



WYKORZYSTANIE JĘZYKA UML W INSPIRE (WARSZTATY)

Agnieszka Chojka
Wrocławski Instytut Zastosowań Informatyki
Przestrzennej i Sztucznej inteligencji

Warszawa, październik-listopad 2017





PROGRAM SZKOLENIA

- WPROWADZENIE/PRZYPOMNIENIE
 - INSPIRE i SDI
 - Interoperacyjność
 - Schematy aplikacyjne i ich rola w SDI
- PRZEGLĄD DIAGRAMÓW UML
- REGUŁY BUDOWY SCHEMATÓW APLIKACYJNYCH UML
 - Podstawy notacji UML
 - Mechanizmy rozszerzenia
 - Etapy budowy schematu aplikacyjnego UML
 - Stereotypy stosowane w INSPIRE
- PODSUMOWANIE



WROCLAW GLOWNY

budimex
50

WPROWADZENIE PRZYPOMNIENIE



- **IN**frastructure for **SP**atial **InfoR**mation in **E**urope
 - dyrektywa 2007/2/WE Parlamentu Europejskiego i Rady z dnia 14 marca 2007 r. *ustanawiająca infrastrukturę informacji przestrzennej we Wspólnocie Europejskiej*
- **Cel**
 - utworzenie **Europejskiej Infrastruktury Danych Przestrzennych** (ESDI, ang. *European Spatial Data Infrastructure*)
- **Beneficjenci**
 - instytucje zarządzające (administracja publiczna)
 - ustawodawcy
 - obywatele i ich organizacje na poziomie krajowym i międzynarodowym



INFRASTRUKTURA DANYCH PRZESTRZENNYCH

- Zespół środków

- prawnych
- organizacyjnych
- ekonomicznych
- technicznych

które zapewniając powszechny dostęp do danych i usług geoinformacyjnych, dotyczących określonego obszaru, przyczyniają się do

- efektywnego stosowania geoinformacji dla zrównoważonego rozwoju tego obszaru
- umożliwiają racjonalne gospodarowanie zasobami geoinformacyjnymi

[wg Gaździcki, Internetowy leksykon geomatyczny]



INFRASTRUKTURA DANYCH PRZESTRZENNYCH

- Obejmuje
 - powiązane ze sobą, zdolne do współdziałania systemy i bazy danych przestrzennych zawierające dane i metadane o odpowiedniej treści i jakości
 - technologie teleinformatyczne i geoinformacyjne stosujące powszechnie akceptowane standardy
 - przepisy prawne, struktury organizacyjne, rozwiązania ekonomiczne i zasoby ludzkie
 - użytkowników tworzących społeczeństwo geoinformacyjne

[wg Gaździcki, Internetowy leksykon geomatyczny]



INFRASTRUKTURA DANYCH PRZESTRZENNYCH

- **Perspektywa danych** (ang. *data-centric view*)
 - schematy aplikacyjne
 - metadane

- **Perspektywa usług** (ang. *service-centric view*)
 - interoperacyjność
 - architektura zorientowana na usługi



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

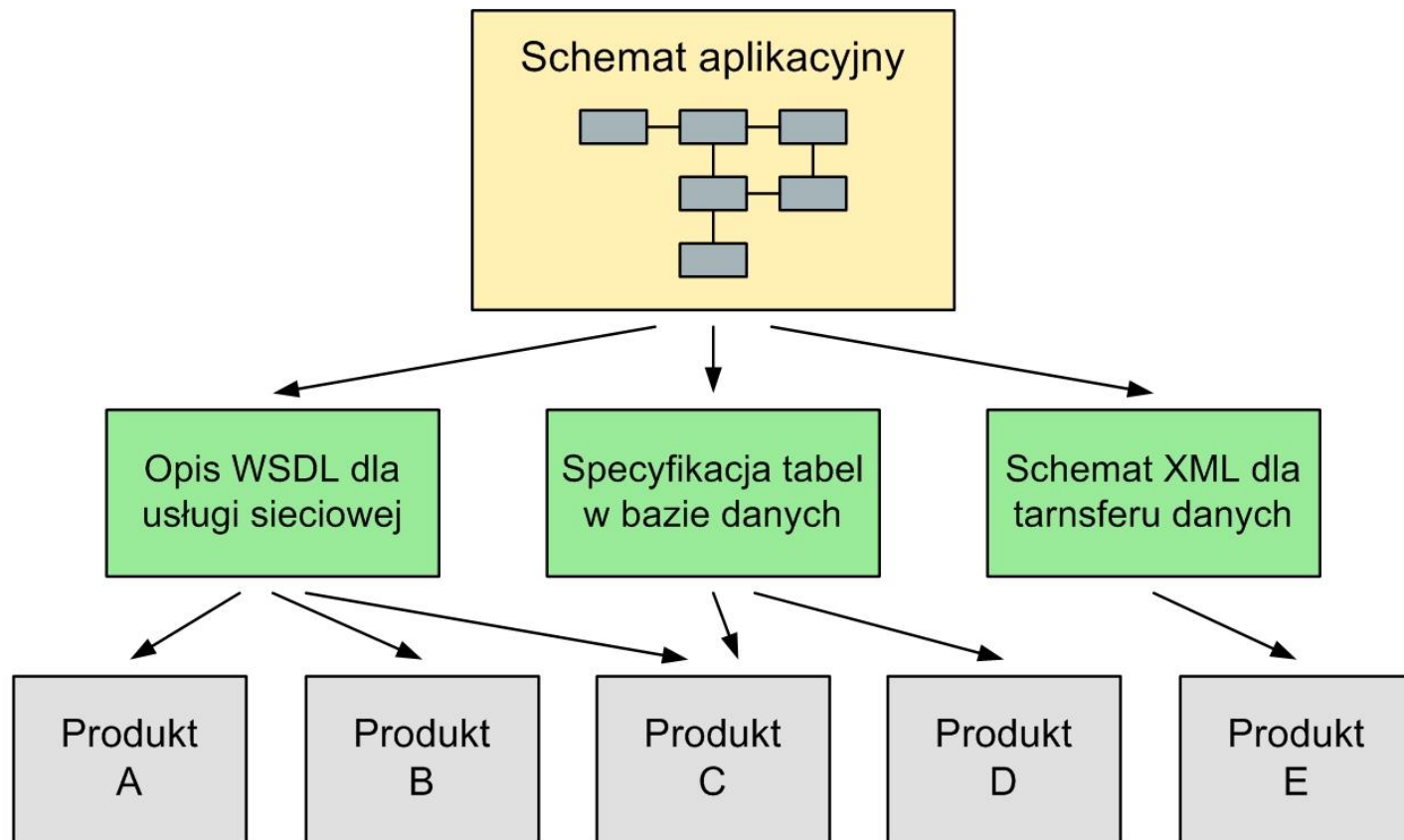


PODEJŚCIE OPARTE NA MODELU

niezależny od implementacji opis semantyki i logiczna struktura danych

specyfikacje dla różnych technik

implementacje



[wg CEN/TC 287, 2012. CEN/TR 15449-3 Geographic information – Spatial data infrastructures – Part 3: Data centric view.]



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)

 **INTEROPERACYJNOŚĆ**
(ang. *interoperability*)



INTEROPERACYJNOŚĆ

- **Zdolność do współdziałania**
 - m.in. zdolność do komunikowania ► dokonywania **bezstratnej wymiany danych** między różnymi systemami
- **Cel**
 - **norm ISO serii 19100**
 - zdolność do zapewnienia udziału w IG zróżnicowanych środowisk narzędziowych, przedmiotowych, instytucjonalnych i innych
 - **INSPIRE**
 - możliwość łączenia zbiorów danych przestrzennych oraz interakcji usług danych przestrzennych bez powtarzalnej interwencji manualnej, w taki sposób, aby wynik był spójny, a wartość dodana zbiorów i usług danych przestrzennych została zwiększona



INTEROPERACYJNOŚĆ

- **Techniczna**

- aspekt systemowy (np. urządzenia, protokoły komunikacji, systemy operacyjne)
- aspekt syntaktyczny (np. formaty danych, języki reprezentacji)
- architektura zorientowana na usługi (SOA), wykorzystująca technologię Web Services (usługi sieciowe)

- **Semantyczna**

- właściwe, jednoznaczne rozumienie wymienianej i upowszechnianej informacji (np. tezaury)

- **Organizacyjna**

- przyzwolenia na interoperacyjność (np. przepisy prawne, struktury i procedury organizacyjne, czynniki ekonomiczne i kadrowe)
[wg Gaździcki, Internetowy leksykon geomatyczny]



INTEROPERACYJNOŚĆ

- Jak osiągnąć? (1)
 - metodologia modelowania informacji geograficznej za pomocą schematów pojęciowych
 - podstawa dla spójnych realizacji GIS w odmiennych środowiskach sprzętowo-programowych
 - zapewnia współdziałanie różnych realizacji GIS
 - języki modelowania informacji geograficznej
 - UML (ISO 19109, ISO 19103)
 - XML (ISO 19118)
 - GML (ISO 19136)



INTEROPERACYJNOŚĆ

- **Jak osiągnąć? (2)**
 - harmonizacja
 - baz danych przestrzennych
 - doprowadzenie do interoperacyjności w zakresie baz danych przestrzennych
 - danych przestrzennych (INSPIRE)
 - zapewnienie dostępu do danych przestrzennych w reprezentacjach, które umożliwiają łączenie tych danych w sposób spójny z innymi zharmonizowanymi danymi, korzystając z usług sieciowych oraz stosując wspólne specyfikacje produktów danych
 - zbiorów danych (ustawa o IIP)
 - działania o charakterze prawnym, technicznym i organizacyjnym, mające na celu doprowadzenie do wzajemnej spójności tych zbiorów oraz ich przystosowanie do wspólnego i łącznego wykorzystywania



INTEROPERACYJNOŚĆ

- Jak osiągnąć? (3)
 - SOA (ang. *Service Oriented Architecture*)
 - architektura zorientowana na usługi
 - architektura systemów informatycznych umożliwiająca udostępnianie usług w ramach organizacji
 - wykorzystuje m.in. technologię **Web Services** (usługi sieciowe), która pozwala na automatyczny dostęp do zasobów poprzez Internet
 - rozwiązanie sprawdzające się w przypadku dużych i złożonych organizacji, których działanie opiera się na różnych systemach o złożonych architekturach (różni dostawcy aplikacji)

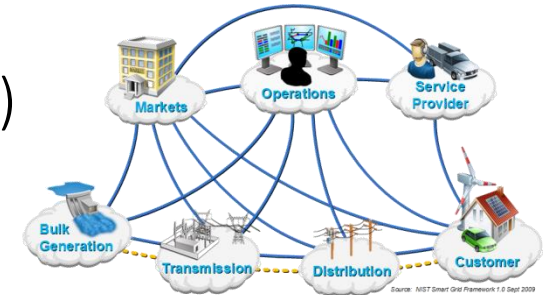


TERMINOLOGIA

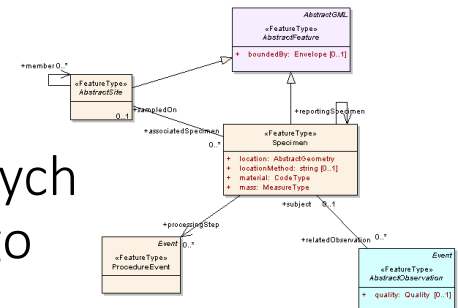
- **Model pojęciowy** (ang. *conceptual model*)
 - model definiujący pojęcia z pewnej przestrzeni rozważań (przedmiotu zainteresowań)



- **Schemat pojęciowy** (ang. *conceptual schema*)
 - formalny opis modelu pojęciowego w określonym języku schematu pojęciowego

