

Inżynieria wiedzy

Agnieszka Ławrynowicz

Instytut Informatyki
Poznań, 2017

Wiedza

- dane: "surowe sygnały"
 - informacja: znaczenie dołączone do sygnałów
 - wiedza: cel, kompetencje dołączone do informacji, działanie
-

Inżynieria wiedzy: klasyczne role

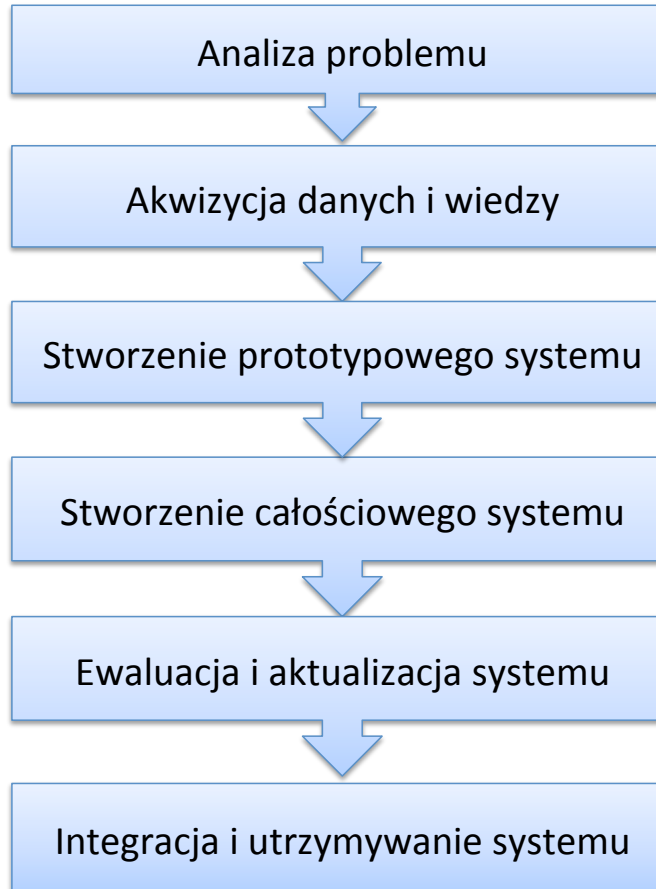
Ekspert dziedzinowy

- "klient"
- dostarcza wiedzy i procesów potrzebnych do zrozumienia problemu

Inżynier wiedzy

- pozyskuje wiedze od eksperta dziedzinowego
 - odwzorowuje wiedzę i procesy do bazy wiedzy / systemu informacyjnego
-

Inżynieria wiedzy: etapy



proces: pozyskiwanie, strukturalizacja, formalizacja, działanie operacyjne

Akwizycja wiedzy: *hackathony*

OpenML, Lorentz Center (Holandia), 2016



Akwizycja wiedzy: arkusze kalkulacyjne połączone z ontologią

Populous

The screenshot shows a spreadsheet application window titled "Populous - /Users/simon/Documents/e-lico/kupo_cells_sept10_Populous.xls". The spreadsheet has four columns: A (Cell Term), B (cell label), C (part_of), and D (participates_in). The rows list various biological terms and their relationships. For example, "kidney cell" is a "renal cell" which is a "part_of" "kidney".

Cell Term	cell label	part_of	participates_in
kidney cell	renal cell	kidney	
kidney epithelial cell		kidney epithelium	
renal tubule epithelial cell		renal tubule epithelium	
kidney glomerular epithelial cell		kidney glomerular epithelium	
renal tubule cell		renal tubule	
kidney cortex cell	renal cortex cell	kidney cortex	
renal cortex tubule cell		renal cortex tubule	
kidney medulla cell	renal medullary cell	kidney medulla	
kidney outer medulla cell	renal outer	outer renal medulla	
kidney inner medulla cell	renal inner medullary	inner renal medulla	
inner renal medulla loop of henle cell		inner renal medulla loop of henle	
juxtaglomerular complex cell	juxtaglomerular	juxtaglomerular complex	regulation of glomerular filtration, regulation of blood circulation
kidney blood vessel cell	renal blood vessel	kidney blood vessel	
kidney arterial blood vessel cell	renal arterial blood	kidney arterial blood vessel	
kidney capillary endothelial cell	renal capillary cell	kidney capillary, capillary endothelium	
kidney venous blood vessel cell	renal venous blood vessel cell	kidney venous blood vessel	
renal corpuscle cell	cell of the renal	renal corpuscle	
mesangial cell		mesangium	phagocytosis, extracellular matrix constituent
glomerular mesangial cell		glomerular mesangium	
extraglomerular mesangial cell		extraglomerular mesangium	
podocyte	visceral epithelial	glomerular visceral epithelium	glomerular filtration, regulation of glomerular anatomical structure arrangement
bowmans capsule epithelial cell	epithelial cell of the	bowmans capsule	
parietal epithelial cell	glomerular parietal	glomerular parietal epithelium	
glomerular cell		glomerulus	
glomerular capillary endothelial cell	glomerular capillary	glomerular capillary endothelium	glomerular filtration, regulation of glomerular anatomical structure arrangement
renal afferent arteriole cell	afferent arteriole cell	afferent arteriole	regulation of glomerular filtration
renal afferent arteriole endothelial cell	afferent arteriole	afferent arteriole, arteriole endothelium	
juxtaglomerular cell		part of afferent arteriole forming	renin secretion into blood stream, detection
renal afferent arteriole smooth muscle	afferent arteriole	afferent arteriole, arteriole smooth	
renal afferent arteriole cell	afferent arteriole cell	afferent arteriole	regulation of glomerular filtration
renal afferent arteriole endothelial cell	afferent arteriole	afferent arteriole, arteriole endothelium	
renal afferent arteriole smooth muscle	afferent arteriole	afferent arteriole, arteriole smooth	
proximal tubule epithelial cell	PTEC, proximal	renal proximal tubule	renal sodium ion absorption, potassium ion
proximal convoluted tubule epithelial		proximal convoluted tubule	
proximal straight tubule cell		proximal straight tubule	
loop of henle epithelial cell	henle's loop	loop of henle	extracellular matrix constituent secretion
loop of henle ascending limb	henle's loop	loop of henle ascending limb	renal sodium ion absorption, potassium ion
loop of henle thick ascending limb	henle's loop thick	loop of henle ascending limb thick	
loop of henle thin ascending limb	henle's loop thin	loop of henle ascending limb thin	
loop of henle medullary thick	henle's loop	distal straight tubule premacula	

The right-hand panel shows the "CELLS: B17:B17" selection. Below it is the "HIERARCHY" section with a search bar and a list of terms: Thing, abdominal tone, abdominal tone value, abiotic stress sensitivity, abiotic stress sensitivity value, absolute acceleration, absolute acceleration value, absolute activity, absolute activity value, absolute age, absolute age value, and absolute alternation. Below the hierarchy is the "TYPE OF ALLOWED VALUES" section with radio buttons for "Free text" (selected), "Direct subclasses", "Subclasses", "Instances", and "Direct instances". At the bottom is the "ALLOWED VALUES" section with a text box containing "Any".

Akwizycja wiedzy: Semantyczne Wiki

Search

Go Search

[Main page](#)
[Random page](#)
[Free GNU/Linux distros](#)
[Donate to the FSF](#)
[FSF store](#)

Interaction

[Participation guide](#)
[About the FSD](#)
[Recent changes](#)
[Mailing list](#)
[Active users](#)
[Backlog](#)

Navigation


[Pages pending approval](#)
[Recent changes](#)
[New pages](#)
[Help](#)

Tools

[Special pages](#)


The Free Software Directory

Welcome to the **Free Software Directory**, a catalog of free software that [anyone can edit](#).



Free software  is a matter of liberty, not price. Think of "free" as in "free speech", not as in "free beer". Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software.

Contribute

- Have you found a bug or an entry that needs to be updated?
- Would you like to submit a new entry?
- Or are you eager to join a team of other community members to help maintain, grow, and rebuild the Free Software Directory?

