

OWL – język definiowania ontologii w semantycznej sieci WWW¹

Jolanta Cybulka

Politechnika Poznańska, Instytut Automatyki i Inżynierii Informatycznej

Pl. M. Skłodowskiej-Curie 5, 60-965 Poznań

e-mail: Jolanta.Cybulka@put.poznan.pl

Streszczenie. W pracy scharakteryzowano język OWL (skr. ang. *Web Ontology Language*) przeznaczony do definiowania ontologii rozumianych jako inżynierskie artefakty, wykorzystywane do oznaczania zawartości dokumentów w sieci WWW. Język OWL, będąc następcą kilku wcześniejszych języków, m.in. RDF, DAML, OIL i DAML+OIL, został zaproponowany przez konsorcjum W3C jako standard w rozważanej klasie.

Słowa kluczowe: EuroWordNet, metadane, ontologia, OWL, semantyczna sieć WWW, SIMPLE

1. Wprowadzenie

Efektom burzliwego rozwoju sieci WWW są ogromne repozytoria informacyjne, których sprawne automatyczne przetwarzanie wymaga wiedzy o znaczeniu zawartości zgromadzonych w repozytoriach dokumentów. Powyższa idea legła u podstaw tzw. „semantycznej sieci WWW” [3] i, jak dotychczas, zaowocowała powstaniem kilku uniwersalnych języków służących do definiowania tzw. „ontologii” (semantycznych metadanych związanych z dokumentami).

Termin ontologia² został zapożyczony z filozofii i znalazł zastosowanie m.in. w lingwistyce (leksykony semantyczne), sztucznej inteligencji (specyfikacje konceptualizacji), a także inżynierii systemów informatycznych (ontologie znaczenia opisów rzeczywistości, [10]). Za pośrednictwem sztucznej inteligencji, ontologie wkroczyły w obszar zainteresowań semantycznej sieci WWW. Są one tu rozumiane jako inżynierskie artefakty służące do specyfikowania konceptów (pojęć, a także wiążących je relacji) wybranej dziedziny zastosowań, tak, by agenty programowe lub ludzie mogli się w rozważanej dziedzinie porozumiewać. Najprostszym rodzajem ontologii jest taksonomia, oparta na relacji hierarchizującej pojęcia, która bywa nazywana relacją typu *is-a*, relacją *hiponimii* (wraz z odwrotną relacją *hiperonimii*) lub relacją *specjalizacji* (z odwrotną relacją *generalizacji*), w zależności od rozważanego obszaru zastosowań. Stosowanie tego rodzaju relacji jest dobrze ugruntowane zarówno teoretycznie, jak i praktycznie. Bardziej zaawansowane kategorie ontologii mogą zawierać również inne rodzaje powiązań. Nie istnieją ogólnie przyjęte zbiory relacji ontologicznych, i dlatego proponuje się różne rozwiązania, z których jedynie część bazuje na dobrze ufundowanych podstawach, nawiązujących np. do ontologii formalnych (mereologiczna

¹ Pracę wykonano w ramach tematu DS-45-083/2004 finansowanego przez Politechnikę Poznańską.

² Pojęcie „ontologia”, które wprowadził w 1613 roku R. Goclenius (w *Lexicon Philosophicum...*), używane jest powszechnie do określenia działu filozofii, jakim jest teoria bytów realnie istniejących [14].

teoria relacji typu całość-część), czy też do arystotelesowskich rozważań na temat sposobów „bytowania materii” (jakość, ilość, relacja, czas, miejsce, położenie, działanie, doznawanie i posiadanie) i wynikającej z nich „jakościowej struktury znaczenia” (tzw. „struktury qualii”), [11, 12, 14]. Jakościowa struktura znaczenia pozwala wyodrębnić pojęcia i relacje czterech kategorii: związanej z pochodzeniem bytu (rola agentywna), z jego wewnętrzną strukturą (rola konstytucjonalna), z jego formą, wyróżniającą go spośród innych bytów (rola formalna) oraz z jego przeznaczeniem (celem istnienia – rola teliczna). Opisany sposób pojmowania relacji specyfikujących sensy pojęć oraz wyodrębniania pojęć podstawowych (z zakresu tzw. „ontologii szczytowych”³ – ang. *top ontologies*) zastosowano m.in. w dwóch europejskich projektach związanych z budową wielojęzycznych leksykonów semantycznych: EuroWordNet [15] i SIMPLE [13]. Jednakże, w ogólności, problem doboru relacji wiążących pojęcia w ontologiach, pozostaje otwarty.