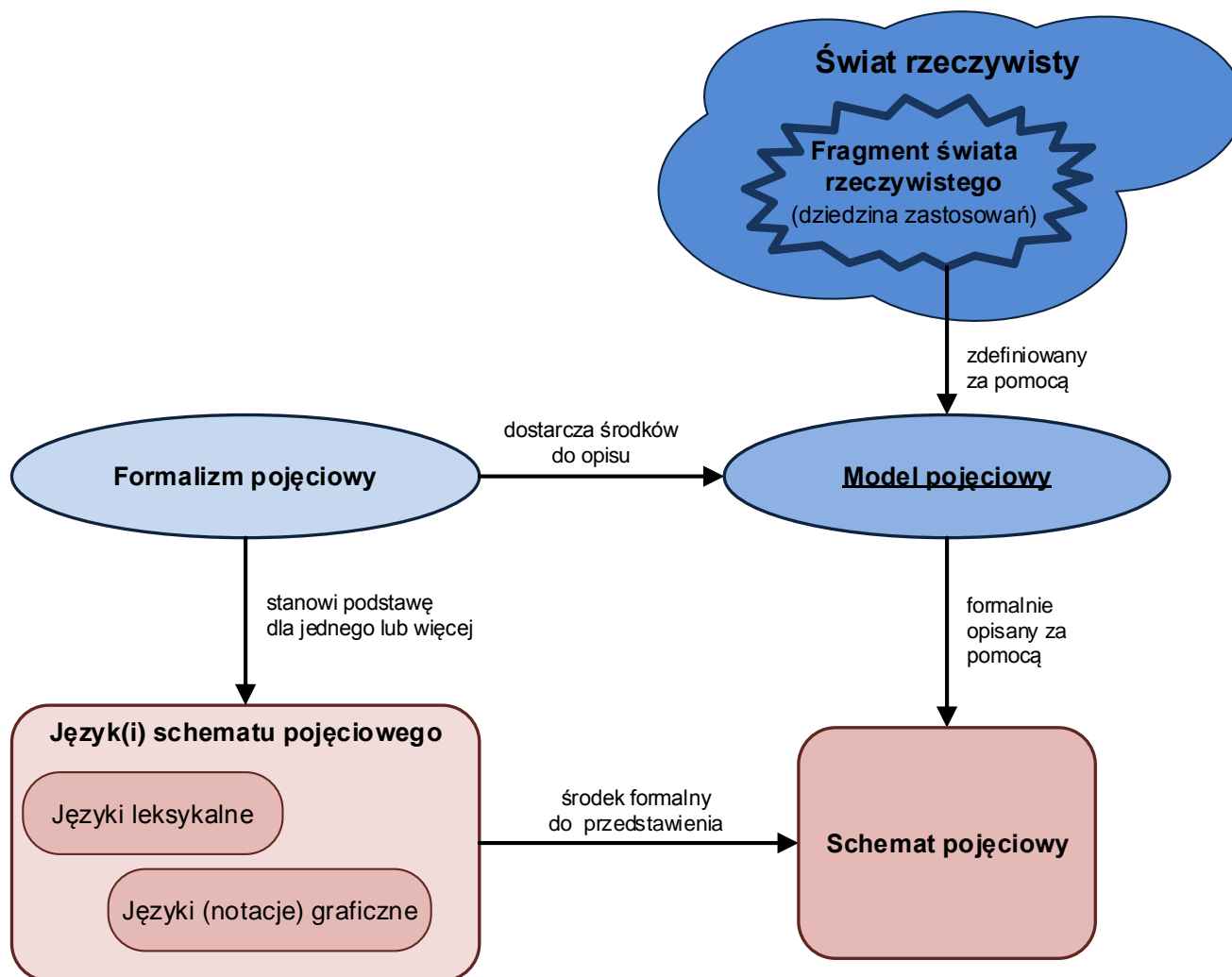


- **Schemat aplikacyjny** (ang. *application schema*)
 - schemat pojęciowy dla danych wykorzystywanych przez jedną lub więcej aplikacji, dla określonego zakresu przedmiotowego



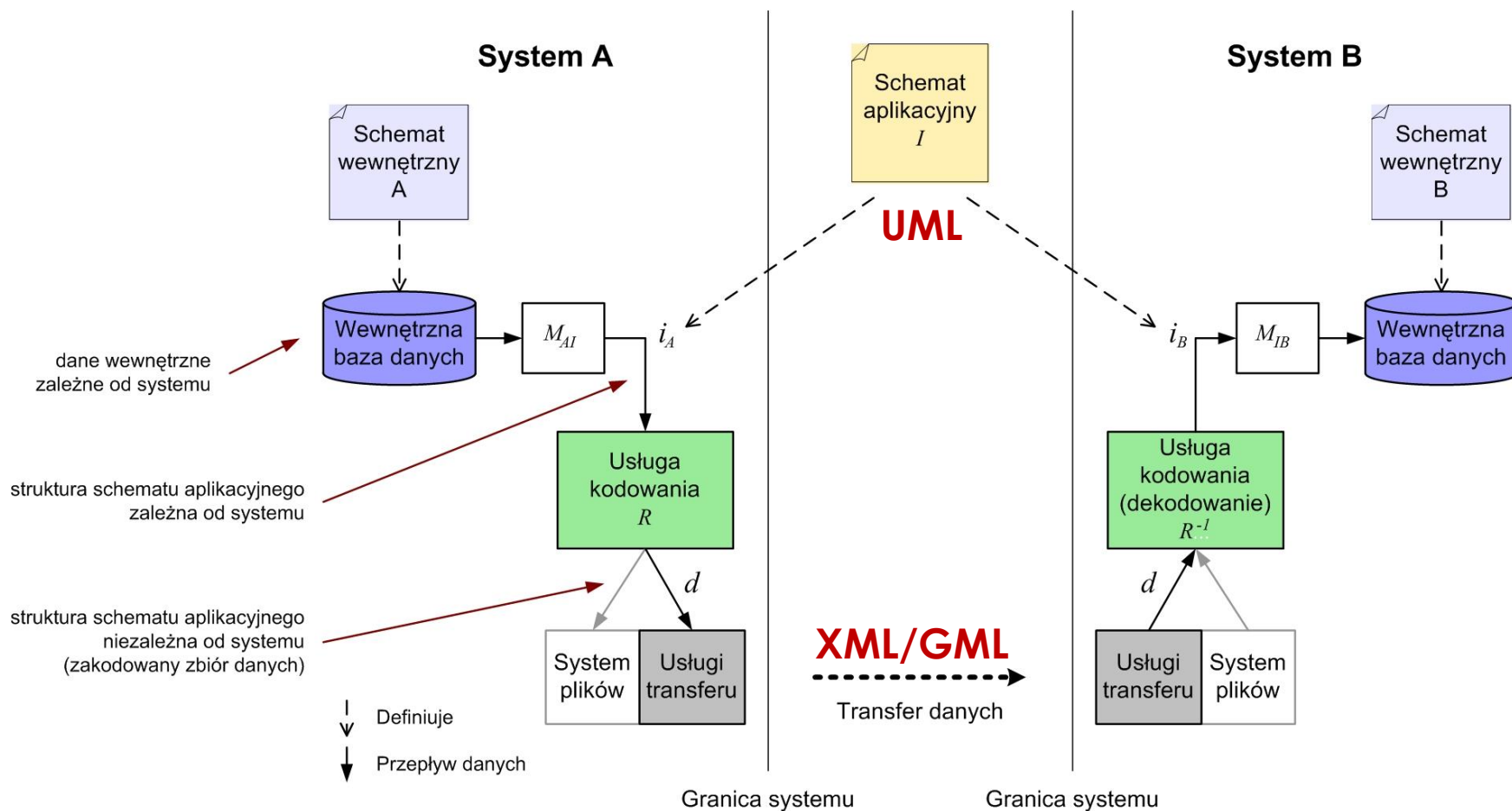


MODELOWANIE POJĘCIOWE





INTEROPERACYJNA WYMIANA DANYCH



[wg ISO/TC 211, 2011. ISO 19118 Geographic information – Encoding.]



UML

- ang. *Unified Modeling Language*
- zunifikowany (ujednolicony) graficzny język modelowania
- **Informatyka**
 - opis świata obiektów w analizie obiektowej i programowaniu obiektowym
 - wymiana informacji o systemach i oprogramowaniu za pomocą diagramów oraz uzupełniającego je tekstu
 - m.in. definiowanie wymagań, projektowanie architektury systemu, modelowanie struktury danych, modelowanie zachowania (działania) systemu



- **Geoinformatyka**

- środek formalny modelowania informacji geograficznej
- opis świata obiektów rzeczywistych
- zalecany przez normy ISO serii 19100, jako język schematu pojęciowego
- wykorzystuje się przede wszystkim możliwości modelowania obiektowego oferowane przez diagramy klas i pakietów
 - w modelach uwzględnia się głównie klasy z atrybutami, bez metod



SCHEMAT APLIKACYJNY UML

- Model definiujący pojęcia z pewnej dziedziny (przestrzeni rozważań, przedmiotu zainteresowań)
- Zapisany w języku schematu pojęciowego UML
 - według zasad określonych w standardach ISO 19109 oraz ISO 19103
 - za pomocą oznaczeń klas i powiązań między nimi (diagram klas), właściwych dla języka UML
- Opis struktur logicznych danych przestrzennych oraz opis semantyki ich zawartości
- Opis niezależny od platformy sprzętowo-programowej



GML

- ang. *Geography Markup Language*
- język znaczników geograficznych?!
- geograficzny język znaczników?!

- język znaczników przeznaczony do opisu/zapisu danych geograficznych (przestrzennych)



- Geoinformatyka
 - język formalny do opisu struktur danych
 - zalecany przez normy ISO serii 19100
 - wykorzystuje gramatykę języka **XML** (ang. *eXtensible Markup Language*)
 - umożliwia zapis w języku **XML Schema** określonych właściwości przestrzennych i nieprzestrzennych (zdefiniowanych w normach ISO serii 19100) obiektów geograficznych
 - otwarty **format wymiany** danych przestrzennych (wektorowych i opisowych) pomiędzy różnymi systemami geoinformacyjnym (GIS)
 - m.in. np. poprzez wykorzystanie jednej z usług geoinformacyjnych – usługi WFS, która na żądanie dostarcza dane zakodowane w tym formacie



SCHEMAT APLIKACYJNY GML

- Schemat aplikacyjny zapisany w języku **XML Schema**, zgodnie z regułami określonymi w ISO 19136



PRZEGLĄD DIAGRAMÓW UMI



DIAGRAMY W UML

- W UML istnieje szereg diagramów
 - każdy diagram UML pokazuje projektowany system z innej perspektywy
 - w procesie budowy systemu bierze udział wielu uczestników (klient, analityk, projektant, programista), a każdy z nich zainteresowany jest innym aspektem tworzonego systemu
- Diagram
 - graf, którego wierzchołkami są elementy, a krawędziami związki



DIAGRAMY W UML

RODZAJE

- **Diagramy statyczne (modelowanie strukturalne)**
 - diagram pakietów (ang. *package diagram*)
 - diagram klas (ang. *class diagram*)
 - diagram obiektów (ang. *object diagram*)
 - diagram komponentów/składników (ang. *component diagram*)
 - diagram wdrożenia (ang. *deployment diagram*)
 - diagram struktur złożonych (ang. *composite structure diagram*)
- **Diagramy dynamiczne (modelowanie behawioralne)**
 - diagram przypadków użycia (ang. *use case diagram*)
 - diagram czynności/aktywności (ang. *activity diagram*)
 - diagram maszyny stanowej (ang. *state machine diagram*)
 - diagram czasowy/przebiegów czasowych (ang. *timing diagram*)
 - diagram komunikacji (ang. *communication diagram*)
 - diagram przebiegu/sekwencji (ang. *sequence diagram*)
 - diagram przeglądu interakcji (ang. *interaction overview diagram*)

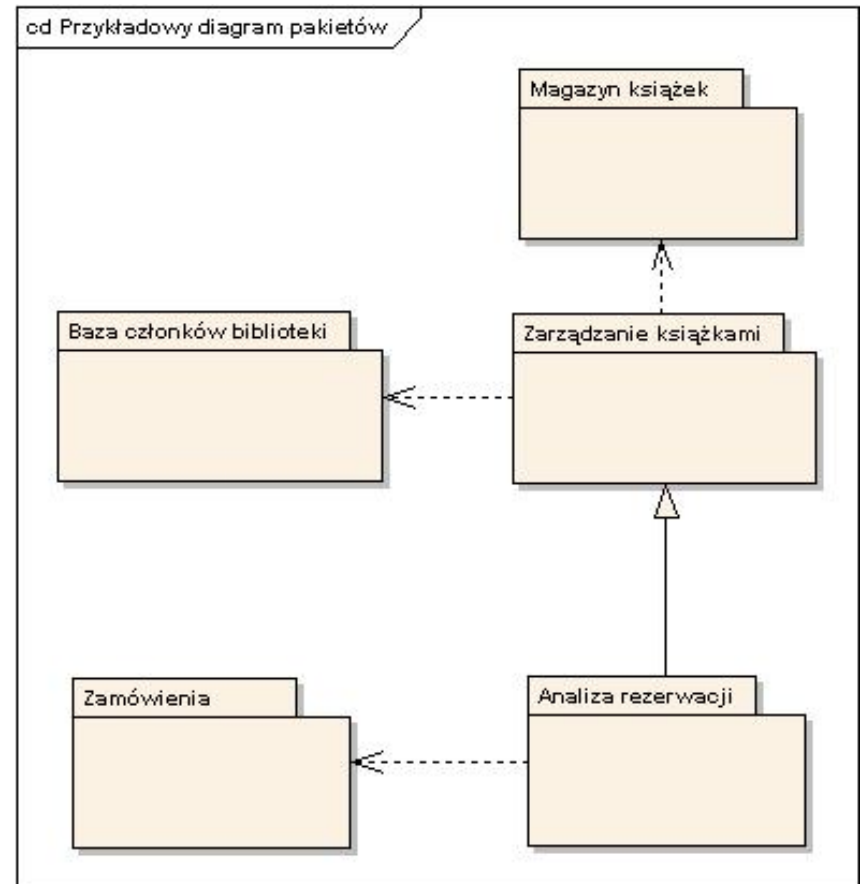
diagram
interakcji



DIAGRAMY W UML STATYCZNE

• Diagram pakietów

- porządkuje strukturę zależności w systemie, który ma bardzo wiele klas, przypadków użycia itp.
- podział systemu z logicznego punktu widzenia

