

Wrocławski Instytut Zastosowań Informatyki
Przestrzennej i Sztucznej Inteligencji

Materiały szkoleniowe

GML – warsztaty praktyczne w zakresie
niezbędnym do realizacji zadań związanych
z INSPIRE

2016-10-19

Spis treści

1	Wstęp	3
2	Rozdział I – Podstawy GML	4
2.1	Ćwiczenie – Tworzenie prostego pliku GML.....	4
2.2	Ćwiczenie – Kontrola wczytywania plików GML do QGIS	6
2.3	Ćwiczenie – Zapis geometrii w GML	9
3	Rozdział III – Konwersja UML do XSD	13
3.1	Instruktaż – Opracowanie modelu danych w UML	13
3.2	Ćwiczenie – Konwersja XMI/EAP do formatu XSD	17
3.3	Ćwiczenie – Ręczna konwersja z notacji UML do schematu XSD	20
4	Rozdział V – Konwersja danych przestrzennych z powszechnych formatów do GML	26
4.1	Instruktaż – Transformacja w Hale.....	26
4.1.1	Wczytywanie danych źródłowych do aplikacji HALE	26
4.1.2	Analiza zbioru danych oraz tabela mapowania	28
4.1.3	Implementacja procesu Harmonizacji w HALE.....	34
4.2	Ćwiczenie – Samodzielne przejście procesu harmonizacji w HALE	40
5	Walidacja plików GML względem schematu XSD	44
5.1	Instruktaż – Walidacja pliku GML w edytorze tekstowym Notepad++	44
5.2	Ćwiczenie – Walidacja pliku GML – tryb wsadowy	46
5.2.1	Walidacja pojedynczego pliku GML	46
5.2.2	Walidacja plików GML [Zaawansowane].....	46

1 Wstęp

Ćwiczenia mają za zadanie sprawdzić zrozumienie omawianego tematu przez uczestników. Zadania należy wykonywać samodzielnie, na koniec zajęć przekazane zostaną uczestnikom zeszyty ćwiczeń wraz z przykładowymi odpowiedziami. Zadania ćwiczeniowe nie są liniowe, w większości są to zadania otwarte, w związku z czym mogą mieć wiele prawidłowych rozwiązań. Dodatkowo po każdym ćwiczeniu trener zaprezentuje przykładowe rozwiązanie.

Część z ćwiczeń przeprowadzona zostanie przez trenera w postaci instruktażu, gdzie przebieg będzie omawiany oraz prezentowany przez trenera.

Niezbędne oprogramowanie:

- Notepad++ (freeware)
- QGIS w wersji 2.0+ (open source)
- AltovaXML community edition 2013 (freeware)
- HALE w wersji 3.0.0 (32 bit)

2 Rozdział I – Podstawy GML

2.1 Ćwiczenie – Tworzenie prostego pliku GML

Celem ćwiczenia będzie zmodyfikowanie istniejącego pliku XML, zawierającego informacje o pomnikach przyrody, do postaci zgodnej z GML. W rezultacie powinniśmy otrzymać plik, który bez problemu wyświetli się w oprogramowaniu QGIS.

Plik z danymi wejściowymi zostanie przekazany uczestnikom przez trenera. Jego fragment zawierający informacje dotyczące jednego obiektu został zamieszczony poniżej:

```
<?xml version = "1.0" encoding = "utf-8"?>
<zbior>
  <obiekt id="30478" >
    <nazwa>Chochoł</nazwa>
    <typ>drzewo</typ>
    <kodInspire>PL.ZIPOP.1393.PP.0201011.64</kodInspire>
    <lokalizacja>
      <kodKraju>PL</kodKraju>
      <województwo>dolnośląskie</województwo>
      <wspolrzedne układ="epsg:4326" >
        <szerokosc>51.2647</szerokosc>
        <dlugosc>15.5621</dlugosc>
      </wspolrzedne>
    </lokalizacja>
  </obiekt>
  <obiekt id="30479" >
  <obiekt id="30480" >
</zbior>
```

Rysunek 1 - Plik XML - dane wejściowe

W celu modyfikacji istniejącego pliku XML do postaci zgodnej z GML należy wykonać następujące kroki:

- Zastąpić element główny w zbiorze elementem gml:FeatureCollection, z uwzględnieniem przestrzeni nazw dla GML => xmlns:gml="http://www.opengis.net/gml",
- Poszczególne obiekty przestrzenne w zbiorze będą typu gml:featureMember, należy dla każdego z nich dodać atrybut „gml:id” – który będzie identyfikatorem obiektu przestrzennego w obrębie pliku,
- Istniejący sposób zapisu współrzędnych zamienić na definicję punktu zgodną ze standardem GML, czyli zgodny z poniższą strukturą:

```
<gml:Point>
  <gml:pos> [TUTAJ WSPÓŁRZĘDNE]</gml:pos>
</gml:Point>
```
- Zamienić istniejącą deklarację układu współrzędnych na zapis zgodny ze standardem GML, czyli dodać atrybut „srsName” z odpowiednim kodem EPSG. W rezultacie plik powinien wczytywać się w aplikacji QGIS, bez wywoływania prośby o wskazanie układu współrzędnych.

Przykładowe rozwiązanie ćwiczenia 2.1:

Zbiór zawiera trzy obiekty, dla zwiększenia czytelności w całości rozwinięto tylko pierwszy z nich.

```
<?xml version = "1.0" encoding = "utf-8"?>
<gml:FeatureCollection xmlns:gml="http://www.opengis.net/gml">
  <gml:featureMember gml:id= "1">
    <obiekt fid="30478" >
      <nazwa>Chochoł</nazwa>
      <typ>drzewo</typ>
      <kodInspire>PL.ZIPOP.1393.PP.0201011.64</kodInspire>
      <lokalizacja>
        <kodKraju>PL</kodKraju>
        <województwo>dolnośląskie</województwo>
        <wspolrzedne>
          <gml:Point srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:pos>15.5621 51.2647</gml:pos>
          </gml:Point>
        </wspolrzedne>
      </lokalizacja>
    </obiekt>
  </gml:featureMember>
  <gml:featureMember gml:id= "2">
  <gml:featureMember gml:id= "3">
</gml:FeatureCollection>
```

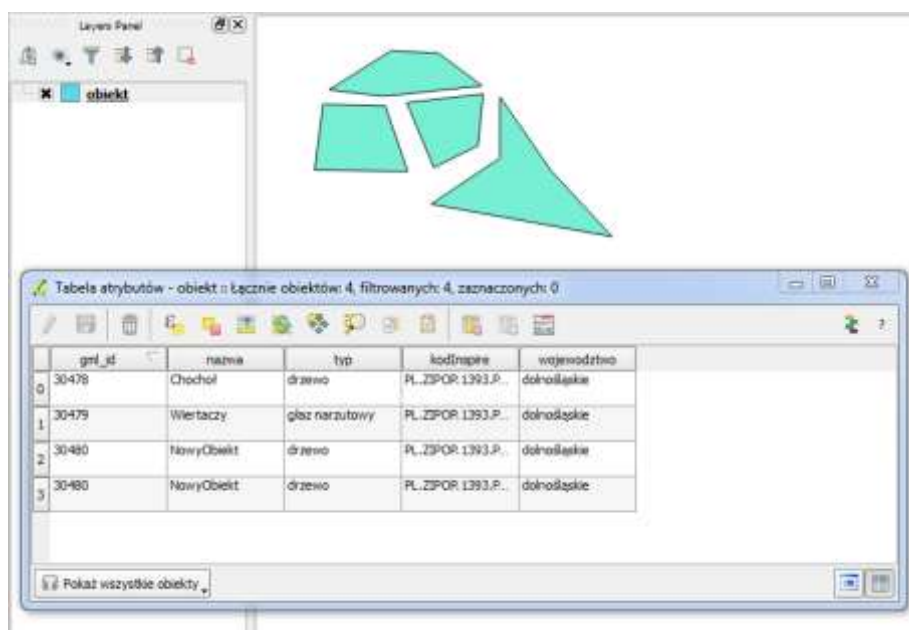
Rysunek 2 - Rozwiązanie zadania 2.1

2.2 Ćwiczenie – Kontrola wczytywania plików GML do QGIS

Pliki GML zazwyczaj posiadają strukturę zagnieżdżoną, prezentując tym samym podejście obiektowe. Dlatego warto zapoznać się ze sposobem kontrolowanego wczytywania ich do aplikacji GISowej, w tym wypadku na przykładzie QGIS.

```
<?xml version = "1.0" encoding = "utf-8"?>
<gml:FeatureCollection xmlns:gml="http://www.opengis.net/gml">
  <gml:FeatureMember gml:id= "1">
    <obiekt fid="30478" >
      <nazwa>Chochol</nazwa>
      <typ>drzewo</typ>
      <kodinspire>PL.ZIPOP.1393.PP.0201011.64</kodinspire>
      <lokalizacja kraj="PL">
        <województwo>dolnośląskie</województwo>
        <centroid układ="EPSG:4326">
          <gml:Point srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:pos>51.2762 15.7376</gml:pos>
          </gml:Point>
        </centroid>
        <otulina układ="EPSG:4326">
          <gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:outerBoundaryIs><gml:LinearRing><gml:coordinates>
              51.341473945783129,15.796944804216867
              51.19870082831325,15.780519578313251
              51.249239984939756,15.659225602409638
              51.331997853915659,15.700288667168675
              51.341473945783129,15.796944804216867
            </gml:coordinates></gml:LinearRing></gml:outerBoundaryIs></gml:Polygon>
          </otulina>
        </lokalizacja>
      </obiekt>
    </gml:FeatureMember>
    <gml:FeatureMember gml:id= "2">
    <gml:FeatureMember gml:id= "3">
    <gml:FeatureMember gml:id= "4">
  </gml:FeatureCollection>
```

Rysunek 3 - Struktura źródłowego GML



Rysunek 4 - Domyślnie wczytany plik GML

Źródłowy plik zawiera dwie geometrie, jak widać na załączniku powyżej aplikacja QGIS domyślnie wczytała jedną klasę obiektów z geometrią powierzchniową.

Celem ćwiczenia będzie wprowadzić zmiany w pliku "pomniki_przyrody.gfs" tak, aby:

- Do aplikacji QGIS dane wczytane zostały dwukrotnie jako dwie klasy obiektów o odmiennej geometrii: klasa "Centroidy" zawierająca geometrię punktową oraz "Otulina" zawierająca geometrię powierzchniową.

W celu zmodyfikowania zadeklarowania geometrii należy zmodyfikować wartość elementu <GeometryType>, która może przyjmować następujące wartości:

- 0 – dowolny typ geometrii,
- 1 – geometria punktowa,
- 2 – geometria liniowa,
- 3 – geometria powierzchniowa,
- 4 – geometria typu „MultiPoint”,
- 5 – geometria typu „MultiLinestring”,
- 6 – geometria typu „MultiPolygon”,
- 7 – kolekcja geometrii – obiekty różnego typu.

Oraz podać ścieżkę do elementu z geometrią jako wartość <GeometryElementType>.

- Klasy mają się wczytać pod nazwami „Otulina” oraz „Centroidy”.
- Każda z klas po wczytaniu ma mieć następującą strukturę (w podanej kolejności): kodInspire, nazwa, województwo, kodKraju, układEPSG.

Przed rozpoczęciem ćwiczenia struktura plik QFS zostanie omówiona przez prowadzącego szkolenie.

Rozwiązanie ćwiczenia 2.2:

```

<GMLFeatureClassList>
  <GMLFeatureClass>
    <Name>centroidy</Name>
    <ElementPath>obiekt</ElementPath>
    <GeometryType>1</GeometryType>
    <GeometryElementPath>lokalizacja|centroid
    </GeometryElementPath>
    <SRSName>EPSG:4326</SRSName>
    <DatasetSpecificInfo>
      <FeatureCount>4</FeatureCount>
      <ExtentXMin>51.02624</ExtentXMin>
      <ExtentXMax>51.58280</ExtentXMax>
      <ExtentYMin>15.53035</ExtentYMin>
      <ExtentYMax>15.87718</ExtentYMax>
    </DatasetSpecificInfo>
    <PropertyDefn>
      <Name>kodInspire</Name>
      <ElementPath>kodInspire</ElementPath>
      <Type>String</Type>
      <Width>27</Width>
    </PropertyDefn>
    <PropertyDefn>
      <Name>nazwa</Name>
      <ElementPath>nazwa</ElementPath>
      <Type>String</Type>
      <Width>10</Width>
    </PropertyDefn>
    <PropertyDefn>
      <Name>województwo</Name>
      <ElementPath>lokalizacja|województwo</ElementPath>
      <Type>String</Type>
      <Width>14</Width>
    </PropertyDefn>
    <PropertyDefn>
      <Name>kodKraju</Name>
      <ElementPath>lokalizacja@kraj</ElementPath>
      <Type>String</Type>
      <Width>15</Width>
    </PropertyDefn>
    <PropertyDefn>
      <Name>ukladEPSG</Name>
      <ElementPath>lokalizacja|centroid@uklad</ElementPat
      <Type>String</Type>
      <Width>15</Width>
    </PropertyDefn>
  </GMLFeatureClass>
</GMLFeatureClassList>

<GMLFeatureClassList>
  <GMLFeatureClass>
    <Name>otulina</Name>
    <ElementPath>obiekt</ElementPath>
    <GeometryType>3</GeometryType>
    <GeometryElementPath>lokalizacja|otulina
    </GeometryElementPath>
    <SRSName>EPSG:4326</SRSName>
    <DatasetSpecificInfo>
      <FeatureCount>4</FeatureCount>
      <ExtentXMin>51.02624</ExtentXMin>
      <ExtentXMax>51.58280</ExtentXMax>
      <ExtentYMin>15.53035</ExtentYMin>
      <ExtentYMax>15.87718</ExtentYMax>
    </DatasetSpecificInfo>
    <PropertyDefn>
      <Name>kodInspire</Name>
      <ElementPath>kodInspire</ElementPath>
      <Type>String</Type>
      <Width>27</Width>
    </PropertyDefn>
    <PropertyDefn>
      <Name>nazwa</Name>
      <ElementPath>nazwa</ElementPath>
      <Type>String</Type>
      <Width>10</Width>
    </PropertyDefn>
    <PropertyDefn>
      <Name>województwo</Name>
      <ElementPath>lokalizacja|województwo</ElementPath>
      <Type>String</Type>
      <Width>14</Width>
    </PropertyDefn>
    <PropertyDefn>
      <Name>kodKraju</Name>
      <ElementPath>lokalizacja@kraj</ElementPath>
      <Type>String</Type>
      <Width>15</Width>
    </PropertyDefn>
    <PropertyDefn>
      <Name>ukladEPSG</Name>
      <ElementPath>lokalizacja|otulina@uklad</ElementPath>
      <Type>String</Type>
      <Width>15</Width>
    </PropertyDefn>
  </GMLFeatureClass>
</GMLFeatureClassList>

