Wrocławski Instytut Zastosowań Informacji Przestrzennej i Sztucznej Inteligencji

# Interoperacyjność i proces harmonizacji w ramach INSPIRE (poziom zaawansowany)

Szkolenia dla pracowników resortu środowiska w zakresie tematyki dotyczącej wdrożenia dyrektywy INSPIRE

# Spis treści

**1 Informacje ogólne o dokumencie ..... 3**

*1.1 Adresaci dokumentacji ..... 3*

*1.2 Zakres przedmiotowy dokumentacji ..... 3*

**2 Zadanie I – Tworzenie dokumentów XML. ..... 3**

**3 Zadanie 2 – Xpath – język ścieżek. ..... 5**

**4 Zadanie 3 – Opracowanie schematu XSD dla pliku XML. ..... 6**

**5 Zadanie 4 – Walidacja pliku GML. ..... 8**

**6 Zadanie 5 – XSLT w praktyce. ..... 9**

**7 Zadanie 6 – HALE – przykładowa harmonizacja. ..... 10**

**8 Harmonizacja w HALE – zadanie dodatkowe. ..... 20**

# 1 Informacje ogólne o dokumencie

## 1.1 Adresaci dokumentacji

Adresatami instrukcji są użytkownicy, którzy są lub mogą być zainteresowani przetwarzaniem danych przestrzennych zapisanych w formacie XML/GML.

## 1.2 Zakres przedmiotowy dokumentacji

Materiały szkoleniowe w części ćwiczeniowej składają się z kilku zadań wykonywanych w trakcie szkolenia. Zadania te rozpoczyna, omawia i nadzoruje prowadzący, kierując uczestników do celu i omawiając zagadnienia niezbędne dla pełnego zrozumienia tematu. Opracowany program ma również na celu przekazanie uczestnikom kompetencji i wiedzy, dzięki którym będą potrafili samodzielnie wykorzystywać poznane narzędzia.

Zadania do realizacji w czasie ćwiczeń, zostały przygotowane w porządku, wskazującym zalecaną kolejność ich wykonania. Każde z nich oznaczone kolejnym numerem, tj. 1, 2, … oraz tytułem, składa się z dwu części A i B. Część A w kilku zdaniach przedstawia problem i precyzuje zadanie do rozwiązania, stawiając wyraźny cel i określając rezultat. Cześć B natomiast podaje w sposób ogólny kolejne kroki, prezentujące poprawny plan realizacji zadania. Agenda z części B jest realizowana przez wszystkich uczestników, a szczegółowo omawia ją i wykonuje prowadzący. Materiał przedstawiający ćwiczenia został również zilustrowany zrzutami ekranowymi.

# 2 Zadanie I – Tworzenie dokumentów XML.

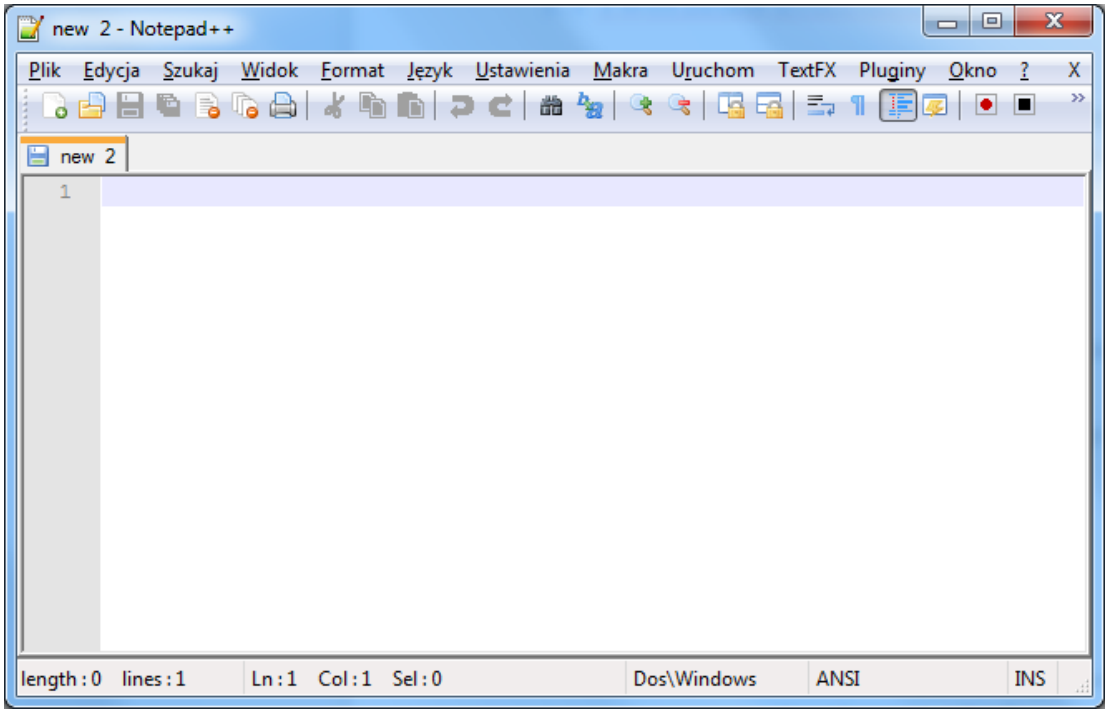
A) Opis zadania:

Celem pierwszego, wprowadzającego do zagadnienia XML, ćwiczenie będzie utworzenie prostego dokumentu XML w edytorze tekstu Notepad++ z wykorzystaniem dodatku XML Tools. Dokument powinien być poprawnie sformatowanym dokumentem XML (well-formed) opisującym dane kontaktowe i zawierającym:

- Nagłówek XML,
- Element nadrzędny „kontakty” ,
- Dwa elementy „kontakt” reprezentujące pojedyncze kontakty zawierające:
  - ✓ atrybut „plec”,
  - ✓ element „imie”,
  - ✓ element „nazwisko”,
  - ✓ element „telefon”,
  - ✓ element „mail”,
  - ✓ element złożony „adres” posiadający elementy potomne:
    - ✓ element „ulica”,
    - ✓ element „miejscowosc”,
    - ✓ element „kod-pocztowy”.
- Elementy i atrybuty powinny być wypełnione przykładowymi danymi.

B) Realizacja:

a) Pierwszym krokiem jest uruchomienie programu Notepad++ posiadającego zainstalowane rozszerzenie XML Tools.




*Rysunek 1 Okno programu Notepad++*

b) W edytorze należy wskazać, że tworzony plik będzie plikiem XML o kodowaniu „UTF-8”. W tym celu z menu „Format” należy zaznaczyć opcje „Koduj w UTF-8 (bez BOM)”. Następnie w menu „Język” wybrać „XML”.

c) Pierwszym etapem tworzenia nowego pliku XML powinno być wprowadzenie nagłówka XML deklarującego wersję i sposób kodowania (który jest zgodny z tym ustalonym w edytorze tekstu).

```
<?xml version="1.0" encoding="UTF-8" ?>
```

d) Po zdefiniowaniu nagłówka tworzonego pliku można przystąpić do edycji zawartości. Elementem wymaganym do poprawnego utworzenia pliku XML jest element nadrzędny, który w tym ćwiczeniu powinien nazywać się „kontakty”. Pod nagłówkiem należy dopisać więc:

```
<?xml version="1.0" encoding="UTF-8" ?>
<kontakty></kontakty>
```

e) Wewnątrz elementu kontaktu należy dodać element „kontakt” reprezentujący pojedynczy kontakt.

```
<kontakty xmlns="http://test.org/test">
      <kontakt></kontakt>
</kontakty>
```

f) Element „kontakt” powinien posiadać elementy proste (posiadające zawartość) - „imie”, „nazwisko”, „telefon”, „mail” np.:

```
<?xml version="1.0" encoding="UTF-8" ?>
<kontakty xmlns="http://test.org/test">
      <kontakt>
            <imie>Janusz</imie>
            <nazwisko>Dzięcioł</nazwisko>
```

```
            <telefon>+4812345678</telefon>
            <mail>janusz.dzieciol@mail.com</mail>
        </kontakt>
</kontakty>
```

g) Kolejnym krokiem jest dodanie elementu złożonego „adres” wewnątrz elementu „kontakt” posiadającego elementy proste: „ulica”, „miejscowosc” oraz „kod-pocztowy”:

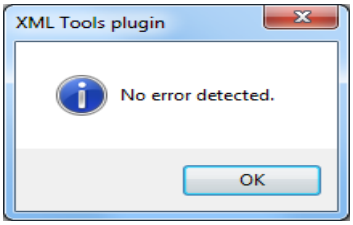
```
<adres>
      <ulica>Grabiszyńska 241a</ulica>
      <miejscowosc>Wrocław</miejscowosc>
      <kod-pocztowy>53-234</kod-pocztowy>
</adres>
```

h) Ostatnim krokiem jest dodanie do elementu „kontakt” atrybutu „plec”:

```
<kontakt plec="m">
```

i) Plik powinien zawierać jeszcze jedną pozycję kontaktu, którą należy zdefiniować podobnie jak w krokach 5-7. Po utworzeniu dokumentu należy do zapisać.