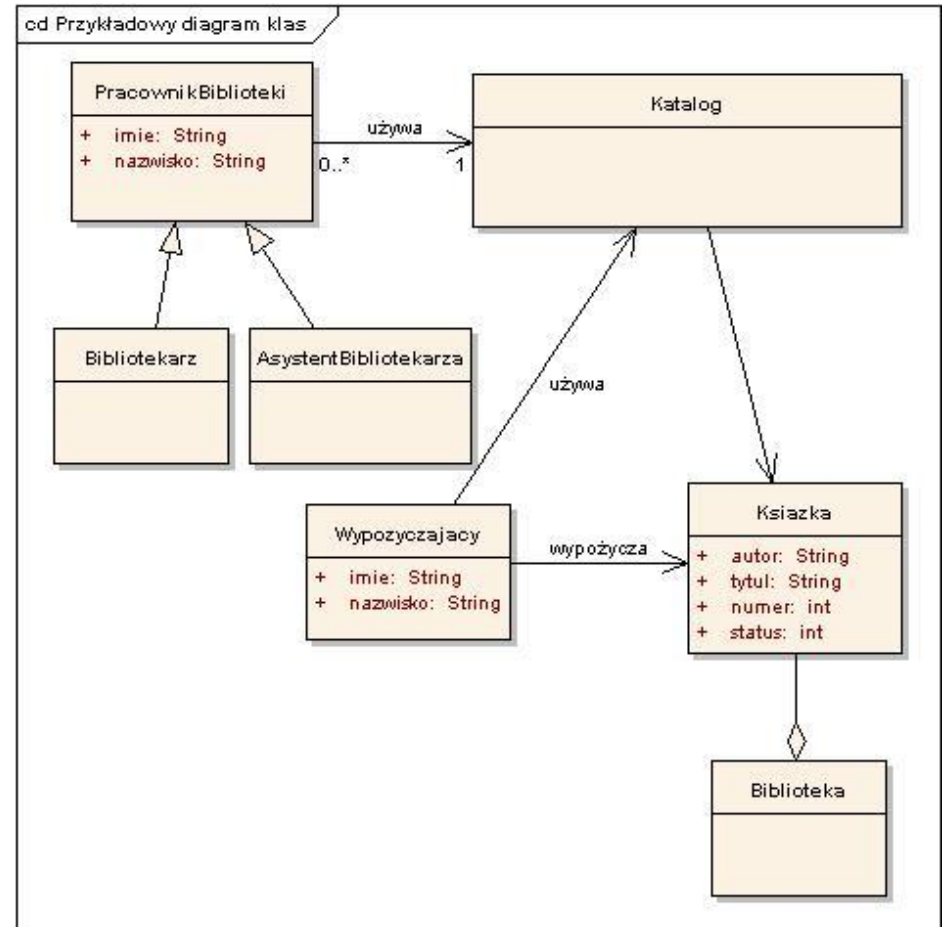


<http://www.erudis.pl/>



DIAGRAMY W UML STATYCZNE

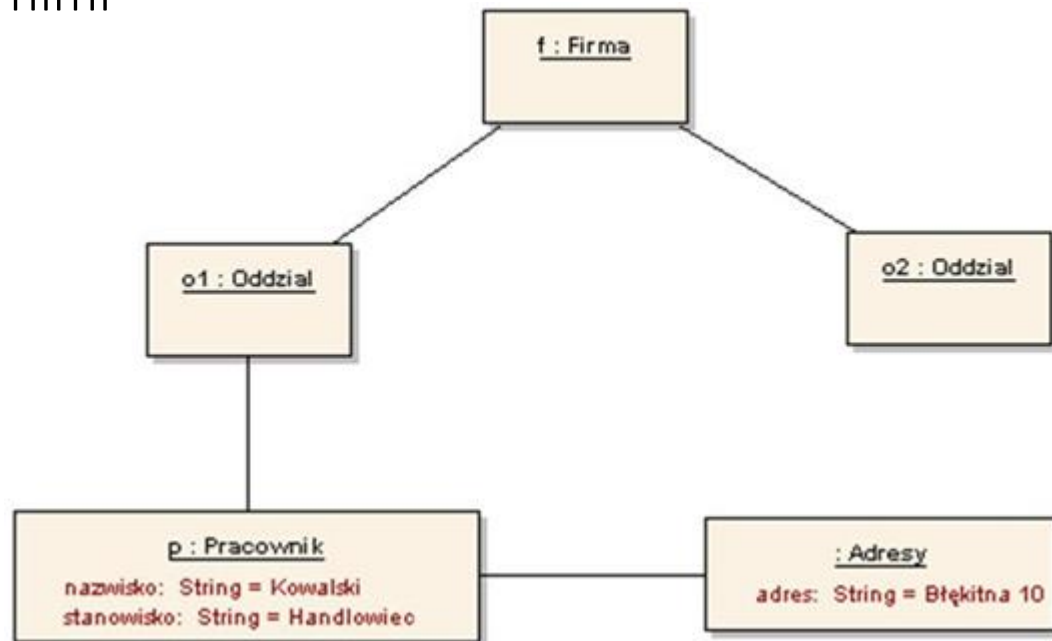
- Diagram klas
 - złożony z klas i związków między nimi





DIAGRAMY W UML STATYCZNE

- Diagram obiektów
 - przedstawia obiekty i związki między nimi

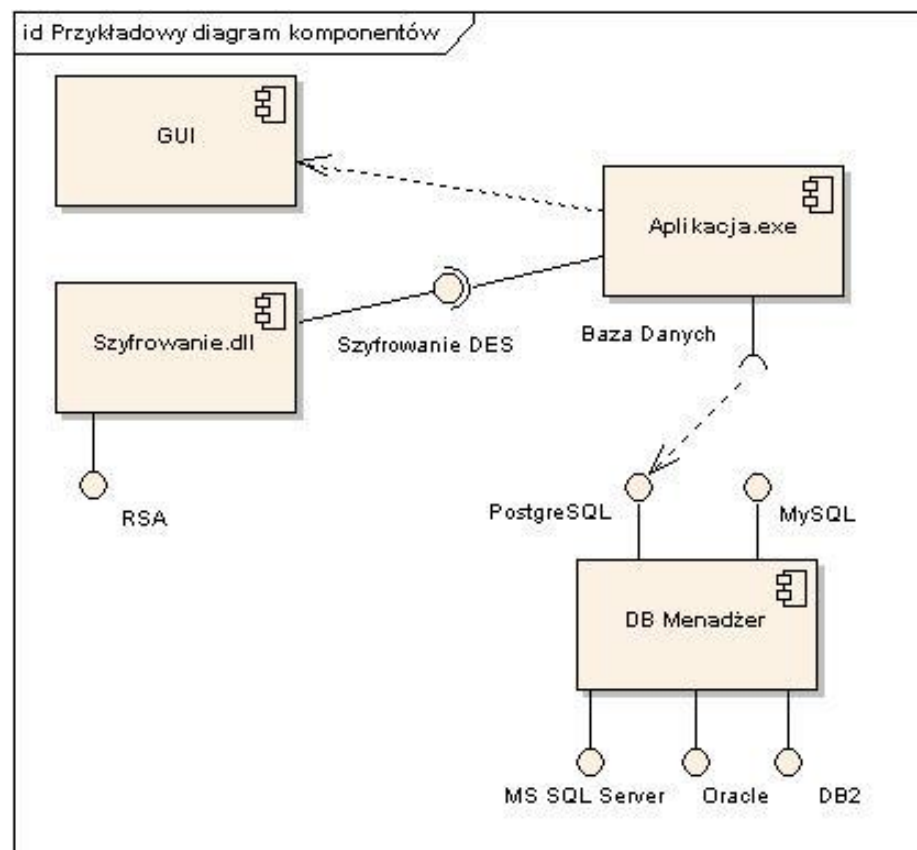




DIAGRAMY W UML STATYCZNE

- Diagram komponentów

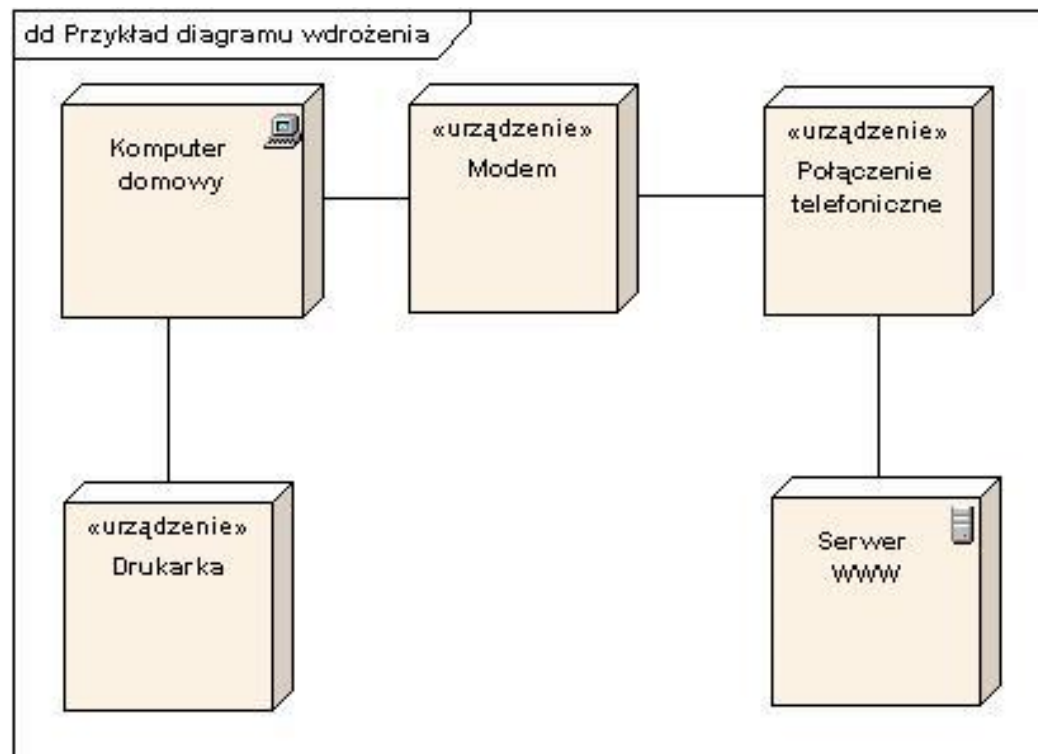
- podział systemu na fizyczne elementy oprogramowania
 - pliki, biblioteki, aplikacje itp.





DIAGRAMY W UML STATYCZNE

- Diagram wdrożenia
 - obrazuje wdrożenie i konfigurację zaprojektowanego systemu

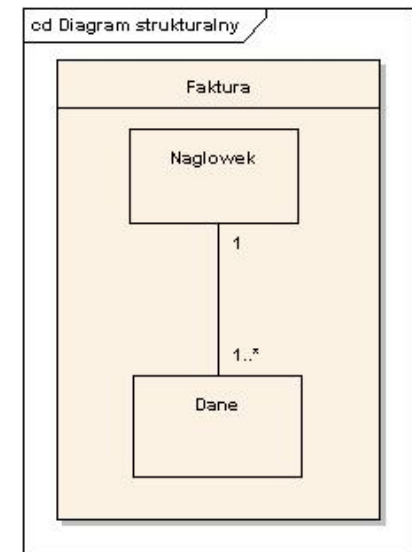
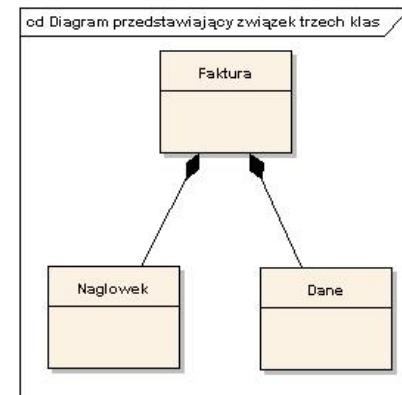
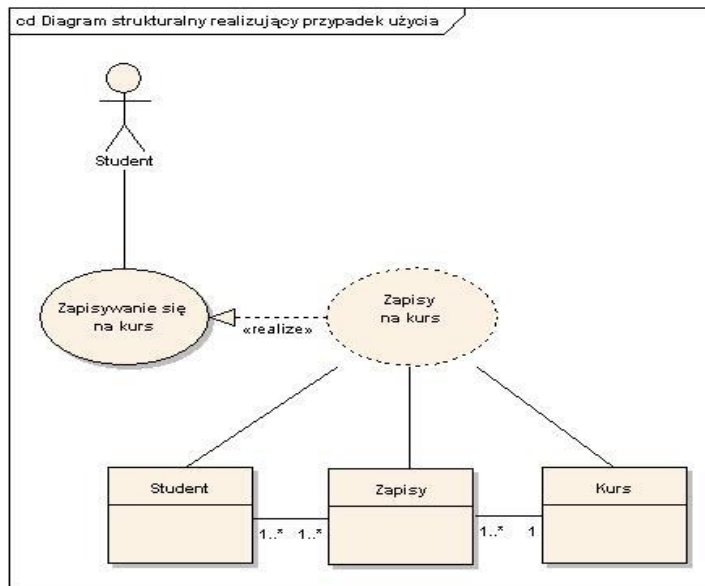




