GDAŃSK UNIVERSITY OF TECHNOLOGY

Bazy wiedzy

Krzysztof Goczyła
Aleksander Waloszek
Wojciech Waloszek (wowal@eti.pg.gda.pl)
Teresa Zawadzka
Michał Zawadzki

Katedra Inżynierii Oprogramowania
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

GDAŃSK UNIVERSITY OF TECHNOLOGY

Źródła

- **Podstawy:**
  - Goczyła K.: *Ontologie w systemach informatycznych*, 2011
  - RDF Primer, W3C, 2004,
  - RDF Concepts and Abstract Syntax, W3C, 2004,
  - OWL 2 Web Ontology Language Primer, W3C, 2009

- **Wnioskowanie:**
  - RDF Semantics, W3C, 2004,
  - OWL Direct Semantics, W3C, 2009,
  - ter Horst, Herman J.: *Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary*, Journal of Web Semantics, 3(2-3): 79–115, 2005

GDAŃSK UNIVERSITY OF TECHNOLOGY

Wstęp

GDAŃSK UNIVERSITY OF TECHNOLOGY

Sławna osoba

GDAŃSK UNIVERSITY OF TECHNOLOGY

Semantic Web

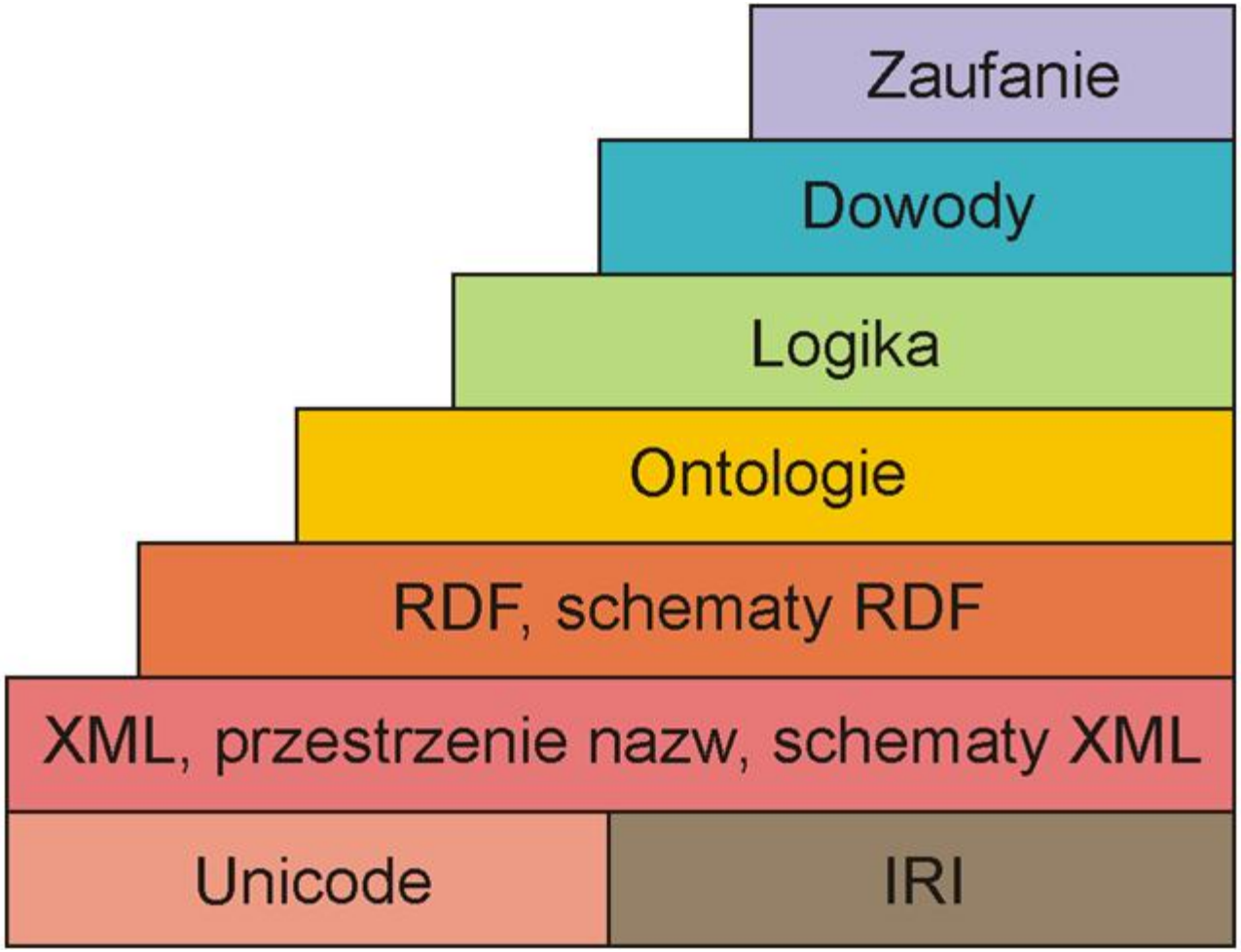
- **Semantic Web**

  Rozszerzenie istniejącej sieci WWW o mechanizmy semantyczne, tak aby informacje dostępne w tej sieci były dobrze zdefiniowane i umożliwiały lepszą współpracę komputerom i ludziom.

  T.Berners-Lee, J.Hendler, O.Lassila. "The Semantic Web", Scientific American, 2001.

GDAŃSK UNIVERSITY OF TECHNOLOGY

„Torcik” Semantic Web



www.semanticweb.org

GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF

GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF jako język reprezentacji wiedzy

1. Resource Description Framework (RDF) to język do zapisu informacji o zasobach Internetu.

2. Trójki RDF to najprostsze jednostki semantyczne, z których budowana jest wiedza w Semantycznym Internecie.

GDAŃSK UNIVERSITY OF TECHNOLOGY

Model wiedzy RDF

Resource Description Framework - model opisu *zasobów*.

Zasób - wszystko, co można zidentyfikować poprzez Internationalized Resource Identifier (IRI) lub poprzez wartość (literał)

Model RDF: zbiór trójek:

	podmiot	**orzeczenie**	**dopełnienie**
inaczej:			
	podmiot	predykat	dopełnienie
inaczej:			
	obiekt	właściwość	wartość
	subject	**predicate**	**object**

GDAŃSK UNIVERSITY OF TECHNOLOGY

Trójki

- Trójki to najprostsze zdania:
  - Jan kocha Joannę,
  - Joanna lubi Jana,
- Według idei RDF *wszystkie* informacje zapisujemy za pomocą takich zdań,
- Według idei RDF *wszystko jest zasobem*.

