

Inżynieria ontologii

Mikołaj Morzy
Agnieszka Ławrynowicz

Instytut Informatyki
Poznań, rok akademicki 2012/2013

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; dostarczają wskazówek i rekomendacji twórcom ontologii.

Po co tworzyć ontologie?

- **współdzielone rozumienie struktury informacji**
 - między ludźmi, między programami (agentami)
 - ułatwienie jej automatycznego przetwarzania
- **ponowne użycie wiedzy dziedzinowej**
 - uniknięcie ponownego „odkrywania Ameryki”
 - wprowadzenie standardów
- **jawna reprezentacja założeń przyjętych w danej dziedzinie**
 - pozwala na ich łatwą zmianę, w odróżnieniu od ich sztywnego zapisu w kodzie
- **oddzielenie wiedzy dziedzinowej od wiedzy operacyjnej**

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

Nie istnieje żadna „jedyna poprawna” metoda wytwarzania ontologii.

Ogólne reguły:

1. Możliwe wiele sposobów modelowania danej dziedziny. Najlepsze rozwiązanie zależy od tworzonej aplikacji i przewidywanych rozszerzeń.
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)

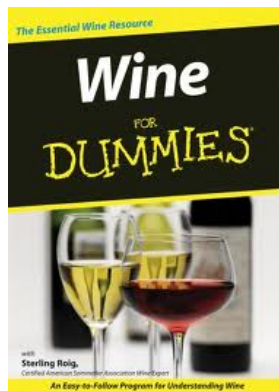
Budowa ontologii realizowana w cyklicznych etapach obejmujących:



Przykład – budowa ontologii o winach i jedzeniu



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 Ligonia Trzebnica		
	Karta Wina	
	Lampka 100ml	Bufla
BIŁE WINA POLSKIE		
Francia		
Chateau Bellevue	10zł	75zł
WINA Czerwone		
Francia		
Cateau Haut Laborde	14zł	90zł
Argentyna		
Quada Sierra / Cabernet	9zł	70zł
Hiszpania		
El Coto Crianza	10zł	70zł
Senorio de Santa Rita Crianza	13zł	80zł
Coto de Imaz Rioja Reserva	16zł	98zł
Włochy		
Hardan Mourmelon	14zł	90zł
Węgry		
Classique	18zł	99zł
Crème de vin	13zł	

Krok 1 - ustal dziedzinę i zakres ontologii

- Na jakiego rodzaju pytania ontologia ma udzielać odpowiedzi?
 - **musi zawierać: pojęcia opisujące rodzaje i główne typy win, dobre i złe połączenia danego typu wina z danym jedzeniem**
 - **mało prawdopodobne: pojęcia nt. zarządzania magazynem w winiarni lub nt. pracowników restauracji**
- Kto będzie korzystał z ontologii i ją utrzymywał?
 - **ontologia służąca:**
 - a) klientom restauracji - powinna zawierać ceny detaliczne,
 - b) 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.

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

Pytania kompetencyjne

- Jakie własności 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

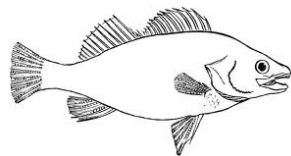
Czy istnieją ontologie dotyczące naszej dziedziny, które moglibyśmy rozszerzyć i zmodyfikować, tak by pasowały do naszych wymagań?

- **ontologie nadrzędne/fundacjonistyczne (ang. top-level/upper/foundational ontologies)**
- **repozytoria ontologii (TONES)**
- **wyszukiwarki ontologii (Swoogle)**
- **zasoby nie-ontologiczne (baza francuskich win może już istnieć, taka zaimportowana baza może stanowić nasze „pierwsze podejście” w tworzeniu klasyfikacji win) + ich re-inżynieria**

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
 - np. najpierw klasa *Wino*, później 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. najpierw klasy *Pauillac* i *Margaux*, później ich wspólna nadklasa – *Medoc* (z kolei podklasa *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
 - Jeżeli klasa A jest nadklasą B, wtedy każda instancja B jest też instancją A (np. Pinot noir to podklasa Wino czerwone)



Krok 4 - zdefiniuj klasy i ich hierarchię

- Relacja „jest” lub „jest rodzajem” (ang. **is-a**) - główna relacja obowiązująca w hierarchii klas (**Chardonnay jest rodzajem Wina Białego, Odrzutowiec jest rodzajem samolotu**)
- Przechodniość relacji hierarchicznych
 - **jeśli B jest podklasą A i C jest podklasą B, to C jest podklasą A**
 - **Unikanie cykli – Jeśli klasa B jest zdefiniowana jako podklasa A, oraz A jako podklasa B, to obie klasy są sobie równoważne**

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)



Nowa klasa czy nowa wartość własności?

