



# GML – WARSZTATY PRAKTYCZNE W ZAKRESIE NIEZBĘDNYM DO REALIZACJI ZADAŃ ZWIĄZANYCH Z INSPIRE

Przemysław Biegański, Michał Kubicki  
Wrocławski Instytut Zastosowań Informatyki  
Przestrzennej i Sztucznej inteligencji



Warszawa, październik 2016 r



# CEL I ZAKRES SZKOLENIA

- Wprowadzenie do GML
- Podstawy języka GML
- Rola GML w INSPIRE
- Słowniki i listy kodowe GML
- Mapowanie schematu aplikacyjnego UML do schematu GML, zgodnie z zasadami zdefiniowanymi w aneksie E ISO 19136
- Konwersja danych przestrzennych z powszechnych formatów do struktur GML
- Walidacja GML względem schematu aplikacyjnego GML



WROCLAW GLOWNY

budime  
5

# WPROWADZENIE DO GML



# STANDARYZACJA

- Standaryzacja umożliwia porozumiewanie się w warunkach różnorodności.
- Celem standaryzacji danych przestrzennych jest zapewnienie ich odpowiedniej jakości przy uwzględnieniu postępu technologicznego oraz rosnących wymagań ze strony użytkowników.
- Standaryzacja w GISie – zapewnienie interoperacyjności systemów.



# INTEROPERACYJNOŚĆ

- Wspólny protokół komunikacji (np. HTTP jako protokół transportowy, standardy OGC).
- Wspólny format przesyłanych danych (dane binarne, **format XML** dla wymiany danych w sieci WWW).
- Wspólny schemat aplikacyjny.





# GML A XML

- Wspólny protokół komunikacji (np. HTTP jako protokół transportowy, **standardy OGC**).
- Wspólny format przesyłanych danych (dane binarne, **format XML** dla wymiany danych w sieci WWW).
- Wspólny **schemat aplikacyjny**.



<http://0.tqn.com/d/pcsupport/1/S/h/Y/-/-/xml-file.png>



# GML A XML

- XML jest naturalnym formatem wymiany danych w sieci WWW.
- Konieczna stała się standaryzacja sposobu kodowania danych przestrzennych w formacie XML.
- Takim standardem stał się GML.



<https://cdn1.iconfinder.com/data/icons/file-format-set/64/2547-256.png>



# CZYM JEST GML

- Geography Markup Language.
- Opracowany przez OGC, oparty na XML język dziedziny służący do opisu danych przestrzennych.
- Format wymiany danych przestrzennych pomiędzy różnymi systemami GIS w tym również za pośrednictwem sieci WWW.





# CZYM JEST GML

*"reprezentacja XML służąca do transportu i przechowywania informacji geograficznej łączącej zarówno właściwości przestrzenne jak i nieprzestrzenne obiektów geograficznych"*



# CZYM JEST GML

Definicje:

- obiektów przestrzennych,
- typów geometrycznych.

Definicje obiektów przestrzennych wykorzystywanych w aplikacji.

Instancje obiektów przestrzennych



gml.xsd



schemat aplikacyjny  
GML Schema



plik GML



# CZYM JEST GML

- Transport danych w sieci WWW – problem objętości plików.
- Możliwość wymiany danych pomiędzy dwoma stronami znającymi schemat aplikacyjny.
- Schematy XSD zapewniają walidację wymienianych danych.
- Możliwość łatwej transformacji w inny schemat aplikacyjnych (harmonizacja danych).



# CZYM NIE JEST GML

*GML **nie** jest językiem prezentacji danych przestrzennych.*

- GML nie definiuje sposobu wyświetlania danych przestrzennych.
- Aplikacje mogą dowolnie interpretować i wyświetlać zawartość plików GML.
- Możliwa jest transformacja do innych formatów gotowych do wyświetlenia:
  - HTML (atrybuty poza geometryczne),
  - SVG (grafika wektorowa).



# CZYM NIE JEST GML

*GML **nie** jest językiem przetwarzania danych przestrzennych.*

- GML jest oparty na XML, który jest tylko językiem opisu.
- Nie można za jego pomocą przeprowadzać analiz przestrzennych.
- Konieczne jest wykorzystanie wyspecjalizowanych aplikacji GIS.



# PODSTAWY JĘZYKA GML





# DLACZEGO GML?

## PLUSY:

- Otwarty format oparty na XML.
- Silnie wspierany przez konsorcjum OGC.
- W znacznym stopniu wspiera interoperacyjność systemów.
- Szeroki zakres modelowanych aspektów.
- Podejście obiektowe.

## MINUSY:

- Wysoka złożoność
- Skomplikowane przejście z modelu relacyjnego
- Złożone schematy aplikacyjne



# OBIEKT PRZESTRZENNY

Obiekt przestrzenny (GML Feature) może zawierać:

- atrybuty przestrzenne (geometria itp.),
- atrybuty nieprzestrzenne (identyfikatory, nazwy, liczby itp.).

W relacyjnym GISie reprezentowany jako wiersz w odpowiednio zdefiniowanej tabeli z kolumną geometryczną.



# OBIEKT PRZESTRZENNY

```
<complexType name="ObiektPrzestrzennyType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <!-- atrybuty obiektu przestrzennego -->
    </extension>
  </complexContent>
</complexType>
```

Każdy obiekt przestrzenny zawarty w schemacie aplikacyjnym bazuje oraz rozszerza **gml:AbstractFeature**.



# OBIĘKT PRZESTRZENNY – PRZYKŁAD XSD

```
<element name="Budynek" substitutionGroup="gml:AbstractFeature" ← Wymienialność
type="test:BudynekType"/>
  <complexType name="BudynekType">
    <complexContent>
      <extension base="gml:AbstractFeatureType" ← Rozszerzenie
        <sequence>
          <element maxOccurs="8" minOccurs="8" name="identyfikator" type="string"/>
          <element name="liczba_kondygnacji" type="integer"/>
          <element name="nazwa" type="string"/>
          <element name="obszar" type="gml:SurfacePropertyType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="BudynekPropertyType">
    <sequence minOccurs="0">
      <element ref="test:Budynek"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  </complexType>
```



# OBIEKT PRZESTRZENNY – PRZYKŁAD GML

```
<Budynek>
  <identyfikator>00000001</identyfikator>
  <liczba_kondygnacji>2</liczba_kondygnacji>
  <nazwa>Przykładowy budynek</nazwa>
  <obszar>
```

Pełen zapis geometrii

```
    <gml:Polygon srsName="urn:ogc:def:crs:EPSG::2177">
      <gml:outerBoundaryIs>
        <gml:LinearRing>
          <gml:coordinates>5466762.7163669551,5906412.1436833404
5466763.6555850096,5906414.9627388529 5466767.3497115085,5906413.7506035957
5466766.4119072855,5906410.8822248299
5466762.7163669551,5906412.1436833404</gml:coordinates>
        </gml:LinearRing>
      </gml:outerBoundaryIs>
    </gml:Polygon>
```

```
  </obszar>
</Budynek>
```

Współrzędne obiektu



# KOLEKCJA OBIEKTÓW PRZESTRZENNYCH

W celu utworzenia hierarchii (zawierania) obiektów przestrzennych należy wykorzystać element potomny rozszerzający element GML **`gml:AbstractMemberType`**.

```
<complexType name="KolekcjaObiektowPrzestrzennychType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <element maxOccurs="unbounded" minOccurs="0" name="obiekt-przestrzenny">
        <extension base="gml:AbstractMemberType">
          <sequence>
            <element ref="ObiektPrzestrzenny"/>
          </sequence>
        </extension>
      </element>
    </extension>
  </complexContent>
</complexType>
```





# KOLEKCJA OBIEKTÓW PRZESTRZENNYCH

```
<KolekcjaObiektowPrzestrzennych>  
  <obiekt-przestrzenny>  
    <Tresc>  
    </Tresc>  
  </obiekt-przestrzenny>  
  <obiekt-przestrzenny>  
    <Tresc>  
    </Tresc>  
  </obiekt-przestrzenny>  
  <obiekt-przestrzenny>  
    <Tresc>  
    </Tresc>  
  </obiekt-przestrzenny>  
</KolekcjaObiektowPrzestrzennych>
```

member

collection



# GML – ATRYBUTY

Atrybuty w plikach GML możemy podzielić na:

- nieprzestrzenne – typy proste, złożone, enumeracje. Definiowane w sposób standardowy w XML Schema.
- Przestrzenne – elementy reprezentujące geometrię są typu `gml:AbstractGeometryType`.



# GML – GEOMETRIA

```
<complexType name="prostokat">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <sequence>
        <element name="x1" type="double"/>
        <element name="y1" type="double"/>
        <element name="x2" type="double"/>
        <element name="y2" type="double"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Definicja w XML Schema

Faktyczny  
obiekt w GML

```
<prostokat srsName="urn:ogc:def:crs:EPSG::4326"
srsDimension="2">
  <x1>14.07</x1>
  <y1>54.50</y1>
  <x2>24.09</x2>
  <y2>49.00</y2>
</prostokat>
```



# TYPY GEOMETRYCZNE

Podstawowe figury geometryczne  
(GML Primitives) to m.in.:

- Punkt (gml:Point)
- Łamana (gml:LineString)
- Pierścień (gml:LinearRing)
- Wielokąt (gml:Polygon)

Ale GML pozwala na więcej...

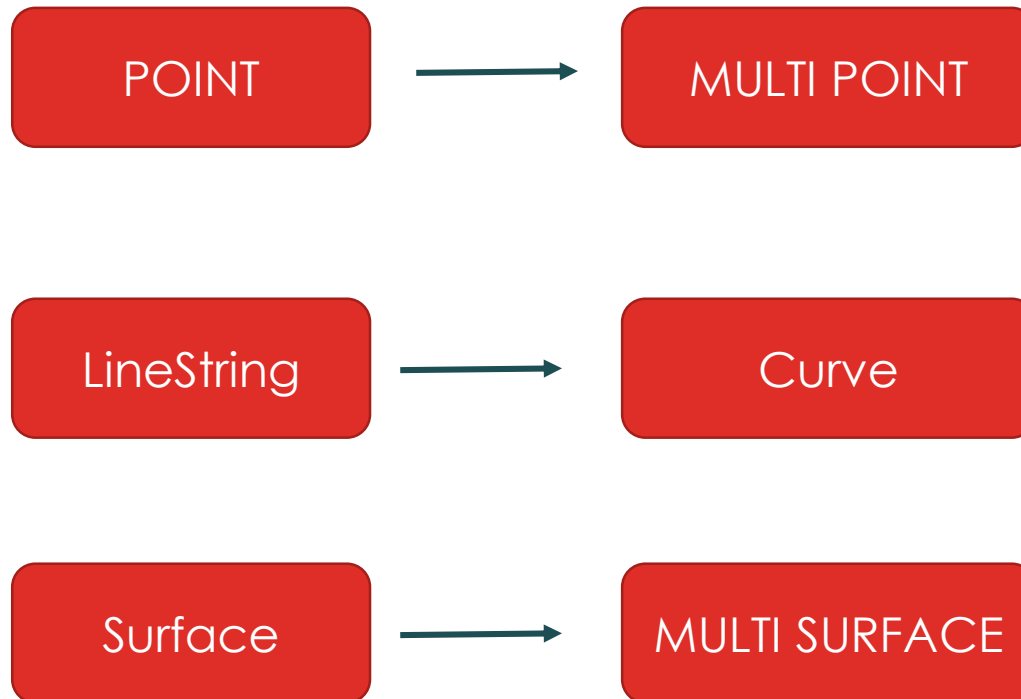


# TYPY GEOMETRYCZNE

gml:Curve	gml:CompositeSurface	gml:Clothoid	
gml:Surface	gml:CompositeSolid	gml:CubicSpline	
gml:PolyhedralSurface	gml:GeometricComplex	gml:GeodesicString	
gml:TriangulatedSurface	gml:MultiGeometry	gml:LineStringSegment	
gml:Tin	gml:MultiPoint	gml:OffsetCurve	
gml:Solid	gml:MultiCurve	gml:AbstractSurfacePatch	
gml:OrientableCurve	gml:MultiSurface	gml:AbstractGriddedSurface	
gml:OrientableSurface	gml:MultiSolid	gml:AbstractParametricCurveSurface	
gml:Ring	gml:MultiGeometry	gml:Cone	
gml:Shell	gml:AbstractCurveSegment	gml:Cylinder	gml:Bezier
gml:LineString	gml:Arc	gml:Geodesic	gml:BSpline
gml:Polygon	gml:ArcByBulge	gml:PolygonPatch	gml:Circle
gml:LinearRing	gml:ArcByCenterPoint	gml:Rectangle	gml:CircleByCenterPoint
gml:Point	gml:ArcString	gml:Sphere	
gml:CompositeCurve	gml:ArcStringByBulge	gml:Triangle	



# GEOMETRIA – AGREGACJE







# GEOMETRIA GML:MULTIPOINT