```

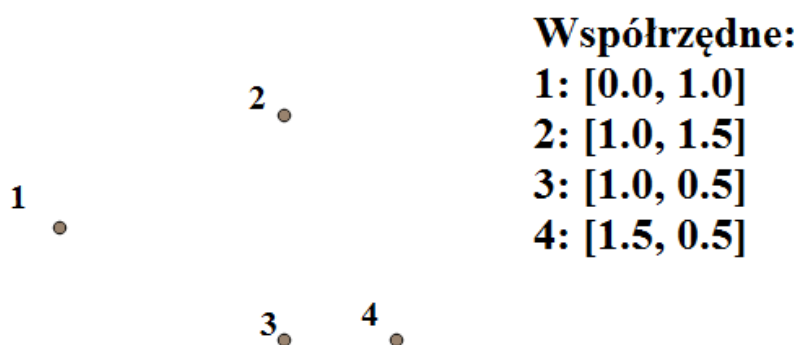
Rysunek 5 - Rozwiązanie zadania - pliki QFS dla obu klas

2.3 Ćwiczenie – Zapis geometrii w GML

Jednym z najczęstszych błędów podczas opracowywania/przetwarzania plików GML są błędy związane z zapisem geometrii. Ćwiczenie to ma na celu zapoznanie uczestników ze składnią związaną z najczęstszymi typami geometrycznymi.

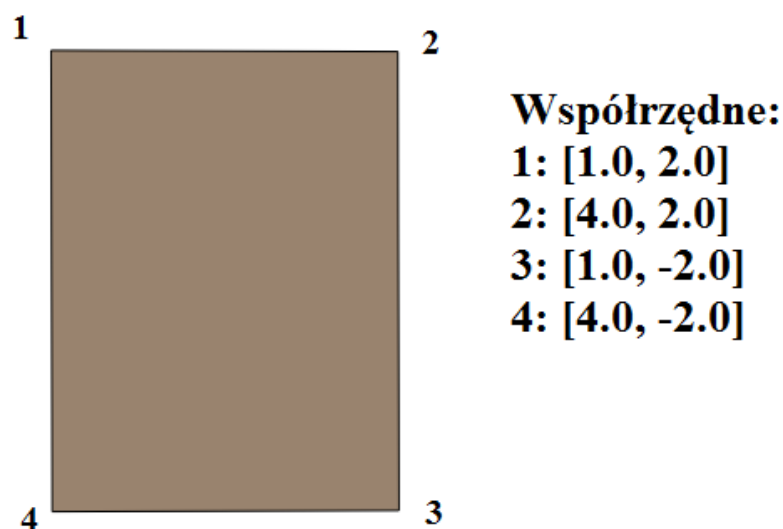
Proszę opracować prosty plik GML przedstawiający zaprezentowane poniżej figury geometryczne. Podczas opracowywania pliku należy skorzystać z szablonu przekazanego przez prowadzącego zawierającego szkielet pliku GML.

a) Multipunkt



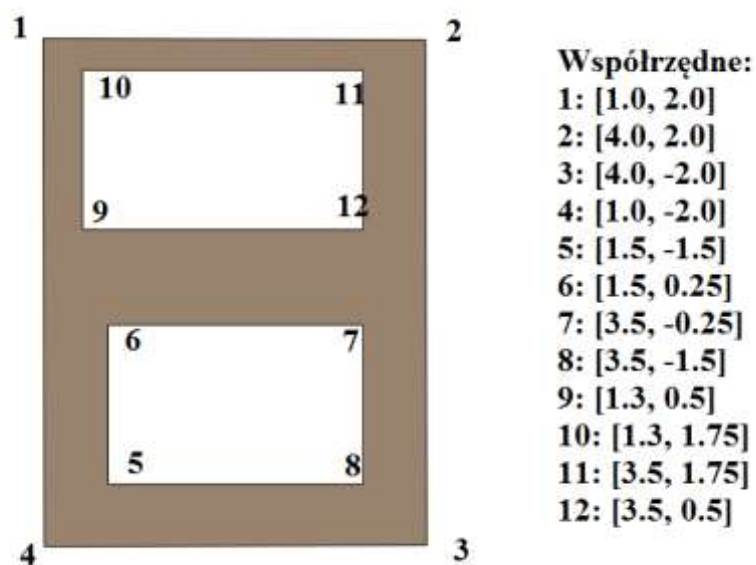
Rysunek 6 - Multipunkt do opracowania

b) Poligon



Rysunek 7 - Poligon do opracowania

c) Poligon z enklawą



Rysunek 8 - Poligon z enklawą do opracowania

Rozwiązanie ćwiczenia 2.3:

Multipunkt:

```

<?xml version="1.0" encoding="utf-8" ?>
<gml:FeatureCollection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="" xmlns:ogr="http://ogr.maptools.org/" xmlns:gml="
http://www.opengis.net/gml">
  <gml:featureMember>
    <ogr:multipoint fid="multipoint.0">
      <ogr:geometryProperty>
        <gml:MultiPoint srsName="urn:ogc:def:crs:EPSG::4326">
          <gml:pointMember>
            <gml:Point>
              <gml:coordinates>0.50,1.00</gml:coordinates>
            </gml:Point>
          </gml:pointMember>
          <gml:pointMember>
            <gml:Point>
              <gml:coordinates>0.50,1.50</gml:coordinates>
            </gml:Point>
          </gml:pointMember>
          <gml:pointMember>
            <gml:Point>
              <gml:coordinates>1.50,1.00</gml:coordinates>
            </gml:Point>
          </gml:pointMember>
          <gml:pointMember>
            <gml:Point>
              <gml:coordinates>1.00,0.00</gml:coordinates>
            </gml:Point>
          </gml:pointMember>
        </gml:MultiPoint>
      </ogr:geometryProperty>
    </ogr:multipoint>
  </gml:featureMember>
</gml:FeatureCollection>
    
```

Poligon:

```

<?xml version="1.0" encoding="utf-8" ?>
<gml:FeatureCollection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="" xmlns:ogr="http://ogr.maptools.org/" xmlns:gml="
http://www.opengis.net/gml">
  <gml:featureMember>
    <ogr:polygon_wzor fid="polygon_wzor.0">
      <ogr:geometryProperty>
        <gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326">
          <gml:exterior>
            <gml:LinearRing>
              <gml:coordinates>2.00,1.00 -2.00,1.00 -2.00,4.00
              2.00,4.00 2.00,1.00</gml:coordinates>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </ogr:geometryProperty>
    </ogr:polygon_wzor>
  </gml:featureMember>
</gml:FeatureCollection>
    
```

Poligon z enklawą:

```

<?xml version="1.0" encoding="utf-8" ?>
<gml:FeatureCollection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="" xmlns:ogr="http://ogr.maptools.org/" xmlns:gml="
http://www.opengis.net/gml">
  <gml:featureMember>
    <ogr:polygon_z_enklawa fid="polygon_z_enklawa.0">
      <ogr:geometryProperty>

        <gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326">

          <gml:exterior>
            <gml:LinearRing>
              <gml:coordinates>2.0,1.0 2.0,4.0 -2.0,4.0 -2.0,1.0
                2.0,1.0</gml:coordinates>
            </gml:LinearRing>
          </gml:exterior>

          <gml:interior>
            <gml:LinearRing>
              <gml:coordinates>1.75,1.30 0.5,1.30 0.5,3.50 1.75,3.50
                1.75,1.30</gml:coordinates>
            </gml:LinearRing>
          </gml:interior>

          <gml:interior>
            <gml:LinearRing>
              <gml:coordinates>-0.25,1.50 -1.50,1.50 -1.50,3.50
                -0.25,3.50 -0.25,1.50</gml:coordinates>
            </gml:LinearRing>
          </gml:interior>
        </gml:Polygon>
      </ogr:geometryProperty>
    </ogr:polygon_z_enklawa>
  </gml:featureMember>
</gml:FeatureCollection>
    
```

3 Rozdział III – Konwersja UML do XSD

Schematy aplikacyjne GML to po prostu dokumenty XML Schema wykorzystujące elementy zdefiniowane w standardzie GML i udostępniane przez konsorcjum OGC. Ręczne tworzenie rozbudowanych schematów nie jest zadaniem trywialnym. Możliwe jest wykorzystanie aplikacji umożliwiających konwersję odpowiednio opisanych diagramów UML do schematów aplikacyjnych GML. Niestety w tej wysoce wyspecjalizowanej dziedzinie brak narzędzi open source, które umożliwiłyby przejście pełnej ścieżki produkcyjnej – od UML do schematu aplikacyjnego GML.

Do tego celu wykorzystane zostaną narzędzia:

- Enterprise Architect – do modelowania diagramów UML,
- ShapeChange – do konwersji diagramów na schemat GML (narzędzie darmowe, aczkolwiek wspiera wyłącznie pliki projektowe z Enterprise Architect oraz pliki XML, niestety tylko te wyeksportowane z EA).

3.1 Instruktaż – Opracowanie modelu danych w UML

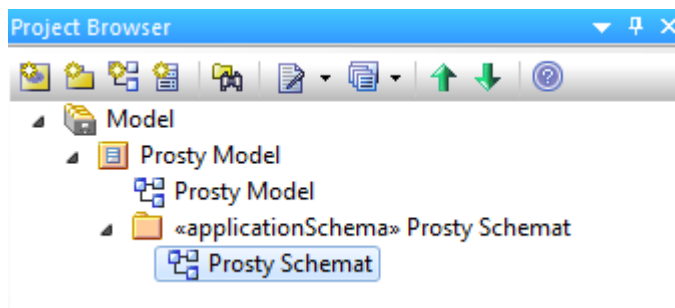
Punkt 3.1 – zostanie zaprezentowany oraz dokładnie omówiony przez prowadzącego.

Celem tej części ćwiczenia jest przygotowanie prostego diagramu klas UML w aplikacji Enterprise Architect. Modelowany schemat będzie posiadał:

- Enumerację „RodzajBudynku” przyjmującą wartości {mieszkalny, usługowy, przemysłowy},
- Typ danych „Adres” posiadający atrybuty:
 - ulica o typie znakowym,
 - miejscowość o typie znakowym,
 - kod pocztowy o typie znakowym.
- Obiekt przestrzenny „Budynek” posiadający atrybuty:
 - geometrię o typie gml:Surface,
 - identyfikator o typie znakowym,
 - liczba kondygnacji o typie liczba całkowita,
 - nazwa budynku o typie znakowym,
 - rodzaj budynku o typie enumeracyjnym „RodzajBudynku”,
 - adres budynku o typie „Adres”.
- Obiekt „BazaDanychBudyneków” agregujący obiekty przestrzenne „Budynek”.

Kroki do wykonania:

1. W aplikacji Enterprise Architect należy utworzyć projekt „schemat_gml.eap”, który będzie stanowił bazę dla tworzonego schematu aplikacyjnego.
2. Aby rozpocząć tworzenie diagramu należy w oknie Project Browser rozwinąć obiekty i dwukrotnie kliknąć na pozycję „Prosty Schemat”. W oknie głównym otwarte zostanie puste okno edycji diagramów.