j) Poprawność pliku XML można zweryfikować wybierając opcję „Check XML syntax now” z menu „Pluginy→XML Tools”. Jeśli dokument został stworzony poprawnie to powinien pojawić się komunikat:




*Rysunek 2Test składniowy.*

# 3 Zadanie 2 – Xpath – język ścieżek.

A) Opis zadania:

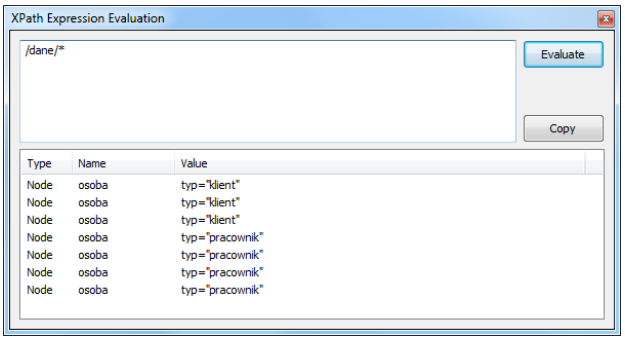
Ćwiczenie ma na celu zapoznanie uczestników warsztatów z językiem XPath wykorzystywanym w arkuszach przekształceń XSLT. Pozwala on na lokalizowanie i wydobywanie elementów z dokumentu XML.

B) Realizacja:

a) W celu przetestowania kilku zapytań języka XPath należy w edytorze Notepad++ otworzyć dokument „kontakty_duze.xml”. Zawiera on listę kontaktów do pracowników i klientów pewnej firmy.

b) Zapytania XPath można wprowadzać w okno aktywowane za pomocą opcji „Evaluate XMLPath expression dostępne z menu „Pluginy→XML Tools”. Po wprowadzeniu formuły i wciśnięciu przycisku „Evaluate” powinny pojawić się wyniki zapytania (o ile formuła była prawidłowa).

*c)* Wyświetlenie wszystkich elementów znajdujących się w elemencie nadrzędnym „dane” poprzez zapytanie */dane/**.




*Rysunek 3Wynik zapytania.*

d) Wyświetlenie wszystkich imion znajdujących się w kontaktach:
`//imie`

e) Wyświetlenie wszystkich imion i nazwisk:
`//imie | //nazwisko`

f) Wyświetlenie nazwisk klientów:
`//osoba[@typ="pracownik"]/nazwisko`

g) Wyświetlenie prywatnych numerów telefonów pracowników:
`//osoba[@typ="pracownik"]/telefon[@typ="prywatny"]`

h) Wyliczenie wszystkich osób:
`count(/dane/osoba)`

i) Nazwiska pracowników zarabiających więcej niż 4000 zł:
`//nazwisko[../pensja>4000]`

# 4 Zadanie 3 – Opracowanie schematu XSD dla pliku XML.

A) Opis zadania:

Celem ćwiczenia jest uzupełnienie brakujących definicji elementów w przygotowanym schemacie XML Schema walidującym wcześniej przygotowane pliki XML zawierające kontakty. Schemat posiada zdefiniowane:

- element nadrzędny „kontakty”,
- element „kontakt”,
- wystąpienie elementu „adres” bez definicji,
- wystąpienie atrybutu „plec” o typie „plec”, który nie jest zdefiniowany

W celu poprawnej walidacji należy więc dodać w schemacie:

- dodanie do elementu „kontakt” definicji elementów prostych „imie, „nazwisko”, „telefon” i „mail” o typach „xs:string”,
- zdefiniowanie typu prostego „plec”, który będzie enumeracją o dozwolonych wartościach {„m”,”k”},
- zdefiniowanie elementu złożonego „adres”, który będzie zawierał elementy proste (typ xs:string): „ulica”, „miejscowosc”, „kod-pocztowy”.

B) Realizacja:

a) Elementy proste „imie”, „nazwisko”, „telefon” i „mail” powinny być zdefiniowane wewnątrz tagu <xs:sequence> w definicji elementu „kontakt” jako:
<xs:element name="**imie**" type="**xs:string**"/>

Należy uzupełnić schemat o definicje tych elementów na podstawie przykładu powyżej.

b) Definicja typu prostego „plec”, który ma być enumeracją powinna znajdować się bezpośrednio w elemencie nadrzędnym schematu, czyli w xs:schema. Definicja enumeracji:

```
<xs:simpleType name="plec">
  <xs:restriction base="xs:string">
      <xs:enumeration value="m" />
      <xs:enumeration value="k" />
  </xs:restriction>
</xs:simpleType>
```

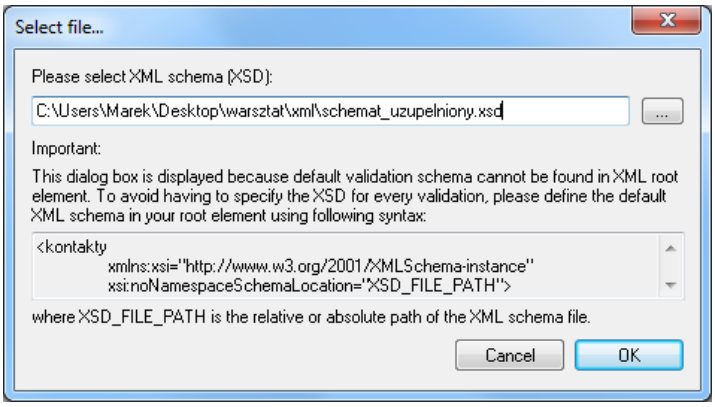
c) Ostatnim krokiem w edytowaniu schematu jest dodanie elementu złożonego adres (w elemencie xs:schema). Wewnątrz elementu „xs:sequence” należy umieścić definicje elementów prostych „ulica”, „miejscowosc”, „kod-pocztowy” o typie „xs:string”, podobnie jak dla elementu „kontakt”.

```
<xs:element name="adres" type="adres"/>
<xs:complexType name="adres">
      <xs:sequence>

      </xs:sequence>
</xs:complexType>
```

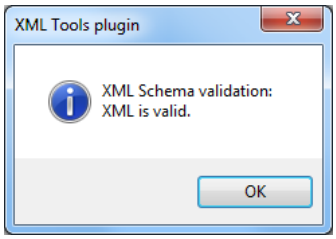
d) Po dokonaniu edycji plik schematu należy zapisać jako plik z rozszerzeniem „.xsd”.

e) Kolejnym krokiem jest dokonanie walidacji wcześniej utworzonego pliku XML zawierającego kontakty. W tym celu w edytorze Notepad++ należy otworzyć ten plik, a następnie z menu „Pluginy→XML Tools ” wybrać opcję „Validate now”.




*Rysunek 4Walidacja wcześniej uzupełnionego pliku.*

f) Należy wybrać plik schematu i zatwierdzić poprzez wciśnięcie przycisku „OK”. Jeśli walidacja przebiegnie poprawnie wyświetlony zostanie komunikat:




*Rysunek 5- Rezultat walidacji pliku XML schematem XSD.*

# 5 Zadanie 4 – Walidacja pliku GML.

A) Opis zadania:

Zadanie ma na celu poprawę błędnego pliku GML do postaci zgodnej z danym schematem XSD. Poprawa musi również zawierać zlikwidowanie błędów syntaktycznych pliku. Plik nie przechodzi walidacji zarówno sam w sobie, jak i względem schematu XSD.

Pliki ćwiczeniowe dostępne są w następującej lokalizacji:

\\materialy szkoleniowe\Zadanie 4 - Walidacja pliku GML\plik_z_bledami.gml

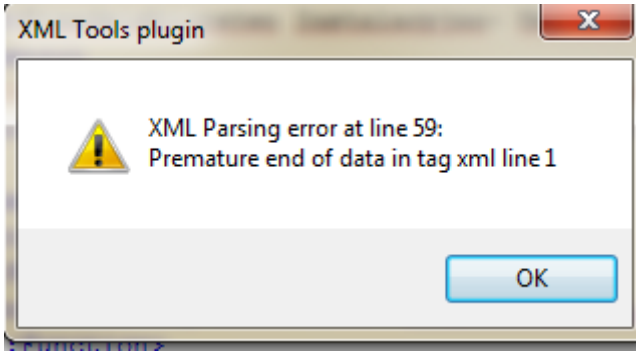
\\materialy szkoleniowe\Zadanie 4 - Walidacja pliku GML\ProductionAndIndustrialFacilities.xsd

Walidacja syntaktyczna pliku będzie przeprowadzana w aplikacji Notepad++, natomiast zgodność ze schematem XSD w ogólnodostępnym walidatorze internetowym pod adresem: http://www.utilities-online.info/xsdvalidation/

B) Realizacja:

Plik zawiera dwa obiekty przynależne do tematu Production And Industrial Facilities. Do kodu pliku wprowadzono 6 błędów dla ułatwienia zaznaczono ich przybliżoną lokalizację komentarzami. Komentarze w plikach XML mają następującą składnie: <!—treść komentarze →.