If so, visit our [Participate page](#) to find out how you can get started today. Also make sure to join the [directory-discuss](#)  mailing list to keep up on discussions or send in any questions you have about the directory.

Get involved!

Join us each Friday from 12 p.m. to 3 p.m. EST  for the FSD IRC meetings! For more details, check the [Free Software Directory blog](#) .

Query or browse

 [Browse GNU Projects \(439\)](#)


 [Query on license type](#) 

[Browse All \(15959\)](#)

[Random entry](#)

Most recent changes

 [New packages RSS feed](#)

 [Query on descriptions](#)

 [Recent updates RSS feed](#)

Akwizycja wiedzy: Crowdsourcing

[WHY AI](#)[USE CASES](#)[SUCCESS STORIES](#)[PRICING](#)[BLOG](#)[HERE TO TASK?](#)[LOGIN](#)[START TRIAL](#)[SENTIMENT ANALYSIS](#)[SEARCH RELEVANCE](#)[CONTENT MODERATION](#)[DATA COLLECTION](#)[DATA CATEGORIZATION](#)[TRANSCRIPTION](#)[IMAGE ANNOTATION](#)[OTHER USE CASES](#)

Data Collection

Enhance your business data at massive scale

Get rich, reliable data on your customers, your leads, and your business. Whether you're looking to locate URLs, addresses, investor information, or any other crucial business information, CrowdFlower can help. Just tell our contributors what to find and they'll find it for you.

[Learn more](#)

Data Categorization

Turn big data into rich data

Whether you need to tag images, distinguish parent from subsidiary companies, categorize business websites, or organize your own domain, CrowdFlower gets there efficiently. Our quality control measures make sure your data is categorized accurately, while our giant contributor-base makes sure it happens fast.



Inżynieria ontologii

Inżynieria ontologii zajmuje się zasadami, metodami i narzędziami do konstrukcji i utrzymywania ontologii.

Metodologie inżynierii ontologii dotyczą procesu i metodologicznych aspektów konstrukcji i utrzymywania ontologii i dostarczają wskazówek i rekomendacji twórcom ontologii.

Ontology Development 101: A Guide to Creating Your First Ontology (Noy &McGuinness)

Nie istnieje żadna „jedyna poprawna” metoda wytwarzania ontologii.

Ogólne reguły:

1. Najlepsze rozwiązanie zależy od aplikacji.
 2. Proces wytwarzania ontologii jest iteracyjny.
 3. Pojęcia zdefiniowane w ontologii powinny odpowiadać obiektom (fizycznym lub logicznym) i relacjom z opisywanej dziedziny.
 - w zdaniach opisujących dziedzinę: rzeczowniki (obiekty) i czasowniki (relacje)
-

Ontology Development 101: A Guide to Creating Your First Ontology (Noy &McGuinness)

Budowa ontologii realizowana w cyklicznych etapach obejmujących:



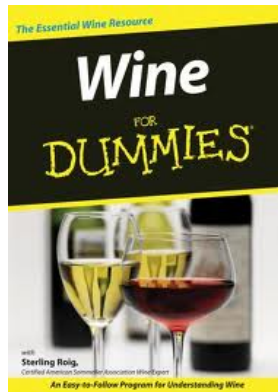
Przykład – budowa ontologii o winach i jedzeniu

Jakie wino pasuje do owoców morza?

WeLoveSeafood.com



Współdzielona ontologia o winach i jedzeniu



Typy win



Krok 1 - ustal dziedzinę i zakres ontologii

- Jakiej dziedziny dotyczy ontologia?
 - Do czego jej użyjemy?
 - Na jakiego rodzaju pytania ontologia ma udzielać odpowiedzi?
 - Kto będzie korzystał z ontologii i ją utrzymywał?
-

Krok 1 - ustal dziedzinę i zakres ontologii

- Jakiej dziedziny dotyczy ontologia?
 - reprezentacja win i jedzenia
- Do czego jej użyjemy?
 - aplikacje sugerujące wybór posiłków i win w restauracji



Restauracja Laguna Tronka		
Seria Win		
	Lampa 200ml	Butełka
BIAŁE WINA POLSKIE		
Francja		
Chateau Bellevue	100	750
WINA Czerwone		
Francja		
Chateau Haut-Lalande	140	900
Argentyna		
Davidi Sierra / Cabernet	90	700
Hiszpania		
El Coto Crianza	100	700
Senorio de Santa Rita Crianza	110	800
Coto de Inca Rioja Reserva	160	980
Włochy		
Yardan Mount-Henon	140	900
USA		
Chateau	180	950
Czerwone wino		
	110	

Krok 1 - ustal dziedzinę i zakres ontologii

- Na jakiego rodzaju pytania ontologia ma udzielać odpowiedzi?
 - musi zawierać: pojęcia opisujące typy win, dobre i złe połączenia danego typu z danym jedzeniem
 - mało prawdopodobne: np. pojęcia nt. zarządzania magazynem w winiarni
 - Kto będzie korzystał z ontologii i ją utrzymywał?
 - ontologia służąca klientom restauracji powinna zawierać ceny detaliczne,
 - ontologia służąca właścicielom restauracji - ceny hurtowe
-

Krok 1 - ustal dziedzinę i zakres ontologii

Pytania kompetencyjne (ang. *competency questions*) – lista pytań, na które powinna odpowiedzieć oparta na ontologii baza wiedzy. Służą za test:

- Czy ontologia zawiera dostatecznie dużo informacji, aby na nie odpowiedzieć?
 - Czy odpowiedzi są wystarczająco szczegółowe?
-

Pytania kompetencyjne

- Jakie właściwości wina powinno się rozważyć przy jego wyborze?
- Czy Bordeaux to wino czerwone czy białe?
- Czy Cabernet Sauvignon pasuje do owoców morza?
- Jakie jest najlepsze wino do grillowanego mięsa?
- Czy bukiet zależy od rocznika?
- Jakie są dobre roczniki Napa Zinfandel?
- ...



Zgodnie z listą pytań ontologia powinna zawierać informacje o cechach i typach win, dobrych i złych rocznikach, rodzajach jedzenia, dla których ważny jest odpowiedni wybór pasującego wina, polecanych kombinacjach jedzenie – wino.

Krok 2 – rozważ użycie istniejących ontologii

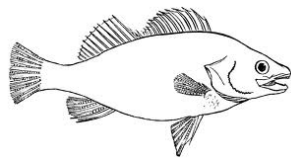
Czy istnieją ontologie dotyczące naszej dziedziny, które moglibyśmy rozszerzyć i zmodyfikować, tak by pasowały do naszych wymagań?

- ontologie nadrzędne/fundacjonistyczne (ang. top-level/upper/foundational ontologies)
 - repozytoria ontologii (TONES)
 - wyszukiwarki ontologii (Swoogle)
 - re-inżynieria zasobów nie-ontologicznych (istniejąca już baza francuskich win może stanowić nasze „pierwsze podejście” w tworzeniu klasyfikacji win)
-

Krok 3 - wylistuj najważniejsze terminy w ontologii

- **Najważniejsze terminy związane z daną dziedziną**
- **Na początku ważne aby lista terminów była obszerna i wyczerpująca, nawet jeśli niektóre pojęcia, relacje lub właściwości się pokrywają**

wino, winogrono, winiarnia, jej lokalizacja, kolor wina, smak i zawartość cukru; różne rodzaje jedzenia np. ryby czy wołowina; podtypy wina – wino białe, czerwone



Krok 4 - zdefiniuj klasy i ich hierarchię

- podejście **top-down**: najpierw zdefiniowanie najbardziej ogólnych pojęć w dziedzinie, a później ich specjalizacja
 - Zaczynamy od zdefiniowania ogólnych pojęć nt. jedzenia i wina, by później rozwinąć klasę Wino tworząc jej podklasy – Wino białe, Wino czerwone, Wino różowe
 - podejście **bottom-up**: najpierw zdefiniowanie najbardziej specyficznych klas – liści w hierarchii, by później grupować te klasy w bardziej ogólne pojęcia
 - Np. Zaczynamy definiując klasy *Pauillac* i *Margaux*. Później tworzymy ich wspólną nadklasę – *Medoc* (z kolei podklasę *Bordeaux*)
 - podejście **mieszane** – najpierw definiujemy bardziej istotne pojęcia, a potem tworzymy ich generalizacje i specjalizacje
-

Krok 4 - zdefiniuj klasy i ich hierarchię

- z listy terminów (krok 3) wybieramy odpowiadające rzeczywistym obiektom, a nie opisujące te obiekty (ich własności)
- układamy hierarchię klas - czy obiekt będący instancją jednej klasy, jest koniecznie (np. przez definicję) instancją innej klasy?
 - Jeżeli klasa A jest nadklasą B, wtedy każda instancja B jest też instancją A (np. Pinot noir podklasą Wino czerwone)



Kiedy wprowadzić nową klasę?