Rysunek 9 - „Drzewko” projektowe

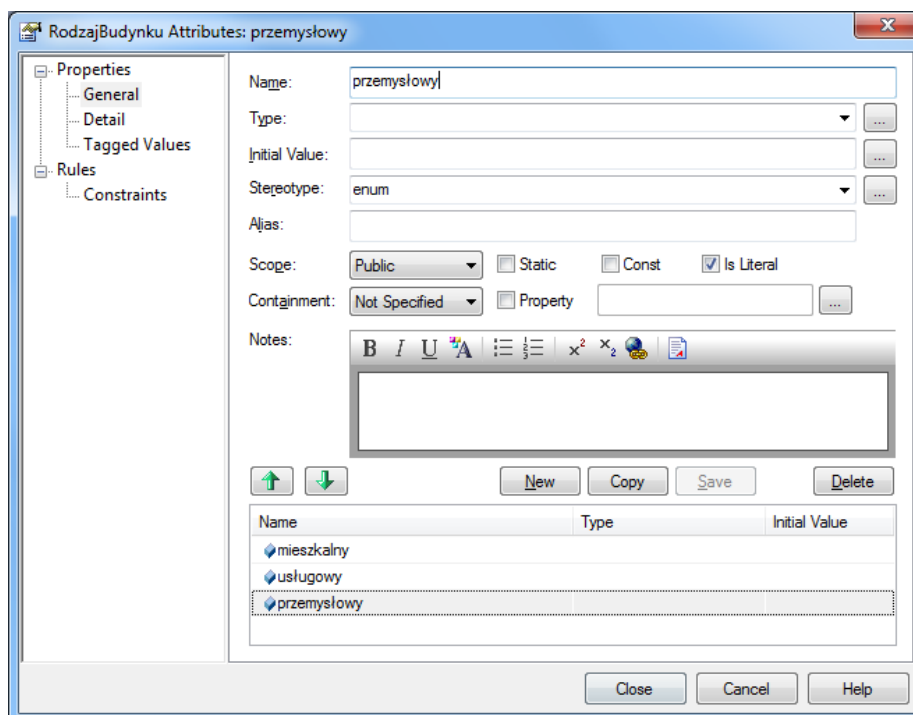
3. Aby dodać nowy element należy kliknąć prawym przyciskiem myszy w oknie głównym i wybrać pozycję „New Element or Connector” i wskazać obiekt Class.
4. Pierwszym elementem, jaki zostanie dodany będzie enumeracja „RodzajBudynku”. Po wprowadzeniu nazwy nowej klasy należy pamiętać o ustawieniu stereotypu na „enumeration”.

Name: RodzajBudynku

Stereotype: enumeration Status: Proposed

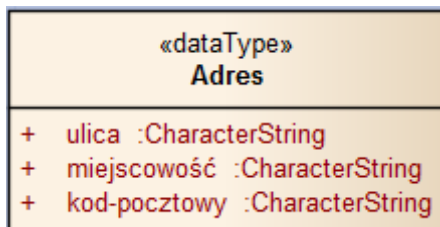
Rysunek 10 - Enumeracja - "Rodzaj budynku"

5. Aby dodać do enumeracji dozwolone wartości należy kliknąć prawym przyciskiem myszy na enumerację i wybrać opcję „Features & Properties” i wskazać opcję „Attributes”. Po wprowadzeniu nazwy należy zapisać atrybut za pomocą przycisku „Save”. Aby utworzyć nową wartość należy wybrać opcję „New”.



Rysunek 11 - Enumeracja dodawanie kolejnych wartości

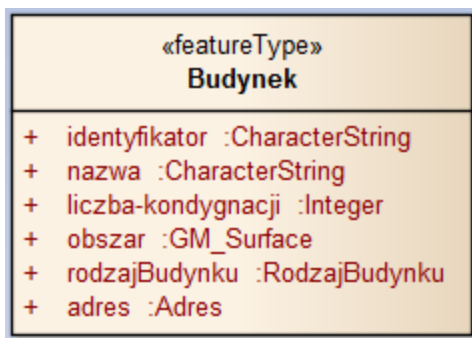
6. Następnie należy zdefiniować nowy typ danych „Adres”. W tym celu należy stworzyć nową klasę, nazwać ją i ustalić stereotyp na „DataType”.
7. Utworzony typ danych powinien posiadać wcześniej określone elementy. Tworzy się je podobnie jak w przypadku enumeracji, należy tylko pamiętać, aby ustawić zasięg (Scope) na public oraz odpowiedni typ danych (CharacterString dla wszystkich 3 atrybutów).



Rysunek 12 - Złożony typ danych "Adres"

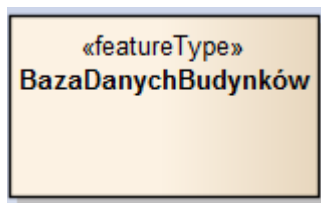
8. Kolejnym krokiem jest utworzenie pierwszego i głównego obiektu przestrzennego „Budynek”. W tym celu należy utworzyć nową klasę, nazwać ją i nadać jej stereotyp „FeatureType”.
9. Nowe atrybuty powinny mieć publiczny zasięg i posiadać następujące typy danych:
 - identyfikator – CharacterString,
 - nazwa – CharacterString,
 - liczba-kondygnacji – Integer,
 - obszar – GM_Surface,
 - rodzajBudynku – RodzajBudynku,
 - adres – Adres.

Aby wybrać utworzony wcześniej typ danych („RodzajBudynku” lub „Adres”) przy wprowadzaniu typu atrybutu można kliknąć w przycisk „...” i wybrać zdefiniowaną wcześniej klasę.



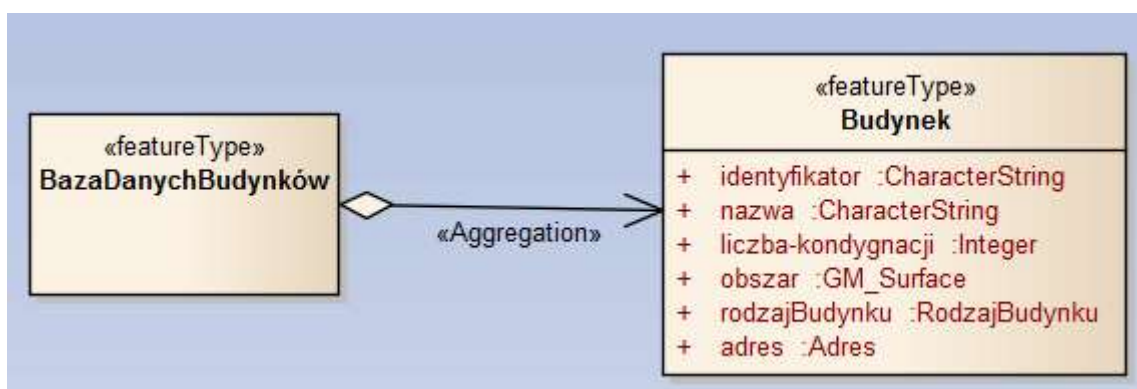
Rysunek 13 - obiekt przestrzenny "Budynek"

10. Kolejnym elementem do dodania będzie klasa agregująca „BazaDanychBudynków” o stereotypie „FeatureType”, która nie będzie posiadała żadnych atrybutów.



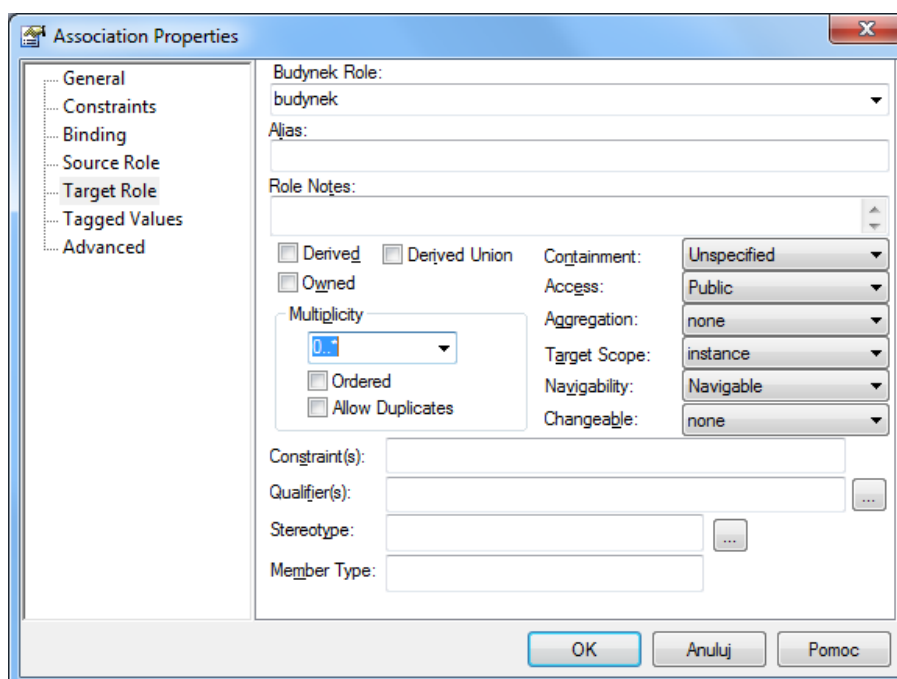
Rysunek 14 - Klasa agregująca "BazaDanychBudynków"

11. Należy teraz wskazać za pomocą odpowiedniej asocjacji, że obiekt „BazaDanychBudynków” zawierać może wiele elementów typu „Budynek”. Dodatkowo należy zaznaczyć, że będzie to agregacja.



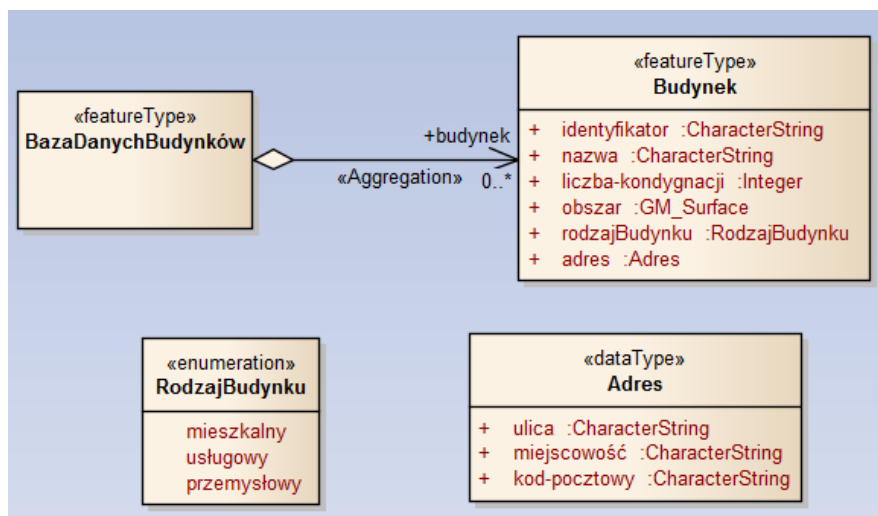
Rysunek 15 - Asocjacja z agregacją

12. Należy jeszcze wskazać krotność tej agregacji – budynków może być wiele, ale również nie musi być żadnego. W tym celu należy dwukrotnie kliknąć na asocjację i w oknie właściwości wybrać (z lewej strony) opcję „Target Role” nazwać rolę „budynek” i wskazać krotność „Multiplicity” jako „0...*”



Rysunek 16 - Ustawienia krotności agregacji

13. Utworzony schemat aplikacyjny będzie wyglądał następująco:



Rysunek 17 - Gotowy schemat aplikacyjny UML

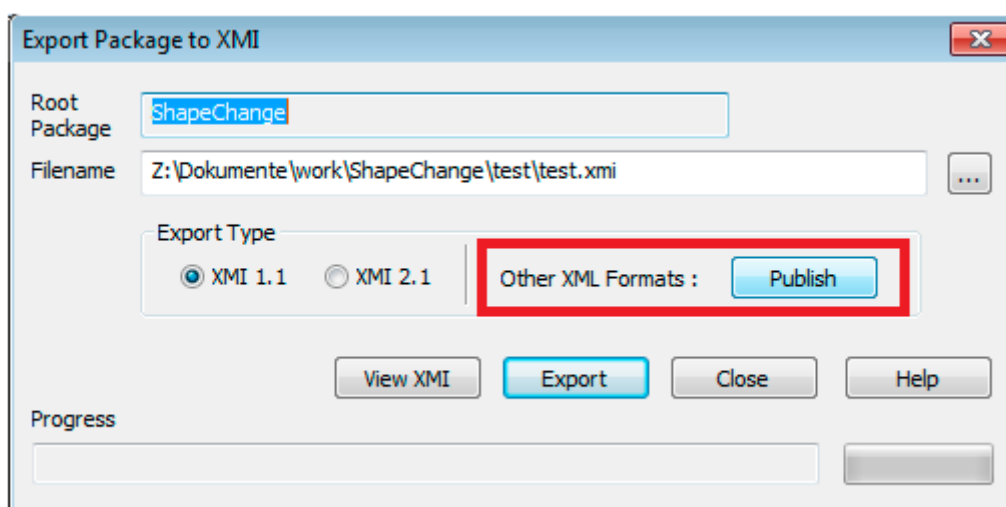
3.2 Ćwiczenie – Konwersja XMI/EAP do formatu XSD

Utworzony wcześniej diagram UML zostanie przekonwertowany do schematu aplikacyjnego GML za pomocą aplikacji ShapeChange. Aplikacja ShapeChange może przetwarzać zarówno pliki projektowe EA, jak i pliki w formacie XMI 1.0.