DIAGRAMY W UML STATYCZNE

- Diagram struktur złożonych

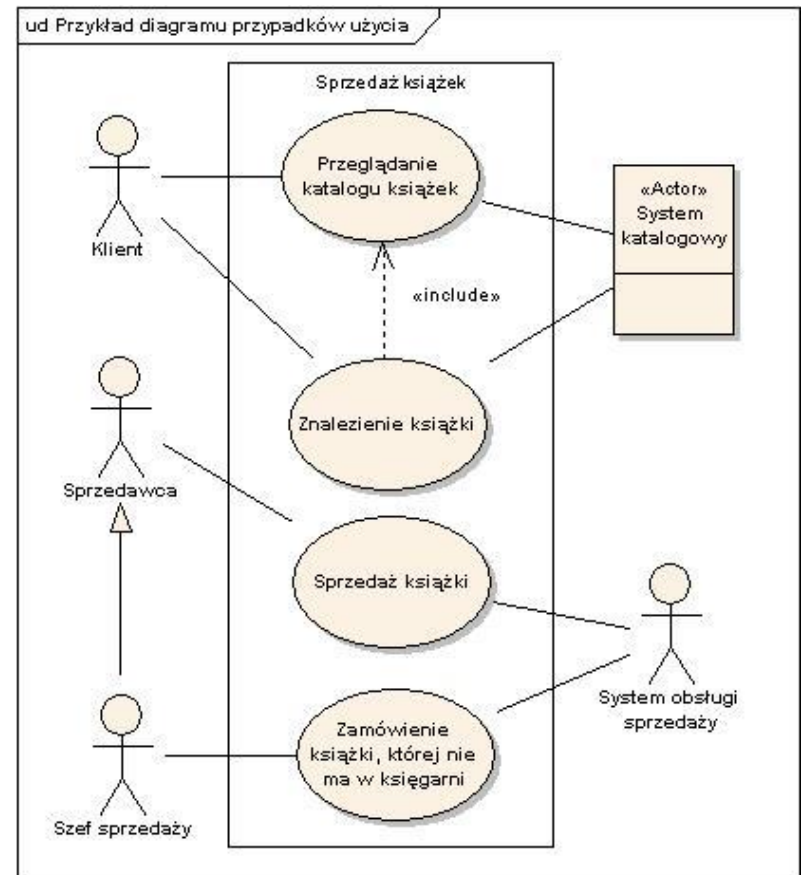
- obrazuje elementy systemu wykonujące wspólne zadanie, typowe sposoby użycia elementów systemu, związki między nimi, które trudno wyrazić na innych diagramach





DIAGRAMY W UML DYNAMICZNE

- Diagram przypadków użycia
 - przedstawia system z punktu widzenia użytkowników
 - pokazuje **co** system robi (nie **jak** to robi)



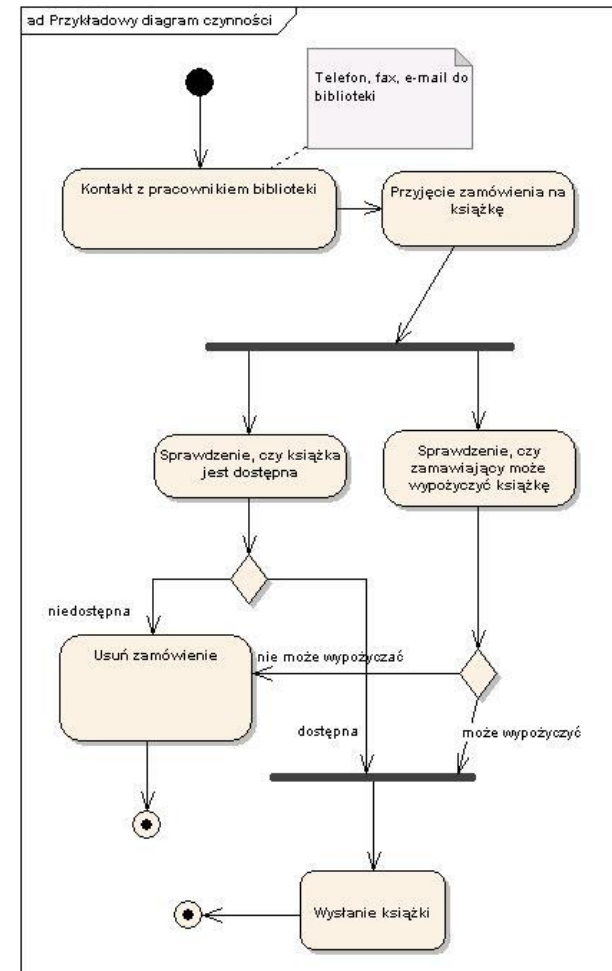
<http://www.erudis.pl/>



DIAGRAMY W UML DYNAMICZNE

• Diagram czynności

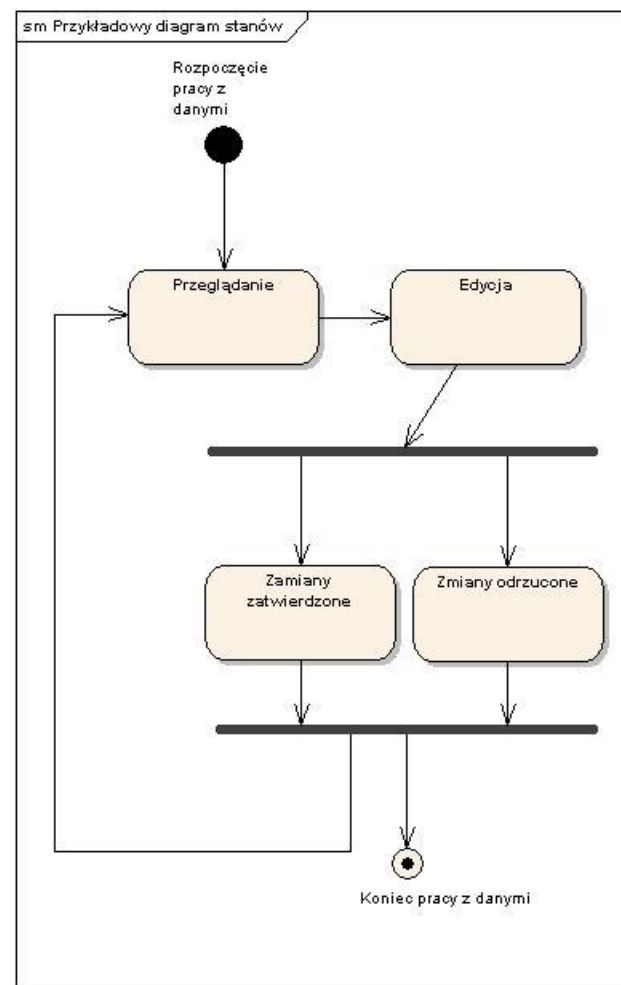
- przedstawia strumień kolejno wykonywanych czynności
- skupia się na opisanu jakiegoś procesu, w którym uczestniczy wiele obiektów





DIAGRAMY W UML DYNAMICZNE

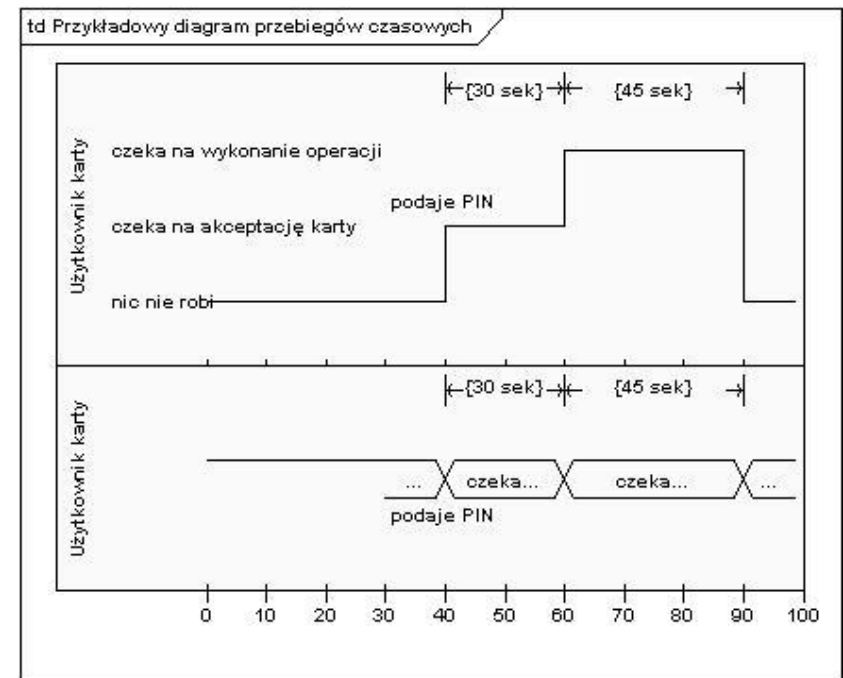
- Diagram maszyny stanowej
 - obrazuje maszynę stanową składającą się ze stanów, przejść, zdarzeń i czynności
 - pokazuje, jakie są możliwe stany konkretnego obiektu spowodowane zdarzeniami





DIAGRAMY W UML DYNAMICZNE

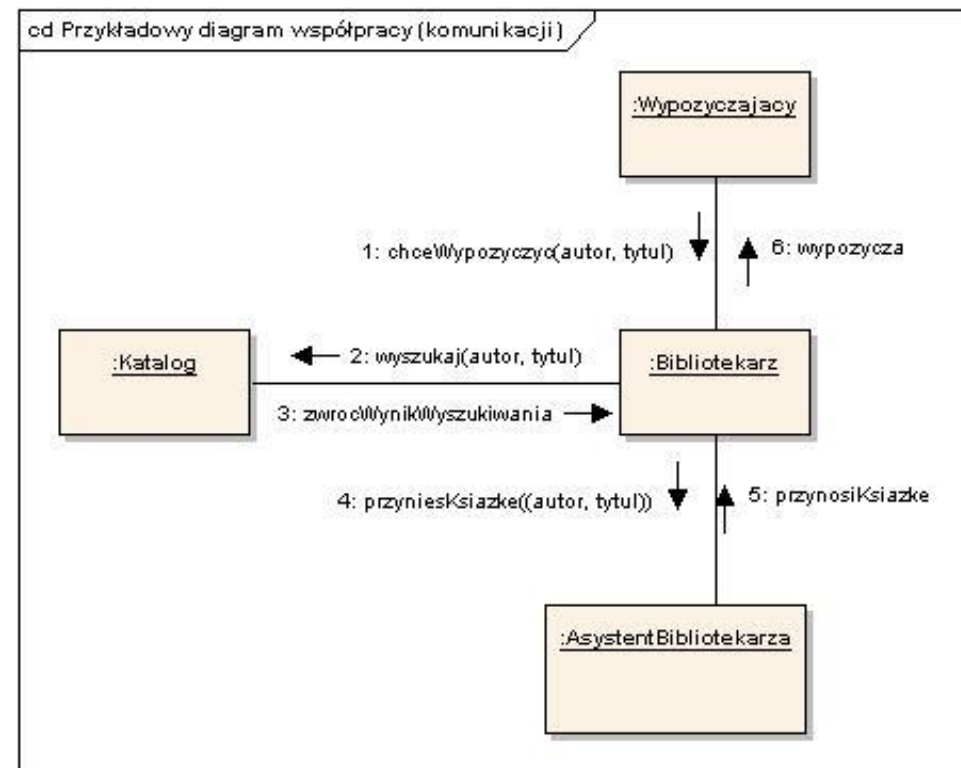
- Diagram przebiegów czasowych
 - obrazuje zachowanie obiektu z naciskiem na dokładne określenie czasu, w którym obiekt jest poddawany jakimś zamianom lub sam wykonuje jakieś działanie