- Podklasa zazwyczaj zawiera:
 - dodatkowe własności, których nie ma nadklasa
 - dodatkowe ograniczenia inne niż dla nadklasy
 - bardziej specyficzne ograniczenia
- Synonimy nazwy pojęcia nie stanowią różnych klas
 - Klasy reprezentują pojęcia a nie ich nazwy!
 - Nazwa klasy może się zmienić, ale będzie reprezentować to samo pojęcie

(Koźlarz i Kozak)

- popularne jest użycie identyfikatorów numerycznych



Nowa klasa czy nowa wartość własności?

- Czy określone rozróżnienie (np. białe, czerwone, różowe wino) jest własnością czy zestawem klas?
 - Decyzja zależna od zakresu dziedziny oraz aplikacji
 - Czy nowa klasa ma dla nas istotne znaczenie?

W firmie produkującej butelki kolor wina nie ma większego znaczenia, więc nowe klasy mogły by być zbędne, inaczej było by w aplikacji ułatwiającej dobór wina do posiłku.



Krok 5 - zdefiniuj własności klas

Dla klasy Wino: kolor, smak, zawartość cukru

Dla klasy Winiarnia: położenie

- Możliwe typy własności:
 - W przypadku obiektów strukturalnych, ich fizyczne lub abstrakcyjne „części” (dania w posiłku)
 - Relacje klas z innymi obiektami
 - wytwórca wina – łączący wino i winiarnię oraz winogrono z którego wino jest zrobione
 - dla klasy Wino: nazwa, pochodzenie, wytwórca, winogrono
 - Własności są dziedziczone przez podklasy, dlatego powinny być związane z najbardziej ogólną klasą, która posiada daną własność
-

Krok 6 - zdefiniuj ograniczenia własności

- Liczność (ograniczenia ilościowe) – określa ile własność może mieć wartości
 - Wino jest zrobione z co najmniej $N=1$ gatunku winogron
 - wino wykonane z maksimum $M=1$ gatunku winogron
 - Własności literałowe
 - cena wina (String)
 - musujące? tak/nie (Boolean)
 - Dziedzina i przeciwdziedzina
 - zakresem własności „produkuje” jest klasa „Wino”
 - dziedziną własności „produkuje” jest klasa „Winiarnia”
-

Krok 7 – wprowadź instancje

- Zdefiniowanie pojedynczej instancji klasy:
 - Wybranie klasy
 - Stworzenie instancji klasy
 - Stworzenie ograniczeń własności

Chateau-Morgon-Beaujolais jest instancją klasy Beaujolais

Kolor: czerwone

Smak: delikatny

Poziom taniny: niski

Winogrono: Gamay (instancja klasy Winogrono)

Wytwórca: Chateau-Morgon (instancja klasy Winiarnia)

Region: Beaujolais (instancja klasy Region)

Zawartość cukru: wytrawne

Klasa czy instancja?

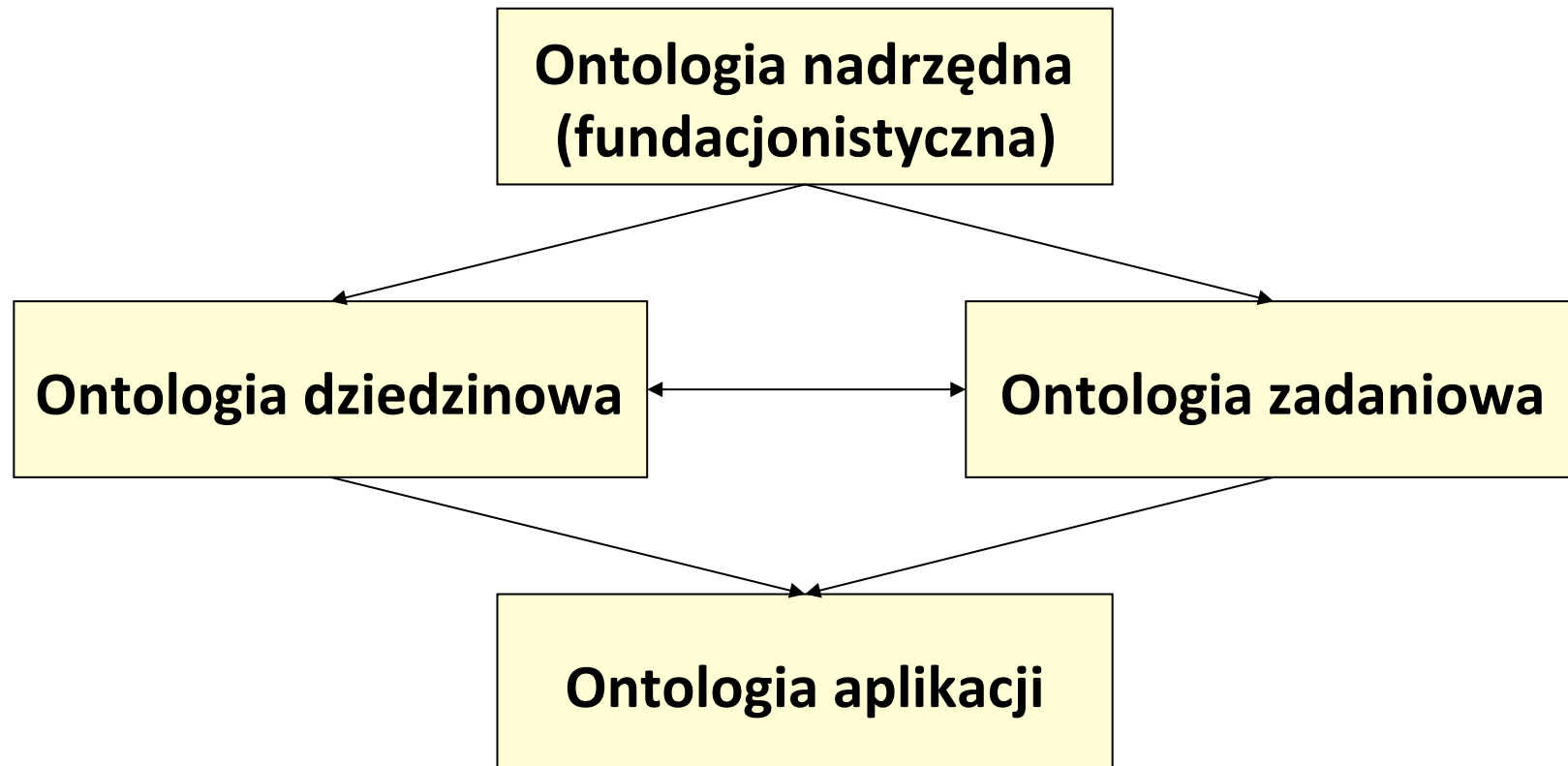
- **Jaki powinien być poziom granularności w ontologii? Jakie są najbardziej specyficzne obiekty? (będą one reprezentowane jako instancje)**
 - **Aplikacja doboru wina do posiłku: instancją będzie np. Sterling Vineyards Merlot**
 - **Aplikacja do inwentaryzacji win w restauracji: instancjami będą pojedyncze butelki wina**
 - **Pojęcia, które formują naturalną hierarchię - reprezentowane jako klasy**
 - w trakcie rozwoju ontologii, niektóre gałęzie pojęć w hierarchii mogą zyskać kolejne poziomy szczegółowości, a ich instancje zostać redefiniowane jako podklasy
 - **Pojęcia, dla których nie możemy sobie wyobrazić ich instancji – modelowane jako instancje (np. nazwy własne)**
 - Niekiedy, w związku z wymogami aplikacji, można zamodelować klasę mającą pojedynczą instancję
-

Adnotacje

- **Klasy/relacje powinny posiadać dokumentację w postaci adnotacji**
 - Opis klasy/relacji w języku naturalnym
 - Lista założeń dziedzinowych odnoszących się do definicji klasy/relacji
 - Lista synonimów
 - **Dokumentacja równie ważna jak ta w przypadku dokumentowania kodów źródłowych programu!**
-

Które istniejące ontologie wykorzystać i jak?

