

PROBLEMATYKA MODELOWANIA INFORMACJI GEOGRAFICZNEJ W SCHEMATACH GML

PROBLEMS OF GEOGRAPHIC INFORMATION MODELING IN GML SCHEMAS

Alina Kmiecik

Instytut Informatyki, Politechnika Łódzka

Słowa kluczowe: modelowanie informacji geograficznej, języki modelowania schematów pojęciowych, UML, GML, konwersja schematów UML do GML

Keywords: modeling of geographic information, conceptual schema languages, UML GML, UML to GML schemas conversion

Wprowadzenie

Rozwój technologii informacyjnych, związany z tym spadek kosztów pozyskania i utrzymania danych oraz upowszechnienie internetu zmieniły sposób traktowania zasobów geograficznych, które ze zbiorów dostępnych wąskiej grupie użytkowników stały się dobrem powszednim, nierzadko udostępnianym publicznie i wymienianym pomiędzy niezależnymi organizacjami w ramach szeroko rozumianego współdziałania. Potwierdzeniem takiego scenariusza jest projekt infrastruktury danych przestrzennych INSPIRE (*IN*frastructure for *S*patial *I*nfoRmation i *E*urope), którego podstawowym założeniem jest publiczny dostęp do zasobów geograficznych krajów członkowskich Unii Europejskiej.

Współdzielenie danych geograficznych w zróżnicowanym technologicznie i koncepcyjnie środowisku nie jest zadaniem prostym. Każdy system informacyjny przechowuje dane w formacie, który zazwyczaj nie jest rozumiany przez inne systemy. Dlatego współdzielenie danych między organizacjami bazującymi na heterogenicznych systemach informacyjnych wymaga niezależnego sprzętowo i narzędziowo standardu zapisu danych geograficznych. Standard ten powinien wyznaczać nie tylko fizyczny format plików i notację przyjętą do zapisu danych; powinien także dawać możliwość przedstawienia logicznej organizacji danych, która pozwoli odbiorcom jednoznacznie zrozumieć ich znaczenie. Jaką wartość niesie bowiem udostępniony publicznie zbiór postaci {‘2’, ‘35’, ‘Θεσσαλονίκη’, ‘#123’} bez wskazania kontekstu i struktury organizacyjnej informacji? Owo znaczenie danych powinno być wyrażone w postaci modeli (schematów) danych.

Konsorcjum OGC (*Open Geospatial Consortium*) opracowując język zapisu danych geograficznych GML (*Geography Markup Language*) uwzględniło wagę modelu danych w pro-

cesie współdzielenia zasobów. Leżąca u podstaw języka format¹ XML (*eXtensible Markup Language*) pozwala w jednolity sposób przedstawić zarówno dane geograficzne jak również model logiczny (tzw. schemat aplikacyjny) opisujący znaczenie i kontekst tych danych. Formalny charakter języka wymagany na potrzeby zautomatyzowanego przetwarzania przez systemy komputerowe powoduje jednak, że budowanie schematów aplikacyjnych bezpośrednio w GML nie jest proste. Potrzeba dogłębnej znajomości notacji XML oraz specyficznych reguł modelowania ustanowionych przez twórców GML w sposób znaczący zawęża grono osób będących w stanie modelować dane geograficzne w tym języku. Cechy stanowiące potęgę GML dla zapisu danych, takie jak drzewiasta organizacja i opis znacznikowy danych, stają się słabą stroną języka w przypadku modelowania, powodując znaczący wzrost plików, zmniejszając czytelność i możliwości analizy struktur i ich związków.

Celem publikacji jest przedstawienie koncepcji rozwiązania problemu modelowania schematów w języku GML ukierunkowanych na uproszczenie procesu oraz przesunięcie odpowiedzialności modelowania danych ze środowisk informatycznych na społeczności geoinformacyjne. Kolejne rozdziały omawiają odpowiednio: podstawy języka GML, modelowanie schematów w języku GML i problemy z tym związane oraz proponowane rozwiązania dla tych problemów.