W celu sprawdzenia składni pliku GML w aplikacji Notepad++ należy skorzystać z możliwości pluginu „XML Tools -> Check XML syntax now". Jeśli składnia jest nieprawidłowa program poinformuje o błędzie.

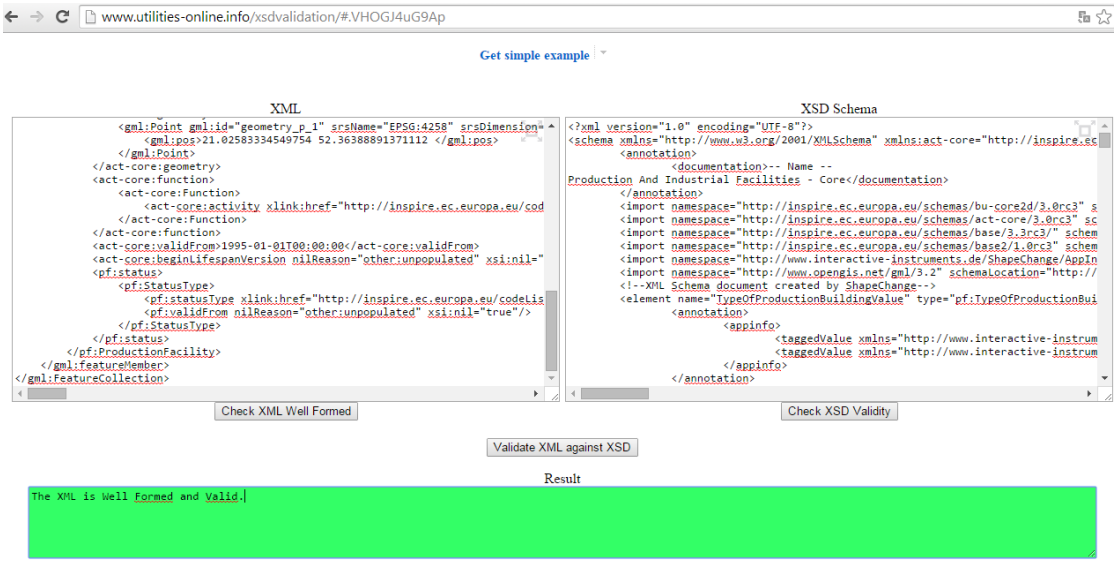



*Rysunek 6 - Przykładowy komunikat o błędzie.*

Gdy aplikacja Notepad++ nie wykaże żadnych nieprawidłowości należy przejść do walidacji. W tym celu kopiujemy zawartość pliku GML do schowka, a następnie wklejamy w oknie oznaczonym XML. Klikamy na „Check XML well formed". Jeśli test przechodzi prawidłowo, należy skopiować zawartość pliku XSD

do sąsiedniego okna „XML Schema”. Sprawdzenie poprawności schematu XSD poprzez „Check XSD Validity” jest opcjonalne, gdyż korzystamy z gotowego schematu.

Finalnym krokiem jest konfrontacja GML z XSD poprzez polecenie „Validate XML against XSD”. Jeśli plik spełnia wymagania stawiane przez schemat użytkownik powinien otrzymać następującą informację: **The XML is Well Formed and Valid.**




*Rysunek 7- Pozytywna walidacja pliku GML.*

# 6 Zadanie 5 – XSLT w praktyce.

A) Opis zadania:

Celem zadania jest praktyczne zapoznanie się ze składnią transformacji XSL. Przykładowy plik XSL dla tematu Obiekty Produkcyjne i Przemysłowe został bogato opisany komentarzami, które zostaną prześledzone i omówione przez trenera. Przeprowadzanie drobnych modyfikacji w kodzie i weryfikacja rezultatów.

B) Realizacja:

Lokalizacja do plików niezbędnych do wykonania ćwiczenia:

- Plik z transformacją XSL: \\materialy szkoleniowe\Zadanie 5 - Opracowanie pliku XSL\plik_transformacji.xsl
- Plik wykonywalny dla przeprowadzenia procesu harmonizacji: \\materialy szkoleniowe\Zadanie 5 - Opracowanie pliku XSL\process.bat
- Schemat XSD niezbędny do weryfikacji poprawności harmonizacji: \\materialy szkoleniowe\Zadanie 5 - Opracowanie pliku XSL\schemat xsd\ProductionAndIndustrialFacilities.xsd
- Dane źródłowe do transformacji: \\materialy szkoleniowe\Zadanie 5 - Opracowanie pliku XSL\1_dane_wejsciowe\oczyszczalnie.gml
- Folder zapisu danych zharmonizowanych: \\materialy szkoleniowe\Zadanie 5 - Opracowanie pliku XSL\2_wynik_transformacji

# 7 Zadanie 6 – HALE – przykładowa harmonizacja.

A) Opis zadania:

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.

B) Realizacja:

W celu realizacji zadania niezbędne są następujące dane źródłowe:

- schemat aplikacyjny danych źródłowych (XSD), dostępny w lokalizacji:

//Desktop/Zadanie 6 – HALE/dane wejściowe/oczyszczalnie.xsd

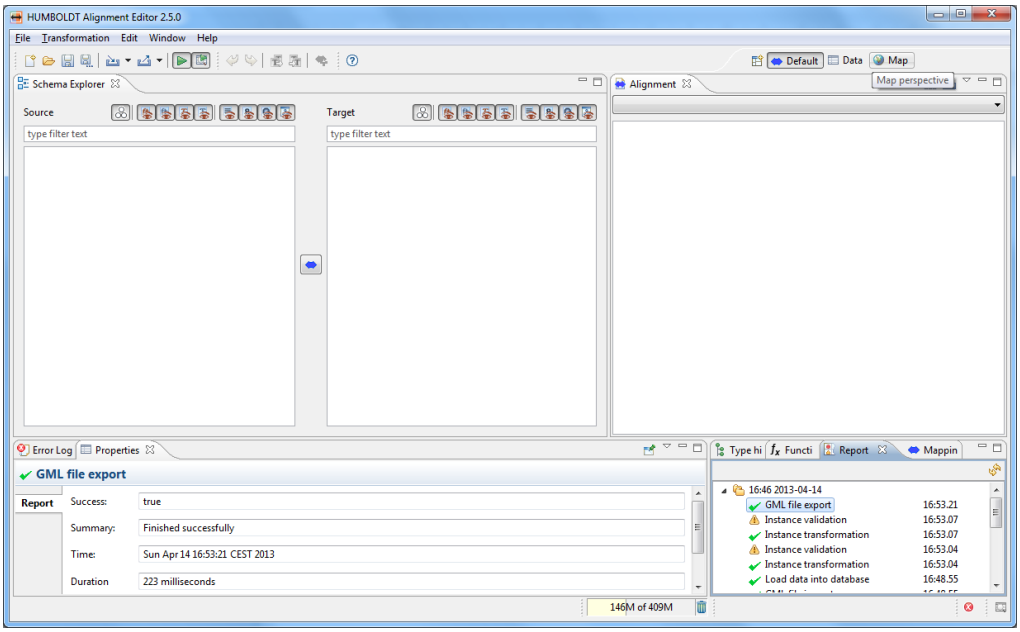
- dane źródłowe w formacie XML/GML, dostępne w lokalizacji:

//Desktop/Zadanie 6 – HALE/dane wejściowe/oczyszczalnie.gml

- docelowy schemat aplikacyjny zgodny z INSPIRE (XSD), dostępny w lokalizacji:

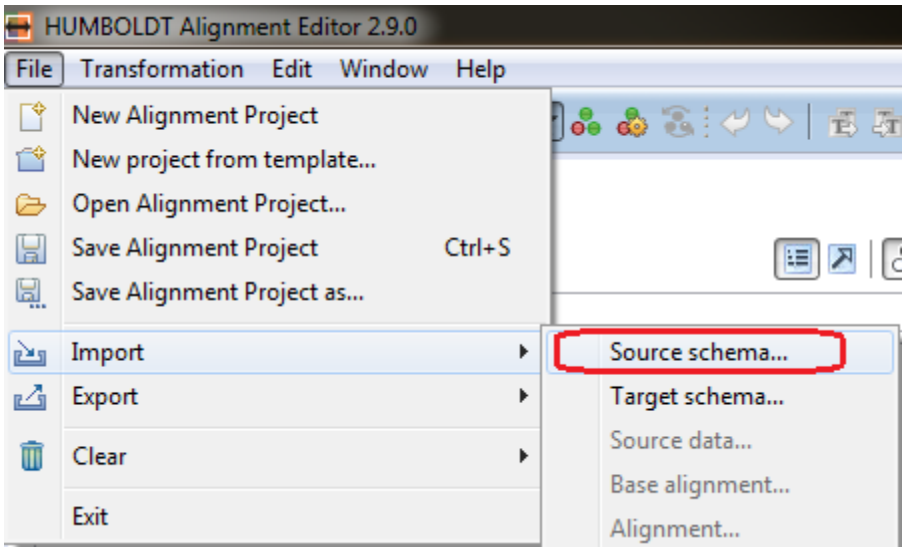
//Desktop/Zadanie 6 – HALE/dane wejściowe/ProductionAndIndustrialFacilities.xsd

