Kompleksowa organizacja i przeprowadzenie szkoleń dotyczących wdrażania dyrektywy INSPIRE i budowy krajowej infrastruktury informacji przestrzennej dla pracowników administracji publicznej, w tym dla pracowników Służby Geodezyjnej i Kartograficznej

Materiały szkoleniowe

SZKOLENIE EKSPERCKIE

SESJA 3

WARIANT ZAAWANSOWANY

Dotyczy realizacji umowy nr ZP/BO-4-2500-2/GI-2500-18/2010 z dnia 2010-09-21

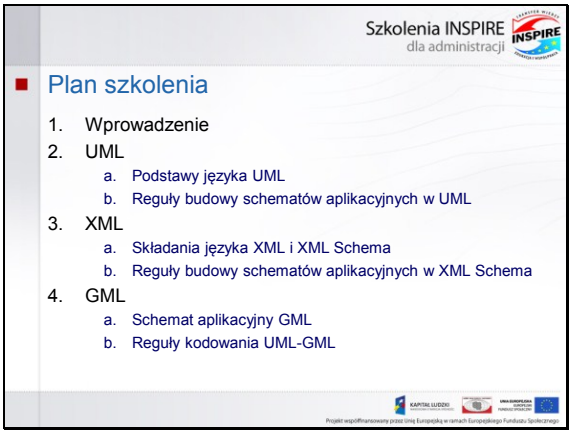
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

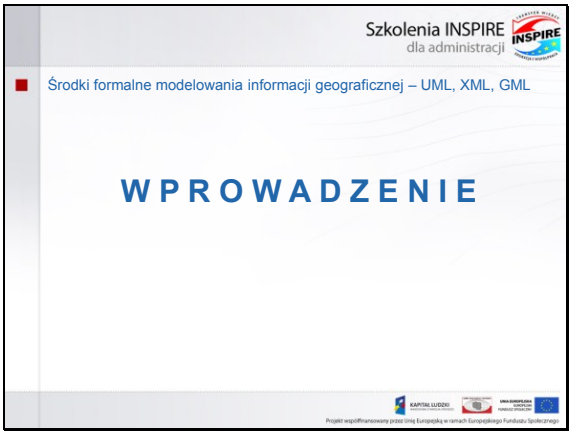
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spis treści

1. Wprowadzenie





Szkolenia INSPIRE dla administracji

Infrastruktura danych przestrzennych

- **Perspektywa danych** (ang. *data-centric view*):
  - schematy aplikacyjne,
  - metadane.
- **Perspektywa usług** (ang. *service-centric view*):
  - interoperacyjność,
  - architektura zorientowana na usługi.

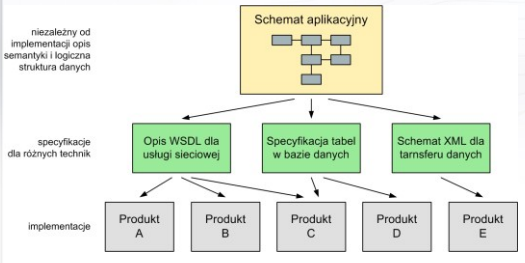
Szkolenia INSPIRE dla administracji

Perspektywa danych

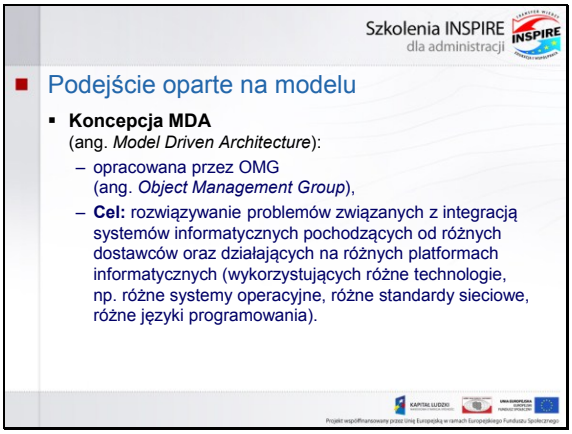
- **Podejście oparte na modelu** (ang. *Model Driven Approach*):
  - **niezależny od implementacji schemat aplikacyjny** zostaje odwzorowany na **różne specyfikacje** (wykorzystujące różne technologie, np. usługi sieciowe, relacyjne bazy danych, XML),
  - **specyfikacje** mogą zostać zaimplementowane (wdrożone) na **różnych platformach** sprzętowo-programowych.

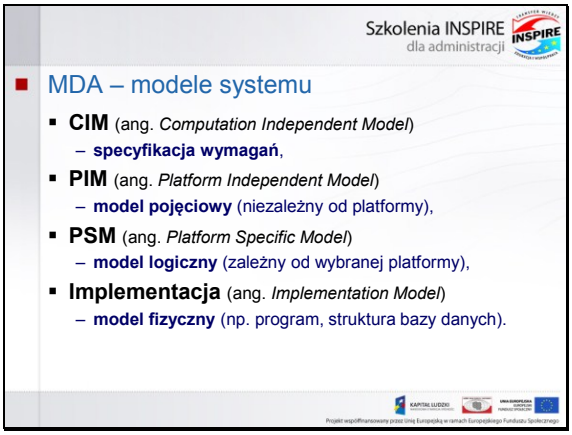
Szkolenia INSPIRE dla administracji

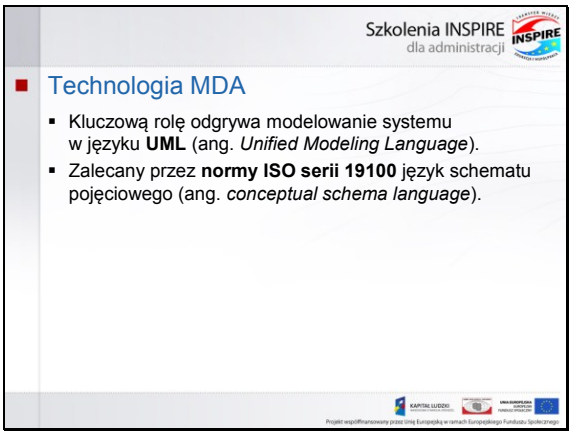
Podejście oparte na modelu

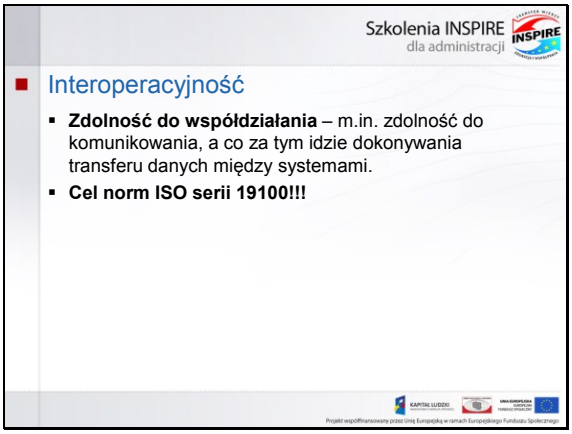


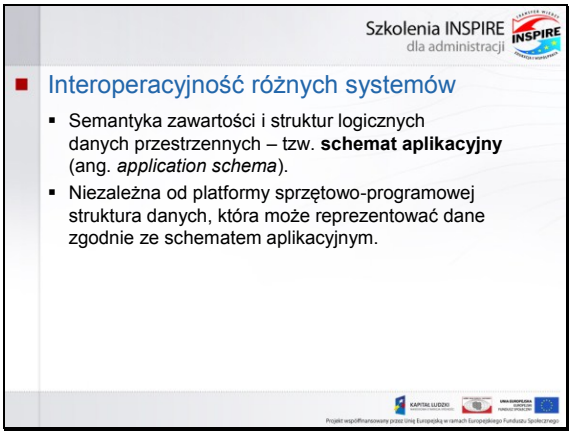
Podejście oparte na modelu
Źródło: prCEN/TR 15449, 2006.

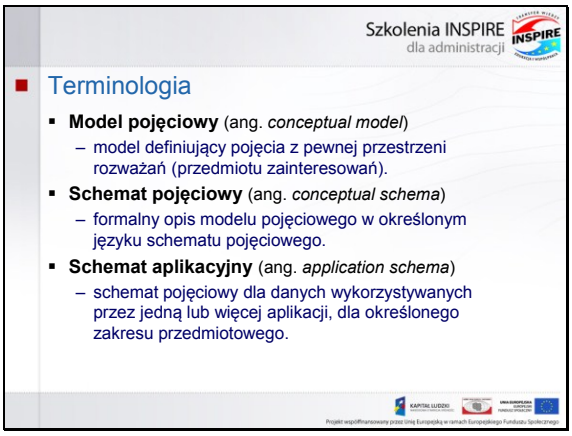

Szkolenia INSPIRE
dla administracji
Podejście oparte na modelu
Koncepcja MDA
(ang. Model Driven Architecture):
– opracowana przez OMG
(ang. Object Management Group),
– Cel: rozwiązywanie problemów związanych z integracją systemów informatycznych pochodzących od różnych dostawców oraz działających na różnych platformach informatycznych (wykorzystujących różne technologie, np. różne systemy operacyjne, różne standardy sieciowe, różne języki programowania).


Szkolenia INSPIRE
dla administracji
MDA – modele systemu
CIM (ang. Computation Independent Model)
– specyfikacja wymagań,
PIM (ang. Platform Independent Model)
– model pojęciowy (niezależny od platformy),
PSM (ang. Platform Specific Model)
– model logiczny (zależny od wybranej platformy),
Implementacja (ang. Implementation Model)
– model fizyczny (np. program, struktura bazy danych).


Szkolenia INSPIRE
dla administracji
Technologia MDA
Kluczową rolę odgrywa modelowanie systemu w języku UML (ang. Unified Modeling Language).
Zalecany przez normy ISO serii 19100 język schematu pojęciowego (ang. conceptual schema language).


Szkolenia INSPIRE
dla administracji
Interoperacyjność
Zdolność do współdziałania – m.in. zdolność do komunikowania, a co za tym idzie dokonywania transferu danych między systemami.
Cel norm ISO serii 19100!!!


Szkolenia INSPIRE
dla administracji
Interoperacyjność różnych systemów
Semantyka zawartości i struktur logicznych danych przestrzennych – tzw. schemat aplikacyjny (ang. application schema).
Niezależna od platformy sprzętowo-programowej struktura danych, która może reprezentować dane zgodnie ze schematem aplikacyjnym.


Szkolenia INSPIRE
dla administracji
Terminologia
Model pojęciowy (ang. conceptual model)
– model definiujący pojęcia z pewnej przestrzeni rozważań (przedmiotu zainteresowań).
Schemat pojęciowy (ang. conceptual schema)
– formalny opis modelu pojęciowego w określonym języku schematu pojęciowego.
Schemat aplikacyjny (ang. application schema)
– schemat pojęciowy dla danych wykorzystywanych przez jedną lub więcej aplikacji, dla określonego zakresu przedmiotowego.

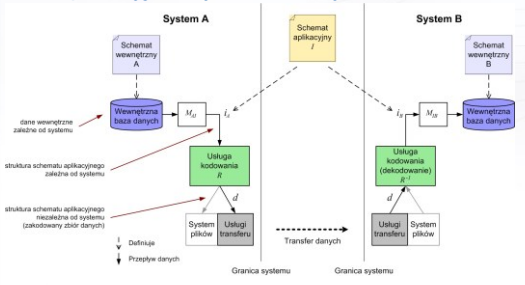
Szkolenia INSPIRE dla administracji

Schemat aplikacyjny

- Podstawa pomyślnej wymiany danych:
  - definiuje możliwą zawartość oraz strukturę danych.
- Do wymiany danych:
  - zapisany w języku schematu pojęciowego UML, zgodnie z normami ISO/TS 19103 i ISO 19109 (reguły budowy schematów aplikacyjnych).
- Dostępny dla obu uczestników procesu wymiany danych (nadawcy i odbiorcy):
  - aby zapewnić pomyślny przebieg wymiany danych.

Szkolenia INSPIRE dla administracji

Interoperacyjna wymiana danych

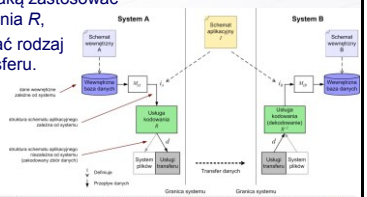


Ogólna idea wymiany danych między dwoma systemami
Źródło: prCEN/TR 15449, 2006; ISO/DIS 19118:2005

Szkolenia INSPIRE dla administracji

Interoperacyjna wymiana danych

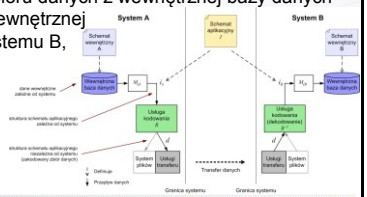
- System A chce przesłać zbiór danych do systemu B. W tym celu A i B muszą:
  - uzgodnić wspólny schemat aplikacyjny *I*,
  - zdecydować, jaką zastosować regułę kodowania *R*,
  - jaki wykorzystać rodzaj protokołu transferu.



Szkolenia INSPIRE dla administracji

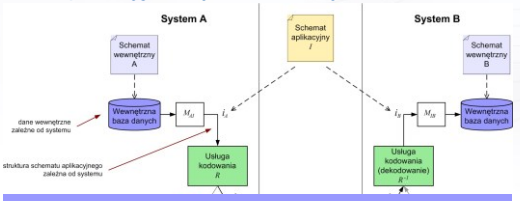
Interoperacyjna wymiana danych

- Oba systemy, A i B, przechowują dane w wewnętrznych bazach danych zgodnie ze swoimi wewnętrznymi schematami, ale zazwyczaj te schematy są różne.
- Przeniesienie zbioru danych z wewnętrznej bazy danych systemu A do wewnętrznej bazy danych systemu B, musi przebiegać w następujący sposób:



Szkolenia INSPIRE dla administracji

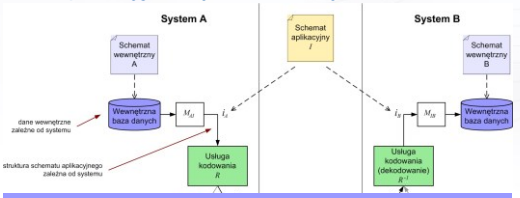
Interoperacyjna wymiana danych



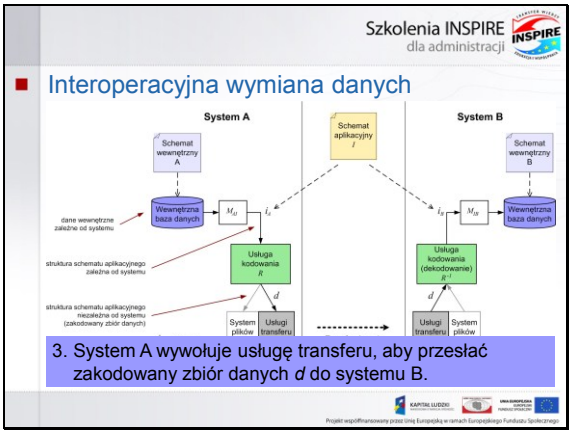
1. System A tłumaczy swoje wewnętrzne dane na strukturę danych zgodną ze wspólnym schematem aplikacyjnym *I* – powstaje schemat aplikacyjny określonej struktury danych *iA*.

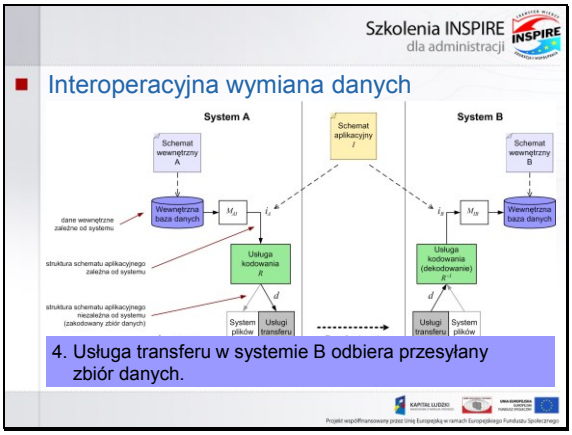
Szkolenia INSPIRE dla administracji

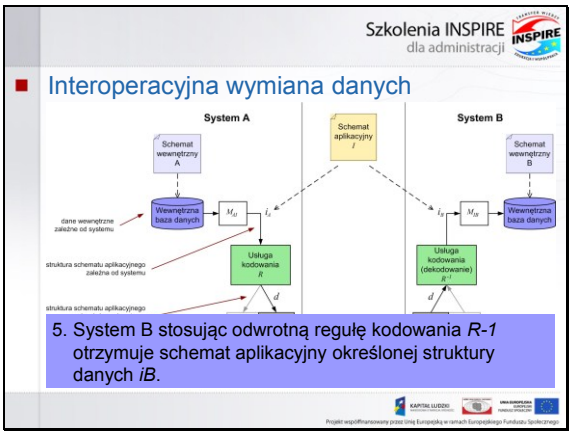
Interoperacyjna wymiana danych



2. Zostaje wykorzystana usługa kodowania, która stosuje regułę kodowania *R* do stworzenia struktury danych *d* niezależnej od systemu, a więc odpowiedniej do transferu.

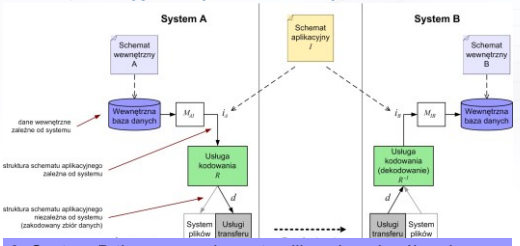

Szkolenia INSPIRE
dla administracji
Interoperacyjna wymiana danych
System A
System B
Schemat wewnętrzny A
Schemat aplikacyjny
Schemat wewnętrzny B
Wewnętrzna baza danych
Usługa kodowania
Usługa kodowania (dekodowanie)
System plików
Usługi transferu
3. System A wywołuje usługę transferu, aby przesłać zakodowany zbiór danych *d* do systemu B.


Szkolenia INSPIRE
dla administracji
Interoperacyjna wymiana danych
System A
System B
Schemat wewnętrzny A
Schemat aplikacyjny
Schemat wewnętrzny B
Wewnętrzna baza danych
Usługa kodowania
Usługa kodowania (dekodowanie)
System plików
Usługi transferu
4. Usługa transferu w systemie B odbiera przesyłany zbiór danych.


Szkolenia INSPIRE
dla administracji
Interoperacyjna wymiana danych
System A
System B
Schemat wewnętrzny A
Schemat aplikacyjny
Schemat wewnętrzny B
Wewnętrzna baza danych
Usługa kodowania
Usługa kodowania (dekodowanie)
5. System B stosując odwrotną regułę kodowania *R-1* otrzymuje schemat aplikacyjny określonej struktury danych *iB*.

Szkolenia INSPIRE dla administracji

Interoperacyjna wymiana danych



6. System B tłumaczy schemat aplikacyjny określonej struktury danych *iB* na swoją wewnętrzną bazę danych.

Szkolenia INSPIRE dla administracji

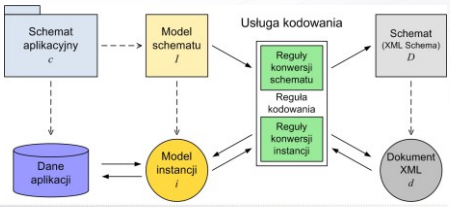
Reguły kodowania

- Normy ISO serii 19100 określają dwie reguły kodowania oparte na języku **XML** (ang. *Extensible Markup Language*):
  - **ISO 19136 (GML), Załącznik E**
    - reguła kodowania oparta na XML dla schematów aplikacyjnych zgodnych z normą ISO 19109, które mogą być przedstawione za pomocą ograniczonego profilu UML, pozwalającego na konwersję do języka XML Schema,
  - **ISO/TS 19139**
    - reguła kodowania oparta na XML dla schematów pojęciowych określających typy, które opisują zasoby geograficzne, np. metadane według ISO 19115 i katalogi obiektów według ISO 19110.