Typy ontologii



Które istniejące ontologie wykorzystać i jak?

- Ontologie fundacjonistyczne dostarczają ogólnych wysokopoziomowych zbiorów kategorii
 - pomoc na starcie budowy ontologii (podstawowe rozróżnienia – obiekty stałe w czasie a procesy, podstawowa struktura, podstawowe relacje)
 - Ontologie dziedzinowe
 - częściowo pokrywają się z zakresem nowo tworzonej ontologii
-

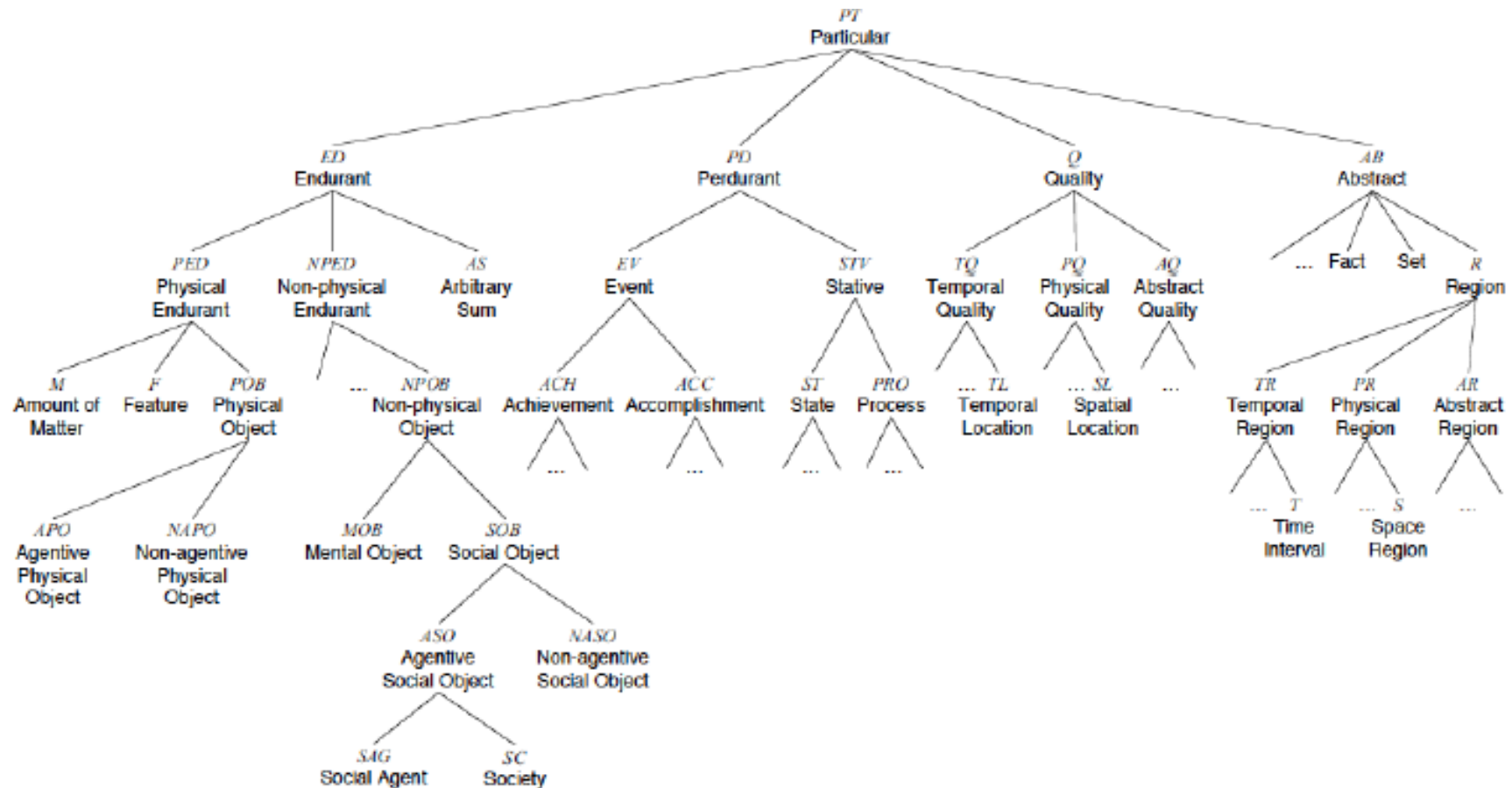
Ontologie fundacjonistyczne

- **DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering)**
 - obiekty i pojęcia związane z językiem naturalnym i kognitywistyką
 - **BFO (Basic Formal Ontology)**
 - zorientowana w kierunku badań naukowych, zastosowania w bioinformatyce
 - **SUMO (Suggested Upper Merged Ontology)**
 - przewidziana dla wielu zastosowań (wyszukiwanie, lingwistyka, wnioskowanie), zawiera zbiór ontologii dziedzinowych
 - **Cyc/Open Cyc**
 - duży projekt z zakresu Sztucznej Inteligencji, w ramach którego tworzona jest ontologia i baza wiedzy dotycząca życia codziennego
-

Typowe ontologiczne podziały klas

- **encje fizyczne a abstrakcyjne**
 - encje fizyczne potrzebują czasu lub przestrzeni żeby istnieć, pozostałe to encje abstrakcyjne
 - **encje „trwające” (ang. Continuant, Endurant) - obiekty stałe w czasie a „wydarzające się” (ang. Occurrent, Perdurant) – procesy**
 - encje „trwające” - postrzegane jako całościowy (kompletny) obiekt w dowolnym wycinku czasu
 - encje „wydarzające się” – w danym wycinku czasu, istnieje jedynie ich fragment
-

Taksonomia głównych kategorii DOLCE



Taksonomia BFO

bfo:Entity

 snap:Continuant

 snap:DependentContinuant

 snap:GenericallyDependentContinuant

 snap:SpecificallyDependentContinuant

 snap:Quality

 snap:RealizableEntity

 snap:Disposition

 snap:Function

 snap:Role

 snap:IndependentContinuant

 snap:MaterialEntity

 snap:Object

 snap:FiatObjectPart

 snap:ObjectAggregate

 snap:ObjectBoundary

 snap:Site

 snap:SpatialRegion

 snap:ZeroDimensionalRegion

 snap:OneDimensionalRegion

 snap:TwoDimensionalRegion

 snap:ThreeDimensionalRegion

span:Occurrent

 span:ProcessualEntity

 span:Process

 span:ProcessBoundary

 span:FiatProcessPart

 span:ProcessAggregate

 span:ProcessualContext

 span:SpatiotemporalRegion

 span:ConnectedTemporalRegion

 span:SpatiotemporalInstant

 span:SpatiotemporalInterval

 span:ScatteredSpatiotemporalRegion

 span:TemporalRegion

 span:ConnectedSpatiotemporalRegion

 span:TemporalInstant

 span:TemporalInterval

 span:ScatteredTemporalRegion

Wzorce projektowe w konstrukcji ontologii

- analogia do wzorców projektowych w programowaniu
 - **wzorzec projektowy dla ontologii (ang. ontology design pattern - ODP):**
 - podsumowuje najlepsze praktyki do stosowania w modelowaniu ontologii
 - jest rozwiązaniem powtarzającego się problemu modelowania w konstrukcji ontologii
-

Wzorce projektowe w konstrukcji ontologii

- moduły (małe ontologie):
 - wykorzystanie tylko adekwatnych fragmentów już istniejącej ontologii
 - dobra dokumentacja
 - najlepsze praktyki re-inżynierii wiedzy
-

<http://ontologydesignpatterns.org>

Ontology Design Patterns . org (ODP)

OntologyDesignPatterns.org is a Semantic Web portal dedicated to ontology design patterns (ODPs). The portal was started under the [NeOn project](#), which still partly supports its development.