Do wyrażania ontologii niezbędne jest zdefiniowanie języka. Można go wzorować na językach logiki (głównie logiki pierwszego rzędu i jej ograniczeń), językach reprezentacji wiedzy za pomocą ram, czy też językach modelowania systemów informatycznych. W rozważanej grupie miejsce szczególne zajmują języki przeznaczone do wyrażania ontologii w semantycznej sieci WWW. Mają one swą kilkuletnią historię. W 1999 roku, w ramach konsorcjum W3C opracowano standard RDF (skr. ang. *Resource Description Framework*, [9]), służący do opisywania danych-zasobów internetowych. Zasobem jest każdy byt udostępniany w Internecie za pomocą adresu URI. Standard RDF oferuje zestaw środków do modelowania danych (zasobów) oraz język do wyrażania modeli. Model danych w rozważanym standardzie składa się ze zbioru stwierdzeń (ang. *statements*), które opisują wartości (ang. *object*) przypisywane właściwościom (atrybutom, związkom – ang. *properties*) opisywanych zasobów (ang. *resources*). Wartościami (ang. *object*) mogą być literały znakowe (ang. *literals*) lub zasoby. Dopuszcza się także możliwość tworzenia stwierdzeń „wyższych rzędów” (stwierdzenia o stwierdzeniach) za pośrednictwem mechanizmu reifikacji (ang. *reification*) – wtedy poddane reifikacji stwierdzenie staje się zasobem. Standard RDF poszerzono o system schematów RDFS (skr. ang. *Resource Description Framework Schema*, [4]) zawierających środki do definiowania pojęć, relacji i typów danych. Schematy RDFS, dysponując znaczną siłą wyrazu, nie mają jednak formalnie zdefiniowanej semantyki. Ograniczenia tego nie ma rozwijany w następnych latach język OIL (skr. ang. *Ontology Inference Language*, [7, 8]), który definiuje zestaw mechanizmów do modelowania ontologii wraz z ich składnią i formalną semantyką. Projektanci języka OIL przyjęli ważne założenie o jego warstwowej budowie i rozszerzalności. Warstwy *Core OIL* i *Standard OIL* dostarczają pojęć do modelowania ontologii, a warstwa *Instance OIL* umożliwia posługiwanie się przykładami pojęć („obiettami klas”). Zaproponowano także warstwę *Heavy OIL*, która może zawierać dalsze rozszerzenia. Projektując warstwowy model języka i godząc się na ograniczoną „siłę wyrazu” jego trzech warstw niższych, chciano zapewnić efektywność działania bazujących na OIL narzędzi programistycznych (m.in. systemów wnioskujących), a tym samym możliwość powszechnego ich stosowania w Internecie. Równoległe z językiem

³ Ontologie szczytowe stanowią klasyfikację najogólniejszych kategorii semantycznych. Ich uszczegółowieniami mogą być np. ontologie: dziedzinowe (ang. *domain ontologies*), zorientowane na zadania (ang. *task ontologies*) lub zorientowane na szczególne zastosowania (ang. *application ontologies*).

OIL rozwijano język DAML (skr. ang. *DARPA Agent Markup Language*), a projektanci obu języków postanowili połączyć swe siły w projekcie DAML+OIL, który następnie przekształcono w projekt OWL (skr. ang. *Web Ontology Language*, [1, 16]). Charakterystykę języka OWL przedstawiamy w następnym punkcie.