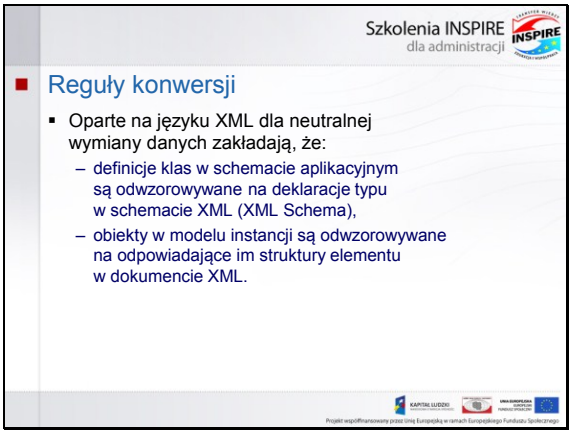
Szkolenia INSPIRE dla administracji

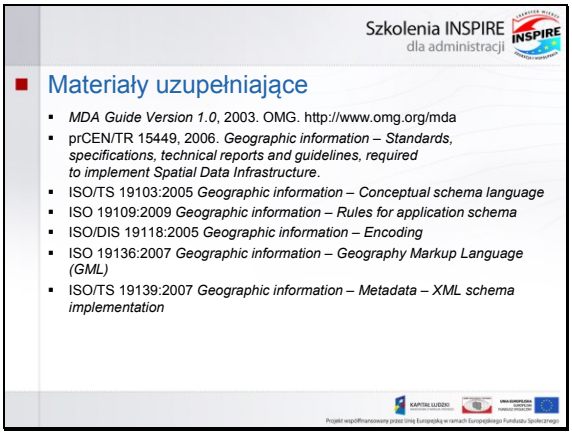
Reguła kodowania

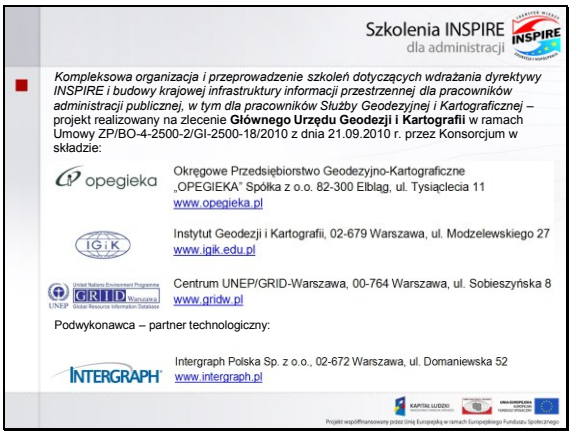
- Powinna określać:
  - reguły konwersji schematu,
  - reguły konwersji instancji.



Reguła kodowania oparta na XML
Źródło: ISO/DIS 19118:2005


Szkolenia INSPIRE
dla administracji
Reguły konwersji
Oparte na języku XML dla neutralnej wymiany danych zakładają, że:
– definicje klas w schemacie aplikacyjnym są odwzorowywane na deklaracje typu w schemacie XML (XML Schema),
– obiekty w modelu instancji są odwzorowywane na odpowiadające im struktury elementu w dokumencie XML.

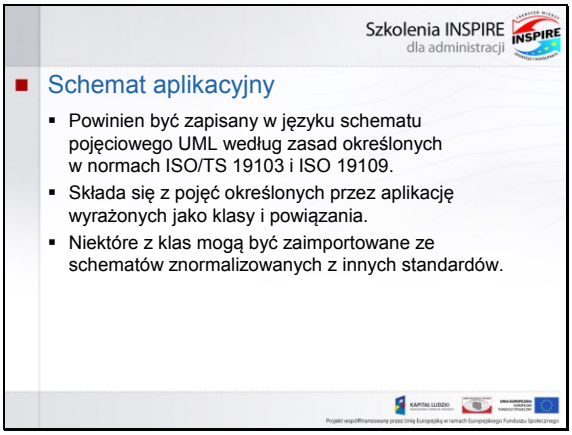

Szkolenia INSPIRE
dla administracji
Materiały uzupełniające
MDA Guide Version 1.0, 2003. OMG. http://www.omg.org/mda
prCEN/TR 15449, 2006. Geographic information – Standards, specifications, technical reports and guidelines, required to implement Spatial Data Infrastructure.
ISO/TS 19103:2005 Geographic information – Conceptual schema language
ISO 19109:2009 Geographic information – Rules for application schema
ISO/DIS 19118:2005 Geographic information – Encoding
ISO 19136:2007 Geographic information – Geography Markup Language (GML)
ISO/TS 19139:2007 Geographic information – Metadata – XML schema implementation

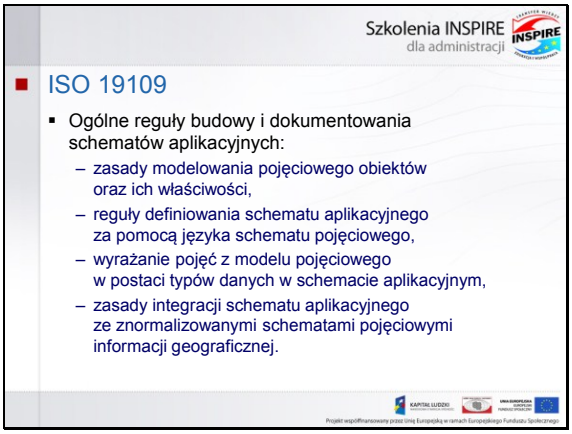

Szkolenia INSPIRE
dla administracji
Kompleksowa organizacja i przeprowadzenie szkoleń dotyczących wdrażania dyrektywy INSPIRE i budowy krajowej infrastruktury informacji przestrzennej dla pracowników administracji publicznej, w tym dla pracowników Służby Geodezyjnej i Kartograficznej – projekt realizowany na zlecenie Głównego Urzędu Geodezji i Kartografii w ramach Umowy ZP/BO-4-2500-2/GI-2500-18/2010 z dnia 21.09.2010 r. przez Konsorcjum w składzie:
opegieka
Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne „OPEGIEKA" Spółka z o.o. 82-300 Elbląg, ul. Tysiąclecia 11
www.opegieka.pl
IGiK
Instytut Geodezji i Kartografii, 02-679 Warszawa, ul. Modzelewskiego 27
www.igik.edu.pl
UNEP GRID Warszawa
Centrum UNEP/GRID-Warszawa, 00-764 Warszawa, ul. Sobieszyńska 8
www.gridw.pl
Podwykonawca – partner technologiczny:
INTERGRAPH
Intergraph Polska Sp. z o.o., 02-672 Warszawa, ul. Domaniewska 52
www.intergraph.pl

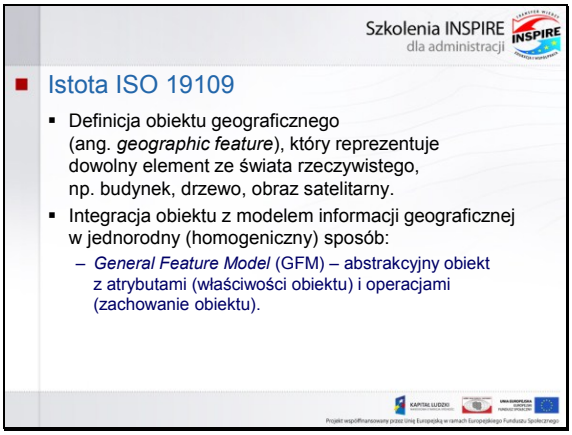

Szkolenia INSPIRE
dla administracji
Szkolenie eksperckie
Środki formalne
modelowania informacji geograficznej
– UML, XML, GML
wykładowca:
IMIĘ NAZWISKO
Zajęcia przygotowane na podstawie autorskiego programu opracowanego przez:
dr inż. Agnieszkę Chojkę z Uniwersytetu Warmińsko-Mazurskiego w Olsztynie

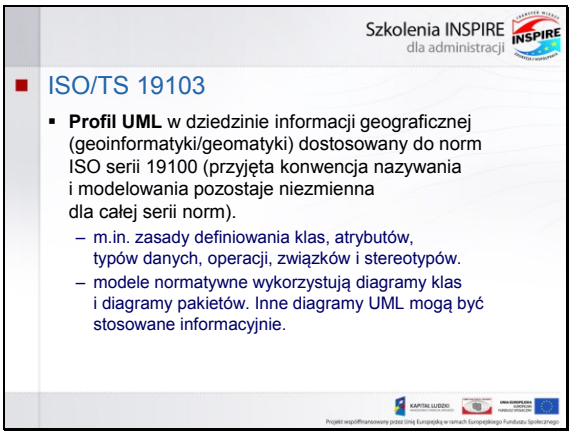
2. UML


Szkolenia INSPIRE
dla administracji
Środki formalne modelowania informacji geograficznej – UML, XML, GML
U M L
Reguły budowy schematów
aplikacyjnych w UML


Szkolenia INSPIRE
dla administracji
Schemat aplikacyjny
Powinien być zapisany w języku schematu pojęciowego UML według zasad określonych w normach ISO/TS 19103 i ISO 19109.
Składa się z pojęć określonych przez aplikację wyrażonych jako klasy i powiązania.
Niektóre z klas mogą być zaimportowane ze schematów znormalizowanych z innych standardów.


Szkolenia INSPIRE
dla administracji
ISO 19109
Ogólne reguły budowy i dokumentowania schematów aplikacyjnych:
– zasady modelowania pojęciowego obiektów oraz ich właściwości,
– reguły definiowania schematu aplikacyjnego za pomocą języka schematu pojęciowego,
– wyrażanie pojęć z modelu pojęciowego w postaci typów danych w schemacie aplikacyjnym,
– zasady integracji schematu aplikacyjnego ze znormalizowanymi schematami pojęciowymi informacji geograficznej.


Szkolenia INSPIRE
dla administracji
Istota ISO 19109
Definicja obiektu geograficznego (ang. geographic feature), który reprezentuje dowolny element ze świata rzeczywistego, np. budynek, drzewo, obraz satelitarny.
Integracja obiektu z modelem informacji geograficznej w jednorodny (homogeniczny) sposób:
– General Feature Model (GFM) – abstrakcyjny obiekt z atrybutami (właściwości obiektu) i operacjami (zachowanie obiektu).


Szkolenia INSPIRE
dla administracji
ISO/TS 19103
Profil UML w dziedzinie informacji geograficznej (geoinformatyki/geomatyki) dostosowany do norm ISO serii 19100 (przyjęta konwencja nazywania i modelowania pozostaje niezmienna dla całej serii norm).
– m.in. zasady definiowania klas, atrybutów, typów danych, operacji, związków i stereotypów.
– modele normatywne wykorzystują diagramy klas i diagramy pakietów. Inne diagramy UML mogą być stosowane informacyjnie.

Szkolenia INSPIRE dla administracji

UML

- ang. *Unified Modeling Language*
- pol. *zunifikowany (ujednolicony) język modelowania*

- Język formalny służący do opisu świata obiektów w analizie obiektowej oraz programowaniu obiektowym.

Szkolenia INSPIRE dla administracji

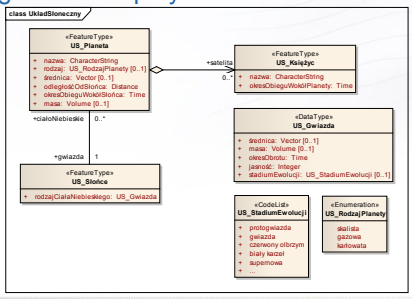
Diagram klas w UML

- Opis:
  - zbioru danych,
  - zależności pomiędzy nimi.

- Podstawowe elementy:
  - oznaczenia klas,
  - związków pomiędzy nimi,
  - oznaczenia pomocnicze (np. stereotypy, metki, ograniczenia itp.).

Szkolenia INSPIRE dla administracji

Diagram klas – przykład



Przykład diagramu klas w UML

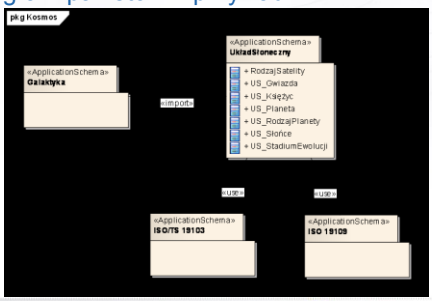
Szkolenia INSPIRE dla administracji

Diagram pakietów w UML

- Zawiera m.in. klasy i diagramy klas.
- Porządkuje strukturę zależności w systemie, w który można wyróżnić bardzo wiele klas, przypadków użycia itp.
- Przedstawia podział systemu z logicznego punktu widzenia.

Szkolenia INSPIRE dla administracji

Diagram pakietów – przykład

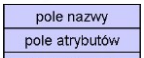


Przykład diagramu klas w UML

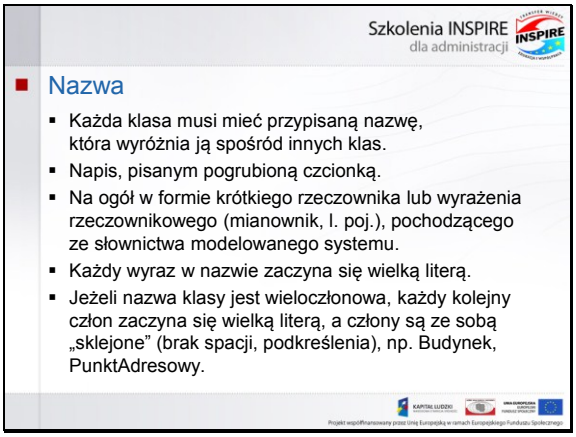
Szkolenia INSPIRE dla administracji

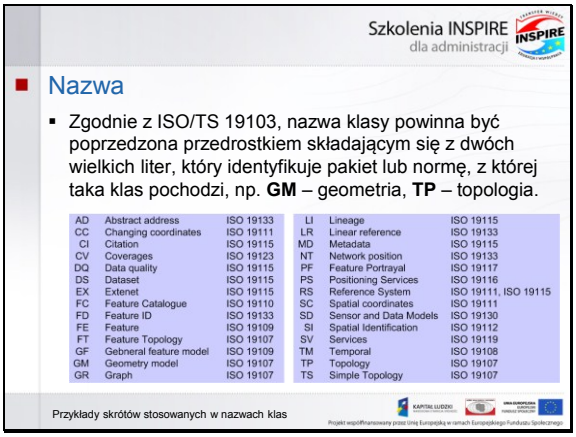
Klasa

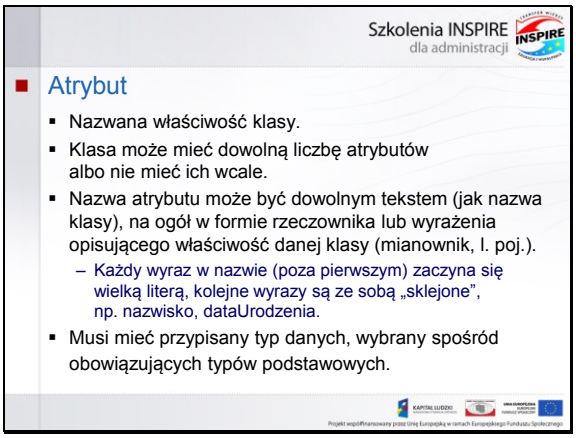
- Opis zbioru obiektów, które mają takie same atrybuty, operacje, związki i znaczenie (wzorzec dla tworzonego obiektu).
- Według ISO/TS 19103 każda klasa powinna być zdefiniowana wewnątrz określonego pakietu.
- **Symbol graficzny:**
  - Prostokąt podzielony na trzy pola:
    - **pole nazwy** – nazwa klasy, stereotyp, lista wartości etykietowanych,
    - **pole atrybutów** – lista atrybutów klasy,
    - **pole operacji** – lista operacji (metod) zdefiniowanych dla klasy.

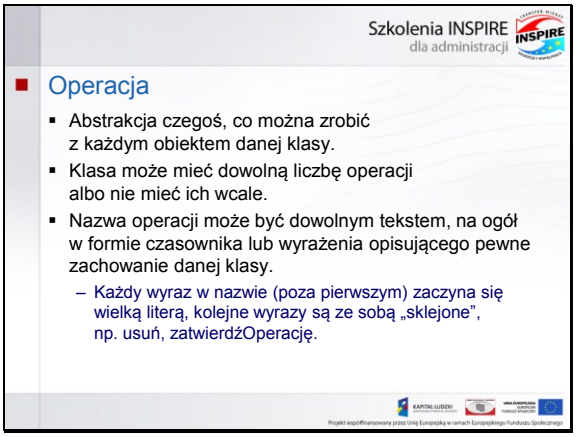


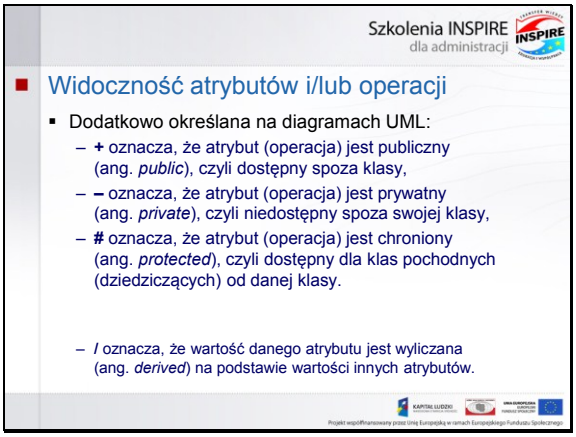
Klasa w UML

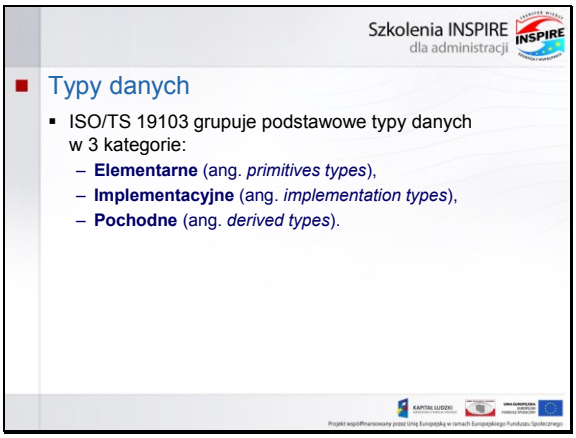

Szkolenia INSPIRE
dla administracji
Nazwa
Każda klasa musi mieć przypisaną nazwę, która wyróżnia ją spośród innych klas.
Napis, pisanym pogrubioną czcionką.
Na ogół w formie krótkiego rzeczownika lub wyrażenia rzeczownikowego (mianownik, l. poj.), pochodzącego ze słownictwa modelowanego systemu.
Każdy wyraz w nazwie zaczyna się wielką literą.
Jeżeli nazwa klasy jest wieloczłonowa, każdy kolejny człon zaczyna się wielką literą, a człony są ze sobą „sklejone" (brak spacji, podkreślenia), np. Budynek, PunktAdresowy.

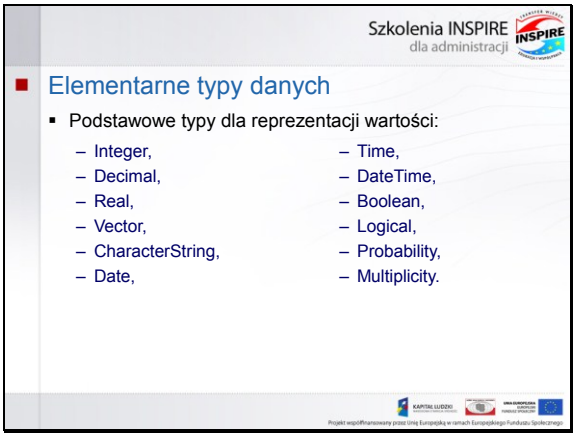

Szkolenia INSPIRE
dla administracji
Nazwa
Zgodnie z ISO/TS 19103, nazwa klasy powinna być poprzedzona przedrostkiem składającym się z dwóch wielkich liter, który identyfikuje pakiet lub normę, z której taka klas pochodzi, np. GM – geometria, TP – topologia.
AD Abstract address ISO 19133
CC Changing coordinates ISO 19111
CI Citation ISO 19115
CV Coverages ISO 19123
DQ Data quality ISO 19115
DS Dataset ISO 19115
EX Extenet ISO 19115
FC Feature Catalogue ISO 19110
FD Feature ID ISO 19133
FE Feature ISO 19109
FT Feature Topology ISO 19107
GF Gebneral feature model ISO 19109
GM Geometry model ISO 19107
GR Graph ISO 19107
LI Lineage ISO 19115
LR Linear reference ISO 19133
MD Metadata ISO 19115
NT Network position ISO 19133
PF Feature Portrayal ISO 19117
PS Positioning Services ISO 19116
RS Reference System ISO 19111, ISO 19115
SC Spatial coordinates ISO 19111
SD Sensor and Data Models ISO 19130
SI Spatial Identification ISO 19112
SV Services ISO 19119
TM Temporal ISO 19108
TP Topology ISO 19107
TS Simple Topology ISO 19107
Przykłady skrótów stosowanych w nazwach klas

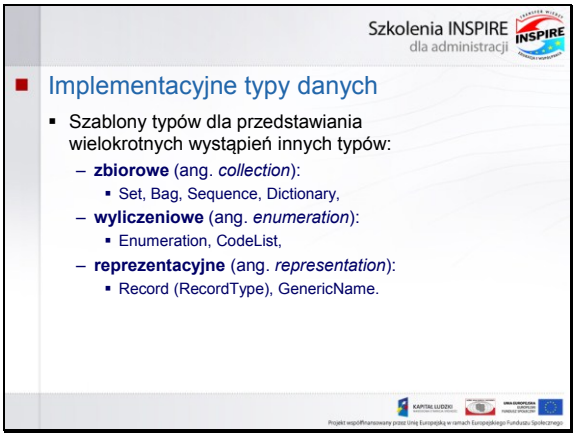

Szkolenia INSPIRE
dla administracji
Atrybut
Nazwana właściwość klasy.
Klasa może mieć dowolną liczbę atrybutów albo nie mieć ich wcale.
Nazwa atrybutu może być dowolnym tekstem (jak nazwa klasy), na ogół w formie rzeczownika lub wyrażenia opisującego właściwość danej klasy (mianownik, l. poj.).
– Każdy wyraz w nazwie (poza pierwszym) zaczyna się wielką literą, kolejne wyrazy są ze sobą „sklejone", np. nazwisko, dataUrodzenia.
Musi mieć przypisany typ danych, wybrany spośród obowiązujących typów podstawowych.

