem GML do zapisu danych mapy zasadniczej

Nasze prace związane wykorzystaniem GML do zapisu danych mapy zasadniczej rozpoczęły się w roku 2010 od prac wykonanych w ramach projektu „*Wypracowanie i wdrożenie innowacyjnych metod integracji danych katastralnych, mapy zasadniczej i Bazy Danych Topograficznych oraz modernizacja usług publicznych świadczonych przez Służbę Geodezyjną i Kartograficzną*” realizowanego przez Samorząd Województwa Mazowieckiego we współpracy z Głównym Geodetą Kraju oraz samorządami powiatu piaseczyńskiego oraz miasta Płocka.

W ramach naszych prac przygotowaliśmy dane w postaci **GML** na podstawie baz danych systemu **GEO-MAP** oraz przygotowanego przez **BGWM** modelu danych. Przedstawiony model posiadał niestety wiele wad, o których istnieniu na początku nikt nie chciał słyszeć. Wydawało się, że jakoś to będzie. Można przypuszczać, że model nie był sprawdzany żadnymi eksperymentami polegającymi chociażby na zapisie pojedynczych obiektów. Bazował jedynie na wewnętrznej spójności formalnej i zaufaniu do jego twórców. Dopiero kiedy firma **COMARCH S.A.**, wykonująca aplikację na jego bazie, i my podczas przygotowywania danych wykazaliśmy jego niekwestionowane wady, rozpoczął się proces naprawy.

Na podstawie poczynionych analiz podczas realizacji prac do podstawowych wad modelu należy zaliczyć:

1. Brak możliwości przenoszenia elementów redakcyjnych związanych z danymi, co w przypadku mapy zasadniczej jest bardzo ważne. Firmy przygotowujące dane poświęcały redakcji wiele czasu, a potem nie było mechanizmu do jej przekazania.
2. Opracowany model był mało czytelny aczkolwiek potężny objętościowo, bo wszystkie dokumenty liczyły razem ok. 1000 stron. Niestety w dużej mierze obszerność wynikała z powtarzania wielu rozdziałów, co prowadziło również do ich niespójności. Przy powielonych rozdziałach wielokrotnie autorzy wprowadzając poprawki w jednym z rozdziałów nie zadawali sobie sprawy, że ten rozdział był powielony w innym dokumencie.
3. Występowały drobne aczkolwiek, z punktu widzenia języka GML, istotne, rozbieżności w schematach używanych do walidacji ze schematami udostępnionymi jako wzorcowe. Powodowało to, że utworzone na bazie wzorcowych schematów pliki nie przechodziły procesu walidacji. W tym wypadku chodziło głównie o literówki i inne drobne błędy.
4. Funkcjonowanie modelu jedynie w sferze teoretycznej spowodowała, że pojawiły się trudności z zapisem informacji o wielu obiektach, których reprezentacja i sposób pomiaru jest ugruntowany w praktyce geodezyjnej. W szczególności problem dotyczył wymagalności wielu atrybutów, które niestety nie zostały pozyskane na etapie pomiaru, a były konieczne na etapie transferu. W takich przypadkach obiekty musiały być albo pomijane albo zapisywane z atrybutami zawierającymi fałszywe dane. Przykładem mogą być budynki, z których żaden nie posiadał wszystkich danych niezbędnych do prawidłowego zapisania go w modelu. W zamieszczonym poniżej przykładzie przedstawiono fikcyjne uzupełnienie atrybutów m.in.: data wyceny, wartość, tak aby dane można było zapisać zgodnie z przyjętym schematem.

```
<egb:Budynek>
.....