GDAŃSK UNIVERSITY OF TECHNOLOGY

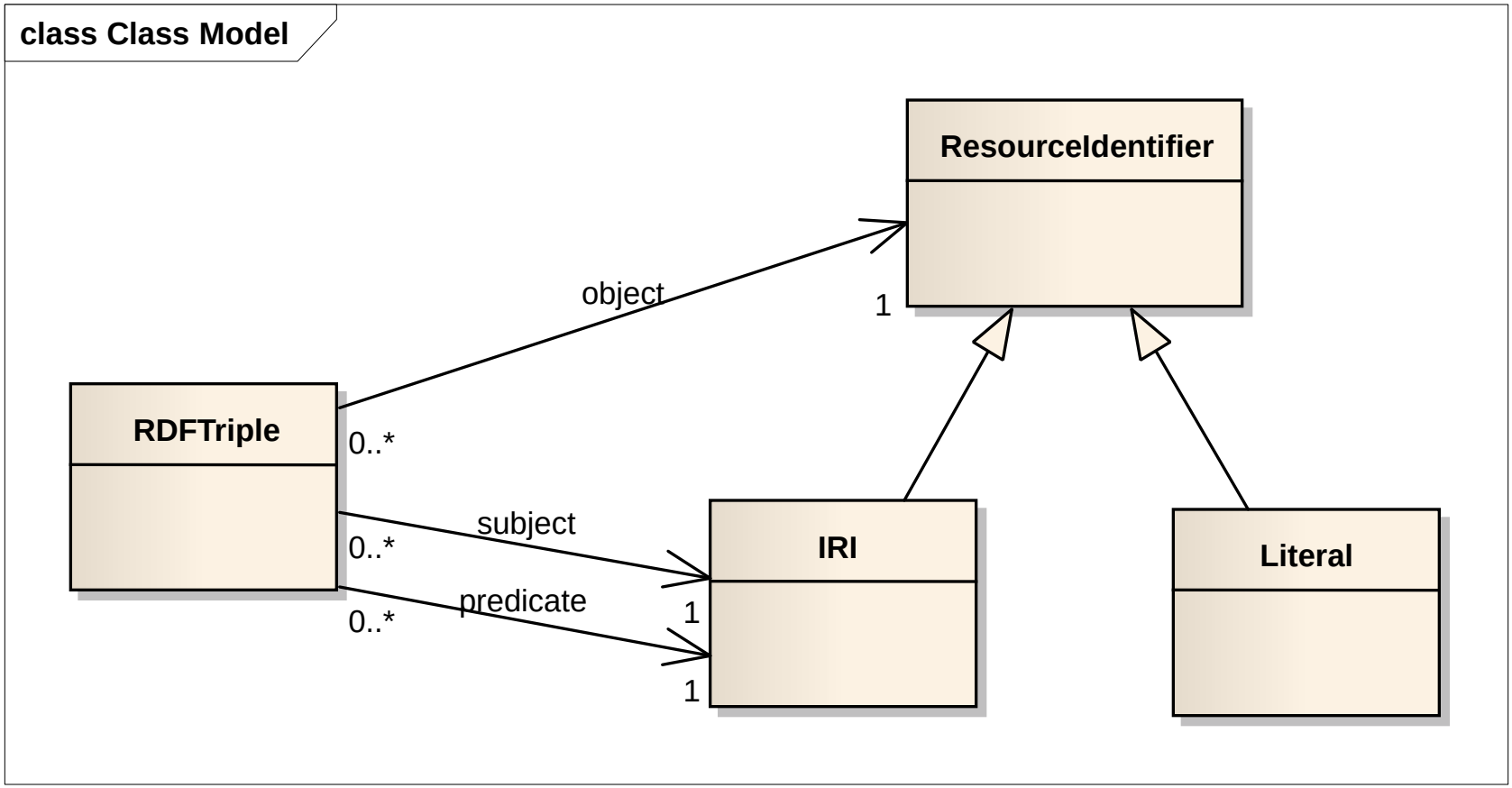
Co to jest zasób?

- RFC 1630 (1994) – Object of the Web
- RFC 2396 (1998) – anything that has identity (ale przykłady dotyczyły jedynie rzeczy materialnych),
- RFC 3986 (2005) – abstract concepts [also] can be resources

Wszystko jest zasobem

GDAŃSK UNIVERSITY OF TECHNOLOGY

Trójki RDF (ENG)