```
<gml:MultiPoint>

  <gml:pointMember>
    <gml:Point>
      <gml:pos>0.0 0.0</gml:pos>
    </gml:Point>
  </gml:pointMember>

  <gml:pointMember>
    <gml:Point>
      <gml:pos>1.0 1.0</gml:pos>
    </gml:Point>
  </gml:pointMember>

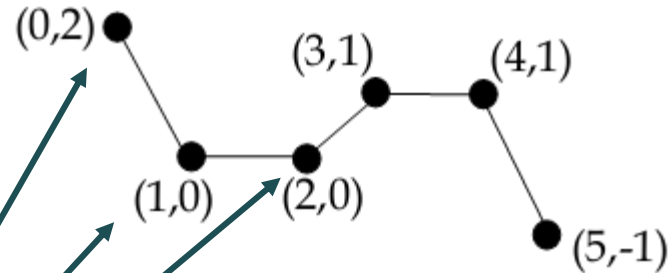
</gml:MultiPoint>
```

← Agregat typu  
punktowego.

← segment



# GEOMETRIA GML:LINESTRING



```
<gml:LineString>  
  <gml:pos>0 2</gml:pos>  
  <gml:pos>1 0</gml:pos>  
  <gml:pos>2 0</gml:pos>  
  <gml:pos>3 1</gml:pos>  
  <gml:pos>4 1</gml:pos>  
  <gml:pos>5 -1</gml:pos>  
</gml:LineString>
```



# GEOMETRIA GML:CURVE

```
gml:Curve>
  <gml:segments>

    <gml:LineStringSegment>
      <gml:pos>0 2</gml:pos>
      <gml:pos>1 0</gml:pos>
      <gml:pos>2 0</gml:pos>
    </gml:LineStringSegment>

    <gml:LineStringSegment>
      <gml:pos>3 1</gml:pos>
      <gml:pos>4 1</gml:pos>
      <gml:pos>5 -1</gml:pos>
    </gml:LineStringSegment>

  </gml:segments>
</gml:Curve>
```

← Agregat typu liniowego.

← segment

Segmentami mogą być m.in:

- gml:LineStringSegment
- gml:Arc
- gml:Circle



# GEOMETRIA GML:POLYGON

```
<gml:Polygon>
```

```
  <gml:exterior>
```

```
    <gml:LinearRing>
```

```
      <!-- 5 punktów -->
```

```
    </gml:LinearRing>
```

```
  </gml:exterior>
```

```
  <gml:interior>
```

```
    <gml:LinearRing>
```

```
      <!-- 4 punkty -->
```

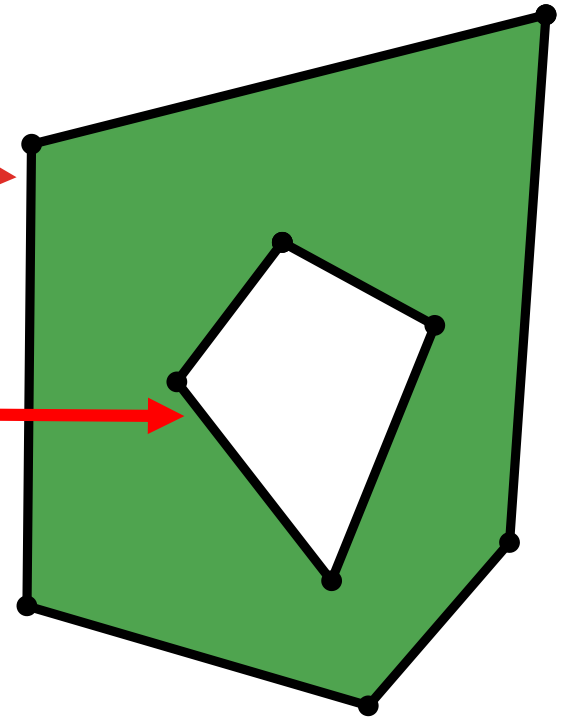
```
    </gml:LinearRing>
```

```
  </gml:interior>
```

```
</gml:Polygon>
```

**zewnątrzny**

**wewnętrzny**

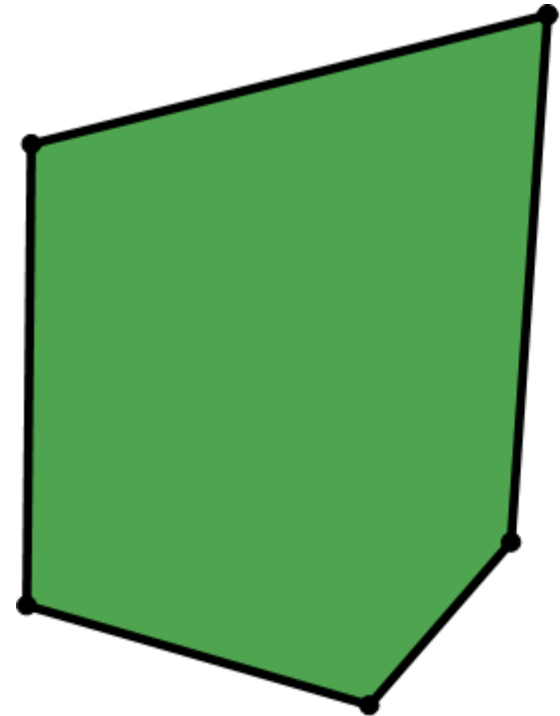




# GEOMETRIA GML:POLYGON

W przypadku gdy w wielokącie nie ma „dziur”, składnia może zostać uproszczona.

```
<gml:Polygon>  
  <gml:LinearRing>  
    <!-- 5 punktów -->  
  </gml:LinearRing>  
</gml:Polygon>
```





# GEOMETRIA GML:SURFACE

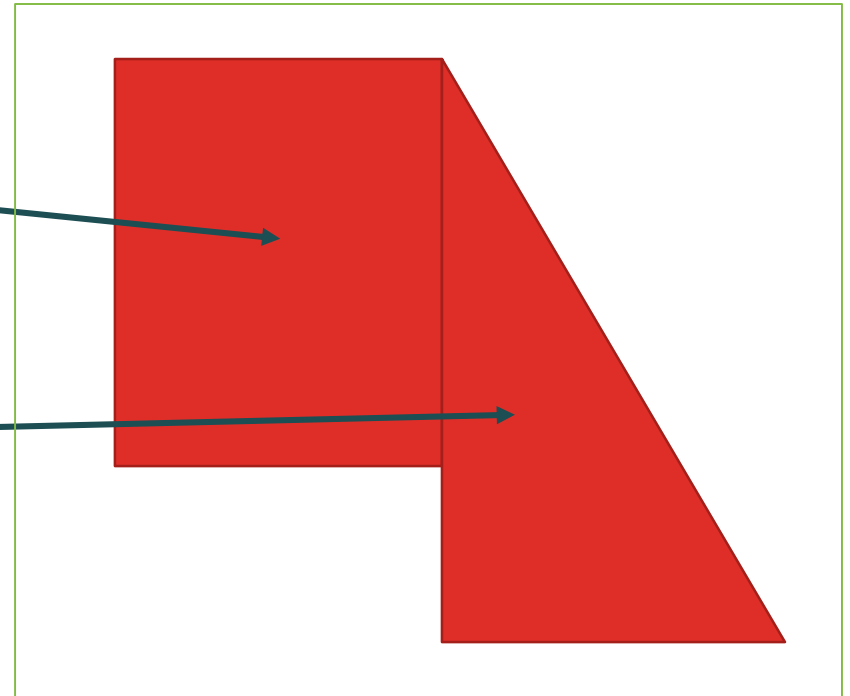
Surface

```
<gml:Surface>
  <gml:patches>

    <gml:PolygonPatch>
      <!-- wielokąt -->
    </gml:PolygonPatch>

    <gml:PolygonPatch>
      <!-- wielokąt -->
    </gml:PolygonPatch>

  </gml:patches>
</gml:Surface>
```







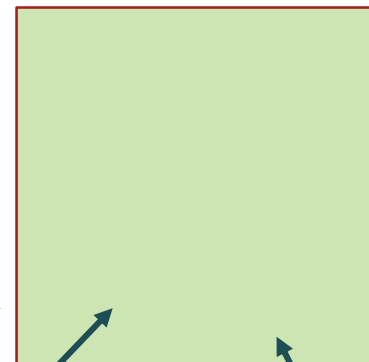
# GEOMETRIA – ZAPIS WSPÓŁRZĘDNYCH

```
<gml:Polygon >
  <gml:exterior>
    <gml:LinearRing>
      <gml:posList>0 0 0 1 1 1 1 0 0 0</gml:posList>
    </gml:LinearRing>
  </gml:exterior>
</gml:Polygon>
```

```
<gml:Surface>
  <gml:patches>
    <gml:PolygonPatch>
      <gml:exterior>
        <gml:LinearRing>
          <gml:posList>0 0 0 1 1 1 1 0 0 0</gml:posList>
        </gml:LinearRing>
      </gml:exterior>
    </gml:PolygonPatch>
  </gml:patches>
</gml:Surface>
```

```
<gml:Polygon>
  <gml:exterior>
    <gml:LinearRing>
      <!-- deprecated -->
      <gml:coordinates>0,0 0,1 1,1 1,0 0,0</gml:coordinates>
    </gml:LinearRing>
  </gml:exterior>
</gml:Polygon>
```