  <omg:funOgolnaBudynku>budynekHandlowyUslugowy</omg:funOgolnaBudynku>
  <omg:funSzczegolowaBudynku>pawiloHandlowoUslugowy</omg:funSzczegolowaBudynku>
  <omg:klasyfikacjaPKOB>budynkiHandlowoUslugowe</omg:klasyfikacjaPKOB>
  <omg:liczbaKondygnacjiNadziemnych>0</omg:liczbaKondygnacjiNadziemnych>
  <egb:liczbaKondygnacjiPodziemnych>0</egb:liczbaKondygnacjiPodziemnych>
  <omg:nrRejestruZabytkow/>
  <egb:dataWyceny>1900-01-01</egb:dataWyceny>
  <egb:liczbaLokaliInnych>0</egb:liczbaLokaliInnych>
  <egb:liczbaLokaliOdrebnych>0</egb:liczbaLokaliOdrebnych>
  <egb:materialScianZewn>mur</egb:materialScianZewn>
  <egb:nrLokaliInnych/><egb:nrLokaliOdrebnych/>
  <egb:powPomieszczenPrzynal uom="m2">0</egb:powPomieszczenPrzynal>
  <egb:powUzytkowaBudynku uom="m2">0</egb:powUzytkowaBudynku>
  <egb:powUzytkowaLokali uom="m2">0</egb:powUzytkowaLokali>
  <egb:powZabudowy uom="m2">0</egb:powZabudowy>
  <egb:rokZakonczeniaBudowy>1972</egb:rokZakonczeniaBudowy>
  <egb:wartosc uom="PLN">0</egb:wartosc>
  <egb:lokalizacja/><egb:jednostkaRejestrowa/>
.....
</egb:Budynek>
```

5. W wielu przypadkach występowały problemy z zapisem geometrii. Obiektom w modelu często przypisano rodzaj reprezentacji geometrycznej mocno odbiegający od rzeczywistości. Jako przykład przedstawiamy jezdnię, która wg definicji mogła być ograniczona maksymalnie dwoma krawężnikami (**maxOccurs=2**)

```
<element name="Jezdnia" type="otmz:JezdniaType" substitutionGroup="omg:SiecTransportowa"/>
<complexType name="JezdniaType">
  <complexContent>
    <extension base="omg:SiecTransportowaType">
      <sequence>
        <element name="idUlicy" type="bdot:UlicaType"/>
        <element name="klasaDrogi" type="omg:KlasaDrogiType"/>
        <element name="nrDrogi" type="string"/>
        <element name="rodzajNawierzchni" type="bdot:RodzajNawierzchniType"/>
        <element name="ulica" type="bdot:CzyUlicaType"/>
        <element name="krawedzJezdni" type="otmz:KraweznikPropertyType" minOccurs="0" maxOccurs="2"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

W schematach przygotowanych na potrzeby kolejnego projektu BGWM „Wsparcie prac związanych z przekształceniem ewidencji gruntów i budynków (EGiB) do postaci cyfrowej, uzupełnienie danych dotyczących budynków oraz modernizacja tego rejestru, przekształcenie mapy zasadniczej do postaci numerycznej, prace związane z przekształceniem do postaci cyfrowej innych danych przestrzennych, skanowanie dokumentów uzasadniających wpisy w EGiB” wyeliminowano lub raczej ograniczono część z opisywanych wcześniej błędów. Polegało to przeważnie na dopuszczeniu pomijania części atrybutów lub ustawiania ich jako pustych. W przypadku danych generowanych dla powiatu garwolińskiego (projekt z 2012 roku) najpoważniejszy błąd, uniemożliwiający zapisanie jakichkolwiek danych, występował w schemacie odnoszącym się do osnowy geodezyjnej. Dla każdego z punktów osnowy należało wskazać obserwacje wykonane przy ustalaniu współrzędnych tego punktu, w tym parametry osnowy wykorzystywane przy pomiarach. Z kolei żeby zapisać osnowę wykorzystywaną do pomiarów należało wskazać, tak jak wcześniej, wszystkie obserwacje i kolejne punkty, a dla nich kolejne obserwacje i kolejne punkty itd. Problem ten jasno pokazał, że podobnie jak poprzednio nie były wykonywane żadne testy praktyczne.