GDAŃSK UNIVERSITY OF TECHNOLOGY

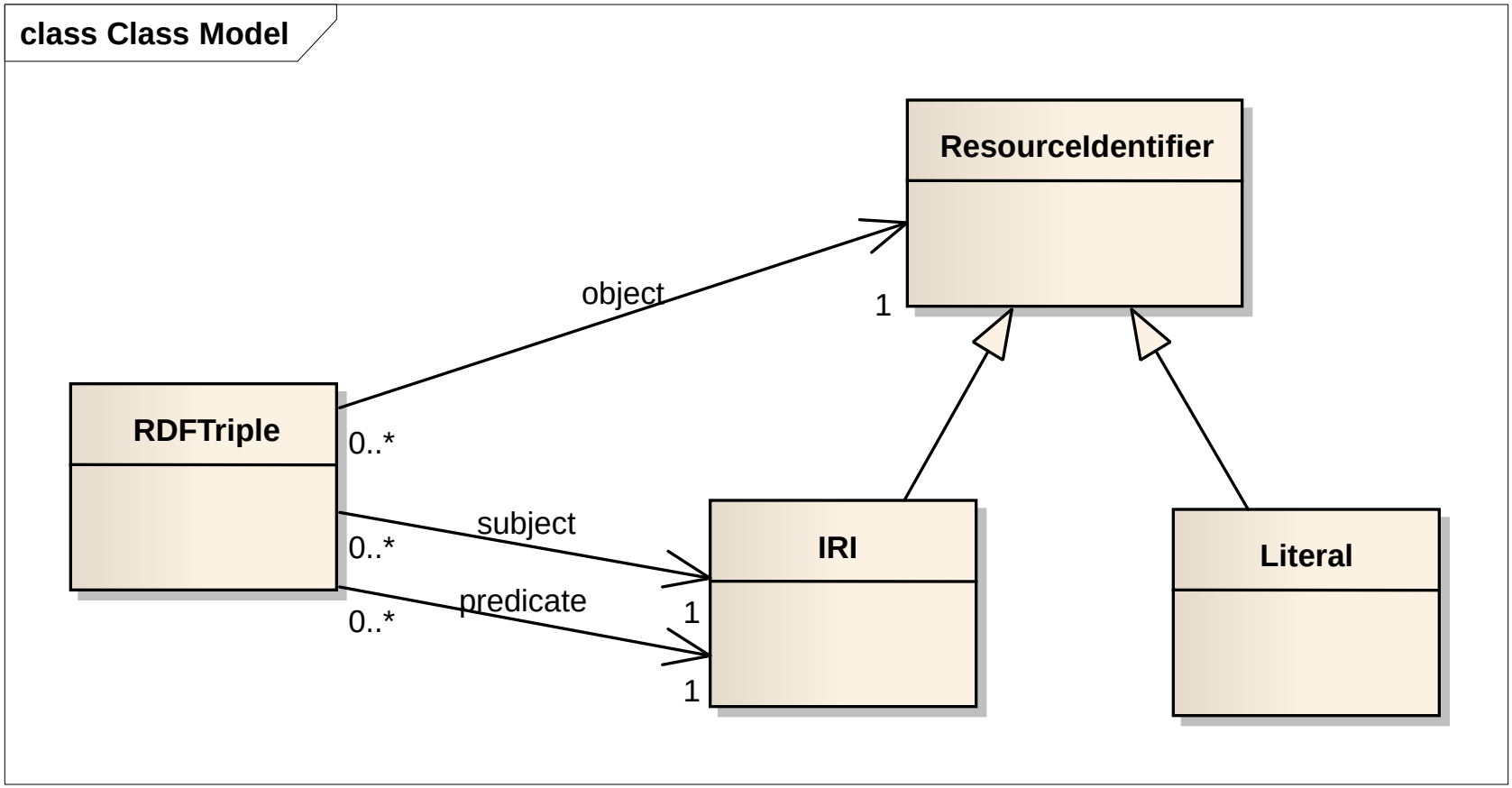
RDF – praktyka

- Jan kocha Joannę .
- Jan kocha Joanna .
- Jan004351 kocha Joanna007832 .
- http://eti.pg.edu.pl/osoby#Jan004351
  http://normalizacjauczuc.gov.pl/uczuciaWyzsze/milosc
  http://eti.pg.edu.pl/osoby#Joanna007832 .

Teraz już wiesz, dlaczego nie należy wyznawać miłości w RDF

GDAŃSK UNIVERSITY OF TECHNOLOGY

Trójki RDF (ENG)



GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF – literały

- Literały reprezentują wartości „konkretne”, tj. wartości z pewnej znanej dziedziny symbolicznej, jak np. łańcuchy znaków, liczby całkowite itp.,
- Literały mogą występować jedynie jako dopełnienia w trójkach,
- Literałom można nadać dowolny typ zdefiniowany w ramach XML Schema (także typy zdefiniowane przez użytkownika zgodnie z XML Schema).

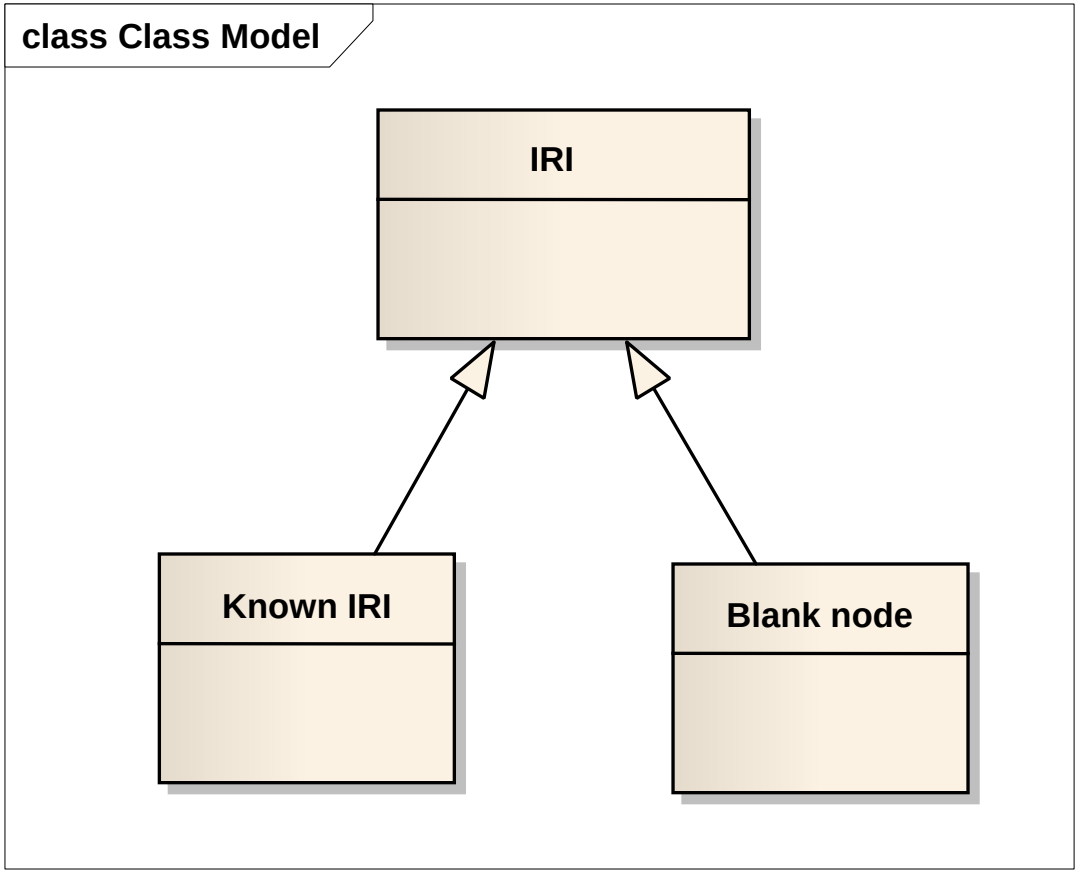
GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF – praktyka – literały

- Jan ma 18 lat .
- Jan maWiek 18 .
- Jan004351 maWiek „18”(liczba całkowita) .
- http://eti.pg.edu.pl/osoby#Jan004351
  http://metrykicywilne.gov.pl/maWiek
  "18"^^xsd:integer .

GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF – i ostatni haczyk



GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF – węzły anonimowe

- Węzły anonimowe reprezentują zasoby (wszystko jest zasobem), których nie chcemy nazywać, ale do których odnosimy się, aby przekazać zamierzoną informację,
- Węzły anonimowe służą również jako „jokery” (węzły reprezentujące „dowolny zasób”) do formułowania zdań, których semantykę można wyrazić przykładowo jako „pracuje w *pewnym projekcie*”, „ma *jakąś stronę WWW*”.

GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF – praktyka – węzły anonimowe

- Joanna kocha (kogoś) kto ma 21 lat.
- Joanna kocha (kogoś) . (Ten ktoś) ma wiek 21 lat.
- Joanna007832 kocha (ktoś1) .
  (ktoś1) maWiek „21”(liczba całkowita) .
- http://eti.pg.edu.pl/osoby#Joanna007832
  http://normalizacjauczuc.gov.pl/uczuciaWyzsze/milosc
  _:ktos1 .
  _:ktos1
  http://metrykicywilne.gov.pl/maWiek
  "21"^^xsd:integer .

GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF – notacje

- Dla RDF zdefiniowano wiele notacji:
  - RDF/XML
  - RDF/JSON
  - N-Triples
  - Turtle

RDF – notacja RDF/XML

```
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:feel="http://normalizacjauczuc.gov.pl/uczuciaWyzsze/"
        xmlns:normy="http://metrykicywilne.gov.pl/">
  …
  <rdf:Description rdf:about="http://eti.pg.edu.pl/osoby#Jan004351">
    <feel:milosc>
      <rdf:Description rdf:about="http://eti.pg.edu.pl/osoby#Joanna007832">
        <feel:lubienie>
          <rdf:Description rdf:about="http://eti.pg.edu.pl/osoby#Jan007832"/>
        </feel:lubienie>
      </rdf:Description>
    </feel:milosc>
  </rdf:Description>
  …
```

RDF – notacja RDF/XML hardcore

```
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:feel="http://normalizacjauczuc.gov.pl/uczuciaWyzsze/"
        xmlns:normy="http://metrykicywilne.gov.pl/">
  …
  <rdf:Description rdf:about="http://eti.pg.edu.pl/osoby#Jan004351">
    <feel:milosc>
      <rdf:Description rdf:about="http://eti.pg.edu.pl/osoby#Joanna007832">
        <feel:lubienie>
          <rdf:Description rdf:about="http://eti.pg.edu.pl/osoby#Jan007832"/>
        </feel:lubienie>
        <rdf:Description>
          <normy:maWiek
            rdf:datatype="http://www.w3.org/2001/XMLSchema#int">19</normy:maWiek>
        </rdf:Description>
      </rdf:Description>
    </feel:milosc>
  </rdf:Description>
  …
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF – notacja RDF/JSON-LD

```
{
    "@context": {
        "kocha": "http://normalizacjauczuc.gov.pl/uczuciaWyzsze/milosc",
        "lubi": "http://normalizacjauczuc.gov.pl/uczuciaWyzsze/lubienie"
    {
    "@id": "http://eti.pg.edu.pl/osoby#Jan004351",
    "kocha": {
        "@id": "http://eti.pg.edu.pl/osoby#Joanna007832",
        "lubi": "http://eti.pg.edu.pl/osoby#Jan004351"
    }
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF – notacja RDF/JSON-LD hardcore

OCENZUROWANE!

GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF – notacja RDF/JSON-LD hardcore

```
{
    "@context": {
        "kocha": "http://normalizacjauczuc.gov.pl/uczuciaWyzsze/milosc",
        "lubi": "http://normalizacjauczuc.gov.pl/uczuciaWyzsze/lubienie",
        "maWiek": "http://metrykicywilne.gov.pl/maWiek"
    {
    "@id": "http://eti.pg.edu.pl/osoby#Jan004351",
    "kocha": {
        "@id": "http://eti.pg.edu.pl/osoby#Joanna007832",
        "lubi": "http://eti.pg.edu.pl/osoby#Jan004351",
        "kocha": {
            "maWiek": {
                "@type": "http://www.w3.org/2001/XMLSchema#int",
                "@value": "21"
            }
        }
    }
}
```

RDF – notacja N-Triple

- Wygląda akurat tak, jak to, co pisaliśmy do tej pory:

http://eti.pg.edu.pl/osoby#Jan004351
http://normalizacjauczuc.gov.pl/uczuciaWyzsze/milosc
http://eti.pg.edu.pl/osoby#Joanna007832 .

http://eti.pg.edu.pl/osoby#Joanna007832
http://normalizacjauczuc.gov.pl/uczuciaWyzsze/lubienie
http://eti.pg.edu.pl/osoby#Jan004351 .

http://eti.pg.edu.pl/osoby#Joanna007832
http://normalizacjauczuc.gov.pl/uczuciaWyzsze/milosc
_:ktos1 .

_:ktos1
http://metrykicywilne.gov.pl/maWiek
"21"^^xsd:integer .

RDF – notacja Turtle

- To, co pisaliśmy do tej pory, ale z ulepszeniami(-aczami):

```
@base <http://eti.pg.edu.pl/osoby#> .
@prefix feel <http://normalizacjauczuc.gov.pl/uczuciaWyzsze/> .

Jan004351 feel:milosc Joanna007832 .
Joanna007832 feel:lubienie Jan004351 .
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF – notacja Turtle hardcore

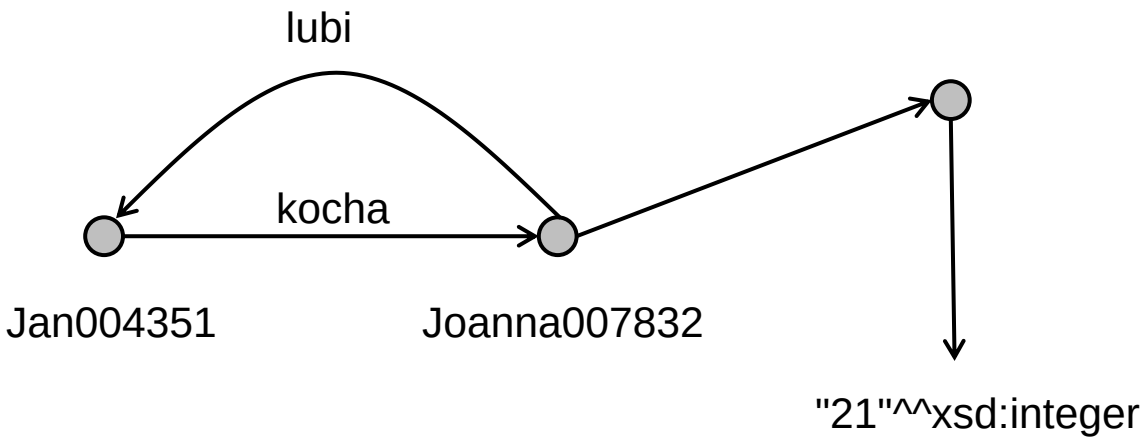
- To, co pisaliśmy do tej pory, ale z ulepszeniami(-aczami):

```
@base <http://eti.pg.edu.pl/osoby#> .
@prefix feel <http://normalizacjauczuc.gov.pl/uczuciaWyzsze/> .
@prefix normy <http://metrykicywilne.gov.pl/> .
@prefix xsd <http://www.w3.org/2001/XMLSchema#> .