2. Ontologie w języku OWL

Język OWL jest głównie przeznaczony do definiowania semantyki dokumentów w systemie WWW. Umożliwia formułowanie ontologii będących zbiorami definicji klas (pojęć danej dziedziny) i obiektów (przykładów pojęć), a także relacji (atrybutów⁴ pojęć, związków pomiędzy pojęciami i ich przykładami). Podobnie, jak język OIL, język OWL ma strukturę warstwową (o rosnącym stopniu skomplikowania), a warstwy zwane są „gatunkami sów” (ang. *OWL species*), przy czym:

- warstwa *OWL Lite* umożliwia głównie tworzenie taksonomii pojęć opartej na relacji *is-a*, a definiowanie pojęć poprzez nakładanie więzów (ang. *constraints*) na relacje (ang. *properties*) jest ograniczone do prostych więzów licznosciowych (ang. *cardinality constraints*) o wartościach 0 i 1 (tzn. można zadeklarować, że pewne pojęcie jest zbiorem przykładów powiązanych z przynajmniej jednym albo co najwyżej jednym, albo dokładnie jednym przykładem innego pojęcia); w warstwie *OWL Lite* nie można formułować ekstensjonalnych definicji pojęć;
- warstwa *OWL DL*, odpowiadająca semantycznie logikom deskrypcyjnym (ang. *description logics*, [2]), umożliwia tworzenie złożonych struktur pojęciowych poprzez nakładanie kilku rodzajów więzów na relacje; ograniczeniem w procesie ontologicznego modelowania dziedziny jest niemożność definiowania relacji zachodzących pomiędzy pojęciami (jedyną relacją tej klasy jest wbudowana relacja hierarchizująca typu *is-a*), a jedynie pomiędzy przykładami pojęć; jest to konsekwencją założenia o separacji typów, tzn. klasa (pojęcie) nie może być traktowana intensjonalnie jak indywiduum albo jak relacja, podobnie, relacja nie może być traktowana ani jak indywiduum, ani jak klasa; efektem wprowadzenia tych ograniczeń jest obliczeniowa efektywność i rozstrzygalność⁵ systemów realizujących tę warstwę języka;
- warstwa *OWL Full* nie zawiera omówionych ograniczeń, za to nie ma formalnie zdefiniowanej semantyki i zagwarantowanej efektywności i rozstrzygalności.

W dalszej części zilustrujemy cechy języka na podstawie przykładowej ontologii dotyczącej zwierząt, której wersję zapisaną w języku OIL zaprezentowano w pracy [5]. Ontologia jest zmodyfikowanym wariantem przykładu zaczerpniętego z pracy [7]. Modyfikacja polega na wykorzystaniu ontologii szczytowej i zestawu relacji zaproponowanych w projekcie EuroWordNet. Należy nadmienić, że język OWL jest „zanurzony” składniowo w języku XML, a niektóre jego elementy pochodzą z języka RDF/RDFS. Zatem, zakłada się znajomość obu wymienionych języków.

Ontologia może mieć strukturę modułową, odpowiadającą procesowi jej tworzenia, tzn. nowe ontologie mogą „importować” ontologie istniejące i rozszerzać definicje

⁴ Atrybut jest relacją unarną.

⁵ Logiki deskrypcyjne stanowią rozstrzygalne podzbiory logiki pierwszego rzędu.

zawartych w nich pojęć (wprowadzane rozszerzenia nie mogą negować prawdziwych dotychczas faktów – fakty i wnioski z nich wynikające mogą być jedynie dodawane, nigdy usuwane). W sensie składniowym, ontologia jest dokumentem RDF (wyrażonym w języku XML), który może być powiązany z innymi dokumentami, np. z importowanymi ontologiami. Powiązania mogą mieć także charakter techniczny, dotyczący tzw. XML-owych „przestrzeni nazw” (ang. *namespaces*), umożliwiających generowanie unikatowych identyfikatorów.