- Czy określone rozróżnienie (np. białe, czerwone, różowe wino) jest własnością czy zestawem klas?
 - **Decyzja zależy od dziedziny oraz zadania jakie ma spełniać ontologia**
 - **Czy nowa klasa ma dla nas istotne znaczenie?**

W firmie produkującej butelki kolor wina nie ma większego znaczenia (nowe klasy są 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
 - **przeciwdziedzina własności „produkuje” jest klasa „Wino”**
 - **dziedzina 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?

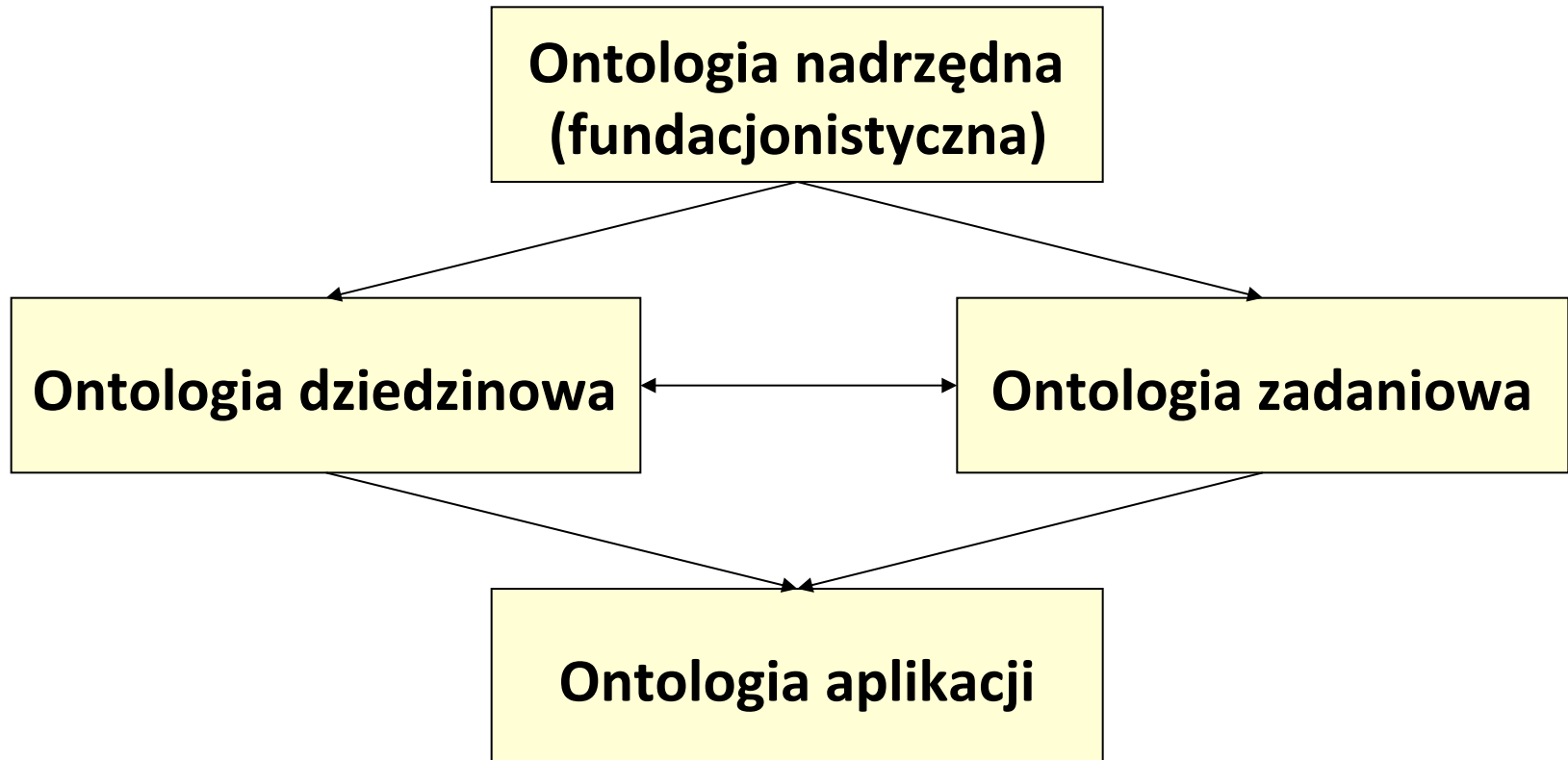
- **Jaki powinien być poziom granularności w ontologii? Jakie są najbardziej specyficzne obiekty? (będą one reprezentowane jako instancje)**
 - **Aplikacja doboru wina do posiłku: instancja to np. Sterling Vineyards Merlot**
 - **Aplikacja do inwentaryzacji win w restauracji: instancje to pojedyncze 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**