- W pierwszej kolejności należy uruchomić aplikację HALE:



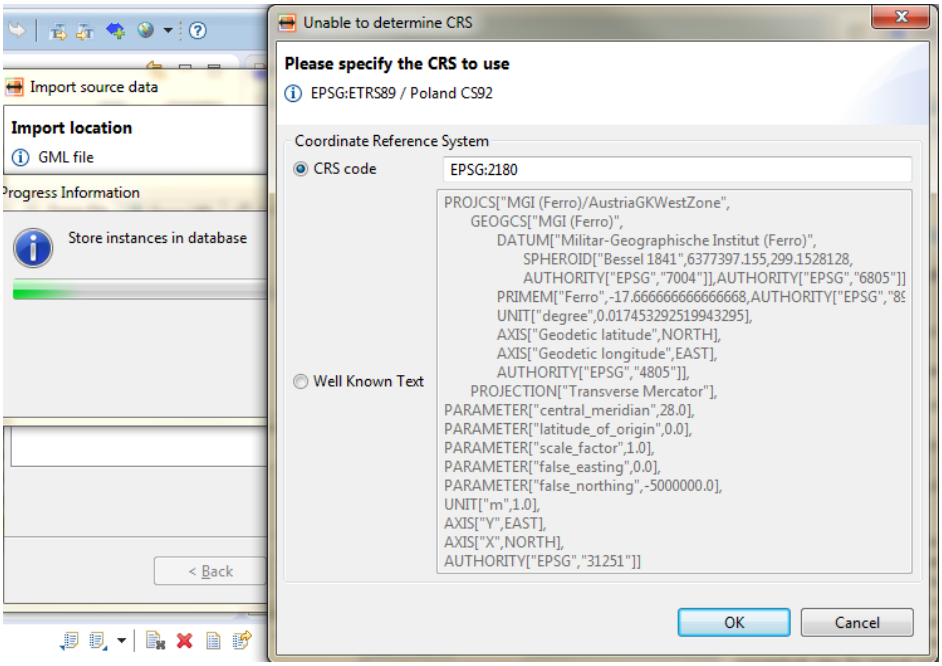

*Rysunek 8 - Okno aplikacji HALE.*

- Następnym krokiem jest załadowanie źródłowego schematu aplikacyjnego i źródłowych danych GML. Z menu „File→Import” należy wybrad opcję „Source Schema” i wskazad plik oczyszczalnie.xsd.

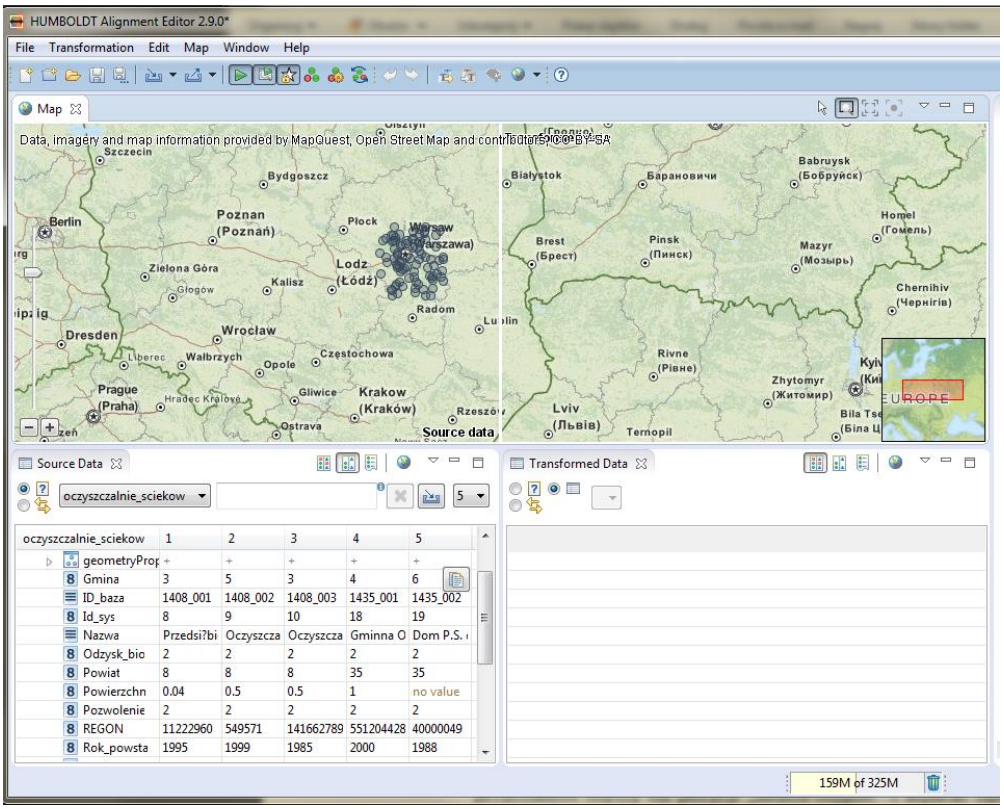



*Rysunek 9- wczytywanie schematu źródłowego.*

Następnie należy załadować dane źródłowe poprzez wybranie polecenia „Import -> Source data”. Po wskazaniu pliku GML użytkownik zostanie poproszony o wskazanie układu współrzędnych pliku. Należy podać układ EPSG:2180 (Poland CS1992).



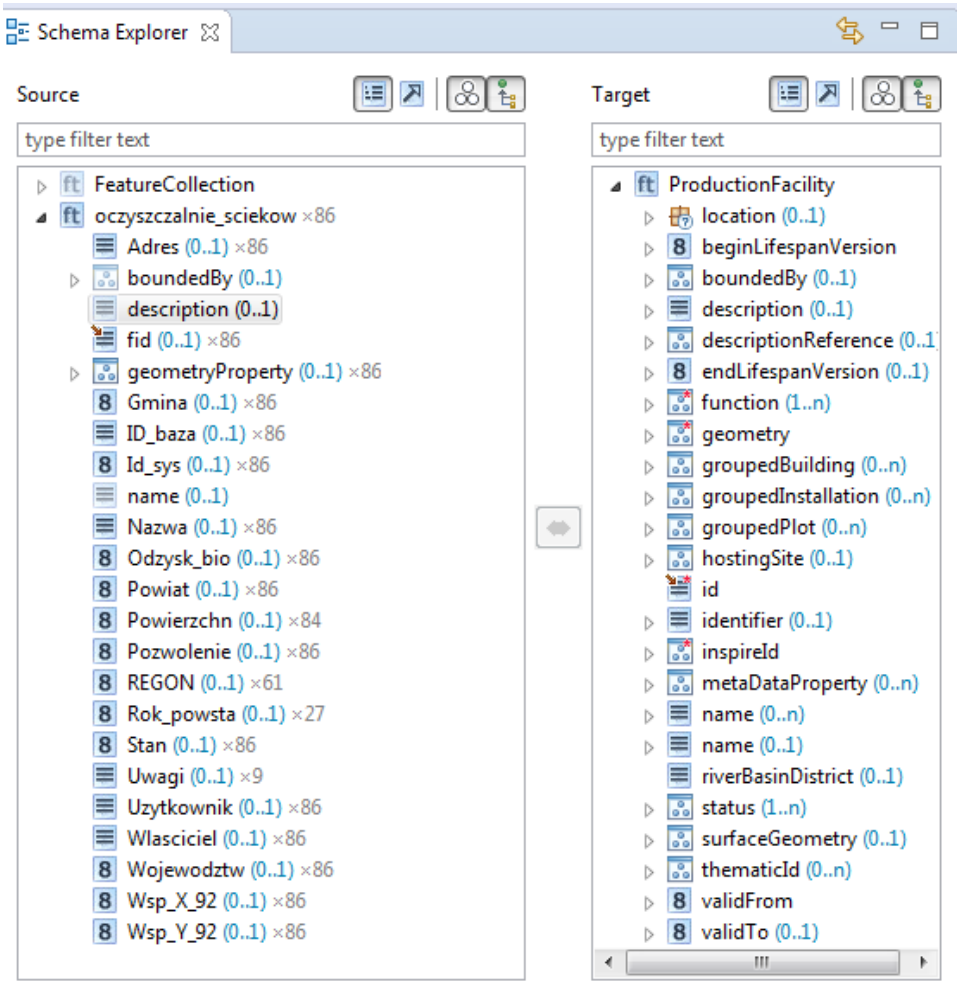

*Rysunek 10- Wskazanie układu współrzędnych.*




*Rysunek 11- Załadowane dane źródłowe w aplikacji HALE.*

Wczytywanie docelowego schematu aplikacyjnego zgodnego z INSPIRE. W tym celu należy wczytać plik ProductionAndIndustrialFacilities.xsd za pomocą znanej już funkcji „Import => Target schema".

Wybieramy „Edit mapping relevant target types" z paska narzędzi w celu wyświetlenia wszystkich wczytanych klas obiektów. W celu zwiększenia czytelności z zakładki [http://inspire.ec.europa.eu/schemas/pf/3.0rc3](http://inspire.ec.europa.eu/schemas/pf/3.0rc3)




*Rysunek 12 - Tabele ze schematami*

W celu przeprowadzenia harmonizacji należy przenieść założenia wynikające z tabeli mapowania do aplikacji HALE.

