STUDIA INFORMATICA 2006
Volume 27 Number 2 (67)

Krzysztof GOCZYŁA, Teresa ZAWADZKA
Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki

ONTOLOGIE W SIECI SEMANTYCZNEJ

Streszczenie. Wraz z nastaniem ery Internetu i jego gwałtownym rozwojem, zasadniczym problemem dla współczesnej informatyki stała się automatyzacja pozyskiwania olbrzymich zasobów wiedzy ludzkiej w nim zgromadzonych. Wiedza ta ma bardzo zróżnicowany charakter z uwagi na wielość formatów zapisu danych, a przede wszystkim z uwagi na różny stopień jej ustrukturalizowania. Jedną z najbardziej popularnych idei, dążących do systematycznego podejścia do pozyskiwania wiedzy z Internetu stała się, tzw. „inicjatywa Semantic Web". Zgodnie z tą ideą, wiedza ludzka powinna być strukturalizowana w formie ontologii publikowanych w Internecie, w powszechnie zaakceptowanym, precyzyjnym i możliwym do przetwarzania przez komputery, formacie. W tym artykule bliżej prezentujemy problemy, związane z budową ontologii i ich wykorzystaniem w ramach Semantic Web (Sieci Semantycznej). Problemy te przedstawiane są na szerszym tle różnych metod maszynowej reprezentacji wiedzy. Prezentujemy także podejście alternatywne, polegające na automatycznym tworzeniu „ontologii wszystkiego", na podstawie analizy tekstu stron WWW w języku naturalnym.

Słowa kluczowe: reprezentacja wiedzy, ontologie, ramki, logika opisowa, Sieć Semantyczna

ONTOLOGIES OVER THE SEMANTIC WEB

Summary: One of the major and most challenging tasks for modern Information Technology is development of methods aimed at automatically acquiring and processing knowledge stored in the biggest information repository the man has ever created – the Internet. This knowledge is of miscellaneous nature, mainly due to the fact that it is stored in many languages and in numerous formats with different levels of structuring. The "Semantic Web initiative" strives to achieve this goal by structuring the contents of Internet into publicly available and shared ontologies formulated in a commonly accepted, machine readable format. In this paper we discuss problems of building ontologies and using them throughout Semantic Web. Relevant topics are presented in the broader context of knowledge representation methods. We also present an alternative approach based on processing textual Web contents, extracting seman-

tics from them and creating a "general ontology" to be used to present knowledge to a user.

Keywords: knowledge representation, ontologies, frames, description logics, Semantic Web

1. Wprowadzenie

Wiedza – ostatnio to słowo coraz częściej słyszy się w kontekstach, związanych z informatyką. O ile wiek XX był bez wątpienia – dzięki komputerom oczywiście – wiekiem informacji, o tyle mówi się, że wiek XXI będzie wiekiem wiedzy (także dzięki coraz „mądrzejszym" komputerom), a społeczeństwo tego wieku będzie „społeczeństwem opartym na wiedzy" (*knowledge-based society*). A zatem, wypadałoby postawić pytanie, czym jest wiedza? O ile w wypadku wiedzy ludzkiej intuicyjnie czujemy, czym ona jest, o tyle „wiedza komputerowa" wydaje się mieć wiele różnych interpretacji i przybliżeń. Ale zacznijmy od wiedzy ludzkiej. Najprościej można powiedzieć, że wiedza jest to zbiór wiadomości i umiejętności nabytych przez człowieka drogą uczenia się. Ale czy ta prosta definicja daje nam jakieś wskazówki co do tego, czym jest „wiedza komputerowa"? W pierwszej chwili może wydawać się, że tak. Komputery też się uczą. *Machine learning* jest przecież poważnym działem informatyki albo raczej poważnym działem sztucznej inteligencji. Jednak wszyscy zdajemy sobie sprawę z tego, że uczenie się człowieka od „uczenia się" (cudzysłów użyty bardzo celowo!) komputerów wciąż jeszcze dzieli przepaść. A zatem, definicja wiedzy w wypadku komputerów powinna być inna, a brzmieć może na przykład tak: „Wiedza jest to zbiór danych i informacji, zapisanych w pamięci komputera oraz takich danych i informacji, które komputer potrafi z tych pierwszych wywnioskować". [1]

Aby komputery mogły „wiedzieć", musimy tę wiedzę dla nich odpowiednio przygotować. To przygotowanie polega oczywiście na odpowiednim zapisie. Co to znaczy „odpowiednim"? To znaczy jednoznacznym, a zarazem adekwatnym (dostatecznie ekspresywnym) dla zagadnienia, które chcemy komputerowi opisać. W niniejszym artykule zajmiemy się właśnie tym problemem, w kontekście wiedzy zgromadzonej w nieprzebranych zasobach Internetu. Opowiemy o tym, jak informatycy próbują tę wiedzę uporządkować i uczynić zrozumiałą dla komputerów (i – paradoksalnie – przy okazji także dla ludzi).

Układ tego artykułu jest następujący: W rozdziale 2 wprowadzimy Czytelnika w różne metody reprezentacji wiedzy, koncentrując się na metodach ontologicznych. W rozdziale 3

[1] Czytelnik może spytać, czym się różnią dane od informacji. Nie będziemy tu rozwodzić się nad niuansami terminologicznymi, dlatego też w powyższej definicji występują zarówno dane, jaki i informacje. W najprostszym ujęciu przykładem danej jest liczba „2" lub napis „komputer", a informacją to, że „Komputer Jacka ma 2 GB pamięci".

opiszemy Sieć Semantyczną i czym są ontologie w tej Sieci. W rozdziale 4 zajmiemy się pokrótce głównymi problemami inżynierii ontologii. W rozdziale 5 naszkicujemy pewne podejście alternatywne, bazujące na przetwarzaniu języka naturalnego. Na koniec dokonamy podsumowania, na tyle, na ile można podsumować coś, co rozwija się tak dynamicznie.

2. Metody reprezentacji wiedzy

Bez wątpienia najpowszechniejszą metodą reprezentacji wiedzy jest jej zapis w języku naturalnym. Bez wielkiego ryzyka można założyć, że cała, istotna wiedza ludzkości zapisana jest w ten właśnie sposób (nazwiemy go niestrukturalnym). Ale jest to zarazem sposób, który jest najmniej odpowiedni dla komputerów, właśnie z uwagi na ów brak struktury, którego – we wszystkich zastosowaniach – tak nie lubią komputery. Stąd też już wiele lat temu podjęto badania w dziedzinie maszynowych metod reprezentacji wiedzy, czyli metod przeznaczonych dla zapisu wiedzy w pamięci komputerów. Nie znaczy to jednak, że zarzucono badania w dziedzinie metod niestrukturalnych. Wręcz przeciwnie – dziedzina informatyki zwana przetwarzaniem języka naturalnego (*Natural Language Processing*, *NLP*) rozwija się dynamicznie i prowadzone są w niej poważne projekty, ukierunkowane na wydobywanie semantyki z tekstu, zamieszczonego na przykład na stronach HTML. Wspomnimy o tym jeszcze w rozdziale 5.

Skupmy się na razie na metodach strukturalnych. Wszystkie z nich oparte są na solidnych podstawach logiki matematycznej. Wprowadzimy tu dość arbitralny podział tych metod na:

- metody regułowe,
- metody ontologiczne.

W kolejnych punktach przybliżymy te metody, koncentrując się na metodach ontologicznych.

2.1. Metody regułowe

W metodach regułowych wiedza jest reprezentowana jako zbiór reguł. Spośród różnych rodzajów reguł, największą popularność zyskały klauzule Horna o postaci:

$$\mathbf{A}_1,\ldots, \mathbf{A}_n \rightarrow \mathbf{B},$$

gdzie $\mathbf{A}_i$ i **B** są atomowymi formułami logicznymi. Znaczenie takiej reguły jest następujące: jeśli wszystkie $\mathbf{A}_i$ są prawdziwe, to także **B** jest prawdziwe. W szczególnym przypadku wszystkie formuły $\mathbf{A}_i$ mogą być puste, co oznacza bezwarunkową prawdziwość **B**. Dla reguł tego typu opracowano efektywne metody przetwarzania (dowodzenia i wnioskowania). Wystarczy tu wspomnieć język Prolog i jego możliwości w dowodzeniu „w przód" (*forward*

chaining) i „do tyłu” (*backward chaining*). Regułowy zapis wiedzy zawiera nie tylko reguły, ale także asercje unarne i binarne (które jednak mogą być traktowane jako specjalny przypadek klauzul Horna), zwane faktami. Na przykład:

jestMatką (x, y), jestSiostrą (z, x) $\rightarrow$ jestCiotką (z, y)

jest jednoelementowym zbiorem reguł, a

jestMatką (Janina, Ala); jestSiostrą (Anna, Janina)

jest dwuelementowym zbiorem faktów. Oczywiście, nietrudno udowodnić, że prawdziwa jest też asercja jestCiotką (Anna, Ala).

Przytoczone wyżej reguły są regułami dedukcyjnymi (tj. takimi, w których **B** jest formułą logiczną). Ale można sobie również wyobrazić inny rodzaj reguł, w których **B** jest czynnością (zwaną też akcją). Takie reguły nazywamy reaktywnymi lub proceduralnymi. Reprezentują one wiedzę dynamiczną, w odróżnieniu od wiedzy statycznej, zapisywanej regułami dedukcyjnymi. Reguły proceduralne znajdują duże zastosowanie, na przykład w systemach agentowych. Ponieważ programowe agenty odgrywają dużą rolę w Sieci Semantycznej (o czym więcej później), to nie dziw, że w ramach konsorcjum W3C, promującym Sieć Semantyczną, opracowywane są specjalne języki do zapisu reguł. Zainteresowanego czytelnika odsyłamy do dokumentacji języka RuleML [18], który umożliwia zapis zarówno reguł dedukcyjnych, jak i proceduralnych.