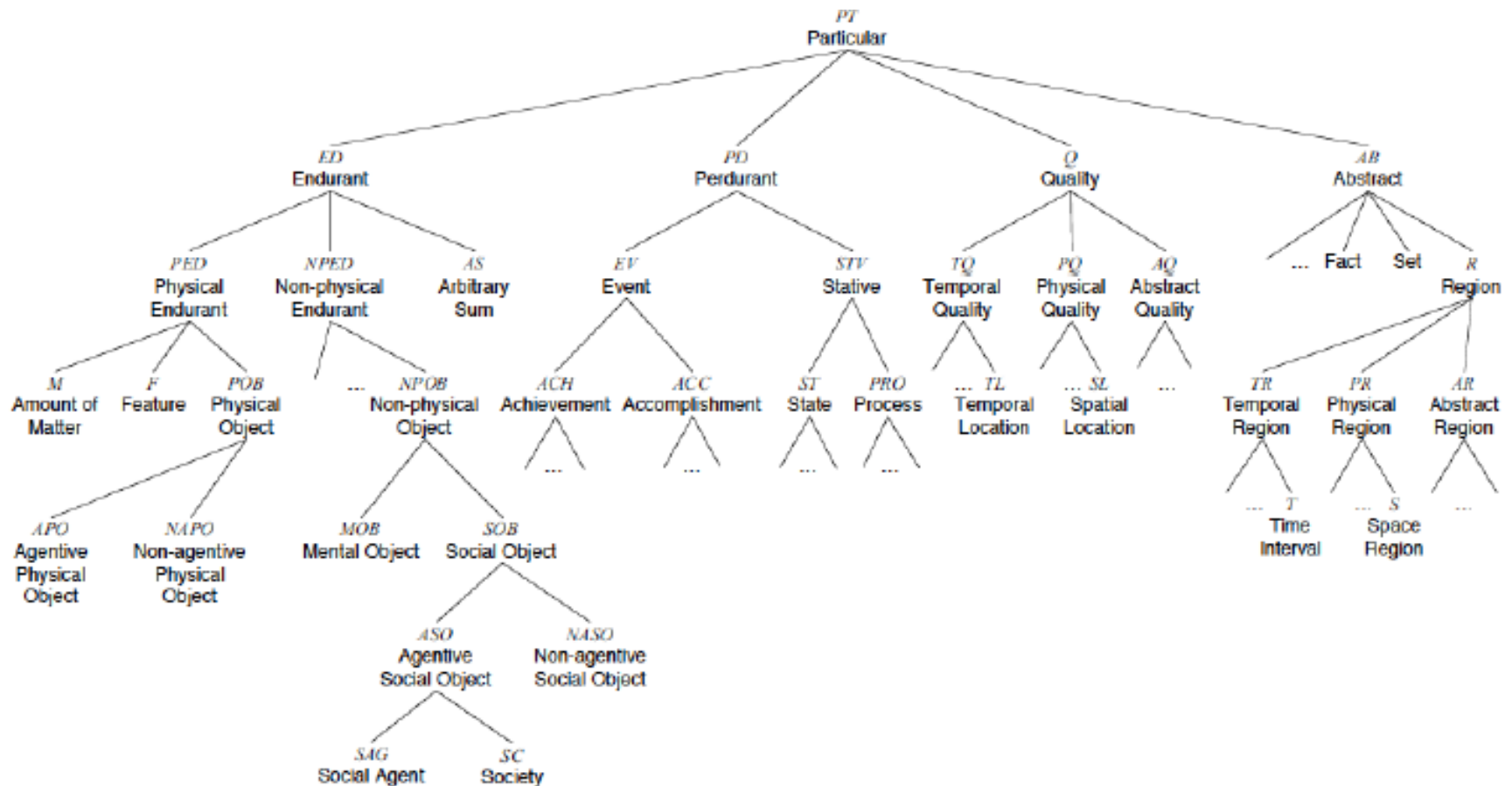
Ontologie fundacjonistyczne

- **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

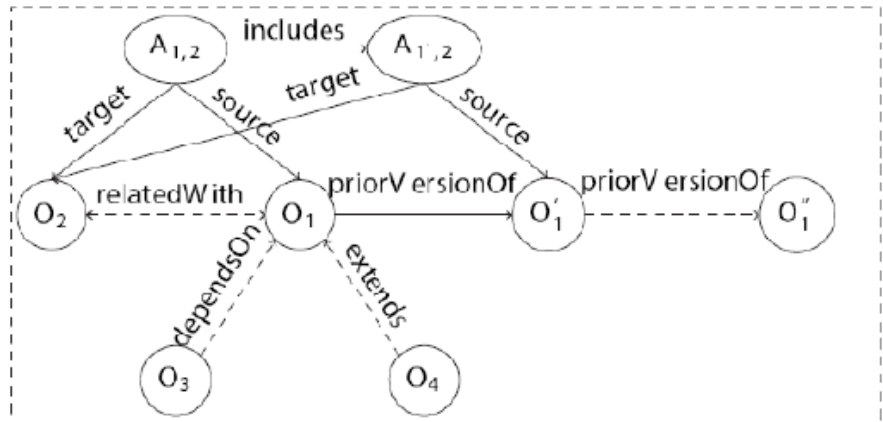
 span:ScatteredTemporalRegion

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 wiedzą dziedzinową**
 - **kolaboratywny i dyskusyjny tryb pracy nad tworzeniem ontologii**
 - **tworzenie *sieci ontologii*, w przeciwieństwie do tworzenia dedykowanych, nowych ontologii „od zera”**

NEON: specyfikacja wymagań

Ontology Requirements Specification Document (ORSD):

- cel ontologii
- zakres ontologii
- język implementacji
- przewidziani użytkownicy
- przypadki użycia
- ponowne wykorzystanie zasobów ontologicznych

Przykład: **Digital Multimedia Repositories Ontology**

http://129.194.69.119/public/dmro/DMRO_ORSD.pdf



NEON: wykorzystanie istniejących zasobów

- ontologie nadrzędne
- wyrażenia z innych ontologii
- wzorce projektowe
- zasoby nie-ontologiczne



Wyszukiwanie wyrażeń w innych ontologiach

Wtyczka Watson do NEON Toolkit

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

- Ontology Navigator:** A tree view showing the ontology structure. The 'DigitalResource' class is highlighted under the 'ore:AggregatedResource' class.
- Entity Properties:** A panel showing the identifier for the selected entity: `<http://www.e-lico.eu/ontologies/dmro/DMRO-AgentRole#Participant>`. It also includes an 'Annotations' table with columns for 'Annotation Property', 'Value', 'Type', and 'Language'. Below the table is a 'Create new:' section with input fields.
- Watson Results View:** A panel showing search results for 'Digital-Resource'. The results are displayed in a table with columns for the property name, the range, and an 'Add relation' button.

Annotation Property	Value	Type	Language
Create new:			
<input type="text"/>	<input type="text"/>	<input type="text"/>	--

Property	Range	Action
subClassOf	InformationRealization	Add relation from Digital-Resource
isDigitallyReproducedBy	range	Add relation to Digital-Resource
Archive	subClassOf	Add relation to Digital-Resource
HexadecimalCode	subClassOf	Add relation to Digital-Resource