Ustalenia wynikające z tabeli mapowań dla obiektu Oczyszczalnie Ścieków (zestawienie zbiorcze, uproszczone. Nazwa atrybutu Inspire, następnie informacja na podstawie czego należy go uzupełnić):

- InspireID ->
  - LocalID na podstawie połączenia (konkatenacji) wartości stałej PF.ProductionFacility oraz wartości atrybutu ID_baza.
  - Namespace – przestrzeń nazw dla obiektów ze zbioru. W tym wypadku będzie to wartość stała PL.ZIPGL.325.3.PF.3.8. Składająca się z identyfikatora obiektu w ewidencji zbiorów i usług oraz frazy dotyczącej tematu INSPIRE. W tym wypadku „PF.3.8” – dwuliterowy skrót tematu oraz jego numer.
- Name -> wartość uzupełniana bez zmian na podstawie wartości atrybutu Nazwa.
- Geometry -> bezpośrednie mapowanie geometrii obiektów. Zarówno w zbiorze źródłowym, jak i docelowym są to obiekty punktowe.
- Function ->
  - Activity – uzupełnione zostanie wartością stałą dla tego typu obiektów. Wartość ta pochodzi z listy kodowej NACE (zalecanej przez INSPIRE w celu zachowania interoperacyjności) i jej wartość wynosi „E.37.0.0”.

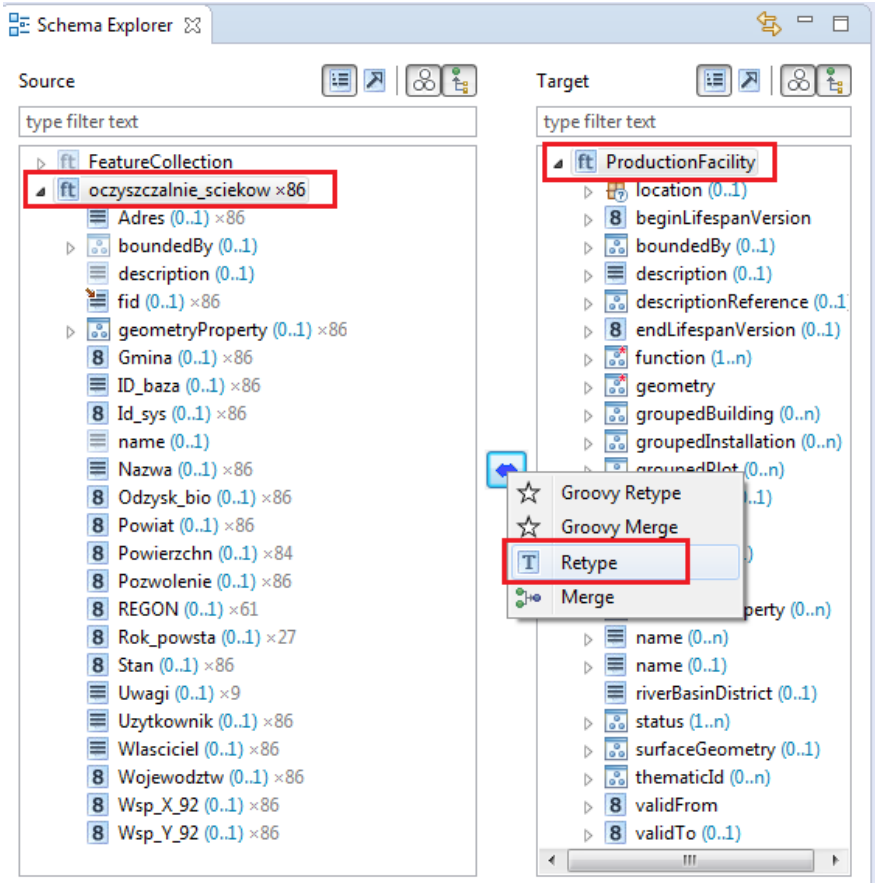
- Description – opcjonalny opis słowny wartości zawartej w Activity. W tym przypadku będzie to „Sawerage”.

- ❖ ValidFrom -> wartość bazuje na atrybucie „Rok powstania” z danych źródłowych. Dane podane są z dokładnością roczną. Dlatego też w celu spełnienia wymogów INSPIRE co do daty, przyjęte zostało założenie, że data ta odnosi się do dnia 1 stycznia. Stąd też do daty dodane zostało „-01-01 00:00:00”.
- ❖ ValidTo ->-> wartość bazuje na atrybucie „Rok zakończenia” z danych źródłowych. Dane podane są z dokładnością roczną. Dlatego też w celu spełnienia wymogów INSPIRE co do daty, przyjęte zostało założenie, że data ta odnosi się do dnia 1 stycznia. Stąd też do daty dodane zostało „-01-01 00:00:00”.
- ❖ BeginLifeSpanVersion -> atrybut nie zasilany. Otrzymuje stereotyp nilReason="other:unpopulated" dla wszystkich obiektów.
- ❖ Status -> mapowanie wartości listy kodowej pochodzącej ze zbioru źródłowego do listy kodowej przewidzianej w schemacie INSPIRE. Na podstawie atrybutu Stan. Mapowane są dwie wartości dla stan = 1 (działający) wartość ta wynosić będzie „FUNCTIONAL„, natomiast dla stan = 2(zamknięty) wartość ta wynosić będzie „DECOMISSIONED”.

Proces mapowania podzielić można dla uproszczenia i zachowania porządku na kilka etapów:

a) Przypisanie odpowiadających sobie klas pomiędzy schematami.

Pierwszym krokiem będzie zmapowanie klas sobie odpowiadających – w tym wypadku „oczyszczalnie ścieków => Production Facility”. Należy wykorzystać funkcję „retype” (rysunek 11 ).




*Rysunek 13- Mapowanie klas.*

b) Uzupełnienie wartości stałych

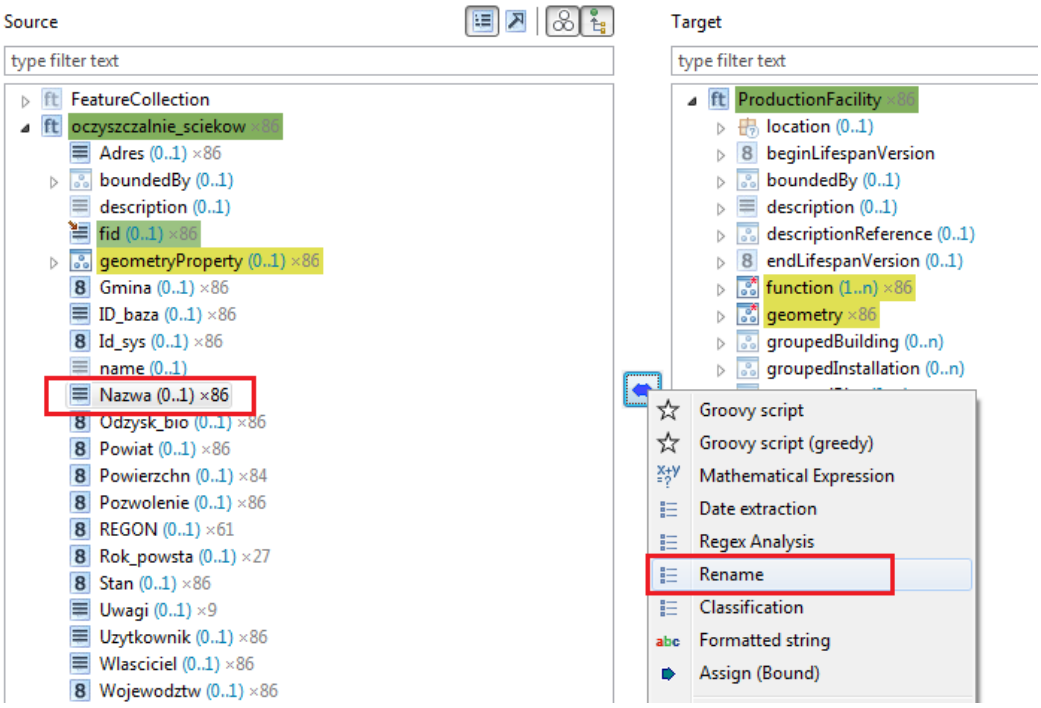
Uzupełniamy wartości, które można przyjąć za wartość stałą dla wszystkich obiektów w klasie. Czyli uzupełnieniu ulegną elementy takie jak: function (activity: E37.0.0 oraz description: Sawerage) oraz namespace.

Przypisanie stałych wartości odbywa się za pomocą polecenia Assign (bound), przy czym należy wybrać pole ze schematu wejściowego, które musi być uzupełnione, aby wartość została dodana do schematu docelowego. W tym przypadku wybrać należy atrybut, który zawsze jest uzupełniany – np. ID obiektu (rysunek 12).

*Rysunek 14- przykład Assign Bound*

c) Mapowanie atrybutów, których wartości przechodzą pomiędzy schematami w sposób nienaruszony.