2.2. Ramki Minsky’ego i sieci semantyczne

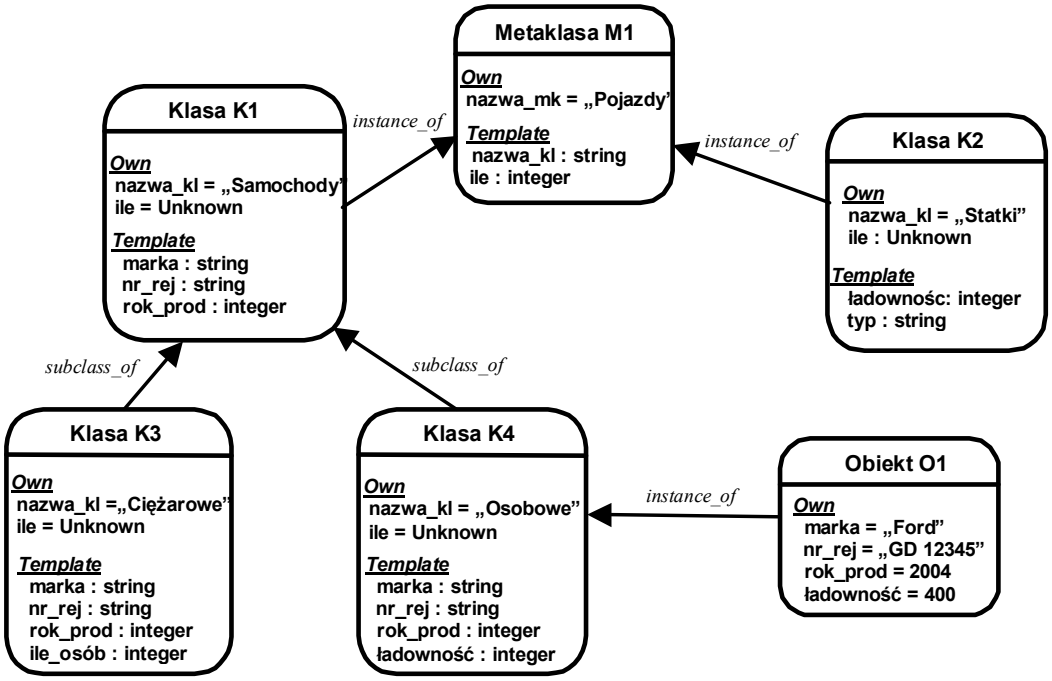
Centralną tematyką tego artykułu są jednak ontologie, a więc skupmy się teraz na ontologicznych metodach reprezentacji wiedzy, odsyłając Czytelnika, zainteresowanego metodami regułowymi, do bogatej literatury przedmiotu (np. [19]). Zacznijmy od samego terminu „ontologia”. Ma on swoje korzenie w filozofii: ontologia – inaczej teoria bytu (lub bytów) – to podstawowy dział filozofii, zajmujący się badaniem charakteru i struktury rzeczywistości. W informatyce termin ten tak naprawdę niewiele odbiega od swojego znaczenia filozoficznego, choć nabrał sensu bardziej materialnego (co znalazło swoje odzwierciedlenie w tym, że używamy liczby mnogiej: „ontologie”). Nieformalnie mówiąc, ontologia jest strukturalnym (formalnym) opisem pojęć, występujących w danej dziedzinie. Ontologię możemy traktować jako encyklopedię lub słownik, do których sięgamy po wiedzę na pewien temat. Inna, nieco bardziej precyzyjna, definicja ontologii głosi, że ontologia jest to bezpośrednia (tzn. wyrażona w określonym języku, najlepiej: zrozumiałym dla komputera) specyfikacja konceptualizacji, rozumianej jako abstrakcyjny model pewnych aspektów świata, opisany w terminach konceptów, ich właściwości i relacji pomiędzy konceptami. W tej definicji pojawia się pojęcie konceptu, które jest zasadnicze dla każdej ontologicznej metody reprezentacji wiedzy.

Za najwcześniejszą ontologiczną metodę reprezentacji wiedzy (mamy na myśli oczywiście metody ukierunkowane na przetwarzanie komputerowe) uważane są ramki (*frames*), wprowadzone przez M. Minsky'ego w 1975 roku [6], [15]. Według Minsky'ego, każdy element świata może być reprezentowany w postaci bytu, zwanego ramką. Dla celów ontologicznej reprezentacji świata, najważniejsze ramki to klasy (będące w istocie odpowiednikiem konceptów). Każda klasa zawiera klatki (*slots*), reprezentujące właściwości klasy. Klatki też są ramkami, do których dodatkowo mogą być przypisane, tzw. fasety (*facets*) – właściwości klatek, zwykle nakładające ograniczenia na klatki. Klatki dowolnej ramki mogą być albo własne (*own*), albo szablonowe (*template*). Klatki własne – jak sama nazwa wskazuje – są własnością danej ramki i ich wartości są prywatne dla tej ramki. Ramki szablonowe natomiast służą do produkowania innych ramek, będących wystąpieniami (*instances*) tych pierwszych. Zasada jest taka, że klatki szablonowe danej klasy stają się klatkami własnymi wszystkich jej wystąpień. Jeśli z kolei wystąpienie danej klasy ma swoje klatki szablonowe, to ta pierwsza klasa staje się metaklasą („klasą klas"). Klasy mogą też przejmować (dziedziczyć) od siebie klatki, zarówno własne, jak i szablonowe. Ramka, która nie ma klatek szablonowych, nie może mieć swoich wystąpień i wówczas jest obiektem. Zauważmy jeszcze, że na wszystkie ramki, składające się na daną ontologię, mogą być dodatkowo nałożone aksjomaty wyrażone w języku logiki.

Te, na pierwszy rzut oka, dość skomplikowane zasady zilustrujemy przykładową ontologią (rys. 1). W korzeniu tej ontologii występuje ramka oznaczona M1. Ma ona jedną klatkę własną „nazwa_mk" i dwie klatki szablonowe: „nazwa_kl" i „ile". Napisy „string" i „integer", występujące po nazwach klatek są w istocie fasetami, przypisanymi do odpowiednich klatek. Zauważmy, że klatka „nazwa_mk" ma ustaloną wartość „Pojazdy". Ramka M1 ma dwa wystąpienia: ramki K1 i K2. Klatki szablonowe ramki M1 stały się klatkami własnymi ramek K1 i K2, z konkretnymi wartościami (choć niektóre z tych wartości mogą nie być znane na etapie tworzenia ontologii, jak klatki „ile"). Ponieważ ramki te mają swoje klatki szablonowe, są klasami, a zatem M1 jest metaklasą. Wszystkie klatki własne i szablonowe klasy K1 są dziedziczone przez ramki K3 i K4, które w ten sposób też są klasami, zwanymi podklasami klasy K1 (K1 nie jest jednak metaklasą, gdyż żadna z ramek szablonowych K1 nie staje się klatką własną K3 lub K4). Ramka O1 jest wystąpieniem klasy K4, gdyż klatki szablonowe K4 stały się klatkami własnymi O1 i ponadto, O1 nie ma klatek szablonowych (w przeciwnym razie ramka O1 byłaby klasą, a ramka K4 stałaby się metaklasą).

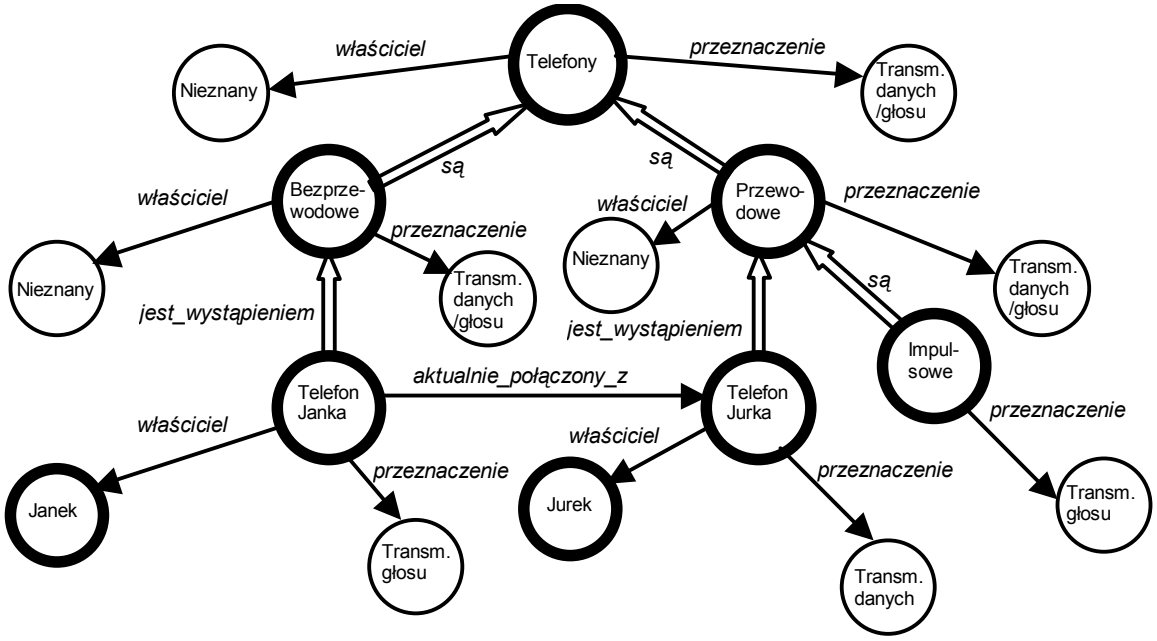
Idei ramek Minsky'ego zarzucano zbytnią ogólność, niejednoznaczność, a także brak ścisłej, formalnej definicji (nieco później opracowano formalizm dla ramek o nazwie F-Logic [21]). Wszystko to powoduje komplikacje w komputerowym przetwarzaniu i wnioskowaniu, a pojęcie metaklasy prowadzi nawet do nierozstrzygalności pewnych problemów wnioskowania. Jednak mimo tych niedoskonałości ramki stały się podstawą standardowego interfejsu

OKBC (*Open Knowledge Base Connectivity*), przeznaczonego do jednolitej reprezentacji wiedzy i dostępu do ontologicznych baz wiedzy. Ponadto, jak nietrudno zauważyć, idea ramek jako elementów świata, rozszerzona o aspekty behawioralne, znalazła swoje wyraziste odzwierciedlenie w obiektowych metodykach inżynierii oprogramowania.