Slide	subClassOf	Add relation to Digital-Resource
ComputerBackup	subClassOf	Add relation to Digital-Resource
digitallyReproduces	domain	Add relation to Digital-Resource
Visualization	subClassOf	Add relation to Digital-Resource
View	subClassOf	Add relation to Digital-Resource
Image	subClassOf	Add relation to Digital-Resource



Wzorce projektowe w konstrukcji ontologii

- trudność w wykorzystaniu tylko użytecznych fragmentów istniejącej ontologii, nakład pracy może być wyższy niż stworzenie nowej ontologii „od zera”
- potrzeba małych (lub z ręcznie zmodularyzowanych) ontologii:
 - dobra dokumentacja
 - najlepsze praktyki re-inżynierii wiedzy
- analogia do wzorców projektowych w programowaniu
- **wzorzec projektowy dla ontologii (ang. ontology design pattern - ODP):**
 - podsumowuje najlepsze praktyki modelowania ontologii
 - jest rozwiązaniem powtarzającego się problemu modelowania w konstrukcji ontologii

<http://ontologydesignpatterns.org>

Log in



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 account](#)

help

- [About ODP](#)
- [What is a pattern?](#)
- [What is an exemplary ontology?](#)
- [How to post a pattern](#)
- [Training](#)

catalogues

- [Content ODPs](#)
- [Reengineering ODPs](#)
- [Alignment ODPs](#)
- [Logical ODPs](#)

[ontology design patterns . org \(odp\)](#) [discussion](#) [view source](#) [history](#)

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 3rd Workshop on Ontology Patterns (WOP2012) will be held at ISWC2012 (November 11 or 12) in Boston, US.
- A VoCamp-related event on ODPs was held on October 28-29 2011 in Paris
- Several ODP-related presentations at ISWC 2011, e.g. in the [Formal Ontologies and Patterns session](#)

Navigation

 **List of Patterns**
You can find lists here, detailing all available ontology design patterns.


 **Pattern types**
Ontology patterns are of several types. Here are details about pattern types and their taxonomy.

 **Domains**
Ontology patterns can cover, or be related to, a particular domain. Here is a list.


 **Modeling Issues**
See all loaded modeling issues. Modeling issues are linked to ontology patterns that solve a defined problem.