```
<gml:Polygon>
  <gml:exterior>
    <gml:LinearRing>
      <gml:pos>0 0</gml:pos>
      <gml:pos>0 1</gml:pos>
      <gml:pos>1 1</gml:pos>
      <gml:pos>1 0</gml:pos>
      <gml:pos>0 0</gml:pos>
    </gml:LinearRing>
  </gml:exterior>
</gml:Polygon>
```





# UKŁADY WSPÓŁRZĘDNYCH

GML umożliwia również definiowanie i opisywanie układów współrzędnych (CRS – Coordinate Reference System):

- geodezyjne, kartezjańskie, sferyczne, elipsoidalne, cylindryczne, użytkownika ...
- opisywanie osi układu,
- opisywanie operacji na układach.



# UKŁADY WSPÓŁRZĘDNYCH

Atrybutami służącymi do opisu układu odniesienia w pliku GML są:

- **srsName** – zapis kodowania układu współrzędnych pliku GML (type="anyURI")
- **srsDimension** – zawierający informację o wielowymiarowości współrzędnych (type="positiveInteger")

Przykład:

```
<prostokat srsName="urn:ogc:def:crs:EPSG::4326"  
srsDimension="2">
```



# UKŁADY WSPÓŁRZĘDNYCH

Podczas pracy z plikami GML można spotkać się z następującymi sposobami kodowania EPSG w atrybucie srsName (gdzie "xxxx" to kod układu):

- „urn:x-ogc:def:crs:EPSG:xxxx” (URN)
- „urn:ogc:def:crs:EPSG::xxxx” (URN)
- „http://www.opengis.net/def/crs/EPSG/0/xxxx” – URI rekomendowane dla Web Feature Service 2.0 \*
- „EPSG:xxxx” – nie zalecane dla GML (nie jest to prawidłowy URI), jednak używane przez liczne oprogramowanie GIS

\* - [http://docs.opengeospatial.org/is/09-025r2/09-025r2.html#\\_ftn40](http://docs.opengeospatial.org/is/09-025r2/09-025r2.html#_ftn40)



# UKŁADY WSPÓŁRZĘDNYCH

Kodowanie układu EPSG

```
<gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326">
  <gml:outerBoundaryIs>
    <gml:LinearRing srsName="urn:ogc:def:crs:EPSG::4326">
      <gml:coordinates>
        15.796944804216867,51.341473945783129
        15.780519578313251,51.19870082831325
        15.659225602409638,51.249239984939756
        15.700288667168675,51.331997853915659
        15.796944804216867,51.341473945783129
      </gml:coordinates>
    </gml:LinearRing>
  </gml:outerBoundaryIs>
</gml:Polygon>
```



# CZAS W GML

Oprócz geometrii GML pozwala na wyrażanie obiektów czasoprzestrzennych (spatiotemporal features), czyli dołączanie do istniejącej geometrii czy topo-geometrii znaczników czasu:

- gml:RelatedTimeType,
- gml:TimePosition,
- gml:TimeInstant, gml:TimePeriod,
- gml:timeLength, gml:duration, gml:timeInterval.



# CZAS W GML

Jednak w większości przypadków wystarczy wykorzystanie podstawowych typów XML Schema:

- xsd:Date,
- xsd:DateTime,
- xsd:timeStamp.



# JEDNOSTKI MIARY

GML pozwala zdefiniować jednostki miary (units of measurement), w których wyrażone są parametry:

- systemy miar – gml:BaseUnit,
- skalary – gml:AreaType, gml:VolumeType, gml:SpeedType, gml:TimeType, gml:AngleType,
- liczność elementów.







# TECHNOLOGIE XML: XLINK



**XLink** – specyfikacja W3C służąca do tworzenia łączy w dokumentach XML. **Xlink** można używać zarówno do tworzenia odnośników, jak również łączenia dokumentów z innymi.

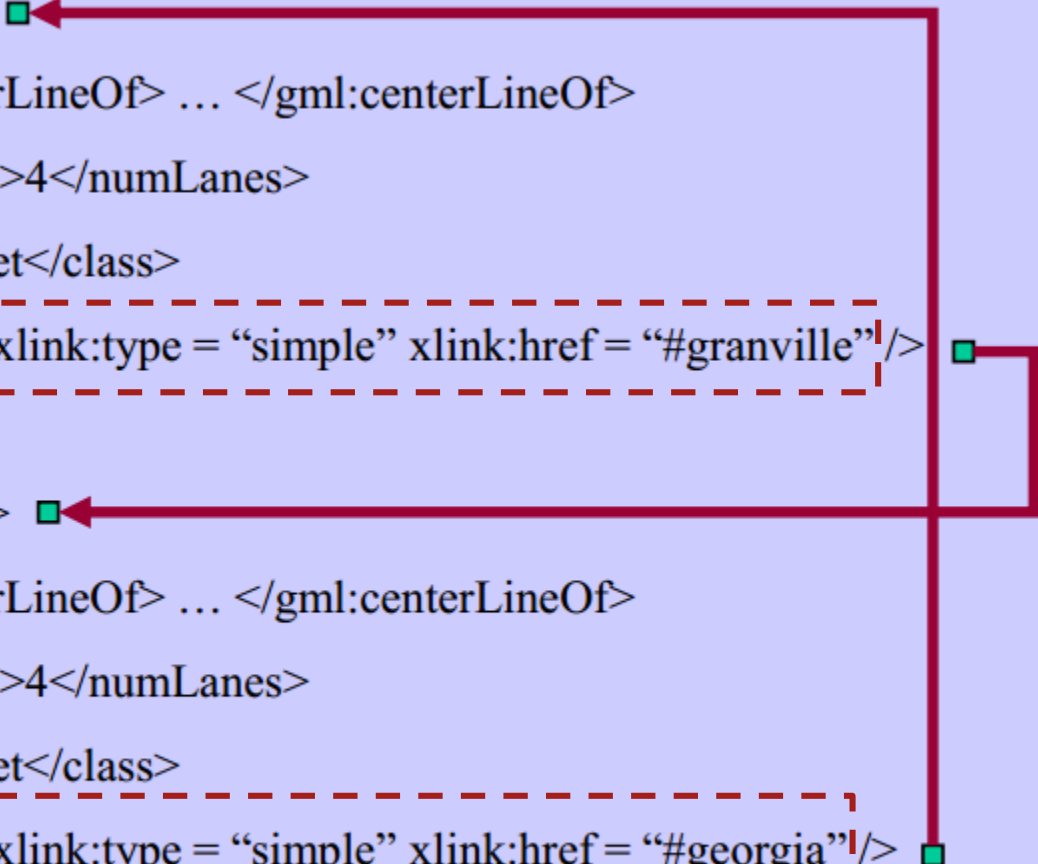
Standard ten jest elastyczniejszy niż dobrze znane linkowanie HTML, ponieważ wskazywać można na praktycznie dowolny zasób.



# TECHNOLOGIE XML: XLINK

```
<Road fid = "georgia"> 
  <gml:centerLineOf> ... </gml:centerLineOf>
  <numLanes>4</numLanes>
  <class>street</class>
  <intersects xlink:type = "simple" xlink:href = "#granville"/> 
</Road>

<Road fid = "granville"> 
  <gml:centerLineOf> ... </gml:centerLineOf>
  <numLanes>4</numLanes>
  <class>street</class>
  <intersects xlink:type = "simple" xlink:href = "#georgia"/> 
</Road>
```





# TECHNOLOGIE XML: XLINK

Obecność atrybutu "xlink:href" we właściwości GML oznacza, że wartość w tym miejscu należy utożsamiać z zasobem znajdującym się w referencji.

Xlink:href – zasobem dla referencji może być inny element w tym samym pliku GML, jak również dowolny zasób znajdujący się w Internecie.

Za pomocą xlink:href wewnątrz pliku GML wskazuje się również obiekty powiązane rolą asocjacji.

XLINK nie jest standardem wspieranym przez przeglądarki w związku z czym niemożliwa jest podróż od zasobu do zasobu, analogicznie jak to ma miejsce w przypadku hiperłączy HTML.



# CZEŚĆ PRAKTYCZNA (INSTRUKTAŻ ORAZ ĆWICZENIA)

## **Materiały szkoleniowe – Zeszyt ćwiczeń: Rozdział – Podstawy GML**

Ćwiczenia:

- Ćwiczenie – Tworzenie prostego pliku GML
- Ćwiczenie – Kontrola wczytywania plików GML do QGIS
- Ćwiczenie – Zapis geometrii w GML

# ROLA GML W INSPIRE





# ZASTOSOWANIE

GML w INSPIRE jest używany do prezentacji wszystkich danych zbiorów zharmonizowanych, które zazwyczaj mają odniesienie przestrzenne:

- korzystają z przestrzeni nazw charakterystycznych dla GML,
- przechowują informacje o współrzędnych obiektów.



# IDENTYFIKATORY OBIEKTÓW I ICH UŻYCIE W GML

Idea użycia GML jako formatu przechowywania zharmonizowanych danych INSPIRE implikuje tworzenie dwóch rodzajów identyfikatorów:

- wynikających ze specyfiki formatu danych: gml:id, gml:identifier,
- wynikających ze struktury danych zharmonizowanych: inspireId, thematicId.



# IDENTYFIKATORY OBIEKTÓW I ICH UŻYCIE W GML – GML:ID

Jest to standardowy w GML identyfikator każdego dającego się wyróżnić obiektu w zbiorze w formie atrybutu elementu głównego tego obiektu:

```
<gml:featureMember>
  <sd:SpeciesDistributionUnit gml:id="Article17SpeciesDistribution_1424">

  <gml:MultiSurface gml:id="_8651e8e9-a74f-4d23-bceb-ff67f68fb85c" srs!
    <gml:surfaceMember>
      <gml:Polygon gml:id="_2353e040-d49d-4e98-b011-b6a1f83f677d">
```

- musi jednoznacznie identyfikować każdy obiekt w dokumencie – służy do wewnętrznego adresowania,
- dziedzina gml:id to CharacterString ograniczony do liter, podkreślników, myślników, kropek i cyfr (NCName).





# IDENTYFIKATORY OBIEKTÓW I ICH UŻYCIE W GML – GML:IDENTIFIER

gml:identifier został wprowadzony od standardu GML w wersji 3.2. Jest jednym ze standardowych elementów, które tworzy się dla każdego obiektu GML:

```
<gml:identifier  
codeSpace="http://inspire.ec.europa.eu/ids">https://msdi.data.gov.  
mt/so/sd/SpeciesDistributionUnit/Article17SpeciesDistribution/1391  
</gml:identifier>
```

- podawany razem z odniesieniem do listy możliwych wartości (codeSpace)
- nie jest elementem obligatoryjnym – licznosc [0..1]



# IDENTYFIKATORY OBIEKTÓW I ICH UŻYCIE W GML: INSPIREID

Atrybut inspireId, będący identyfikatorem obiektu wynikającym ze struktury zharmonizowanego zbioru danych składa się z 3 podatrybutów:

- localId – unikalny identyfikator obiektu w obrębie zbioru,
- namespace – unikalny identyfikator zbioru,
- versionId – wersja identyfikatora.

```
<sd:inspireId>
  <base:Identifier>
    <base:localId>1391</base:localId>
    <base:namespace>Article17SpeciesDistribution</base:namespace>
  </base:Identifier>
</sd:inspireId>
```

Jego zastosowanie nie zawsze jest wymagane, niemniej jednak uważa się za wysoce zalecane jego utworzenie zawsze wtedy, gdy jest to możliwe.



# IDENTYFIKATORY OBIEKTÓW I ICH UŻYCIE W GML: THEMATICID

ThematicId jest dodatkowym, alternatywnym identyfikatorem obiektu. Może wystąpić wtedy, gdy zasadne jest opisanie elementów zbioru dodatkowym identyfikatorem, przepisany wprost ze zbioru źródłowego. Jego atrybuty składowe to:

- `identifier` – unikalny identyfikator obiektu w obrębie założonego schematu,
- `identifierScheme` – identyfikator schematu użytego do przedzielenia wartości *identifier*.