What's new

- The 2nd [Workshop on Ontology Patterns](#) to be held on November 8, in conjunction with ISWC2010. **Submission deadline extended to September 1st!**
- [eXtreme Design camp in Bologna](#)

Navigation

 [List of Patterns](#)
You can find lists here, detailing all available ontology design patterns.

Contribute

 [Submit Pattern](#)
Start here if you want to submit an ontology pattern.

 [Post Modeling Issue](#)

News

 [Latest ODP News!](#)
[2nd Workshop on Ontology Patterns \(WOP\) accepted at ISWC 2010!](#)

Navigation

- Main page
- List patterns
- Pattern types
- Modeling Issues
- Domains
- Training
- Events

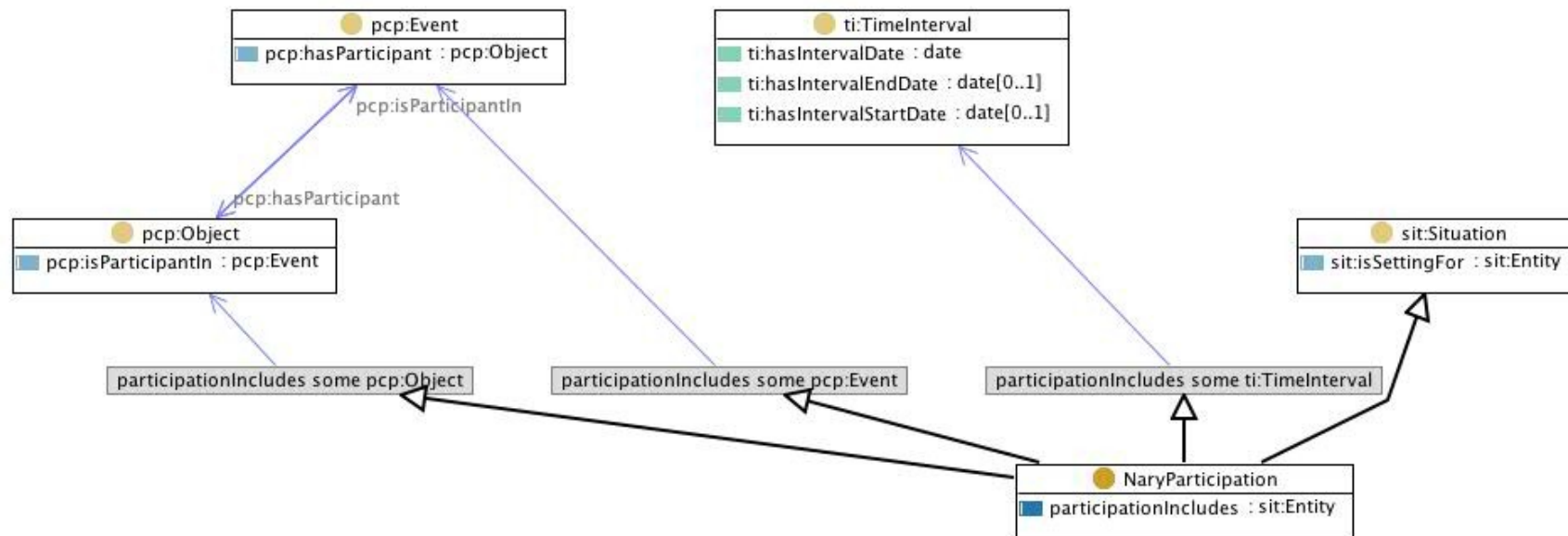
contribute

- Submit a pattern
- Submit an exemplary ontology
- Post a modeling issue
- Review a pattern
- Feedback about the portal
- Request an ODP

ontology design patterns . org (odp) | discussion | view source | history

Log in

Przykład: „n-ary participation” (graficzna reprezentacja)



Przykład: „n-ary participation” (ogólny opis)

General description

Name:	Nary Participation
Submitted by:	Aldo Gangemi
Also Known As:	
Intent:	To represent events with their participants, time, space, etc.
Domains:	General
Competency Questions:	<ul style="list-style-type: none">■ What are the participants in that event at this time?■ What events had what participants in that location? ---
Solution description:	
Reusable OWL Building Block:	http://www.ontologydesignpatterns.org/cp/owl/naryparticipation.owl (40)
Consequences:	All sorts of relations denoting events with multiple participants, space-time indexing, etc. can be represented with this pattern. When objects participate at the event at different times or with different parts, more elementary nary-participation instances must be created, and made parts of the main one.
Scenarios:	The match lasted 95 minutes, and the winner used a new nano-carbon racquet.
Known Uses:	
Web References:	
Other References:	
Examples (OWL files):	<ul style="list-style-type: none">■ http://www.ontologydesignpatterns.org/cp/examples/naryparticipation/naryparticipationex.owl #
Extracted From:	<ul style="list-style-type: none">■ http://www.iaa.cnr.it/ontologies/DUL.owl #
Reengineered From:	
Has Components:	<ul style="list-style-type: none">■ Submissions:Participation■ Submissions:TimeInterval
Specialization Of:	<ul style="list-style-type: none">■ Submissions:Situation
Related CPs:	<ul style="list-style-type: none">■ Submissions:Co-participation■ Submissions:Participation

Inne metodologie

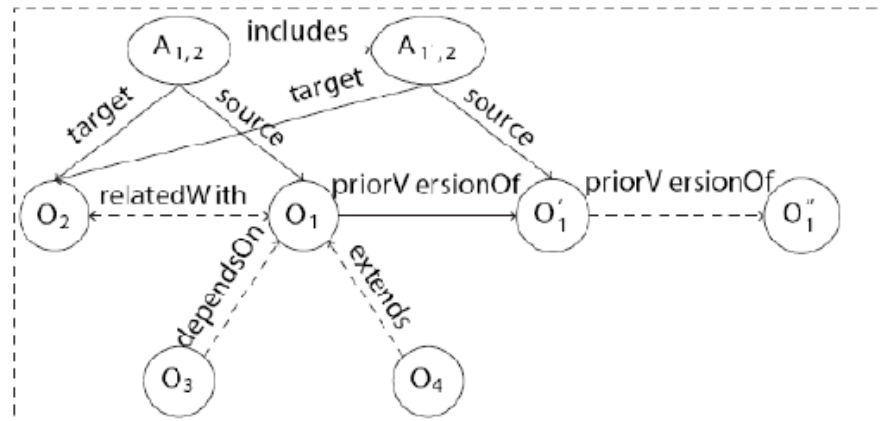
- **konstrukcja pojedynczej ontologii**

- Methontology
- On-To-Knowledge
- KACTUS

- **konstrukcja „sieci ontologii”**

- NEON:

duża liczba ontologii,
osadzonych w sieciach ontologii,
budowanych przez rozproszone,
współpracujące ze sobą grupy



Metodologia NEON

- **wykaz czynności** („Glossary of Activities,„) identyfikuje i definiuje ponad 50 czynności wykonywanych podczas kolaboratywnego budowania sieci ontologii
 - *-formalizacja, -diagnostyka, -lokalizacja, -dopasowywanie, -wzbogacanie* itp.
 - nacisk na:
 - **ponowne użycie i reinżynierię zasobów zawierających wiedzę dziedzinową**
 - **kolaboratywny i dyskusyjny tryb pracy nad tworzeniem ontologii**
 - **tworzenie *sieci ontologii*, w przeciwieństwie do tworzenia dedykowanych, nowych ontologii „od zera”**
-