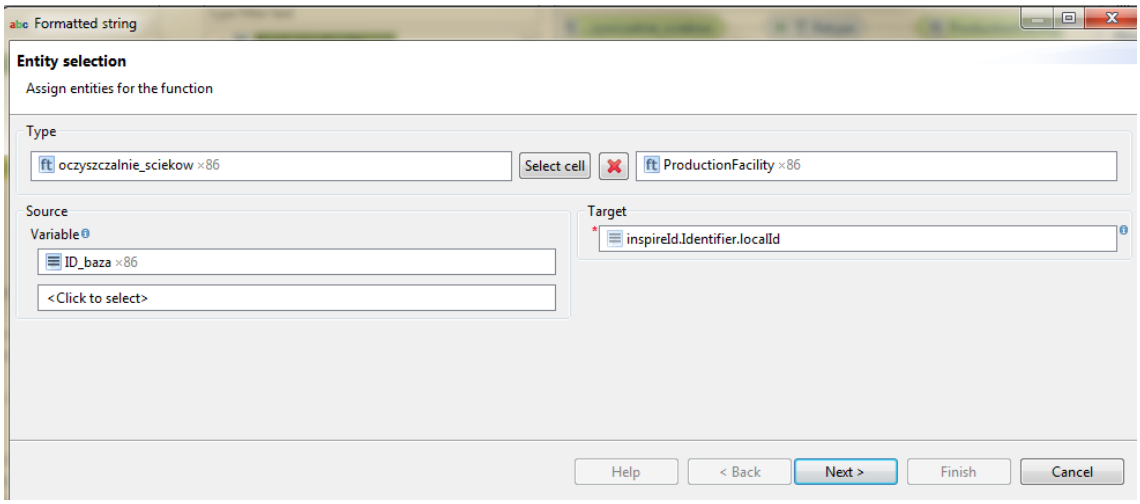
Mapowanie takie odbywa się poprzez funkcję „RENAME" – która w praktyce kopiuje wartość ze wskazanego źródłowego atrybutu do docelowego atrybutu. W celu uzyskania okna pomocy dotyczącego jakiejkolwiek funkcji należy wskazać funkcję po czym wybrać „Dynamic Help" – F1. Dotyczy atrybutu „nazwa -> name".



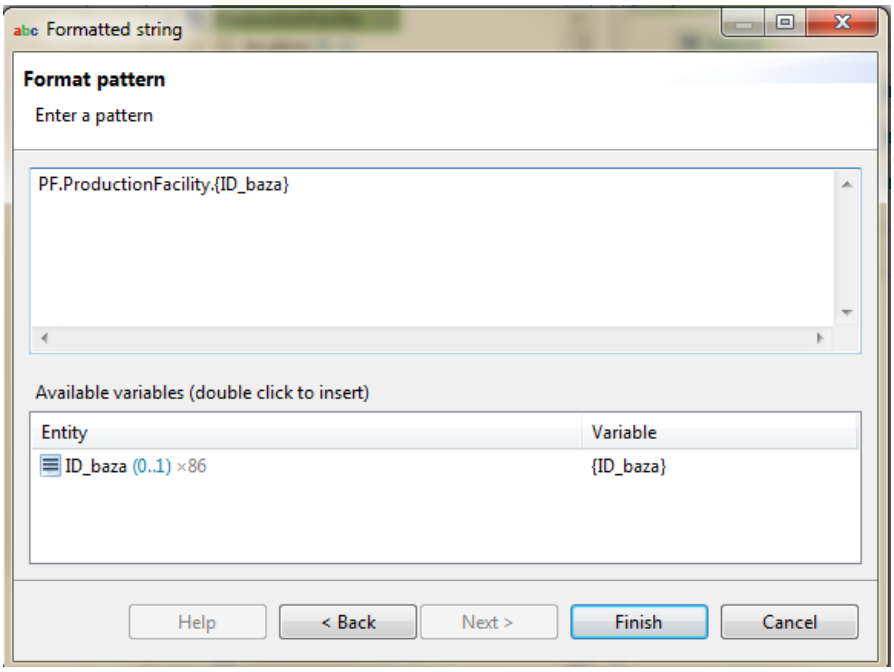

*Rysunek 15- kopiowanie atrybutów metodą 'Rename'.*

d) Mapowanie atrybutów, w przypadku, których w zbiorze docelowym powstaje nowa wartość powstała poprzez np. konkatenację atrybutów ze zbioru źródłowego.

Punkt dotyczy uzupełnienia localID, validFrom oraz validTo. Należy wybrać zasilany atrybut oraz funkcję Fromatted String. Funkcja ta buduje wartość na podstawie podanych wzorców oraz zmiennych pochodzących z danych źródłowych.




*Rysunek 16- Budowanie localID 1/2.*



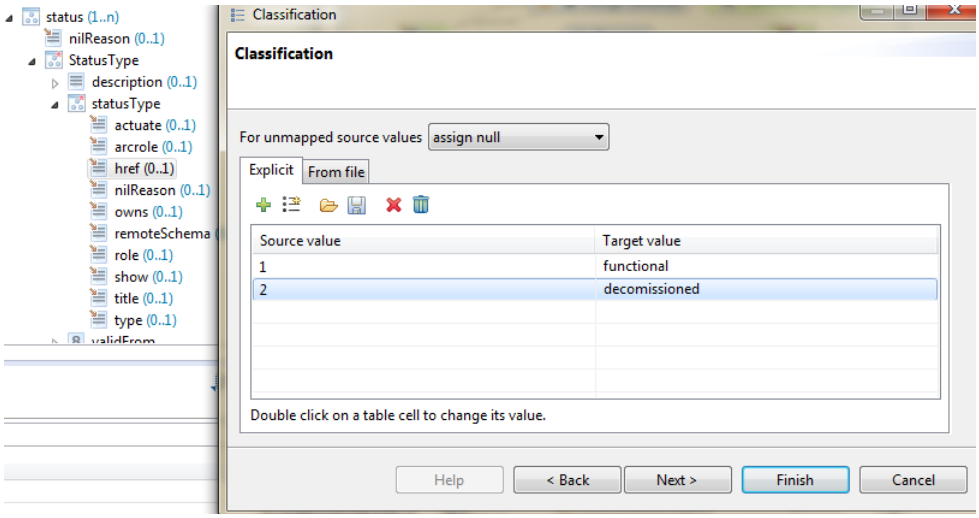

*Rysunek 17- Budowanie localID 2/2*

Analogicznie przy pomocy tej samej funkcji należy uzupełnić wartości dla atrybutów „validFrom” oraz „validTo”.

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 list. Dotyczy atrybutu status.

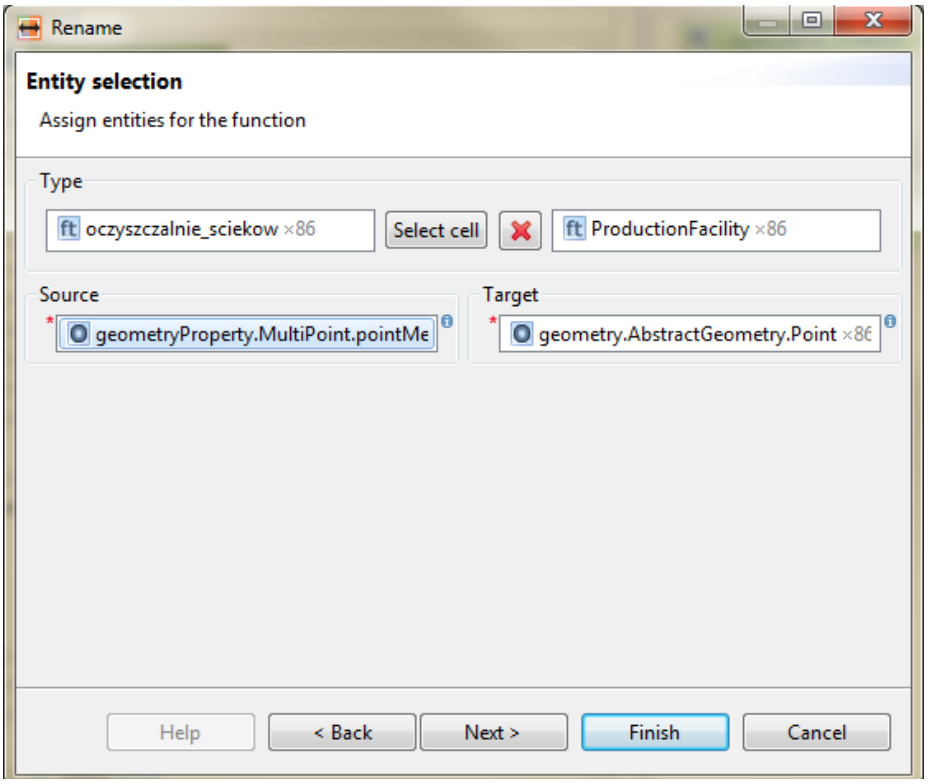
Do przypisania sobie odpowiadających atrybutów wykorzystywana jest funkcja klasyfikacji „Classification”. Klasyfikację możemy ustalić ręcznie od podstaw lub wczytać z pliku (o ile taki plik został przygotowany wcześniej).