Rys. 1. Przykładowa ontologia wyrażona w formie zbioru ramek
Fig. 1. An exemplary ontology as a set of frames

Inną, rozwijaną równolegle z ramkami, ontologiczną metodą reprezentacji wiedzy, są sieci semantyczne (*semantic networks*) [6]. Sieć semantyczna jest zbiorem klas i obiektów, powiązanych ze sobą różnorodnymi relacjami. Relacje te tworzą dwie hierarchie: hierarchię abstrakcji (oznaczoną na rys. 2 strzałkami blokowymi) oraz hierarchię właściwości (na rys. 2 strzałki zwykłe). Hierarchia abstrakcji obejmuje relację uszczegółowienia pomiędzy klasami („są") oraz relację egzemplifikacji („jest wystąpieniem"), zachodzącą pomiędzy obiektem a klasą. Hierarchia właściwości może obejmować dowolne zależności pomiędzy obiektami i klasami, a także pomiędzy obiektami i klasami a wartościami. Na rys. 2 klasy i obiekty oznaczone są okręgami pogrubionymi, a wartości – zwykłymi. Niektóre relacje mają predefiniowane znaczenie, jak na przykład relacja „zawiera", oznaczająca agregację. W zależności od przyjętego modelu sieci, podklasa dziedziczy wszystkie atrybuty lub też mogą istnieć atrybuty niedziedziczone (dopuszczalne są wyjątki). W sieci z rys. 2 klasa „Impulsowe" nie dziedziczy od klasy „Przewodowe" atrybutu „właściciel" (dziedziczonego z kolei przez klasę „Przewodowe" od klasy „Telefony"). Należy to odczytać tak, że atrybut „właściciel" jest ważny dla wszystkich telefonów, z wyjątkiem telefonów impulsowych.



Rys. 2. Przykład sieci semantycznej
Fig. 2. A semantic network

Nietrudno zauważyć analogie pomiędzy reprezentacją ramkową a sieciami semantycznymi. Wprawdzie w sieciach semantycznych nie występują metaklasy, to jednak pod względem możliwości wnioskowania mają wszystkie inne wady ramek, przede wszystkim nieprecyzyjność, zbytnią ogólność i trudności w przetwarzaniu komputerowym.

2.3. Ontologie oparte na logice opisowej

Ścisłym formalizmem, służącym do zapisu ontologii, jest logika opisowa, zwana też logiką deskrypcyjną (*Description Logic – DL* [2]). Formalizm ten, choć rozwijany od blisko 30 lat, ostatnio zyskuje bardzo na popularności, z uwagi na rozwój idei Sieci Semantycznej ([4, 22] i języka OWL-DL [16]. Logika opisowa jest formalizmem, będącym rozstrzygalnym podzbiorem rachunku predykatów pierwszego rzędu. Z tego też względu nadaje się do algorytmizacji i przetwarzania komputerowego. Formalizm ten ma jawną interpretację teoriomnogościową, co sprawia, że jest łatwo zrozumiały dla osób przyzwyczajonych do myślenia w kategoriach bazodanowych. Logika opisowa w istocie obejmuje całą rodzinę języków, przeznaczonych do definiowania ontologii o różnym stopniu ekspresji i różnych możliwościach wnioskowania. Podejście do reprezentowania wiedzy o świecie w formie ontologii, opartych na logice opisowej (w skrócie: ontologii DL), bazuje na następujących, prostych i naturalnych, a zarazem bardzo ogólnych założeniach:

1. Istnieje pewne **uniwersum** (zwane też dziedziną zainteresowań), które chcemy opisać w formie ontologii.
2. Elementy tego uniwersum, zwane **osobnikami**, są wystąpieniami **konceptów**.
3. Koncepty są ze sobą powiązane binarnymi relacjami, zwanymi **rolami**.

Zgodnie z tymi założeniami, ontologia DL składa się z dwóch części. Pierwsza z nich, *TBox*, zawiera **terminologię** ontologii, na którą składa się zbiór konceptów, zbiór ról oraz zbiór aksjomatów, definiujących ograniczenia nałożone na koncepty i role. Druga część ontologii, *ABox*, to **opis świata**, obejmujący zbiór asercji unarnych, opisujących wystąpienia konceptów, oraz zbiór asercji binarnych, opisujących wystąpienia ról. Ontologia DL nie musi zawierać obu tych części; bardzo często mamy do czynienia z ontologiami, zawierającymi tylko część terminologiczną, choć można też wyobrazić sobie ontologię zawierającą tylko opis świata.

Ontologie stanowią dane dla baz wiedzy. Aby ontologie mogły być przetwarzane w bazach wiedzy, w szczególności – aby mogły być przedmiotem wnioskowania, baza wiedzy musi posiadać moduł (silnik) wnioskujący (*inference engine*), a także musi istnieć określony język zapisu ontologii. Każdy język z licznej rodziny języków logiki opisowej zawiera pewne, podstawowe elementy składowe. Są to:

- koncepty atomowe, w tym koncept uniwersalny $\top$ (Top), reprezentujący uniwersum, oraz koncept pusty $\bot$ (Bottom), reprezentujący koncept, który nie może mieć żadnych wystąpień;
- role atomowe;
- konstruktory służące do tworzenia złożonych konceptów i ról.

Jako język zapisu ontologii przyjmiemy język oznaczany symbolem $\mathcal{ALC}$. Język ten jest dość prosty, ale jego siła ekspresji wystarcza do definiowania nietrywialnych, użytecznych w praktyce ontologii. W tabeli 1 przedstawiono konstruktory tego języka.

Tabela 1

Konstruktory języka $\mathcal{ALC}$

Konstruktor	**Znaczenie**
$\neg$C	Negacja konceptu
C $\sqcap$ D	Przecięcie (część wspólna) konceptów
C $\sqcup$ D	Suma (unia) konceptów
$\exists$R.C	Kwantyfikacja egzystencjalna: zbiór takich osobników, które są powiązane przynajmniej jeden raz rolą *R* z osobnikiem należącym do konceptu *C*
$\forall$R.C	Kwantyfikacja ogólna: zbiór takich osobników, których wszystkie istniejące powiązania rolą *R* dotyczą osobników należących do konceptu *C* (obejmuje także takie osobniki, które nie są powiązane rolą *R* z żadnymi osobnikami)

Dla ilustracji wprowadzonych konstruktorów rozpatrzmy prostą ontologię, opisującą koligacje rodzinne. Ontologia ta zawiera koncepty: Osoba, Mężczyzna, Kobieta i Rodzic oraz rolę maDziecko. Wzajemne zależności pomiędzy elementami tej ontologii oraz opis świata zawarto w tabeli 2.

Tabela 2

Przykładowa ontologia DL

TBox	**ABox**
Mężczyzna ⊑ Osoba	Kobieta (Anna)
Kobieta ⊑ Osoba	Kobieta (Joanna)
Kobieta ⊓ Mężczyzna ≡ ⊥	Mężczyzna (Karol)
Rodzic ≡ Osoba ⊓ ∃maDziecko.Osoba	maDziecko (Anna, Joanna)
Ojciec ≡ Mężczyzna ⊓ Rodzic	maDziecko (Anna, Karol)
Matka ≡ Kobieta ⊓ Rodzic	

TBox tej ontologii zawiera aksjomaty stwierdzające, że Mężczyzna i Kobieta to Osoby, nikt nie może być jednocześnie Mężczyzną i Kobietą. Rodzic to Osoba, która ma choć jedno dziecko będące Osobą, a Ojciec i Matka to Rodzice, którzy są odpowiednio Mężczyzną i Kobietą. ABox zawiera pewne znane fakty o Annie, Joannie i Karolu. Na przykładzie tej ontologii wyjaśnimy podstawowe problemy wnioskowania z ontologii DL.

Zawieranie (*subsumption*): Koncept C zawiera (*subsumes*) koncept D (oznaczenie: $D \sqsubseteq C$), jeśli zbiór wystąpień konceptu D jest zawsze podzbiorem zbioru wystąpień konceptu C. W naszym przykładzie niektóre zależności zawierania podane są jawnie, w postaci aksjomatów. Inne, niejawne zależności zawierania, to np. Ojciec ⊑ Osoba i Matka ⊑ Osoba.

Spełnialność (*satisfiability)*: Koncept C jest spełnialny, jeśli może zawierać wystąpienia. Innymi słowy, koncept C nie jest spełniany, jeśli $C \equiv \bot$. W naszym przykładzie wszystkie koncepty są spełnialne. Gdybyśmy jednak dodali do naszej przykładowej ontologii koncept Opiekun i aksjomat Opiekun ≡ Matka ⊓ Ojciec, to Opiekun byłby konceptem niespełnialnym.

Rozłączność (*disjointness*): Koncepty C i D są rozłączne, jeśli zbiory ich wystąpień są zawsze rozłączne ($C \sqcap D \equiv \bot$). W naszym przykładzie pary: Kobieta i Mężczyzna oraz Matka i Ojciec, to pary konceptów rozłącznych.

Równoważność (*equivalence*): Koncepty C i D są równoważne ($C \equiv D$), jeśli zbiory ich wystąpień są zawsze równe.

Powyższe problemy dotyczą składowej terminologicznej ontologii (TBox). Nie są one od siebie niezależne. Na przykład od razu widać, że sprawdzenie rozłączności dwóch konceptów jest równoważne sprawdzeniu spełnialności ich przecięcia. Nietrudno pokazać, że wszystkie cztery problemy można sprowadzić do problemu zawierania.

Uwzględnianie składowej asercjonalnej (ABox) prowadzi do kolejnych podstawowych problemów wnioskowania. W oznaczeniach symbolicznych tych problemów zastosujemy notację funkcyjną interfejsu DIG [3].

Wystąpienia konceptu (*retrieval*): Problem ten polega na podaniu wszystkich osobników, którzy są wystąpieniami danego konceptu (oznaczenie: *instances* (*C*)). Na przykład odpowiedzią na zapytanie *instances* (Matka) jest jednoelementowy zbiór osobników {Anna}.

Przynależność do konceptu (*instance check*): Problem ten polega na sprawdzeniu, czy dany osobnik jest wystąpieniem danego konceptu (oznaczenie: *instance*(*x*, *C*)). Na przykład odpowiedzią na zapytanie *instance* (Anna, Matka) jest *true*.