Zapewne podobne problemy czekają nas już wkrótce, bo zaczyna obowiązywać rozporządzenie Ministra Administracji i Cyfryzacji w sprawie „bazy danych geodezyjnej ewidencji sieci uzbrojenia terenu, bazy danych obiektów topograficznych oraz mapy zasadniczej”, które również oparte jest na czysto teoretycznym modelu danych, bez jakichkolwiek testów i niesie wiele problemów przy wymianie danych. Wprawdzie wyciągnięto wnioski z błędów we wcześniejszych dwóch modelach danych, ale wiele problemów niestety pozostaje i ujawni się na etapie zastosowań praktycznych. Uważam, że w stosunku do tego, jak i wielu wydanych ostatnio przez GUGiK, rozporządzenia błędem jest powielanie w każdym z nich schematu modelu podstawowego. Oprócz zaciemniania istoty poszczególnych rozporządzeń w przyszłości może powodować wiele problemów przy modyfikacji przepisów. Istnienie definicji (opisu) „**modelu podstawowego**” jako samodzielnego tworu np. rozporządzenia byłoby w tej kwestii najlepszym rozwiązaniem, bo do niego można wtedy odnosić wszystkie inne modele szczegółowe dotyczące danych przestrzennych.

### 3.2. Doświadczenia z rozporządzeniem w sprawie EMUiA

W przypadku EMUiA nasze doświadczenia rozpoczęły się w roku 2012 i bazują na prawie 500 wdrożeniach w polskich gminach. Problemy w tej materii są mniejsze niż opisywane poprzednio, ale również występują i standardowo dotyczą przede wszystkim nieprzystosowania modelu danych do rzeczywistości. W tym wypadku zwracamy uwagę na obiekt „AD\_Ulica”, który zgodnie z zapisami rozporządzenia może być zapisane jedynie jako linia.

<b>Klasa: AD_Ulica</b>	
<i>Nazwa:</i>	geometria
<i>Nazwa (pełna):</i>	geometria
<i>Dziedzina:</i>	GM_Object
<i>Liczność:</i>	1
<i>Definicja:</i>	Dla obiektu typu ulica jest to łamana wyznaczona na podstawie danych ewidencji gruntów, BDOT500 lub mapy zasadniczej przez osie symetrii pasa terenu stanowiącego ulicę. Dla obiektu typu plac oraz rondo jest to powierzchnia ograniczona linią łamaną, wyznaczoną przez odcinki zewnętrznych granic tych obiektów, ustalone na podstawie danych ewidencji gruntów i budynków, danych BDOT500 lub mapy zasadniczej.
<b>Klasa: AD_Ulica</b>	
<b>Ograniczenie:</b>	
<i>Nazwa:</i>	Geometria placu jest powierzchnią
<i>Język naturalny:</i>	Geometria obiektu typu 'plac' musi być powierzchnią.
<i>OCL:</i>	inv: self.typ = 'plac' implies self.geometria.ocllsTypeOf(GM_Surface)
<b>Ograniczenie:</b>	
<i>Nazwa:</i>	Geometria ulicy jest linią
<i>Język naturalny:</i>	Geometria obiektu typu 'ulica' musi być linią.
<i>OCL:</i>	inv: self.typ = 'ulica' implies self.geometria.ocllsTypeOf(GM_Curve)

Rys. 1 Fragment rozporządzenia w sprawie EMUiA dotyczący obiektu ulica

Rzeczywistość w tym względzie jest znacznie bogatsza i brak ciągłości geometrycznej ulicy jest zjawiskiem dosyć często w niej występującym. Jeden z prostszych przypadków (ulica PCK w Dęblinie) przedstawiono na poniższym rysunku.