 **Training Area**


Contribute

 **Submit Pattern**
Start here if you want to submit an ontology pattern.

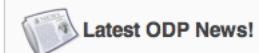
 **Post Modeling Issue**
If you have an unsolved modeling problem you wish to share with the community, post it here!

 **Submit an Exemplary Ontology**
Start here if you want to submit an exemplary ontology.

 **Post Review About a Pattern**
Review a submission to contribute to the certification process.

 **Post Your Feedback**
If you have issues about the web site, can't find information you need, or simply wish to propose

News



3rd Workshop on Ontology Patterns accepted for ISWC2012

29 May 2012 09:09:06 - by [EvaBlomqvist](#)

Pattern Camp - Paris

22 November 2011 08:08:23 - by [EvaBlomqvist](#)

ODP Seminar in Lim&Bio Lab - Paris 13 University - 16th June 2011

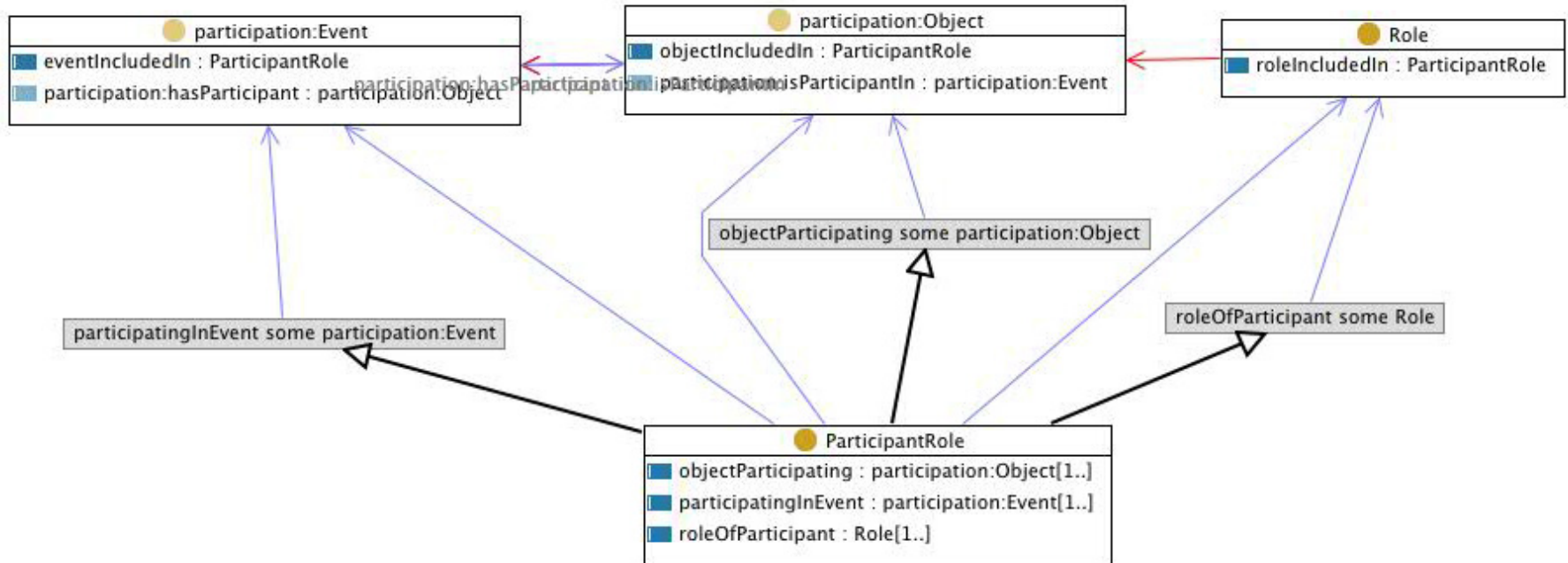
27 October 2011 23:23:04 - by [RimDJEDIDI](#)

2nd Workshop on Ontology Patterns (WOP) accepted at ISWC 2010!