Język GML

Geography Markup Language jest standardem de facto zapisu danych opracowanym przez konsorcjum OGC na potrzeby wymiany informacji geograficznej. Zasadniczo GML dostarcza wyrażen służących do sformalizowanego opisu aspektów geograficznych świata rzeczywistego takich jak obiekty geograficzne, ich reprezentacje geometryczne i zależności topologiczne, systemy odniesienia, czas czy jednostki miary. Podstawowe pojęcia zamodelowane w języku GML zostały pierwotnie zaczerpnięte ze *Specyfikacji Abstrakcyjnej Open Geospatial Consortium* (OGC, 1999), a następnie dostosowane do postanowień norm ISO serii 19100. Warto nadmienić, że najnowsza wersja 3.1.1 języka (OGC, 2005) stanowi podstawę opracowania projektu normy *ISO/DIS 19136 Geographic information – Geography Markup Language*. Publikacja dokumentu normatywnego planowana jest na lipiec 2007 roku.

Zasadniczo język GML realizuje ogólny model obiektu *General Feature Model* (ISO, 2005) przyjęty w rodzinie norm ISO 19100. Zakłada, że informacja geograficzna stanowi kolekcję obiektów. Każdy obiekt (ang. *feature*) reprezentuje pewną *abstrakcję zjawisk występujących w świecie rzeczywistym*. Obiekt ten jest obiektem geograficznym, jeżeli jest powiązany z położeniem względem Ziemi. Stan każdego obiektu jest określony przez zbiór *właściwości*. Właściwości obiektów mogą być proste lub złożone, mogą mieć charakter nieprzestrzenny lub przestrzenny. Liczba właściwości jakie może posiadać obiekt, jak również nazwy i typy tych właściwości zależą od definicji *typu* tego obiektu. Obiekt może mieć powiązanie (związek) z innymi obiektami. Powiązanie może reprezentować silną zależność typu całość –

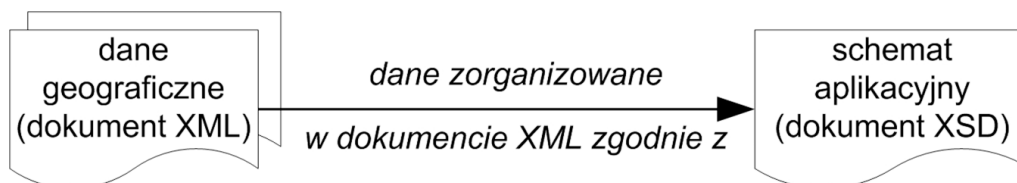
¹ **Komentarz Redakcji.** Według dokumentu *XML 1.0 W3C Recommendation* język XML jest *podzbiorem* (dialektem) języka SGML określonego w normie ISO 8879. Oba te języki, SGML i XML, są przez ich twórców określane jako formalne metajęzyki służące do definiowania języków znacznikowych. W tym przypadku termin metajęzyk oznacza rolę, jaką pełnią, są one bowiem przeznaczone wyłącznie do definiowania innych języków, w tym przypadku znacznikowych. GML jest językiem znacznikowym, jest zdefiniowany przy pomocy języka (ściślej – metajęzyka) XML i nie jest jego dialektem.

część tzn. jeden obiekt jest częścią składową drugiego obiektu lub słabą zależność, w której jeden obiekt odwołuje się do drugiego obiektu. Powiązanie musi być jawnie zdefiniowane między typami danego obiektu i opisane rolami, jakie każdy z obiektów pełni w związku. Właściwości przestrzenne obiektu geograficznego są opisywane z wykorzystaniem elementów geometrycznych i topologicznych, które – tak jak obiekty – posiadają typ, właściwości i powiązania. Ze względu na złożoność struktury tych elementów właściwości przestrzenne obiektów geograficznych są modelowane jako powiązania. Właściwości nieprzestrzenne obiektu o złożonej strukturze także mogą być reprezentowane jako osobne obiekty i wówczas mogą być modelowane przy pomocy powiązania (np. obiekt p1 typu Działka może mieć powiązanie z obiektem w1 reprezentującym Właściciela).

Cyfrowa reprezentacja świata rzeczywistego może być modelowana jako *kolekcja obiektów*. Kolekcja obiektów jest zbiorem obiektów, który sam może być postrzegany w kategorii obiektu. Oznacza to, że kolekcja obiektów ma swój typ i może posiadać właściwości.

Pomimo mylącej nazwy GML nie jest kolejnym językiem kodowania informacji. Wykorzystuje notację i zasady modelowania informacji właściwe językowi XML – opracowanego przez konsorcjum W3C (Word Wide Web) języka znacznikowego dedykowanego niezależnej sprzętowo i narzędziowo wymianie danych między systemami spiętymi w sieci informatycznej (internet, intranet). W tym sensie GML jest więc tylko zastosowaniem XML w obszarze informacji geograficznej; formą biblioteki zawierającej definicje wszystkich struktur wspierających modelowanie informacji geograficznej. Dlatego podstawą zrozumienia zapisów danych prezentowanych w GML jest znajomość składni języka XML. Ze względu na ograniczenia wprowadzone przez organizatorów publikacji, charakterystyka języka XML odniesiona do języka GML zostanie przedstawiona w dużym skrócie. Zainteresowanych szczegółami składni i gramatyki języka XML autorka zachęca do odwiedzenia strony konsorcjum W3C (<http://www.w3c.org>).