Spójność bazy wiedzy (*consistency*): Ontologia (lub baza wiedzy) jest spójna (oznaczenie: *consistent*(*K*)), jeśli wszystkie koncepty z części terminologicznej są spełnialne, a część asercjonalna nie zawiera osobników „fałszywych" (tj. takich, którzy nie są wystąpieniem żadnego konceptu). Nasza przykładowa ontologia jest spójna, ale gdybyśmy dodali do niej następujące dwie asercje: Kobieta (Maria), Mężczyzna (Maria), to odpowiedzią na zapytanie *consistent* (*K*) byłoby *false*.

Rozszerzmy naszą ontologię o rolę maSyna. Jasne jest, że jeśli ktoś ma syna, to ma dziecko. W terminach języka $\mathcal{ALC}$ zapiszemy to jako następujący aksjomat:

$$\text{maSyna} \sqsubseteq \text{maDziecko}.$$

Ale jak wyrazić to, że syn zawsze jest Mężczyzną? Innymi słowy, jak określić **zakres** roli maSyna? Można to zrobić, wprowadzając do terminologii kolejny aksjomat:

$$\exists \text{maSyna}.\neg\text{Mężczyzna} \equiv \bot.$$

Ten aksjomat można zinterpretować następująco: jeśli jakiś osobnik ma syna, który nie jest Mężczyzną, to ten osobnik jest wystąpieniem konceptu pustego, czyli jest osobnikiem fałszywym (a zatem nie istnieje). Podobnie możemy określić, że tylko Rodzice mogą mieć synów (czyli określić **dziedzinę** roli). Odpowiedni aksjomat wygląda następująco:

$$\exists \text{maSyna}.\top \sqsubseteq \text{Rodzic},$$

co można odczytać jako stwierdzenie, że jeśli ktoś ma syna, to jest wystąpieniem konceptu Rodzic (ale nie wszyscy rodzice mają synów, stąd operator $\sqsubseteq$ w tym aksjomacie). Zauważmy, że to ograniczenie na dziedzinę roli maSyna nie wynika z istniejącego już w naszej ontologii aksjomatu $\text{Rodzic} \equiv \text{Osoba} \sqcap \exists \text{maDziecko}.\text{Osoba}$, gdyż aksjomat ten nie wyklucza, że rola maDziecko może wiązać z osobnikami konceptu Osoba osobniki innych konceptów niż Rodzic.

Do tak rozszerzonej ontologii dodajmy następującą asercję:

maSyna(Karol, Jan)

i wystosujmy do naszej bazy wiedzy następujące dwa zapytania: types(Jan) i types(Karol). Zapytanie types(x) żąda podania wszystkich konceptów z terminologii, których wystąpieniem jest x. Ponieważ Karol nie jest osobnikiem fałszywym, to Jan jest Mężczyzną (tak zdefiniowaliśmy zakres roli maSyna), a zatem także Osobą. Stąd odpowiedzią na pierwsze zapytanie jest {Mężczyzna, Osoba}. Natomiast odpowiedzią na drugie zapytanie jest {Rodzic, Mężczyzna, Osoba}.

Zadaniem tego artykułu nie jest wykład z logiki opisowej, zatem nie będziemy dalej wnikać w szczegóły procesu wnioskowania. Powiemy tylko, że w ogólności wnioskowanie z terminologii DL ma wykładniczą złożoność obliczeniową w funkcji liczby konceptów. Jednak w praktyce takie „nieprzyjazne" ontologie nie zdarzają się i istnieją efektywne systemy, wnioskujące z ontologii DL, takie jak Racer [11], FaCT [12], Jena [20] (choć ten wnioskuje z ograniczonej postaci ontologii – z tzw. trójek RDF), a także KaSeA [9] – system realizowany na Politechnice Gdańskiej [9].

2.4. Ontologie a reguły

Metody regułowe i metody ontologiczne to dwie różne metody reprezentacji wiedzy. Żadna z nich nie zawiera drugiej. Dla ilustracji rozpatrzmy następujący przykład. Zdefiniujmy koncept StudentMiejscowy jako obejmujący takich studentów, którzy studiują w tym mieście, w którym mieszkają. Definicję tę w postaci reguły (klauzuli Horna) możemy zapisać następująco:

$$\text{studiuje}(x, y), \text{mieszka}(x, z), \text{mieści\_się\_w}(y, z) \rightarrow \text{StudentMiejscowy}(x).$$

Nie można natomiast w taki generyczny sposób zdefiniować tego konceptu w języku $\mathcal{ALC}$. Z drugiej strony, rozważmy typowo encyklopedyczną definicję osoby jako kobiety albo mężczyzny. Ontologicznie zapiszemy to bardzo prosto:

$$\text{Osoba} \equiv \text{Kobieta} \sqcup \text{Mężczyzna}, \text{Kobieta} \sqcap \text{Mężczyzna} \equiv \bot,$$

natomiast nie jesteśmy w stanie zapisać tego w formie klauzul Horna.

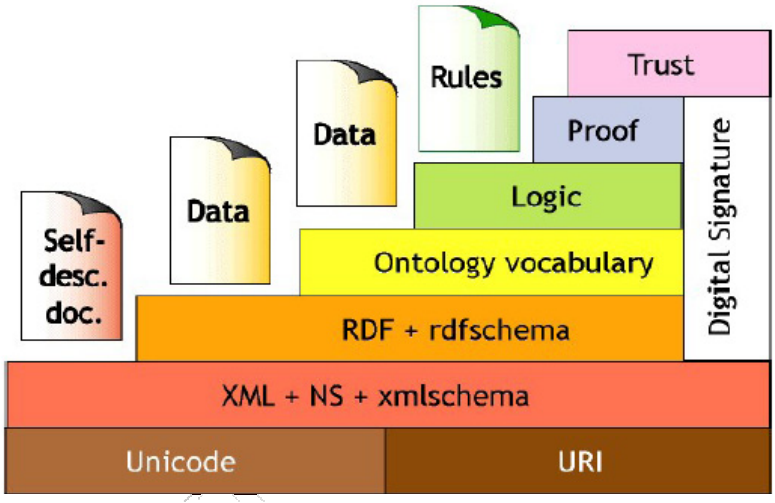
Widzimy więc, że w istocie te dwie metody uzupełniają się nawzajem. Stąd też w systemach zarządzania wiedzą stosowane są obie: reguły do zapisu wiedzy generycznej, w której jedno pojęcie wynika z poprzednich, a ontologie do zapisu wiedzy są o charakterze aksjomatycznym. Jednak wiele stwierdzeń można sformułować na oba sposoby, jak na przykład to, że jeśli ktoś ma dziecko, to jest rodzicem: $\text{maDziecko}(x) \rightarrow \text{Rodzic}(x)$ (reguła) i $\exists \text{maDziecko}.\top \sqsubseteq \text{Rodzic}$ (aksjomat).

3. Sieć Semantyczna

Czym jest Sieć Semantyczna (*Semantic Web*)? (Nie mylić z sieciami semantycznymi pisanymi małymi literami, czyli z *semantic networks*). Zgodnie z klasyczną już dziś definicją [4]: „Sieć Semantyczna jest to rozszerzenie istniejącej sieci WWW o mechanizmy semantyczne, tak aby informacje dostępne w tej sieci były dobrze zdefiniowane i umożliwiały lepszą współpracę komputerom i ludziom".

Idea budowy Sieci Semantycznej została podjęta jako odpowiedź na gwałtowane rozrastanie się Internetu, a w szczególności tych zasobów informacyjnych, które nazywamy *Word Wide Web* (WWW). Bez możliwości uczynienia tych zasobów możliwymi do przetwarzania przez komputer (*machine-readable*) będziemy skazani na mozolniejsze poszukiwania coraz szerszej wiedzy, w bardziej nieuporządkowanych zakamarkach globalnej sieci.

Logiczną konstrukcję Sieci Semantycznej bardzo dobrze ilustruje, tzw. „torcik" Sieci Semantycznej (rys. 3). Dolną warstwę Sieci, jej podstawę, stanowi system jednoznacznego adresowania zasobów Sieci identyfikatorami URI (*Uniform Resource Identifier*). Warto zaznaczyć, że nie wystarcza posługiwanie się adresami typu URL, gdyż pod jednym adresem URL może znajdować się wiele zasobów. Drugim elementem podstawy jest powszechny alfabet Unicode, pozwalający na binarne kodowanie znaków dowolnego alfabetu.



Rys. 3. Torcik Sieci Semantycznej [www.semanticweb.org]
Fig. 3. Semantic Web Cake [www.semanticweb.org]

Drugą warstwę torcika stanowi baza syntaktyczna, jaką jest język XML (*eXtensible Markup Language*). Na tym języku oparto składnię pozostałych warstw, w szczególności warstw semantycznych Sieci. Warstwa XML umożliwia definiowanie typów dokumentów (*XML Schema*), co stwarza możliwości lepszego strukturalizowania, a co za tym idzie – uporządkowania zasobów Sieci. Istotnym elementem tej warstwy jest też możliwość definiowania przestrzeni nazw (*name spaces – NS*), co pozwala na unikanie konfliktów w sytuacji, gdy

w różnych miejscach Sieci, pod tymi samymi nazwami rozumie się różne pojęcia. Warstwa XML stwarza też możliwości wyrażania znaczenia treści dokumentów, poprzez stosowanie semantycznych (a nie, jak w dokumentach HTML, czysto edytorskich) znaczników w dokumentach Sieci i poprzez strukturalizowanie tych dokumentów w formy drzewiaste, o zagnieżdżających się elementach. Stąd też dokumenty XML nazywa się czasami dokumentami samoopisującymi się (*self-describing documents*).

```
<rdf:resource    rdf:about="http://pg#JK"
                 pg:nazwisko="Kowalski">
                 <pg:stronaDomowa rdf:resource="http://www.pg/page.html"/>
</rdf:resource>
```

Opisywany jest tu – w postaci dwóch trójek RDF – zasób o identyfikatorze „http://pg#-JK". Ma on właściwość „nazwisko" o wartości literalnej „Kowalski" oraz właściwość „stronaDomowa", która sama jest zasobem o nazwie „http://www.pg/page.html" (i oczywiście też może mieć swoje właściwości). Właściwości „nazwisko" i „stronaDomowa" pochodzą z pewnej przestrzeni nazw „pg" (i tam też pewnie opisane jest ich znaczenie), natomiast właściwości „resource" i „about" należą do przestrzeni nazw „rdf", w której zdefiniowano syntaktykę i semantyką RDF. (Czytelnika, chcącego zagłębić się w zawiłości składni i semantyki RDF odsyłamy do [26]).