29 May 2010 12:12:43 - by [EvaBlomqvist](#)

VoCamp in Paris - #vocampparis2010

Przykład: „participant role” (diagram)



Przykład: „participant role” (ogólny opis)

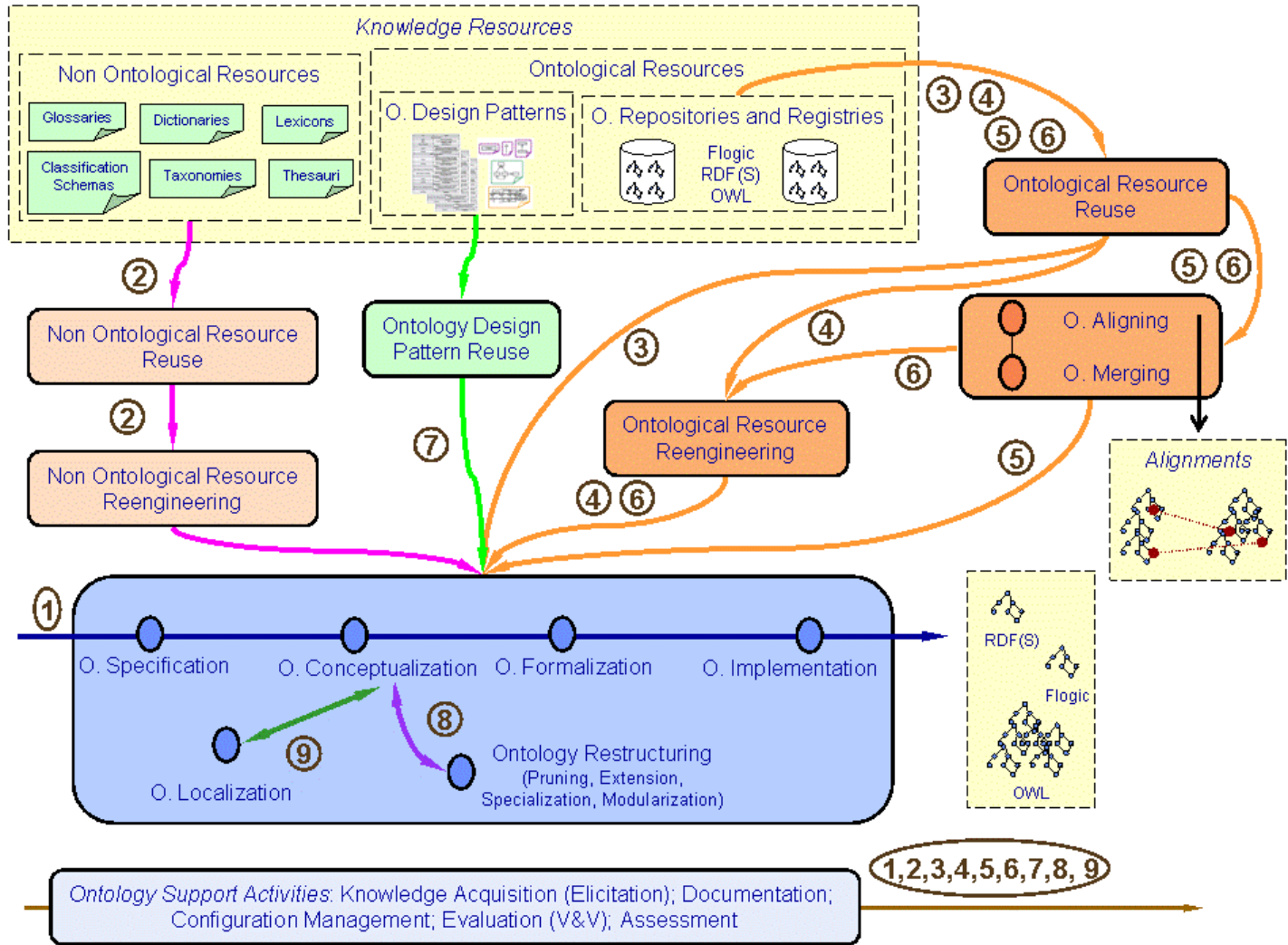
General description

Name:	ParticipantRole
Submitted by:	EvaBlomqvist
Also Known As:	
Intent:	To represent participants in events holding specific roles in that particular event.
Domains:	General , Organization
Competency Questions:	<ul style="list-style-type: none">■ What is the role of this object in this event?■ What is the object holding this role in this event?■ In what event did this object hold this role?
Solution description:	The pattern introduces a situation that connects the object to its role in a particular event.
Reusable OWL Building Block:	http://www.ontology.se/odp/content/owl/ParticipantRole.owl (73)
Consequences:	This pattern does not take into account time aspects of the participation, for such aspects see the timeindexedparticipation-pattern .
Scenarios:	John was elected the meeting secretary of today's board meeting. During the party we used the blue cup as a vase.
Known Uses:	
Web References:	
Other References:	
Examples (OWL files):	
Extracted From:	
Reengineered From:	
Has Components:	<ul style="list-style-type: none">■ Submissions:Situation■ Submissions:Participation
Specialization Of:	<ul style="list-style-type: none">■ Submissions:Participation
Related CPs:	<ul style="list-style-type: none">■ Submissions:Time indexed participation

