RDF Schema

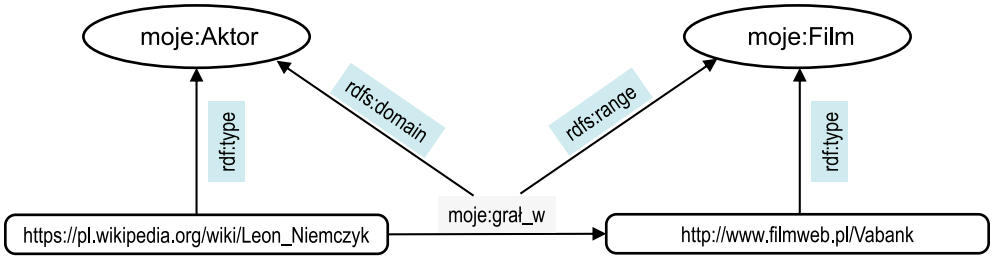
Ponieważ RDF daje zupełną dowolność w wyborze słowników, niewątpliwie spełniony jest postulat (2). Należy jednak pamiętać, że źródła informacji posługujące się niepowiązanymi słownikami uniemożliwią skuteczne skojarzenie oznaczonych, co prawda, informacji. Dlatego znaczenia nabierają również postulaty (3) – aby umożliwić definiowanie związków pomiędzy klasami pojęć – oraz (4) – aby zminimalizować liczbę dublujących się ontologii. Standard o nazwie RDF Schema (RDFS) dostarcza podstawowych pojęć umożliwiających określanie zależności pomiędzy klasami obiektów oraz typami własności tych obiektów. W szczególności, wprowadza pojęcie klasy (`rdfs:Class`) wraz z szeregiem własności, umożliwiając definiowanie hierarchii klas i hierarchii własności. RDFS pozwala również na określenie, do jakich typów podmiotu (`rdfs:domain`) i dopełnienia (`rdfs:range`) wolno zastosować konkretne orzeczenie. Stanowi zatem kompletną technologię określania zawartości poszczególnych klas i zależności pomiędzy nimi – czyli budowy lub opisania dowolnej ontologii. Jeśli utworzymy własną, prostą ontologię, wymuszającą przynależność podmiotu i dopełnienia do określonych typów[6]:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix moje: <http://mojepojęcia/>
moje:grał_w rdfs:domain moje:Aktor
moje:grał_w rdfs:range moje:Film
```

to poprzednie zdanie RDF musimy uzupełnić o deklaracje typu dla podmiotu i orzeczenia:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
<https://pl.wikipedia.org/wiki/Leon_Niemczyk> rdf:type moje:Aktor
<http://www.filmweb.pl/Vabank> rdf:type moje:Film
```

Pierwotne zdanie, wraz ze zdefiniowanymi ograniczeniami, można zaprezentować w postaci grafu przedstawionego na rys. 5.4. Obiekty umieszczono w zaokrąglonych prostokątach, a własne pojęcia – w elipsach.



Rysunek 5.4. Przykładowe zdanie wraz z prostą ontologią ograniczającą zakres stosowania orzeczenia do obiektów określonego typu. Pojęcia dostarczane przez RDF i RDFS umieszczono na niebieskim tle

[6] Polecenie @prefix wprowadza makrodefinicje zastępujące powtarzające się części URL. W przykładzie posługujemy się pojęciami zdefiniowanymi w RDFS i wprowadzamy dwie nowe, własne klasy: aktora i filmu.

Już samo wspomnienie rozmaitości przedstawionych tutaj konkurencyjnych technologii oznaczania każe wątpić w spełnienie postulatu (4). Nie należy jednak dramatyzować z powodu promowania przez potentatów własnych rozwiązań, tym bardziej jeśli, jak w przypadku mikrodanych, jest to wynikiem współpracy konkurentów. Niezależnie istnieje bowiem ekosystem ontologii opracowanych przez urzędy i organizacje non-profit, wraz z rozmaitością udostępnionych przez nie danych zorganizowanych według tych ontologii. Cała inicjatywa nosi nazwę Linked Open Data (LOD). Aktualne powiązania między ontologiami, tj. wykorzystywanie już zdefiniowanych słowników pojęć przez inne, nowsze słowniki, są przedstawione w postaci grafu pod adresem `http://lod-cloud.net`. Obecnie DBPedia oraz GeoNames definiują bardzo wiele pojęć i w sposób naturalny są również wykorzystywane przez masę innych ontologii, które stopniowo pokrywają kolejne dziedziny życia. W tym ekosystemie, jak w każdym innym, trwa również walka o dominację – przy czym najskuteczniejszym „materiałem promocyjnym" dla ontologii jest upublicznienie odpowiednio licznych i atrakcyjnych danych nią się posługujących.

SPARQL

Semantyczne oznaczenie danych z użyciem ontologii jest zaledwie punktem wyjścia do ich dalszej analizy. Dane, zwłaszcza te dostępne z wielu heterogenicznych i rozproszonych źródeł, muszą być w tym celu efektywnie przechowywane. Używa się do tego specjalizowanych *baz trójek danych* (*triple store*)[7]. Baza trójek danych jest bardzo pojemnym narzędziem, podobnie jak sam RDF. Można więc w niej przechowywać zarówno fakty, jak i wykorzystywane ontologie. System informatyczny należycie przechowujący takie dane, wraz z mechanizmem wykonywania zapytań, może być już uznany [51] za *graf wiedzy* (*knowledge graph*) – narzędzie do sprawnej eksploracji wiedzy z danych w postaci sieci semantycznej.

Głównym celem tworzenia baz typu *triple store* jest umożliwienie bardzo elastycznego wyszukiwania informacji. Służy do tego technologia SPARQL, która oznacza zarówno standard składni języka zapytań, jak i standard protokołu komunikacyjnego. Podstawowym typem zapytania, analogicznie do SQL, jest `SELECT`, które zwraca listę wyników z możliwością jej uporządkowania (`ORDER BY`) i grupowania (`GROUP BY`). Jednak analogie kończą się, gdy w grę wchodzi filtrowanie danych klauzulą `WHERE`. Jako że zdania przechowywane są we wspólnym obszarze, mechanizm filtrowania umożliwia precyzyjne specyfikowanie relacji pomiędzy wydobywanymi danymi. Zapytanie to nie ma klauzuli `JOIN` – wszak wszystkie dane w bazie są już połączone w jednej tabeli!

Wykorzystajmy SPARQL do znalezienia odpowiedzi na początkowe pytanie: „którzy aktorzy filmowi wspólnie grali najczęściej?". Istnieje wiele, współczesnych, otwartych baz danych, które albo można pobrać w postaci RDF (np. Biblioteka Kongresu USA), albo odpytać online, wykorzystując udostępnioną usługę SPARQL (Europejski Portal Danych, DBpedia, LinkedGeoData. GeoNames.org). Skorzystamy z możliwości

[7] Spośród wielu dostępnych obecnie baz trójek danych warto wskazać dwa popularne rozwiązania o otwartym kodzie źródłowym: OpenLink Virtuoso oraz Apache Jena.