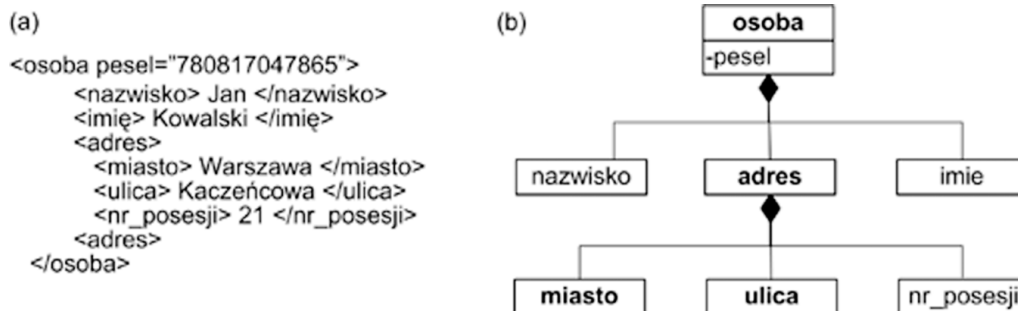
Zapis danych w języku GML jest dwuczęściowy: dane zapisywane są w dokumentach GML (plik z rozszerzeniem XML); opis deklaratywny danych określający ich logiczną organizację w pliku podany jest w schemacie aplikacyjnym GML (plik z rozszerzeniem XSD). Zawartość obu plików stanowią tzw. elementy XML (rys. 1).



Rys. 1. Dwuczęściowy zapis danych w języku GML

Element XML reprezentuje określoną porcję informacji: definicje lub dane o obiektach świata rzeczywistego, elementach geometrycznych opisujących właściwości przestrzenne obiektów itd. Element może mieć atrybuty oraz zawartość. Atrybuty są wykorzystywane do zapisu danych atomowych takich jak np. identyfikator obiektu. Zawartość elementu służy do zapisu złożonych struktur danych. Zawartość może być pusta, może być sekwencją różnych elementów, wybranym elementem z alternatywnej listy elementów, powtórzeniem elementów, zwykłym tekstem lub kombinacją elementów i tekstu. Zakres elementu wyznaczony jest bądź to parą znaczników elementu, pomiędzy którymi zapisana jest zawartość ele-

mentu. Rysunek 2 podaje przykład zapisu elementu *osoba* posiadającego atrybut *pesel* o wartości *780817047865*; na zawartość elementu *osoba* składają się trzy zagnieżdżone elementy *nazwisko*, *imię* oraz *adres* (2a). Dwa pierwsze elementy posiadają zawartość w postaci tekstu odpowiednio: *Jan* i *Kowalski*. Element *adres* natomiast zawiera wewnętrzną strukturę zbudowaną z elementów *miasto*, *ulica* i *nr_posesji*. Drzewiasta struktura elementu została dodatkowo zobrazowana na diagramie (2b).



Rys. 2. Przykłady zapisu elementów w notacji XML:

a) zapis elementu *osoba*; b) diagram UML reprezentujący złożoną strukturę elementu *osoba*

Ważną regułą języka GML jest, że element XML reprezentujący obiekt geograficzny posiada atrybut *id*, który pozwala jednoznacznie identyfikować element XML w dokumencie. Właściwości obiektów są zawsze modelowane jako znaczniki zagnieżdżone w ramach elementów reprezentujących obiekty i nie mogą posiadać identyfikatora. Dla odmiany żaden obiekt geograficzny nie może być zagnieżdżony wewnątrz innego obiektu, a powiązania między elementami mogą być realizowane jedynie za pomocą właściwości.

Dokument GML stanowi plik wymiany danych geograficznych. Składa się z elementów XML które poprzez zagnieżdżanie odzwierciedlają logiczną strukturę informacji. Organizacja danych taka jak kolejność zagnieżdżeń elementów jest ustalona w schemacie aplikacyjnym GML.