Oprogramowanie dostępne jest adresem: <http://shapechange.net/>

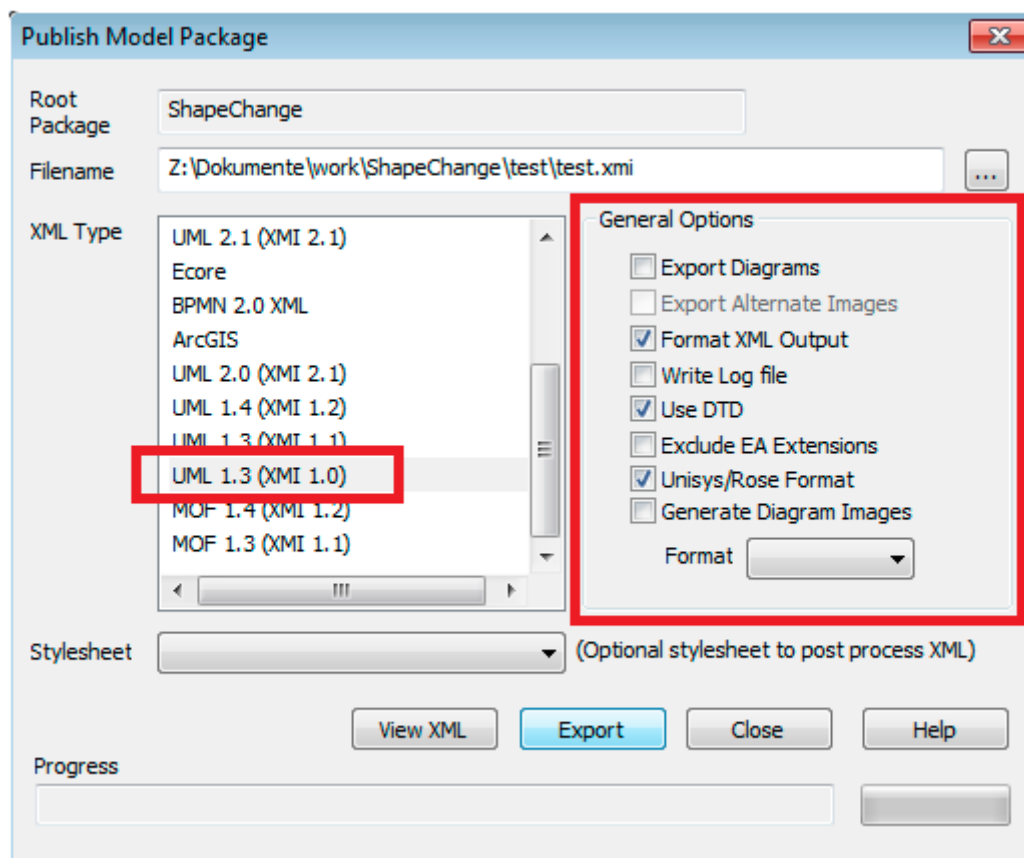
Eksport pliku projektowego EA do formatu XMI:

- W aplikacji należy wybrać opcję „Export Package to XMI”. Domyślnie aplikacja eksportuje dane do plików XMI w wersji 1.1 oraz 2.1 - w związku z czym musimy wskazać „Other XML Formats”, żeby odszukać wersję 1.0.



Rysunek 18 - Eksport do formatu XMI

Opcje eksportu muszą odpowiadać tym przedstawionym na poniższym zrzucie ekranu.



Rysunek 19 - Konfiguracja eksportu

Przygotowanie aplikacji do współpracy z Enterprise Architect:

- Konwersja plików EAP jest możliwa tylko wtedy, gdy na komputerze zainstalowana jest aplikacja Enterprise Architect.
- Należy odszukać plik SSJavaCom.dll zlokalizowany w folderze, w którym zainstalowany został Enterprise Architect (schemat ścieżki: <EA installation folder>/Java API).
Następnie skopiować go do następującego folderu:
 - ✓ <Windows folder>/System32 – dla systemu 32-bitowego,
 - ✓ <Windows folder>/SysWOW64 – dla systemu 64-bitowego.

Aplikacja nie musi być instalowana na komputerze, wystarczy, że zostanie rozpakowana. W pierwszej kolejności należy przeprowadzić test „online” aplikacji i sprawdzić czy nie zwróci nam żadnych błędów. W tym celu należy uruchomić plik „test.bat” w głównym folderze aplikacji. Wykonuje on konwersję schematu bazując na plikach konfiguracyjnych zamieszczonych online.

```
java -jar ShapeChange-2.1.0.jar -Dfile.encoding=UTF-8 -c  
http://shapechange.net/resources/test/testXMI.xml  
java -jar ShapeChange-2.1.0.jar -Dfile.encoding=UTF-8 -c  
http://shapechange.net/resources/test/testEA.xml
```

pause

Rysunek 20 - Treść pliku "test.bat"

Pliki, które będą ulegać modyfikacji:

„testLokalnie.bat” – plik uruchamiający, wywołujący funkcję JAVA oraz odwołujący się do zadanego pliku konfiguracyjnego.

„resources/config/testEA.xml” – plik konfiguracyjny, który zawiera parametry konfiguracyjne dla konwersji plików EAP. W pliku dostosować należy lokalizację pliku wejściowego oraz zarówno folder, jak i nazwę pliku wyjściowego.

- Dane wejściowe: `<parameter name="inputFile" value="test/schematTestowy.eap"/>`
- Folder wyjściowy: `<parameter name="outputDirectory" value="test/ea"/>`
- Plik wyjściowy: `<parameter name="outputFilename" value="test "/>`

„resources/config/testXMI.xml” – plik konfiguracyjny, który zawiera parametry konfiguracyjne dla konwersji plików XMI. W pliku dostosować należy lokalizację pliku wejściowego oraz zarówno folder, jak i nazwę pliku wyjściowego.

- Dane wejściowe: `<parameter name="inputFile" value="test/schematTestowy.xmi"/>`
- Folder wyjściowy: `<parameter name="outputDirectory" value="test/xmi"/>`
- Plik wyjściowy: `<parameter name="outputFilename" value="test "/>`

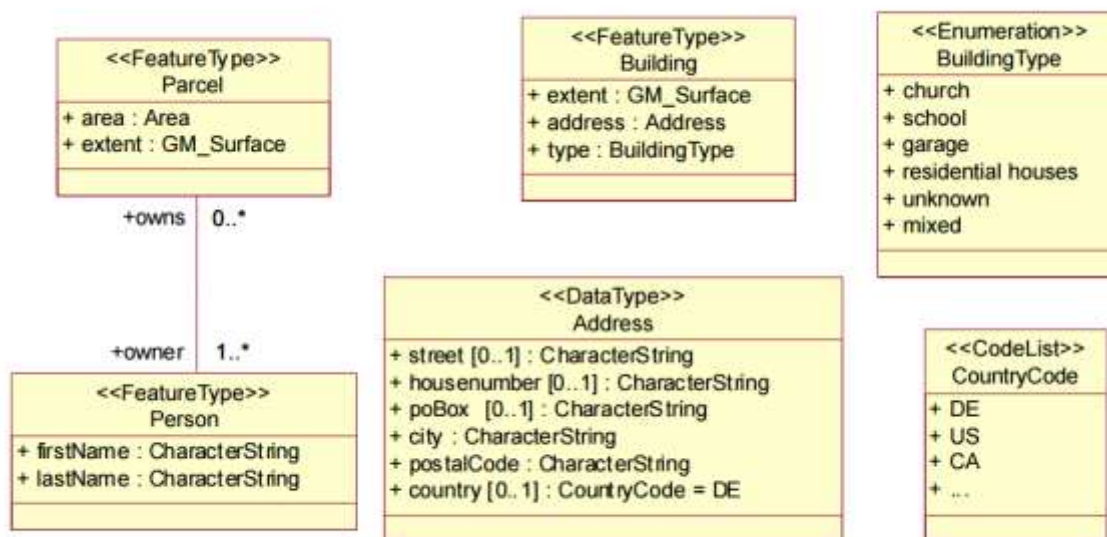
Dane wejściowe zwyczajowo kopiujemy do folderu „test”. W każdym pliku XMI musi znaleźć się odwołanie do pliku UMLX13.dtd. Odwołanie takie ma postać: **<!DOCTYPE XMI SYSTEM "UMLX13.dtd">**.

Domyślnie dane wynikowe (schemat XSD oraz katalog obiektów w HTML) zapisywane będą:

- W lokalizacji „test/ea/INPUT” – w przypadku plików EAP,
- W lokalizacji „test/xmi/INPUT” – w wypadku plików XMI.

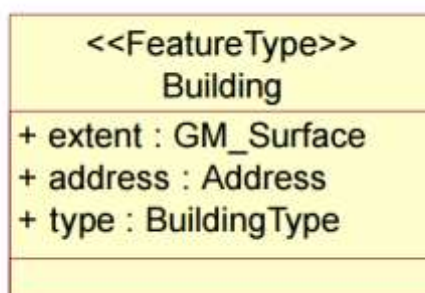
3.3 Ćwiczenie – Ręczna konwersja z notacji UML do schematu XSD

Celem ćwiczenia jest przejście przez proces mapowania z zaprezentowanego poniżej modelu do postaci schematu XSD zgodnie z ISO 19136. Zadanie zostało podzielone na mniejsze fragmenty, tak by zwiększyć czytelność zadania. Każdy z punktów zostanie omówiony przez trenera. Na końcu rozwiązanie zostanie scalone w jednym pliku.



Rysunek 21 - Docelowy model UML do przemapowania do XSD

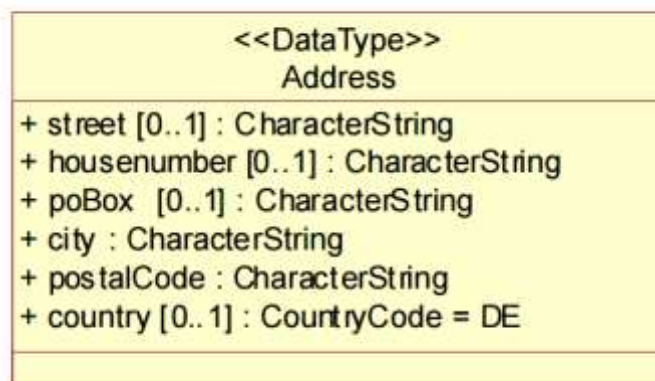
a) Klasa obiektów "Building"



Rysunek 22 - Klasa "Building"

Do opracowania proszę skorzystać z dostarczonego przez trenera pliku XSD zawierającego szkielet schematu, który należy dokończyć.

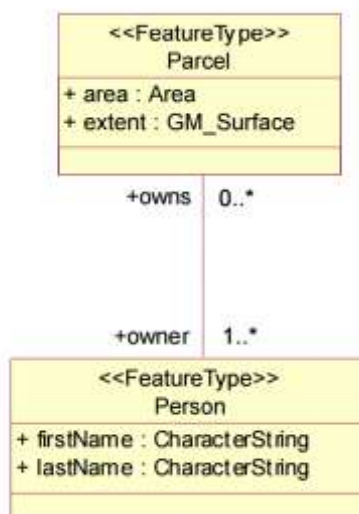
b) Typ danych „Address”



Rysunek 23 - Typ danych "Address"

Do opracowania proszę skorzystać z dostarczonego przez trenera pliku XSD zawierającego szkielet schematu, który należy dokończyć.

c) Klasy obiektów „Parcel” oraz „Person”



Rysunek 24 - Klasy "Parcel" oraz "Person"

Do opracowania proszę skorzystać z dostarczonego przez trenera pliku XSD zawierającego szkielet schematu, który należy dokończyć.

d) Enumeracja "BuildingType"



Rysunek 25 - Enumeracja "BuildingType"

Do opracowania proszę skorzystać z dostarczonego przez trenera pliku XSD zawierającego szkielet schematu, który należy dokończyć.

e) Lista kodowa "CountryCode"



Rysunek 26 - Lista kodowa "CountryCode"

Do opracowania proszę skorzystać z dostarczonego przez trenera pliku XSD zawierającego szkielet schematu, który należy dokończyć.

Rozwiązanie ćwiczenia 3.3:

a) Building

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.someorg.de/example" xmlns="
http://www.w3.org/2001/XMLSchema"
xmlns:ex="http://www.someorg.de/example" xmlns:gml="http://www.opengis.net/gml"
elementFormDefault="qualified"
version="1.0">
  <element name="Building" substitutionGroup="gml:_Feature">
    <complexType>
      <complexContent>
        <extension base="gml:AbstractFeatureType">
          <sequence>
            <element name="extent" type="gml:SurfacePropertyType"/>
            <element name="address">
              <complexType>
                <sequence>
                  <element name="Address" type="ex:AddressType"/>
                </sequence>
              </complexType>
            </element>
            <element name="type" type="ex:BuildingType"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
</schema>
```

b) Address

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.someorg.de/example" xmlns="
http://www.w3.org/2001/XMLSchema"
xmlns:ex="http://www.someorg.de/example" xmlns:gml="http://www.opengis.net/gml"
elementFormDefault="qualified"
version="1.0">
  <complexType name="AddressType">
    <sequence>
      <element name="street" type="string" minOccurs="0"/>
      <element name="houseNumber" type="string" minOccurs="0"/>
      <element name="poBox" type="string" minOccurs="0"/>
      <element name="city" type="string"/>
      <element name="postalCode" type="string"/>
      <element name="country" type="ex:CountryCode" minOccurs="0" default="DE"/>
    </sequence>
  </complexType>
</schema>
```


c) ParcelPerson

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.someorg.de/example" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ex="http://www.someorg.de/example" xmlns:gml="http://www.opengis.net/gml" elementFormDefault=
"qualified"
version="1.0">
  <element name="Parcel" substitutionGroup="gml:_Feature">
    <complexType>
      <complexContent>
        <extension base="gml:AbstractFeatureType">
          <sequence>
            <element name="area" type="gml:AreaType"/>
            <element name="extent" type="gml:SurfacePropertyType"/>
            <element name="owner" type="ex:PersonPropertyType">
              <annotation>
                <appinfo source="urn:x-gml:reverseProperty">owns</appinfo>
              </annotation>
            </element>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <complexType name="ParcelPropertyType">
    <sequence>
      <element ref="ex:Parcel" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <element name="Person" substitutionGroup="gml:_Feature">
    <complexType>
      <complexContent>
        <extension base="gml:AbstractFeatureType">
          <sequence>
            <element name="firstName" type="string"/>
            <element name="lastName" type="string"/>
            <element name="owns" type="ex:ParcelPropertyType">
              <annotation>
                <appinfo source="urn:x-gml:reverseProperty">owner</appinfo>
              </annotation>
            </element>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <complexType name="PersonPropertyType">
    <sequence>
      <element ref="ex:Person" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
</schema>
```


d) BuildingType

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.someorg.de/example" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ex="http://www.someorg.de/example" xmlns:gml="http://www.opengis.net/gml" elementFormDefault=
"qualified"
version="1.0">
  <simpleType name="BuildingTypeType">
    <restriction base="string">
      <enumeration value="church"/>
      <enumeration value="school"/>
      <enumeration value="garage"/>
      <enumeration value="residential houses"/>
      <enumeration value="unknown"/>
      <enumeration value="mixed"/>
    </restriction>
  </simpleType>
</schema>
```