Przykładową definicję ontologiczną rozpoczyna nagłówek dokumentu XML, po którym występują jednostki (ang. *entities*) definiujące skrótowe nazwy dla trzech przedrostków opisujących przestrzenie nazw znaczników (w celu ułatwienia posługiwania się nimi w definicjach ontologicznych). Jednostki te uzupełniają standardową definicję DTD typu dokumentu RDF. Są nimi: przestrzeń owl znaczników języka OWL, przestrzeń xsd znaczników języka schematów XML oraz hipotetyczna przestrzeń EWN nazw znaczników związanych z językiem EWN (EuroWordNet). Definicję ontologiczną rozpoczyna znacznik `<rdf:RDF ...>` zawierający specyfikacje przestrzeni nazw (poza omówionymi, także: `rdf` i `rdfs` oraz znaczniki `dc` i `dcq` standardu Dublin Core⁶). Nazwy niekwalifikowane przedrostkami należą do definiwanej ontologii, o hipotetycznym adresie:

`http://www.example.pl/animals.`

```
<?xml version="1.0" ?>
<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY EWN "...#" > ]>
<rdf:RDF
  xmlns      = "http://www.example.pl/animals#"
  xmlns:owl  = "http://www.w3.org/2002/07/owl#"
  xmlns:rdf  = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd  = "http://www.w3.org/2001/XMLSchema#"
  xmlns:dc   = "http://purl.org/dc/elements/1.1/"
  xmlns:dcq  = "http://purl.org/dc/qualifiers/1.0/"
  xmlns:EWN  = "...">
```

Właściwą definicję ontologiczną rozpoczyna nagłówek (ang. *ontology header*) specyfikujący metadane związane z definiowaną ontologią (odpowiada im para znaczników `<owl:Ontology>` i `</owl:Ontology>`), jak np.: jej adres URI, tytuł, autorów, temat, wydawcę, typ, język, dokument źródłowy oraz krótki opis w języku naturalnym. W prezentowanym przykładzie, metadane opisano za pomocą znaczników Dublin Core oraz `rdfs`.

```
<owl:Ontology rdf:about="http://www.example.pl/animals">
  <rdfs:comment> A didactic example ontology describing
    African and Asian animals with the use of
    semantic relations a slots and basic concepts
    from the linguistic top ontology of EWN
```

⁶ Dublin Core ([6] – standard definiowania metadanych), które są stosowane do opisywania zasobów np. w bibliotekach elektronicznych, muzeach, urzędach itp.

```

</rdfs:comment>
<dc:Title>African nad Asian animals</dc:Title>
<dc:Creator>J. Cybulka, J. Martinek based on I.Horrocks
</dc:Creator>
<dc:Subject>animal, food, vegetarians</dc:Subject>
  <dc:Description>A didactic example ontology describing
    African and Asian animals with the use of
    semantic relations as slots and basic
    concepts from the linguistic top ontology
    of EWN
</dc:Description>
<dc:Description>
  <rdf:Description>
    <dcq:descriptionType>Release</dcq:descriptionType>
    <rdf:value>1.01</rdf:value>
  </rdf:Description>
</dc:Description>
<dc:Publisher>J. Cybulka, J. Martinek</dc:Publisher>
<dc>Type>ontology</dc>Type>
<dc:Source>
  http://www.africa.com/nature/animals.html</dc:Source>
<dc:Language>OWL</dc:Language>
<dc:Language>en-uk</dc:Language>
<rdfs:label>Example Animals Ontology</rdfs:label>
</owl:Ontology>

```

Zasadnicza część ontologii składa się z definicji klas, relacji oraz ich przykładów. W przedstawianym przykładzie wszystkie relacje pochodzą z projektu EuroWordNet, zatem ich definicje nie pojawiają się jawnie, są jedynie inkorporowane za pomocą nazw z przestrzeni EWN. Przykładem prostej definicji pojęcia (klasy) jest następująco zdefiniowany termin giraffe:

```
<owl:Class rdf:ID="giraffe"/>
```

Z przytoczonej definicji wynika jedynie, że zbiór bytów będących przykładami pojęcia o identyfikatorze giraffe jest podzbiorem predefiniowanego uniwersum bytów o nazwie Thing. W rozważanej ontologii posłużymy się terminami z zakresu ontologii szczytowej EWN, m.in. pojęciem &EWN;Animals, określającym zbiór bytów będących zwierzętami. Stosując relację taksonomiczną subclassOf (z przestrzeni rdfs) możemy zadeklarować, że klasa giraffe jest szczególnym przypadkiem (podzbiorem) klasy (zasobu w sensie RDF) &EWN;Animals. W definicji pojawia się atrybut rdf:about odnoszący się do istniejącego i wcześniej zdefiniowanego bytu (za jego pomocą można wprowadzać rozszerzenia do istniejących definicji):

```

<owl:Class rdf:about="giraffe">
  <rdfs:subclassOf rdf:resource="&EWN;Animal" />
</owl:Class>