```
<base2:ThematicIdentifier>
```

```
  <base2:identifier>PL51</base2:identifier>
```

```
  <base2:identifierScheme>NUTS</base2:identifierScheme>
```

```
</base2:ThematicIdentifier>
```

Informacje przechowywane przez `thematicId` są analogiczne do tych, które przechowuje `gml:identifier`, stąd można uznać za zasadne pozyskiwanie tylko jednego z tych elementów w zbiorze zharmonizowanym.



# ENUMERACJE I LISTY KODOWE

Enumeracje i listy kodowe określają zakres dziedzinowy danego atrybutu poprzez podanie jego dopuszczalnych wartości.

- **Enumeracja** to zamknięta lista dopuszczalnych wartości, których definicje są aksjomatyczne. Ma zastosowanie wtedy, gdy zjawisko opisywane przez atrybut ma ściśle określone stany, np. typ zadrzewienia: iglasty, liściasty, mieszany.
- **Lista kodowa** to otwarta bądź zamknięta lista dopuszczalnych wartości, która jest wzbogacona o definicje pojęć. Ma zastosowanie wtedy, gdy zjawisko opisywane przez atrybut może przyjąć wartości dotychczas nigdy nie występujące w zbiorze, np. lista występujących w kraju gatunków zwierząt.



# ENUMERACJE I LISTY KODOWE W INSPIRE GML

Zastosowanie formatu GML jako właściwego dla danych INSPIRE pozwala na kontrolę poprawności wartości atrybutów, które są zebrane w postaci list kodowych bądź enumeracji. Kontrola jest możliwa poprzez walidację danych zharmonizowanych względem schematu XSD, pod warunkiem poprawnego zdefiniowania restrykcji w temacie.

# ENUMERACJE I LISTY KODOWE W INSPIRE GML – SPOSÓB KODOWANIA W PLIKU XSD

Ponieważ wartości z list kodowych są przechowywane w repozytorium poza plikiem XSD, podanie ich wartości następuje poprzez typ danych `gml:ReferenceType`.

```
<element name="countingMethod" type="gml:ReferenceType">
  <annotation>
    <documentation>-- Name --
counting method

-- Definition --
Method of providing a number for the indication of the abundance of a species within a specific species distribution unit.

-- Description --
To obtain a density or abundance estimate the data set provider can either count, estimate or calculate the population abundance.</documentation>
    <appinfo>
      <taggedValue xmlns="http://www.interactive-instruments.de/ShapeChange/AppInfo" tag="obligation">implementingRule</taggedValue>
    </appinfo>
  </annotation>
</element>
```

# ENUMERACJE I LISTY KODOWE W INSPIRE GML – SPOSÓB KODOWANIA W PLIKU INSPIRE GML

Typ danych `gml:ReferenceType` określa podanie wartości elementu za pomocą atrybutu `xlink:href`.

```
<sd:countingMethod xlink:href="http://inspire.ec.europa.eu/codelist/CountingMethodValue/calculated"/>
```



# REJESTR INSPIRE

Rejestr INSPIRE stanowi centralny punkt dostępu do wielu zarządzanych centralnie rejestrów INSPIRE. Zawartość tych rejestrów bazuje na dyrektywie INSPIRE, Zasadach Implementacji i Wytycznych Technicznych. Znajduje się na stronie [www](http://inspire.ec.europa.eu/registry/):

<http://inspire.ec.europa.eu/registry/>





# REJESTR INSPIRE – ZAWARTOŚĆ

Zawartość rejestru INSPIRE obejmuje:

- wykaz tematów
- schematy aplikacyjne
- listy kodowe
- enumeracje wartości danych
- referencje do dokumentów
- listy kodowe metadanych INSPIRE
- wykaz warstw INSPIRE
- słowniki terminów i definicji

# MAPOWANIE SCHEMATU APLIKACYJNEGO UML DO SCHEMATU APLIKACYJNEGO GML [ISO 19136]



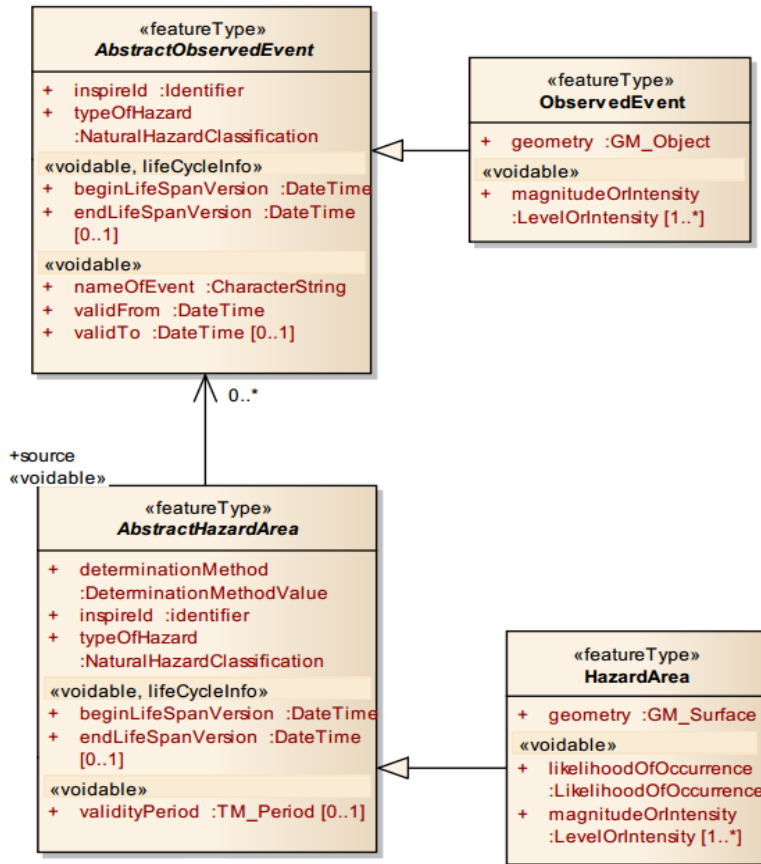
# SCHEMATY APLIKACYJNE GML

Schematy aplikacyjne GML mogą być tworzone:

- **ręcznie,**
- **z wykorzystaniem aplikacji wspomagających tworzenie schematów XML,**
- **z wykorzystaniem odpowiednio opisanych diagramów UML i aplikacji konwertujących.**



# UML – PRZYPOMNIENIE



Kluczowe pojęcia:

- Klasa,
- Atrybuty,
- Asocjacja,
- Generalizacja,
- Stereotyp.



# UML – PRZYPOMNIENIE

Diagramy klas można tworzyć w dedykowanych aplikacjach m.in.:

- Enterprise Architect,
- Papyrus UML,
- WhiteStarUML (open source),
- Microsoft Visio.

Na podstawie odpowiednio opisanego (za pomocą stereotypów) diagramu UML można wygenerować gotowy schemat aplikacyjny GML.



# GENEROWANIE SCHEMATÓW GML

W tym celu można wykorzystać narzędzie ShapeChange pozwalające wczytać:

- projekty Enterprise Architect (.eap),
- pliki XMI 1.0,
- GSIP Microsoft Access Database.



# GENEROWANIE SCHEMATÓW GML

Schemat aplikacyjny UML zgodny z ISO 19109 może być przekształcony do schematu aplikacyjnego GML zgodnie z regułami określonymi w standardzie ISO 19136, załącznik E.

- **ISO 19109** – definiuje zasady opracowywania schematów aplikacyjnych, w tym reguły definiowania klas obiektów.
- **ISO 19136** – norma określająca standard kodowania dla plików GML.





# KODOWANIE DIAGRAMÓW UML

## Stereotypy klas

STEREOTYP	ALIASY	DEFINICJA
Brak stereotypu	<<abstract>>	Standardowa klasa obiektów, w niektórych modelach pojawia się również stereotyp „Abstract” lecz powinno się takiego zapisu unikać.
<<FeatureType>>	<<feature>>	Oznaczenie klasy jako "FeatureType" w rozumieniu ISO 19136. Transformowany do <b>gml:AbstractFeatureType</b> .
<<DataType>>	<<request>> <<response>>	Element globalny definiujący typ danych. Transformowany do <b>complexType</b> z ograniczeniem sequence.
<<Enumeration>>		Typ wyliczeniowy. Transformowany do <b>enumeracji XSD</b> .
<<CodeList>>	<<conceptScheme>> <<vocabulary>>	Typ słownikowy, który może być rozszerzany o dodatkowe wartości.
<<Union>>		Element transformowany do <b>complexType</b> z ograniczeniem choice.





# KODOWANIE DIAGRAMÓW UML

## Stereotypy atrybutów

STEREOTYP	ALIASY	DEFINICJA
<<voidable>>		Atrybut może, ale nie musi wystąpić. W przypadku braku, przyczyna braku zamiast wartości.
<<version>>		Dotyczy sytuacji, gdy obiekt jest powiązany rolą asocjacji. Wskazuje, że wartość dotyczy konkretnej wersji obiektu, nie natomiast obiektu w sposób ogólny.
<<property>>		Standardowy atrybut.



# KODOWANIE TYPÓW DANYCH

Oznaczenie UML	Typ XML Schema
Character	xsd:string
CharacterString	xsd:string
Integer	xsd:integer
Boolean	xsd:boolean
Number	xsd:double
Real	xsd:double
Date	xsd:date
DateTime	xsd:dateTime
Decimal	xsd:double
Length	gml:LengthType
Distance	gml:LengthType
Angle	gml:AngleType
Volume	gml:VolumeType



# KODOWANIE GEOMETRII

Oznaczenie w UML	Typ XML Schema
GM_Point	gml:Point
GM_Curve	gml:Curve
GM_Surface	gml:Surface
GM_MultiPoint	gml:MultiPoint
GM_LineString	gml:LineString
GM_Polygon	gml:PolygonPatch
GM_Circle	gml:Circle
GM_ArcString	gml:ArcString



# KLASY OBIEKTÓW

```
<element name="Budynek" substitutionGroup="gml:AbstractFeature"
type="test:BudynekType"/>
<complexType name="BudynekType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element maxOccurs="8" minOccurs="8" name="identyfikator" type="string"/>
        <element name="liczba_kondygnacji" type="integer"/>
        <element name="nazwa" type="string"/>
        <element name="obszar" type="gml:SurfacePropertyType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Należy pamiętać o publicznym  
dostępie dla atrybutów. →

«featureType» Budynek	
+	identyfikator :CharacterString [8]
+	liczba_kondygnacji :Integer
+	nazwa :CharacterString
+	obszar :GM_Surface



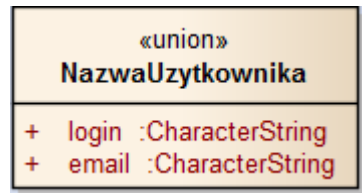
# TYP DANYCH

```
<element name="Adres" substitutionGroup="gml:AbstractObject" type="test:AdresType"/>
  <complexType name="AdresType">
    <sequence>
      <element name="ulica" type="string"/>
      <element name="miejscowość" type="string"/>
      <element name="kod-pocztowy" type="string"/>
    </sequence>
  </complexType>
```

«dataType» Adres	
+	ulica :CharacterString
+	miejscowość :CharacterString
+	kod-pocztowy :CharacterString