Metodologia NEON

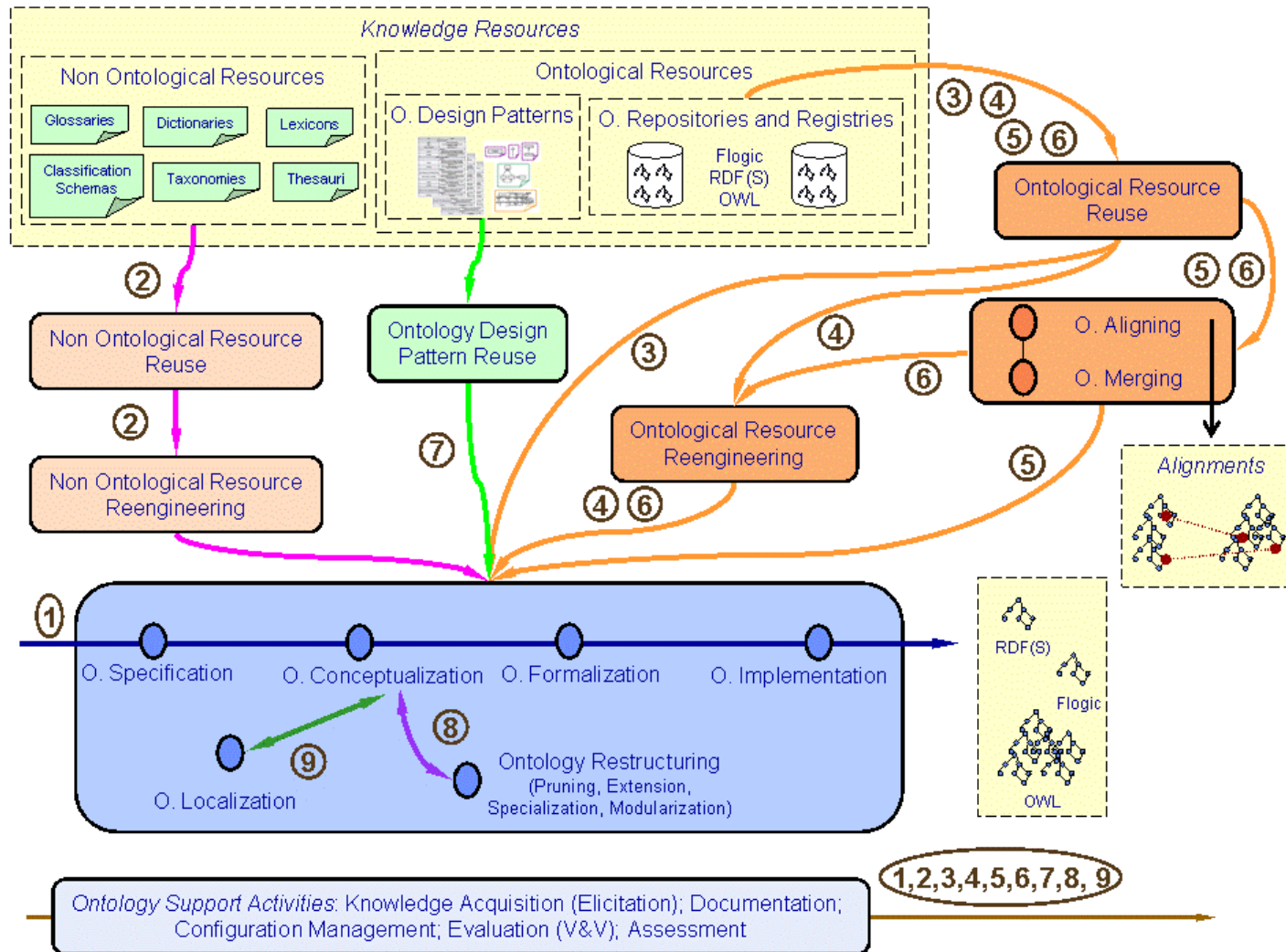
Dokument ze Specyfikacją Wymagań dla Ontologii

1. cel
 2. zakres (np. jednostki osadnicze z okresu średniowiecza)
 3. język implementacji (np. OWL)
 4. użytkownicy końcowi (np. doktorant w dziedzinie historii, nauczyciel w szkole podstawowej itd.)
 5. scenariusze użycia (np. wyszukiwanie jednostek danego typu)
 6. wymagania nie-funkcjonalne (np. modularyzacja, użycie ontologii fundacjonistycznej, re-użycie wybranego zasobu jak Dublin Core)
 7. wymagania funkcjonalne: pytania kompetencyjne
 8. wstępny "glosariusz" pojęć (na podstawie pytań kompetencyjnych i odpowiedzi)
-

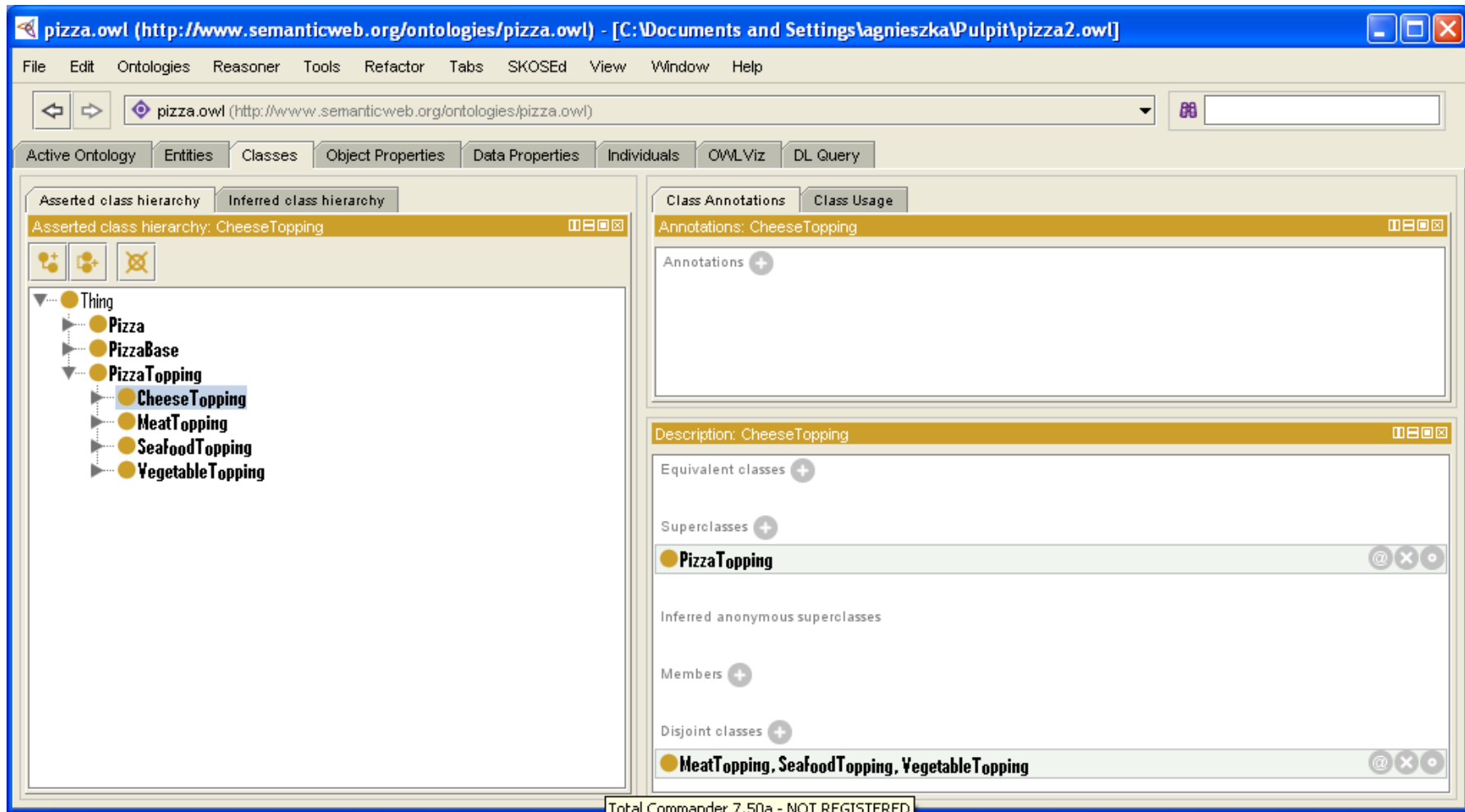
Wykorzystanie nie-ontologicznych zasobów