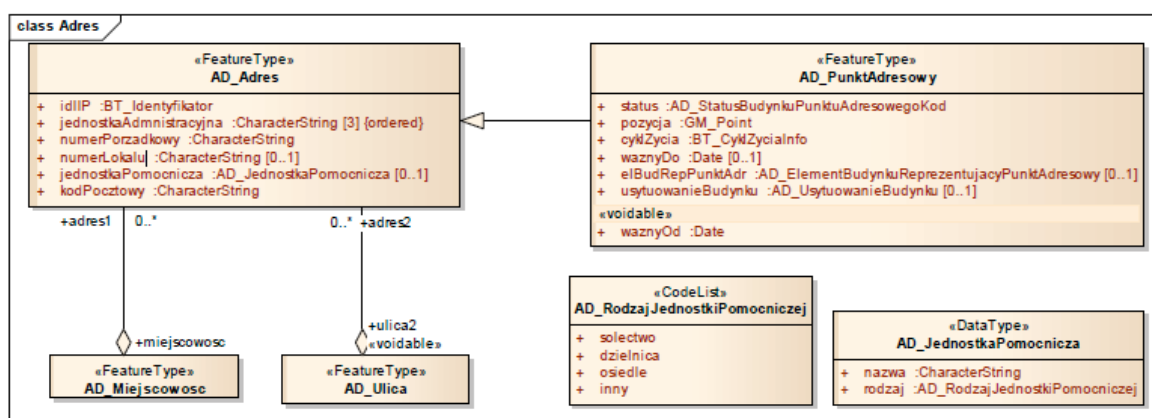


Rys. 2 Przykład ulicy, której geometrii nie można zapisać zgodnie z rozporządzeniem ws. EMUiA

W rozporządzeniu występują także inne problemy dotyczący definicji atrybutów. Każdy obiekt i każdy atrybut powinien być jasno opisany, a zapisy w treści dokumentu i w pliku ze schematem powinny być zgodne. Niestety w rozporządzeniu znajdziemy przypadki niespójności. Najlepszym przykładem jest informacja o numerze lokalu, która pojawia się w schemacie, a brak jest jakiegokolwiek opisu w treści zasadniczej rozporządzenia. Najprawdopodobniej wynika to z niejednoznaczności rozporządzenia dotyczącej dwóch pojęć: „adres budynku” i „adres”. Czasami pojęcia te są traktowane jako tożsame, a czasami z rozporządzenia wynika, że dotyczą zupełnie innych pojęć. Jak bowiem interpretować zapis §6 ust. 1 gdzie napisane jest:

- § 6. 1. Na **adres budynku** składają się informacje:
- 1) nazwa województwa i jego identyfikator TERYT;
  - 2) nazwa powiatu i jego identyfikator TERYT;
  - 3) nazwa gminy i jej identyfikator TERYT;
  - 4) rodzaj i nazwa jednostki pomocniczej (sołectwo, dzielnica, osiedle i inne);
  - 5) nazwa miejscowości o statusie miasta lub wsi i jej identyfikator TERYT;
  - 6) nazwa miejscowości stanowiącej część miasta lub wsi i jej identyfikator TERYT;
  - 7) nazwa ulicy lub placu i jej identyfikator TERYT;
  - 8) numer porządkowy;
  - 9) kod pocztowy.

podczas gdy najprawdopodobniej autorom chodziło o ogólne pojęcie **adresu**. Co byłoby zgodne z interpretacją modelu, w którym obiekt „AD\_Adres” przybiera postać:



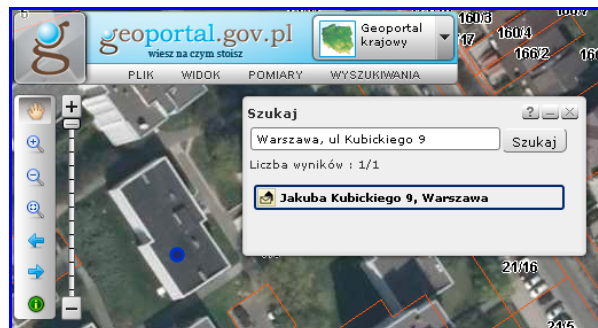
Rys. 3 Schemat klasy „AD\_Adres” z rozporządzeniem ws. EMUiA

i jak widać jest dziedziczony przez obiekt „AD\_PunktAdresowy”, który posiada pozycję (lokalizację) oraz inne niezbędne parametry.



Może zamierzenia autorów szły w kierunku, zresztą bardzo logicznym, że adresy jako takie nie mają przyporządkowanej lokalizacji. Adresy są przypisane do miejscowości wprost, albo przez ulicę, a dopiero dzięki związaniu z punktem adresowym zyskują lokalizację przestrzenną. Co oznacza, że jeśli rozpatrujemy adresy wszystkich lokali w budynku zlokalizowanym w Warszawie przy ulicy **Kubickiego** i oznaczonym numerem **9** to są one związane z jednym punktem adresowym dotyczącym budynku (ilustracja poniżej). Wtedy jednak nie powinno być użyte dziedziczenie lecz **adresom** powinno się przyporządkować identyfikator odpowiedniego **punktu adresowego**.