Schemat aplikacyjny GML wyznacza zakres schematu wymiany, definicje typów i deklaracje elementów XML wykorzystywanych w dokumencie GML. Typy określają listę atrybutów i zawartość dla elementów XML. Deklaracje elementów ustanawiają nazewnictwo, które jest następnie stosowane w dokumencie XML do zapisu poszczególnych elementów. Każdy schemat aplikacyjny zgodny ze specyfikacją języka GML powinien importować *predefiniowane schematy języka GML* oraz spełniać reguły budowy schematów aplikacyjnych podane w specyfikacji języka.

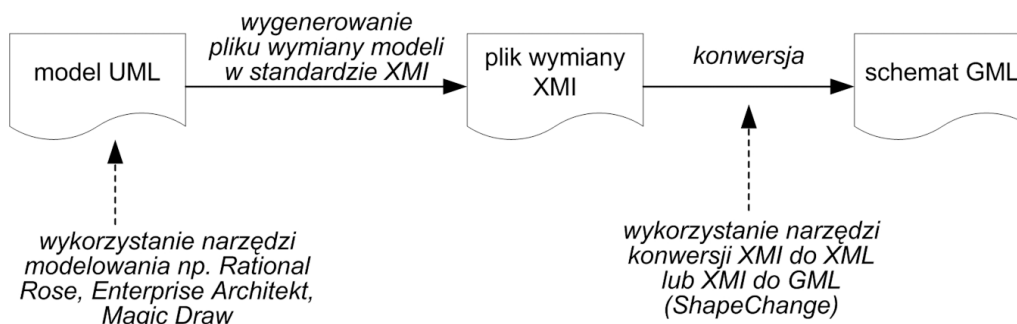
Predefiniowane schematy GML. GML stanowi zbiór schematów XML, które definiują podstawowe struktury danych do zapisu informacji geograficznej. W najnowszej wersji (3.1.1) języka jest ich aż dwadzieścia dziewięć. Język GML korzysta także ze schematów języka XLink (*XML Linking Language*) oraz SMIL (*Synchronized Multimedia Integration Language*) w wersji 2.0.

Modelowanie informacji geograficznej w języku GML

Wykorzystanie języka GML jako języka modelowania schematów pojęciowych nie jest proste. To co było zaletą języka w przypadku zapisu danych, dla modelowania staje się prawdziwym utrudnieniem. Tekstowa reprezentacja modelu, drzewiasta organizacja i znacznikowy opis struktur powodują znaczny rozrost pliku, co nawet przy ograniczonej dziedzinie objętej procesem modelowania praktycznie uniemożliwia ogarnięcie całości modelowanych struktur i analizę występujących między nimi związków. Przykładowo, zapis uproszczonego modelu dla sieci uzbrojenia terenu zajmuje w GML blisko 600 linii kodu; przykładowy zapis GML jedynie trzech struktur sieci uzbrojenia terenu: przewodu, odcinka przewodu i osi odcinka przewodu został zamieszczony na rysunku 3. Brak wsparcia narzędziowego innego niż tekstowe edytory języka XML sprawia, że budowa schematów wymiany danych wymaga doskonałej znajomości reguł semantycznych oraz zasad syntaktycznych języka, co ogranicza krąg potencjalnych twórców schematów do wąskiego grona specjalistów o doświadczeniu informatycznym.

Jednym z proponowanych rozwiązań dla problemu modelowania struktur GML jest zastosowanie języków schematów pojęciowych (CSL, *Conceptual Schema Language*). Większość języków modelowania pojęciowego zawiera reprezentację graficzną, która w zwięzłej formie pozwala przedstawić modelowane struktury i ich zależności, a dzięki temu szybciej znajduje zrozumienie u odbiorców. Przykładem takiego języka powszechnie wykorzystywanego do modelowania informacji geograficznej jest UML (*Unified Modeling Language*). Co więcej, język ten posiada szerokie wsparcie ze strony ogromnej rzeszy narzędzi modelowania stosowanych w inżynierii oprogramowania. Warto dodać, że komitet techniczny ISO/TC 211 Geographic information/Geomatics przyjęła UML za podstawowy język modelowania pojęć w rodzinie norm ISO 19100. Przykład uproszczonego modelu sieci uzbrojenia terenu w języku UML przedstawiony jest na rysunku 4 (warto porównać ten diagram z zapisem struktury GML).