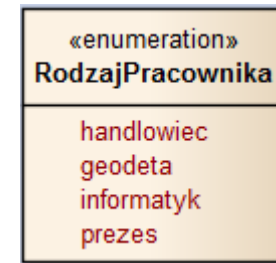


# UNIA ORAZ TYP ENUMERACYJNY



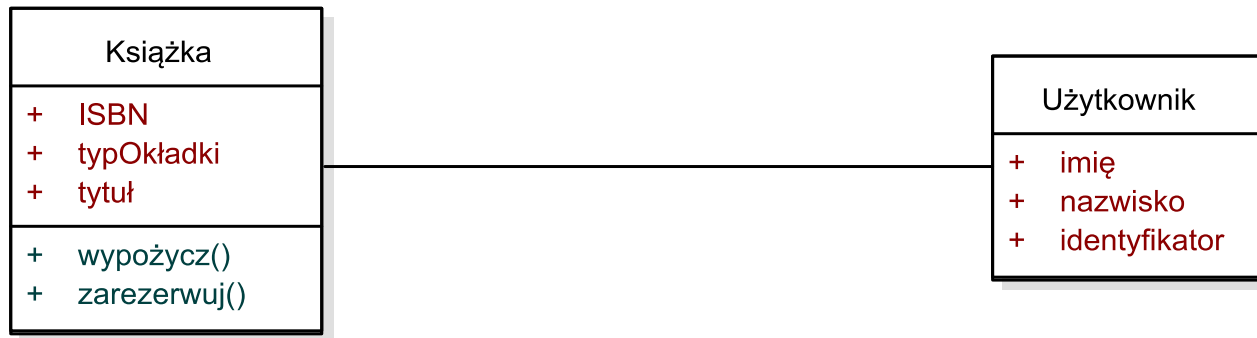
```
<complexType name="NazwaUzytkownikaType">  
  <choice>  
    <element name="login" type="string"/>  
    <element name="email" type="string"/>  
  </choice>  
</complexType>
```

```
<simpleType name="RodzajPracownikaType">  
  <restriction base="string">  
    <enumeration value="handlowiec"/>  
    <enumeration value="geodeta"/>  
    <enumeration value="informatyk"/>  
    <enumeration value="prezes"/>  
  </restriction>  
</simpleType>
```





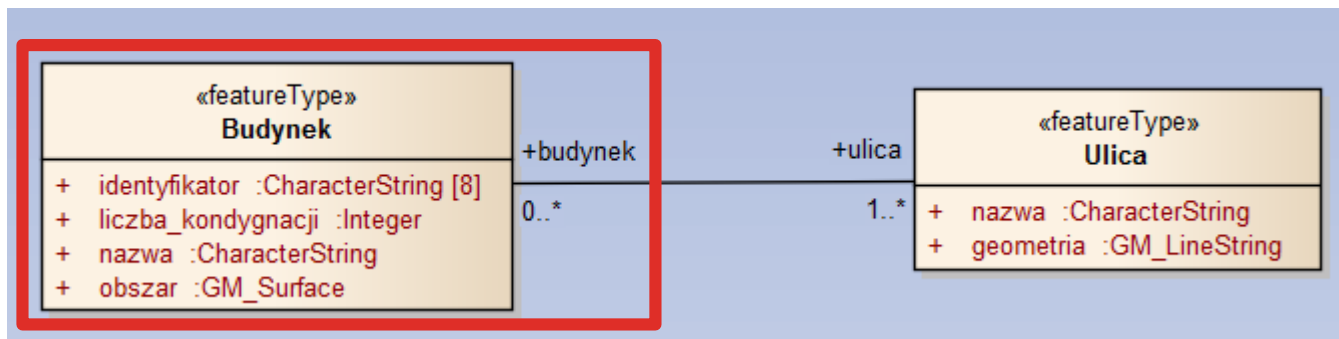
# ASOCJACJE PRZYPOMNIENIE DEFINICJI



- Związek dwukierunkowy.
- Obie klasy są świadome związku.



# ASOCJACJE

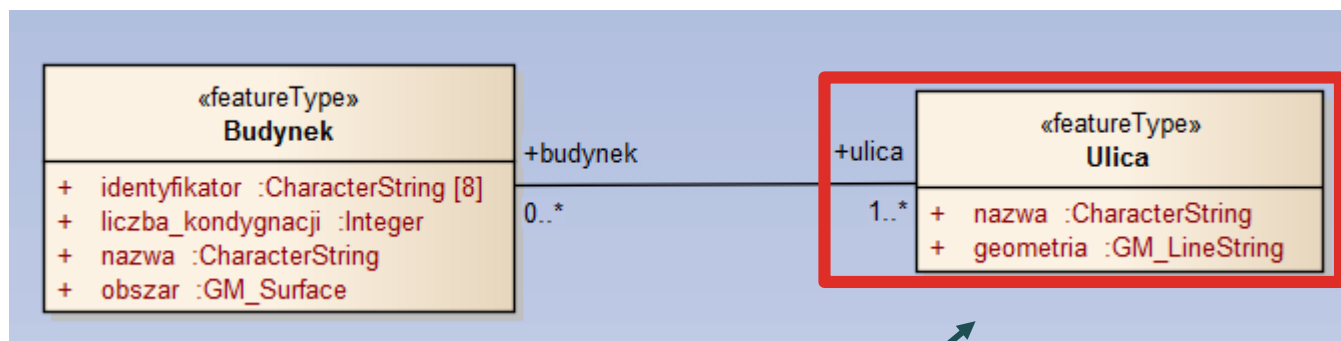


```
<complexType name="BudynekType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <!-- pozostałe atrybuty -->
        <element maxOccurs="unbounded"
          minOccurs="0" name="ulica"
          type="test:UlicaPropertyType">
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```





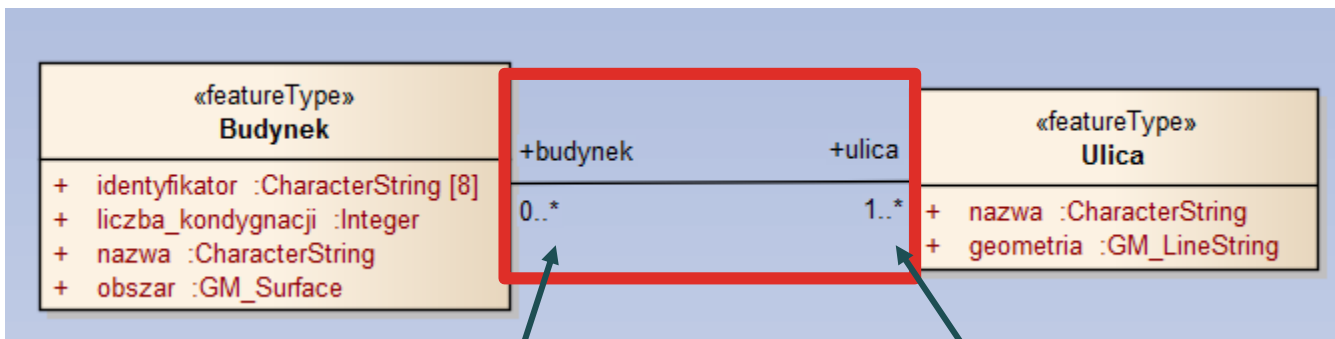
# ASOCJACJE



```
<complexType name="UlicaType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="nazwa" type="string"/>
        <element maxOccurs="unbounded" name="budynek,,
          type="test:BudynekPropertyType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```



# ASOCJACJE



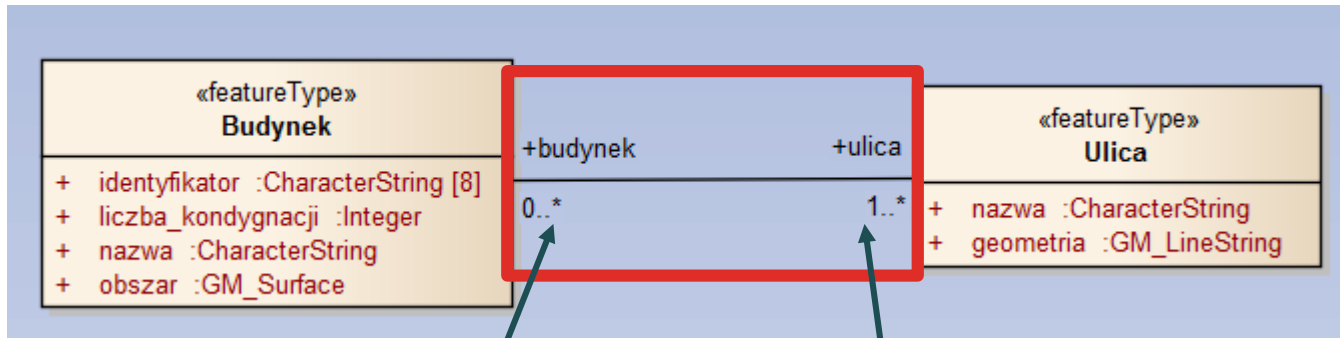
```
<element name="budynek" type="gml:ReferenceType" minOccurs="0"
maxOccurs="unbounded">
  <annotation>
    <appinfo source="urn:gml:targetElement">Budynek</appinfo>
    <appinfo source="urn:gml:reverseProperty">ulica</appinfo>
  </annotation>
</element>
```

Zapis asocjacji  
poprzez referencje.

```
<element name="ulica" type="gml:ReferenceType" minOccurs="1"
maxOccurs="unbounded">
  <annotation>
    <appinfo source="urn:gml:targetElement">Ulica</appinfo>
    <appinfo source="urn:gml:reverseProperty">budynek</appinfo>
  </annotation>
</element>
```



# ASOCJACJE



```
<element name="budynek" minOccurs="0"
maxOccurs="unbounded">
  <complexType>
    <sequence>
      <element ref="Budynek"/>
    </sequence>
  </complexType>
</element>
```

```
<element name="ulica" minOccurs="1"
maxOccurs="unbounded">
  <complexType>
    <sequence>
      <element ref="Ulica"/>
    </sequence>
  </complexType>
</element>
```

Zapis asocjacji  
poprzez wartość.