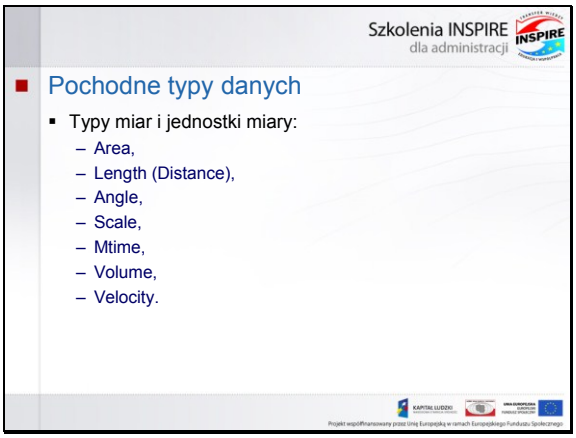

Szkolenia INSPIRE
dla administracji
Operacja
Abstrakcja czegoś, co można zrobić z każdym obiektem danej klasy.
Klasa może mieć dowolną liczbę operacji albo nie mieć ich wcale.
Nazwa operacji może być dowolnym tekstem, na ogół w formie czasownika lub wyrażenia opisującego pewne zachowanie danej klasy.
Każdy wyraz w nazwie (poza pierwszym) zaczyna się wielką literą, kolejne wyrazy są ze sobą „sklejone”, np. usuń, zatwierdźOperację.


Szkolenia INSPIRE
dla administracji
Widoczność atrybutów i/lub operacji
Dodatkowo określana na diagramach UML:
+ oznacza, że atrybut (operacja) jest publiczny (ang. public), czyli dostępny spoza klasy,
– oznacza, że atrybut (operacja) jest prywatny (ang. private), czyli niedostępny spoza swojej klasy,
oznacza, że atrybut (operacja) jest chroniony (ang. protected), czyli dostępny dla klas pochodnych (dziedziczących) od danej klasy.
/ oznacza, że wartość danego atrybutu jest wyliczana (ang. derived) na podstawie wartości innych atrybutów.


Szkolenia INSPIRE
dla administracji
Typy danych
ISO/TS 19103 grupuje podstawowe typy danych w 3 kategorie:
Elementarne (ang. primitives types),
Implementacyjne (ang. implementation types),
Pochodne (ang. derived types).


Szkolenia INSPIRE
dla administracji
Elementarne typy danych
Podstawowe typy dla reprezentacji wartości:
Integer,
Decimal,
Real,
Vector,
CharacterString,
Date,
Time,
DateTime,
Boolean,
Logical,
Probability,
Multiplicity.


Szkolenia INSPIRE
dla administracji
Implementacyjne typy danych
Szablony typów dla przedstawiania wielokrotnych wystąpień innych typów:
zbiorowe (ang. collection):
Set, Bag, Sequence, Dictionary,
wyliczeniowe (ang. enumeration):
Enumeration, CodeList,
reprezentacyjne (ang. representation):
Record (RecordType), GenericName.


Szkolenia INSPIRE
dla administracji
Pochodne typy danych
Typy miar i jednostki miary:
Area,
Length (Distance),
Angle,
Scale,
Mtime,
Volume,
Velocity.

Szkolenia INSPIRE dla administracji

Związki

- **Zależność** (ang. *dependency*)
- **Powiązanie** (ang. *association*)
- **Uogólnienie** (ang. *generalization*)

Szkolenia INSPIRE dla administracji

Zależność

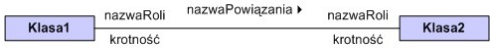
- Związek użycia między dwoma elementami.
- Zmiany dokonane w specyfikacji jednego elementu (np. klasy *Zamówienie*) mogą mieć wpływ na inny element, który używa (korzysta) tego pierwszego (np. na klasę *ZamówienieSpecjalne*), ale niekoniecznie na odwrót.
- **Graficznie** – linia przerywana z grotem skierowanym na element, od którego coś zależy:

Przykład zależności w UML

Szkolenia INSPIRE dla administracji

Powiązanie (asocjacja)

- Związek strukturalny równorzędnych partnerów, tzn. żadna klasa nie jest ważniejsza.
- Wskazuje, iż obiekty jednego elementu są połączone z obiektami innego.
- Dla każdej pary klas/obiektów można zdefiniować dowolnie wiele powiązań, z których każde reprezentuje inną zależność.
- **Graficznie** – linia ciągła łącząca klasy:



Powiązanie w UML

Szkolenia INSPIRE dla administracji

Powiązanie

- Można opisać za pomocą:
  - **nazwy**,
  - **ról**,
  - **krotności**.

Szkolenia INSPIRE dla administracji

Nazwa powiązania

- Powiązanie może mieć przypisaną nazwę, która określa istotę danego związku.
- Aby uniknąć niejednoznaczności, można podać kierunek odczytu (trójkątny znacznik przy nazwie).
- Nie jest konieczna, gdy określone są role.
- Zwykle jest to czasownik w czasie teraźniejszym, w 3os. l. poj.

Przykład nazwy powiązania

Szkolenia INSPIRE dla administracji

Rola

- Klasa biorąca udział w powiązaniu ogrywa w nim określoną rolę.
- „Oblicze", które klasa przy jednym końcu powiązania prezentuje klasie przy drugim końcu.
- Dana klasa może odgrywać tę samą albo inną rolę w różnych powiązaniach.

Przykład ról i krotności w powiązaniu

Szkolenia INSPIRE dla administracji

Krotność (liczność)

- Oznacza ile obiektów może być połączonych przez jeden egzemplarz powiązania.
- Przykłady możliwych krotność dla powiązania w UML:

Krotność	Znaczenie	Np.
–	1 lub nie określono krotności	–
N	dokładnie N	5
*	dowolnie wiele	–
m..n	od m do n	1..5
m..*	od m do dowolnie wiele	2..*

Przykłady możliwych krotności w powiązaniu

Szkolenia INSPIRE dla administracji

Powiązania w UML

- **Nawigacja** (ang. *navigation*)
- **Agregacja zwykła** (ang. *aggregation*)
- **Agregacja całkowita, kompozycja** (ang. *composition*)

Szkolenia INSPIRE dla administracji

Nawigacja

- Powiązanie jednokierunkowe, tzn. tylko jedna z klas biorących udział w powiązaniu „wie” o drugiej klasie.
- **Graficznie** – strzałka dołączona do zakończenia linii powiązania i wskazująca na daną klasę. Brak strzałki oznacza, że powiązaniem można „poruszać się” w obu kierunkach (obie klasy wiedzą o sobie).

Nawigacja w UML

Szkolenia INSPIRE dla administracji

Agregacja zwykła

- Związek „**posiadania**” między klasami, klasa-całość składa się z zestawu innych klas, które stanowią jej składniki (klasy-części); całość „ma/posiada” części.
- Klasy-części mogą być składnikami innych agregacji, zaś ich czas życia nie jest ograniczony do czasu życia klasy-całości.
- **Graficznie** – pusty romb umieszczonego na końcu linii powiązania od strony klasy-całości:

Agregacja zwykła w UML

Szkolenia INSPIRE dla administracji

Agregacja całkowita (kompozycja)

- Związek „**zawierania**” między klasami, gdzie klasa-całość zawiera inne klasy, które stanowią jej składniki (klasy-części); całość „zawiera” części.
- Klasy-części nie mogą być składnikami innych klas-całości, zaś ich czas życia jest ograniczony do czasu życia klasy-całości.
- **Graficznie** – wypełniony romb umieszczony na końcu linii powiązania od strony klasy-całości:

Agregacja całkowita w UML

Szkolenia INSPIRE dla administracji

Uogólnienie (dziedziczenie)

- Związek między elementem ogólnym (generalizacją, nadklasą, przodkiem) a jego pewnym specyficznym rodzajem (specjalizacją, podklasą, potomkiem).
- Potomek może wystąpić wszędzie tam, gdzie jest spodziewany przodek, ale nie na odwrót (potomek może zawsze zastąpić przodka).
- Potomek dziedziczy wszystkie właściwości przodka:
  - atrybuty,
  - operacje,
  - powiązania,
  - ograniczenia.
- Najczęściej potomek ma jeszcze własne cechy, poza odziedziczonymi.

Szkolenia INSPIRE dla administracji

Uogólnienie (dziedziczenie)

- Klasa może nie mieć przodka, a może mieć jednego lub więcej.
- Może mieć nazwę, ale zwykle nie jest ona potrzebna.
- **Graficznie** – linia ciągła zakończona niewypełnionym trójkątem wskazującym przodka:

Dziedziczenie w UML

Szkolenia INSPIRE dla administracji

Mechanizmy rozszerzenia w UML

- Pozwalają na rozszerzenie semantyki (słownictwa) UML:
  - umożliwiają dostosowanie UML do potrzeb konkretnego zadania i pozwalają na przystosowanie go do nowych technologii.
  - pozwalają na uszczegółowienie modelu, zdefiniowanie nowych elementów, przystosowanie notacji do specyficznej dziedziny lub preferencji użytkownika.
- Są to:
  - **stereotypy** (ang. *stereotypes*),
  - **metki** (ang. *tagged values*),
  - **ograniczenia** (ang. *constraints)*.

Szkolenia INSPIRE dla administracji

Stereotyp

- Wyrażenie umożliwiające klasyfikację elementów modelu.
- Wspólna, nazwana własność obiektów, klas, powiązań, atrybutów, ograniczeń.
- Dla każdego rodzaju elementu UML istnieje lista dozwolonych stereotypów.
- ISO/TS 19103 definiuje 11 stereotypów dla informacji geograficznej (9 z nich to standardowe stereotypy UML), np.:
  - **<<Enumeration>>**,
  - **<<CodeList>>**,
  - **<<DataType>>**.

Szkolenia INSPIRE dla administracji

Metka (wartość etykietowana)

- Pozwala dołączyć do elementu dodatkowe właściwości.
- Para „*klucz = wartość*”.
- Listę wartości etykietowanych oddzielonych przecinkami umieszcza się w nawiasach klamrowych {}:

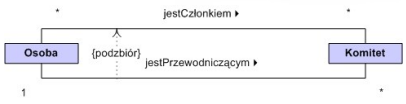
{ autor = ‘Marcin Woicki’, dataWydania = ‘2000’ }

Przykład metki w UML

Szkolenia INSPIRE dla administracji

Ograniczenie

- Określa restrykcje nakładane na elementy modelu.
- Może stanowić wyrażenia języka naturalnego, języka formalnego (np. OCL w UML), może przyjmować postać formuły matematycznej lub fragmentu kodu.
- OCL (ang. *Object Constraint Language*) – formalny język zapisu ograniczeń w UML.

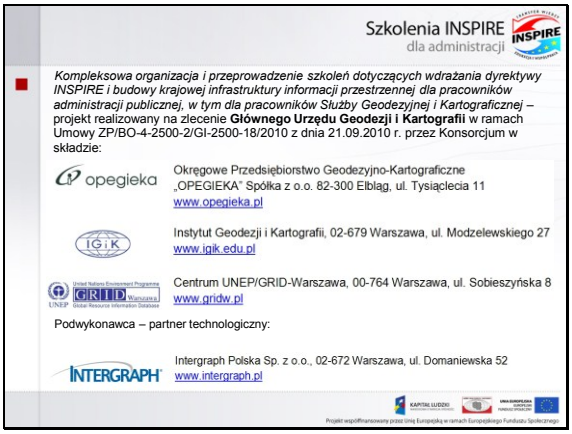


Przykład ograniczenia w UML

Szkolenia INSPIRE dla administracji

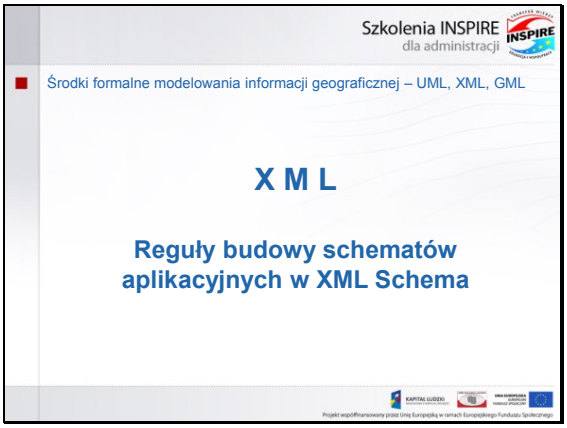
Materiały uzupełniające

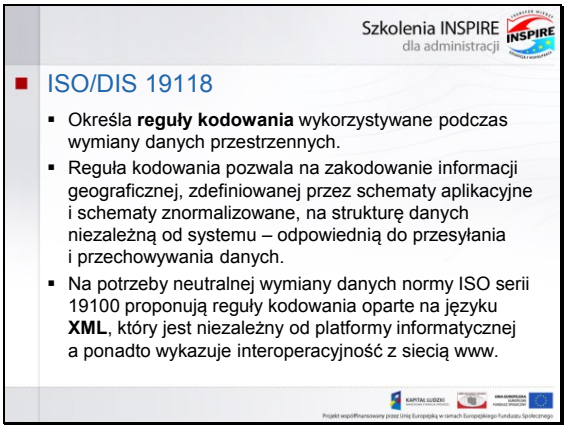
- *Booch G., Rumbaugh J., Jacobson I. 2002. UML. Przewodnik użytkownika. WNT, Warszawa.*
- *Fowler M., Scott K. 2002. UML w kropelce. Oficyna wydawnicza LTP, Warszawa.*
- *Schmuller J. 2003. UML dla każdego. Helion, Gliwice.*
- *Subieta K. 1999. Wprowadzenie do obiektowych metodyk projektowania i notacji UML. Jedenasta Górska Szkoła PTI Szczyrk.*
- *Object Management Group (OMG), UML. http://www.uml.org*
- *Portal UML. Projekty systemów informatycznych z UML. http://www.uml.com.pl*
- *ISO/TS 19103:2005 Geographic information – Conceptual schema language*
- *ISO 19109:2009 Geographic information – Rules for application schema*


Szkolenia INSPIRE
dla administracji
Kompleksowa organizacja i przeprowadzenie szkoleń dotyczących wdrażania dyrektywy INSPIRE i budowy krajowej infrastruktury informacji przestrzennej dla pracowników administracji publicznej, w tym dla pracowników Służby Geodezyjnej i Kartograficznej – projekt realizowany na zlecenie Głównego Urzędu Geodezji i Kartografii w ramach Umowy ZP/BO-4-2500-2/GI-2500-18/2010 z dnia 21.09.2010 r. przez Konsorcjum w składzie:
opegieka
Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne „OPEGIEKA" Spółka z o.o. 82-300 Elbląg, ul. Tysiąclecia 11
www.opegieka.pl
IGiK
Instytut Geodezji i Kartografii, 02-679 Warszawa, ul. Modzelewskiego 27
www.igik.edu.pl
GRID Warszawa
Centrum UNEP/GRID-Warszawa, 00-764 Warszawa, ul. Sobieszyńska 8
www.gridw.pl
Podwykonawca – partner technologiczny:
INTERGRAPH
Intergraph Polska Sp. z o.o., 02-672 Warszawa, ul. Domaniewska 52
www.intergraph.pl


Szkolenia INSPIRE
dla administracji
Szkolenie eksperckie
Środki formalne
modelowania informacji geograficznej
– UML, XML, GML
wykładowca:
IMIĘ NAZWISKO
Zajęcia przygotowane na podstawie autorskiego programu opracowanego przez:
dr inż. Agnieszkę Chojkę z Uniwersytetu Warmińsko-Mazurskiego w Olsztynie

3. XML

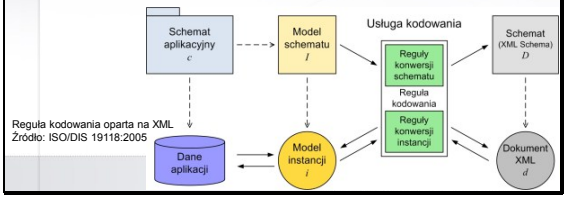

Szkolenia INSPIRE
dla administracji
Środki formalne modelowania informacji geograficznej – UML, XML, GML
X M L
Reguły budowy schematów
aplikacyjnych w XML Schema


Szkolenia INSPIRE
dla administracji
ISO/DIS 19118
Określa reguły kodowania wykorzystywane podczas wymiany danych przestrzennych.
Reguła kodowania pozwala na zakodowanie informacji geograficznej, zdefiniowanej przez schematy aplikacyjne i schematy znormalizowane, na strukturę danych niezależną od systemu – odpowiednią do przesyłania i przechowywania danych.
Na potrzeby neutralnej wymiany danych normy ISO serii 19100 proponują reguły kodowania oparte na języku XML, który jest niezależny od platformy informatycznej a ponadto wykazuje interoperacyjność z siecią www.

Szkolenia INSPIRE dla administracji

Reguła kodowania oparta na XML

- reguły konwersji **schematu aplikacyjnego UML** na **schemat struktur danych zapisany w XML Schema**,
- reguły konwersji **instancji** na **struktury danych zapisane w dokumencie XML**.



Reguła kodowania oparta na XML
Źródło: ISO/DIS 19118:2005

Szkolenia INSPIRE dla administracji

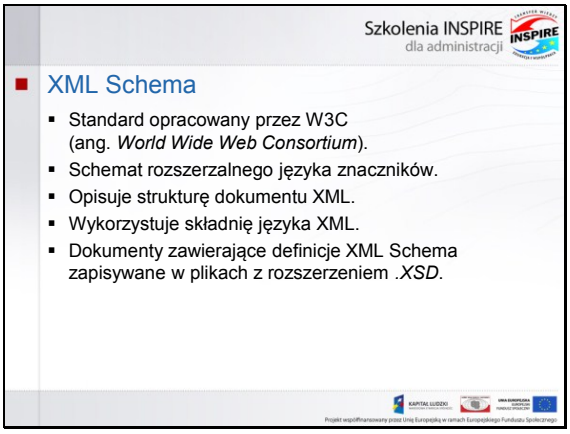
Schemat XML (XSD)

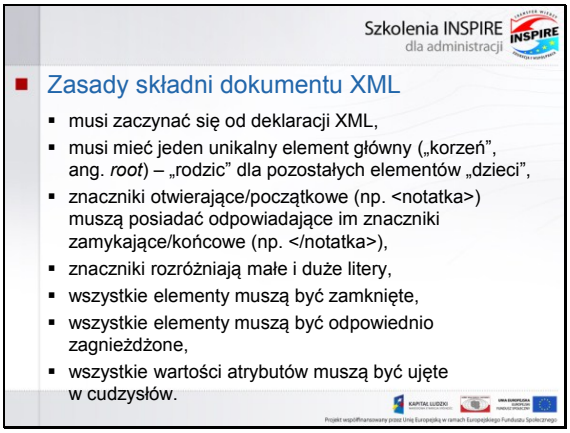
- Reguły konwersji schematu definiują jak utworzyć **dokument schematu XML** (**XSD**, ang. *XML Schema Definition*) zgodnie ze schematem aplikacyjnym wyrażonym w UML.
- Główny cel – zapewnienie, że dokumenty XML wytworzone przy użyciu reguł konwersji danych będą poprawne.
- Powinien zawierać definicje typów, deklaracje atrybutów i elementów, które odpowiadają klasom zdefiniowanym w schemacie aplikacyjnym.
- Fizycznie może być pojedynczym dokumentem schematu lub może być podzielony na kilka oddzielnych.

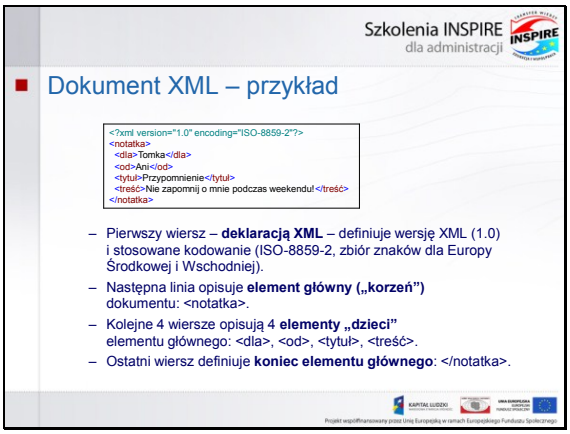
Szkolenia INSPIRE dla administracji

XML

- Standard opracowany przez W3C (ang. *World Wide Web Consortium*).
- Rozszerzalny język znaczników.
- Zaprojektowany do przesyłania i przechowywania danych.
- Język znaczników, podobnie jak język HTML (ang. *HyperText Markup Language*).
- Programowo i sprzętowo niezależne narzędzie przenoszenia informacji (HTML odpowiada jedynie za jej wyświetlanie).