```

Powyższe związki można wyrazić jednostopniowo za pomocą następującej definicji, która dodatkowo zawiera etykietę umożliwiającą nazwanie definiowanej klasy bytów po polsku:

```

<owl:Class rdf:ID="giraffe">
  <rdfs:subClassOf rdf:resource="&EWN;Animal" />
  <rdfs:label xml:lang="pl">żyrafa</rdfs:label>
</owl:Class>

```

Załóżmy, że Zeta jest imieniem pewnej żyrafy. W sensie ontologicznym, Zeta jest przykładem pojęcia *giraffe*, o czym orzeka następujące zdanie:

```

<giraffe rdf:ID="Zeta"/>

```

W prezentowanej ontologii nie definiujemy jawnie relacji, jednakże w celach ilustracyjnych pokażemy, w jaki sposób można by je zdefiniować. W języku OWL występują dwie kategorie relacji binarnych: zachodzących pomiędzy elementami klas (tzw. *ObjectProperty*) oraz definiujących związki pomiędzy elementami klas i danymi innych typów (*DatatypeProperty* – wydaje się, że te drugie z pewnością byłyby traktować jako atrybuty klas). Rozpatrujemy relację „posiadania całości” *HAS_HOLO_PART* jako podrelację ogólniejszej relacji holonimii *HAS_HOLONYM*. Rozważana relacja może zachodzić pomiędzy elementami klas będących uszczegółowieniami klasy *&EWN;Part* bytów stanowiących konstytucjonalnie części i elementami klas będących subclassami klasy *&EWN;Group*, która obejmuje całości. Dodatkowo, relacja holonimii jest przechodnia:

```

<owl:ObjectProperty rdf:ID="HAS_HOLO_PART">
  <rdfs:SubPropertyOf rdf:resource="#HAS_HOLONYM"/>
  <rdf:type rdf:resource="&owl;TransitiveProperty" />
  <rdfs:domain rdf:resource="&EWN;Part" />
  <rdfs:range rdf:resource="&EWN;Group" />
</owl:ObjectProperty>

```

Za pomocą mechanizmu inwersji relacji, można w następujący sposób zdefiniować relację „posiadania części” (*HAS_MERO_PART*):

```

<owl:ObjectProperty rdf:ID="HAS_MERO_PART">
  <owl:inverseOf rdf:resource="#HAS_HOLO_PART"/>
</owl:ObjectProperty>

```

Niektóre relacje, np. relacja reprezentująca „posiadanie koloru” (*BE_IN_STATE/colour*), są funkcjami, co można wyrazić za pomocą następującej definicji:

```

<owl:ObjectProperty rdf:ID="BE_IN_STATE/colour">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
</owl:ObjectProperty>

```

Przykładem relacji definiującej wartość atrybutu „wiek” pewnego typu obiektów jest *BE_IN_STATE/age*, definiowana jako relacja kategorii *DatatypeProperty*, której zakresem są liczby całkowite nieujemne zdefiniowane jako typ w schemacie XML:

```

<owl:DatatypeProperty rdf:ID="BE_IN_STATE/age">
  <rdfs:range rdf:resource="&xsd;NonNegativeInteger"/>
</owl:DatatypeProperty>