- standardy metadanych
 - schematy danych w istniejących systemach (np. schematy baz danych)
 - tezaury, słowniki, leksykony
 - folksonomie
 - itd.
-

NEON - scenariusze budowy ontologii



Narzędzia - Protégé



Narzędzia – NEON Toolkit



The screenshot displays the NeOn Toolkit interface with the following components:

- Ontology Navigator:** A tree view on the left showing a project named "Demo 2.1 [FLogic]" with various URIs and files like "beer.rdfs".
- Mapping Editor:** The central workspace shows two ontologies being mapped. The left ontology (from "http://www.CarDemoDB.com") has a class "cars" with properties "cars_name" and "prices_car". The right ontology (from "http://www.CarDemoIntegration") has a class "car" with properties "hasName" and "Properties from product".
- Visual Diagrams:** Two diagrams illustrate the mapping. The top diagram shows a one-to-one mapping between "cars" and "car". The bottom diagram shows a many-to-many mapping between "cars" and "car".
- Attributes/Relations:** Two panels below the diagrams show the selected classes and their properties for each ontology.
- Instances:** Two tables at the bottom show instance data for the mapped classes.
- Transformations:** A panel at the bottom center explains the transformation needed for the "hasName" property: "From 'cars_name(' to the value of 'hasName' the following transformation is necessary:" with radio buttons for "mathematical transformation", "string transformation", and "other transformation".

instance	cars_name(prices_car

instance	hasName

Narzędzia – Top Braid Composer

The screenshot displays the TopBraid Composer interface within the Eclipse SDK. The main workspace shows a semantic network graph with three nodes: Holger Knublauch, Ralph Hodgson, and Dean Allemang. Holger Knublauch is connected to both Ralph Hodgson and Dean Allemang via 'knows' relationships. The interface includes several panels:

- Classes:** A tree view showing the ontology structure, including `owl:Thing` (29), `foaf:Agent` (25), `foaf:Group`, `foaf:Organization`, `foaf:Person` (25), `foaf:Document` (3), `foaf:Image` (1), `foaf:OnlineAccount`, `foaf:Project`, `geo:SpatialThing` (25), `foaf:Person` (25), `geo:Point`, and `owl:Nothing`.
- Properties:** A list of properties for the selected resource, including `homepage` (`<http://www.knublauch.com>`), `is primary topic of` (`<http://www.knublauch.com/foaf.rdf>`), `knows` (with instances `dalleman:me` and `rhodgson:me`), `made` (`<http://www.knublauch.com/foaf.rdf>`), `past project` (`<http://protege.stanford.edu>`), `weblog` (`<http://composing-the-semantic-web.blogspot.com/>`), `work info homepage` (`<http://www.topquadrant.com/topbraidsuite.html>`), and `workplace homepage` (`<http://www.topquadrant.com/>`).
- SPARQL Query Editor:** A query editor showing a SPARQL query:

```
SELECT ?person ?worksAt
WHERE { ?person foaf:workplaceHomepage ?worksAt }
```

 Below the query, a table displays the results:

[person]	worksAt
dalleman:me	<http://www.topquadrant.com>
rhodgson:me	<http://www.topquadrant.com>
- Geography:** A map view showing the location of the resources. A red pin is placed on a map of the area around S De Anza Blvd and John Dr. The map includes a search bar and a 'Search' button.

Narzędzia - FluentEditor

The screenshot displays the FluentEditor interface with the following components:

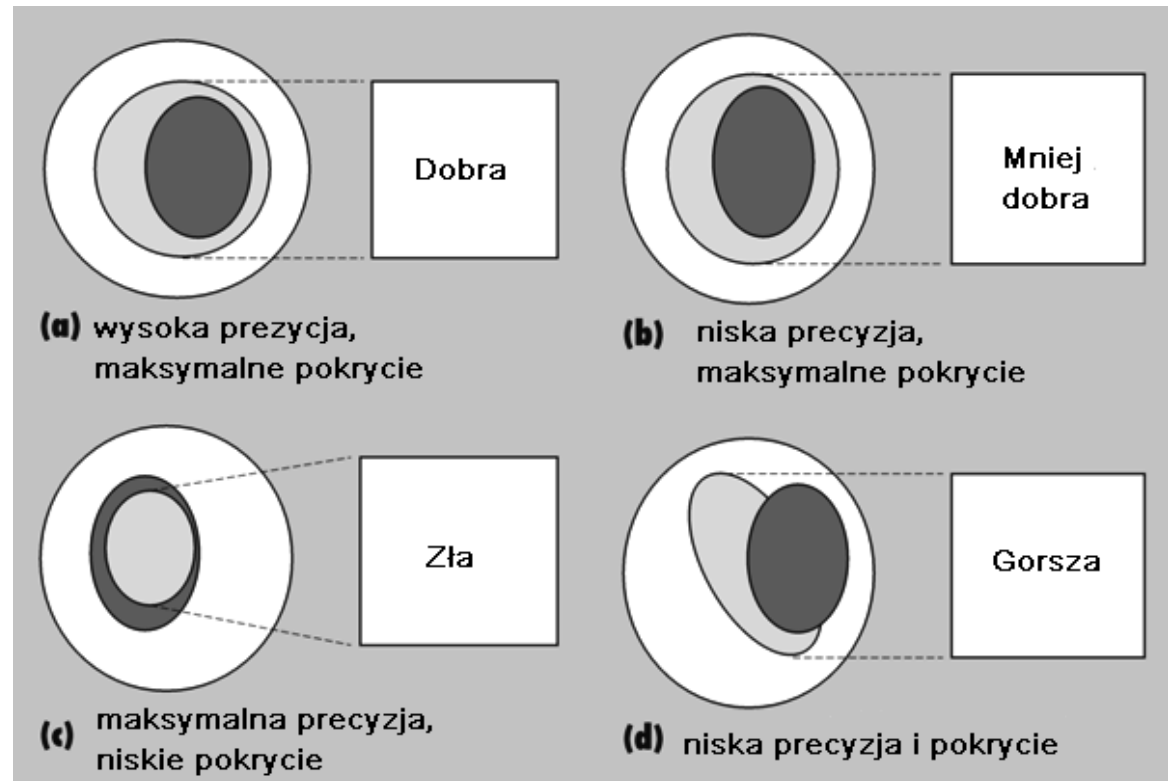
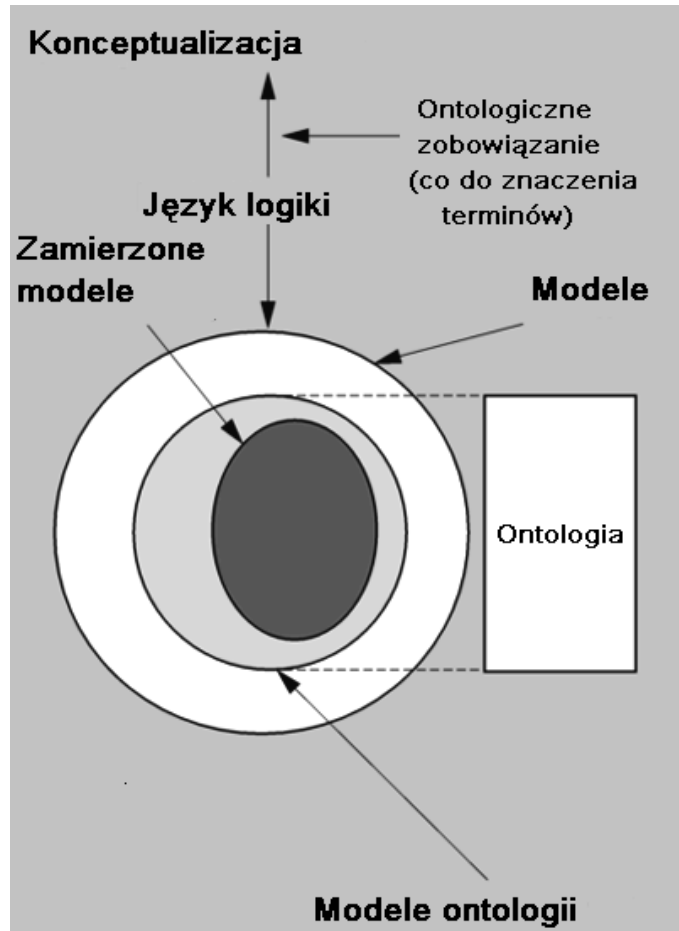
- Document:** A text editor showing logical rules and comments. A red box highlights the following text:

```
114 No herbivore eats animal and-or eats thing that has-part animal.  
115 Tasty-Pizza is pizza and has-part an animal .  
116 Vegan-Pizza is pizza and has-part a plant.  
117 Sophie eats Tasty-Pizza.
```
- Reasoner (HermiT):** A window for executing queries. A red box highlights the query input field containing "Who-Or-What is Sophie?" and an adjacent "Enter" button.
- Taxonomy Tree:** A hierarchical tree structure on the right side of the interface. The root node is "thing", which branches into "animal", "pizza", and "plant". The "animal" node further branches into "carnivore", "giraffe", "herbivore", "lion", and "omnivore". The "herbivore" node branches into "elephant" and "impala". The "lion" node branches into "Leo". The "pizza" node branches into "carnivorous-plant", "grass", "palm-tree", "tasty-plant", and "tree". The "plant" node branches into "carnivorous-plant", "grass", "palm-tree", "tasty-plant", and "tree". The "plant-part" node branches into "branch" and "fruiting-body". The "fruiting-body" node branches into "apple".