Szkolenia INSPIRE
dla administracji
XML Schema
Standard opracowany przez W3C (ang. World Wide Web Consortium).
Schemat rozszerzalnego języka znaczników.
Opisuje strukturę dokumentu XML.
Wykorzystuje składnię języka XML.
Dokumenty zawierające definicje XML Schema zapisywane w plikach z rozszerzeniem .XSD.


Szkolenia INSPIRE
dla administracji
Zasady składni dokumentu XML
musi zaczynać się od deklaracji XML,
musi mieć jeden unikalny element główny („korzeń”, ang. root) – „rodzic” dla pozostałych elementów „dzieci”,
znaczniki otwierające/początkowe (np. <notatka>) muszą posiadać odpowiadające im znaczniki zamykające/końcowe (np. </notatka>),
znaczniki rozróżniają małe i duże litery,
wszystkie elementy muszą być zamknięte,
wszystkie elementy muszą być odpowiednio zagnieżdżone,
wszystkie wartości atrybutów muszą być ujęte w cudzysłów.


Szkolenia INSPIRE
dla administracji
Dokument XML – przykład
<?xml version="1.0" encoding="ISO-8859-2"?>
<notatka>
<dla>Tomka</dla>
<od>Ani</od>
<tytuł>Przypomnienie</tytuł>
<treść>Nie zapomnij o mnie podczas weekendu!</treść>
</notatka>
– Pierwszy wiersz – deklaracją XML – definiuje wersję XML (1.0) i stosowane kodowanie (ISO-8859-2, zbiór znaków dla Europy Środkowej i Wschodniej).
– Następna linia opisuje element główny („korzeń”) dokumentu: <notatka>.
– Kolejne 4 wiersze opisują 4 elementy „dzieci” elementu głównego: <dla>, <od>, <tytuł>, <treść>.
– Ostatni wiersz definiuje koniec elementu głównego: </notatka>.

Szkolenia INSPIRE dla administracji

XML a HTML – przykład

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<notatka>
  <dla>Tomka</dla>
  <od>Ani</od>
  <tytul>Przypomnienie</tytul>
  <treść>Nie zapomnij o mnie podczas weekendu!</treść>
</notatka>
```

Dokument XML

<?xml version="1.0" encoding="ISO-8859-2" ?>
- <notatka>
  <dla>Tomka</dla>
  <od>Ani</od>
  <tytul>Przypomnienie</tytul>
  <treść>Nie zapomnij o mnie podczas weekendu!</treść>
</notatka>

Widok dokumentu XML w przeglądarce internetowej

```
<html>
  <head>
    <title>Notatka</title>
  </head>
  <body>
    <table align="center" width="350" bgcolor="red">
      <tr><td align="center"><h2><u>PRZYPOMNIENIE</u></h2></td></tr>
      <tr><td align="center"><h3>Nie zapomnij o mnie podczas weekendu!</h3>
      <tr><td align="right"><h3><i>Ania</i></h3></td></tr>
    </table>
  </body>
</html>
```

Dokument HTML

PRZYPOMNIENIE

Nie zapomnij o mnie podczas weekendu!

Ania

Widok dokumentu HTML w przeglądarce internetowej

Szkolenia INSPIRE dla administracji

Elementy i atrybuty

- Dokument XML zawiera elementy, które mogą mieć atrybuty.
  - **Elementem XML** jest wszystko to, co znajduje się między znacznikiem początkowym (łącznie z nim) elementu a znacznikiem końcowym (łącznie z nim) elementu.
  - **Atrybuty** dostarczają dodatkowych informacji o elementach.
- Element może zawierać:
  - inne elementy,
  - tekst,
  - atrybuty,
  - lub wszystkie powyższe.

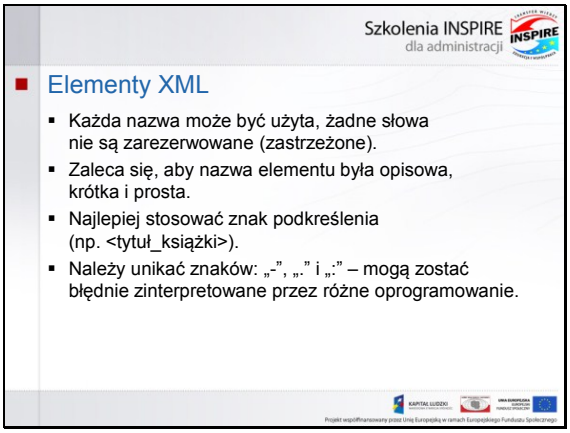
Szkolenia INSPIRE dla administracji

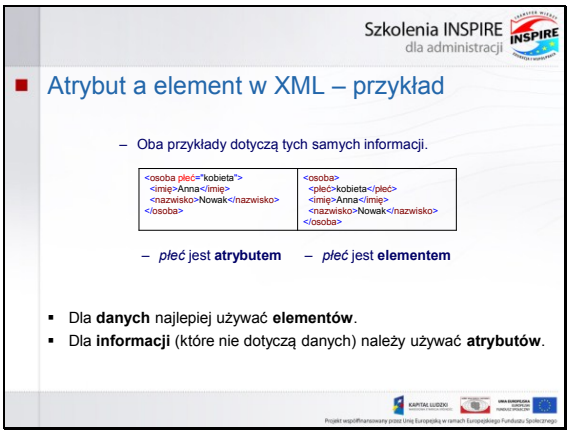
Elementy i atrybuty – przykład

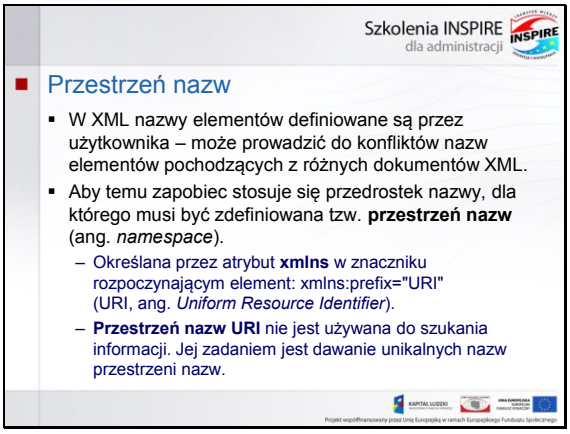
```
<?xml version="1.0" encoding="ISO-8859-2"?>
<księgarnia>
  <książka kategoria="DZIECI">
    <tytul>Harry Potter i czara ognia</tytul>
    <autor>J.K. Rowling</autor>
    <rok>2001</rok>
    <cena>49.99</cena>
  </książka>
  <książka kategoria="KRYMINAŁ">
    <tytul>Byczki w pomidorach</tytul>
    <autor>J. Chmielewska</autor>
    <rok>2010</rok>
    <cena>38.99</cena>
  </książka>
</księgarnia>
```

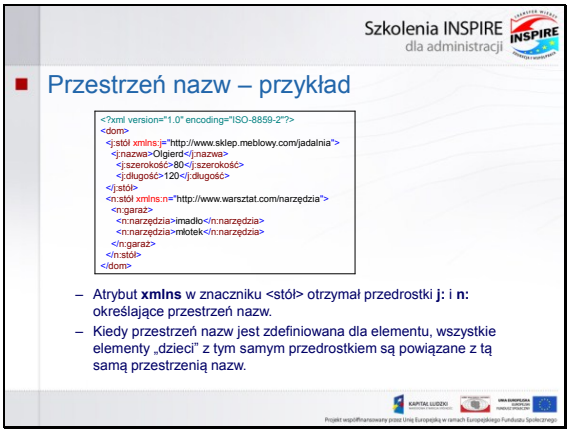
- Dokument XML składa się z **elementu głównego** <księgarnia>, który zawiera **element** <książka>.
- <książka> składa się z **elementów** <tytuł>, <autor>, <rok>, i <cena>, które zawierają tekst.
- **Element** <książka> posiada również **atrybut** kategoria.

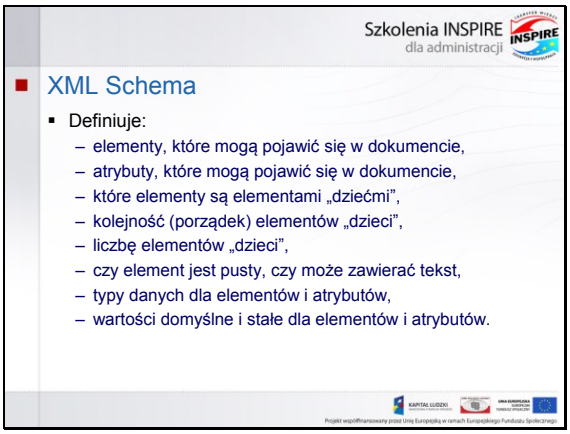

Szkolenia INSPIRE
dla administracji
Elementy XML
Muszą stosować się do poniższych zasad nazywania:
– nazwy mogą zawierać litery, liczby i inne znaki,
– nazwy nie mogą rozpoczynać się od liczby lub znaku przestankowego (np. . , : ; „ ‘ ? -),
– nazwy nie mogą rozpoczynać się od liter xml (lub XML, lub Xml, itp.),
– nazwy nie mogą zawierać spacji.

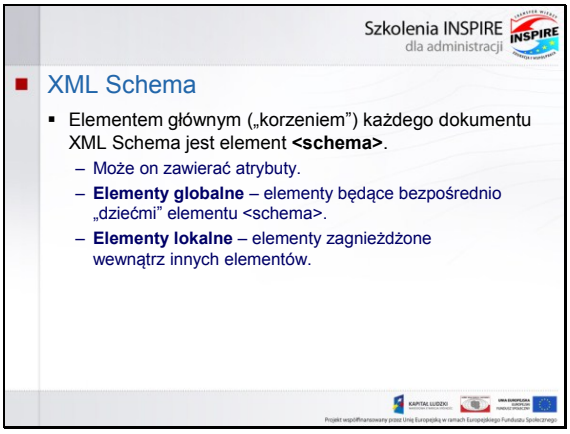

Szkolenia INSPIRE
dla administracji
Elementy XML
Każda nazwa może być użyta, żadne słowa nie są zarezerwowane (zastrzeżone).
Zaleca się, aby nazwa elementu była opisowa, krótka i prosta.
Najlepiej stosować znak podkreślenia (np. <tytuł_książki>).
Należy unikać znaków: „-”, „.” i „:” – mogą zostać błędnie zinterpretowane przez różne oprogramowanie.

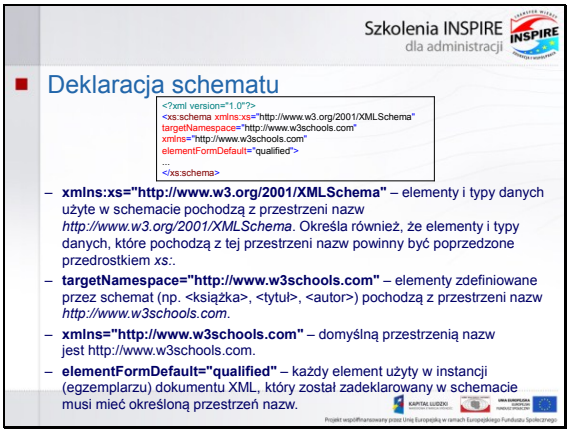

Szkolenia INSPIRE
dla administracji
Atrybut a element w XML – przykład
– Oba przykłady dotyczą tych samych informacji.
<osoba płeć="kobieta">
<imię>Anna</imię>
<nazwisko>Nowak</nazwisko>
</osoba>
<osoba>
<płeć>kobieta</płeć>
<imię>Anna</imię>
<nazwisko>Nowak</nazwisko>
</osoba>
– płeć jest atrybutem
– płeć jest elementem
Dla danych najlepiej używać elementów.
Dla informacji (które nie dotyczą danych) należy używać atrybutów.

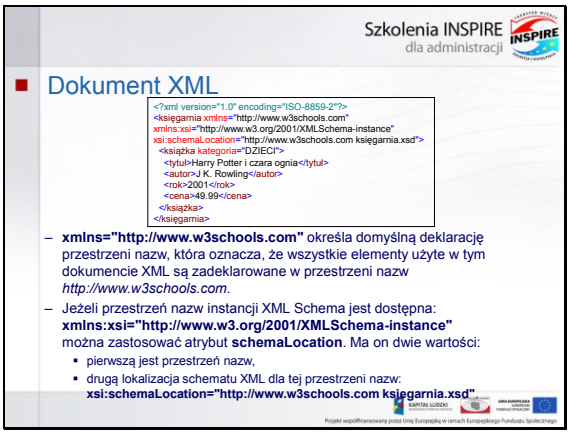

Szkolenia INSPIRE
dla administracji
Przestrzeń nazw
W XML nazwy elementów definiowane są przez użytkownika – może prowadzić do konfliktów nazw elementów pochodzących z różnych dokumentów XML.
Aby temu zapobiec stosuje się przedrostek nazwy, dla którego musi być zdefiniowana tzw. przestrzeń nazw (ang. namespace).
Określana przez atrybut xmlns w znaczniku rozpoczynającym element: xmlns:prefix="URI" (URI, ang. Uniform Resource Identifier).
Przestrzeń nazw URI nie jest używana do szukania informacji. Jej zadaniem jest dawanie unikalnych nazw przestrzeni nazw.

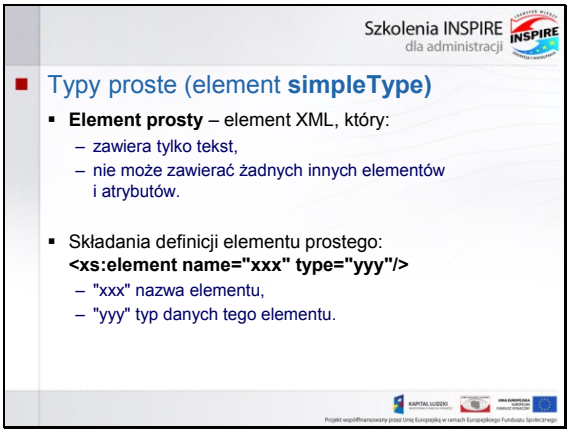

Szkolenia INSPIRE
dla administracji
Przestrzeń nazw – przykład
<?xml version="1.0" encoding="ISO-8859-2"?>
<dom>
<j:stół xmlns:j="http://www.sklep.meblowy.com/jadalnia">
<j:nazwa>Olgierd</j:nazwa>
<j:szerokość>80</j:szerokość>
<j:długość>120</j:długość>
</j:stół>
<n:stół xmlns:n="http://www.warsztat.com/narzędzia">
<n:garaż>
<n:narzędzia>imadło</n:narzędzia>
<n:narzędzia>młotek</n:narzędzia>
</n:garaż>
</n:stół>
</dom>
Atrybut xmlns w znaczniku <stół> otrzymał przedrostki j: i n: określające przestrzeń nazw.
Kiedy przestrzeń nazw jest zdefiniowana dla elementu, wszystkie elementy „dzieci" z tym samym przedrostkiem są powiązane z tą samą przestrzenią nazw.

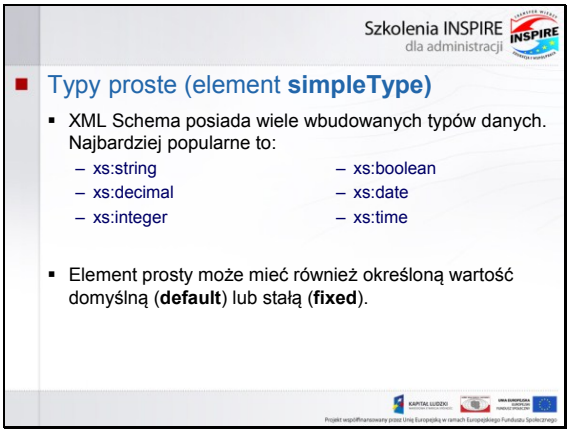

Szkolenia INSPIRE
dla administracji
XML Schema
Definiuje:
elementy, które mogą pojawić się w dokumencie,
atrybuty, które mogą pojawić się w dokumencie,
które elementy są elementami „dziećmi",
kolejność (porządek) elementów „dzieci",
liczbę elementów „dzieci",
czy element jest pusty, czy może zawierać tekst,
typy danych dla elementów i atrybutów,
wartości domyślne i stałe dla elementów i atrybutów.

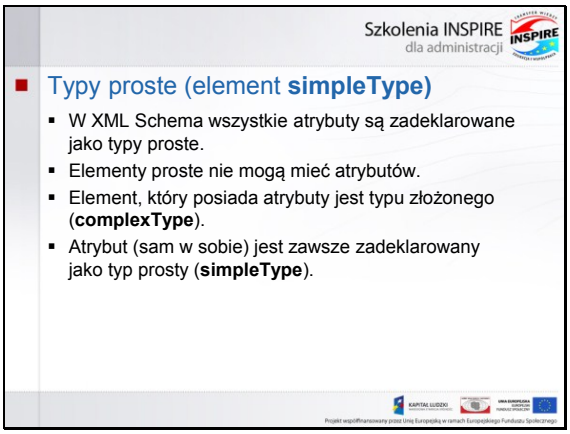

Szkolenia INSPIRE
dla administracji
XML Schema
Elementem głównym („korzeniem") każdego dokumentu XML Schema jest element <schema>.
Może on zawierać atrybuty.
Elementy globalne – elementy będące bezpośrednio „dziećmi" elementu <schema>.
Elementy lokalne – elementy zagnieżdżone wewnątrz innych elementów.


Szkolenia INSPIRE
dla administracji
Deklaracja schematu
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">
...
</xs:schema>
xmlns:xs="http://www.w3.org/2001/XMLSchema" – elementy i typy danych użyte w schemacie pochodzą z przestrzeni nazw http://www.w3.org/2001/XMLSchema. Określa również, że elementy i typy danych, które pochodzą z tej przestrzeni nazw powinny być poprzedzone przedrostkiem xs:.
targetNamespace="http://www.w3schools.com" – elementy zdefiniowane przez schemat (np. <książka>, <tytuł>, <autor>) pochodzą z przestrzeni nazw http://www.w3schools.com.
xmlns="http://www.w3schools.com" – domyślną przestrzenią nazw jest http://www.w3schools.com.
elementFormDefault="qualified" – każdy element użyty w instancji (egzemplarzu) dokumentu XML, który został zadeklarowany w schemacie musi mieć określoną przestrzeń nazw.


Szkolenia INSPIRE
dla administracji
Dokument XML
<?xml version="1.0" encoding="ISO-8859-2"?>
<księgarnia xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.com księgarnia.xsd">
<książka kategoria="DZIECI">
<tytuł>Harry Potter i czara ognia</tytuł>
<autor>J K. Rowling</autor>
<rok>2001</rok>
<cena>49.99</cena>
</książka>
</księgarnia>
xmlns="http://www.w3schools.com" określa domyślną deklarację przestrzeni nazw, która oznacza, że wszystkie elementy użyte w tym dokumencie XML są zadeklarowane w przestrzeni nazw http://www.w3schools.com.
Jeżeli przestrzeń nazw instancji XML Schema jest dostępna: xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" można zastosować atrybut schemaLocation. Ma on dwie wartości:
pierwszą jest przestrzeń nazw,
drugą lokalizacja schematu XML dla tej przestrzeni nazw: xsi:schemaLocation="http://www.w3schools.com księgarnia.xsd"


Szkolenia INSPIRE
dla administracji
Typy proste (element simpleType)
Element prosty – element XML, który:
– zawiera tylko tekst,
– nie może zawierać żadnych innych elementów i atrybutów.
Składania definicji elementu prostego:
<xs:element name="xxx" type="yyy"/>
– "xxx" nazwa elementu,
– "yyy" typ danych tego elementu.


Szkolenia INSPIRE
dla administracji
Typy proste (element simpleType)
XML Schema posiada wiele wbudowanych typów danych. Najbardziej popularne to:
– xs:string
– xs:decimal
– xs:integer
– xs:boolean
– xs:date
– xs:time
Element prosty może mieć również określoną wartość domyślną (default) lub stałą (fixed).


Szkolenia INSPIRE
dla administracji
Typy proste (element simpleType)
W XML Schema wszystkie atrybuty są zadeklarowane jako typy proste.
Elementy proste nie mogą mieć atrybutów.
Element, który posiada atrybuty jest typu złożonego (complexType).
Atrybut (sam w sobie) jest zawsze zadeklarowany jako typ prosty (simpleType).

Szkolenia INSPIRE dla administracji

Typy proste (element **simpleType**)

- Składania definiowania atrybutu:
  <xs:attribute name="xxx" type="yyy"/>
  - "xxx" nazwa atrybutu,
  - "yyy" typ danych atrybutu.
- Atrybut może:
  - mieć określoną wartość domyślną lub stałą,
  - być opcjonalny lub wymagany (**use="required"**). Domyślnie atrybut jest opcjonalny.

Szkolenia INSPIRE dla administracji

Typy proste (element **simpleType**)

- W XML Schema można zdefiniować ograniczenie (**restriction**) zawartości elementu lub atrybutu, które jest stosowane do określania akceptowalnych wartości elementów i atrybutów XML.

Ograniczenie	Opis
enumeration	Definiuje listę dopuszczalnych wartości
fractionDigits	Określa max liczbę dozwolonych miejsc dziesiętnych. Musi być większe lub równe 0
length	Określa dokładną liczbę znaków lub listę elementów dozwolonych. Musi być większe lub równe 0
maxExclusive	Określa górną granicę dla wartości numerycznych (jej wartość musi być mniejsza niż ta wartość)
maxInclusive	Określa górną granicę dla wartości numerycznych (jej wartość musi być mniejsza lub równa tej wartości)
maxLength	Określa max liczbę znaków lub listę elementów dozwolonych. Musi być większe lub równe 0
minExclusive	Określa dolną granicę dla wartości numerycznych (jej wartość musi być większa niż ta wartość)
minInclusive	Określa dolną granicę dla wartości numerycznych (jej wartość musi być większa lub równa tej wartości)
minLength	Określa min liczbę znaków lub listę elementów dozwolonych. Musi być większe lub równe 0
pattern	Definiuje dokładną sekwencję dopuszczalnych znaków
totalDigits	Określa dokładną liczbę dozwolonych cyfr. Musi być większe od 0
whiteSpace	Określa jak „białe" znaki (whitespace – przesunięcia o wiersz, tabulatory, spacje i powroty karetki) są obsługiwane

Ograniczenia dla typów danych
Źródło: W3Schools Online Web Tutorials

Szkolenia INSPIRE dla administracji

Ograniczenie – przykład

```
<xs:element name="haslo">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="5"/>
      <xs:maxLength value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- Element prosty "haslo" z ograniczeniem zawartości.
- Długość "hasła" musi wynosić co najmniej 5 i co najwyżej 8 znaków.