```

Ciekawym mechanizmem, wynikającym ze stosowania logik deskrypcyjnych, jest możliwość tworzenia klas poprzez nakładanie więzów na relacje. Na przykład, można

zdefiniować klasę zawierającą wyłącznie (AllValuesFrom) lub również (SomeValuesFrom) elementy będące częściami wskazanych obiektów. Ta nowa klasa⁷ powstanie poprzez ograniczenie zakresu relacji „posiadania całości”. W charakterze przykładu, zdefiniujemy klasę tree jako uszczegółowienie klasy &EWN;Plant a także klasy: branch i leaf. Klasa branch jest, po pierwsze, uszczegółowieniem klasy &EWN;Natural wskazującym na naturalne pochodzenie definiowanego bytu. Po drugie, wśród przykładów pojęcia branch istnieje przynajmniej jeden, który jest częścią drzewa (inne mogą być na przykład częściami krzewów). Obie wspomniane właściwości wyznaczają zbiory elementów, a zbiór wynikowy (zbiór przykładów definiowanej klasy branch) powstaje na ich przekroju, mimo że w definicji nie używa się jawnie operatora przecięcia zbiorów. Podobne właściwości ma klasa leaf. Dodatkowo, wobec przechodniości relacji holonimii, systemy wnioskujące powinny wydedukować, że holonimem liścia jest nie tylko gałąź, lecz również drzewo. Oto opisane definicje:

```
<owl:Class rdf:ID="tree">
  <rdfs:subClassOf rdf:resource="&EWN;Plant" />
</owl:Class>
<owl:Class rdf:ID="branch">
  <rdfs:subClassOf rdf:resource="&EWN;Natural" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&EWN;HAS_HOLO_PART" />
      <owl:someValuesFrom rdf:resource="#tree" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="leaf">
<rdfs:subClassOf rdf:resource="&EWN;Natural" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&EWN;HAS_HOLO_PART" />
      <owl:someValuesFrom rdf:resource="#branch" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Szczególnym rodzajem więzu jest hasValue umożliwiający tworzenie zbioru elementów mających pewną określoną, pojedynczą wartość jako zakres relacji – na przykład: słonie, poza tym, że są roślinożercami, są także zwierzętami o szarym kolorze skóry (indywiduum określające kolor szary jest zakresem relacji BE_IN_STATE/colour):

```
<owl:Class rdf:ID="elephant">
  <rdfs:subClassOf rdf:resource="#herbivore" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&EWN;BE_IN_STATE/colour" />
      <owl:hasValue rdf:resource="#grey" />
    </owl:Restriction>
```

⁷ Będzie to tzw. „klasa anonimowa”, zdefiniowana lokalnie i nie mająca nazwy.

```

    </rdfs:subClassOf>
</owl:Class>

```

Można także nakładać ograniczenia na liczbę elementów powiązanych za pomocą wskazanej relacji. Istnieją trzy warianty takich więzów: liczba maksymalna (*maxCardinality*), minimalna (*minCardinality*) i dokładna (*Cardinality*). Jeśli klasę dorosłych słońi określimy jako uszczegółowienie klasy słońi z dodatkowym ograniczeniem na wiek (co najmniej 15 lat) i wagę (z przedziału 1500-3000 kg), to uzyskamy zapis:

```

<owl:Class rdf:ID="adult_elephant">
  <rdfs:subClassOf rdf:resource="#elephant" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&EWN;BE_IN_STATE/age" />
      <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger"> 15
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&EWN;BE_IN_STATE/weight" />
      <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger"> 1500
      </owl:minCardinality>
      <owl:maxCardinality
        rdf:datatype="&xsd;nonNegativeInteger"> 3000
      </owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

W dotychczasowych definicjach pojęć posługiwaliśmy się tzw. „klasami prostymi”. Język OWL umożliwia tworzenie złożonych wyrażeń klasowych (ang. *class expressions*), w których znajdują zastosowanie operatory mnogościowe: sumy (*unionOf*), przecięcia (*intersectionOf*), dopełnienia (*complementOf*), a także operator ekstensjonalnego tworzenia skończonych zbiorów (*oneOf*) i operator wskazywania rozłączności zbiorów (*disjointWith*). Zaczernięte z ontologii szczytowej słownika EuroWordNet pojęcie *Animal* możemy rozszerzyć wskazując, że jest ono sumą rozłącznych parami pojęć: *herbivore*, *carnivore* i *omnivore*, z których każde jest uszczegółowieniem pojęcia wyjściowego *Animal*:

```

<owl:Class rdf:ID="herbivore">
  <rdfs:subClassOf rdf:resource="&EWN;Animal" />
</owl:Class>
<owl:Class rdf:ID="carnivore">
  <rdfs:subClassOf rdf:resource="&EWN;Animal" />
</owl:Class>
<owl:Class rdf:ID="omnivore">
  <rdfs:subClassOf rdf:resource="&EWN;Animal" />
</owl:Class>