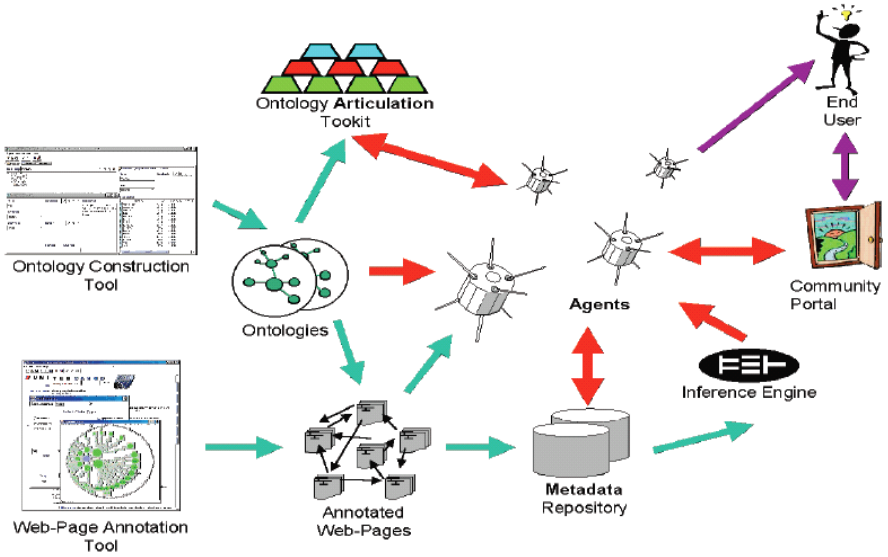
Ważne jest to, że RDF umożliwia definiowanie typów zasobów (służy do tego język RDF Schema). Można na przykład zdefiniować typ zasobu „Osoba", który ma właściwości: „nazwisko", „adres" itd., określonego typu. Umożliwia to budowanie prostych ontologii, w których występują koncepty – typy – i ich wystąpienia – zasoby. Popularnym narzędziem do wydobywania informacji (wnioskowania) z takich trójek jest, wspomniany już wcześniej, pakiet Jena.

Na kolejnym poziomie „torcika" znajdujemy ontologie bardziej ekspresywne niż zbiory trójek RDF, w szczególności ontologie zapisane w języku OWL-DL. Język ten aktualnie stał się *de facto* standardem zapisu ontologii w Sieci Semantycznej. Jest rozwijany w ramach inicjatywy *Semantic Web* przez konsorcjum W3C (www.w3c.org). Sama składnia języka OWL jest oparta na trójkach RDF, co czyni ją dość zawiłą, jednak trwają prace nad nowym standardem, który – jak można mieć nadzieję – składnię tę uporządkuje. Semantyka języka OWL-DL oparta jest na logice opisowej, a ściślej na dialekcie o nazwie $\mathcal{SHIOQ}(\mathcal{D})$ który oprócz konstruktorów języka $\mathcal{ALC}$ zawiera takie konstrukcje, jak ograniczenia liczebnościowe, dziedziny konkretne, role przechodnie, odwrotne i symetryczne, i inne. Pełny opis OWL-DL można znaleźć w jego specyfikacji [16].

Kolejne warstwy Sieci Semantycznej – logika (*Logic*) i dowody (*Proof*) – obejmują ontologie wyrażone za pomocą mechanizmów ekspresji, dostarczanych przez warstwy niższe (logika) oraz moduły wnioskujące (dowody). W warstwie dowodów ontologie rozszerzane są o reguły dedukcyjne i proceduralne, zapisane na przykład w języku RuleML lub podobnym.

Nad wszystkimi warstwami Sieci Semantycznej „czuwa" warstwa zaufania (*Trust*). Jest to niezwykle ważna warstwa, gdyż użytkownicy Sieci Semantycznej powinni ufać składającym się na nią źródłom wiedzy, a przynajmniej zdawać sobie sprawę ze stopnia zaufania, jakim mogą obdarzyć dane źródło wiedzy i – tym samym – ontologie prezentowane w Sieci przez to źródło. Jest kilka środków zwiększania takiego zaufania. Jednym z nich jest powszechne stosowanie podpisu cyfrowego i związanej z nim odpowiedniej infrastruktury bezpieczeństwa. Istotne jest bowiem, by na etapie wymiany informacji pomiędzy źródłami wiedzy oraz podczas przekazywania informacji ze źródła wiedzy do użytkownika informacja nie ulegała przekłamaniu, celowemu sfałszowaniu i innym zakłóceniom. Drugim środkiem jest ocenianie wiarygodności źródeł wiedzy przez niezależne instytucje rankingowe typu *non-profit* i udostępnianie tych ocen potencjalnym użytkownikom tych źródeł. Jest to szczególnie ważne w wypadku informacji krytycznych, potencjalnie szkodliwych dla użytkowników, takich jak informacje medyczne. Przykładem takiej instytucji rankingowej, właśnie z dziedziny medycyny, jest Health On the Net (www.hon.ch).

Jak wspomnieliśmy już w podrozdziale 2.1, ważną rolę w Sieci Semantycznej odgrywają agenty, rozumiane jako autonomicznie działające programy komputerowe, które potrafią samodzielnie – komunikując się w razie potrzeby z innymi agentami – rozwiązać zadanie postawione im przez człowieka. O ile ontologie stanowią statyczną część Sieci Semantycznej, to jej dynamiczną składową wyznaczają właśnie agenty. Dobrze ilustruje to tzw. „łańcuch pokarmowy" Sieci Semantycznej (rys. 4).



Rys. 4. „Łańcuch pokarmowy" Semantic Web [www.semanticweb.org]
Fig. 4. Semantic Web Food Chain [www.semanticweb.org]

Każdy użytkownik Sieci Semantycznej dysponuje zbiorem agentów (*agents*), wyspecjalizowanych w określonych dziedzinach. Przykładowo: jeden agent może zajmować się spra-

wami żywieniowymi, poszukując informacji o restauracjach, oferujących menu odpowiadające gustom „właściciela" tego agenta, inny może poszukiwać w sieci interesujących wydarzeń kulturalnych, inny jeszcze może nadzorować kwestie zdrowotne użytkownika, komunikując się w tym celu z odpowiednią usługą WWW (*Web service*) itd. Wyobraźnia może podsuwać kolejne dziedziny i zastosowania. Aby sprostać tym zadaniom, agenty nie tylko poszukują w Sieci istotnych dokumentów, ale także korzystają z aktywnych usług WWW, opisanych stosownymi ontologiami (zapisanymi w dialekcie OWL-DL o nazwie OWL-S), a także z usług motorów wnioskujących, aby za ich pomocą uzyskać dodatkowe informacje, bezpośrednio niedostępne. Uzyskaną w ten sposób wiedzę przekazują swojemu „właścicielowi" do „konsumpcji".

Jak widzimy to na rys. 4., istotnymi elementami łańcucha pokarmowego – dostarczycielami pokarmu – są elementy związane z ontologiami: edytory ontologii (*ontology construction tools*), usługi do prezentowania ontologii i ich fragmentów (*ontology articulation toolkits*), narzędzia do znacznikowania stron WWW według zadanych ontologii (*Web-page annotation tools*) i efekty ich działań (*annotated Web pages*), a także różnorodne repozytoria metadanych, jak leksykony, tezaurusy i in. Stąd też jednym z zasadniczych problemów dla realizacji Sieci Semantycznej jest inżynieria ontologii, będąca przedmiotem następnego rozdziału.

4. Inżynieria ontologii w Sieci Semantycznej

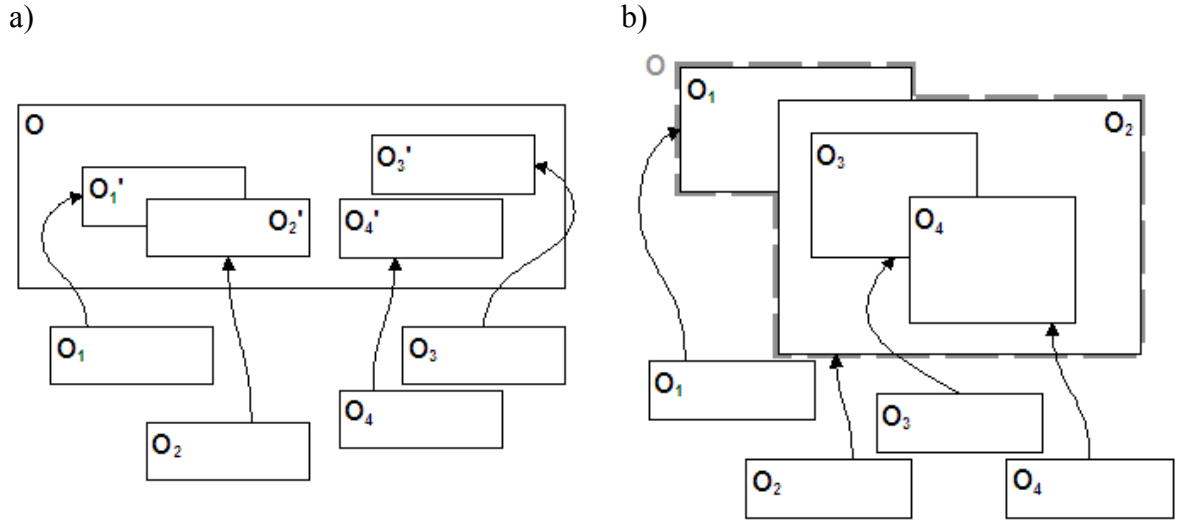
Rozwój Sieci Semantycznej to również rozwój ontologii, gdyż właśnie te stoją u podstaw całego procesu wymiany wiedzy i jej dalszego wykorzystania. Bez ontologii nie ma Sieci Semantycznej, i to ontologii spełniających pewne określone wymagania. Możemy posłużyć się analogią do oprogramowania. Wytworzenie danego oprogramowania ma sens tylko wtedy, gdy to oprogramowanie spełnia pewne kryteria. W wypadku ontologii sytuacja jest analogiczna. Zapożyczając terminologię z inżynierii oprogramowania, mówimy o **poprawności** ontologii. Ontologia jest poprawna, jeśli poprawnie realizuje stawiane jej wymagania. Zapewnieniu poprawności ontologii służy proces weryfikacji ontologii (*ontology verification*) [21]. Drugi warunek, jaki musi spełniać ontologia, dotyczy znaczenia ontologii. Model formalny świata zapisany, w ontologii musi odpowiadać rzeczywistej dziedzinie, dla której dana ontologia została stworzona. Do tego procesu odnosi się termin **walidacja** ontologii (*ontology validation*) [21].