Szkolenia INSPIRE dla administracji

Typy złożone (element **complexType**)

- **Element złożony** – element XML, który:
  - zawiera inne elementy i/lub
  - atrybuty.
- Wyróżnia się 4 rodzaje elementów złożonych:
  - elementy puste,
  - elementy, które zawierają tylko inne elementy,
  - elementy, które zawierają tylko tekst,
  - elementy, które zawierają zarówno inne elementy jak i tekst.

Każdy z tych elementów może również zawierać atrybuty.

Szkolenia INSPIRE dla administracji

Deklaracja elementu złożonego

- W XML Schema można go zdefiniować na dwa sposoby:
  - bezpośrednio zadeklarować element złożony przez nazwanie go lub
  - przypisać elementowi złożonemu nazwę i atrybut **type**, który odnosi się do nazwy.
    - Wtedy kilka elementów w schemacie może odwoływać się do tego samego typu złożonego.

Szkolenia INSPIRE dla administracji

Deklaracja elementu złożonego – przykład

- **Pusty element złożony**
  - nie może mieć żadnej zawartości poza atrybutami.

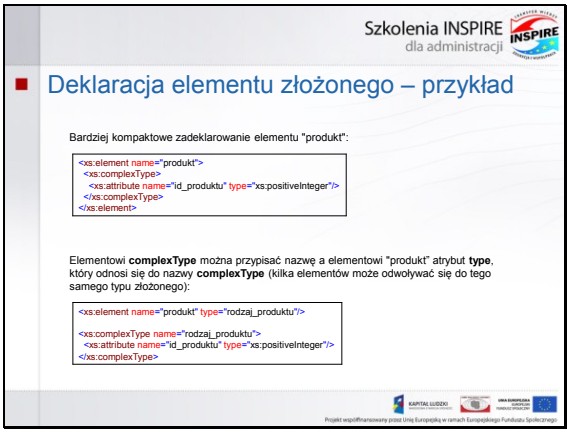
Pusty element w XML:

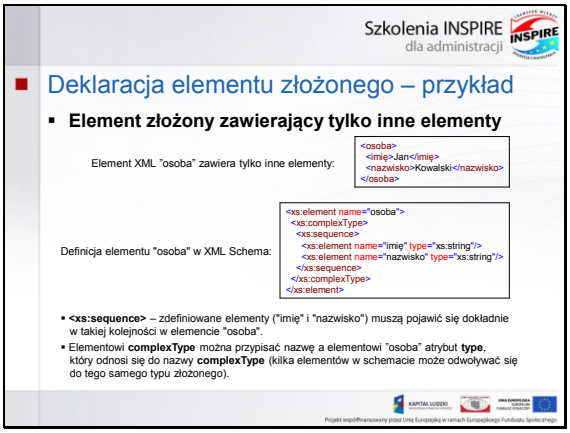
```
<produkt id="1345"/>

<produkt id="1345"></produkt>
```

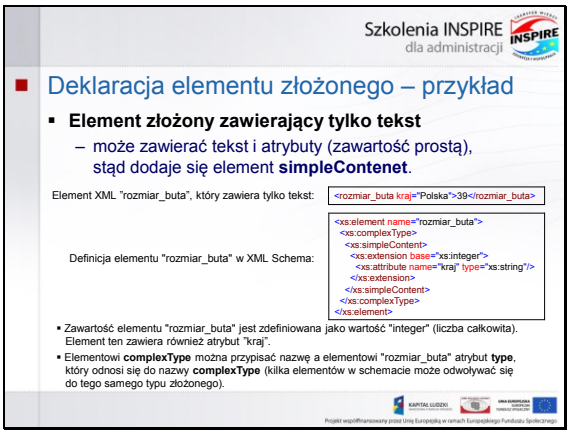
Definicja typu bez zawartości w XML Schema wymaga zdefiniowania typu (który zezwala na występowanie elementów w swojej zawartości, ale faktycznie nie deklaruje się żadnych elementów):

```
<xs:element name="produkt">
  <xs:complexType>
    <xs:complexContent>
      <xs:restriction base="xs:integer">
        <xs:attribute name="id_produktu" type="xs:positiveInteger"/>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

- Element **complexContent** sygnalizuje, że model zawartości typu złożonego zostanie ograniczony lub rozszerzony.
- Ograniczenie typu **xs:restriction base="xs:integer"** deklaruje jeden atrybut, ale nie wprowadza żadnego elementu zawartości.


Szkolenia INSPIRE
dla administracji
Deklaracja elementu złożonego – przykład
Bardziej kompaktowe zadeklarowanie elementu "produkt":
<xs:element name="produkt">
<xs:complexType>
<xs:attribute name="id_produktu" type="xs:positiveInteger"/>
</xs:complexType>
</xs:element>
Elementowi complexType można przypisać nazwę a elementowi "produkt" atrybut type, który odnosi się do nazwy complexType (kilka elementów może odwoływać się do tego samego typu złożonego):
<xs:element name="produkt" type="rodzaj_produktu"/>
<xs:complexType name="rodzaj_produktu">
<xs:attribute name="id_produktu" type="xs:positiveInteger"/>
</xs:complexType>


Szkolenia INSPIRE
dla administracji
Deklaracja elementu złożonego – przykład
Element złożony zawierający tylko inne elementy
Element XML "osoba" zawiera tylko inne elementy:
<osoba>
<imię>Jan</imię>
<nazwisko>Kowalski</nazwisko>
</osoba>
Definicja elementu "osoba" w XML Schema:
<xs:element name="osoba">
<xs:complexType>
<xs:sequence>
<xs:element name="imię" type="xs:string"/>
<xs:element name="nazwisko" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:sequence> – zdefiniowane elementy ("imię" i "nazwisko") muszą pojawić się dokładnie w takiej kolejności w elemencie "osoba".
Elementowi complexType można przypisać nazwę a elementowi "osoba" atrybut type, który odnosi się do nazwy complexType (kilka elementów w schemacie może odwoływać się do tego samego typu złożonego).


Szkolenia INSPIRE
dla administracji
Deklaracja elementu złożonego – przykład
Element złożony zawierający tylko tekst
– może zawierać tekst i atrybuty (zawartość prostą), stąd dodaje się element simpleContenet.
Element XML "rozmiar_buta", który zawiera tylko tekst:
<rozmiar_buta kraj="Polska">39</rozmiar_buta>
Definicja elementu "rozmiar_buta" w XML Schema:
<xs:element name="rozmiar_buta">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:integer">
<xs:attribute name="kraj" type="xs:string"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
Zawartość elementu "rozmiar_buta" jest zdefiniowana jako wartość "integer" (liczba całkowita). Element ten zawiera również atrybut "kraj".
Elementowi complexType można przypisać nazwę a elementowi "rozmiar_buta" atrybut type, który odnosi się do nazwy complexType (kilka elementów w schemacie może odwoływać się do tego samego typu złożonego).

Szkolenia INSPIRE dla administracji

Deklaracja elementu złożonego – przykład

- **Element złożony z mieszaną zawartością**
  - może zawierać atrybuty, elementy i tekst.

Element XML "list", który zawiera tekst i inne elementy:

```
<list>
  Szanowny Panie<nazwisko>Kowalski</nazwisko>.
  Pana zamówienie<id_zamówienia>1032</id_zamówienia>
  zostanie zrealizowane<data_dostawy>2011-06-25</data_dostawy>
</list>
```

Definicja "list" elementu w XML Schema:

```
<xs:element name="list">
 <xs:complexType mixed="true">
  <xs:sequence>
   <xs:element name="nazwisko" type="xs:string"/>
   <xs:element name="id_zamówienia" type="xs:positiveInteger"/>
   <xs:element name="data_dostawy" type="xs:date"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

- Aby w elemencie "list" mogły pojawić się dane typu znakowego między elementami „dziećmi", atrybut **mixed** musi być ustawiony na "true".
- **<xs:sequence>** – zdefiniowane elementy ("nazwisko", "id_zamówienia" i "data_dostawy") muszą pojawić się dokładnie w takiej kolejności wewnątrz elementu "list".
- Elementowi **complexType** można przypisać nazwę a elementowi "list" atrybut **type**, który odnosi się do nazwy **complexType** (kilka elementów w schemacie może odwoływać się do tego samego typu złożonego).

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego.

Szkolenia INSPIRE dla administracji

Deklaracja elementu złożonego – przykład

- Definicje nowych typów prostych i złożonych (elementy **simpleType** i **complexType**) można oprzeć na elementach już istniejących w schemacie
  - rozszerzyć je i dodać do nich nowe elementy (**extension**):

```
<xs:element name="pracownik" type="pelne_dane_osobowe"/>

<xs:complexType name="dane_osobowe">
 <xs:sequence>
  <xs:element name="imię" type="xs:string"/>
  <xs:element name="nazwisko" type="xs:string"/>
 </xs:sequence>
</xs:complexType>

<xs:complexType name="pelne_dane_osobowe">
 <xs:complexContent>
  <xs:extension base="dane_osobowe">
   <xs:sequence>
    <xs:element name="ulica" type="xs:string"/>
    <xs:element name="kod_pocztowy" type="xs:string"/>
    <xs:element name="miasto" type="xs:string"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

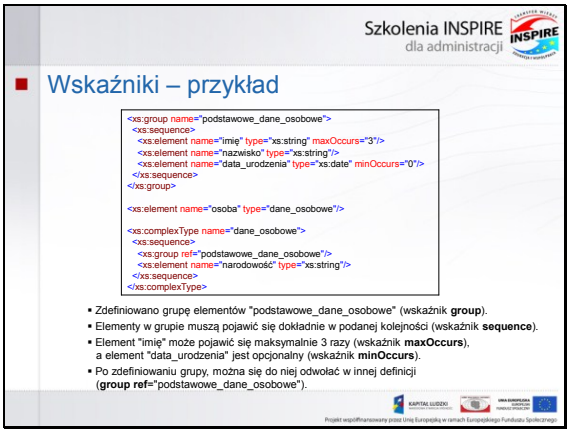
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego.

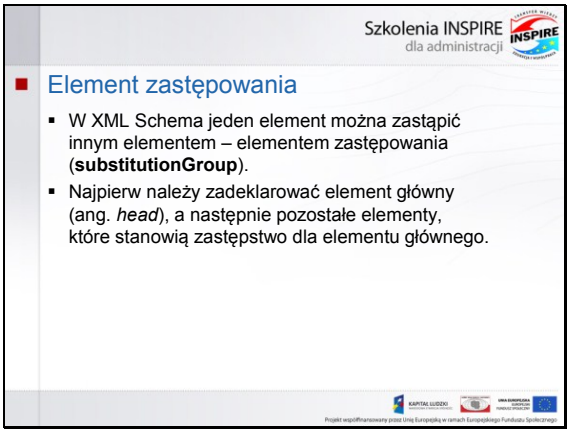
Szkolenia INSPIRE dla administracji

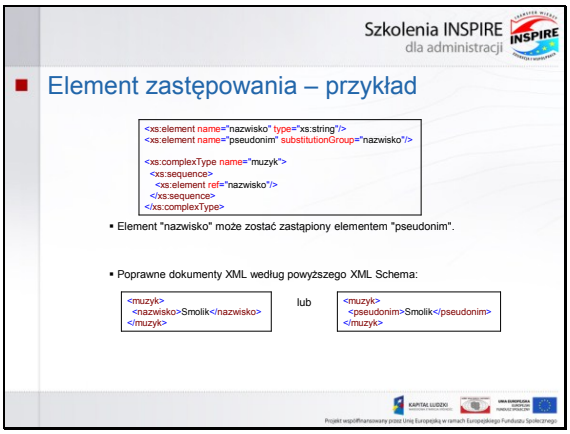
Wskaźniki

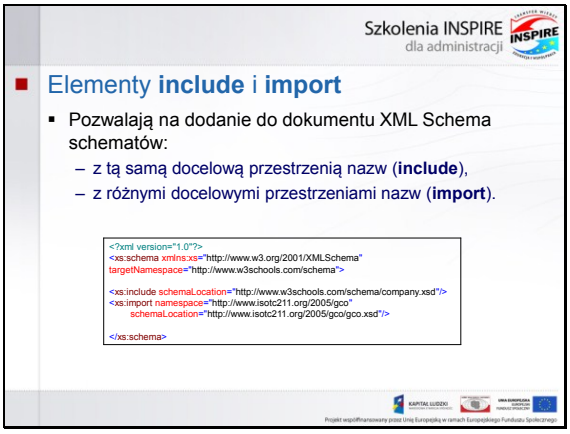
- Pozwalają kontrolować sposób używania elementów w dokumencie:
  - **Wskaźniki porządkowe** – definiują kolejność elementów:
    - *all* – elementy „dzieci" mogą pojawić się w dowolnej kolejności, i każdy element „dziecko" musi pojawić się tylko raz,
    - *choice* – jeden albo więcej elementów „dziecko" może wystąpić,
    - *sequence* – elementy „dzieci" muszą pojawić się w określonej kolejności.
  - **Wskaźniki występowania** – definiują częstość występowania elementów:
    - *maxOccurs* – max ilość wystąpień elementu ,
    - *minOccurs* – min ilość wystąpień elementu.
  - **Wskaźniki grupy** – definiują powiązane zbiory elementów:
    - *group* – nazwana grupa elementów,
    - *attributeGroup* – nazwana grupa atrybutów.

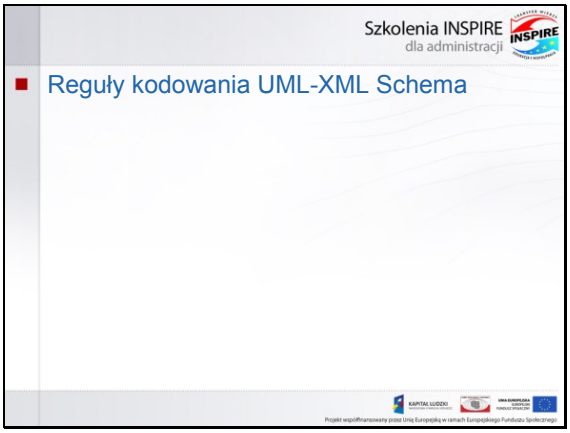
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego.

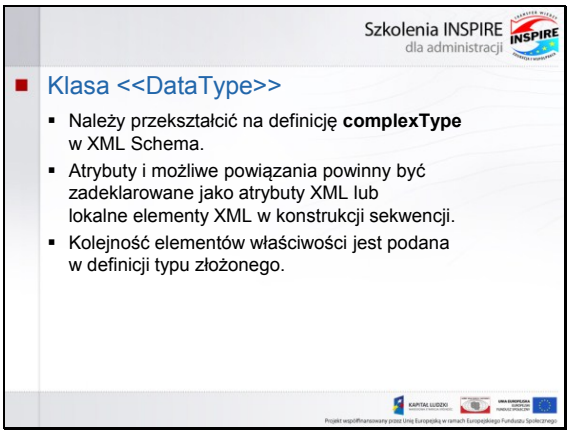

Szkolenia INSPIRE
dla administracji
Wskaźniki – przykład
<xs:group name="podstawowe_dane_osobowe">
<xs:sequence>
<xs:element name="imię" type="xs:string" maxOccurs="3"/>
<xs:element name="nazwisko" type="xs:string"/>
<xs:element name="data_urodzenia" type="xs:date" minOccurs="0"/>
</xs:sequence>
</xs:group>
<xs:element name="osoba" type="dane_osobowe"/>
<xs:complexType name="dane_osobowe">
<xs:sequence>
<xs:group ref="podstawowe_dane_osobowe"/>
<xs:element name="narodowość" type="xs:string"/>
</xs:sequence>
</xs:complexType>
Zdefiniowano grupę elementów "podstawowe_dane_osobowe" (wskaźnik group).
Elementy w grupie muszą pojawić się dokładnie w podanej kolejności (wskaźnik sequence).
Element "imię" może pojawić się maksymalnie 3 razy (wskaźnik maxOccurs), a element "data_urodzenia" jest opcjonalny (wskaźnik minOccurs).
Po zdefiniowaniu grupy, można się do niej odwołać w innej definicji (group ref="podstawowe_dane_osobowe").


Szkolenia INSPIRE
dla administracji
Element zastępowania
W XML Schema jeden element można zastąpić innym elementem – elementem zastępowania (substitutionGroup).
Najpierw należy zadeklarować element główny (ang. head), a następnie pozostałe elementy, które stanowią zastępstwo dla elementu głównego.


Szkolenia INSPIRE
dla administracji
Element zastępowania – przykład
<xs:element name="nazwisko" type="xs:string"/>
<xs:element name="pseudonim" substitutionGroup="nazwisko"/>
<xs:complexType name="muzyk">
<xs:sequence>
<xs:element ref="nazwisko"/>
</xs:sequence>
</xs:complexType>
Element "nazwisko" może zostać zastąpiony elementem "pseudonim".
Poprawne dokumenty XML według powyższego XML Schema:
<muzyk>
<nazwisko>Smolik</nazwisko>
</muzyk>
lub
<muzyk>
<pseudonim>Smolik</pseudonim>
</muzyk>


Szkolenia INSPIRE
dla administracji
Elementy include i import
Pozwalają na dodanie do dokumentu XML Schema schematów:
– z tą samą docelową przestrzenią nazw (include),
– z różnymi docelowymi przestrzeniami nazw (import).
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com/schema">
<xs:include schemaLocation="http://www.w3schools.com/schema/company.xsd"/>
<xs:import namespace="http://www.isotc211.org/2005/gco"
schemaLocation="http://www.isotc211.org/2005/gco/gco.xsd"/>
</xs:schema>


Szkolenia INSPIRE
dla administracji
Reguły kodowania UML-XML Schema


Szkolenia INSPIRE
dla administracji
Klasa <<DataType>>
Należy przekształcić na definicję complexType w XML Schema.
Atrybuty i możliwe powiązania powinny być zadeklarowane jako atrybuty XML lub lokalne elementy XML w konstrukcji sekwencji.
Kolejność elementów właściwości jest podana w definicji typu złożonego.

Szkolenia INSPIRE dla administracji

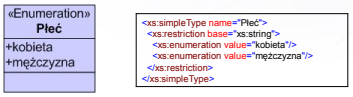
Klasa <<BasicType>>

- Należy przekształcić na deklarację **simpleType** w XML Schema.
- Każdy z typów danych zdefiniowany w XML Schema może zostać użyty jako moduł konstrukcyjny do określenia typów podstawowych zdefiniowanych przez użytkownika.
- Podstawowe typy danych zdefiniowane przez ISO/TS 19103 należy przekształcić na odpowiadające im typy danych w XML Schema.

Szkolenia INSPIRE dla administracji

Klasa <<Enumeration>>

- Należy przekształcić na typ prosty, który ogranicza łańcuch tekstowy do kilku wartości wyliczeniowych.



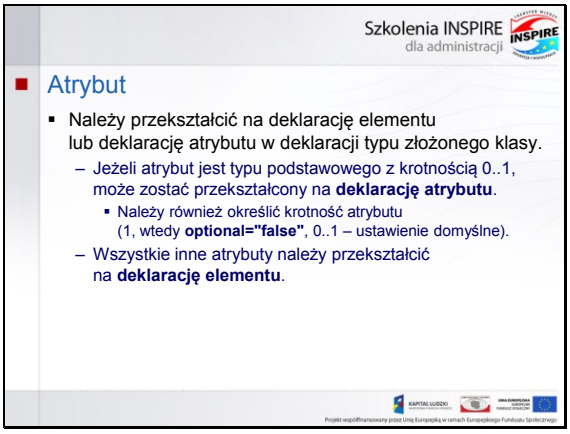