```



```

<owl:Class rdf:about="&EWN;Animal">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#carnivore" />
    <owl:Class rdf:about="#herbivore" />
    <owl:Class rdf:about="#omnivore" />
  </owl:unionOf>
</owl:Class>
<owl:Class rdf:about="#carnivore">
  <owl:disjointWith rdf:resource="#herbivore" />
</owl:Class>
<owl:Class rdf:about="#herbivore">
  <owl:disjointWith rdf:resource="#omnivore" />
</owl:Class>
<owl:Class rdf:about="#omnivore">
  <owl:disjointWith rdf:resource="#carnivore" />
</owl:Class>

```

Przykładem zastosowania operatora tworzenia zbioru (pojęcie `grey_colour`) przez wyliczenie jego elementów może być definicja jednoelementowej klasy zawierającej indywiduum reprezentujące kolor szary:

```

<owl:Class rdf:ID="grey_colour">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#grey" />
  </owl:oneOf>
</owl:Class>
<grey_colour rdf:ID="grey" />

```

W języku OWL można wskazywać zachodzenie związków pomiędzy wskazanymi parami indywiduów: jeśli Azja jest przykładem pojęcia kontynent, a Indie są przykładem pojęcia kraj oraz założymy, że kraje są częściami kontynentów, to możemy użyć relacji `HAS_HOLO_PART` do powiązania rozważanych indywiduów:

```

<continent rdf:ID="Asia" />
<country rdf:ID="India">
  <HAS_HOLO_PART rdf:resource="#Asia" />
</country>

```

O powszechności stosowania ontologii w dużym stopniu zdecyduje istnienie współdzielonych definicji ontologicznych (rozproszonych bibliotek ontologii), adaptowalnych w różnych aplikacjach. W takich wypadkach, w procesie dołączania ontologii bibliotecznych często będzie pojawiać się konieczność utożsamiania pojęć, relacji i indywiduów pochodzących z różnych źródeł. Język OWL udostępnia mechanizmy `equivalentClass` i `equivalentProperty` do wskazywania równoważności, odpowiednio, pojęć i relacji oraz mechanizm `sameIndividualAs`, umożliwiający utożsamianie przykładów pojęć. W odniesieniu do indywiduów, istnieje również odwrotna do omówionej

możliwość rozróżniania przykładów pojęć za pośrednictwem deklaracji `differentIndividualFrom`.

3. Podsumowanie

W pracy scharakteryzowano uniwersalny język przeznaczony do formułowania ontologii, które mogą być wykorzystywane w semantycznej sieci WWW. Język OWL ma budowę trójwarstwową, przy czym warstwy pierwsza i druga bazują semantycznie na logikach deskrypcyjnych, co ułatwia tworzenie efektywnych narzędzi programistycznych, jak np. edytory strukturalne, systemy wnioskujące, filtry i wyszukiwarki danych. Do najbardziej popularnych systemów umożliwiających tworzenie ontologii poziomu OWL DL należą: Protégé (uniwersytet Stanford), KAON (uniwersytet w Karlsruhe) i OilEd (uniwersytet w Manchester). Znane są również systemy wnioskujące FaCT i RACER dla logik deskrypcyjnych, które można zastosować m.in. do sprawdzania niesprzeczności ontologii, wywnioskowywania niejawnych związków pomiędzy pojęciami i dedukowania przynależności indywidualów do klas.

Pewną niedogodnością języka OWL jest jego składnia, wyrażona w języku XML i zawierająca zbyt wiele elementów technicznych, nieczytelnych dla człowieka. Przydałby się zatem wariant publikacyjny składni języka, ułatwiający zapoznawanie się z jego walorami semantycznymi.