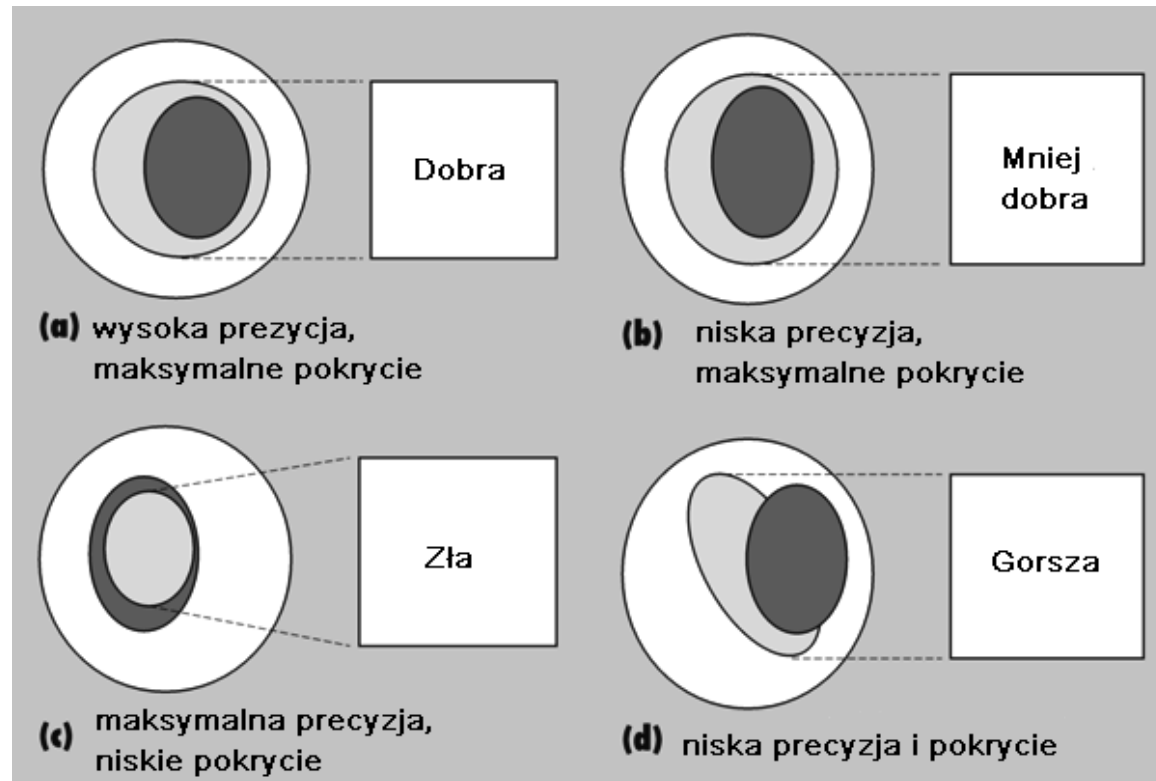
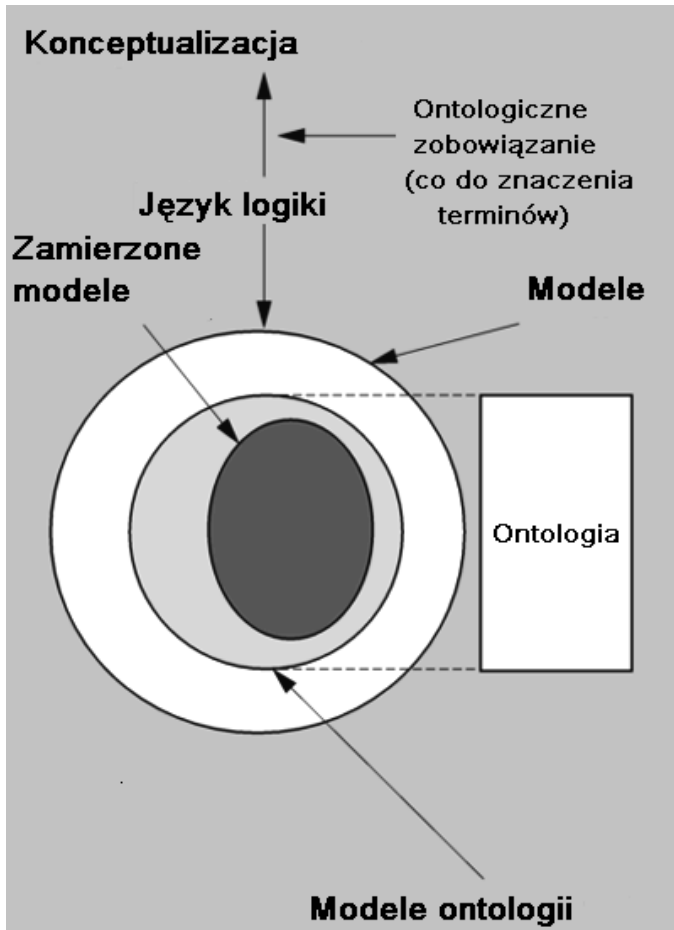
Wykorzystanie nie-ontologicznych zasobów