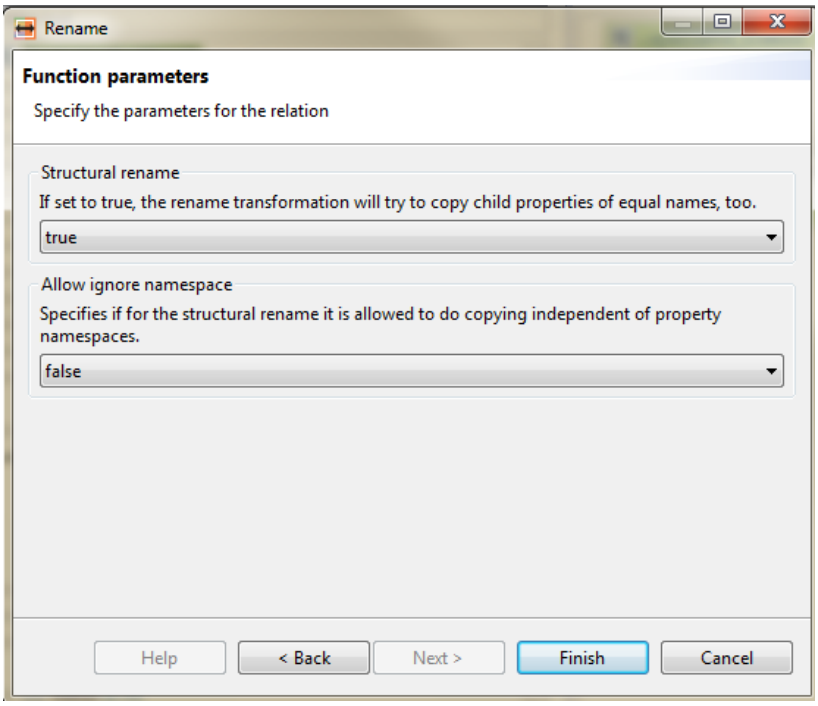

*Rysunek 18- Przykład klasyfikacji.*

f) Mapowanie geometrii pomiędzy schematami.

W obu schematach geometria reprezentowana jest w sposób punktowy. Należy wskazać odpowiadające sobie węzły zawierające informacje punktowe.

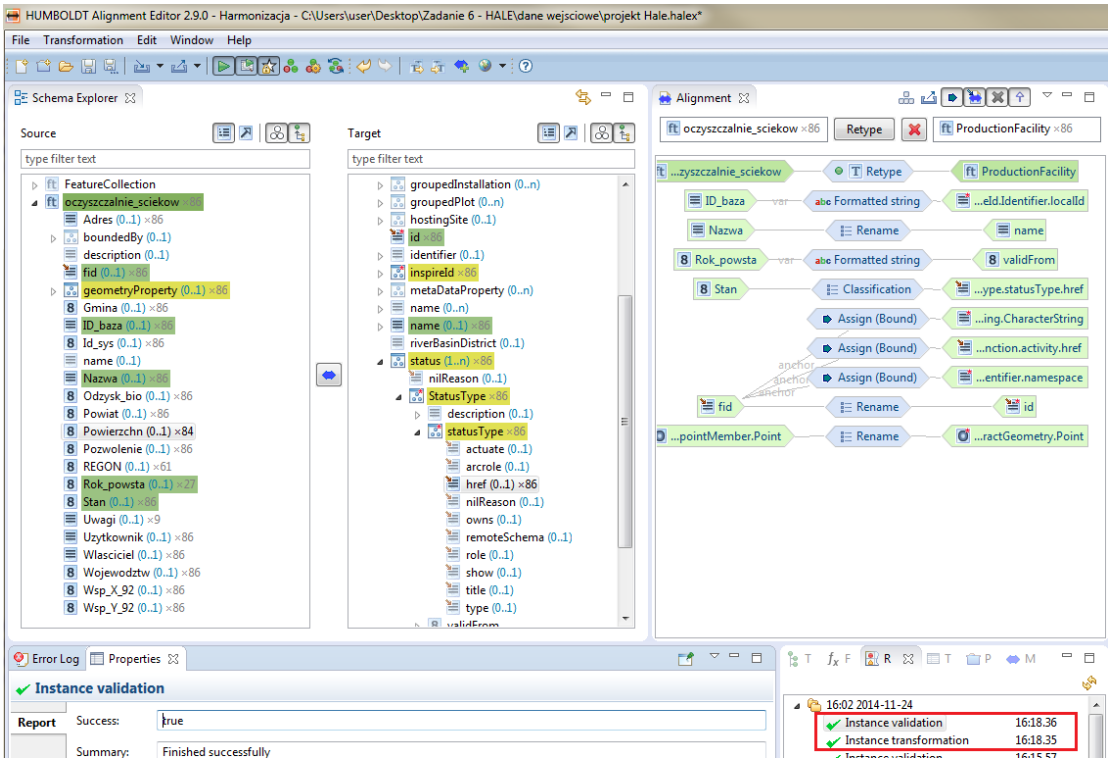



*Rysunek 19-Mapowanie geometrii ½.*




*Rysunek 20 - mapowanie geometrii 2/2.*

W tym miejscu dysponujemy już kompletnym schematem mapowania klasy obiektów do schematu INSPIRE. Dzięki wbudowanym funkcjonalnościom aplikacji 'na żywo' sprawdzana jest poprawność procesu i walidacja rezultatu. Odpowiadają za to "transformation => Live transformation" oraz "transformation => Validate Transformed Instances".



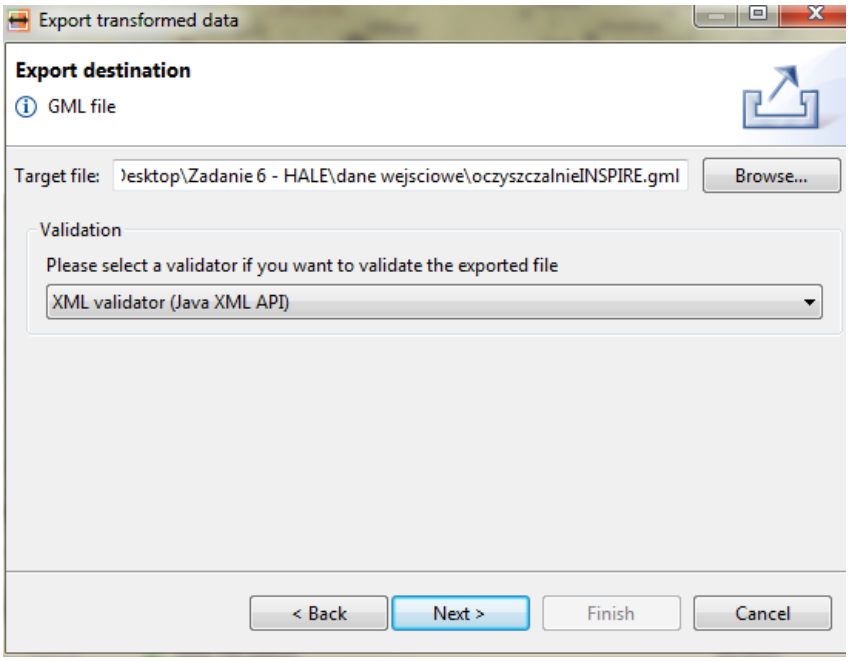

*Rysunek 21- Kompletny proces transformacji.*

Ostatnim krokiem jest eksport przetransformowanych danych do pliku GML. W tym celu należy wybrać z menu File -> Export -> Transformed Data. Po czym użytkownik przeniesiony zostaje do kreatora eksportu danych. W tym celu należy kolejno:

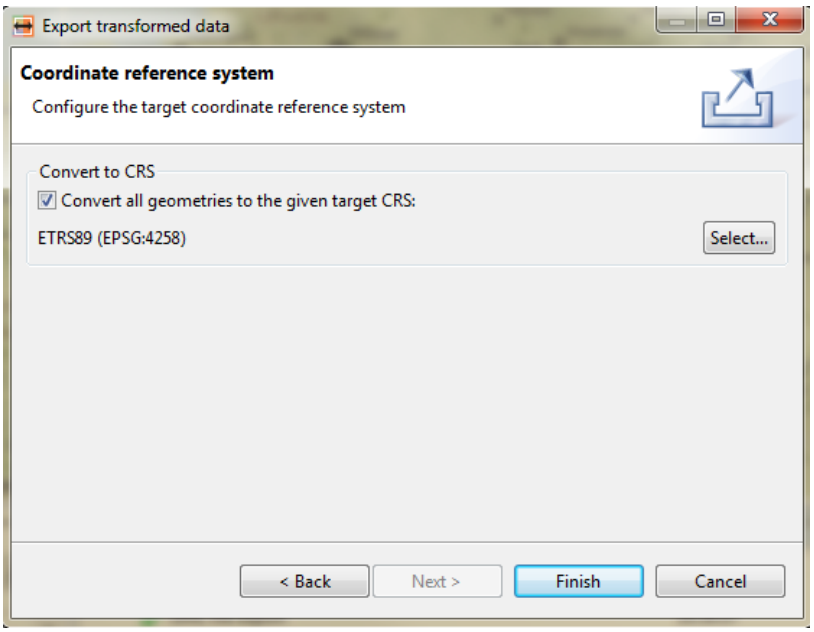
wybrać docelowy format danych – GML (Feature Collection).