Zaprezentowany w pracy przykład pokazuje, że stosunkowo łatwo można stosować w ontologiach OWL-owych osiągnięcia innych dziedzin, na przykład rozwiązania wypracowane w obszarze wielojęzycznych leksykonów semantycznych. Ciekawym wydaje się pomysł zdefiniowania w OWL-u prototypowych wersji ontologii szczytowych SIMPLE i EuroWordNet oraz zaproponowanych tam zestawów relacji i cech semantycznych, a następnie ocena ich właściwości za pomocą systemów wnioskujących.

Literatura

- [1] Antoniou, G., Harmelen, F. van, Web Ontology Language: OWL, w Handbook on Ontologies (Staab, S., Studer, R. eds), Springer Verlag, Berlin, Heidelberg 2004.
- [2] Baader, F. et al. (eds.), *The Description Logic Handbook. Theory, Implementation and Applications*, Cambridge Univ. Press, Cambridge 2003.
- [3] Berners-Lee, T. *Semantic Web road map*, <http://www.w3.org/DesignIssues/semantic.html>.
- [4] Brickley, D., Guha, R. V., *Resource Description Framework (RDF) Schema Specification 1.0*, <http://www.w3.org/TR/2000/CR-rdf-schema-20000327>.
- [5] Cybulka, J., Martinek, J., Semantic Roles as Slots in OIL ontologies, *Computers and the Humanities* 38 (2) (2004), 129-148.
- [6] Dublin Core Metadata, <http://purl.oclc.org/dc/>.
- [7] Horrocks, I., Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., Harmelen, F.

- van, Klein, M., Staab, S., Studer, R., Motta, E., *The Ontology Inference Layer OIL*, raport techniczny, <http://www.cs.vu.nl/~dieter/oil/Tr/oil.pdf>.
- [8] Horrocks, I., *A Denotational Semantics for Standard OIL and Instance OIL*, raport techniczny, <http://www.ontoknowledge.org/oil/download/semantics.pdf>.
- [9] Lassila, O., Swick, R. R., *Resource Description Framework (RDF): Model and Syntax Specification*, rekomendacja W3C, <http://www.w3.org/TR/REC-rdf-syntax/>.
- [10] Łabuzek, M., *Wykorzystanie metamodelowania do specyfikacji ontologii znaczenia opisów rzeczywistości*, <http://e-informatyka.pl/article/show-bw/90>.
- [11] Pustejovsky, J., The Generative Lexicon, *Computational Linguistics*, 17 (4) (1991), 409-441.
- [12] Rodríguez, H., Climent, S., Vossen, P., Bloksma, L., Peters, W., Alonge, A., Bertagna, F., Roventini, A., The Top-Down Strategy for Building EuroWordNet: Vocabulary Coverage, base Concepts and Top Ontology, *Computers and the Humanities*, 32 (1998), 117-152.
- [13] *SIMPLE Work Package 2 Linguistic Specification report Deliverable D2.1, 2000*, <http://www.ub.es/gilcub/SIMPLE/simple.html>.
- [14] Such, J., Szcześniak, M., *Ontologia przyrodnicza, Pisma Filozof.*, t LXXXI, Wyd. Naukowe Instytutu Filozofii UAM, Poznań, 2001.
- [15] Vossen, P. (ed.) *EuroWordNet General Document*, 1999, <http://www.hum.uva.nl/~ewn/docs/GeneralDocPS.zip>.
- [16] *Web Ontology Language (OWL) Guide Version 1.0*, <http://www.w3.org/TR/owl-guide>.

OWL – an Ontology Definition Language for the Semantic Web

Jolanta Cybulka

Poznań University of Technology, Institute of Control and Information Engineering
 Pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Poland
e-mail: Jolanta.Cybulka@put.poznan.pl

Abstract. The paper contains a concise description of the OWL language (*Web Ontology Language*) which is especially useful to formulate ontologies meant as artefacts applied in different domains of interest. Such ontologies may be used, for instance, to semantically annotate the contents of Web documents. OWL has several predecessors, such as RDF, DAML, OIL, DAML+OIL and has been developed by W3C group as a standard in the considered class of languages.

Key words: EuroWordNet, metadata, ontology, OWL, Semantic Web, SIMPLE