Bazujące na tej koncepcji rozwiązania przyjmują następujący scenariusz: struktury danych są modelowane w języku UML, a następnie konwertowane do języka GML z wykorzystaniem dostępnych narzędzi z obszaru inżynierii oprogramowania (rys. 5). Tym nurtem podążają między innymi Roy Gronmo (2002), Clemens Portele (2003) (współautor języka GML) oraz organizacja ISO, która w najnowszej wersji projektu normy ISO przedstawia reguły konwersji modeli z języka UML do schematów GML. Zbadaniu faktycznej przydatno-



Rys. 5. Koncepcja wykorzystania języka UML i dostępnych narzędzi modelowania do budowy schematów GML

ści tego podejścia poświęcone były także październikowe warsztaty „ESDI Conceptual Schema Language and Tools” organizowane wspólnie przez ISO, CEN, oraz INSPIRE (Ispra, październik 2005).

Niemniej pierwsze wyniki analiz przeprowadzonych przez autorkę dowodzą, że dostępne na rynku uniwersalne narzędzia modelowania w języku UML (Rational Rose, Magic Draw, Enterprise Architect) nie są dostosowane do potrzeb budowy schematów GML. Nie posiadają mechanizmów konwersji do języka GML, nie wspierają predefiniowanych typów języka (zwolennicy modelowania schematów UML są zmuszeni najpierw zamodelować bibliotekę predefiniowanych typów danych w UML tj. ponad 400 obiektów), nie są także w stanie weryfikować poprawności semantycznej modelowanych struktur (semantyka rozumiana w ujęciu reguł języka GML, ISO 19136 Geographic information – Geography Markup Language). Dotychczasowe metody przenoszenia modeli UML do schematów języka GML opierają się na manualnym przekładzie elementów wyróżnionych w ramach modelu pojęciowego, co poza wymiarem edukacyjnym nie rozwiązuje problemu i w istotnym stopniu umniejsza wagę rozwiązania. Dodatkowo obciążona jest ryzykiem nadinterpretacji modelu przez osoby implementujące schemat GML. Efektem tego może być niezgodność schematów GML z zakładanym modelem wymiany (w UML), co niewątpliwie negatywnie wpłynie na współdziałanie w ramach infrastruktur danych przestrzennych. Dedykowane narzędzia konwersji UML–XML są z kolei opatrzone wadami, które w istotny sposób zawężają ich wykorzystanie np. restrykcje nałożone na sposób modelowania informacji, częściowa konwersja, konieczność manualnego uzupełniania schematów GML o szczegóły implementacyjne. Prototypowe narzędzie ShapeChange (Portele, 2004) opracowane przez Clemensa Portele częściowo wspiera automatyczną konwersję modelu UML do GML bazując na schematach wymiany modeli pojęciowych XMI (*XML Metamodel Interchange*). Niestety z uwagi na fakt powszechnego niedotrzymania standardu XMI przez producentów narzędzi modelowania, zastosowanie ShapeChange praktycznie ogranicza się do konwersji schematu budowanego jednym konkretnym narzędziem modelowania UML, Rational Rose. Inną kwestią, na którą należy zwrócić uwagę jest fakt, że UML jako język modelowania pojęciowego zezwala na szereg uogólnień i niejednoznaczne zapisy, które nie mają racji bytu na poziomie implementacyjnym. Dlatego zastosowanie narzędzia Clemensa Portele wymusza na modelującym wskazanie wszystkich szczegółów implementacyjnych już na poziomie modelu w UML, co w zasadzie niweczy sens przenoszenia procesu definiowania schematów GML na grunt języka UML. Dodatkowo wymaga zapamiętania nowych reguł właściwych dla języka UML i specyficznych wytycznych Shape Change dotyczących modelowania na potrzeby konwersji².

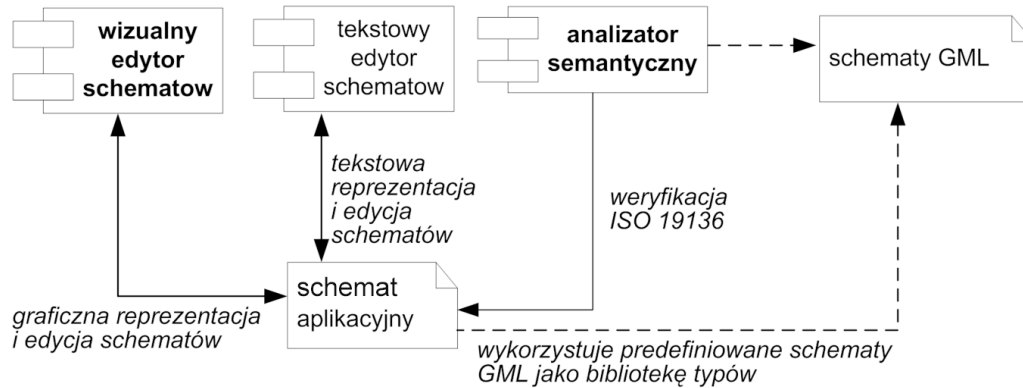
Innym rozwiązaniem problemu modelowania schematów GML rozwijanym przez autorkę publikacji jest bezpośrednie przekształcanie struktur GML na przyjazną dla użytkownika postać graficzną. W podejściu tym modelowanie danych wykonywane jest bezpośrednio na plikach GML, natomiast zadaniem narzędzia wspierającego proces modelowania jest wizuali-