Jan004351 feel:milosc Joanna007832 .
Joanna007832 feel:lubienie Jan004351 ;
                    feel:milosc [ normy:maWiek "21"^^xsd:integer ] .
```

RDF – notacja (dosłownie) graficzna

- Formułując kolejne trójki, budujemy **graf RDF** (skierowany):
  - „rzeczowniki” (podmioty i dopełnienia) stanowią węzły,
  - każda trójka to krawędź skierowana od podmiotu do dopełnienia z etykietą określoną przez orzeczenie,
  - notacja graficzna jest jednak nieformalna.

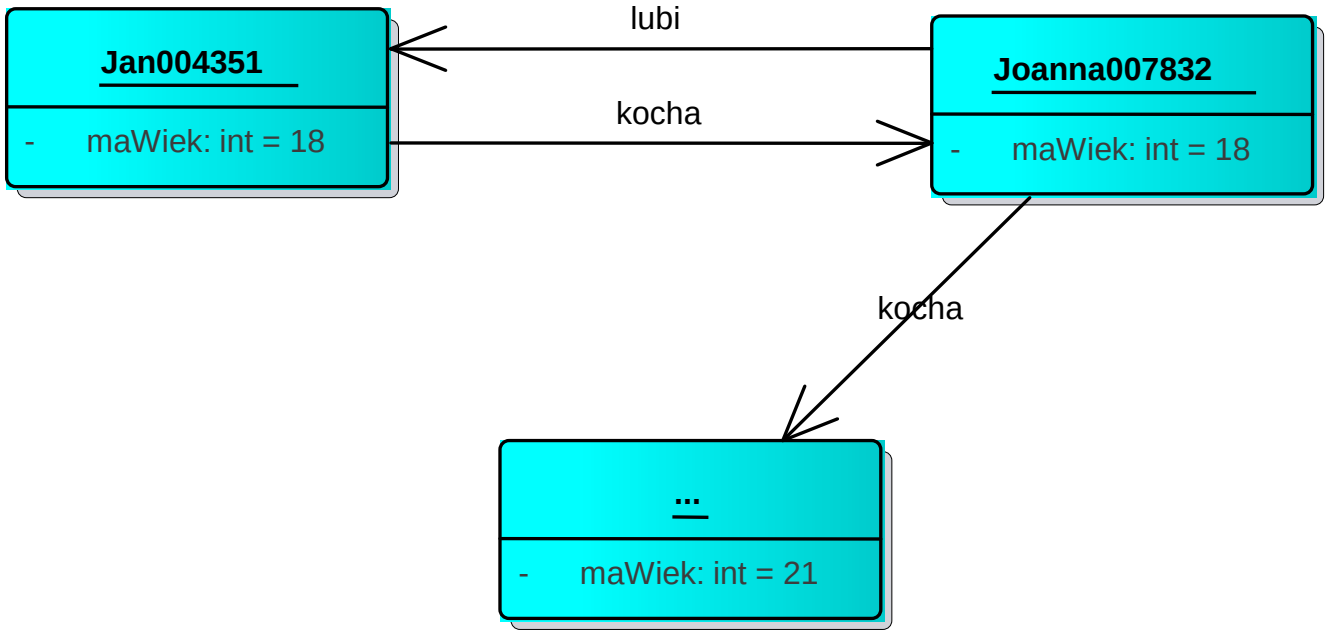


GDAŃSK UNIVERSITY OF TECHNOLOGY

RDFS

Co teraz?

- Dysponując informacjami zdobytymi do tej pory, jesteśmy w stanie swobodnie zapisać wszystkie informacje z, powiedzmy, diagramów obiektów:
  - związki między obiektami – wystąpienia asocjacji,
  - wartości atrybutów obiektów



GDAŃSK UNIVERSITY OF TECHNOLOGY

Co teraz??

- Diagramy mogą w zasadzie być dowolnie rozbudowane, ale… czegoś tu brakuje
- Dzięki RDFS możemy rozszerzyć naszą siłę ekspresji trójek RDF i zaatakować nowy element modelu: diagramy klas
- RDFS, czyli RDF Schema, robi to tak, żeby nie naruszyć podstawowych zasad RDF:
  - nadal wszystko jest zasobem,
  - nadal wszystko zapisujemy za pomocą trójek

GDAŃSK UNIVERSITY OF TECHNOLOGY

Co w takim razie możemy – typy

- Po pierwsze, możemy chcieć powiedzieć, że obiekt jest jakiegoś typu,
- W RDFS służy do tego **specjalne orzeczenie**,
- To orzeczenie to **rdf:type**

```
@base <http://eti.pg.edu.pl/osoby#> .
@prefix rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix feel <http://normalizacjauczuc.gov.pl/uczuciaWyzsze/> .

Jan004351 feel:milosc Joanna007832 .
Joanna007832 feel:lubienie Jan004351 .

Jan004351 rdf:type Student .
Joanna007832 rdf:type Studentka .
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

CwTRM – dziedziny i zakresy

- Możemy też chcieć powiedzieć, że pewne orzeczenie może łączyć tylko obiekty określonych typów: tak jak na diagramie klas łączymy asocjacją konkretne klasy,
- W RDFS służą do tego specjalne orzeczenia,
- Te orzeczenia to **rdfs:domain** i **rdfs:range**

```
@base <http://eti.pg.edu.pl/osoby#> .
@prefix rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs <http://www.w3.org/2000/01/rdf-schema#> .
@prefix feel <http://normalizacjauczuc.gov.pl/uczuciaWyzsze/> .

feel:milosc rdfs:domain Osoba .
feel:milosc rdfs:range Osoba .
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

CwTRM – klasy i dziedziczenie

- Atakujemy dalej diagram klas, chcąc powiedzieć, że dwie klasy dziedziczą od siebie,
- W RDFS służy do tego specjalne orzeczenie,
- To orzeczenie to **rdfs:subClassOf**

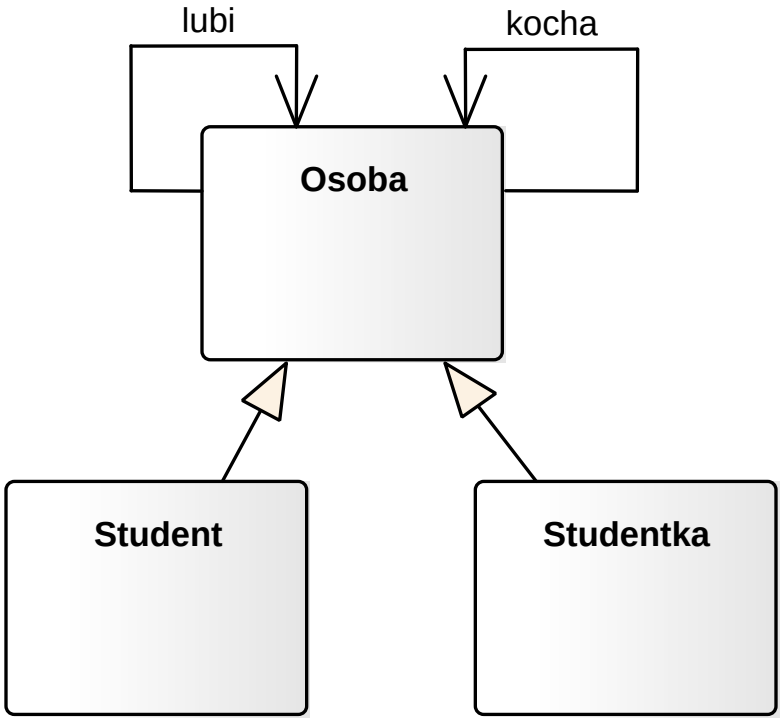
```
@base <http://eti.pg.edu.pl/osoby#> .
@prefix rdfs <http://www.w3.org/2000/01/rdf-schema#> .