```
<xs:simpleType name="Płeć">
  <xs:restriction base="xs:string">
    <xs:enumeration value="kobieta"/>
    <xs:enumeration value="mężczyzna"/>
  </xs:restriction>
</xs:simpleType>
```

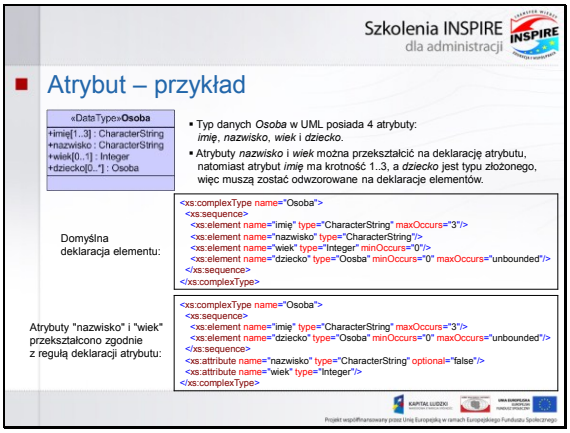
Przekształcenia klasy **<<Enumeration>> Płeć** na odpowiednią strukturę w XML Schema

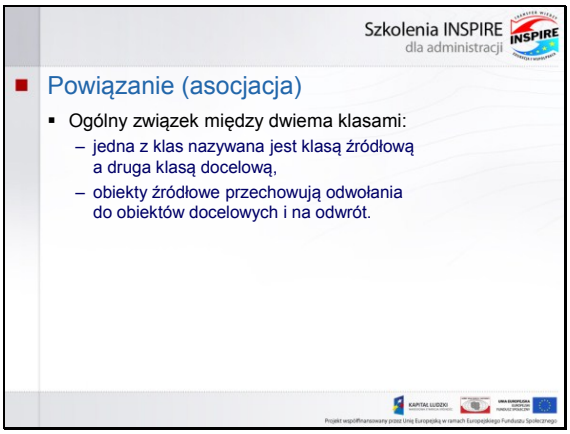
Szkolenia INSPIRE dla administracji

Klasa <<CodeList>>

- Nie powinna być przekształcana.
- Może być odwzorowana na słownik, który przechowuje pary: *kod* i *wartość* zdefiniowane w liście kodowej.
- Słownik powinien być dostępny publicznie a jego adres www należy podać jako URI.
- Atrybut typu lista kodowa należy kodować jako wartość łańcucha znaków (*string*).


Szkolenia INSPIRE
dla administracji
Atrybut
Należy przekształcić na deklarację elementu lub deklarację atrybutu w deklaracji typu złożonego klasy.
Jeżeli atrybut jest typu podstawowego z krotnością 0..1, może zostać przekształcony na deklarację atrybutu.
Należy również określić krotność atrybutu (1, wtedy optional="false", 0..1 – ustawienie domyślne).
Wszystkie inne atrybuty należy przekształcić na deklarację elementu.


Szkolenia INSPIRE
dla administracji
Atrybut – przykład
«DataType»Osoba
+imię[1..3] : CharacterString
+nazwisko : CharacterString
+wiek[0..1] : Integer
+dziecko[0..*] : Osoba
Typ danych Osoba w UML posiada 4 atrybuty: imię, nazwisko, wiek i dziecko.
Atrybuty nazwisko i wiek można przekształcić na deklarację atrybutu, natomiast atrybut imię ma krotność 1..3, a dziecko jest typu złożonego, więc muszą zostać odwzorowane na deklaracje elementów.
Domyślna deklaracja elementu:
<xs:complexType name="Osoba">
<xs:sequence>
<xs:element name="imię" type="CharacterString" maxOccurs="3"/>
<xs:element name="nazwisko" type="CharacterString"/>
<xs:element name="wiek" type="Integer" minOccurs="0"/>
<xs:element name="dziecko" type="Osoba" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
Atrybuty "nazwisko" i "wiek" przekształcono zgodnie z regułą deklaracji atrybutu:
<xs:complexType name="Osoba">
<xs:sequence>
<xs:element name="imię" type="CharacterString" maxOccurs="3"/>
<xs:element name="dziecko" type="Osoba" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="nazwisko" type="CharacterString" optional="false"/>
<xs:attribute name="wiek" type="Integer"/>
</xs:complexType>


Szkolenia INSPIRE
dla administracji
Powiązanie (asocjacja)
Ogólny związek między dwiema klasami:
jedna z klas nazywana jest klasą źródłową a druga klasą docelową,
obiekty źródłowe przechowują odwołania do obiektów docelowych i na odwrót.

Szkolenia INSPIRE dla administracji

Powiązanie (asocjacja)

- Typ złożony odpowiadający klasie źródłowej powinien zawierać deklarację elementu (jeśli asocjacja jest nawigowalna i klasa docelowa jest identyfikowana przez nazwę roli).
  - Nazwa elementu powinna być nazwą roli identyfikującej klasę docelową a typem powinno być albo odwołanie się do obiektu docelowego (*IM_ObjectReference*), albo typ, który oparty jest na lub posiada atrybut zdefiniowany w odwołaniu się do obiektu docelowego (*IM_ObjectReference*).
- Typ złożony odpowiadający klasie docelowej powinien zawierać deklarację elementu (jeśli asocjacja jest nawigowalna i klasa źródłowa jest identyfikowana przez nazwę roli).
  - Nazwa elementu powinna być nazwą roli identyfikującej klasę źródłową a typem powinno być albo odwołanie się do obiektu docelowego (*IM_ObjectReference*), albo typ, który oparty jest na lub posiada atrybut zdefiniowany w odwołaniu się do obiektu docelowego (*IM_ObjectReference*).

Szkolenia INSPIRE dla administracji

Powiązanie (asocjacja) – przykład

- Powiązanie między klasą *Firma* i *Osoba*, gdzie tylko klasa *Firma* wie o klasie *Osoba* – zdefiniowano tylko jedną nazwę roli.

Przekształcenie asocjacji na odpowiednią deklarację w XML Schema:

```
<xs:complexType name="Firma">
  <xs:sequence>
    <xs:element name="pracownik" type="OdwolanieDoOsoba" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="OdwolanieDoOsoba">
  <xs:attributeGroup ref="IM_ObjectReference"/>
</xs:complexType>
```

Szkolenia INSPIRE dla administracji

Agregacja zwykła

- Słaby związek typu „całość-część” między klasą-całość a klasą-część.
- Posiadanie jest słabe, tzn. „części” mogą być członkami więcej niż jednej „całości” w tym samym czasie.
- Obiekt „część” może być współdzielony przez więcej niż jeden obiekt „całość”.
- Ogólnie „całość” może tylko przechowywać odniesienia do swoich „części”, ale w przypadku całkowitego posiadania może zawierać odpowiednie „części”.

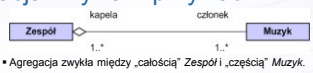
Szkolenia INSPIRE dla administracji

Agregacja zwykła

- Typ złożony odpowiadający klasie „całość” powinien zawierać deklarację elementu, gdzie nazwa odpowiada nazwie roli identyfikującej klasę „część”.
  - Typ elementu powinien być oparty na odwołaniu się do obiektu docelowego (*IM_ObjectReference*) i może zawierać zero lub jeden element typu, który odpowiada klasie „część”.
- Typ złożony odpowiadający klasie „część” powinien zawierać deklarację elementu (jeśli powiązanie jest nawigowalne i klasa docelowa jest identyfikowana przez nazwę roli).
  - Nazwa elementu powinna odpowiadać nazwie roli identyfikującej klasę „całość” a typ powinien być oparty na odwołaniu się do obiektu docelowego (*IM_ObjectReference*).

Szkolenia INSPIRE dla administracji

Agregacja zwykła – przykład



- Agregacja zwykła między „całością” *Zespół* i „częścią” *Muzyk*.
- *Zespół* identyfikuje *Muzyka* przez nazwę roli *członek* oraz *Muzyk* identyfikuje *Zespół* przez nazwę roli *kapela*.

Przekształcenie agregacji zwykłej na odpowiednią deklarację w XML Schema:

```
<xs:complexType name="Zespół">
 <xs:sequence>
  <xs:element name="członek" minOccurs="1" maxOccurs="unbounded">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="Muzyk" type="Muzyk" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attributeGroup ref="IM_ObjectReference"/>
   </xs:complexType>
  </xs:element>
 </xs:sequence>
</xs:complexType>

<xs:complexType name="Muzyk">
 <xs:sequence>
  <xs:element name="kapela" type="IM_ObjectReference" minOccurs="1" maxOccurs="unbounded"/>
 </xs:sequence>
</xs:complexType>
```

Szkolenia INSPIRE dla administracji

Agregacja całkowita (kompozycja)

- Mocny związek typu „całość-część” między klasą-całość a klasą-część.
- Posiadanie jest mocne, tzn. „część” może być członkiem dokładnie jednego obiektu „całość”.
- Kompozycja powinna zawierać swoje odpowiednie „części”.

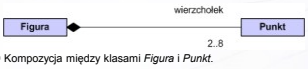
Szkolenia INSPIRE dla administracji

Agregacja całkowita (kompozycja)

- Typ złożony odpowiadający klasie „całość" powinien zawierać deklarację elementu, gdzie nazwa odpowiada nazwie roli identyfikującej klasę „część" a typ odpowiada typowi klasy „część".
- Typ złożony odpowiadający klasie „część" nie powinien zawierać żadnej deklaracji elementu nawet (jeśli nawa roli identyfikuje klasę „całość").
  - „Część" zawsze będzie zawierać się wewnątrz klasy „całość".

Szkolenia INSPIRE dla administracji

Agregacja całkowita (kompozycja) – przykład



- Kompozycja między klasami *Figura* i *Punkt*.
- *Figura* identyfikuje *Punkt* przez docelową nazwę roli *wierzchołek*.

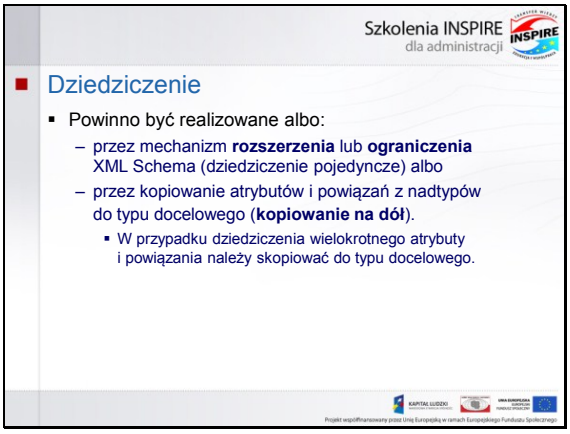
Przekształcenie kompozycji na odpowiednią deklarację w XML Schema:

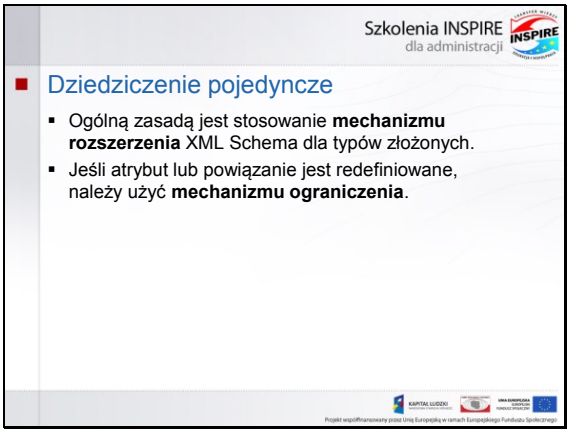
```
<xs:complexType name="Figura">
  <xs:sequence>
    <xs:element name="wierzchołek" type="Punkt" minOccurs="2" maxOccurs="8"/>
  </xs:sequence>
</xs:complexType>
```

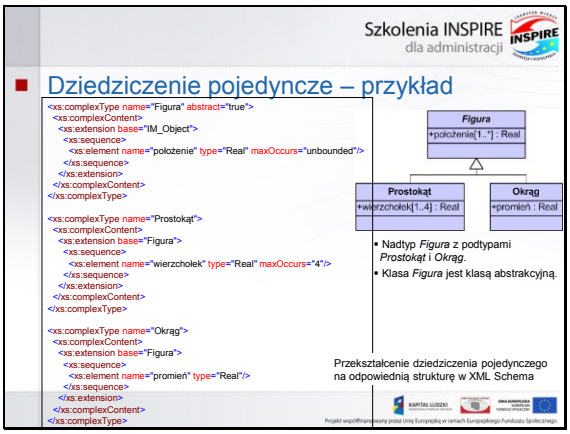
Szkolenia INSPIRE dla administracji

Dziedziczenie

- Pozwala podtypowi dziedziczyć atrybuty i powiązania jego nadtypu.
  - **Dziedziczenie pojedyncze** – typ może dziedziczyć tylko z jednego nadtypu.
  - **Dziedziczenie wielokrotne** – typ może dziedziczyć z więcej niż jednego typu.
- **UML** dopuszcza dziedziczenie pojedyncze i wielokrotne.
- **XML Schema** obsługuje tylko dziedziczenie pojedyncze. Dziedziczenie wielokrotne należy zasymulować.


Szkolenia INSPIRE
dla administracji
Dziedziczenie
Powinno być realizowane albo:
– przez mechanizm **rozszerzenia** lub **ograniczenia** XML Schema (dziedziczenie pojedyncze) albo
– przez kopiowanie atrybutów i powiązań z nadtypów do typu docelowego (**kopiowanie na dół**).
W przypadku dziedziczenia wielokrotnego atrybuty i powiązania należy skopiować do typu docelowego.


Szkolenia INSPIRE
dla administracji
Dziedziczenie pojedyncze
Ogólną zasadą jest stosowanie **mechanizmu rozszerzenia** XML Schema dla typów złożonych.
Jeśli atrybut lub powiązanie jest redefiniowane, należy użyć **mechanizmu ograniczenia**.


Szkolenia INSPIRE
dla administracji
Dziedziczenie pojedyncze – przykład
<xs:complexType name="Figura" abstract="true">
<xs:complexContent>
<xs:extension base="IM_Object">
<xs:sequence>
<xs:element name="położenie" type="Real" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Prostokąt">
<xs:complexContent>
<xs:extension base="Figura">
<xs:sequence>
<xs:element name="wierzchołek" type="Real" maxOccurs="4"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Okrąg">
<xs:complexContent>
<xs:extension base="Figura">
<xs:sequence>
<xs:element name="promień" type="Real"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
Figura
+położenie[1..*] : Real
Prostokąt
+wierzchołek[1..4] : Real
Okrąg
+promień : Real
Nadtyp *Figura* z podtypami *Prostokąt* i *Okrąg*.
Klasa *Figura* jest klasą abstrakcyjną.
Przekształcenie dziedziczenia pojedynczego na odpowiednią strukturę w XML Schema

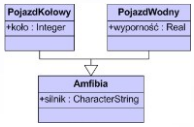
Szkolenia INSPIRE dla administracji

Dziedziczenie wielokrotne

- Procedurę kopiowania atrybutów/powiązań należy rozpocząć od:
  - lewego nadtypu – skopiować jego atrybuty i powiązania,
  - następnie kontynuować z następnym nadtypem na prawo,
  - aż do osiągnięcia nadtypu położonego najbardziej na prawo,
  - atrybuty podtypu dodawane są jako ostatnie.
- Jeśli nadtyp lub podtyp definiuje atrybut lub powiązanie z tą samą nazwą jak wcześniej skopiowane, ma miejsce konflikt nazw.
  - Wtedy ostatni atrybut lub powiązanie powinno mieć pierwszeństwo i zastąpić te wcześniej skopiowane.

Szkolenia INSPIRE dla administracji

Dziedziczenie wielokrotne – przykład



- Typy: *PojazdKołowy*, *PojazdWodny* i *Amfibia*.
- *Amfibia* jest podtypem dwóch pozostałych.

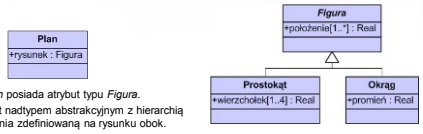
Przekształcenie dziedziczenia wielokrotnego na odpowiednią strukturę w XML Schema

```
<xs:complexType name="PojazdKołowy">
  <xs:sequence>
    <xs:element name="koło" type="Integer"/>
  </xs:sequence>
  <xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>

<xs:complexType name="PojazdWodny">
  <xs:sequence>
    <xs:element name="wyporność" type="Real"/>
  </xs:sequence>
  <xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>