DIAGRAMY W UML DYNAMICZNE

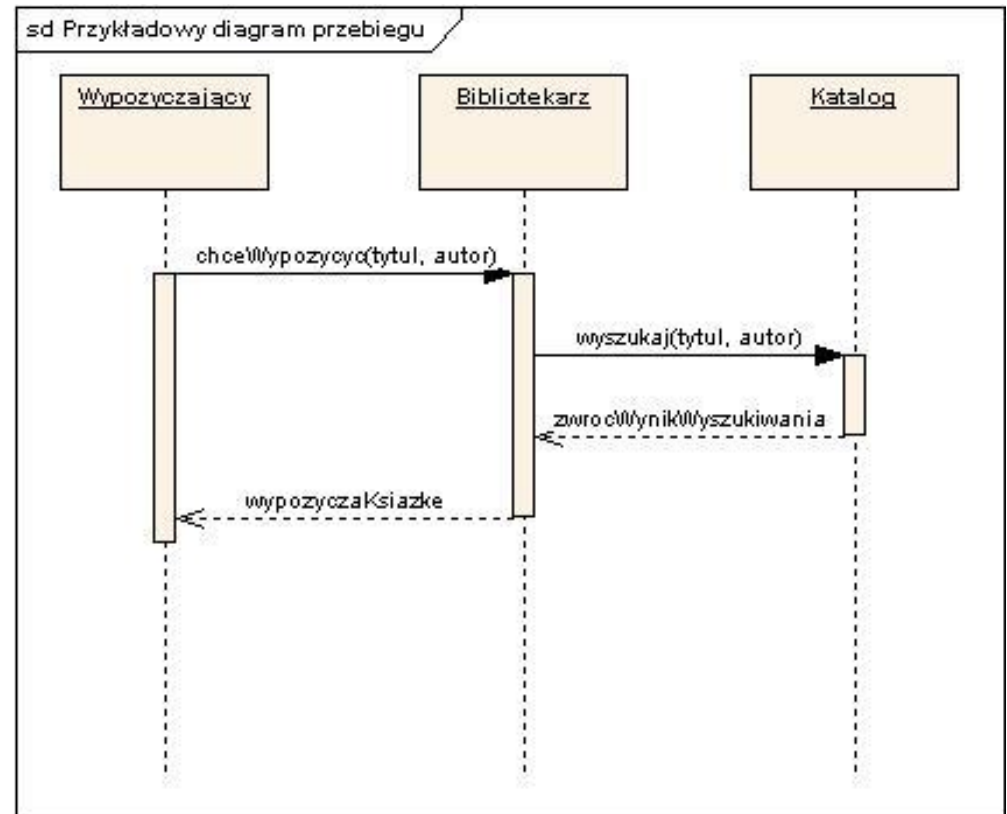
- Diagram komunikacji
 - przedstawia wzajemne oddziaływanie na siebie obiektów oraz komunikaty, jakie między sobą przesyłają





DIAGRAMY W UML DYNAMICZNE

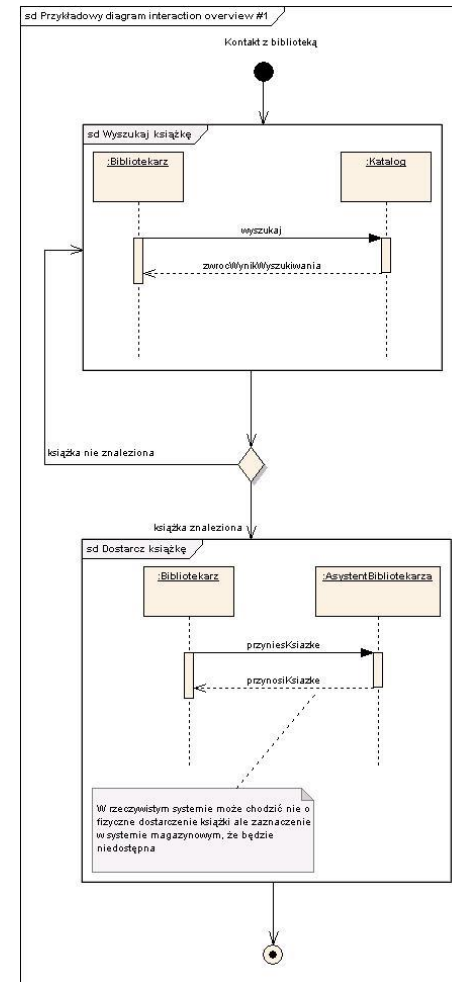
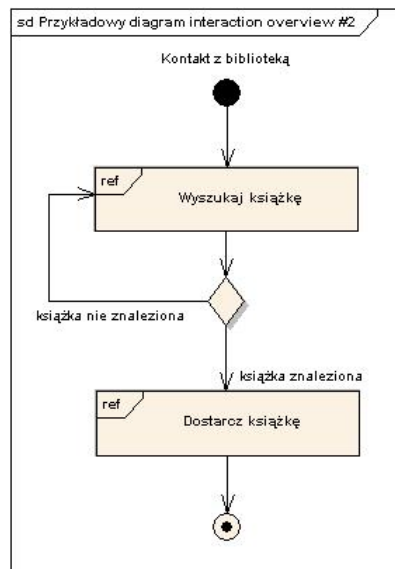
- Diagram przebiegu
 - kładzie nacisk na kolejność przesyłania komunikatów i czas istnienia obiektów





DIAGRAMY W UML DYNAMICZNE

- Diagram przeglądu interakcji
 - stanowi połączenie diagramu czynności i diagramu przebiegu
 - obrazuje współpracę diagramów interakcji



<http://www.erudis.pl/>



REGUŁY BUDOWY SCHEMATÓW APLIKACYJNYCH UML



SCHEMAT APLIKACYJNY UML

- Powinien być zapisany w **języku schematu pojęciowego UML** według zasad określonych w normach **ISO 19109** i **ISO 19103**
- Składa się z pojęć określonych przez aplikację (dziedzinę zastosowań) wyrażonych jako klasy i powiązania
- Niektóre z klas mogą być zaimportowane ze schematów znormalizowanych z innych standardów



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

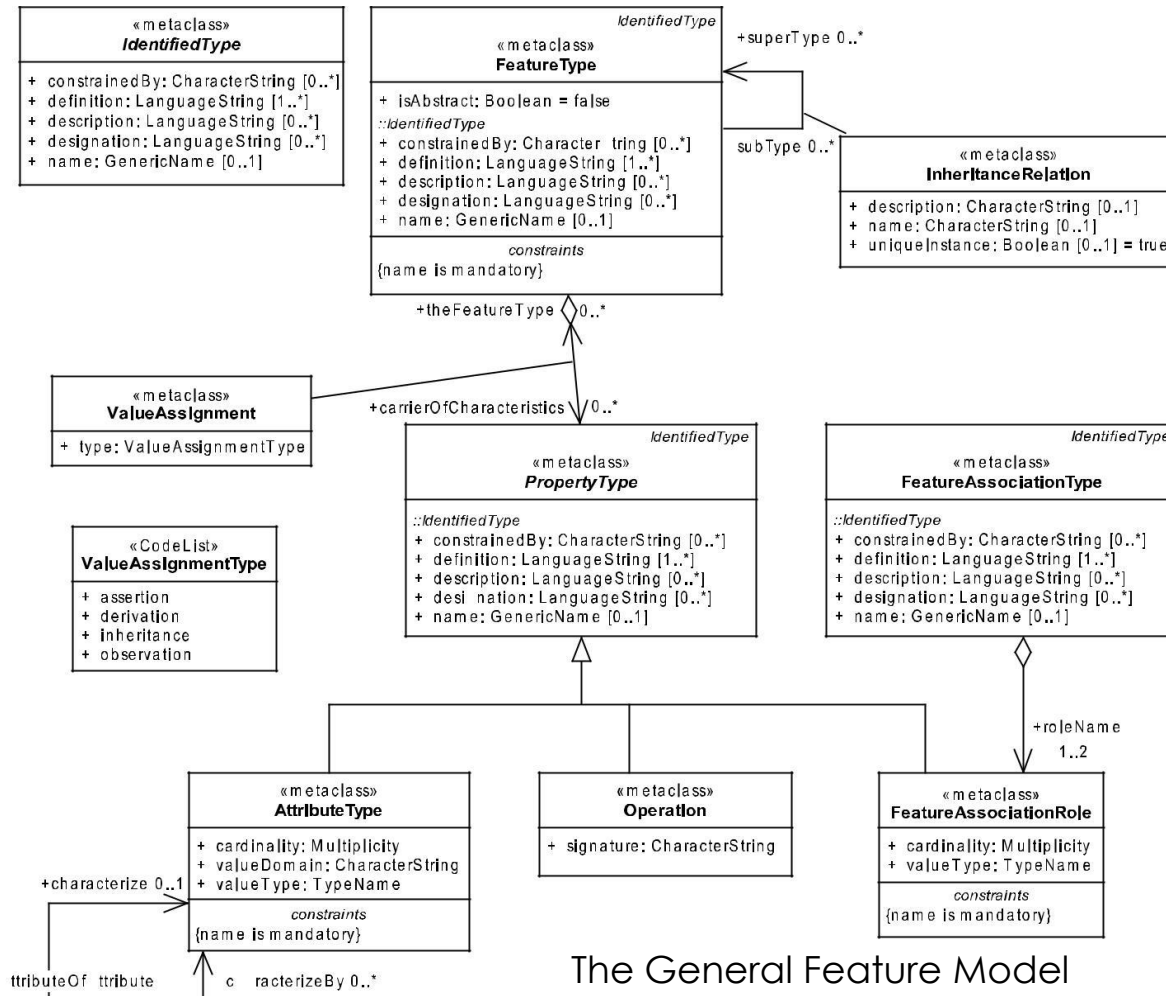


ISTOTA ISO 19109

- 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)



ISTOTA ISO 19109 – GFM



The General Feature Model



ISTOTA ISO 19109 – GFM

- **GFM** definiuje abstrakcyjny obiekt posiadający właściwości
 - **atrybuty** zawierają wszystkie statyczne informacje takie, jak jakość obiektu, jego właściwości geometryczne
 - **operacje** zawierają informacje o zmianie obiektu (np. droga została zamknięta w zimie albo droga jest wyświetlana tylko w zakresie skal 1:5 000 do 1:25 000); taka zmiana jest również nazywana zachowaniem obiektu



ISO 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
 - diagramy pakietów
 - inne diagramy UML mogą być stosowane informacyjnie



DIAGRAM KLAS W UML

- Opis
 - zbioru danych
 - zależności pomiędzy nimi
- Podstawowe elementy
 - oznaczenia klas
 - związków pomiędzy nimi
 - oznaczenia pomocnicze (np. stereotypy, metki, ograniczenia itp.)



DIAGRAM KLAS PRZYKŁAD

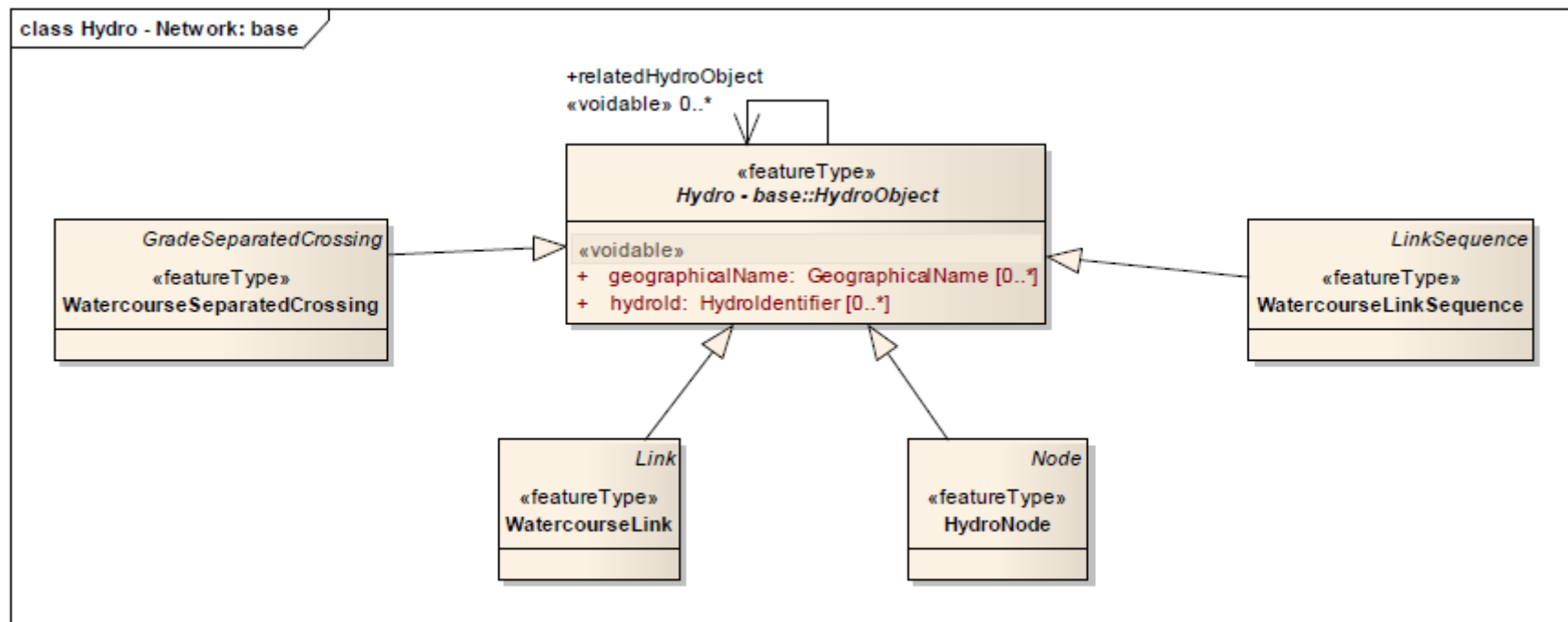


Figure 6 – UML class diagram: Overview of the ‘Hydro – Network’ application schema