Zasadnicze pytanie inżynierii ontologii sprowadza się do tego, w jaki sposób tworzyć ontologie, aby były poprawne i adekwatne do modelowanej rzeczywistości. Przede wszystkim

należy zidentyfikować, na jakiej drodze możemy uzyskać potrzebną ontologię. Możliwych jest kilka sposobów [17]:

- **integrowanie ontologii** (*ontology integration*) poprzez tworzenie nowej ontologii z wykorzystaniem ontologii już istniejących,
- **łączenie ontologii** (*ontology merging*) w jedną ontologię ujednolicającą wszystkie łączone ontologie,
- **budowanie ontologii** (*ontology building*) na nowo.

Należy jednak zauważyć, że rozwój Sieci Semantycznej powoduje zarówno powstawanie nowych ontologii, opisujących różne (często te same) dziedziny nauki, jak i równoległy dynamiczny rozwój istniejących ontologii i dostosowywanie ich do specyficznych potrzeb i nowych aplikacji. Dlatego też budując ontologię, należy zawsze kierować się zasadą, że jeśli tylko jest to możliwe, należy wykorzystywać istniejące ontologie. Wykorzystanie istniejących ontologii występuje w dwóch ważnych procesach: integrowania ontologii i łączenia ontologii. Rysunek 5, omówiony szczegółowo poniżej, obrazuje różnice pomiędzy tymi dwoma sposobami tworzenia ontologii.



Rys. 5. Integrowanie (a) a łączenie (b) ontologii
Fig. 5. Ontology integration (a) and ontology merging (b)

4.1. Integrowanie ontologii

Proces integrowania ontologii polega na tworzeniu nowej ontologii z ontologii istniejących (ontologii integrowanych), w taki sposób, że:

- ontologie integrowane są dostosowywane do konkretnych potrzeb, a następnie
- ontologie te są wzbogacane o nowe terminy (koncepty, role, atrybuty) i zależności między nimi.

Ontologie integrowane mogą być użyte bezpośrednio, jeśli są odpowiednie dla danego zastosowania. Dostosowanie zaś ontologii integrowanych do indywidualnych potrzeb może albo **specjalizować** koncepty, role lub atrybuty, albo je **wzbogacać**.

O specjalizacji mówimy, jeśli terminy pojawiające się w ontologii są zbyt ogólne; np.: potrzebujemy wiedzy na temat pieczywa (koncept Pieczywo), a w ontologii integrowanej zamodelowano wiedzę na temat produktów zbożowych (koncept ProduktZbożowy). W ten sposób, w odniesieniu do konceptu ProduktZbożowy, mówimy o jego specjalizacji. Poprzez wzbogacanie rozumiemy sytuację odwrotną: w ontologii integrowanej mamy zdefiniowany koncept Pieczywo, a potrzebujemy konceptu ProduktZbożowy. W ten sposób wzbogacamy koncept Pieczywo. W procesie transformacji ontologii pewne terminy mogą ulegać specjalizacji, a pewne wzbogacaniu. A zatem, w rzeczywistości mówimy nie o specjalizacji lub wzbogacaniu ontologii, a o specjalizacji lub wzbogacaniu terminów, pojawiających się w ontologii. W wyjątkowych sytuacjach, kiedy część terminów ulega specjalizacji, a pozostałe terminy pozostają bez zmian, możemy mówić o specjalizacji ontologii. Analogiczna sytuacja zachodzi dla wzbogacania ontologii.

W drugim kroku procesu integracji w ontologii zintegrowanej (wynikowej) definiowane są dodatkowe role, atrybuty i koncepty, potrzebne w konkretnym zastosowaniu. Przykładowo: załóżmy, że jedna z ontologii integrowanych zawiera koncept Pieczywo, a inna koncept Lek. Ontologia wynikowa ma opisywać interakcje pomiędzy lekami a pieczywem. W takiej sytuacji w ontologii wynikowej dodatkowo potrzebna jest rola maInterakcjęZ, której dziedziną jest koncept Pieczywo, a zakresem koncept Lek. Dodatkowo, należy określić, że maInterakcjęZ jest rolą symetryczną.

Na rysunku 5 a proces dostosowywania ontologii integrowanych do konkretnych potrzeb został oznaczony strzałkami, symbolizującymi transformację ontologii O_1, O_2, O_3 i O_4 do ontologii O_1', O_2', O_3' i O_4', a ontologią wynikową jest O. Ontologie O_1', O_2', O_3' i O_4' mogą opisywać różne lub te same dziedziny. Dziedziny ontologii integrowanych mogą być rozłączne (np. ontologia O_3' i O_4') lub się pokrywać – w pełni lub częściowo (np. ontologia O_1' i O_2'). Na naszym przykładowym rysunku dziedzina ontologii wynikowej O również nie jest sumą dziedzin ontologii integrowanych. Dzieje się tak w konsekwencji definiowania dodatkowych terminów spoza dziedzin ontologii integrowanych. Nie zawsze jednak taka sytuacja musi zajść.

4.2. Łączenie ontologii

Łączenie ontologii jest w rzeczywistości szczególnym przypadkiem integrowania ontologii. Łączone ontologie nie są dostosowywane do konkretnych potrzeb, ontologia wynikowa nie definiuje żadnych nowych terminów. W rzeczywistości łączenie ontologii jest procesem znajdowania wspólnych terminów pomiędzy różnymi ontologiami i wywiedzenie z nich no-

wej ontologii wynikowej, która umożliwia współpracę systemów komputerowych, opartych na ontologiach łączonych.

Proces ten zilustrowano na rys. 5 b. Ontologie łączone to ontologie O_1, O_2, O_3 i O_4. Zauważmy, że ontologie łączone w tym procesie nie ulegają zmianie. Strzałki na rysunku 5 b, w przeciwieństwie do rysunku 5 a, oznaczają tylko to, że ontologie łączone stają się częścią ontologii wynikowej O, oznaczonej linią przerywaną. Stąd na rysunku 1b ontologia wynikowa składa się właśnie z ontologii O_1, O_2, O_3 i O_4, a nie z ontologii O_1', O_2', O_3' i O_4'. Analogicznie jak w wypadku integrowania ontologii, dziedziny ontologii łączonych mogą pozostawać ze sobą w różnych zależnościach. I tak w naszym przykładzie dziedzina ontologii O_4 zawiera się w dziedzinie ontologii O_2.

Ontologia wynikowa „rozumie" wszystkie terminy, zdefiniowane w ontologiach łączonych oraz dodatkowo „wie", jakie występują zależności między tymi terminami. Posłużmy się przykładem. Załóżmy, że łączone są następujące dwie ontologie A i B:

A: Kobieta $\sqcap$ Mężczyzna $\equiv \bot$
Kobieta $\sqcup$ Mężczyzna $\equiv \top$

B: Człowiek $\equiv \top$
$\exists$maDziecko.$\top \sqsubseteq$ Człowiek

Ontologia wynikowa powstała z połączenia obu ontologii wygląda następująco:

Kobieta $\sqcap$ Mężczyzna $\equiv \bot$
Kobieta $\sqcup$ Mężczyzna $\equiv \top$
Człowiek $\equiv \top$
$\exists$maDziecko.$\top \sqsubseteq$ Człowiek

Zauważmy, że semantyka ontologii wynikowej nie jest – jak mogłoby się wydawać na pierwszy rzut oka – prostą sumą semantyki ontologii połączonych. Ontologia wynikowa poza tym, że zawiera wszystkie terminy z obu ontologii, dodatkowo wie również, że Kobieta $\sqcup$ Mężczyzna $\equiv$ Człowiek, co nie było jawnie wyspecyfikowane w żadnej z ontologii składowych.

Ontologia powstała w wyniku połączenia ontologii może zastąpić ontologie łączone lub być warstwą pośredniczącą pomiędzy systemami opartymi na ontologiach łączonych. Jeśli nowa ontologia pełni rolę warstwy pośredniczącej, mówimy o **odwzorowywaniu ontologii** (*ontology mapping*) [21]. W przypadku odwzorowywania ontologii warstwa pośrednicząca tłumaczy terminy z jednej ontologii na terminy z drugiej ontologii. W opisywanym przykładzie warstwa pośrednicząca musiałaby wiedzieć, że dziedziny obu ontologii są takie same, a więc musiałaby wiedzieć, że Kobieta $\sqcup$ Mężczyzna $\equiv$ Człowiek.

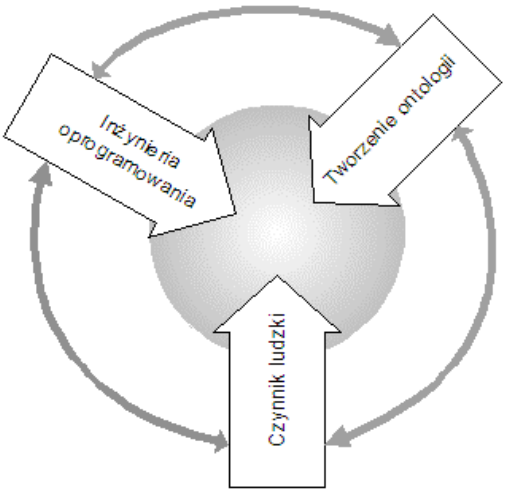
4.3. Budowanie ontologii

Budowanie ontologii na nowo jest procesem, który powinien zostać podjęty tylko po szczegółowej analizie istniejących ontologii i możliwości ich wykorzystania w konkretnym zastosowaniu. Po zdefiniowaniu celu budowania nowej ontologii i analizie istniejących ontologii należy zidentyfikować koncepty, relacje pomiędzy konceptami, atrybuty i role oraz ostatecznie zapisać ontologię w formalnym języku, np. OWL. Proces ten został szczegółowo opisany w metodologii *Enterprise Ontology* [24] i stał się podstawą różnych metod budowania ontologii. Formalny opis modelowanego świata zazwyczaj powstaje z półformalnego opisu rzeczywistości. Na podstawie opisu półformalnego inżynierowie wiedzy formułują taksonomię konceptów. Taka taksonomia wzbogacana jest o dodatkowe zależności między konceptami, inne niż relacja zawierania, oraz odpowiednie role i atrybuty, będące binarnymi relacjami między konceptami oraz pomiędzy konceptami a dziedzinami konkretnymi. Tak zbudowana ontologia staje się podstawą dyskusji z ekspertami dziedzinowymi. Często pomocna staje się wizualizacja ontologii. Eksperci dziedzinowi wprowadzają uwagi do istniejącej ontologii. Uwagi takie powinny zawsze być ściśle udokumentowane. Od tego momentu proces zazwyczaj się powtarza, aż do ustalenia pewnej stabilnej wersji ontologii. Warto zauważyć, że również w przypadku integrowania ontologii udział ekspertów dziedzinowych bywa niezbędny.