<xs:complexType name="Amfibia">
  <xs:sequence>
    <xs:element name="koło" type="Integer"/>
    <xs:element name="wyporność" type="Real"/>
    <xs:element name="silnik" type="CharacterString"/>
  </xs:sequence>
  <xs:attributeGroup ref="IM_ObjectIdentification"/>
</xs:complexType>
```

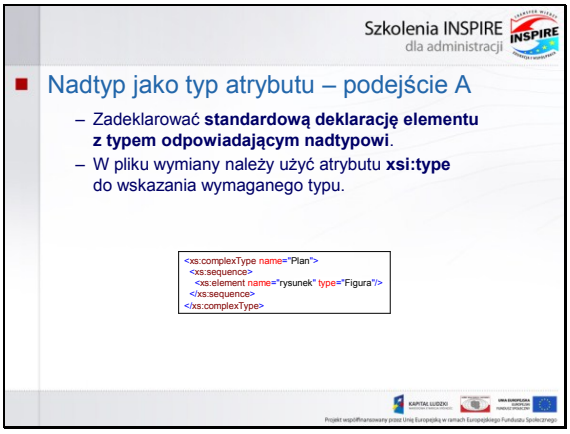
Szkolenia INSPIRE dla administracji

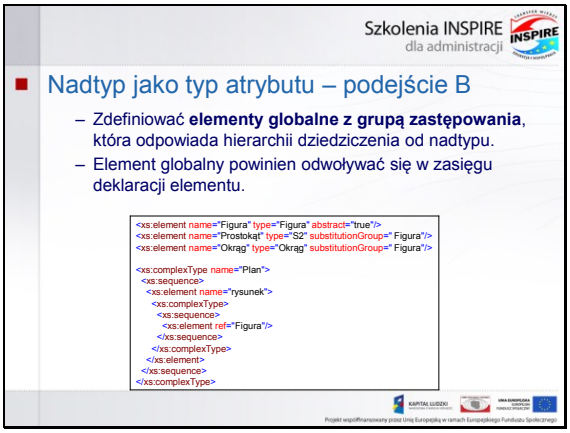
Nadtyp jako typ atrybutu

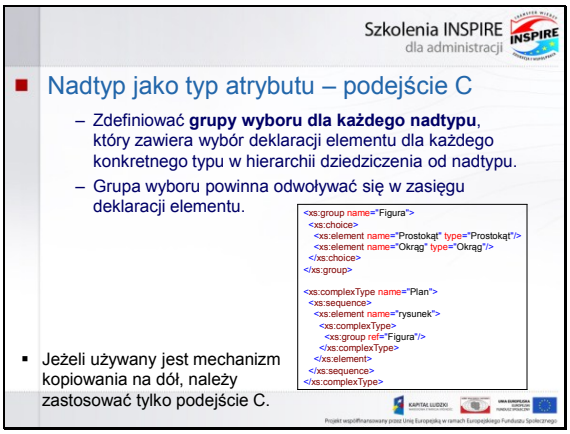
- Instancja atrybutu może być jednym z konkretnych podtypów zdefiniowanych przez hierarchię dziedziczenia od nadtypu.
- Schemat XML nie obsługuje bezpośrednio **mechanizmu typu dynamicznego**, ale można wykorzystać trzy alternatywne podejścia:

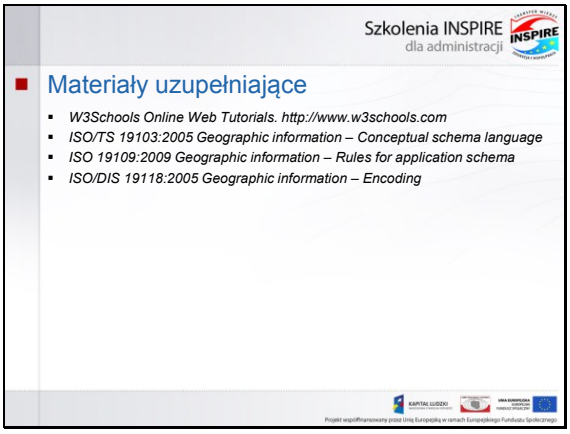


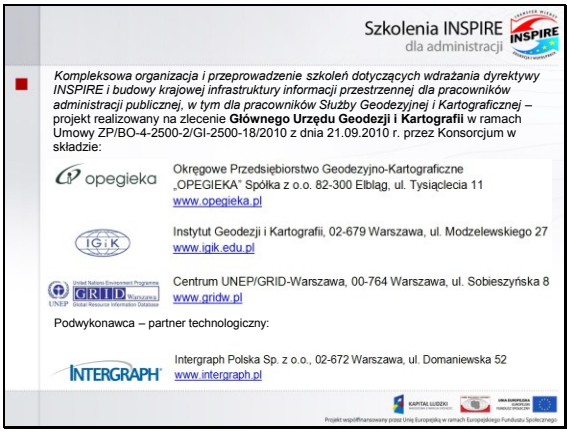
- Klasa *Plan* posiada atrybut typu *Figura*.
- *Figura* jest nadtypem abstrakcyjnym z hierarchią dziedziczenia zdefiniowaną na rysunku obok.


Szkolenia INSPIRE
dla administracji
Nadtyp jako typ atrybutu – podejście A
– Zadeklarować standardową deklarację elementu z typem odpowiadającym nadtypowi.
– W pliku wymiany należy użyć atrybutu xsi:type do wskazania wymaganego typu.
<xs:complexType name="Plan">
<xs:sequence>
<xs:element name="rysunek" type="Figura"/>
</xs:sequence>
</xs:complexType>


Szkolenia INSPIRE
dla administracji
Nadtyp jako typ atrybutu – podejście B
– Zdefiniować elementy globalne z grupą zastępowania, która odpowiada hierarchii dziedziczenia od nadtypu.
– Element globalny powinien odwoływać się w zasięgu deklaracji elementu.
<xs:element name="Figura" type="Figura" abstract="true"/>
<xs:element name="Prostokąt" type="S2" substitutionGroup=" Figura"/>
<xs:element name="Okrąg" type="Okrąg" substitutionGroup=" Figura"/>
<xs:complexType name="Plan">
<xs:sequence>
<xs:element name="rysunek">
<xs:complexType>
<xs:sequence>
<xs:element ref="Figura"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>


Szkolenia INSPIRE
dla administracji
Nadtyp jako typ atrybutu – podejście C
– Zdefiniować grupy wyboru dla każdego nadtypu, który zawiera wybór deklaracji elementu dla każdego konkretnego typu w hierarchii dziedziczenia od nadtypu.
– Grupa wyboru powinna odwoływać się w zasięgu deklaracji elementu.
<xs:group name="Figura">
<xs:choice>
<xs:element name="Prostokąt" type="Prostokąt"/>
<xs:element name="Okrąg" type="Okrąg"/>
</xs:choice>
</xs:group>
<xs:complexType name="Plan">
<xs:sequence>
<xs:element name="rysunek">
<xs:complexType>
<xs:group ref="Figura"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
Jeżeli używany jest mechanizm kopiowania na dół, należy zastosować tylko podejście C.

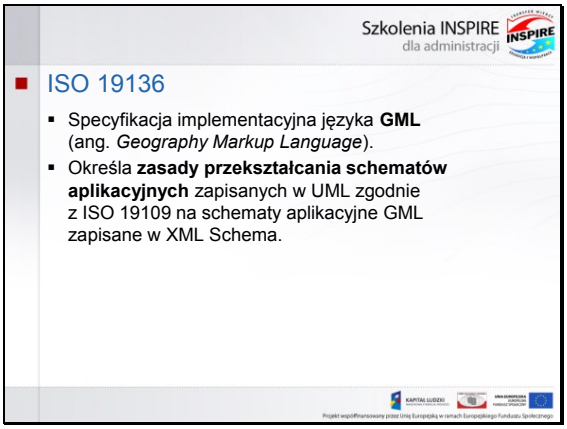

Szkolenia INSPIRE
dla administracji
Materiały uzupełniające
W3Schools Online Web Tutorials. http://www.w3schools.com
ISO/TS 19103:2005 Geographic information – Conceptual schema language
ISO 19109:2009 Geographic information – Rules for application schema
ISO/DIS 19118:2005 Geographic information – Encoding

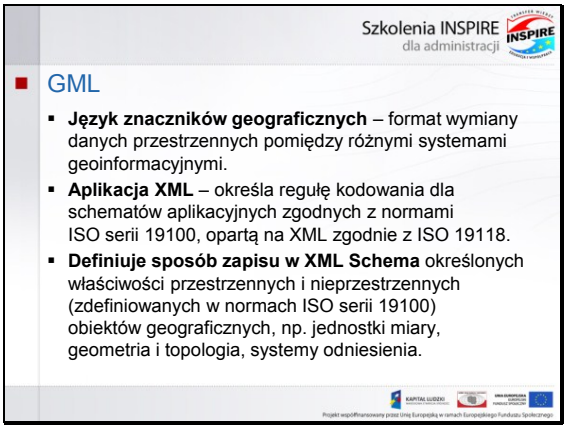

Szkolenia INSPIRE
dla administracji
Kompleksowa organizacja i przeprowadzenie szkoleń dotyczących wdrażania dyrektywy INSPIRE i budowy krajowej infrastruktury informacji przestrzennej dla pracowników administracji publicznej, w tym dla pracowników Służby Geodezyjnej i Kartograficznej – projekt realizowany na zlecenie Głównego Urzędu Geodezji i Kartografii w ramach Umowy ZP/BO-4-2500-2/GI-2500-18/2010 z dnia 21.09.2010 r. przez Konsorcjum w składzie:
opegieka
Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne „OPEGIEKA" Spółka z o.o. 82-300 Elbląg, ul. Tysiąclecia 11
www.opegieka.pl
IGiK
Instytut Geodezji i Kartografii, 02-679 Warszawa, ul. Modzelewskiego 27
www.igik.edu.pl
GRID
Centrum UNEP/GRID-Warszawa, 00-764 Warszawa, ul. Sobieszyńska 8
www.gridw.pl
Podwykonawca – partner technologiczny:
INTERGRAPH
Intergraph Polska Sp. z o.o., 02-672 Warszawa, ul. Domaniewska 52
www.intergraph.pl

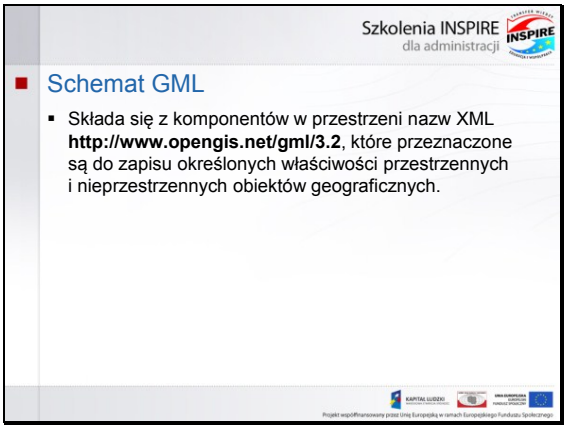

Szkolenia INSPIRE
dla administracji
Szkolenie eksperckie
Środki formalne
modelowania informacji geograficznej
– UML, XML, GML
wykładowca:
IMIĘ NAZWISKO
Zajęcia przygotowane na podstawie autorskiego programu opracowanego przez:
dr inż. Agnieszkę Chojkę z Uniwersytetu Warmińsko-Mazurskiego w Olsztynie

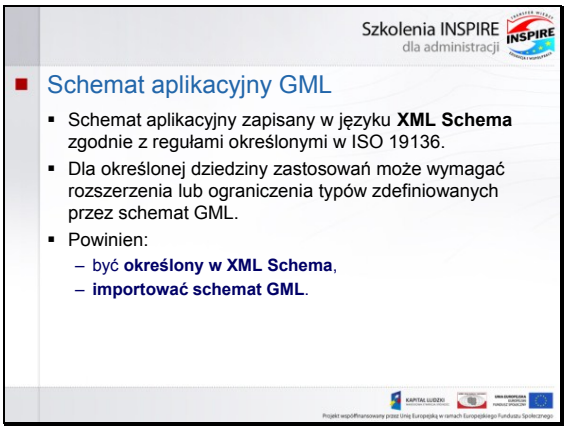
4. GML


Szkolenia INSPIRE
dla administracji
Środki formalne modelowania informacji geograficznej – UML, XML, GML
G M L
Reguły budowy schematów
aplikacyjnych w GML (XML Schema)


Szkolenia INSPIRE
dla administracji
ISO 19136
Specyfikacja implementacyjna języka GML (ang. Geography Markup Language).
Określa zasady przekształcania schematów aplikacyjnych zapisanych w UML zgodnie z ISO 19109 na schematy aplikacyjne GML zapisane w XML Schema.

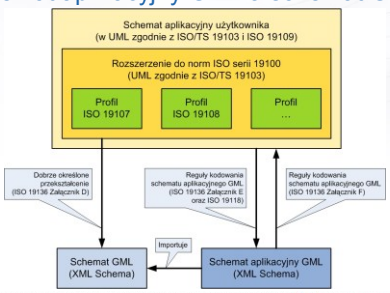

Szkolenia INSPIRE
dla administracji
GML
Język znaczników geograficznych – format wymiany danych przestrzennych pomiędzy różnymi systemami geoinformacyjnymi.
Aplikacja XML – określa regułę kodowania dla schematów aplikacyjnych zgodnych z normami ISO serii 19100, opartą na XML zgodnie z ISO 19118.
Definiuje sposób zapisu w XML Schema określonych właściwości przestrzennych i nieprzestrzennych (zdefiniowanych w normach ISO serii 19100) obiektów geograficznych, np. jednostki miary, geometria i topologia, systemy odniesienia.


Szkolenia INSPIRE
dla administracji
Schemat GML
Składa się z komponentów w przestrzeni nazw XML http://www.opengis.net/gml/3.2, które przeznaczone są do zapisu określonych właściwości przestrzennych i nieprzestrzennych obiektów geograficznych.


Szkolenia INSPIRE
dla administracji
Schemat aplikacyjny GML
Schemat aplikacyjny zapisany w języku XML Schema zgodnie z regułami określonymi w ISO 19136.
Dla określonej dziedziny zastosowań może wymagać rozszerzenia lub ograniczenia typów zdefiniowanych przez schemat GML.
Powinien:
– być określony w XML Schema,
– importować schemat GML.

Szkolenia INSPIRE dla administracji

Schemat aplikacyjny GML a schemat GML



Zależności między normami ISO serii 19100 a normą ISO 19136
Źródło: ISO 19136:2007

Szkolenia INSPIRE dla administracji

Konstrukcja schematu aplikacyjnego GML

- zastosowanie reguł dla schematów aplikacyjnych GML określonych w ISO 19136 dla tworzenia schematów aplikacyjnych bezpośrednio w XML Schema,
- zastosowanie reguł określonych w ISO 19109 dla schematów aplikacyjnych w UML, i dostosowanie ich zarówno do ograniczeń nakładanych na takie schematy, jak i do zasad przekształcania ich na schematy aplikacyjne GML określone w Załączniku E do ISO 19136. Przekształcenie schematu aplikacyjnego UML zgodnego z ISO 19109 na odpowiadający mu schemat aplikacyjny w GML oparte jest na zbiorze reguł kodowania zgodnych z regułami schematów aplikacyjnych GML i ISO 19118.

Szkolenia INSPIRE dla administracji

Reguły kodowania UML-GML

- Przekształcenie schematu aplikacyjnego UML, zgodnego z ISO 19109, na odpowiadający mu schemat aplikacyjny GML, oparte jest na zbiorze reguł kodowania określonych w ISO 19136 Załącznik E.
- **Założenie ogólne**:
  - definicja klasy w schemacie aplikacyjnym UML jest przekształcana na deklarację typu i elementu w Schemacie GML (XML Schema) według następujących zależności:

Szkolenia INSPIRE dla administracji

Reguły kodowania UML-GML

Schemat aplikacyjny UML	Schemat aplikacyjny GML
Pakiet	Jeden dokument XML Schema na pakiet (przekształcenie domyślne)
<<Application Schema>>	Dokument XML Schema
<<DataType>>	Element globalny, którego modelem zawartości jest element *complexType* w XML Schema o zakresie globalnym, typ właściwości
<<Enumeration>>	Ograniczenie *xsd:string* z wartościami wyliczenia
<<CodeList>>	Unia wyliczenia i wzorca (przekształcenie domyślne, przekształcenie alternatywne jest odwołanie do słownika)
<<Union>>	Grupa wyboru, której członkami są obiekty GML lub obiekty odpowiadające *DataTypes*
<<FeatureType>>	Element globalny, którego modelem zawartości jest typ XML Schema o zakresie globalnym, pochodzący z bezpośredniego/pośredniego rozszerzenia *gml:AbstractFeatureType*, typ właściwości
Brak stereotypu lub <<Type>>	Element globalny, którego modelem zawartości jest typ XML Schema o zakresie globalnym, pochodzący z bezpośredniego/pośredniego rozszerzenia *gml:AbstractGMLType*, typ właściwości
Operacje	Nie kodowane
Atrybut	Lokalny *xsd:element*, typ jest również typem właściwości (jeśli typ jest typem złożonym) lub typem prostym
Rola powiązania (asocjacji)	Lokalny *xsd:element*, typ jest zawsze typem właściwości (tylko role nazwane i nawigowalne)
Ograniczenia OCL	Nie kodowane

Reguły kodowania UML-GML
Źródło: ISO 19136:2007

Szkolenia INSPIRE dla administracji

Reguły kodowania UML-GML

- Dla różnych elementów modelu UML można określić metki, które pozwalają kontrolować przekształcanie schematu aplikacyjnego UML na XML Schema (schemat aplikacyjny GML).

Element modelu UML	Stosowana metka
Pakiet	– documentation – xsdDocument – targetNamespace (tylko dla <<Application Schema>>) – xmlns (tylko dla <<Application Schema>>) – version (tylko dla <<Application Schema>>) – gmlProfileSchema (tylko dla <<Application Schema>>)
Klasa	– documentation – noPropertyType – byValuePropertyType – isCollection – asDictionary (tylko dla <<CodeList>>) – xmlSchemaType (tylko dla <<Type>>)
Atrybut lub zakończenie powiązania	– documentation – sequenceNumber – inlineOrByReference – isMetadata

Metki dla elementów modelu UML
Źródło: ISO 19136:2007

Szkolenia INSPIRE dla administracji

Pakiet

- Przekształcany na jeden dokument XML Schema z metką „*xsdDocument*" określającą nazwę pliku XSD.
- Metka „*xsdDocument*":
  - ustawiona – dokument schematu zawiera wszystkie składniki XML Schema wynikające z klas UML zawartych bezpośrednio w pakiecie
  - nie ustawiona – wszystkie składniki schematu deklarowane są w dokumencie schematu pakietu, w którym zawarty jest ten pakiet.
- Ustawienie metki „*xsdDocument*" jest obowiązkowe dla wszystkich pakietów ze stereotypem **<<ApplicationSchema>>**.

Szkolenia INSPIRE dla administracji

Pakiet

- Dla każdego dokumentu schematu należy ustawić wartości atrybutów **targetNamespace** i **version** elementu głównego (odpowiednie metki w pakiecie schematu aplikacyjny UML).
- Należy określić atrybut **xmlns** dla docelowej przestrzeni nazw (metka „*xmlns*”).
- Należy uwzględnić wzajemne zależności między poszczególnymi pakietami:
  - zaimportowanie (**import**) i/lub
  - włączenie (**include**) schematów aplikacyjnych z innych pakietów.

Szkolenia INSPIRE dla administracji

Klasa

- Na schemat aplikacyjny GML przekształcane są jedynie klasy UML:
  - bez stereotypu,
  - ze stereotypem:
    - <<FeatureType>>,
    - <<Type>>,
    - <<DataType>>,
    - <<Union>>,
    - <<CodeList>>,
    - <<Enumeration>>.
- Wszystkie klasy UML powinny mieć zero lub jeden nadtyp.
- Każda klasa UML przekształcana jest na typ nazwany i otrzymuje przyrostek „Type”.

Szkolenia INSPIRE dla administracji

Typy danych

- Zdefiniowane przez normy ISO serii 19100 przekształcane są na odpowiadające im typy danych w XML Schema.

Klasa UML	Element obiektu GML	Typ GML	Typ właściwości GML
GM_Object	gml:AbstractGeometry	gml:AbstractGeometryType	gml:GeometryPropertyType
GM_Point	gml:Point	gml:PointType	gml:PointPropertyType
TP_Edge	gml:Edge	gml:EdgeType	gml:DirectedEdgePropertyType
SC_CRS	gml:AbstractCRS	gml:AbstractCRSType	gml:CRSPropertyType
CharacterString	–	–	xsd:string
Integer	–	–	xsd:integer, xsd:nonPositiveInteger, xsd:negativeInteger, xsd:nonNegativeInteger, xsd:positiveInteger
Length, Distance	–	–	gml:LengthType

Przykłady implementacji typów danych z norm ISO serii 19100 w schemacie aplikacyjnym GML
Źródło: ISO 19136:2007

Szkolenia INSPIRE dla administracji

Klasa <<DataType>>

- Przekształcana na typ złożony (element **complexType**) w XML Schema.
- Jeśli:
  - nie ma nadtypu – nie jest typem pochodnym w XML Schema,
  - jest nadtypem – stanowi rozszerzenie swojego nadtypu, który nie może być typem pochodnym (bezpośrednio lub pośrednio) *gml:AbstractGMLType*. Nadtypy abstrakcyjne bez atrybutów i ról nawigowalnych powiązań są pomijane.

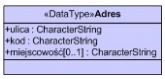
Szkolenia INSPIRE dla administracji

Klasa <<DataType>>

- Należy zdefiniować elementy globalne XML z odpowiednimi ustawieniami dla nazwy (nazwa klasy UML), typu (nazwa klasy z przyrostkiem „*Type*”), abstrakcyjności (jeśli klasa jest abstrakcyjna) i grupą zastępowania (nazwa określająca element nadtypu lub *gml:AbstractObject*, jeśli klasa nie ma nadtypu).
- Należy utworzyć nazwany typ złożony, którego nazwa zawiera nazwę klasy UML z przyrostkiem „*PropertyType*”, jeśli dla klasy nie ustawiono metki „*noPropertyType*” z wartością „*true*”.

Szkolenia INSPIRE dla administracji

Klasa <<DataType>> – przykład




```
<complexType name="AdresType">
 <sequence>
  <element name="ulica" type="string"/>
  <element name="kod" type="string"/>
  <element name="miejscowość" type="string" minOccurs="0"/>
 </sequence>
</complexType>

<element name="Adres" type="np:AdresType"
        substitutionGroup="gml:AbstractObject"/>
<complexType name="AdresPropertyType">
 <sequence>
  <element ref="np:Adres"/>
 </sequence>
 <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

Przekształcenie klasy **<<DataType>> Adres** na odpowiednią strukturę GML

Szkolenia INSPIRE dla administracji

Klasa <<FeatureType>>

- Pochodzi bezpośrednio lub pośrednio z *gml:AbstractFeatureType*.
- Jeśli:
  - nie ma nadtypu – stanowi bezpośrednie rozszerzenie *gml:AbstractFeatureType*,
  - ma nadtyp – stanowi rozszerzenie nadtypu, który powinien pochodzić od *gml:AbstractFeatureType* (bezpośrednio lub pośrednio).
- Należy zdefiniować elementy globalne XML z odpowiednimi ustawieniami dla nazwy (nazwa klasy UML), typu (nazwa klasy z przyrostkiem „*Type*"), abstrakcyjności (jeśli klasa jest abstrakcyjna) i grupą zastępowania (nazwa nadtypu lub *gml:AbstractFeature*).

Szkolenia INSPIRE dla administracji

Klasa <<FeatureType>>

- Jeśli ma pojedyncze powiązanie, które jest agregacją zwykłą lub całkowitą, rola powiązania jest przekształcana na element właściwości, a klasa przenosi metkę „*isCollection*" z wartością „*true*" oraz do typu złożonego dodawana jest grupa atrybutów *gml:AggregationAttributeGroup*.
- Należy utworzyć nazwany typ złożony, którego nazwa zawiera nazwę klasy UML z przyrostkiem „*PropertyType*", jeśli dla klasy nie ustawiono metki „*noPropertyType*" z wartością „true".
- Należy utworzyć nazwany typ złożony, którego nazwa zawiera nazwę klasy UML z przyrostkiem „*PropertyByValueType*", jeśli dla klasy ustawiono metkę „*byValuePropertyType*" z wartością „*true*".