Warszawa, ul. Kubickiego 9 lok. 1  
 Warszawa, ul. Kubickiego 9 lok. 2  
 Warszawa, ul. Kubickiego 9 lok. 3  
 Warszawa, ul. Kubickiego 9 lok. 4  
 Warszawa, ul. Kubickiego 9 lok. 5  
 Warszawa, ul. Kubickiego 9 lok. 6



Rys. 4 Ilustracja powiązania adresów z lokalizującym je punktem adresowym

Niejednoznacznie też brzmi zacytowany poniżej §6 ust. 2 rozporządzenia:

2. Do każdego numeru porządkowego przyporządkowuje się punkt adresowy:

Zgodnie z dołączonym modelem należałoby może użyć zapisu, że **każdemu adresowi przyporządkowuje się punkt adresowy**.

#### 4. Wnioski

Żadna ze wspomnianych prób implementacji formatu GML nie zakończyła się pełnym sukcesem. W każdym z przypadków największe problemy pojawiały się na etapie tworzenia plików z danymi, na podstawie opracowanych schematów aplikacyjnych. W momencie wykrycia błędów już na etapie transferu modyfikacja schematów jest bardzo trudna, bowiem pociąga za sobą nie tylko zmiany w aplikacjach, ale także w aktach prawnych ustanawiających te schematy jako obowiązujące. Błędy w modelach wynikały głównie z niedostatecznej nieznaności modelowanych zagadnień i braku elementarnych testów opracowanych modeli. Z dotychczasowych doświadczeń wynika wprost kilka podstawowych wniosków w zakresie praktycznego wykorzystania języka GML.

1. Akty prawne nie powinny zawierać w swojej treści schematów GML przynajmniej z dwóch powodów:
  - bez zmiany prawa nie można poprawić w nich nawet oczywistych omyłek,
  - restrykcyjne zasady dotyczące przetwarzania plików opartych na języku XML wymagają aby schematy funkcjonowały jako oddzielna, zwarta całość w postaci pliku lub zbioru plików i w taki sposób był dostarczany późniejszym użytkownikom.

Oczywiście przy odrobinie chęci można wymagane schematy publikować na stronach internetowych z historią ewentualnych zmian oraz informacją o wersji, która jest obowiązująca.

2. Należy wydzielić części wspólne występujące w wielu modelach do samodzielnego bytu i się na niego powoływać, zamiast dołączać go do wszystkich modeli zaciemniając istotę oraz powodując zbyteczny rozrost treści np. definicję **modelu podstawowego**, która już dzisiaj występuje w kilku rozporządzeniach, a na tym nie koniec.
3. Schematy nie mogą być niespójne z treścią aktów prawnych. Szczególnie dotyczy to posiadania przez obiekty atrybutów, o których dowiadujemy się dopiero przy analizie schematu, a w treści aktu nie ma o tym ani słowa.
4. Aby stworzyć dobry schemat aplikacyjny GML należy dokładnie zbadać specyfikę modelowanej rzeczywistości, gdyż poprawność formalna modelu GML nie gwarantuje sukcesu w jego praktycznym wykorzystaniu.
5. Przed przyjęciem modelu jako funkcjonującego powinien on zostać przetestowany na rzeczywistych danych w możliwie szerokim zakresie przypadków.

#### 5. Literatura

- [1.] W3C. Extensible Markup Language (XML) 1.0 (Fifth Edition). <http://www.w3.org/TR/REC-xml/>.
- [2.] OpenGIS® Geography Markup Language (GML) Encoding Standard <http://www.opengeospatial.org/standards/gml>
- [3.] Rozporządzenie MAiC w sprawie „Ewidencji miejscowości, ulic i adresów”
- [4.] Rozporządzenie MAiC w sprawie „Bazy danych geodezyjnej ewidencji sieci uzbrojenia terenu, bazy danych obiektów topograficznych oraz mapy zasadniczej”