Student rdfs:subClassOf Osoba .
Studentka rdfs:subClassOf Osoba .
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

CwTRM!

- Wprowadzone do tej pory orzeczenia pozwalają nam właściwie na pokrycie modelem RDF również diagramu klas (pokrywamy dziedziczenie klas i asocjacje – bez liczebności)



GDAŃSK UNIVERSITY OF TECHNOLOGY

CwTRM?

- To w zasadzie wystarcza do bardzo wielu zastosowań, ale RDFS robi w tym momencie coś trochę szalonego,

- Według zasad, że wszystko jest zasobem, a o zasobach mówimy, stosując trójki, RDFS pozwala nam formułowac zdania o klasach, dokładnie takie same, jak moglibyśmy sformułować o „zwykłych” obiektach

GDAŃSK UNIVERSITY OF TECHNOLOGY

CwTRM(!) – typy dla klas

- Jakiej więc klasy są klasy?
- W RDFS mamy do tego **specjalną klasę** (wyróżniony zasób),
- Ta klasa to **rdfs:Class**

@base <http://eti.pg.edu.pl/osoby#> .
@prefix rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs <http://www.w3.org/2000/01/rdf-schema#> .

Student rdf:type rdfs:Class .
Studentka rdf:type rdfs:Class .
Osoba rdf:type rdfs:Class .

(!) Jakiej klasy jest klasa rdfs:Class?

CwTRM(!) – typy dla orzeczeń

- Jakiej więc klasy są orzeczenia?
- W RDFS mamy do tego specjalną klasę,
- Ta klasa to **rdf:Property**

```
@base <http://eti.pg.edu.pl/osoby#> .
@prefix rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs <http://www.w3.org/2000/01/rdf-schema#> .
@prefix feel <http://normalizacjauczuc.gov.pl/uczuciaWyzsze/> .

feel:milosc rdf:type rdf:Property .
feel:lubienie rdf:type rdf:Property .
```

(!) Jakiej klasy są zasoby rdf:type, rdfs:subClassOf, rdfs:domain?
(!) Jaką dziedzinę ma rdfs:domain? A jaki zakres?

CwTRM(!) – typy dla wartości

- Jakiej więc klasy są wartości atrybutów?
- W RDFS mamy do tego specjalną klasę,
- Ta klasa to **rdfs:Literal**

```
@base <http://eti.pg.edu.pl/osoby#> .
@prefix rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs <http://www.w3.org/2000/01/rdf-schema#> .
@prefix normy <http://metrykicywilne.gov.pl/> .

normy:maWiek rdfs:domain Osoba .
normy:maWiek rdfs:range rdfs:Literal .
```

(!) Jakiej klasy jest rdfs:Literal?
(!) Gdybyśmy rozważyli klasę xsd:integer, od czego by dziedziczyła? I jakiego byłaby typu?

CwTRM(!) – typy dla zasobów

- Wreszcie: w RDF wszystko jest zasobem! Jakiej więc klasy są zasoby? (W RDFS mamy do tego specjalną klasę!),
- Ta klasa to **rdfs:Resource**

@base <http://eti.pg.edu.pl/osoby#> .

…

@prefix feel <http://normalizacjauczuc.gov.pl/uczuciaWyzsze/> .

Osoba rdf:type rdfs:Resource .

Student rdfs:subClassOf rdfs:Resource .

feel:milosc rdf:type rdfs:Resource .

(!) Jakiej klasy jest rdfs:Resource?

(!) Od jakiej klasy dziedziczą rdfs:Class, rdf:Property i rdfs:Literal?

GDAŃSK UNIVERSITY OF TECHNOLOGY

CwTRM(!)

- Wykorzystując zaawansowane możliwości RDFS możemy więc opowiadać m.in. o:
  - klasach, np. tworząc dla nich (meta)klasy:

    KlasyUtworzonePrzezMichala rdfs:subClassOf rdfs:Class .
  - orzeczeniach, np. tworząc dla nich (meta)klasy:

    OrzeczeniaUtworzonePrzezMichala rdfs:subClassOf rdf:Property .
  - wartościach konkretnych, np. tworząc nowe typy:

    TypMichala rdfs:subClassOf rdf:Literal .

    TypMichala rdf:type TypyUtworzonePrzezMichala .

- Dzięki temu możemy nawigować po nawet takich grafach danych, których schematu nie znamy, bądź nie znamy do końca

GDAŃSK UNIVERSITY OF TECHNOLOGY

SPARQL

GDAŃSK UNIVERSITY OF TECHNOLOGY

SPARQL – język zapytań dla trójek

- Językiem zapytań dla trójek RDF jest zdefiniowany w ramach rekomendacji W3C SPARQL (http://www.w3.org/TR/rdf-sparql-query/),
- SPARQL (Simple Protocol and Query Language) wykorzystuje „wzorce grafowe” składające się z trójek RDF, w których niektóre elementy zostały zastąpione zmienną,
- Wynikiem zapytania SPARQL są podstawienia zmiennych odpowiadające odnalezionym elementom grafu RDF we fragmentach pasujących do wzorca.