e) CountryCode

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.someorg.de/example" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ex="http://www.someorg.de/example" xmlns:gml="http://www.opengis.net/gml" elementFormDefault=
"qualified"
version="1.0">
  <simpleType name="CountryCodeType">
    <restriction base="string">
      <enumeration value="DE"/>
      <enumeration value="US"/>
      <enumeration value="CA"/>
      <enumeration value="..."/>
    </restriction>
  </simpleType>
</schema>
```

4 Rozdział V – Konwersja danych przestrzennych z powszechnych formatów do GML

4.1 Instruktaż – Transformacja w HALE

Istotną kwestią jest transformacja danych przestrzennych zapisanych zgodnie z jednym schematem aplikacyjnym do dokumentu GML związanego z innym schematem. Proces taki nazywa się harmonizacją danych przestrzennych. Celem ćwiczenia będzie wykonanie prostej transformacji z wykorzystaniem narzędzia HALE.

Konwersja będzie miała na celu przejście z formatu SHP do struktur GML zgodnych z INSPIRE. Ćwiczenie zostanie wykonane w oparciu o schemat dla tematu Hydrografia, dla klasy obiektów DrainageBasin. Wszystkie niezbędne dane wejściowe zostaną przekazane przez prowadzącego.

Niezbędne dane wejściowe:

- Schemat XSD dla tematu Hydrografia – HydroPhysicalWaters.xsd
- Plik SHP zawierający dane środowiskowe dotyczący zlewni – zlewnie.shp

Dane niezbędne do wykonania ćwiczenia:

- Plik SHP zawierający informacje o zlewniach („zlewnie.shp”),
- Plik XSD zawierający docelowy schemat aplikacyjny INSPIRE dla tematu Hydrografia (“HydroPhysicalWaters.xsd”).

4.1.1 Wczytywanie danych źródłowych do aplikacji HALE

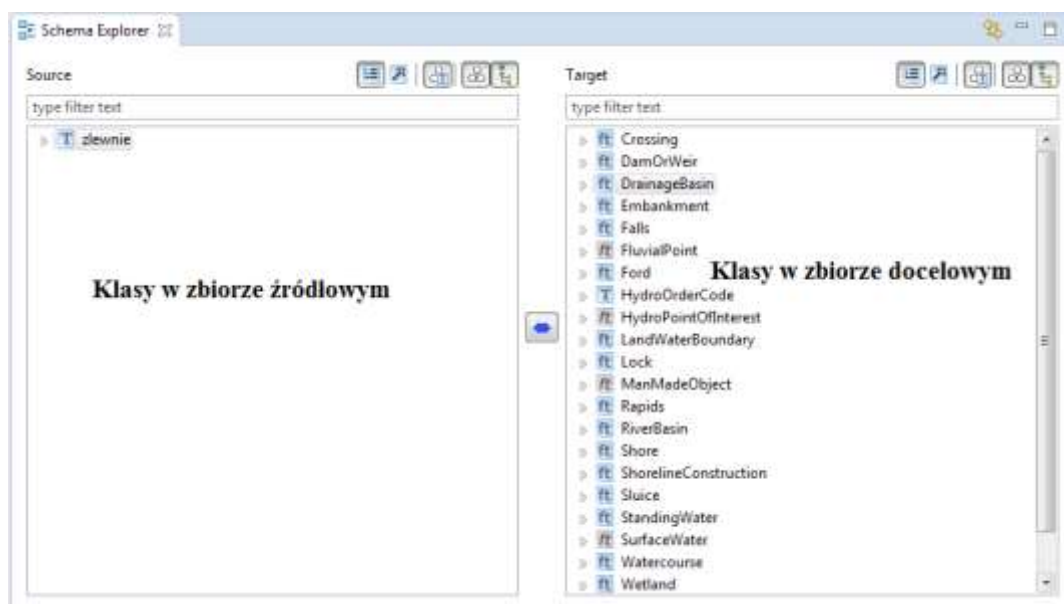
W celu załadowania docelowego schematu aplikacyjnego, z menu górnego wybieramy:

FILE => Import => Target Schemat. Następnie należy wskazać lokalizację pliku XSD ze schematem INSPIRE.

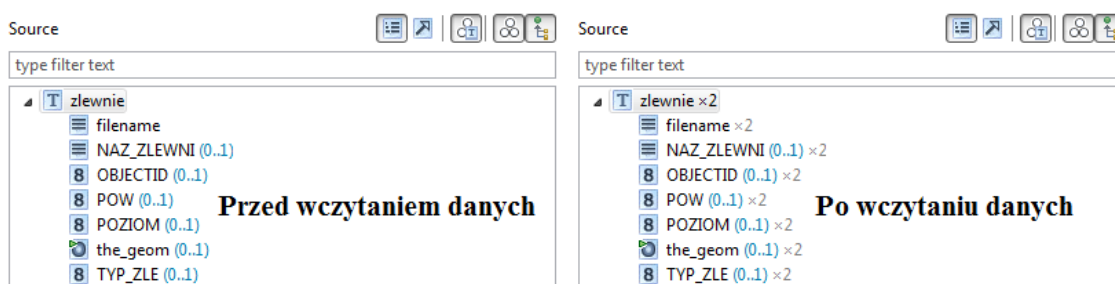
W celu wczytania danych źródłowych, którymi w tym wypadku są dane w formacie SHP musimy w pierwszej kolejności wczytać schemat danych wejściowych, a następnie dopiero same dane. Dla uzyskania schematu danych wejściowych wystarczy wskazać plik SHP, struktura danych zostanie pobrana automatycznie. Schemat: *FILE => Import => Source Schema => From File*, dodatkowo możemy określić kodowanie znaków w zbiorze danych.

W celu załadowania danych źródłowych należy wybrać *FILE => Import => Source Data*. Jest to możliwe dopiero po uprzednim załadowaniu schematu danych wejściowych. Dodatkowo, podobnie jak w przypadku schematu – należy podać kodowanie danych.

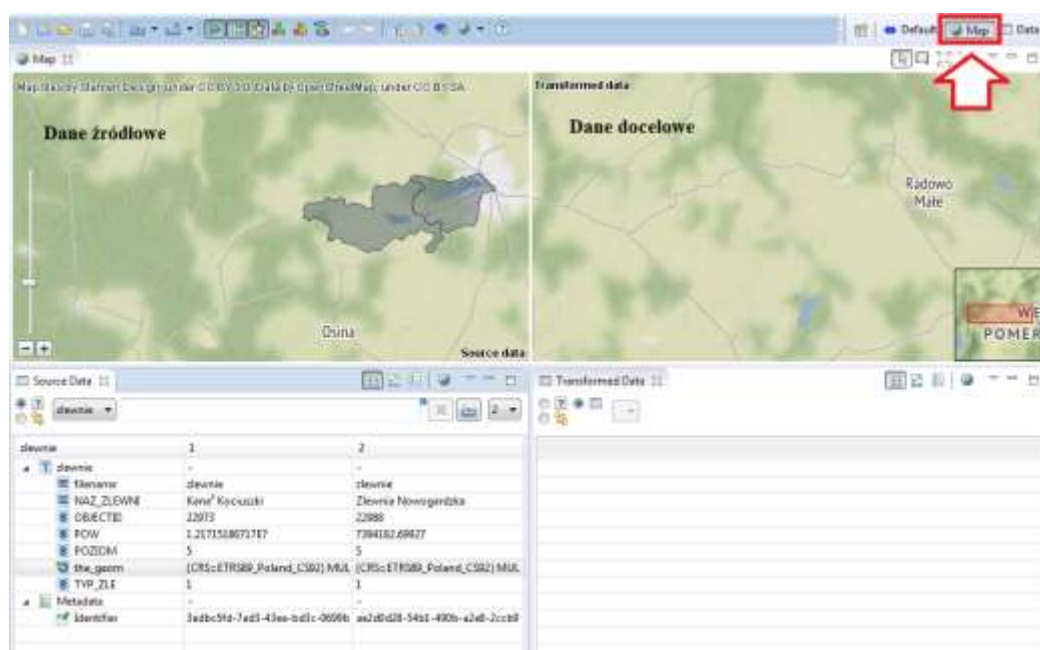
Jeśli wszystko poszło zgodnie z założeniami, po przejściu do zakładki “Map” w aplikacji powinniśmy mieć podgląd na mapę, gdzie wyświetlane będą zarówno dane źródłowe, jak i bieżący stan danych docelowych.



Rysunek 27 - Wczytane schematy aplikacyjne



Rysunek 28 - Załadowanie danych źródłowych



Rysunek 29 - Mapa podglądowa

4.1.2 Analiza zbioru danych oraz tabela mapowania

Pierwszym elementem, jaki należy wykonać jest przeanalizowanie struktury pliku SHP oraz docelowej struktury zbioru w celu uzupełnienia tabeli mapowania. Na cele instruktażowe harmonizować będziemy jedynie jedną klasę obiektów.

Dla ułatwienia poniżej zamieszczona została tabela już uzupełniona o stronę „lewą” z modelem danych INSPIRE. Użytkownik natomiast powinien się skupić na stronie prawej. Dodatkowo można pobrać specyfikację danych dla tematu INSPIRE i skorzystać z zamieszczonego tam katalogu obiektów.

A) Feature Type – DrainageBasin

DrainageBasin	FeatureType/ subtype of hydroObject and link				
Dorzecza.	geographicalName	GeographicalName	voidable 0..*	Złożony typ danych	A geographical name that is used to identify a hydrographic object in the real world. It provides a 'key' for implicitly associating different representations of the Object.
	hydroId	HydroIdentifier	voidable 0..1		An identifier that is used to identify a hydrographic object in the real world. It provides a 'key' for implicitly associating different representations of the object.
	relatedHydroObject	HydroObject	voidable 0..*	association role	A related hydrographic object representing the same real-world feature.
	area	Area	Voidable 1		Size of the drainage basin area.
	basinOrder	HydroOrderCode	voidable 0..1	Złożony typ danych.	Number (or code) expressing the degree of branching/dividing in a drainage basin system.
	beginLifespanVersion	DateTime	Voidable, LifeCycleInfo, 1		Date and time at which this version of the spatial object was inserted or changed in the spatial data set.
	endLifespanVersion	DateTime	Voidable, LifeCycleInfo, 0..1		Date and time at which this version of the spatial object was superseded or retired in the spatial data set.
	geometry	GM_Surface	1		The geometry of the drainage basin, as a surface.
	inspireId	Identifier	1		External object identifier of the spatial object.
	origin	OriginValue	voidable 1		Origin of the drainage basin.
	containsBasin	DrainageBasin	voidable 0..*	Association role	A smaller sub-basin contained within a larger basin.
	outlet	SurfaceWater	voidable 1..*	Association role	The surface water outlet(s) of a drainage basin.

Rysunek 30 - DrainageBasin - model INSPIRE

Poniższą tabelę należy uzupełnić.

Klasa obiektów	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
Dorzecza.	geographicalName			
	hydroId			
	relatedHydroObject			
	area			

	basinOrder			
	beginLifespanVersion			
	endLifespanVersion			
	geometry			
	inspireId			
	origin			
	containsBasin			
	outlet			

Rysunek 31 - Tabela mapowania dla DrainageBasin

B) DataType – HydroOrderCode

HydroOrderCode	FeatureType	Type	Cardinality	Description
A hydrologically meaningful 'order code' for ordering hierarchies of watercourses and drainage basins.	order	CharacterString	1	Number (or code) expressing the degree of branching or dividing in a stream or drainage basin system.
	orderScheme	CharacterString	1	A description of the concept for ordering.
	scope	CharacterString	1	An indicator of the scope or origin for an order code (including whether it is national, supranational or European).

Rysunek 32 - Typ danych - HydroOrderCode

TypDanych	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
Hydro Order Code.	order			
	orderSchema			
	scope			

Rysunek 33 - Tabela mapowania dla HydroOrderCode

C) DataType – GeographicalName

Data type	atrybut	typ	krotność	wartości	definicja
GeographicalName	spelling	SpellingOfName	1..*		
	language	CharacterString	voidable 1		
	nativeness	NativenessValue	voidable 1		Endonym - Name for a geographical feature in an official or well-established language occurring in that area where the feature is situated. exonym - Name used in a specific language for a geographical feature situated outside the area where that language is widely spoken, and differing in form from the respective endonym(s) in the area where the geographical feature is situated.
	nameStatus	NameStatusValue	voidable 1		official - Name in current use and officially approved or established by legislation. Standardised - Name in current use and accepted or recommended by a body assigned advisory function and/or power of decision in matters of toponymy. historical - Historical name not in current use. other - Current, but not official, nor approved name.
	sourceOfName	CharacterString	voidable 1		
	pronunciation	PronunciationName	voidable 1		
	grammaticalGender	grammaticalGenderValue	voidable 0..1		masculine - Masculine grammatical gender. Feminine - Feminine grammatical gender. neuter - Neuter grammatical gender. Common - 'Common' grammatical gender (the merging of 'masculine' and 'feminine').
	grammaticalNumber	grammaticalNumberValue	voidable 0..1		singular - Singular grammatical number. plural - Plural grammatical number. dual - Dual grammatical number.

Rysunek 34 - Tabela mapowania dla GeographicalNames

Klasa obiektów	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
GeographicalName.	spelling			
	language			
	nativeness			
	nameStatus			
	sourceOfName			
	beginLifespanVersion			
	endLifespanVersion			
	pronunciation			
	grammaticalGender			
	grammaticalGender			

Data type	atrybut	typ	krotność	wartości	definicja
SpellingOfName	text	CharacterString			way the name is written
	script	CharacterString	voidable 1		
	transliterationSchema	CharacterString	voidable 0..1		

Rysunek 35 - Tabela mapowania - SpellingOfName

TypDanych	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
Spelling Of Name	<i>text</i>			
	<i>script</i>			
	<i>transliterationSchema</i>			

Rozwiązanie dla ćwiczenia 4.1.2 – gotowe tabele mapowania:

Klasa obiektów	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
Dorzecza.	<i>geographicalName</i>	NAZ_ZLEWNI	złożony	
	<i>hydroId</i>	---		Nie zasilane
	<i>relatedHydroObject</i>	---		Nie zasilane
	<i>area</i>	POW		
	<i>basinOrder</i>	POZIOM	złożony	
	<i>beginLifespanVersion</i>	---		Nie zasilane
	<i>endLifespanVersion</i>	---		Nie zasilane
	<i>geometry</i>	The_geom		
	<i>inspireId</i>	objectid	złożony	
	<i>origin</i>	TYP_ZLE		
	<i>containsBasin</i>	--		Nie zasilane
	<i>outlet</i>	---		Nie zasilane

TypDanych	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
Hydro Order Code.	<i>order</i>	POZIOM		
	<i>orderSchema</i>	„Klasyfikacja szkoleniowa”		Stała wartość
	<i>scope</i>	„10”		Zakres skali

Klasa obiektów	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
GeographicalName.	<i>spelling</i>	NAZ_ZLEWNI	złożony	
	<i>language</i>	„pl”		
	<i>nativeness</i>	„endonym”		

	nameStatus	„official”		
	sourceOfName	--		Nie zasilane
	pronunciation	--		Nie zasilane
	grammaticalGender	--		Nie zasilane
	grammaticalGender	--		Nie zasilane

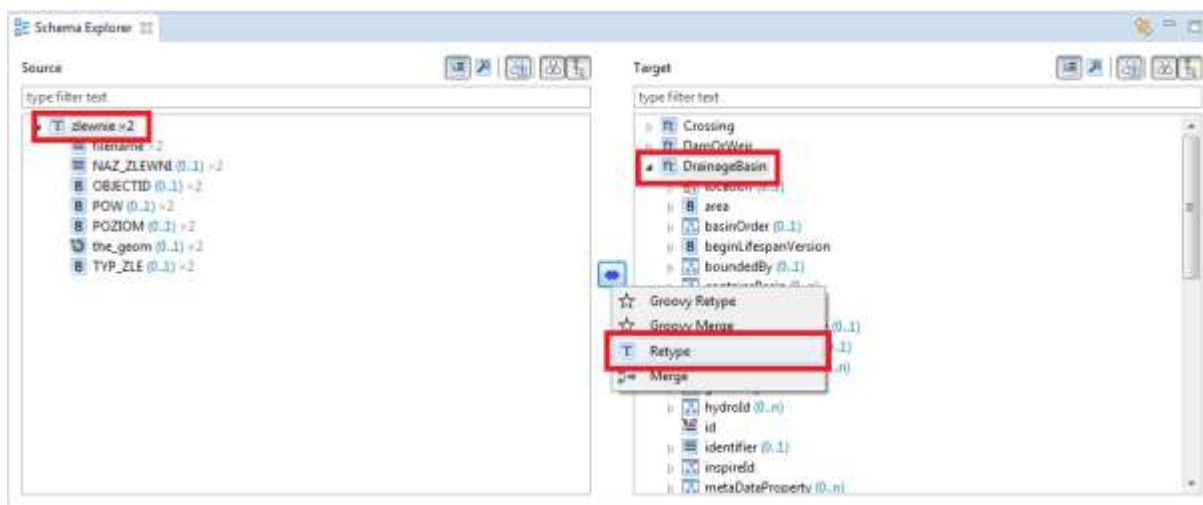
TypDanych	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
Spelling Of Name	<i>text</i>	NAZ_ZLEWNI		
	<i>script</i>	„latn”		Stała wartość
	<i>transliterationSchema</i>	--		Nie zasilane

4.1.3 Implementacja procesu Harmonizacji w HALE

Proces ma na celu opracowanie transformacji w oprogramowaniu, zgodnej z mapowaniem opracowanym w punkcie poprzednim. Należy skupić się nie tylko na tym, w jaki sposób powiązać odpowiadające sobie atrybuty, ale również, co zrobić z wartościami niezasilanymi.

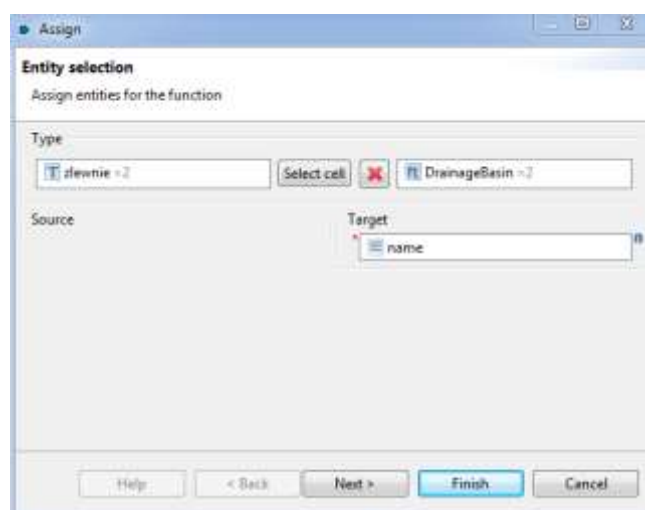
W pierwszej kolejności kilka informacji dotyczących najczęściej wykorzystywanych funkcjonalności oprogramowania:

- a) Przypisanie odpowiadających sobie klas pomiędzy schematami. W tym celu użyć należy funkcji „Retype” w celu połączenia 1:1 lub też „Merge”, w przypadku, gdy kilka klas źródłowych odpowiada jednej klasie docelowej.



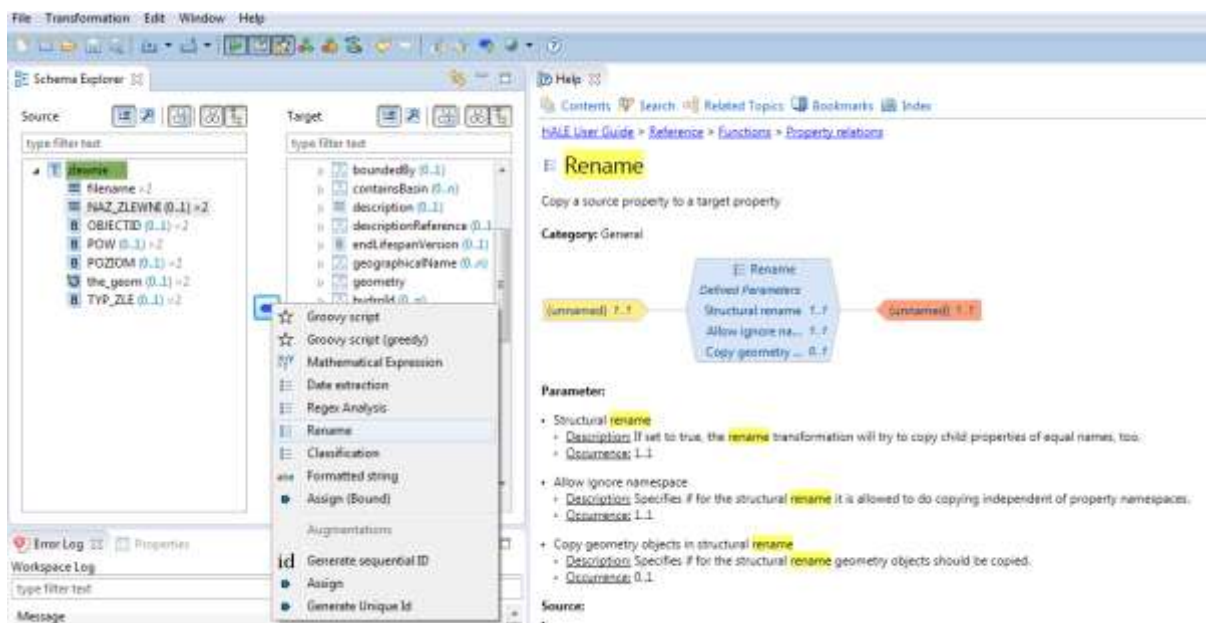
Rysunek 36 - Identyfikacja klas sobie odpowiadających

- b) Uzupełnienie wartości stałych. Uzupełniamy wartości, które można przyjąć za wartość stałą dla wszystkich obiektów w klasie. Przypisanie stałych wartości odbywa się za pomocą polecenia „Assign”, przy czym należy wybrać pole ze schematu wejściowego, które musi być uzupełnione, aby wartość została dodana do schematu docelowego.



Rysunek 37 - Funkcja ASSIGN