4.4. Tworzenie ontologii w procesie wytwarzania oprogramowania

W punktach 4.1 – 4.3 zostały zarysowane zagadnienia, związane z tworzeniem ontologii. W rzeczywistych warunkach proces tworzenia ontologii jest zazwyczaj częścią procesu wytwarzania oprogramowania, szczególnie w systemach opartych na wiedzy, w których ontologie stanowią jedynie jeden z elementów systemu. Każda faza procesu wytwarzania oprogramowania ma swój wpływ na proces tworzenia ontologii. I tak, w fazach początkowych istotne jest ustalenie roli systemu zarządzania wiedzą w całym systemie informatycznym. W fazie specyfikowania wymagań niezbędne jest zebranie wymagań dotyczących wiedzy, jaką system informatyczny ma wykorzystywać. Zazwyczaj właśnie ze specyfikacji wymagań tworzony jest półformalny opis modelowanego świata. W fazach analizy i projektowania powstaje już jego opis formalny.

Również przyjęty cykl wytwarzania oprogramowania ma wpływ na cykl wytwarzania ontologii. Przykładowo, jeśli jest tworzony prototyp systemu, to jest on zazwyczaj budowany z użyciem ontologii prototypowej. Jeśli oprogramowanie jest wytwarzane w cyklu spiralnym, to i w takim cyklu rozwijana jest ontologia.



Rys. 6. Proces tworzenia ontologii w procesie wytwarzania oprogramowania [21]
Fig. 6. Ontology development process in the process of software development [21]

Poza rodzajem wybranego cyklu wytwarzania oprogramowania, na proces tworzenia ontologii ma wpływ również czynnik ludzki. Często podkreśla się, że sukces systemu zarządzania wiedzą silnie zależy od akceptacji ludzi, korzystających z tego systemu. Stąd też niezwykle ważny jest, wspomniany na początku tego rozdziału, proces walidacji ontologii, przeprowadzany z udziałem ekspertów dziedzinowych. Wpływ poszczególnych czynników na proces tworzenia ontologii został zilustrowany na rys. 6.

4.5. Gotowe ontologie

Jak zaznaczyliśmy wcześniej, szalenie ważną rolę w inżynierii ontologii odgrywa wykorzystanie gotowych ontologii. W Internecie dostępnych jest wiele ontologii z różnych dziedzin. Istnieją tzw. biblioteki ontologii, mniej lub bardziej sformalizowane. Obszerny wykaz dostępnych ontologii można znaleźć pod adresem http://www.daml.org/ontologies. Istnieje również biblioteka KACTUS podstawowych ontologii technicznych [13]. Na szczególną uwagę zasługują również szeroko rozpowszechnione ontologie z dziedziny medycyny, takie jak ontologia GALEN [8] i ontologia UMLS [23]. Wspomnijmy również kilka innych ontologii dziedzinowych, takich jak ontologia wspierająca planowanie, stworzona w ramach inicjatywy Rome Laboratory Planning Initiative [1], ontologia z dziedziny fizyki – PhySys [5] czy ontologia z dziedziny matematyki EngMath [10].

Skoro mówimy o ontologiach dziedzinowych, modelujących pewną dziedzinę wiedzy, należy również wspomnieć o dwóch trochę innych rodzajach ontologii: wysokiego poziomu (*upper ontologies*) i leksykalnych (*lexicons ontologies*). Poprzez ontologię wysokiego poziomu rozumiemy taką ontologię, która modeluje pewne ogólne pojęcia i zależności między nimi. W ten sposób tworzy pewne ramy dla innych ontologii, na przykład dziedzinowych. Posłużmy się przykładem. Załóżmy, że mamy pewną ontologię wysokiego poziomu, opisującą

proces podejmowania decyzji. W ramach tej ontologii są zdefiniowane takie koncepty jak Fakt, Przesłanka, Decyzja oraz pewne role. Już w konkretnych ontologiach dziedzinowych tworzone są nowe koncepty, dziedziczące od konceptów zdefiniowanych w ontologii wysokiego poziomu. I tak, w procesie wspomagania decyzji odnośnie wyboru odpowiedniej terapii dla pacjenta mogą być zdefiniowane fakty medyczne, za pomocą konceptu FaktMedyczny oraz fakty pozamedyczne, za pomocą konceptu FaktPozamedyczny. Oba te koncepty muszą dziedziczyć od konceptu Fakt. Natomiast w procesie podejmowania decyzji odnośnie wyboru odpowiedniego samochodu rodzinnego mogą istnieć fakty, opisujące potrzeby rodziny (koncept FaktPotrzebyRodziny) oraz fakty, opisujące finansowe możliwości rodziny (koncept FaktMożliwościRodziny). Również w tej ontologii oba te koncepty dziedziczą od konceptu Fakt. Uzyskujemy w ten sposób pewną jednolitą budowę ontologii dziedzinowej na bazie ontologii wysokiego poziomu.

Ostatni rodzaj gotowych ontologii to leksykony [21]. Leksykon jest zbiorem słów w danym języku wraz z pewną wiedzą o tych słowach. Leksykony mogą być ogólne lub mogą dotyczyć specyficznej dziedziny. Słowa są powiązane między sobą różnymi zależnościami, zarówno semantycznymi, jak i wynikającymi z reguł gramatycznych. Przykłady zależności semantycznych między słowami to:

- relacja symetryczna „jest synonimem”, np. ładny i śliczny;
- relacja „jest rodzajem”, np. hałas i dźwięk: hałas jest rodzajem dźwięku;
- relacja „jest częścią”, np. koło i rower: koło jest częścią roweru;
- relacja „jest przeciwieństwem do”, np. ciepły jest przeciwieństwem do zimny.

Należy zaznaczyć, że nie można traktować leksykonów jako ontologii dziedzinowych. Często zależności między słowami nie przekładają się bezpośrednio na zależności między konceptami w danej dziedzinie. Leksykony są jednak bardzo często używane przy tworzeniu ontologii, zwłaszcza ich łączeniu i integrowaniu. Pomagają one w pewnym stopniu w ocenianiu podobieństwa konceptów na podstawie znaczenia nazw tych konceptów. Do najbardziej znanych leksykonów należy darmowy leksykon języka angielskiego WordNet [25]. Inne ważne źródła, z których można pozyskać leksykony, to serwis ELDA [7], dystrybuujący liczne leksykony różnych europejskich języków, zarówno ogólnego przeznaczenia, jak i ze specyficznych dziedzin, oraz serwis LDC [14], dystrybuujący leksykon CELEX.

5. A może jednak inaczej?

Idea Sieci Semantycznej nie jest jedynym przedsięwzięciem, ukierunkowanym na automatyzację pozyskiwania wiedzy z zasobów WWW. Są też inne, bazujące na podstawowym – i najbardziej naturalnym dla człowieka – sposobie reprezentacji wiedzy ludzkiej, jakim jest

język naturalny. Zastanówmy się, w jaki sposób obecnie najczęściej poszukujemy w sieci WWW interesujących nas informacji. Zazwyczaj czynimy to za pomocą którejś z popularnych wyszukiwarek, np. google.com. Do odpowiedniego pola wpisujemy słowa, które – jak nam się zdaje – powinny znaleźć się w tekście strony HTML. Wyniki takiego wyszukiwania – nazwijmy je syntaktycznym – są różnej jakości. Wynika to czasem z przyczyn czysto technicznych, na przykład z tego, ze autorzy stron HTML umieszczają celowo w nagłówkach stron słowa, które wcale nie odpowiadają treści strony, w ten sposób sztucznie przyciągając internetowych szperaczy. Inne przyczyny są znacznie głębsze i wynikają z osobliwości – a może raczej ze skomplikowanej natury – języka naturalnego. Można przecież napisać spory tekst o miłości, który nie będzie zawierał ani jednego słowa „miłość" lub „kochać" (choć należy wątpić, że będzie to tekst autorstwa informatyka). Z drugiej strony, szukając tekstów o rycerskich zamkach średniowiecznych, można przy okazji sporo dowiedzieć się, jak zamykano drzwi w średniowieczu. Osobną kwestią – jakże utrudniającą wyszukiwanie – jest wielojęzyczność.

Przeanalizujmy nieco bardziej szczegółowo wyszukiwanie syntaktyczne. Załóżmy, że poszukujemy zbioru adresów URL (*Uniform Resource Locator*) miejsc w sieci WWW, zawierających interesujące nas informacje (takie adresy nazwiemy istotnymi). W tym celu formułujemy pewne zapytanie i staramy się – posługując się metodami syntaktycznymi – uzyskać ten zbiór. Przyjmijmy, że faktyczna liczba istotnych adresów WWW wynosi x. W ogólnym przypadku nasze zapytanie zwróci nam y adresów, z których tylko t będzie adresów istotnych, zaś y–t nieistotnych. Pojęcia kompletności $\boldsymbol{R}$ (*recall*) i precyzji $\boldsymbol{P}$ (*precision*) odkrywania definiujemy następująco:

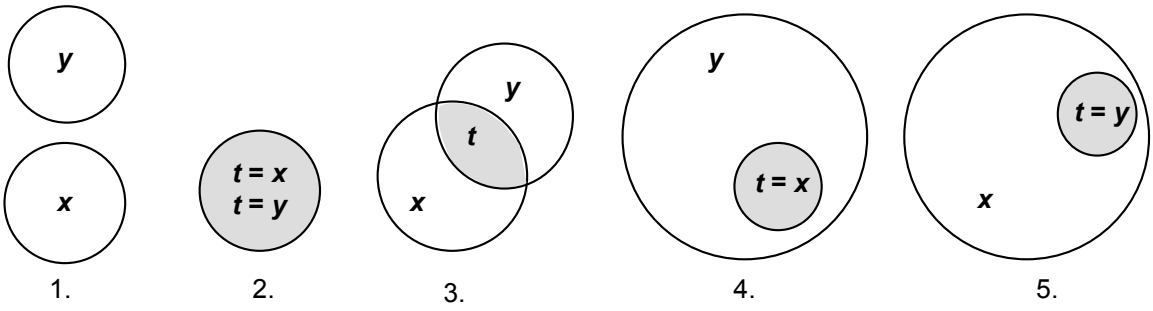
- **Kompletność** odkrywania jest to stosunek liczby **odkrytych, istotnych** adresów do liczby **faktycznie istniejących, istotnych** adresów: $\boldsymbol{R} = t / x$.
- **Precyzja** odkrywania jest to stosunek liczby **odkrytych, istotnych** adresów do liczby **wszystkich odkrytych** adresów: $P = t / y$.