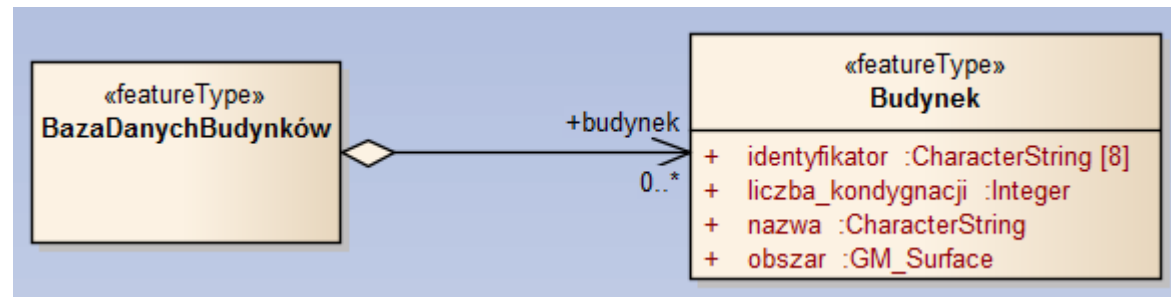


# AGREGACJA

```
<element name="BazaDanychBudynek" substitutionGroup="gml:AbstractFeature"
type="test:BazaDanychBudynekType"/>
  <complexType name="BazaDanychBudynekType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>

          <element maxOccurs="unbounded" minOccurs="0" name="budynek">
            <complexType>
              <complexContent>
                <extension base="gml:AbstractMemberType">
                  <sequence minOccurs="0">
                    <element ref="test:Budynek"/>
                  </sequence>
                  <attributeGroup ref="gml:AssociationAttributeGroup"/>
                </extension>
              </complexContent>
            </complexType>
          </element>
        </sequence>

      </extension>
    </complexContent>
  </complexType>
```





# CZEŚĆ PRAKTYCZNA (INSTRUKTAŻ ORAZ ĆWICZENIA)

## **Materiały szkoleniowe – Zeszyt ćwiczeń: Rozdział – Konwersja UML do XSD**

Instruktaże:

- Instruktaż – Opracowanie modelu danych w UML

Ćwiczenia:

- Ćwiczenie – Konwersja XML do formatu XSD
- Ćwiczenie – Ręczna Konwersja z notacji UML do schematu XSD

# KONWERSJA DANYCH PRZESTRZENNYCH Z POWSZECHNYCH FORMATÓW DO STRUKTUR GML





# GENEROWANIE DANYCH GML

Dane przestrzenne zazwyczaj przechowywane są w relacyjnych bazach danych. Posiadając schemat aplikacyjny GML i dane w przestrzennej bazie danych można otrzymać dane GML:

- ręcznie,
- transformacje XSLT,
- dedykowane aplikacje ETL (FME, HALE),
- aplikacje GIS (eksport do GML).



# GENEROWANIE DANYCH GML

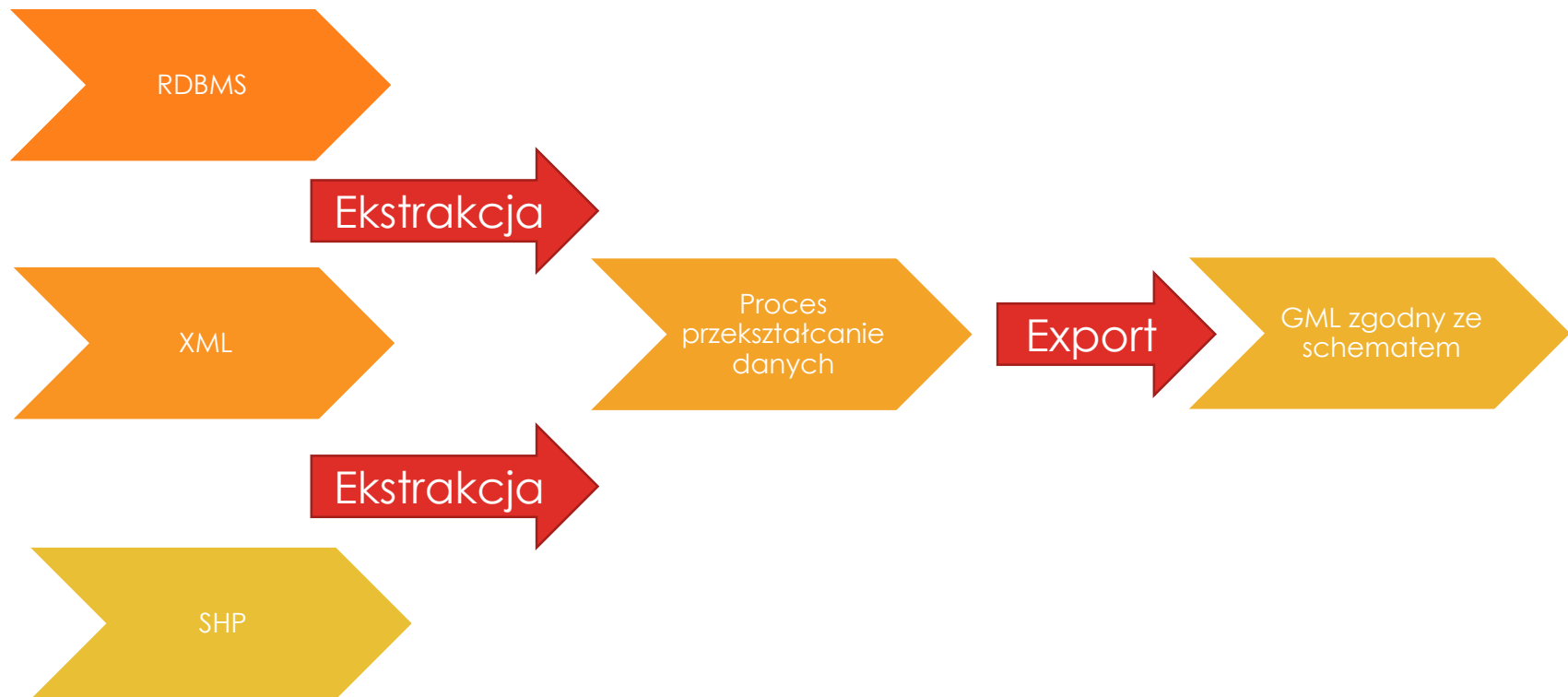


Podejście wykorzystujące transformacje XSL





# GENEROWANIE DANYCH GML



Podejście wykorzystujące narzędzia ETL



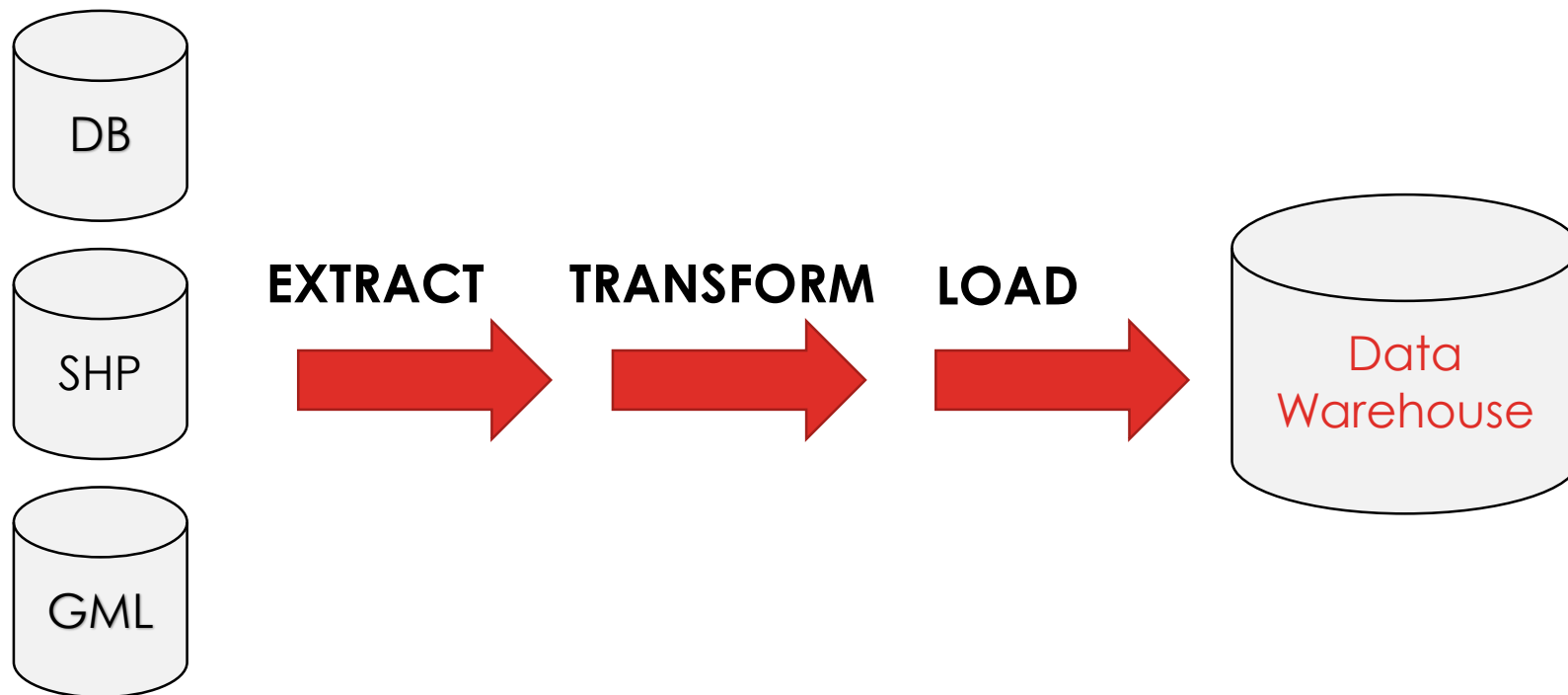
# DEFINICJA PROCESU ETL

Skrót **ETL** (z języka angielskiego **extraction, transformation, loading**) oznacza ekstrakcję, transformację i ładowanie danych. ETL jest procesem, w skład którego wchodzi:

- **Ekstrakcja** danych ze zbiorów źródłowych. Proces ten determinuje bazowe źródła dla hurtowni danych.
- **Transformacja** danych – etap ten może zawierać czyszczenie danych, filtrowanie oraz implementację reguł biznesowych.
- **Ładowanie** danych do hurtowni danych bądź bazy danych będącej repozytorium danych dla aplikacji raportujących.



# DEFINICJA PROCESU ETL





# ETL: TRANSFORMACJA DANYCH

Transformacja danych to zbiór funkcji oraz reguł, które pozwalają dane źródłowe przekształcić do postaci zgodnej ze strukturami oczekiwanymi docelowymi.

*Przykłady transformacji danych:*

- tłumaczenie kodowanych wartości,
- transformacja współrzędnych,
- filtrowanie, sortowanie, agregacja,
- selekcja, łączenie,
- mapowanie list kodowych.