Dodatkowe slajdy



Ocena jakości ontologii – pokrycie i precyzja



Ocena jakości ontologii - metoda OntoClean

- **Problem:** nieuporządkowane taksonomie pojęć względem „co jest czyją podklasą”, jak ułożyć pojęcia we właściwym porządku?
 - Metoda **OntoClean** (Guarino & Welty, 2000) – analiza ontologii na podstawie formalnych, niezależnych od dziedziny metawłasności
 - Pozwala na identyfikację konfliktów logicznych
-

Metawłasności OntoClean

- **tożsamość** (ang. *identity*) – obiekt opisany taką własnością pozostaje wedle niej tym samym obiektem mimo możliwego zmieniania się (np. jednoznaczny identyfikator zasobu)

Jan Kowalski ✓

- **integralność** (ang. *unity*) – możliwość rozpoznania granic i wszystkich części, które składają się na encję

Ocean (np. Ocean Atlantycki) ✓ Woda ✗

- **niezmiennność** (ang. *rigidity*) – zasadnicza własność dla wszystkich instancji

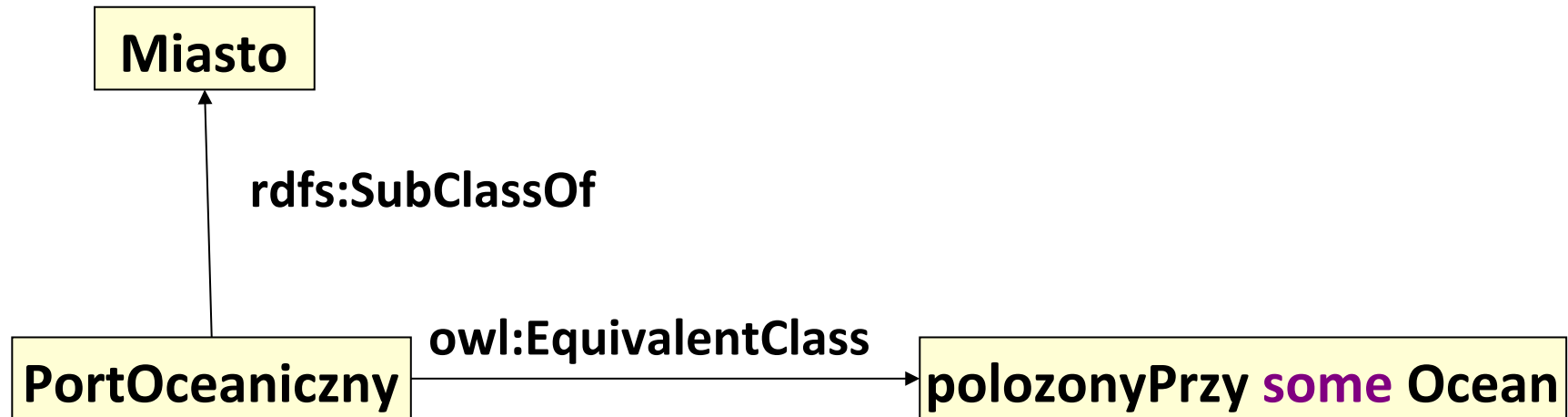
Jan jest osobą ✓

Jan jest studentem ✗

- **zależność** (ang. *dependence*) – każda instancja implikuje istnienie innej instancji, jedna nie istnieje bez drugiej

bycie uczniem (musi istnieć nauczyciel) ✓

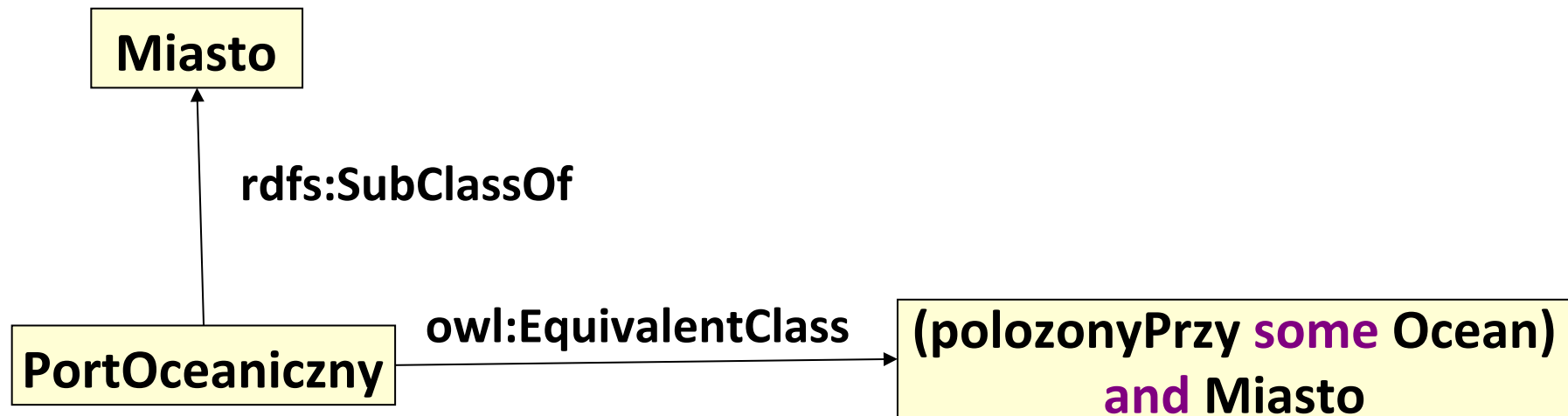
Złe praktyki w modelowaniu ontologii Domyślne wykluczanie innych opcji 1/2



ObjectPropertyAssertion(polozonyPrzy Francja OceanAtlantycki)
ClassAssertion(OceanAtlantycki Ocean)

Francja jest miastem??

Złe praktyki w modelowaniu ontologii Domyślne wykluczanie innych opcji 2/2



ObjectPropertyAssertion(polozonyPrzy Francja OceanAtlantycki)
ClassAssertion(OceanAtlantycki Ocean)

OK

Złe praktyki w modelowaniu ontologii

Za daleko idące modelowanie

- **Próba modelowania wszystkiego związanego z danym obiektem (nawet jeśli nie wskazują na to wymagania)**
 - **Utrudnienie innym wykorzystania modelu, który wychodzi poza dany temat (możliwe konflikty)**
 - **Uwaga na klasy bez ograniczeń własności i bez instancji – czy są potrzebne?**
-