² **Komentarz Redakcji.** Obecnie w literaturze dotyczącej konwersji geoinformatycznych modeli w UML do schematów GML można znaleźć różne podejścia. Obszerny przegląd tych zagadnień zawierają elektroniczne publikacje zbiorowe pod redakcją Simona Coxa, lidera prac nad rozwojem i zastosowaniami GML. Do tych publikacji należą między innymi:

UML and GML: <https://www.seegrid.csiro.au/twiki/bin/view/Xmml/UmlGml>

Basic recipe for converting a UML model to a GML Application Schema: <https://www.seegrid.csiro.au/twiki/bin/view/Xmml/UmL2GMLAS>

Publikacje te zawierają wiele odsyłaczy do innych publikacji z zakresu tej problematyki.



Rys. 6. Projekt architektoniczny autorskiego narzędzia wspierającego modelowanie schematów GML

zacja zapisów GML w postaci notacji graficznej takiej jak UML (rys. 6). Należy zauważyć, że rozwiązanie to w rzeczywistości pomija konwersję modeli UML do schematów GML, a więc unika problemów niezgodności narzędzi modelowania ze schematami wymiany XML. Nie jest też obciążone innymi wadami wcześniej opisanego rozwiązania tj. ograniczeń modelowania, niejednoznaczności w interpretacji zapisów UML i potrzeby budowania biblioteki predefiniowanych typów danych na poziomie języka modelowania. Do wizualizacji i edycji schematów można w pełni wykorzystać technologie W3C takie jak język transformacji dokumentów XML XSLT (*Extensible Stylesheet Language Transformations*), co zapewni rozwiązanie elastyczność i przenośność.

Literatura

- Gronmo R., 2002: Experiences of UML-to-GML Encoding, Proceedings of 5th AGILE Conference on Geographic Information Science.
- ISO, 2005: ISO/DIS 19136 Geographic information – Geography Markup Language.
- ISO, 2005: ISO 19118 Geographic information – Encoding.
- ISO, 2005: ISO 19109 Geographic information – Rules for application schemas.
- OGC, 1999: OpenGIS Abstract Specification, <http://www.opengeospatial.org>.
- Portele C., 2003: Mapping UML to GML Application Schemas-Guidelines and Encoding Rules, <http://www.interactive-instruments.de/ugas/UGAS-Guidelines-and-Encoding-Rules.pdf>.
- Portele C., 2004: Mapping UML to GML Application Schemas: ShapeChange – Architecture and Description, <http://www.interactive-instruments.de/ugas/ShapeChange.pdf>

Summary

Development of information technologies, costs reduction of data collecting and maintenance as well as broad usage of Internet have fundamentally changed the approach to the geographic data, which now are the aim of public access and exchange between independent organizations. The confirmation for this scenario is an INSPIRE project (INfrastructure for SPatial InfoRmation in Europe), whose primary concept is public access to geographic resources for all member countries of the European Union.

Sharing of geographic data in technologically and conceptually different environment is not easy to apply. Each information system stores its data in a format that generally is not understood by other

systems. Therefore, sharing of data between organizations using heterogeneous information systems requires platform and independent standard software for writing geographic data. This standard should include not only the format of files and notation used to write in the data, but also it should offer means for description of logical organization, that allows to understand the meaning of these data. For example, what is the value of data set like { '2', '35', 'Člóóáěđıβęç', '#123' } without indication of its context and logical organization of data structures? That meaning of data should be given in a form of data models (schemas).