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL

Kto kocha Joannę?

```
PREFIX : http://eti.pg.edu.pl/osoby#
PREFIX feel: http://normalizacjauczuc.gov.pl/uczuciaWyzsze/

SELECT ?kto
WHERE  {
        ?kto feel:milosc Joanna007832 .
}
```

?kto
Jan004351

Dalej będzie bez prefixów!!!

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zwykłe

Kogo kocha Joanna?

```
SELECT ?kogo
WHERE  {
        Joanna007832 feel:milosc ?kogo .
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zwykłe..

Kogo lubią jednocześnie Joanna i Jan?

```
SELECT ?kogo
WHERE  {
        Joanna007832 feel:lubienie ?kogo .
        Jan004351 feel:lubienie ?kogo .
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zwykłe…

Kogo i w jakim wieku lubią jednocześnie Joanna i Jan?

```
SELECT ?kogo ?wiek
WHERE  {
        Joanna007832 feel:lubienie ?kogo .
        Jan004351 feel:lubienie ?kogo .
        ?kogo normy:maWiek ?wiek .
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zwykłe….

Kogo lubi lub kocha Joanna?

```
SELECT ?kogo
WHERE  {
        { Joanna007832 feel:lubienie ?kogo . }
        UNION
        { Joanna007832 feel:milosc ?kogo . }
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zwykłe…..

Kto lubi lub kocha Joannę?

```
SELECT ?kto
WHERE  {
        { ?kto feel:lubienie Joanna007832 . }
        UNION
        { ?kto feel:milosc Joanna007832 . }
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zwykłe……

Kogo pełnoletniego lubi Joanna?

```
SELECT ?kogo
WHERE  {
        Joanna007832 feel:lubienie ?kogo .
        ?kogo normy:maWiek ?wiek .
        FILTER ( ?wiek >= 18 )
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zwykłe…….

Kogo pełnoletniego lubi Joanna lub Jan?

```
SELECT ?kogo
WHERE  {
        { Joanna007832 feel:lubienie ?kogo . }
        UNION
        { Jan004351 feel:lubienie ?kogo . }
        ?kogo normy:maWiek ?wiek .
        FILTER ( ?wiek >= 18 )
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zaawansowane (1.1)

- Wyrażenia „regularne” z wykorzystaniem krawędzi:

  ^ to „odwrotna” krawędź,

  | to alternatywa krawędzi,

  / to ciąg krawędzi,

  + to dowolna liczba powtórzeń krawędzi,

  * to dowolna (w tym zerowa) liczba powtórzeń krawędzi.

- EXISTS i NOT EXISTS
- Funkcje agregujące (np. COUNT) i grupowanie

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zaawansowane

Kogo lubi Joanna a nie lubi Jan (NOT EXISTS)?

```
SELECT ?kogo
WHERE  {
        Joanna007832 feel:lubienie ?kogo .
        FILTER NOT EXISTS { Jan004351 feel:lubienie ?kogo . }
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zaawansowane..

Kogo lubi kogoś, kto lubi kogoś, kto lubi (…) Joannę?

```
SELECT ?kto
WHERE  {
        ?kto feel:lubienie/feel:lubienie+ Joanna007832 .
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zaawansowane…

Ile osób lubi Joannę (COUNT)?

```
SELECT (COUNT(?kto) AS ?ile)
WHERE  {
        ?kto feel:lubienie Joanna007832 .
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zaawansowane….

Ile osób lubi Joannę, pogrupuj po wieku?

```
SELECT ?wiek (COUNT(?kto) AS ?ile)
WHERE  {
        ?kto feel:lubienie Joanna007832 .
        ?kto normy:maWiek ?wiek .
}
GROUP BY ?wiek
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zakręcone

- Jednak prawdziwa zabawa rozpoczyna się, gdy zaczynamy wykorzystywać schematy

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zakręcone

Jacy studenci lubią Joannę?

```
SELECT ?kto
WHERE  {
        ?kto feel:lubienie Joanna007832 .
        ?kto rdf:type Student .
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zakręcone..

Wśród jakich rodzajów osób Jan ma osoby (znane i) lubiane?

```
SELECT ?rodzaj
WHERE  {
        Jan004351 feel:lubienie ?kogo .
        ?kogo rdf:type ?rodzaj .
        ?rodzaj rdfs:subClassOf+ Osoba .
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

Zapytanie SPARQL – zakręcone...

Jakie atrybuty przynależne dowolnej osobie ma Jan?