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

NEON - scenariusze budowy ontologii



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) ✓

Narzędzia - Protégé

The screenshot displays the Protégé ontology editor window for the 'pizza.owl' ontology. The interface includes a menu bar (File, Edit, Ontologies, Reasoner, Tools, Refactor, Tabs, SKOS Ed, View, Window, Help) and a toolbar with navigation and search icons. The main workspace is divided into several panes:

- Active Ontology:** Shows the current ontology file: 'pizza.owl (http://www.semanticweb.org/ontologies/pizza.owl)'. Below this are tabs for 'Entities', 'Classes', 'Object Properties', 'Data Properties', 'Individuals', 'OVLViz', and 'DL Query'.
- Class Hierarchy:** A tree view showing the 'Asserted class hierarchy' for 'CheeseTopping'. The hierarchy is: Thing (parent) -> Pizza (child) -> PizzaBase (child) -> PizzaTopping (child) -> CheeseTopping (child). Other children of PizzaTopping are MeatTopping, SeafoodTopping, and VegetableTopping.
- Class Annotations:** A pane titled 'Annotations: CheeseTopping' showing a list of annotations for the selected class.
- Description:** A pane titled 'Description: CheeseTopping' showing the class's properties:
 - Equivalent classes:** A list of classes equivalent to CheeseTopping.
 - Superclasses:** A list of superclasses, including 'PizzaTopping'.
 - Inferred anonymous superclasses:** A list of inferred anonymous superclasses.
 - Members:** A list of members of the class.
 - Disjoint classes:** A list of disjoint classes, including 'MeatTopping, SeafoodTopping, VegetableTopping'.

The status bar at the bottom indicates 'Total Commander 7.50a - NOT REGISTERED'.

Narzędzia – NEON Toolkit



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

- Ontology Navigator:** A tree view on the left showing the project structure, including 'Demo 2_1 [Flogic]' and 'RDF_Demo 2.1 [RDF]'.
- Mapping Panel:** The central area shows a mapping between the 'Car' ontology (left) and the 'car' ontology (right). It includes search boxes, a central diagram with nodes and arrows, and a 'Transformations' section with radio buttons for 'mathematical transformation', 'string transformation', and 'other transformation'.
- Attributes/Relations:** Two panels below the mapping show the attributes and relations for the mapped classes. The left panel shows 'cars' with attributes 'cars_name()' and 'prices_car'. The right panel shows 'car' with attributes 'hasName' and 'Properties from product'.
- Instances:** Two tables below the attributes/relations panels show the instances for the mapped classes. The left table has columns 'instance', 'cars_name()', and 'prices_car'. The right table has columns 'instance' and 'hasName'.

Narzędzia – Top Braid Composer

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

The interface includes several panels:

- Classes:** A tree view showing the ontology structure, including classes like foaf:Agent, foaf:Group, foaf:Organization, foaf:Person, foaf:Document, foaf:Image, foaf:OnlineAccount, foaf:Project, geo:SpatialThing, geo:Point, and owl:Nothing.
- Instances:** A table showing instances of the 'person' class and their 'worksAt' property.
- Query Editor:** A SPARQL query editor with the following query:

```
SELECT ?person ?worksAt
WHERE { ?person foaf:workplaceHomepage ?worksAt }
```
- Map:** A map view showing the location of the TopBraid suite, with a red pin on S De Anza Blvd.

[person]	worksAt
dallelang:me	<http://www.topquadrant.com>
rhodgson:me	<http://www.topquadrant.com>