OGC Consortium (Open Geospatial Consortium) took into account the value of data models while building the GML language (Geography Markup Language). The underlying XML language (eXtensible Markup Language) allows to present in a uniform way the data as well as logical models (i.e. application schema) which describe the meaning and context of data. Nevertheless, the formal character of the language required for automatic data processing in information systems causes that developing of application schema in GML is not an easy task. The features being the power of GML for data description e.g. tree-like data organization, tagged description of data, general readability of files, become weak points in the case of logical structures modeling. Additionally, the need for deep knowledge of XML syntax and GML modeling rules significantly restricts the group of potential schema authors. The aim of this paper is to present the problems of GML schemas modeling and the discussion on the alternative solutions of these problems.

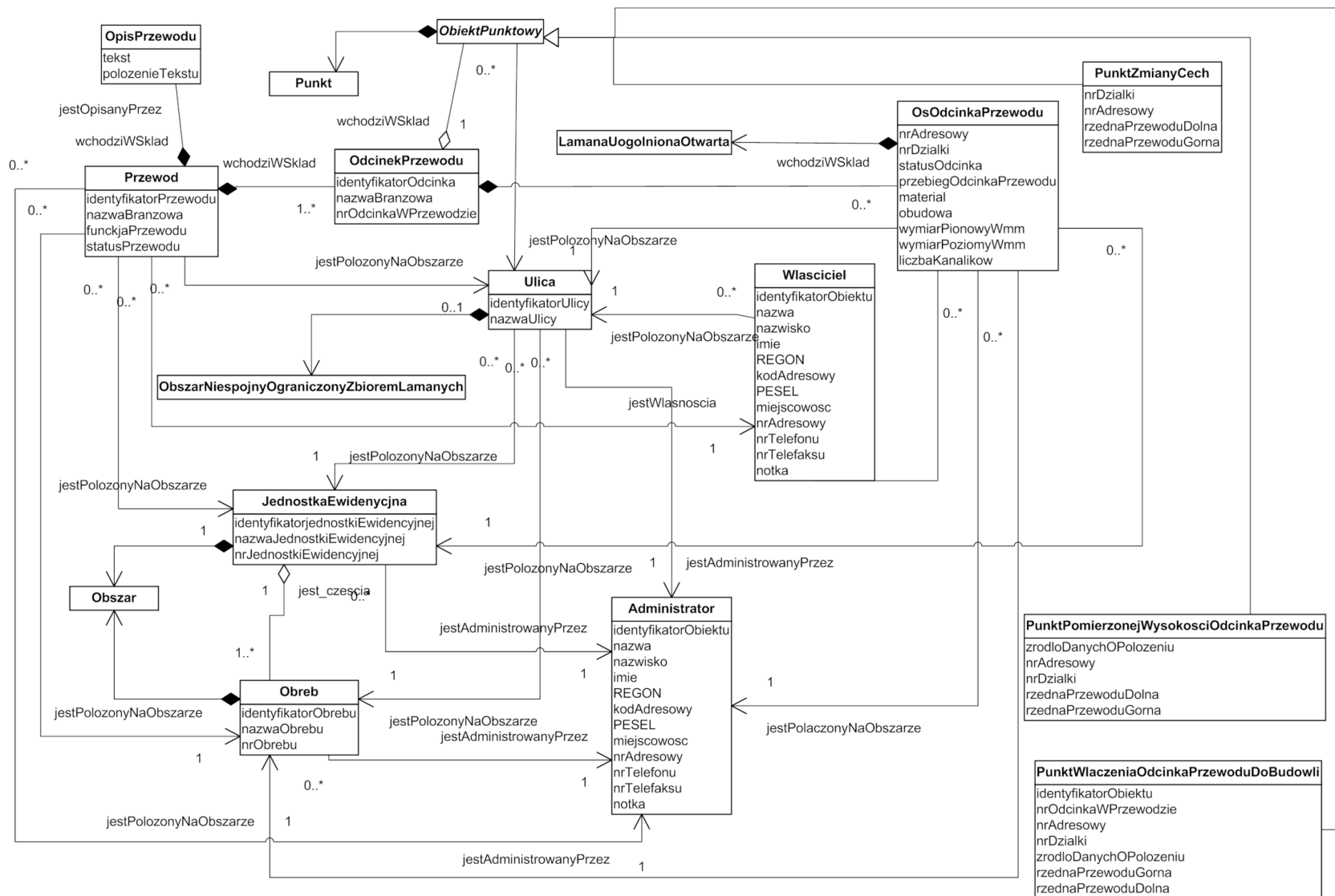
Alina Kmiciek
akmiciek@ics.p.lodz.pl
<http://ics.p.lodz.pl/~akmiciek>


```

<?xml version="1.0" encoding="utf-8" ?>
<xs:schema targetNamespace="http://temporal/gesut.pl" elementFormDefault="qualified"
  xmlns="http://temporal/gesut.pl" xmlns:mstns="http://temporal/gesut.pl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml">
<xs:import namespace="http://www.opengis.net/gml" schemaLocation="GML30/feature.xsd" />
<!--//////////////////////////////////// PRZEWÓD////////////////////////////////////-->
<xs:element type="TypPrzewod" name="Przewod" substitutionGroup="gml:_Feature" />
<xs:complexType name="TypPrzewod" abstract="false">
  <xs:complexContent>
    <xs:extension base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element name="jestPolozonyNaObszarzeJednostkiEwidencyjnej"
          type="TypPowiazanieJednostkaEwidencyjna" />
        <xs:element name="jestPolozonyNaObszarzeUlicy" type="TypPowiazanieUlica" minOccurs="0" />
        <xs:element name="jestPolozonyNaObszarzeObrebu" type="TypPowiazanieObreb" />
        <xs:element name="jestWlasnoscia" type="TypPowiazanieWlasciciel" />
        <xs:element name="jestAdministrowanyPrzez" type="TypPowiazanieAdministrator" />
        <xs:element name="jestOpisanyPrzez" type="TypPowiazanieOpis" minOccurs="0" />
        <xs:element name="identyfikatorPrzewodu" type="xs:string" />
        <xs:element name="nazwaBranzowa" type="xs:string" minOccurs="0" />
        <xs:element name="funkcjaPrzewodu" type="TypFunkcjaPrzewodu" minOccurs="0" />