Folder zapisu zharmonizowanego pliku oraz nadać nazwę (sugerowana nazwa - oczyszczalnieINSPIRE.gml),

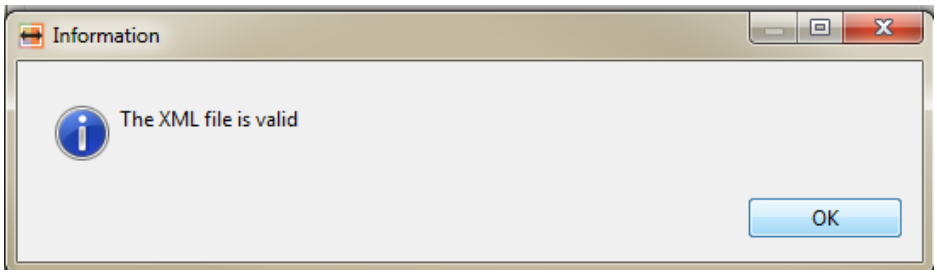
Wybór walidatora XML (jeśli chcemy zwalidować przed eksportem) oraz docelowego układu współrzędnych pliku – w przypadku danych zgodnych z INSPIRE będzie to ETRS89 – EPSG:4258.



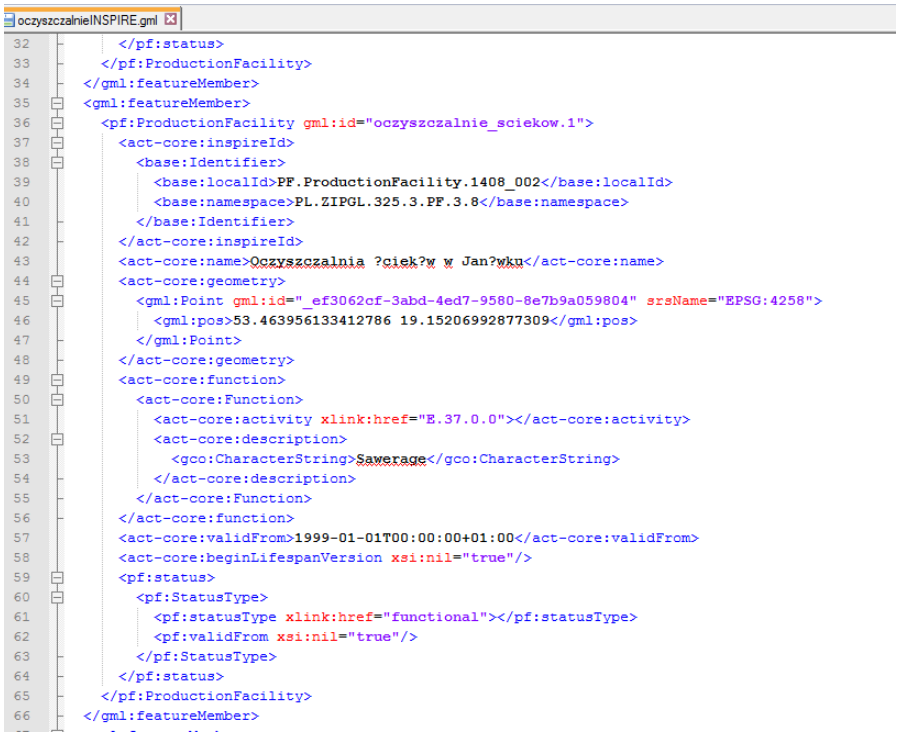

*Rysunek 22 - Export do pliku GML 1/3*




*Rysunek 23 Export do pliku GML 2/3*




*Rysunek 24- Pozytywny wynik walidacji pliku.*



```
oczyszczalnieINSPIRE.gml
32        </pf:status>
33      </pf:ProductionFacility>
34    </gml:featureMember>
35    <gml:featureMember>
36      <pf:ProductionFacility gml:id="oczyszczalnie_sciekow.1">
37        <act-core:inspireId>
38          <base:Identifier>
39            <base:localId>PF.ProductionFacility.1408_002</base:localId>
40            <base:namespace>PL.ZIPGL.325.3.PF.3.8</base:namespace>
41          </base:Identifier>
42        </act-core:inspireId>
43        <act-core:name>Oczyszczalnia ?ciek?w w Jan?wku</act-core:name>
44        <act-core:geometry>
45          <gml:Point gml:id="_ef3062cf-3abd-4ed7-9580-8e7b9a059804" srsName="EPSG:4258">
46            <gml:pos>53.463956133412786 19.15206992877309</gml:pos>
47          </gml:Point>
48        </act-core:geometry>
49        <act-core:function>
50          <act-core:Function>
51            <act-core:activity xlink:href="E.37.0.0"></act-core:activity>
52            <act-core:description>
53              <gco:CharacterString>Sawerage</gco:CharacterString>
54            </act-core:description>
55          </act-core:Function>
56        </act-core:function>
57        <act-core:validFrom>1999-01-01T00:00:00+01:00</act-core:validFrom>
58        <act-core:beginLifespanVersion xsi:nil="true"/>
59        <pf:status>
60          <pf:StatusType>
61            <pf:statusType xlink:href="functional"></pf:statusType>
62            <pf:validFrom xsi:nil="true"/>
63          </pf:StatusType>
64        </pf:status>
65      </pf:ProductionFacility>
66    </gml:featureMember>
```

*Rysunek 25 -Wynikowy plik XML.*

# 8 Harmonizacja w HALE – zadanie dodatkowe.

A) Opis zadania:

Na podstawie procesu opisanego w punkcie numer 7, proszę samodzielnie przeprowadzić transformację danych dotyczących klasy obiektów ELK_Elektrownia.

B) Realizacja:

W celu realizacji zadania niezbędne są następujące dane źródłowe:

- schemat aplikacyjny danych źródłowych (XSD), dostępny w lokalizacji: //Desktop/ Zadanie 7 - HALE (bonus)/dane wejściowe/elektrownia.xsd
- dane źródłowe w formacie XML/GML, dostępne w lokalizacji: //Desktop/ Zadanie 7 - HALE (bonus)/dane wejściowe /elektrownia.gml
- docelowy schemat aplikacyjny zgodny z INSPIRE (XSD), dostępny w lokalizacji: //Desktop/ Zadanie 7 - HALE (bonus)/dane wejściowe /ProductionAndIndustrialFacilities.xsd

Ustalenia wynikające z tabeli mapowań dla obiektu Oczyszczalnie Ścieków (zestawienie zbiorcze, uproszczone. Nazwa atrybutu Inspire, następnie informacja na podstawie czego należy go uzupełnić):

- InspireID ->
  - LocalID na podstawie połączenia (konkatenacji) wartości stałej PF.ProductionFacility.ELK. oraz wartości atrybutu ID_baza.
  - Namespace – przestrzeń nazw dla obiektów ze zbioru. W tym wypadku będzie to wartość stała PL.ZIPGL.325.3.PF.3.8. Składająca się z identyfikatora obiektu w ewidencji zbiorów i usług oraz frazy dotyczącej tematu INSPIRE. W tym wypadku „PF.3.8" – dwuliterowy skrót tematu oraz jego numer.
- Name -> wartość uzupełniana bez zmian na podstawie wartości atrybutu Nazwa.
- Geometry -> bezpośrednie mapowanie geometrii obiektów. Zarówno w zbiorze źródłowym, jak i docelowym są to obiekty punktowe.
- Function ->
  - Activity – uzupełnione zostanie wartością stałą dla tego typu obiektów. Wartość ta pochodzi z listy kodowej NACE (zalecanej przez INSPIRE w celu zachowania interoperacyjności) i jej wartość wynosi „D.35.11".
  - Description – opcjonalny opis słowny wartości zawartej w Activity. W tym przypadku będzie to „Production of electricity".
- ValidFrom -> wartość bazuje na atrybucie „Rok powstania" z danych źródłowych. Dane podane są z dokładnością roczną. Dlatego też w celu spełnienia wymogów INSPIRE co do daty, przyjęte zostało założenie, że data ta odnosi się do dnia 1 stycznia. Stąd też do daty dodane zostało „-01-01 00:00:00".
- BeginLifeSpanVersion -> atrybut nie zasilany. Otrzymuje stereotyp nilReason="other:unpopulated" dla wszystkich obiektów.
- Status -> mapowanie wartości listy kodowej pochodzącej ze zbioru źródłowego do listy kodowej przewidzianej w schemacie INSPIRE. Na podstawie atrybutu Stan. Mapowane są dwie wartości dla stan = 1 (działający) wartość ta wynosić będzie „FUNCTIONAL„, natomiast dla stan = 2(zamknięty) wartość ta wynosić będzie „DECOMISSIONED".