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



DIAGRAM PAKIETÓW PRZYKŁAD

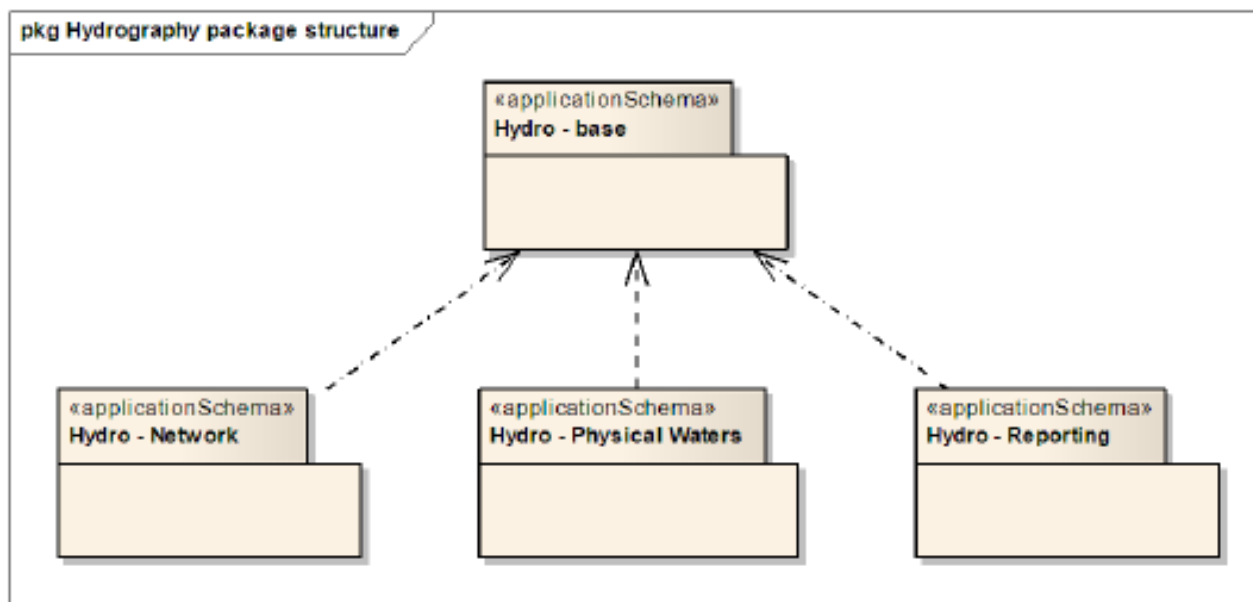
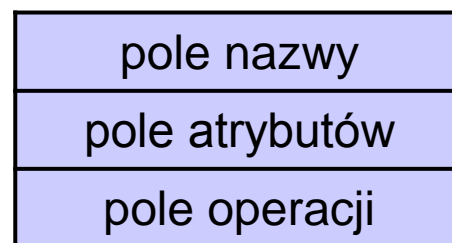


Figure 2 - Package structure of the Hydrography application schemas



KLASA

- Opis zbioru obiektów, które mają takie same atrybuty, operacje, związki i znaczenie (wzorzec dla tworzonego obiektu)
- Według ISO 19103 każda klasa powinna być zdefiniowana wewnątrz określonego pakietu
 - **graficznie:** 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





NAZWA KLASY

- Każda klasa musi mieć przypisaną nazwę, która wyróżnia ją spośród innych klas
 - napis, pisany pogrubioną czcionką
 - na ogół w formie krótkiego rzeczownika lub wyrażenia rzeczownikowego (mianownik, l. poj.), pochodzącego ze słownictwa modelowanego systemu
 - !!! lepiej unikać polskich znaków diakrytycznych
 - każdy wyraz w nazwie zaczyna się wielką literą
 - jeżeli nazwa klasy jest wielocłonowa, każdy kolejny człon zaczyna się wielką literą, a człony są ze sobą „sklejone” (brak spacji, podkreślenia)
 - np. Drzewo, PomnikPrzyrody

PomnikPrzyrody

Drzewo



NAZWA KLASY

- Zgodnie z ISO 19103, nazwa klasy powinna być poprzedzona przedrostkiem składającym się z dwóch wielkich liter, który identyfikuje pakiet lub normę, z której taka klasa pochodzi
 - np. **GM** – geometria, **TP** – topologia

AD	Abstract Address	ISO 19133	LI	Lineage	ISO 19115
CC	Changing Coordinates	ISO 19111	LR	Linear Reference	ISO 19133
CI	Citation	ISO 19115	MD	Metadata	ISO 19115
CV	Coverages	ISO 19123	NT	Network Position	ISO 19133
DQ	Data Quality	ISO 19115	PF	Feature Portrayal	ISO 19117
DS	Dataset	ISO 19115	PS	Positioning Services	ISO 19116
EX	Extenet	ISO 19115	RS	Reference System	ISO 19111
FC	Feature Catalogue	ISO 19110	SC	Spatial Coordinates	ISO 19111
FD	Feature ID	ISO 19133	SD	Sensor and Data Models	ISO 19130
FE	Feature	ISO 19109	SI	Spatial Identification	ISO 19112
FT	Feature Topology	ISO 19107	SV	Services	ISO 19119
GF	General Feature Model	ISO 19109	TM	Temporal	ISO 19108
GM	Geometry Model	ISO 19107	TP	Topology	ISO 19107



ATRYBUT KLASY

- Nazwana właściwość klasy
 - klasa może mieć dowolną liczbę atrybutów albo nie mieć ich wcale
 - nazwa atrybutu może być dowolnym tekstem (podobnie jak nazwa klasy), na ogół w formie rzeczownika lub wyrażenia opisującego właściwość danej klasy (mianownik, l. poj.)
 - !!! lepiej unikać polskich znaków diakrytycznych
 - 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

Pracownik
nazwisko:CharacterString
dataUrodzenia:Date



OPERACJA KLASY

- 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
 - !!! lepiej unikać polskich znaków diakrytycznych
 - każdy wyraz w nazwie (poza pierwszym) zaczyna się wielką literą, kolejne wyrazy są ze sobą „sklejone”
 - np. wycofaj, zatwierdzTransakcje

Transakcja
wycofaj() zatwierdzTransakcje()



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

Pracownik
– nazwisko
– nrTelefonu
haslo
+ ustawNrTelefonu()
zmienHaslo()



TYPY DANYCH

- ISO 19103 grupuje podstawowe typy danych w 3 kategorie
 - **elementarne** (ang. *primitives types*)
 - **implementacyjne** (ang. *implementation types*)
 - **pochodne** (ang. *derived types*)



ELEMENTARNE TYPY DANYCH

- Podstawowe typy dla reprezentacji wartości
 - Integer
 - Decimal
 - Real
 - Vector
 - CharacterString
 - Date
 - Time
 - DateTime
 - Boolean
 - Logical
 - Probability
 - Multiplicity



ELEMENTARNE TYPY DANYCH PRZYKŁADY

Integer	123, -65547
Decimal	12.34
Real	12.34, -1.234E-4
Vector	(123,456,789)
CharacterString	"This is a nice place"
Date	2003-02-19
Time	13. 59:30 or 13:59:30-05:00
DateTime	2003-02-19T13:59:30
Boolean	TRUE, FALSE
Logical	TRUE, FALSE, MAYBE
Probability	0.0 <p< 1.0
Multiplicity	1..*



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



IMPLEMENTACYJNE TYPY DANYCH PRZYKŁADY

Collection types (typy zbiorowe)

Set	zbiór elementów nieuporządkowanych bez powtórzeń
Bag	zbiór elementów nieuporządkowanych z powtórzeniami
Sequence	jak Bag, ale elementy uporządkowane
Dictionary	tabela (array) elementów indeksowanych

Enumeration types (typy wyliczeniowe)

Enumeration (wyliczenie)	np. {Zielony, Czerwony, Niebieski}
CodeList (lista kodów)	np. {2003, 2004, 2005, 2006}

Representation types (typy reprezentacyjne)

Record, Record type	lista par typu (<i>nazwa, wartość</i>); np. (Warszawa, 4000)
Generic name	nazwa rodzajowa, np. GM_Object, TM_TimeInstant



POCHODNE TYPY DANYCH

- Typy miar i jednostki miary
 - Area
 - Length (Distance)
 - Angle
 - Scale
 - Mtime
 - Volume
 - Velocity



ZWIĄZKI

- Zależność (ang. *dependency*)
- Powiązanie/Asocjacja (ang. *association*)
 - Nawigacja (ang. *navigation*)
 - Agregacja zwykła/częściowa (ang. *aggregation*)
 - Agregacja zupełna/całkowita (ang. *composition*)
- Uogólnienie/Generalizacja (ang. *generalization*)



ZALEŻNOŚĆ

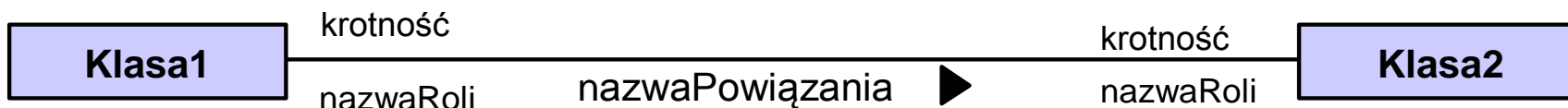
- Związek użycia między dwoma elementami
- Zmiany dokonane w specyfikacji jednego elementu (np. klasy *Zamowienie*) mogą mieć wpływ na inny element, który używa tego pierwszego (np. na klasę *ZamowienieSpecjalne*), ale niekoniecznie na odwrót
- **graficznie:** linia przerywana z grotem skierowanym na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





POWIĄZANIE (ASOCJACJA)

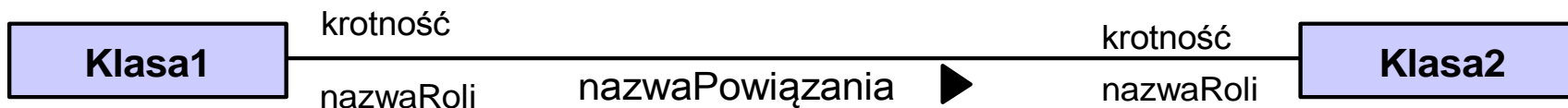
- Związek strukturalny równorzędnych partnerów, tzn. żadna klasa nie jest ważniejsza
- Wskazuje, iż obiekty jednego elementu są połączone z obiektami innego
- Dla każdej pary klas/obiektów można zdefiniować dowolnie wiele powiązań, z których każde reprezentuje inną relację
- **graficznie:** linia ciągła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





POWIĄZANIE (ASOCJACJA)

- Można opisać za pomocą
 - nazwy
 - ról
 - krotności

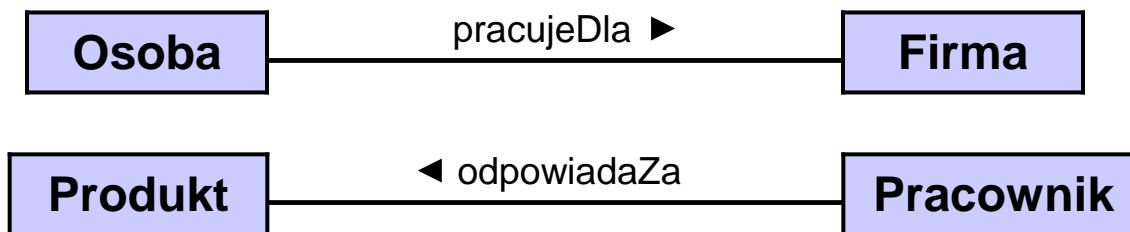




POWIĄZANIE (ASOCJACJA)

- **Nazwa powiązania**

- powiązanie może mieć przypisaną nazwę, która określa istotę danego związku
- aby uniknąć niejednoznaczności, można podać kierunek odczytu (trójkątny znacznik przy nazwie)
- nazwa powiązania nie jest konieczna, gdy określone są role
- zwykle czasownik w czasie teraźniejszym, w 3os. l. poj., np.
 - **konwencja zapisu:** lepiej unikać spacji, można stosować taką samą konwencję zapisu jak w przypadku nazw atrybutów dla klasy

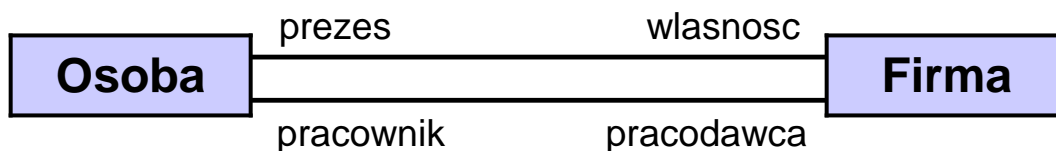




POWIĄZANIE (ASOCJACJA)