```
SELECT ?atrybut ?wartosc
WHERE  {
        Jan004351 ?atrybut ?wartosc .
        ?atrybut rdfs:range rdfs:Literal .
        ?atrybut rdfs:domain ?rodzaj .
        ?rodzaj rdfs:subClassOf+ Osoba .
}
```

GDAŃSK UNIVERSITY OF TECHNOLOGY

SPARQL endpoint(s)

- SPARQL stanowi ważny element koncepcji Semantycznego Internetu,
- Oprócz bycia językiem zapytań pozwalającym na badania schematu jest też językiem dostępowym do repozytoriów danych opublikowanych w ramach Semantycznego Internetu/inicjatywy Open Linked Data
- Repozytoria takie są publikowane w formie tzw. SPARQL endpointów (punktów dostępowych), do których możne kierować zapytania SPARQL

GDAŃSK UNIVERSITY OF TECHNOLOGY

Przechowywanie danych bezpośrednio w RDF

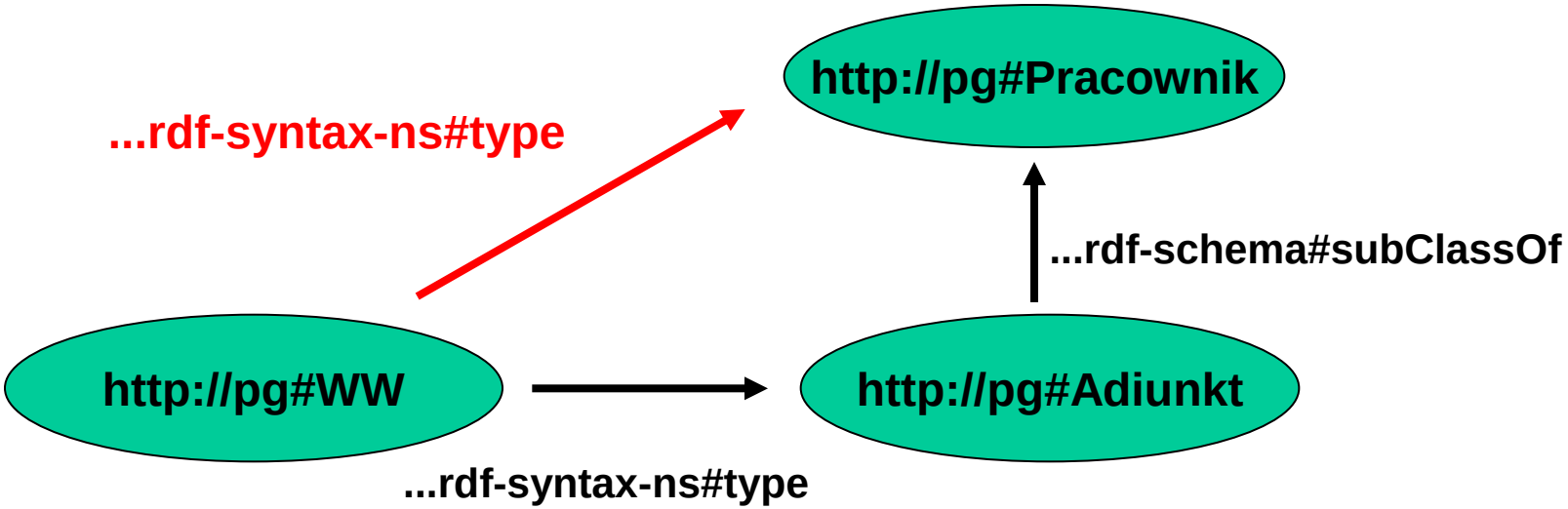
- RDF stanowi też dobrą alternatywą dla baz grafowych jako podstawowy model przechowywania danych,
- Narzędzia do przechowywania trójek RDF oferuje m.in. Oracle (w ramach tzw. Oracle Semantics),
- Dane grafowe sprawdzają się szczególnie tam, gdzie wymagane są zapytania rekurencyjne oraz gdzie występuje zbyt duża liczba złączeń dla efektywnego wykonywania zapytań SQL

GDAŃSK UNIVERSITY OF TECHNOLOGY

Wnioskowanie w Semantycznym Internecie

Wnioskowanie RDFS

- Wnioskowanie w RDFS polega na odkrywaniu nowych prawdziwych trójek na podstawie tych, które już znamy.



...
<#WW> rdf:type <#Pracownik> .
...

(Meta) reguły wnioskowania w bazie RDFS

- Przykład reguły w bazie RDFS:

  Jeżeli C jest podklasą klasy D i a jest obiektem typu C,
  to a jest obiektem typu D

rdfs9 **C rdfs:subClassOf D**
a rdf:type C } **=> a rdf:type D**

- Dodatkowe reguły pozwalają na określenie dziedziny i zakresu predykatów oraz na podobną jak powyżej obsługę hierarchii predykatów,

- Specyfikacja RDFS definiuje kilkanaście reguł wnioskowania.

GDAŃSK UNIVERSITY OF TECHNOLOGY

Przykłady reguł RDFS

rdfs2	**p rdfs:domain C** **a p b**	}	**=> a rdf:type C**
rdfs3	**p rdfs:range C** **a p b**	}	**=> b rdf:type C**
rdfs5	**r rdfs:subPropertyOf s** **p rdfs:subPropertyOf r**	}	**=> p rdfs:subPropertyOf s**
rdfs7	**p rdfs:subPropertyOf r** **a p b**	}	**=> a r b**
rdfs11	**D rdfs:subClassOf E** **C rdfs:subClassOf D**	}	**=> C rdfs:subClassOf E**

GDAŃSK UNIVERSITY OF TECHNOLOGY

Przykłady reguł RDFS (2)

rdf1	**a p b**	}	**=> p rdf:type rdfs:Property**
rdfs4a	**a p b**	}	**=> a rdf:type rdfs:Resource**
rdfs4b	**a p b**	}	**=> b rdf:type rdfs:Resource**

GDAŃSK UNIVERSITY OF TECHNOLOGY

Interpretacja Herbranda

- Efektem wnioskowania w RDF(S) musi być wyprodukowanie nowego zdania,
- Jako wyobrażenie modelu świata w RDF najbardziej naturalnie jest przyjąć graf, ale uzupełniony o wszystkie nowo wyprodukowane trójki,
- Odpowiada to pojęciu *interpretacji Herbranda:* modelem świata są zdania, które możemy wywieść ze zdań nam znanych (model = opis),
- Takie podejście pozwala na wnioskowanie pełne i poprawne nawet przy braku separacji schematu („terminologii”) i danych („opisu świata”),
- **Jednak** ograniczenie tkwi w samym procesie wnioskowania i w definicji modelu.

GDAŃSK UNIVERSITY OF TECHNOLOGY

RDF a OWL

- Pomimo tego, że OWL wykorzystuje słownictwo RDF(S), między obydwoma językami zachodzi znaczna rozbieżność założeń,
- OWL może wyrażać ontologie logiki opisowej:
  - ścisła separacja terminologii od opisu świata,
  - postrzeganie budowy świata poprzez pryzmat modeli teoriozbiorowych (interpretacje Tarskiego),
  - wnioskowanie na podstawie analizy przestrzeni modeli,
- RDF(S) oparty jest na idei przetwarzania grafów:
  - wszystko jest zasobem, „wierzchołkiem”, klasy i wystąpienia są równorzędnymi elementami grafu,
  - świat opisywany jest grafami-zbiorami zdań (interpretacje Herbranda),
  - wnioskowanie poprzez przekształcanie i rozbudowywanie grafów.

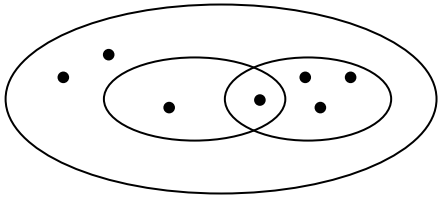
GDAŃSK UNIVERSITY OF TECHNOLOGY

OWL a RDF

OWL

Syntaktyka: zdania RDF

Model: interpretacje Tarskiego



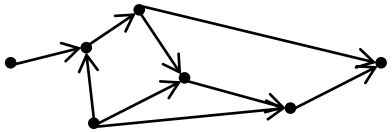
Wnioskowanie: analiza przestrzeni modeli

Pytania: problemy wnioskowania DL

RDF

Syntaktyka: zdania RDF

Model: interpretacje Herbranda



Wnioskowanie: dobudowywanie nowych zdań za pomocą reguł

Pytania: szukanie wzorców grafowych (SPARQL)

GDAŃSK UNIVERSITY OF TECHNOLOGY

Profile OWL 2 – złożoność

Język i semantyka	Złożoność
OWL 2, semantyka RDF	Nierozstrzygalne
OWL 2, semantyka bezpośrednia	(2)NEXPTIME-complete
OWL 2 EL	PTIME-complete
OWL 2 QL	NLogSpace-complete
OWL 2 RL	PTIME-complete
RDF	PTIME-complete