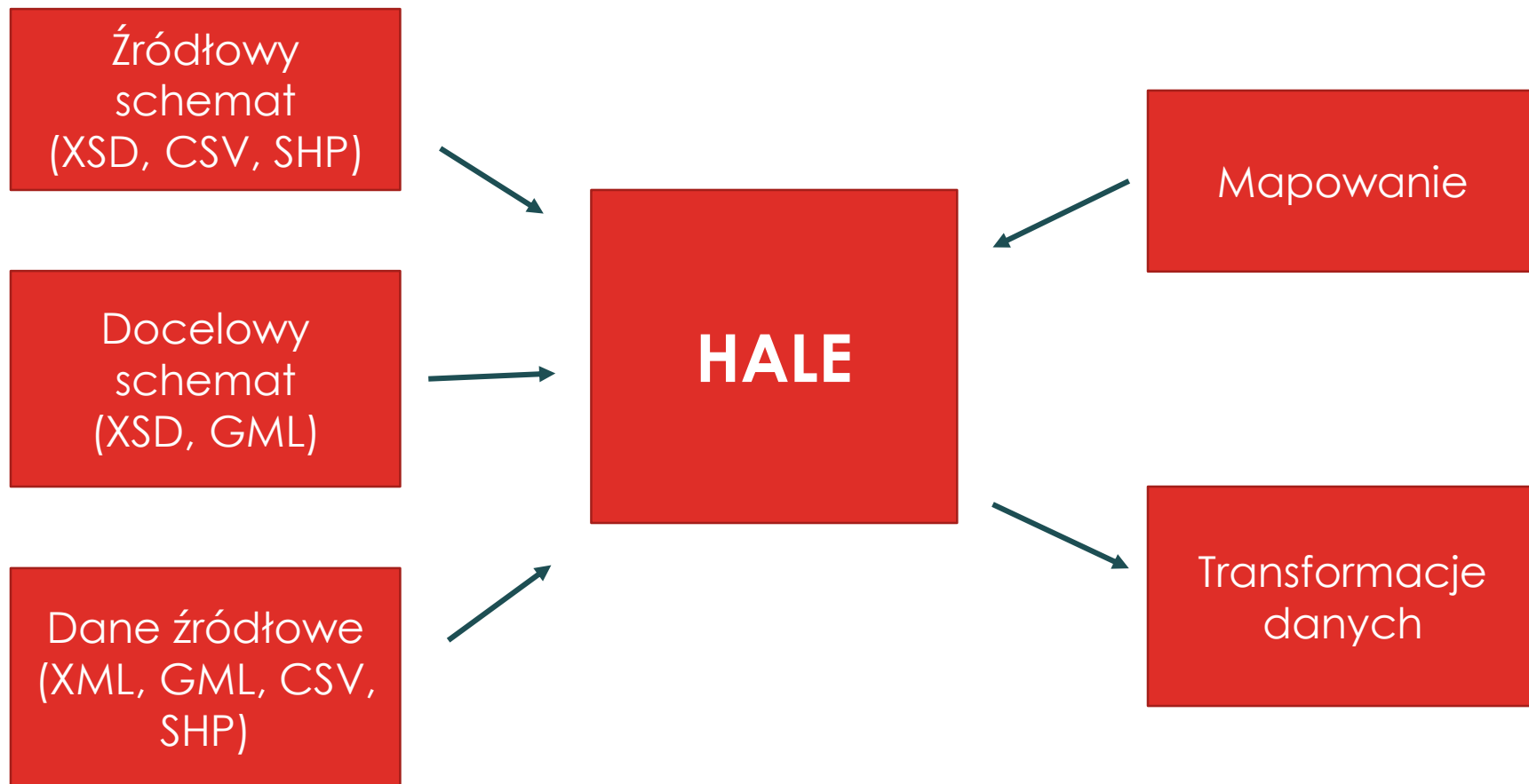
# OPEN SOURCE: HALE

- **HUMBOLDT Alignment Editor (HALE)** jest narzędziem ETL do mapowania pomiędzy różnymi schematami danych. Posiada przystępny graficzny interfejs.
- Dostępne jest na zasadach licencji **Open Source GNU LGPL** licence.
- Obecnie projekt jest rozbudowywany pod nazwą **data harmonisation panel**.





# WPROWADZENIE DO HALE





# WPROWADZENIE DO HALE

- HALE zapewnia rozbudowany graficzny interfejs przystosowany do potrzeb specjalistów GIS, dzięki czemu proces mapowania oraz samej transformacji jest czytelniejszy, a co za tym idzie łatwiejszy do zrozumienia, niż w przypadku transformacji XSL.
- Dodatkowo zapewniany jest bezpośredni podgląd na rezultaty transformacji zapewniający porównanie pomiędzy danymi wejściowymi a wyjściowymi.
- Oprogramowanie rozwijane z myślą o przeprowadzaniu harmonizacji danych przestrzennych INSPIRE.



# HALE – WORKFLOW

1. Import danych źródłowych wraz z ich schematem.
2. Import docelowego schematu danych.
3. Identyfikacja odpowiadających sobie klas obiektów oraz typów danych.





## HALE – WORKFLOW

4. Identyfikacja oraz dodanie w pliku projektowym powiązań oraz relacji pomiędzy źródłowymi a docelowymi typami danych.

- Jakie informacje są dostępne w źródłowym zbiorze danych?
- Jakie informacje są niezbędne dla zasilenia zbioru docelowego?
- W jaki sposób zasilić zbiór docelowy ze źródła?

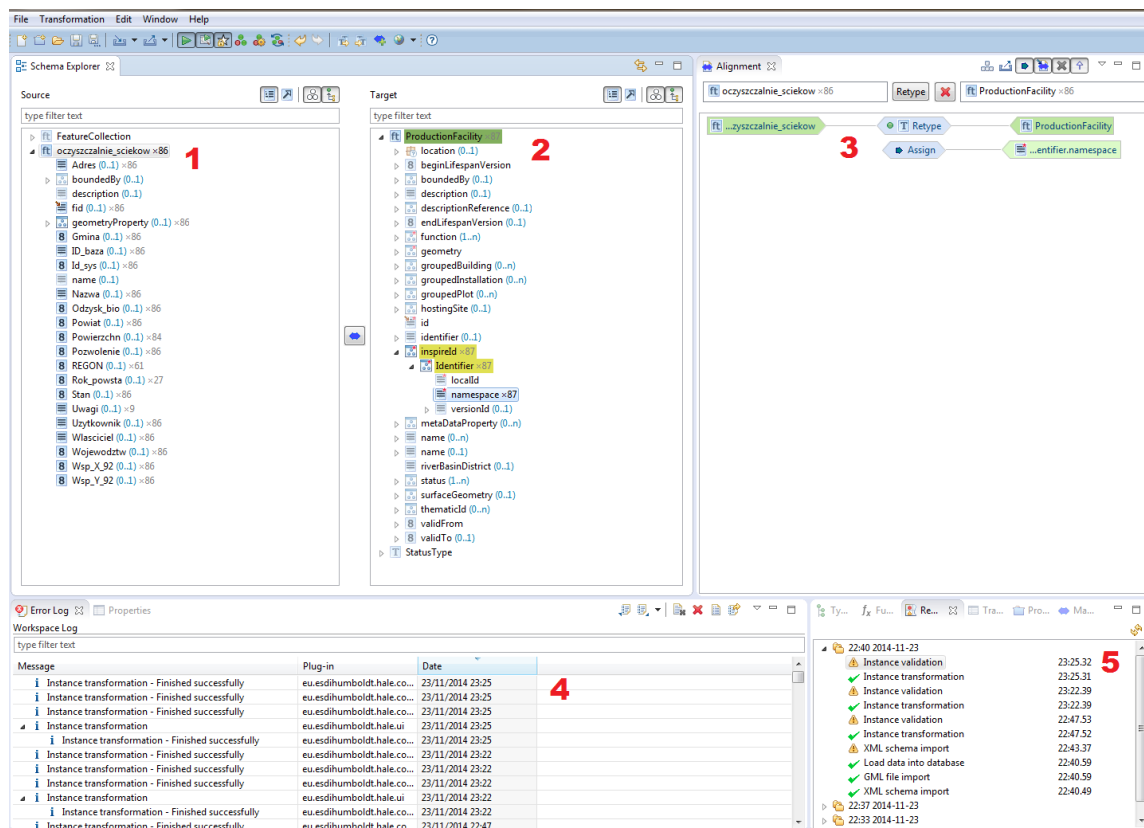


## HALE – WORKFLOW

5. Po każdej zmianie wprowadzonej w mapowaniu powinna nastąpić weryfikacja czy wyniki transformacji są zadowalające. Należy proces powtarzać iteracyjnie, aż rezultat będzie zadowalający.
6. Zapis przetransformowanych danych do pliku. Istnieje możliwość zdefiniowania samego mapowania, w celu późniejszego przetworzenia zbioru danych.



# OPEN SOURCE: HALE OKNO PROGRAMU



- 1) okno schematu źródłowego
- 2) okno schematu docelowego
- 3) okno transformacji
- 4) rejestr błędów/ właściwości
- 5) wydarzenia



# OPEN SOURCE: HALE OKNO MAPY

The screenshot displays the HALE OKNO MAPY application interface. It features a main map area on the left, a 'Transformed data' panel in the center, and a 'Schema Explorer' on the right. The 'Source data' panel at the bottom left shows a table of input data, and the 'Production facility' panel at the bottom center shows a table of output data. The 'Schema Explorer' on the right provides a hierarchical view of the schema elements, including classes and properties.

**Source data**

oczyszczalnie_sciekow	1	2
id	Michałów Grabska, ul Bukowa 8	Janówek, ul Nowodworska 2A
oczyszczalnie_sciekow	oczyszczalnie_sciekow.0	oczyszczalnie_sciekow.1
geometryProperty		
id	3	5
ID_base	1408_001	1408_002
id_syn	8	9
Name	Przedsiębiorstwo Instalacyjno-Usługowe i Oczyszczalnia Tęczyw w Janówku	
Odrys_bie	2	2
Powiat	8	8

**Production facility**

1	2	3
PLPZGK.325.PF.3.8	PLPZGK.325.PF.3.8	PLPZGK.325.PF.3.8
no value	no value	no value
no value	no value	no value
no value	no value	no value
no value	no value	no value
no value	no value	no value
no value	no value	no value
no value	no value	no value
no value	no value	no value

- 1) Mapa z danymi źródłowymi
- 2) Mapa z danymi po transformacji
- 3) Okno pomocy

- 4) Przykład danych wejściowych
- 5) Przykład danych wyjściowych



# CZEŚĆ PRAKTYCZNA (INSTRUKTAŻ ORAZ ĆWICZENIA)

## **Materiały szkoleniowe – Zeszyt ćwiczeń: Rozdział – Konwersja danych przestrzennych z powszechnych formatów do GML**

Instruktaże:

- Instruktaż – Proces transformacji w HALE

Ćwiczenia:

- Ćwiczenie – Samodzielne przejście procesu harmonizacji w HALE





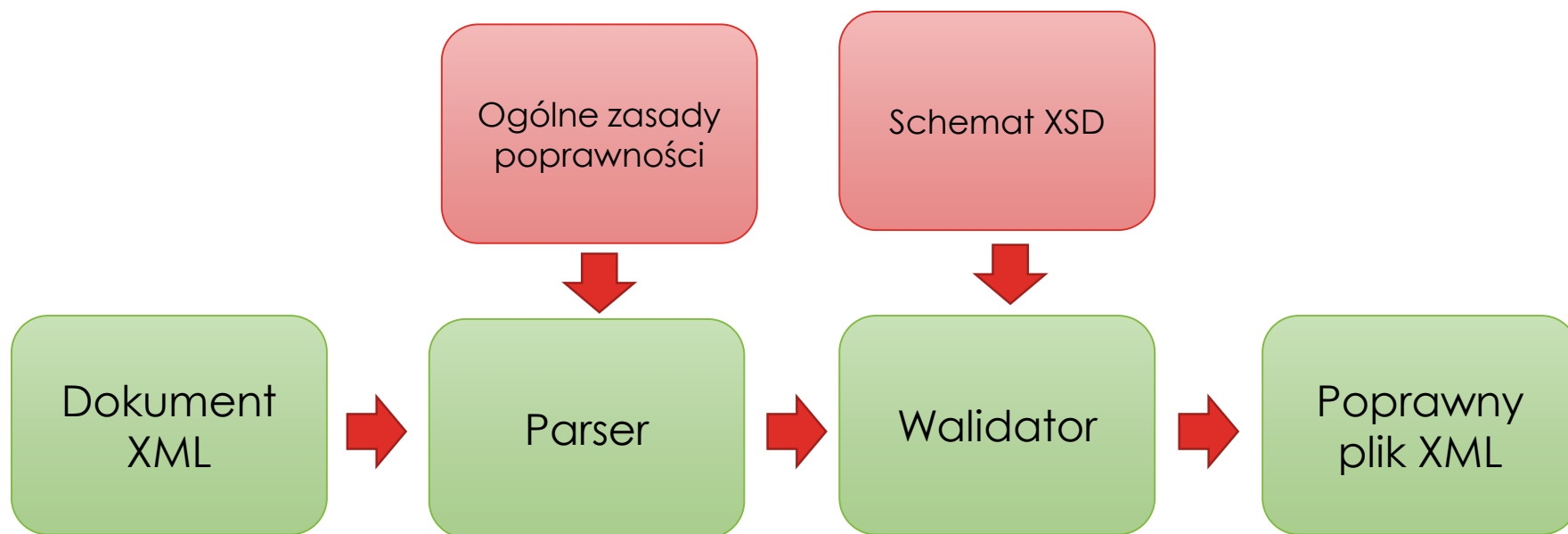
WROCLAW GLOWNY

budimex  
5

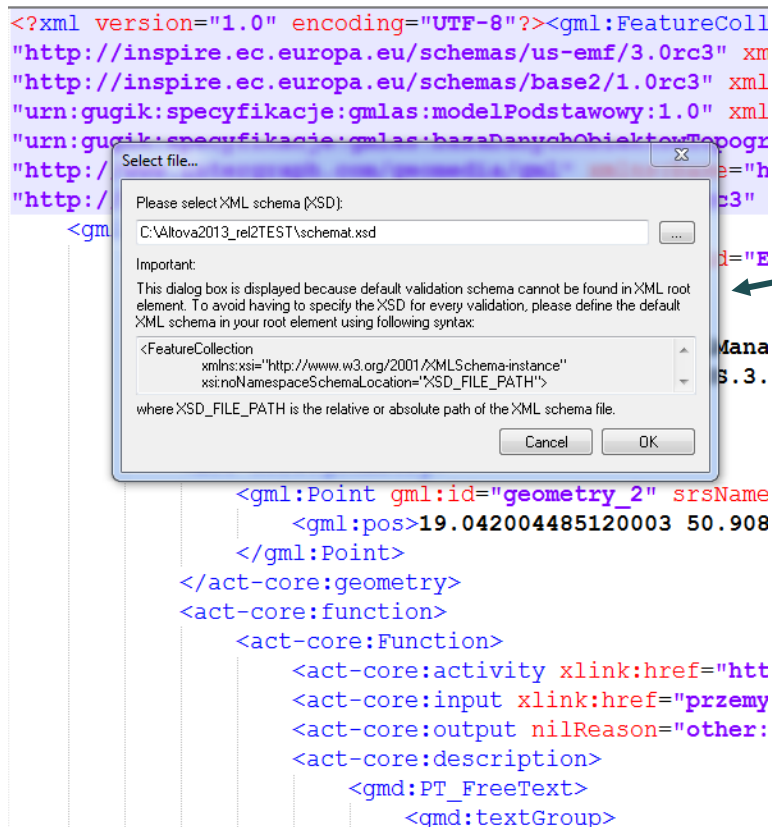
# WALIDACJA GML WZGLĘDEM SCHEMATU APLIKACYJNEGO



# PROCES WALIDACJI



# WALIDACJA EDYTOR TEKSTOWY



Każdorazowo należy wskazać plik XSD, względem którego ma zostać zwalidowany GML

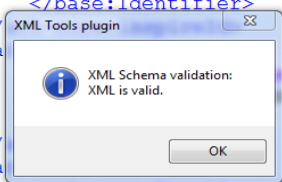
Rozwiązanie dobre, gdy pracujemy z co najwyżej kilkoma plikami. Brak automatyzacji, wszystko trzeba "wyklikać".





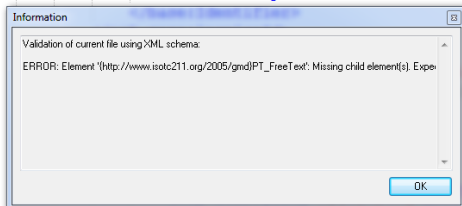
# WALIDACJA EDYTOR TEKSTOWY

```
<base:Identifier>
  <base:localId>US.EnvironmentalManageme
  <base:namespace>PL.ZIPGL.000.US.3.6</base:namespace>
</base:Identifier>
</a>
<a href="#" srsName="EPSG:4485120003 50.9085768"
  <act-core:Function>
    <act-core:activity xlink:href="http://
    <act-core:input xlink:href="przemyslow
    <act-core:output nilReason="other:unpc
    <act-core:description>
      <gmd:PT_FreeText>
        <gmd:textGroup>
          <gmd:LocalisedCharacterStr
        </gmd:textGroup>
      </gmd:PT_FreeText>
    </act-core:description>
  </act-core:Function>
```



Plik poprawny

```
5 <base:Identifier>
6   <base:localId>US.EnvironmentalManagementFacil
7   <base:namespace>PL.ZIPGL.000.US.3.6</base:namespace>
8 </base:Identifier>
9 </a>
10 <a href="#" srsName="EPSG:4256 50.90857681488594"
11   <act-core:Function>
12     <act-core:activity xlink:href="http://inspire
13     <act-core:input xlink:href="przemyslowe"/>
14     <act-core:output nilReason="other:unpopulated"
15     <act-core:description>
16       <gmd:PT_FreeText>
17         <gmd:textGroup>
18           <gmd:LocalisedCharacterString>sta
19       </gmd:PT_FreeText>
20     </act-core:description>
21   </act-core:Function>
22 </a>
23
```



Plik niepoprawny



# WALIDACJA PODEJŚCIE WSADOWE

- Automatyzacja procesu walidacji,
- Możliwość walidacji wielu plików,
- Zapis wyników do pliku tekstowego,
- Minimalizacja czasu bezczynności systemu dzięki pracy bez ingerencji człowieka,
- Uprzednio należy przygotować BAT, ze skryptyem definiującym walidację.



# WALIDACJA PODEJŚCIE WSADOWE

Lokalizacja do folderu z plikami

```
@echo on
```

```
for %%a in (C:\Altova2013_rel2TEST\*.xml) do (
```

```
echo Walidacja: "%%a": >> Plik_zapisu_rezultatow.txt
```

```
"C:\sciezka_do_folderu_z_altova_ce\AltovaXML_COM.exe" /validate "%%a" /schema
```

```
"C:\sciezka_do_pliku\schemat.xsd" >> Plik_zapisu_rezultatow.txt)
```

```
pause
```

Ścieżka do schematu XSD

Nazwa pliku do zapisu logów



# WALIDACJA PODEJŚCIE WSADOWE

Name	Date modified	Type	Size
Altova_log_EMF.txt	2016-10-13 09:32	Text Document	1 KB
plik1.xml	2016-10-12 14:47	XML Document	8 521 KB
plik2.xml	2016-10-12 14:47	XML Document	8 521 KB
plik3.xml	2016-10-13 09:31	XML Document	8 521 KB
plik4.xml	2016-10-12 14:47	XML Document	8 521 KB
schemat.xsd	2013-04-11 08:09	W3C XML Schema	8 KB
walidacja.bat	2016-10-13 09:30	Windows Batch File	1 KB

C:\Altova2013\_rel2TEST\Altova\_log\_EMF.txt - Notepad++

Plik Edycja Szukaj Widok Format Składnia Ustawienia Makra Uruchom Wtyczki Okno

walidacja.bat Altova\_log\_EMF.txt

```
1 Walidacja: "C:\Altova2013_rel2TEST\plik1.xml":
2 The XML data is valid.
3 Walidacja: "C:\Altova2013_rel2TEST\plik2.xml":
4 The XML data is valid.
5 Walidacja: "C:\Altova2013_rel2TEST\plik3.xml":
6 The XML data is invalid.
7 Further elements required under element <base:Identifier>.
8 Walidacja: "C:\Altova2013_rel2TEST\plik4.xml":
9 The XML data is valid.
```

C:\Windows\system32\cmd.exe

```
C:\Altova2013_rel2TEST>for %a in (C:\Altova2013_rel2TEST\*.xml) do (
echo Walidacja: "%a": 1>>Altova_log_EMF.txt
"C:\Program Files (x86)\Altova\AltovaXML2013\AltovaXML_COM.exe" /validate "%a"
/schema "C:\Altova2013_rel2TEST\schemat.xsd" 1>>Altova_log_EMF.txt
)

C:\Altova2013_rel2TEST><
echo Walidacja: "C:\Altova2013_rel2TEST\plik1.xml": 1>>Altova_log_EMF.txt
"C:\Program Files (x86)\Altova\AltovaXML2013\AltovaXML_COM.exe" /validate "C:\A
ltova2013_rel2TEST\plik1.xml" /schema "C:\Altova2013_rel2TEST\schemat.xsd" 1>>A
ltova_log_EMF.txt
)

C:\Altova2013_rel2TEST><
echo Walidacja: "C:\Altova2013_rel2TEST\plik2.xml": 1>>Altova_log_EMF.txt
"C:\Program Files (x86)\Altova\AltovaXML2013\AltovaXML_COM.exe" /validate "C:\A
ltova2013_rel2TEST\plik2.xml" /schema "C:\Altova2013_rel2TEST\schemat.xsd" 1>>A
ltova_log_EMF.txt
)

C:\Altova2013_rel2TEST><
echo Walidacja: "C:\Altova2013_rel2TEST\plik3.xml": 1>>Altova_log_EMF.txt
"C:\Program Files (x86)\Altova\AltovaXML2013\AltovaXML_COM.exe" /validate "C:\A
ltova2013_rel2TEST\plik3.xml" /schema "C:\Altova2013_rel2TEST\schemat.xsd" 1>>A
ltova_log_EMF.txt
)

C:\Altova2013_rel2TEST><
echo Walidacja: "C:\Altova2013_rel2TEST\plik4.xml": 1>>Altova_log_EMF.txt
"C:\Program Files (x86)\Altova\AltovaXML2013\AltovaXML_COM.exe" /validate "C:\A
ltova2013_rel2TEST\plik4.xml" /schema "C:\Altova2013_rel2TEST\schemat.xsd" 1>>A
ltova_log_EMF.txt
)

C:\Altova2013_rel2TEST>pause
Press any key to continue . . .
```



# WALIDACJA CZĘSTE BŁĘDY

- Braki wartości w elementach obligatoryjnych,
- Zła kolejność elementów w obiektach,
- Liczność elementów niezgodna z zakładanym schematem,
- Wartość enumeracyjne danych wejściowych nie pasują do schematu,
- Niezgodności w typach danych.



# KONTROLA SEMANTYCZNA DANYCH

Należy pamiętać, że walidacja nie obejmuje kontroli semantycznej danych, czy zbiór tworzy merytorycznie jednolitą spójną całość.

W celu zapewnienia dodatkowej kontroli należy:

- Dokonać kontroli reprezentatywnych próbek danych,
- Zweryfikować jakość zbioru już na poziomie danych wejściowych,
- Stworzyć dedykowane skrypty w dowolnej technologii, kontrolujące wymagane przez nas warunki niestandardowe.



# CZEŚĆ PRAKTYCZNA (INSTRUKTAŻ ORAZ ĆWICZENIA)

## **Materiały szkoleniowe – Zeszyt ćwiczeń: Rozdział – Walidacja plików GML względem schematu XSD**

Instruktaże:

- Instruktaż – Walidacja pliku GML w edytorze tekstowym Notepad++

Ćwiczenia:

- Ćwiczenie – Walidacja pliku GML – tryb wsadowy