- Rola w powiązaniu

- klasa biorąca udział w powiązaniu odgrywa 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
- zwykle w formie krótkiego rzeczownika lub wyrażenia rzeczownikowego (mianownik, l. poj.); alias klasy
 - **konwencja zapisu:** należy stosować taką samą konwencję zapisu jak w przypadku nazw atrybutów dla klasy

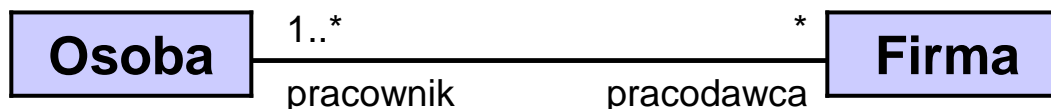




POWIĄZANIE (ASOCJACJA)

- **Krotność (liczność)**
 - ile obiektów może być połączonych przez jeden egzemplarz powiązania

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..*





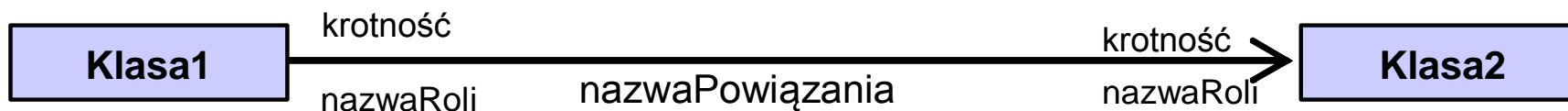
POWIĄZANIE RODZAJE

- Nawigacja
(ang. *navigation*)
- Agregacja zwykła/częściowa
(ang. *aggregation*)
- Agregacja zupełna/całkowita, złożenie, kompozycja
(ang. *composition*)



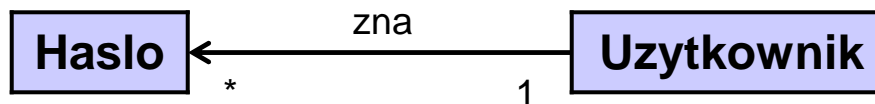
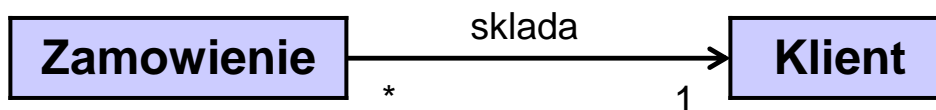
POWIĄZANIE NAWIGACJA

- Powiązanie jednokierunkowe
 - tylko jedna z klas biorących udział w powiązaniu „wie” o drugiej klasie
 - **graficznie:** strzałka dołączona do zakończenia linii powiązania i wskazującej na daną klasę
 - brak strzałki: powiązaniem można poruszać się w obu kierunkach (obie klasy wiedzą o sobie) = powiązanie (asocjacja)
 - kierunek odczytu nazwy powiązania wskazuje grot strzałki (brak trójkątnego znacznika przy nazwie związku)





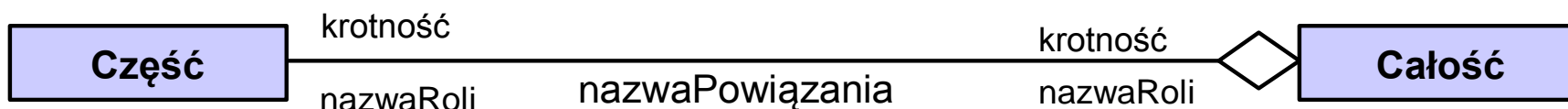
POWIĄZANIE NAWIGACJA PRZYKŁAD





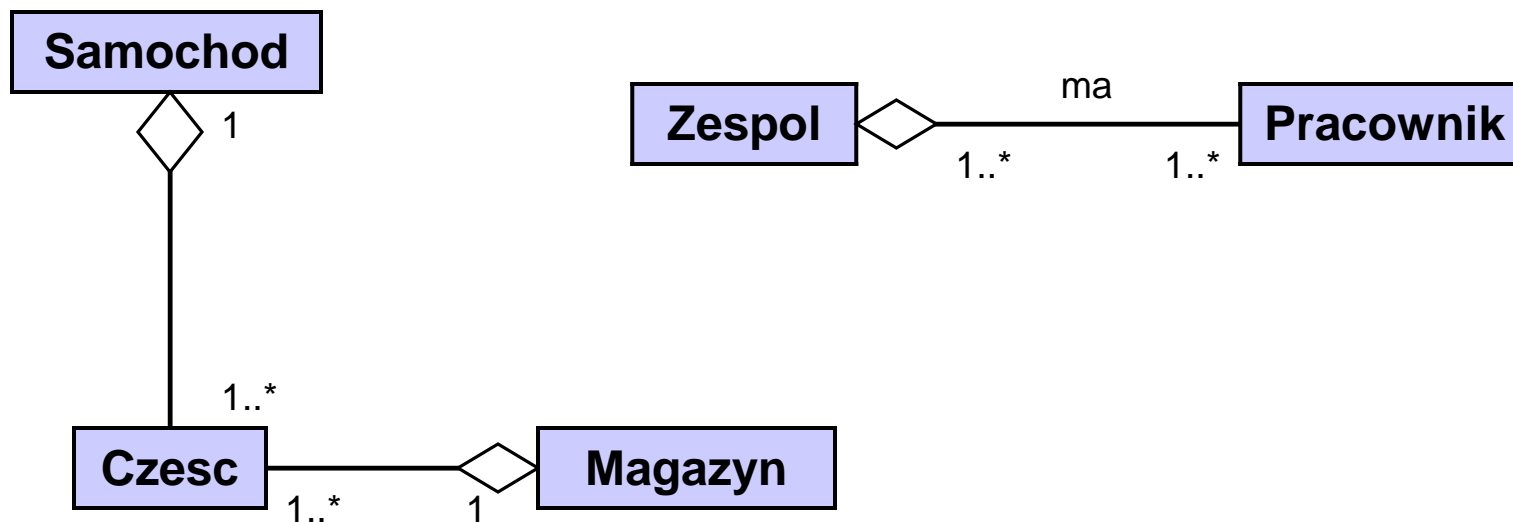
POWIĄZANIE AGREGACJA ZWYKŁA/CZĘŚCIOWA

- **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
 - czas życia klas-części nie jest ograniczony do czasu życia klasy-całości
- **graficznie:** pusty romb umieszczony na końcu linii powiązania od strony klasy całości





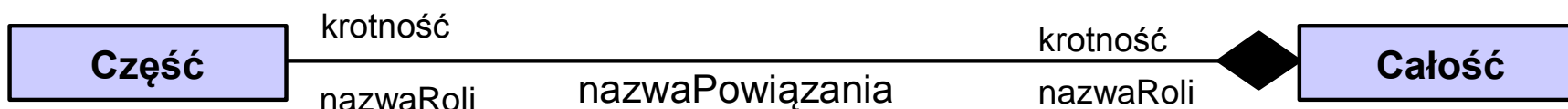
POWIĄZANIE AGREGACJA ZWYKŁA/CZĘŚCIOWA PRZYKŁAD





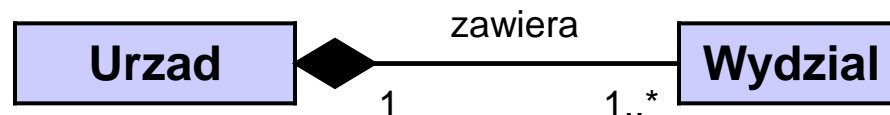
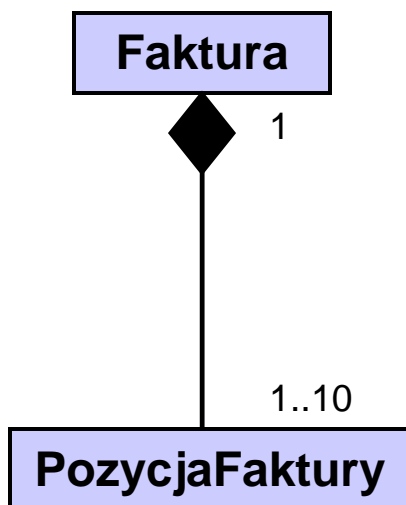
POWIĄZANIE AGREGACJA ZUPEŁNA/CAŁKOWITA (ZŁOŻENIE, KOMPOZYCJA)

- **związek zawierania** między klasami
 - klasa-całość zawiera inne klasy, które stanowią jej składniki (klasy-części)
 - „całość” **zawiera** „części”
 - klasy-części nie mogą być składnikami innych klas-całości
 - czas życia klas-części jest ograniczony do czasu życia klasy-całości
- **graficznie:** wypełniony romb umieszczony na końcu linii powiązania od strony klasy całości



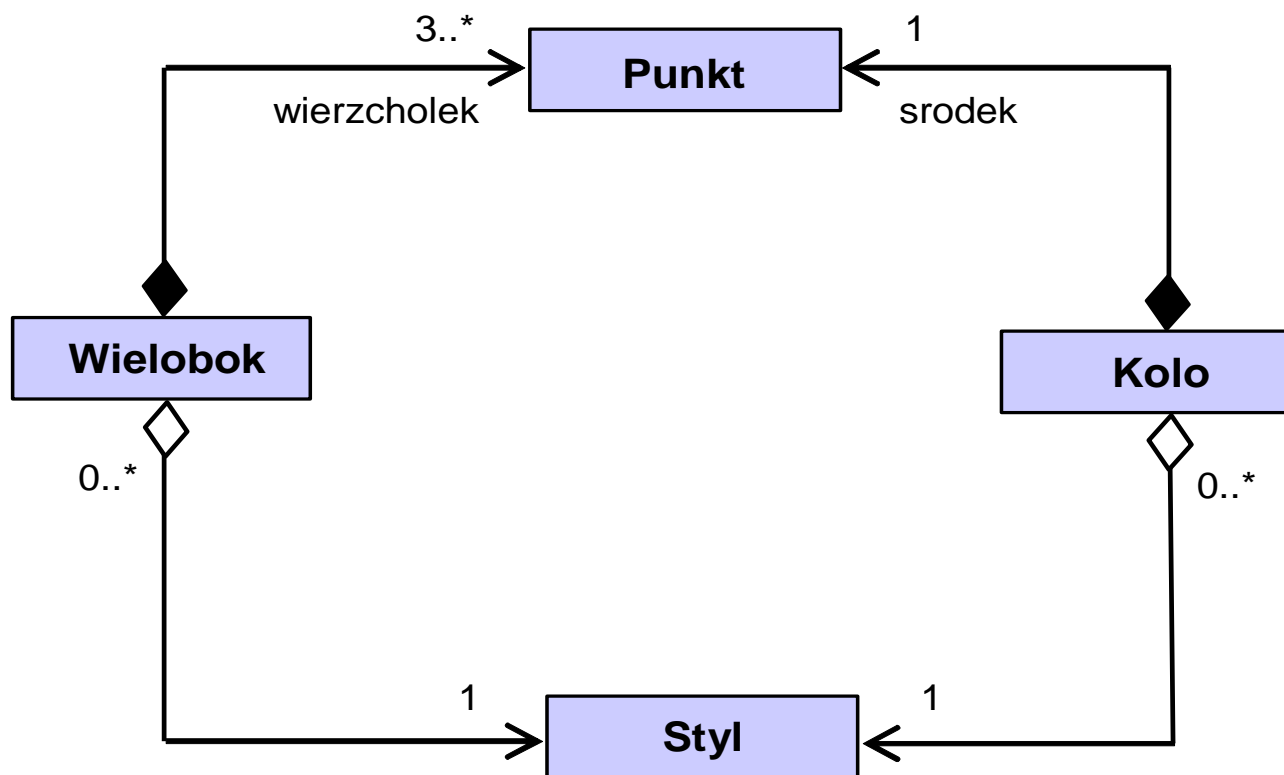


POWIĄZANIE AGREGACJA ZUPEŁNA/CAŁKOWITA PRZYKŁAD





POWIĄZANIE AGREGACJA, NAWIGACJA PRZYKŁAD





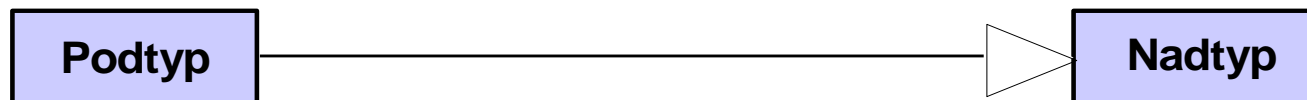
UOGÓLNIENIE (GENERALIZACJA)

- Związek między elementem ogólnym (nadklasą, nadtypem, przodkiem) a jego pewnym specyficznym rodzajem (podklasą, podtypem, potomkiem)
 - **spotykane nazwy:** uogólnienie, generalizacja, dziedziczenie, generalizacja-specjalizacja, nadtyp-podtyp, przodek-potomek
 - potomek dziedziczy wszystkie właściwości przodka
 - atrybuty
 - operacje
 - powiązania przodka
 - ograniczenia
 - najczęściej potomek ma jeszcze własne cechy, poza odziedziczonymi