Szkolenia INSPIRE dla administracji

Klasa <<FeatureType>> – przykład

```
<complexType name="BudynekType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="obszar" type="gml:SurfacePropertyType"/>
        <element name="adres" type="np:AdresPropertyType"/>
        <element name="rodzajBudynku" type="np:RodzajBudynkuType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="BudynekPropertyType">
  <sequence minOccurs="0">
    <element ref="np:Budynek"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>

<complexType name="BudynekPropertyByValueType">
  <sequence>
    <element ref="np:Budynek"/>
  </sequence>
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>

<element name="Budynek" type="np:BudynekType" substitutionGroup="gml:AbstractFeature"/>
```

«FeatureType»**Budynek**
+obszar : GM_Surface
+adres : Adres
+rodzajBudynku : RodzajBudynku

Przekształcenie klasy **<<FeatureType>> Budynek** na odpowiednią strukturę GML

Szkolenia INSPIRE dla administracji

Klasa <<Type>> lub bez stereotypu

- Pochodzi bezpośrednio lub pośrednio z *gml:AbstractGMLType*.
- Jeśli:
  - nie ma nadtypu – stanowi bezpośrednie rozszerzenie *gml:AbstractGMLType*,
  - ma nadtyp – stanowi rozszerzenie nadtypu, który powinien pochodzić od *gml:AbstractGMLType* (bezpośrednio lub pośrednio), ale nie z *gml:AbstractFeatureType* (bezpośrednio lub pośrednio).
- Należy zdefiniować elementy globalne XML z odpowiednimi ustawieniami dla nazwy (nazwa klasy UML), typu (nazwa klasy z przyrostkiem „*Type*”), abstrakcyjności (jeśli klasa jest abstrakcyjna) i grupą zastępowania (nazwa nadtypu lub „*AbstractGML*”).

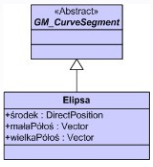
Szkolenia INSPIRE dla administracji

Klasa <<Type>> lub bez stereotypu

- Jeśli ma pojedyncze powiązanie, które jest agregacją zwykłą lub całkowitą, rola powiązania jest przekształcana na element właściwości, a klasa przenosi metkę „*isCollection*” z wartością „*true*” oraz do typu złożonego dodawana jest grupa atrybutów *gml:AggregationAttributeGroup*.
- Należy utworzyć nazwany typ złożony, którego nazwa zawiera nazwę klasy UML z przyrostkiem „*PropertyType*”, jeśli dla klasy nie ustawiono metki „*noPropertyType*” z wartością „*true*”.
- Należy utworzyć nazwany typ złożony, którego nazwa zawiera nazwę klasy UML z przyrostkiem „*PropertyByValueType*”, jeśli dla klasy ustawiono metkę „*byValuePropertyType*” z wartością „*true*”.

Szkolenia INSPIRE dla administracji

Klasa bez stereotypu – przykład



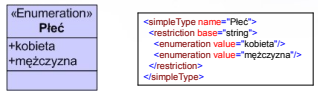
Przekształcenie klasy **Elipsa** na odpowiednią strukturę w GML
Klasa abstrakcyjna *GM_CurveSegment* pochodzi z normy ISO 19107 i reprezentuje prosty element geometryczny – segment krzywej

```
<element name="Elipsa" type="np:ElipsaType" substitutionGroup="gml:AbstractCurveSegment"/>
<complexType name="ElipsaType">
 <complexContent>
  <extension base="gml:AbstractCurveSegmentType">
   <sequence>
    <element name="środek" type="gml:DirectPositionType"/>
    <element name="małaPółoś" type="gml:VectorType"/>
    <element name="wielkaPółoś" type="gml:VectorType"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>
```

Szkolenia INSPIRE dla administracji

Klasa <<Enumeration>>

- Przekształcana na typ prosty (**simpleType**) w XML Schema.
- Typem podstawowym jest *string*, a dziedzina wartości została ograniczona do zbioru wartości określonych przez nazwy atrybutów klasy UML.




```
<simpleType name="Płeć">
  <restriction base="string">
    <enumeration value="kobieta"/>
    <enumeration value="mężczyzna"/>
  </restriction>
</simpleType>
```

Przekształcenia klasy **<<Enumeration>> Płeć** na odpowiednią strukturę GML

Szkolenia INSPIRE dla administracji

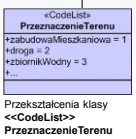
Klasa <<CodeList>>

- Bez ustawionej metki „*asDictionary*" na „*true*" powinna zostać przekształcona tak jak klasa ze stereotypem <<Enumeration>>, ale z tą różnicą, że:
  - Należy dodać wzorzec „*<pattern value='other: \w{2,}'/>*", który dopuszcza inne wartości tekstowe poza zdefiniowanymi (wartości te są poprzedzone przedrostkiem „*other:*").
  - Jeśli określony jest kod dla wartości listy kodowej, tylko kod powinien być reprezentowany jako wzorzec wyliczenia.
  - Wartość zakodowanego kodu powinna być określona za pomocą elementów annotation i appinfo z elementem *gml:description* określającym wartość tekstową wartości wyliczanej.

Szkolenia INSPIRE dla administracji

Klasa <<CodeList>> – przykład

```
<simpleType name="PrzeznaczenieTerenuType">
  <union memberTypes="np:PrzeznaczenieTerenuEnumerationType np:PrzeznaczenieTerenuOtherType"/>
</simpleType>

<simpleType name="PrzeznaczenieTerenuEnumerationType">
  <restriction base="string">
    <enumeration value="1">
      <annotation>
        <appinfo><gml:description>zabudowaMieszkaniowa</gml:description></appinfo>
      </annotation>
    </enumeration>
    <enumeration value="2">
      <annotation>
        <appinfo><gml:description>droga</gml:description></appinfo>
      </annotation>
    </enumeration>
    <enumeration value="3">
      <annotation>
        <appinfo><gml:description>zbiornikWodny</gml:description></appinfo>
      </annotation>
    </enumeration>
  </restriction>
</simpleType>

<simpleType name="PrzeznaczenieTerenuOtherType">
  <restriction base="string">
    <pattern value="other: \w{2,}"/>
  </restriction>
</simpleType>
```



Przekształcenia klasy **<<CodeList>> PrzeznaczenieTerenu** na odpowiednią strukturę w GML

Szkolenia INSPIRE dla administracji

Klasa <<Union>>

- Przekształcana na typ złożony (**complexType**) w XML Schema, podobnie jak klasa ze stereotypem <<DataType>>, przy czym zamiast elementu **sequence** pojawia się element **choice**, który oznacza, że tylko jedna z właściwości może pojawić się w instancji unii.

```
<complexType name="ArtystaType">
  <choice>
    <element name="nazwisko" type="string"/>
    <element name="pseudonim" type="string"/>
  </choice>
</complexType>
```

Szkolenia INSPIRE dla administracji

Atrybut i nazwa roli

- Przekształcane na element lokalny z tą samą nazwą, co typ złożony (**complexType**) definiujący model zawartości typu obiektu.
- Ograniczenia atrybutów **minOccurs** i **maxOccurs** są ustawiane zgodnie z ich definicjami w modelu UML.

Szkolenia INSPIRE dla administracji

Atrybut i nazwa roli

- Typ zależy od typu wartości właściwości w UML. Jeżeli typ wartości właściwości jest zawartością:
  - prostą – typ stosowany jest bezpośrednio (np. *integer*),
  - złożoną – zostanie użyta właściwość.
    - Domyślne kodowanie właściwości typu pozwala na reprezentację:
      - wbudowaną (*inline*) i przez referencję (*by-reference*) dla typów obiektów. Reprezentacja może zostać ograniczona do wbudowanej lub przez referencję poprzez zastosowanie metki „*inlineOrByReference*" z wartością „*inline*" lub „*byReference*" odpowiednio. Jeśli metka nie zostanie ustawiona lub ustawiona na „*inlineOrByReference*", należy zastosować kodowanie domyślne.
      - wbudowaną (*inline*) dla typów danych i unii.

Szkolenia INSPIRE dla administracji

Atrybut i nazwa roli

- Jeśli typ właściwości został już określony w schemacie aplikacyjnym jako typ nazwany (należy sprawdzić metki „*noPropertyType*” i „*byValuePropertyType*”), należy odwołać się do tego składnika schematu, w przeciwnym razie lokalnie zostanie zdefiniowany anonimowy typ właściwości w elemencie właściwości.
- Jeżeli kodowana właściwość jest końcem powiązania i drugi koniec powiązania jest również kodowany na schemat aplikacyjny GML, nazwa właściwości drugiego końca powiązania powinna zostać przekształcona na element *gml:reversePropertyName* w elementach **annotation** i **appinfo** elementu właściwości.

Szkolenia INSPIRE dla administracji

Atrybut i nazwa roli – przykład

```
<element name="właściciel" type="np:OsobaPropertyType" maxOccurs="unbounded">
  <annotation>
    <appinfo>
      <gml:reversePropertyName>np:własność</gml:reversePropertyName>
    </appinfo>
  </annotation>
</element>
...
<complexType name="OsobaPropertyType">
  <sequence minOccurs="0">
    <element ref="np:Osoba"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup" />
</complexType>
```

Dwa alternatywne przekształcenia właściwości typu (roli w powiązaniu) na reprezentację wbudowaną (*inline*) i przez referencję (*by-reference*) w GML

```
<element name="właściciel" maxOccurs="unbounded">
  <annotation>
    <appinfo>
      <gml:reversePropertyName>np:własność</gml:reversePropertyName>
    </appinfo>
  </annotation>
  <complexType>
    <sequence minOccurs="0">
      <element ref="np:Osoba"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
</element>
```

Szkolenia INSPIRE dla administracji

Atrybut i nazwa roli – przykład

– Alternatywnie typ właściwości może posiadać tylko jedną z reprezentacji, wbudowaną lub przez referencję, w zależności od wartości metki „*inlineOrByReference*”.

```
<element name="właściciel" type="np:OsobaPropertyByValueType" maxOccurs="unbounded"/>
...
<complexType name="OsobaPropertyByValueType">
  <sequence>
    <element ref="np:Osoba"/>
  </sequence>
</complexType>
```

Dwa alternatywne przekształcenia właściwości typu (roli w powiązaniu) tylko na reprezentację wbudowaną (*inline*) w GML

```
<element name="właściciel" maxOccurs="unbounded">
  <complexType>
    <sequence>
      <element ref="np:Osoba"/>
    </sequence>
  </complexType>
</element>
```

Szkolenia INSPIRE dla administracji

Atrybut i nazwa roli – przykład

- Jeśli dopuszczalna jest tylko reprezentacja przez referencję – element właściwości należy ograniczyć za pomocą znaczników **annotation** i **appinfo** elementem *gml:targetElement*, który określa nazwę elementu typu docelowego:

```
<element name="elementDocelowy" type="string"/>
```

- Jeśli kodowana właściwość jest końcem powiązania i drugi koniec powiązania jest również kodowany na schemat aplikacyjny GML – nazwę właściwości drugiego końca powiązania należy zapisać za pomocą znaczników **annotation** i **appinfo** oraz elementu *gml:reversePropertyName*.

```
<element name="właściciel" type="gml:ReferenceType" maxOccurs="unbounded">
 <annotation>
  <appinfo>
   <gml:targetElement>np:Osoba</gml:targetElement>
   <gml:reversePropertyName>np:własność</gml:reversePropertyName>
  </appinfo>
 </annotation>
</element>
```

Przekształcenie właściwości typu (roli w powiązaniu) tylko na reprezentację przez referencję (*by-reference*) w GML

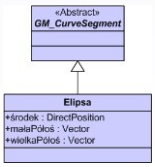
Szkolenia INSPIRE dla administracji

Dziedziczenie

- XML Schema obsługuje tylko **dziedziczenie pojedyncze**.
- Realizowane poprzez:
  - **mechanizm rozszerzenia** (*extension*) lub
  - **mechanizm ograniczenia** (*restriction*) oraz
  - wykorzystanie **elementu zastępowania** (*substitutionGroup*).

Szkolenia INSPIRE dla administracji

Dziedziczenie – przykład

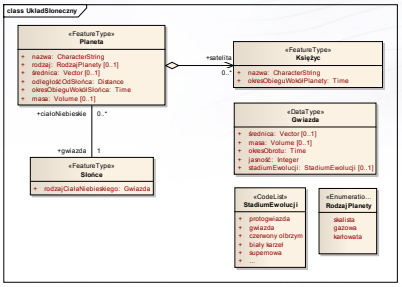


Przekształcenie klasy **Elipsa**, która dziedziczy z klasy abstrakcyjnej ***GM_CurveSegment***, na odpowiednią strukturę w GML

```
<element name="Elipsa" type="np:ElipsaType" substitutionGroup="gml:AbstractCurveSegment"/>
<complexType name="ElipsaType">
 <complexContent>
  <extension base="gml:AbstractCurveSegmentType">
   <sequence>
    <element name="środek" type="gml:DirectPositionType"/>
    <element name="małaPółoś" type="gml:VectorType"/>
    <element name="wielkaPółoś" type="gml:VectorType"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>
```

Szkolenia INSPIRE dla administracji

Przekształcenie UML-GML – przykład



Schemat aplikacyjny w UML

Szkolenia INSPIRE dla administracji

Przekształcenie UML-GML – przykład

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:us="http://www.astronomia.pl/uklad_sloneczny"
xmlns:gml="http://www.opengis.net/gml/3.2" targetNamespace="http://www.astronomia.pl/uklad_sloneczny"
elementFormDefault="qualified" version="1.0">
  <!-- ========================== -->
  <import namespace="http://www.opengis.net/gml/3.2"
schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <!-- ========================== -->
  <element name="US_Planeta" substitutionGroup="gml:AbstractFeature">
    <complexType>
      <complexContent>
        <extension base="gml:AbstractFeatureType">
          <sequence>
            <element name="nazwa" type="string"/>
            <element name="rodzaj" type="us:US_RodzajPlanetyType" minOccurs="0"/>
            <element name="średnica" type="gml:VectorType" minOccurs="0" maxOccurs="unbounded"/>
            <element name="odległośćOdSłońca" type="gml:LengthType"/>
            <element name="okresObieguWokółSłońca" type="gml:TimeType"/>
            <element name="masa" type="gml:VolumeType" minOccurs="0"/>
            <element name="gwiazda" type="us:US_SłońcePropertyType">
              <annotation>
                <appinfo>
                  <gml:reverseProperty>us:cialoNiebieskie</gml:reverseProperty>
                </appinfo>
              </annotation>
            </element>
            <element name="satelita" type="us:US_KsiężycPropertyType" minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
```

Schemat aplikacyjny w GML – fragment 1

Szkolenia INSPIRE dla administracji

Przekształcenie UML-GML – przykład

```
  <element name="US_Księżyc" substitutionGroup="gml:AbstractFeature">
    <complexType>
      <complexContent>
        <extension base="gml:AbstractFeatureType">
          <sequence>
            <element name="nazwa" type="string"/>
            <element name="okresObieguWokółPlanety" type="gml:TimeType"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <complexType name="US_KsiężycPropertyType">
    <sequence minOccurs="0">
      <element ref="us:US_Księżyc"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
  </complexType>
```

Schemat aplikacyjny w GML – fragment 2

Szkolenia INSPIRE dla administracji

Przekształcenie UML-GML – przykład

```
<element name="US_Słońce" substitutionGroup="gml:AbstractFeature">
 <complexType>
  <complexContent>
   <extension base="gml:AbstractFeatureType">
    <sequence>
     <element name="rodzajCiałaNiebieskiego">
      <complexType>
       <sequence>
        <element name="US_Gwiazda" type="us:US_GwiazdaPropertyType"/>
       </sequence>
      </complexType>
     </element>
     <element name="ciałoNiebieskie" type="us:US_PlanetaPropertyType" minOccurs="0" maxOccurs="unbounded">
      <annotation>
       <appinfo>
        <gml:reverseProperty>us:gwiazda</gml:reverseProperty>
       </appinfo>
      </annotation>
     </element>
    </sequence>
   </extension>
  </complexContent>
 </complexType>
</element>
<complexType name="US_SłocePropertyType">
 <sequence minOccurs="0">
  <element ref="us:US_Słońce"/>
 </sequence>
 <attributeGroup ref="gml:AssociationAttributeGroup"/>
```

Schemat aplikacyjny w GML – fragment 3

Szkolenia INSPIRE dla administracji

Przekształcenie UML-GML – przykład

```
<element name="US_Gwiazda" type="us:US_GwiazdaType">
<complexType name="US_GwiazdaType">
 <sequence>
  <element name="średnica" type="gml:VectorType" minOccurs="0"/>
  <element name="masa" type="gml:VolumeType" minOccurs="0"/>
  <element name="okresObiegu" type="gml:TimeType"/>
  <element name="jasność" type="integer"/>
  <element name="stadiumEwolucji" type="us:US_StadiumEwolucjiType" default="gwiazda" minOccurs="0"/>
 </sequence>
</complexType>
<complexType ref="us:US_GwiazdaPropertType">
 <sequence minOccurs="0">
  <element ref="us:US_Gwiazda"/>
 </sequence>
 <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
```

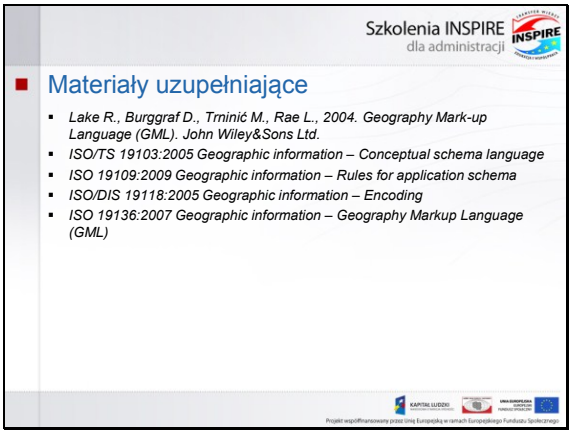
Schemat aplikacyjny w GML – fragment 4

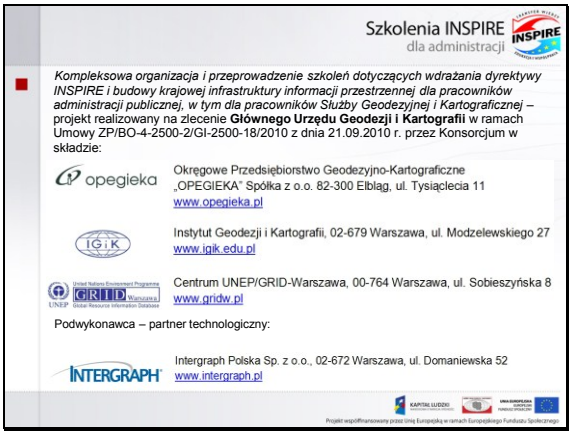
Szkolenia INSPIRE dla administracji

Przekształcenie UML-GML – przykład

```
<simpleType name="US_RodzajPlanetyType">
  <restriction base="string">
   <enumeration value="skalista"/>
   <enumeration value="gazowa"/>
   <enumeration value="karłowata"/>
  </restriction>
</simpleType>
<!-- ================================ -->
<simpleType name="US_StadiumEwolucjiType">
 <union memberTypes="us:US_StadiumEwolucjiEnumerationType us:US_StadiumEwolucjiOtherType"/>
</simpleType>
<simpleType name="US_StadiumEwolucjiEnumerationType">
 <restriction base="string">
  <enumeration value="protogwiazda"/>
  <enumeration value="gwiazda"/>
  <enumeration value="czerwony olbrzym"/>
  <enumeration value="biały karzeł"/>
  <enumeration value="supernowa"/>
  <enumeration value="..."/>
 </restriction>
</simpleType>
<simpleType name="US_StadiumEwolucjiOtherType">
 <restriction base="string">
  <pattern value="other: \w{2,}"/>
 </restriction>
</simpleType>
</schema>
```

Schemat aplikacyjny w GML – fragment 5


Szkolenia INSPIRE
dla administracji
Materiały uzupełniające
Lake R., Burggraf D., Trninić M., Rae L., 2004. Geography Mark-up Language (GML). John Wiley&Sons Ltd.
ISO/TS 19103:2005 Geographic information – Conceptual schema language
ISO 19109:2009 Geographic information – Rules for application schema
ISO/DIS 19118:2005 Geographic information – Encoding
ISO 19136:2007 Geographic information – Geography Markup Language (GML)


Szkolenia INSPIRE
dla administracji
Kompleksowa organizacja i przeprowadzenie szkoleń dotyczących wdrażania dyrektywy INSPIRE i budowy krajowej infrastruktury informacji przestrzennej dla pracowników administracji publicznej, w tym dla pracowników Służby Geodezyjnej i Kartograficznej – projekt realizowany na zlecenie Głównego Urzędu Geodezji i Kartografii w ramach Umowy ZP/BO-4-2500-2/GI-2500-18/2010 z dnia 21.09.2010 r. przez Konsorcjum w składzie:
opegieka
Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne „OPEGIEKA" Spółka z o.o. 82-300 Elbląg, ul. Tysiąclecia 11
www.opegieka.pl
IGiK
Instytut Geodezji i Kartografii, 02-679 Warszawa, ul. Modzelewskiego 27
www.igik.edu.pl
GRID
Centrum UNEP/GRID-Warszawa, 00-764 Warszawa, ul. Sobieszyńska 8
www.gridw.pl
Podwykonawca – partner technologiczny:
INTERGRAPH
Intergraph Polska Sp. z o.o., 02-672 Warszawa, ul. Domaniewska 52
www.intergraph.pl

Notatki