        <xs:element name="statusPrzewodu" type="TypStatusOdcinkaPrzewodu" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<!--//////////////////////////////////// ODCINEK PRZEWODU////////////////////////////////////-->
<xs:element type="TypOdcinekPrzewodu" name="OdcinekPrzewodu" substitutionGroup="gml:_Feature" />
<xs:complexType name="TypOdcinekPrzewodu" abstract="false">
  <xs:complexContent>
    <xs:extension base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element name="wchodziWsklad" type="TypPowiazaniePrzewod" />
        <xs:element name="identyfikatorOdcinka" type="xs:string" />
        <xs:element name="nazwaBranzowa" type="xs:string" minOccurs="0" />
        <xs:element name="nrOdcinkaWPrzewodzie" type="xs:positiveInteger" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<!--//////////////////////////////////// OS ODCINKA PRZEWODU////////////////////////////////////-->
<xs:element name="OsOdcinkaPrzewodu" type="TypOsOdcinkaPrzewodu" substitutionGroup="gml:_Feature" />
<xs:complexType name="TypOsOdcinkaPrzewodu" abstract="false">
  <xs:complexContent>
    <xs:extension base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element name="wchodziWsklad" type="TypPowiazanieOdcinekPrzewodu"/>
        <xs:element name="jestWlasnoscia" type="TypPowiazanieWlasciciel" />
        <xs:element name="jestPolozonyNaObszarzeJednostkiEwidencyjnej"
          type="TypPowiazanieJednostkaEwidencyjna"/>
        <xs:element name="jestPolozonyNaObszarzeUlicy" type="TypPowiazanieUlica" />
        <xs:element name="jestPolozonyNaObszarzeObrebu" type="TypPowiazanieObreb" />
        <xs:element name="jestAdministrowanyPrzez" type="TypPowiazanieAdministrator" />
        <xs:element name="nrAdresowy" type="xs:string" minOccurs="0" />
        <xs:element name="nrDzialki" type="xs:string" minOccurs="0" />
        <xs:element name="statusOdcinka" type="TypStatusOdcinkaPrzewodu" minOccurs="0" />
        <xs:element name="przebiegOdcinkaPrzewodu" type="TypPrzebiegOdcinkaPrzewodu" minOccurs="0" />
        <xs:element name="material" type="TypMaterial" minOccurs="0" />
        <xs:element name="obudowa" type="TypObudowaOdcinkaPrzewodu" minOccurs="0" />
        <xs:element name="wymiarPionowyWmm" type="xs:integer" minOccurs="0" />
        <xs:element name="wymiarPozioomyWmm" type="xs:integer" minOccurs="0" />
        <xs:element name="liczbaKanalikow" type="xs:integer" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
.
.
.

```

Rys. 3. Fragment schematu GML dla sieci uzbrojenia terenu



Rys. 4. Przykład uproszczonego modelu sieci uzbrojenia terenu w języku UML