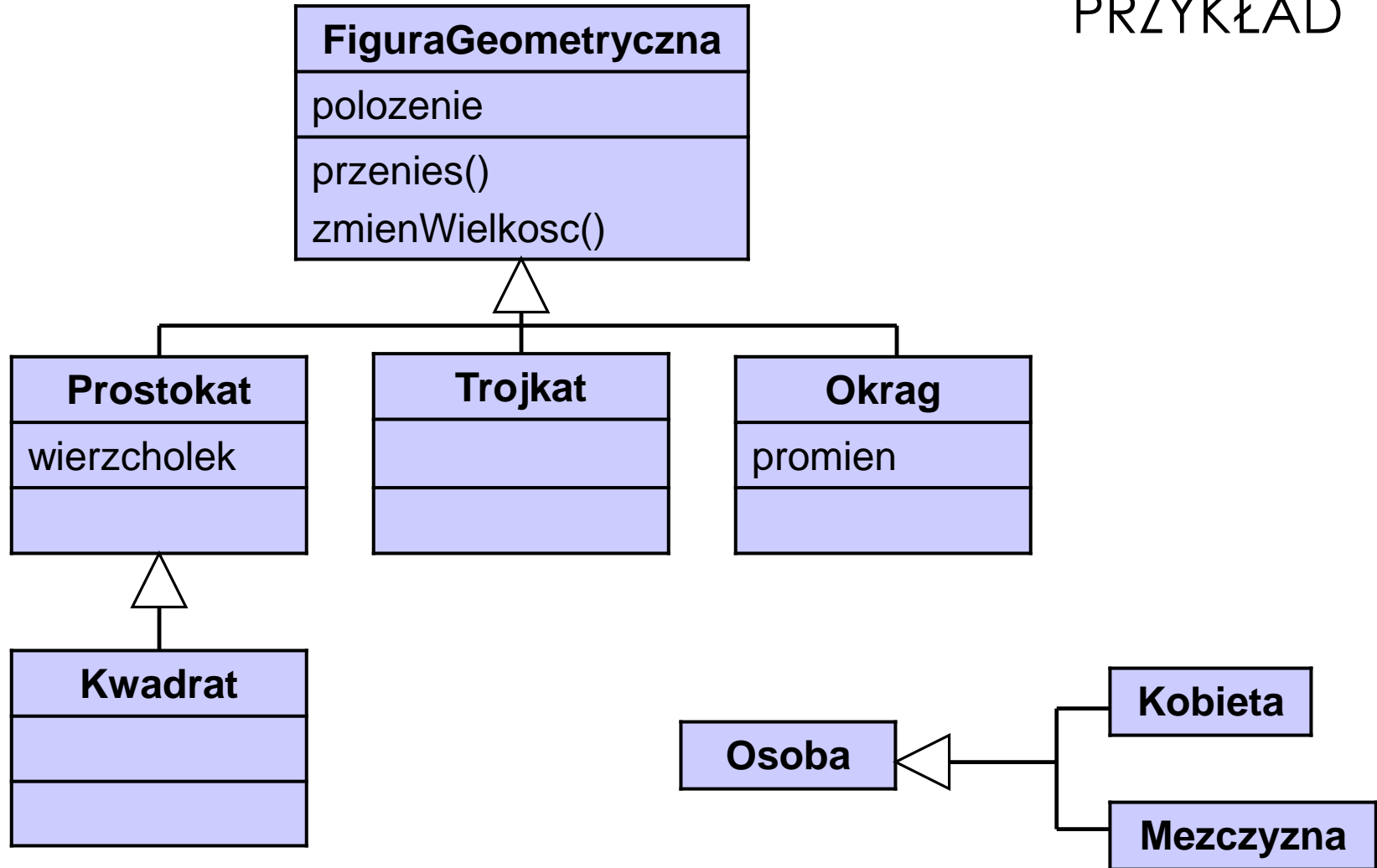
UOGÓLNIENIE (GENERALIZACJA)

- klasa może nie mieć przodka, a może mieć jednego lub więcej
- **korzeń** (klasa podstawowa)
 - klasa bez przodków, ale z co najmniej jednym potomkiem
- **liść**
 - klasa bez potomków
- uogólnienie może mieć nazwę, ale zwykle nie jest potrzebna
- **graficznie:** linia ciągła zakończona niewypełnionym trójkątem wskazującym przodka





UOGÓLNIENIE PRZYKŁAD





MECHANIZMY ROZSZERZENIA

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



MECHANIZMY ROZSZERZENIA STEREOTYP

- Wyrażenie umożliwiające klasyfikację elementów modelu
- Wspólna, nazwana własność obiektów, klas, powiązań, atrybutów, ograniczeń
 - dla każdego rodzaju elementu UML istnieje lista dozwolonych stereotypów
 - ISO 19103 definiuje 11 stereotypów dla informacji geograficznej (9 to standardowe stereotypy UML)
 - «Interface»
 - «Type»
 - «Control»
 - «Entity»
 - «Boundary»
 - «Enumeration»
 - «Exception»
 - «MetaClass»
 - «DataType»
 - «CodeList»
 - «Leaf»



MECHANIZMY ROZSZERZENIA METKA (WARTOŚĆ ETYKIETOWANA)

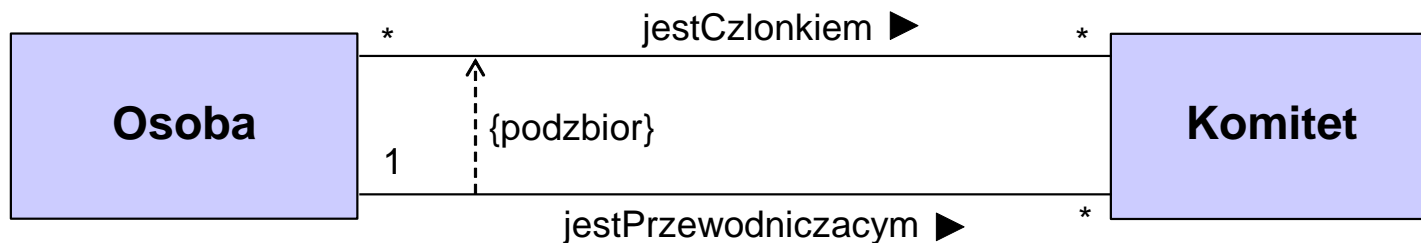
- Pozwala dołączyć do elementu dodatkowe właściwości
 - para *klucz = wartość*
 - listę wartości etykietowanych oddzielonych przecinkami umieszcza się w nawiasach klamrowych {}

{ autor = 'Adam Mickiewicz', dataWydania = '1997' }



MECHANIZMY ROZSZERZENIA OGRANICZENIE

- Określa restrykcje nakładane na elementy modelu
 - może
 - stanowić wyrażenie języka naturalnego
 - stanowić wyrażenie języka formalnego (np. OCL w UML)
 - przyjmować postać formuły matematycznej
 - przyjmować postać fragmentu kodu





MECHANIZMY ROZSZERZENIA OGRANICZENIE JĘZYK OCL

- Język OCL (ang. *Object Constraint Language*)
 - formalny język zapisu ograniczeń w UML
 - podstawowe operatory języka OCL
 - relacje = > < >= <= <>
 - arytmetyczne + - * /
 - logiczne and or xor not implies
 - warunek if .. then .. else .. Endif
 - operatory zbioru count exists forAll includes includesAll isEmpty iterate notEmpty sum



MECHANIZMY ROZSZERZENIA OGRANICZENIE JĘZYK OCL

Class: EnvironmentalMonitoringFacility

Constraint: GeometryRequired Type: Invariant Status: Approved

B I U A | |

```
/* Geometry and representativePoint cannot both be empty.*/  
inv: geometry -> notEmpty() or representativePoint -> notEmpty()
```

Constraint	Type
GeometryRequired	Invariant

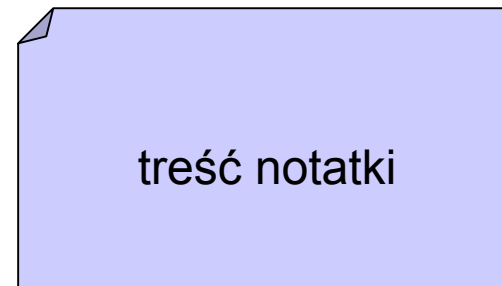
OK Anuluj Zastosuj Pomoc

```
«featureType»  
EnvironmentalMonitoringFacility  
  
«voidable»  
+ representativePoint: GM_Point [0..1]  
+ measurementRegime: MeasurementRegimeValue  
+ mobile: Boolean  
+ resultAcquisitionSource: ResultAcquisitionSourceValue [0..*]  
+ specialisedEMFType: SpecialisedEMFTypeValue [0..1]  
  
constraints  
{GeometryRequired}
```



INNE ELEMENTY NA DIAGRAMACH UML

- **Notatka** (ang. *note*)
 - symbol graficzny umożliwiający przedstawienie dodatkowych ograniczeń i objaśnień do modelu
 - **graficznie:** prostokąt z zagiętym rogiem, z komentarzem tekstowym lub graficznym umieszczonym w środku





ETAPY BUDOWY SCHEMATU APLIKACYJNEGO UML

- Identyfikacja dziedziny problemowej oraz przegląd wymagań
- Opracowanie modelu pojęciowego, w tym identyfikacja
 - typów obiektów (klasy)
 - ich właściwości (atrybuty)
 - związków między nimi
 - ograniczeń
- Zapisanie modelu w języku formalnym
 - schemat aplikacyjny UML
- Integracja opracowanego schematu aplikacyjnego ze schematami znormalizowanymi
 - np. zdefiniowanymi w normach ISO serii 19100 (geometria i topologia, jakość, opis położenia itp.)



STEREOTYPY STOSOWANE W INSPIRE

- Pakiet
 - «ApplicationSchema»
 - schemat aplikacyjny



STEREOTYPY STOSOWANE W INSPIRE

- Klasa
 - «FeatureType»
 - typ obiektu przestrzennego
 - «CodeList»
 - lista predefiniowanych wartości, którą można rozszerzyć
 - «Enumeration»
 - stała lista predefiniowanych wartości (nie można jej rozszerzać)
 - «DataType»
 - definicja strukturalnego typu danych (posiada własne atrybuty)
 - «Union»
 - strukturalny typ danych, dla którego dokładnie jeden z atrybutów musi wystąpić



STEREOTYPY STOSOWANE W INSPIRE

- Atrybut, rola w powiązaniu
 - «voidable»
 - pewna właściwość (cecha) obiektu przestrzennego nie jest prezentowana w zbiorze danych przestrzennych
 - ale może być obecna lub mieć zastosowanie w świecie rzeczywistym



STEREOTYPY STOSOWANE W INSPIRE

- Element oznaczony stereotypem «voidable» może przyjmować wartości
 - *inapplicable* – (*nie stosuje się*) brak wartości, nie ma zastosowania
 - *missing* – (*brak danych*) prawidłowa wartość nie jest łatwo dostępna, może nie istnieć
 - *template* – (*tymczasowy brak danych*) wartość będzie dostępna później
 - *unknown* – (*nieznany*) prawidłowa wartość nie jest znana, ale prawdopodobnie istnieje
 - *withheld* – (*zastrzeżony*) wartość nie została ujawniona, jest zastrzeżona
 - *other: + text* – inne krótkie wyjaśnienie, gdzie *text* jest łańcuchem znaków bez spacji (np. *other:unpopulated*)
 - *anyURI* – referuje na zasób, gdzie opisano przyczynę wyjątku



STEREOTYPY STOSOWANE W INSPIRE

- Atrybut, rola w powiązaniu
 - «lifeCycleInfo»
 - pewna właściwość (cecha) obiektu przestrzennego jest częścią informacji dotyczącej cyklu życia tego obiektu przestrzennego



STEREOTYPY STOSOWANE W INSPIRE

- Przykłady atrybutów oznaczonych stereotypem «lifeCycleInfo»
 - *beginLifespanVersion*
 - data i godzina, w której ta wersja obiektu przestrzennego została wprowadzona do zbioru danych przestrzennych lub zmieniona w tym zbiorze
 - *endLifespanVersion*
 - data i godzina, w której ta wersja obiektu przestrzennego została zastąpiona w zbiorze danych przestrzennych lub wycofana z tego zbioru
 - *validFrom*
 - oficjalna data i godzina, w której dany obiekt przestrzenny został/zostanie zgodnie z prawem ustanowiony (zaczął występować w świecie rzeczywistym)
 - *validTo*
 - data i godzina, w której dany obiekt przestrzenny przestał/przestanie zgodnie z prawem być stosowany (nie występuje już w świecie rzeczywistym)

PODSUMOWANIE





DLACZEGO WARTO ZNAĆ UML?

- **Opracowywanie** standardów danych przestrzennych
- **Tworzenie** zbiorów/baz danych przestrzennych
- **Korzystanie** ze standardów danych przestrzennych



konieczna znajomość UML

UML to uniwersalny środek formalny

- zalecany przez normy ISO serii 19100 do budowy schematów aplikacyjnych
- stosowany w specyfikacjach danych INSPIRE i rozporządzeniach krajowych