- c) Mapowanie atrybutów, których wartości przechodzą pomiędzy schematami w sposób nienaruszony. Mapowanie takie odbywa się poprzez funkcję "Rename", która w praktyce kopiuje wartość ze wskazanego źródłowego atrybutu do docelowego atrybutu. W celu uzyskania okna pomocy dotyczącego jakiegokolwiek funkcji należy wcisnąć F1, a następnie w wyszukiwarce wpisać nazwę funkcji.



Rysunek 38 - Okno schematu wraz z oknem pomocy

- d) Mapowanie atrybutów, w przypadku których w zbiorze docelowym powstaje nowa wartość, np. poprzez konkatencję atrybutów ze zbioru źródłowego. Należy wybrać zasilany atrybut oraz funkcję „Formatted String”. Funkcja buduje wartość na podstawie podanych wzorców oraz zmiennych pochodzących z danych źródłowych.

Formatted string

Entity selection
Assign entities for the function

Type

zlewnie ×2 Select cell DrainageBasin ×2

Source

Variable

OBJECTID ×2

<Click to select>

Target

* inspireId.Identifier.localId

Help < Back Next > Finish Cancel

Formatted string

Format pattern
Enter a pattern

JAKIS_WZORZEC_{OBJECTID}

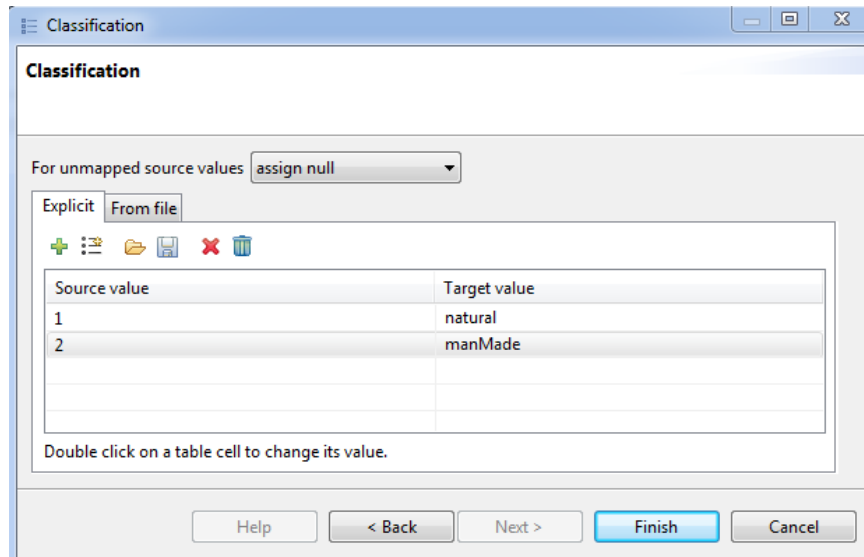
Available variables (double click to insert)

Entity	Variable
OBJECTID (0..1) ×2	{OBJECTID}

Help < Back Next > Finish Cancel

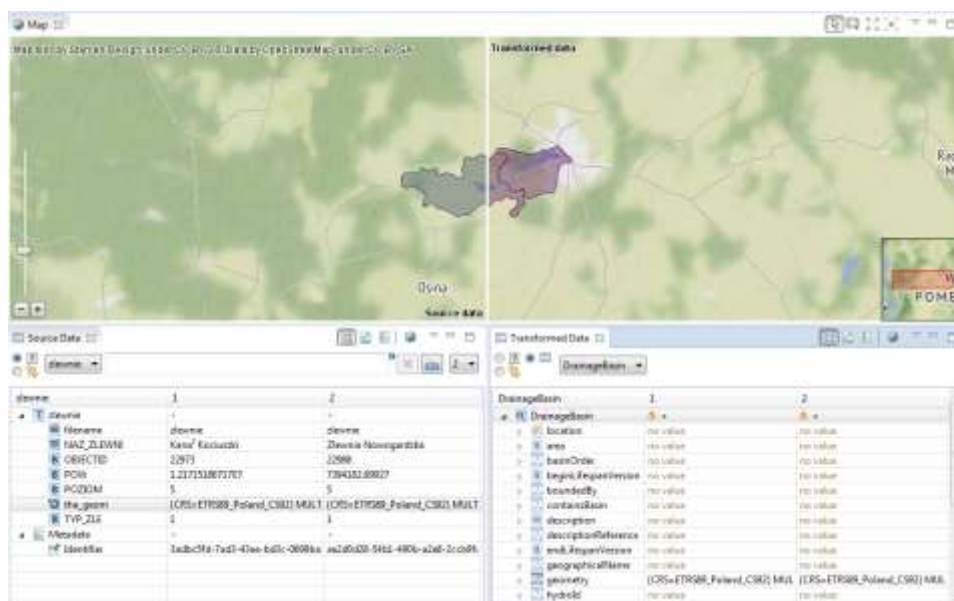
Rysunek 39 - Formatted String - przykład użycia

- e) Mapowanie list kodowych. Dotyczy sytuacji, gdy wartości w obu schematach zapisane są w sposób wyliczeniowy, jednak listy te nie są tożsame, w związku z czym należy przeprowadzić mapowanie pomiędzy wartościami list. Do tego celu wykorzystuje się funkcję "Classification". Klasyfikację możemy utworzyć od podstaw ręcznie, bądź wczytać gotową z pliku.



Rysunek 40 - Przykład klasyfikatora

- f) Mapowanie geometrii pomiędzy schematami. Należy wskazać odpowiadające sobie atrybuty zawierające informacje przestrzenne. Dla zasilonej geometrii w zbiorze docelowym, należy z menu kontekstowego wybrać "Set as default geometry" – dzięki czemu geometria powinna być widoczna również dla zbioru docelowego w panelu mapy.

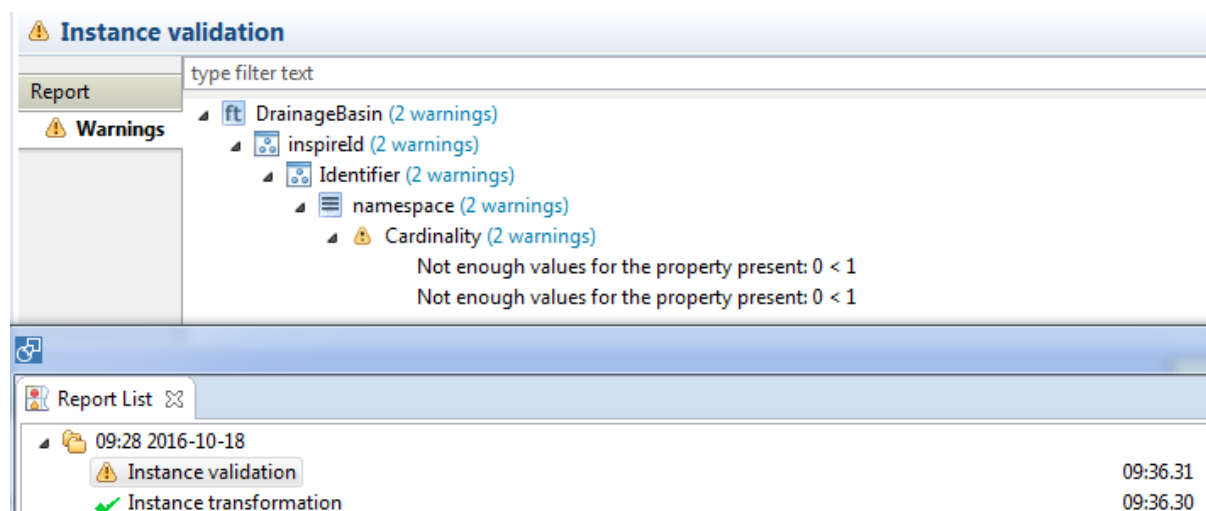


Rysunek 41 - Panel podglądu

Kolejnym krokiem powinno być uzupełnienie schematu w oprogramowaniu HALE zgodnie z przyjętym mapowaniem. W celu komfortowej pracy z menu TRANSFORMATION należy mieć włączone dwie opcje: „Live transformation” oraz „validate transformed instances”.

Live transformation – funkcjonalność, dzięki której na bieżąco widzimy postępy swojej pracy. Dane przekształcane są w locie zgodnie ze zdefiniowanymi przez użytkownika zależnościami. Po każdej wprowadzonej zmianie dane są transformowane, a ich podgląd możliwy do wyświetlenia.

Validate transformed instances – pozostawienie tej funkcji aktywnej pozwala aplikacji na walidację rezultatów po każdej, nawet drobnej, zmianie wprowadzonej w mapowaniu. W przypadku błędów, wyświetla listę niezgodności względem zakładanego modelu.



Rysunek 42 - Rezultat automatycznej walidacji HALE

Rozwiązanie ćwiczenia 4.1.3 – przykładowy proces dostosowawczy:



Rysunek 43 - Schemat przykładowego procesu dostosowawczego

4.2 Ćwiczenie – Samodzielne przejście procesu harmonizacji w HALE

Konwersja będzie miała na celu przejście z formatu SHP do struktur GML zgodnych z INSPIRE. Ćwiczenie zostanie wykonane w oparciu o schemat dla tematu Hydrografia, dla klasy obiektów StandingWater. Wszystkie niezbędne dane wejściowe zostaną przekazane przez prowadzącego.

Niezbędne dane wejściowe:

- Schemat XSD dla tematu Hydrografia – HydroPhysicalWaters.xsd
- Plik SHP zawierający dane środowiskowe dotyczący zlewni – jeziora.shp

W toku ćwiczenia uczestnik będzie musiał wykonać mapowanie danych do schematu docelowego oraz następnie zaimplementować to rozwiązanie w aplikacji HALE tak, aby w rezultacie otrzymać plik GML zgodny ze schematem aplikacyjnym.

Schemat danych źródłowych:

- objectId – identyfikator obiektu w pliku GML,
- ID_JEZ – unikalny identyfikator obiektu,
- nazwa – nazwa jeziora,
- pow – powierzchnia zbiornika wodnego podana w metrach kwadratowych,
- obwod – obwód zbiornika wodnego podany w metrach,
- wys_npm – wysokość nad poziom morza lustra wody,
- głębokosc – średnia głębokość jeziora podana w metrach,
- skala – wartość skali odpowiadająca poziomowi szczegółowości danych wektorowych.

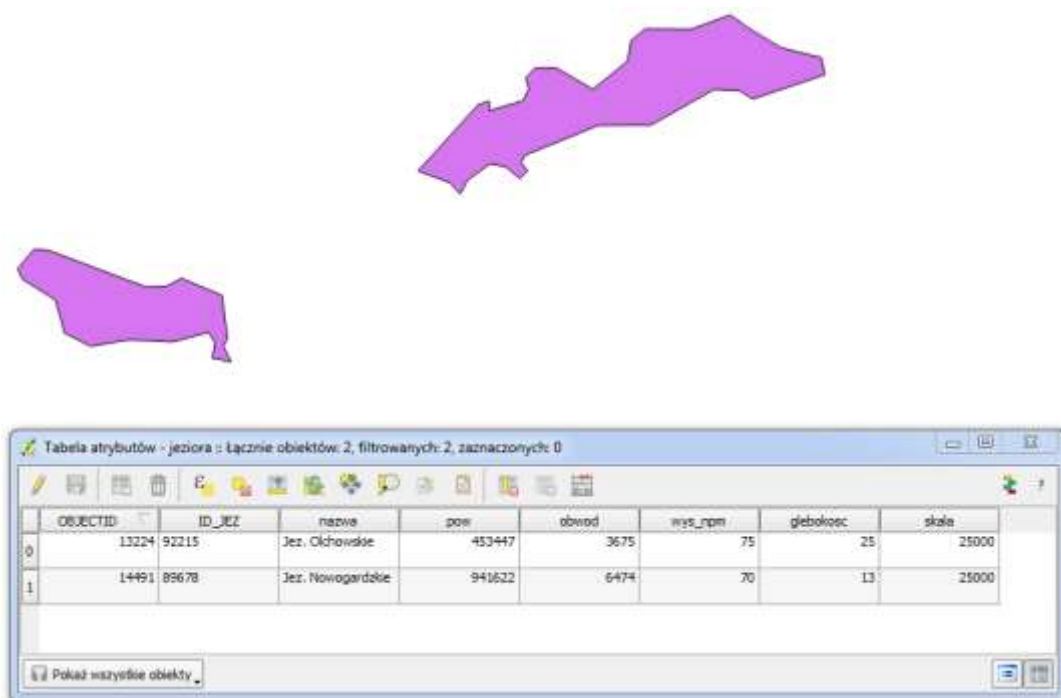


Tabela atrybutów - jeziora : łącznie obiektów: 2, filtrowanych: 2, zaznaczonych: 0

	OBJECTID	ID_JEZ	nazwa	pow	obwod	wys_npm	głębokość	skala
0	13224	92215	Jez. Okchowskie	453447	3675	75	25	25000
1	14491	89678	Jez. Nowogardzkie	941622	6474	70	13	25000

Rysunek 44 - Źródłowy zbiór danych

Dla ułatwienia poniżej zawarto lewą stronę tabeli mapowania wraz z wyróżnieniem, które elementy będą niemapowane.

Standing Water wody stojące	<i>geographicalName</i>	Geographical-Name	voidable 0..*	Data Type	A geographical name that is used to identify a hydrographic object in the real world. It provides a 'key' for implicitly associating different representations of the Object.
	<i>hydroId</i>	HydroIdentifier	voidable 0..1		An identifier that is used to identify a hydrographic object in the real world. It provides a 'key' for implicitly associating different representations of the object.
	<i>relatedHydroObject</i>	HydroObject	voidable 0..*	association role	A related hydrographic object representing the same real-world feature.
	<i>beginLifespanVersion</i>	DateTime	Voidable, LifeCycleInfo, 1		<i>Date and time at which this version of the spatial object was inserted or changed in the spatial data set.</i>
	<i>endLifespanVersion</i>	DateTime	Voidable, LifeCycleInfo, 0..1		<i>Date and time at which this version of the spatial object was superseded or retired in the spatial data set.</i>
	<i>geometry</i>	GM_Primitive	1		<i>The geometry of the surface water: - either a curve or surface for a watercourse; - either a point or surface for a standing water.</i>
	<i>inspireId</i>	Identifier	1		External object identifier of the spatial object.
	<i>levelOfDetail</i>	MD_Resolution	0..1		Resolution, expressed as the inverse of an indicative scale or a ground distance.
	<i>localType</i>	LocalisedCharacterString	voidable 0..1		Provides 'local' name for the type of surface water.
	<i>origin</i>	OriginValue	1	Natural man made	Origin of the surface water.
	<i>persistence</i>	HydrologicalPersistenceValue	voidable 1	Dry ephemeral intermittent perennial	The degree of persistence of water.
	<i>tidal</i>	Boolean	voidable 1		Identifies whether the surface water is affected by tidal water.
	<i>bank</i>	Shore	voidable 0..*	Association role	<i>The bank(s) associated to a surface water.</i>
	<i>drainsBasin</i>	DrainageBasin	voidable 1..*	Association role	The basin(s) drained by a surface water.
	<i>neighbour</i>	SurfaceWater	voidable 0..*	Association role	<i>An association to another instance of the same real-world surface water in another data set.</i>
	<i>elevation</i>	Length	voidable 1		Elevation above mean sea level.
	<i>meanDepth</i>	Length	voidable 1		Average depth of the body of water.
	<i>surfaceArea</i>	Area	voidable 1		Surface area of the body of water.

Klasa obiektów	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
StandingWater	<i>geographicalName</i>			
	<i>hydroid</i>			
	<i>relatedHydroObject</i>			
	beginLifespanVersion			
	endLifespanVersion			
	geometry			
	inspireId			
	levelOfDetail			
	localType			
	origin			
	persistence			
	tidal			
	bank			
	drainsBasin			
	neighbour			
	elevation			
	meanDepth			
	surfaceArea			

Definicje "GeographicalName" oraz "SpellingOfName" należy znaleźć powyżej w zadaniu 4.1 – rysunki 31 oraz 32.

Klasa obiektów	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
GeographicalName.	<i>spelling</i>			
	<i>language</i>			
	<i>nativeness</i>			
	<i>nameStatus</i>			
	<i>sourceOfName</i>			
	<i>pronunciation</i>			
	<i>grammaticalGender</i>			
	<i>grammaticalGender</i>			

TypDanych	Atrybut INSPIRE	Atrybut źródłowy	Typ danych	Komentarz
Spelling Of Name	<i>text</i>			
	<i>script</i>			
	<i>transliterationSchema</i>			

5 Walidacja plików GML względem schematu XSD

Celem ćwiczenia jest przeprowadzenie walidacji pliku GML względem schematu aplikacyjnego INSPIRE zapisanego w formacie XSD. Walidacja zostanie przeprowadzona na pojedynczym pliku z użyciem aplikacji Notepad++, jak również w sposób wsadowy za pomocą skryptu przygotowanego w ćwiczeniu opisanym w punkcie 5.1.

5.1 Instrukcja – Walidacja pliku GML w edytorze tekstowym Notepad++

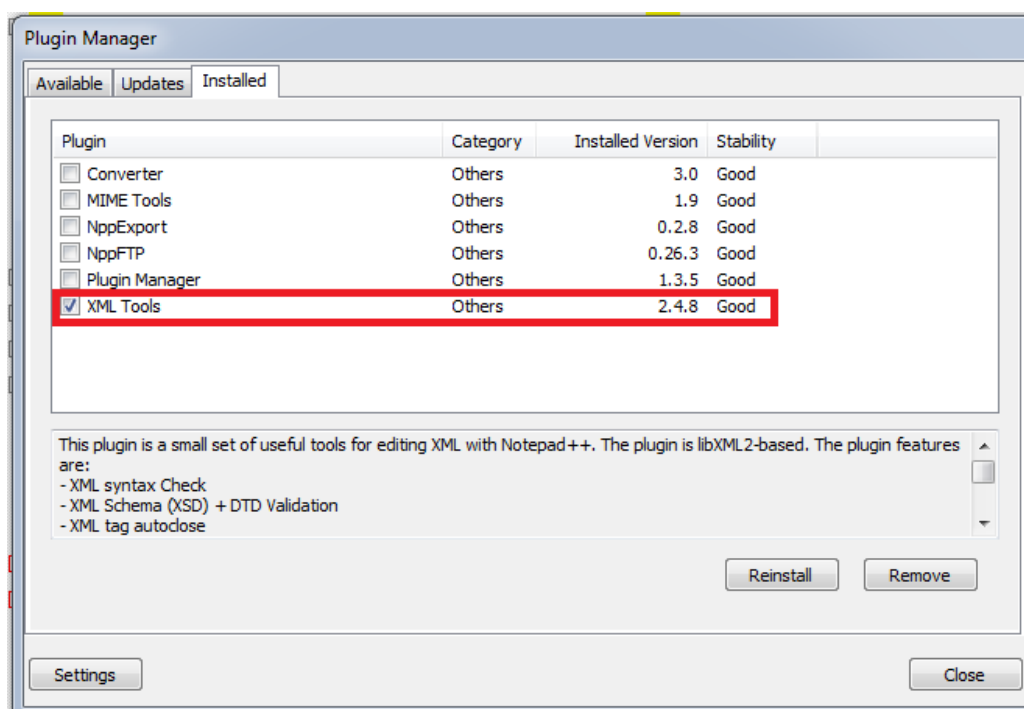
Prowadzący przeprowadzi uczestników krok po kroku przez proces walidacji dokumentu.

UWAGA!

W przypadku walidacji plików GML zgodnych z INSPIRE niezbędne jest połączenie z Internetem. Wynika to stąd, że plik XSD w swojej treści odwołuje się do wielu innych schematów, które walidator pobiera w tle w celu przeprowadzenia kontroli.

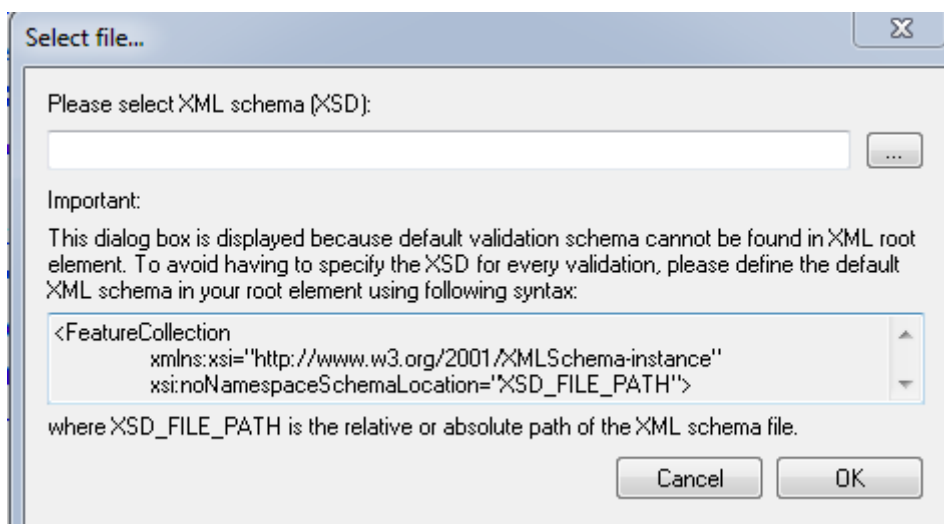
Główny przepływ zdarzeń:

1. Uruchamiamy aplikację Notepad++. Aplikacja powinna posiadać zainstalowane rozszerzenie XML Tools, które umożliwia bardziej zaawansowane przetwarzanie danych XML. Jeśli moduł ten nie jest widoczny w menu "Wtyczki" to znaczy, że należy go doinstalować korzystając z Plugin Managera (Wtyczki => Plugin Manager).



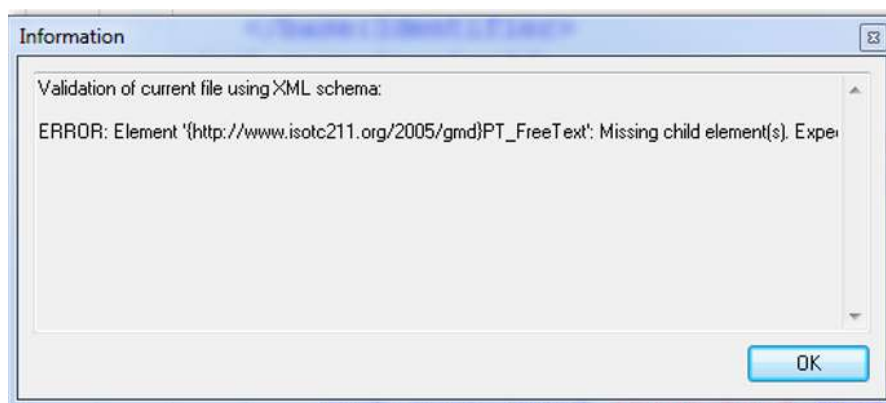
Rysunek 45 - Aplikacja Notepad++ wraz z rozszerzeniem XML Tools

2. Mając otwarty plik GML należy przejść do narzędzi XML Tools i wybrać „Validate Now”. W pierwszej kolejności plik zostanie sparsowany i sprawdzony składniowo. Jeśli posiada jakieś błędy syntaktyczne zostanie wyświetlona informacja na ten temat. W efekcie powinno ukazać się okno z prośbą o wskazanie pliku XSD.



Rysunek 46 - Notepad++ - prośba o wskazanie XSD

3. Po wskazaniu pliku rozpocznie się proces walidacji. Długość trwania jest uzależniona od rozmiaru pliku sprawdzanego, jak i prędkości łącza internetowego. Zalecane jest sprawdzanie w ten sposób plików nie większych niż 25 MB. Po sprawdzeniu ukazać się powinno okno z rezultatem walidacji. Jeśli walidacja zakończyła się niepowodzeniem – wyświetlony zostanie komunikat o błędzie.



Rysunek 47 - Notepad++ - informacja o błędzie

5.2 Ćwiczenie – Walidacja pliku GML – tryb wsadowy

Do przeprowadzenia ćwiczenia niezbędne jest stałe połączenie z Internetem. Przeprowadzenie walidacji w trybie offline jest niemożliwe.

5.2.1 Walidacja pojedynczego pliku GML

Ćwiczenie ma na celu zapoznanie się z procesem konfiguracji pliku wsadowego, jak i interpretacją otrzymanych rezultatów. Komplet plików niezbędnych do wykonania zadania zostanie przekazany przez trenera uczestnikom.

W pierwszej kolejności należy prawidłowo uzupełnić plik konfiguracyjny "walidacja.bat". Plik ten należy uzupełnić o następujące elementy:

- Ścieżka do folderu z plikami XML/GML,
- Ścieżka do pliku "AltovaXML_COM.exe", znajdującego się w lokalizacji, gdzie zainstalowany został program Altova XMLSpy Community Edition (oprogramowanie darmowe),
- Ścieżka do pliku XSD, w oparciu o który ma zostać przeprowadzona walidacja,
- Nazwa pliku tekstowego, w którym zapisane zostaną wyniki walidacji.

Strukturę całego pliku BAT można przedstawić w sposób następujący:

```
@echo on
:: PĘTLA FOR - WYKONAJ ITERACYJNIE DLA WSZYSTKICH PLIKÓW W FOLDERZE
for %%a in (C:\ŚCIEŻKA_DO_FOLDERU\*.xml) do (
:: ZAPISZ DO PLIKU SŁOWO "WALIDACJA"
echo Walidacja: "%%a": >> PLIK_WYNIKOWY.txt
:: ZWALIDUJ PLIK ZA POMOCĄ XSD - WYNIK ZAPISZ DO PLIKU WYNIKOWEGO
"C:\SCIEZKA_DO_OPROGRAMOWANIA\AltovaXML_COM.exe" /validate "%%a"
/schema "C:\SCIEZKA_DO_SCHEMATU\NAZWA_PLIKU.xsd" >>
PLIK_WYNIKOWY.txt)
pause
```

Rysunek 48 - Skrypt walidacyjny - zapis ogólny

Weryfikację poprawności uzupełnienia danych możemy wykonać poprzez uruchomienie skryptu, jeśli proces rozpocznie się – oznacza to, że wszystko jest wpisane prawidłowo. Dla uproszczenia w ćwiczeniu plik XML, XSD oraz BAT znajdują się z tym samym folderze.

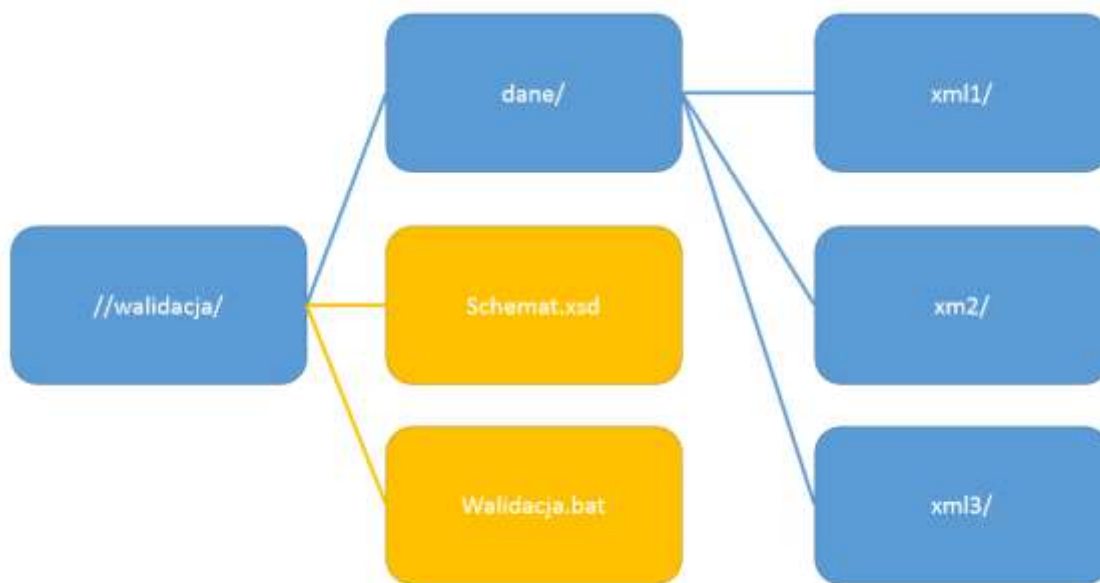
5.2.2 Walidacja plików GML [Zaawansowane]

Ćwiczenie ma na celu przeprowadzenie użytkownika przez bardziej skomplikowany proces walidacji, bliższy sytuacji z życia codziennego. Uczestnicy szkolenia otrzymają od trenera komplet niezbędnych plików.

Przedstawienie zadania:

- Uczestnik posiada zbiór plików w formatach XML oraz GML, które musi zweryfikować pod kątem zgodności z modelem INSPIRE (czyli poddać walidacji).
- Pliki te są zlokalizowane w kilku folderach na dysku, kilka z nich posiada błędy.
- Uczestnik ma za zadanie tak zmodyfikować plik "walidacja.bat", aby zwalidować wszystkie pliki oraz być w stanie wskazać pliki niezgodne ze specyfikacją.

Strukturę plików można przedstawić w sposób następujący:



Rysunek 49 - Struktura plików i katalogów ćwiczeniowych

Zadanie rozwiązać można na wiele sposobów, na zakończenie ćwiczenia prowadzący pokaże oraz przeanalizuje jedno z nich.

Rozwiązanie ćwiczenia 5.2.1:

```
@echo on
for %%a in (*.xml) do (
echo Walidacja: "%%a": >> Wyniki_walidacji.txt
"C:\Program Files (x86)\Altova\AltovaXML2013\AltovaXML_COM.exe" /validate "%%a"
/schema "..\XSD\EnvironmentalManagementFacilities.xsd" >> Wyniki_walidacji.txt)
pause
```

Rozwiązanie ćwiczenia 5.2.2:

```
@echo on
for %%a in (dane\xml1\*.xml) do (
echo Walidacja: "%%a": >> Wyniki_XML1.txt
"C:\Program Files (x86)\Altova\AltovaXML2013\AltovaXML_COM.exe"
/validate "%%a" /schema "schemat.xsd" >> Wyniki_XML1.txt)

for %%a in (dane\xml2\*.xml) do (
echo Walidacja: "%%a": >> Wyniki_XML2.txt
"C:\Program Files (x86)\Altova\AltovaXML2013\AltovaXML_COM.exe"
/validate "%%a" /schema "schemat.xsd" >> Wyniki_XML2.txt)

for %%a in (dane\xml3\*.xml) do (
echo Walidacja: "%%a": >> Wyniki_XML3.txt
"C:\Program Files (x86)\Altova\AltovaXML2013\AltovaXML_COM.exe"
/validate "%%a" /schema "schemat.xsd" >> Wyniki_XML3.txt)
pause
```