Na rys. 7 zilustrowano różne sytuacje, jakie mogą wystąpić na etapie odkrywania źródeł danych. Obszar zacieniowany oznacza zbiór odkrytych istotnych adresów.

W przypadku nr 1 $t = 0$: nie odkryto żadnych istotnych adresów. Zarówno kompletność, jak i precyzja odkrywania wynoszą 0. Ponieśliśmy koszty, a nie odnieśliśmy żadnych zysków. Jedyną – dość wątpliwą – korzyścią jest przekonanie się, że zastosowane słowa kluczowe są niewłaściwe i trzeba je zmienić (lub trzeba zmienić metodę wyszukiwania).

Na drugim biegunie jest przypadek nr 2: znaleźliśmy wszystkie istotne adresy i tylko takie adresy. Osiągnęliśmy zatem idealną kompletność ($t = x$, $\boldsymbol{R} = 1$) przy idealnej precyzji ($t = y$, $\boldsymbol{P} = 1$), a więc i maksymalną efektywność wyszukiwania, rozumianą jako iloczyn kompletności i precyzji. Jasne jest, że taki przypadek – przy ogromie zasobów informacyj-

nych Internetu i ich immanentnym nieuporządkowaniu – jest wyidealizowany i niezwykle mało prawdopodobny.



Rys. 7. Różne przypadki kompletności i precyzji wyszukiwania syntaktycznego
Fig. 7. Different cases of recall and precision of syntactical search

Przypadek nr 3 jest najbardziej typowy: znaleźliśmy istotne adresy, ale nie wszystkie, jakie istnieją ($0 < t < x$), a dodatkowo znaleźliśmy adresy nieistotne ($y > t$). W tej sytuacji zarówno kompletność, jak i precyzja zawierają się w przedziale (0, 1).

W przypadku nr 4 uzyskaliśmy wszystkie istotne źródła danych ($t = x$), ale ponadto znaleźliśmy dużo adresów nieistotnych ($y >> t$). W efekcie osiągnęliśmy idealną kompletność ($\boldsymbol{R} = 1$), jednak kosztem złej precyzji ($\boldsymbol{P}$ bliskie zeru). W efekcie trzeba będzie ponieść dodatkowy nakład pracy na oddzielenie znacznej ilości „plew" od użytecznego „ziarna".

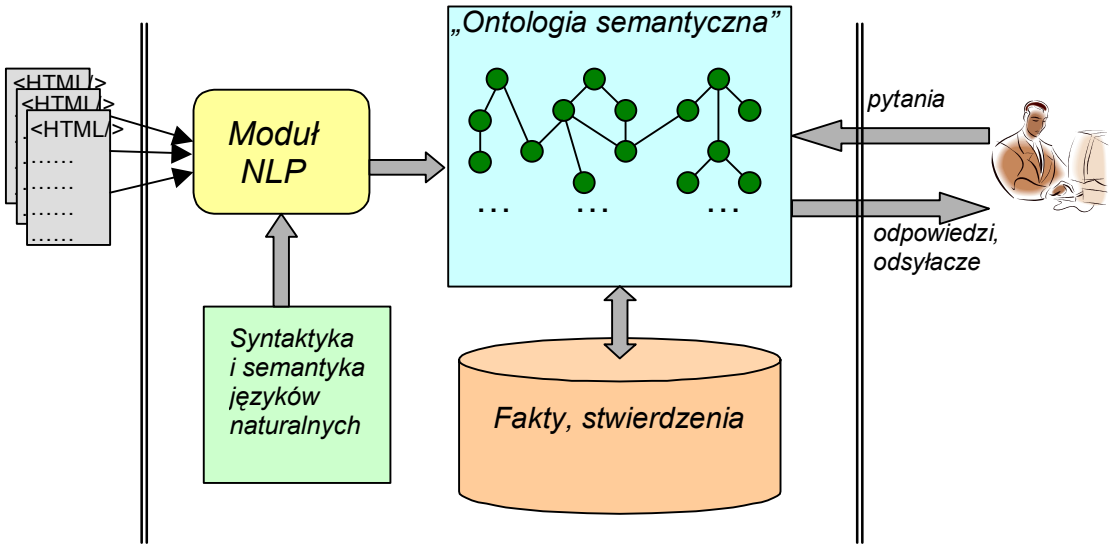
Przypadek nr 5 ilustruje sytuację odwrotną: osiągnęliśmy idealną precyzję ($t = y$, czyli $\boldsymbol{P} = 1$), jednak nie odkryliśmy wielu istotnych źródeł danych ($t << x$, $\boldsymbol{R}$ bliskie zeru). Powstaje tu problem oszacowania, jak wiele istotnych źródeł danych nam umknęło (pozostały „tajemnicą") i czy rzeczywiście są ważne dla całego procesu pozyskiwania informacji z sieci WWW.

Praktyka wskazuje, że te dwa ostatnie często się zdarzają. Ileż to razy, zadawszy zapytanie do wyszukiwarki, uzyskujemy wiele tysięcy odsyłaczy, z których na ekranie widzimy tylko kilkadziesiąt pierwszych? Powstaje problem: czy pracowicie przejrzeć kolejne odsyłacze, bojąc się ominąć istotne źródła, czy też po prostu zrezygnować? A ile razy nie uzyskaliśmy żadnego odsyłacza lub tylko kilka? Czy to znaczy, że źle sformułowaliśmy zapytanie, czy też sieć WWW naprawdę nie zawiera potrzebnych nam danych?

Odpowiedzią na tego typu problemy jest idea wyszukiwania semantycznego. Z grubsza biorąc, polega ona na tym, że wyszukiwarce zadajemy pytanie sformułowane w dowolnym języku naturalnym. W odpowiedzi wyszukiwarka nie tylko podaje nam odsyłacze do stron WWW, zawierających interesujące nas informacje, ale ponadto udziela nam odpowiedzi, również sformułowanej w języku naturalnym. Koncepcję takiego rozwiązania przedstawia rys. 8.

Zasadniczym modułem takiej wyszukiwarki jest moduł przetwarzania języka naturalnego (moduł NLP). Moduł ten czyta strony HTML (lub inne tekstowe) ogólnie dostępne w sieci

WWW, wydobywa z nich semantykę, posługując się przy tym odpowiednią bazą syntaktyczną i semantyczną języków naturalnych oraz metodami logiki rozmytej i statystyki. Uzyskaną w ten sposób wiedzę gromadzi się w, tzw. „ontologii semantycznej" (będącej w istocie „ontologią wszystkiego"), skojarzonej z tworzoną równocześnie bazą faktów i stwierdzeń nieontologicznych. Rozwiązanie to aktualnie jest rozwijane w ramach projektu Hakia. Uczestnicy tego projektu deklarują, że w ten sposób do końca 2006 roku pozyskają całą wiedzę anglojęzyczną, tkwiącą w zasobach sieci WWW i udostępnią ją za pośrednictwem swojej wyszukiwarki semantycznej. Efekty ich pracy można na bieżąco śledzić i testować pod adresem www.hakia.com.



Rys. 8. Koncepcja wyszukiwarki semantyczne
Fig. 8. The idea of a semantic search engine

Sceptycy takiego podejścia twierdzą, że metodą automatycznej analizy i przetwarzania języka naturalnego można pozyskać jedynie niewielki fragment (ok. 10-15%) semantyki, a tym samym i wiedzy, tkwiącej w tekstach, a głównym problemem jest niejednoznaczność zdań w języku naturalnym. I dlatego też uważają, że podejście polegające na tworzeniu ontologii dziedzinowych i oznaczanie tekstów pojęciami z tych ontologii jest bardziej rokujące. Jak zwykle, zostanie to zweryfikowane przez czas i praktykę.

6. Podsumowanie

Zamiast tradycyjnego podsumowania, zastanówmy się, co wiemy, a czego nie wiemy o Sieci Semantycznej i o roli ontologii w Sieci.

Co wiemy:

- Sieć Semantyczna nie ma zastąpić sieci WWW – jest i będzie rozwijana niezależnie.

- Sieć Semantyczna to nie kolejna scentralizowana baza danych – jest to raczej zbiór (powiązanych ze sobą) baz wiedzy.
- Sieć Semantyczna ma ułatwić ludziom i komputerom współużytkowanie wiedzy, zawartej w licznych, rozrzuconych po Internecie, dynamicznie zmieniających się źródłach wiedzy.

Czego nie wiemy? Nie wiemy oczywiście, w jakim stopniu idea Sieci Semantycznej zostanie urzeczywistniona, czy nabierze charakteru globalnego, czy raczej – jak sądzą sceptycy (a może realiści?) – będzie rozwijana lokalnie, w określonych dziedzinach życia. Być może kluczem do sukcesu Sieci Semantycznej będzie faktyczna możliwość ponownego wykorzystania przez inżyniera wiedzy o istniejących (stworzonych przez innych inżynierów wiedzy) komponentach, takich jak ontologii, baz wiedzy, motorów wnioskujących bądź innych elementów „łańcucha pokarmowego", do budowy nowych aplikacji, opartych na wiedzy, użytecznych dla nowoczesnego społeczeństwa.

LITERATURA

1. ARPA/Rome Laboratory Planning Initiative http://aaaipress.org/Library/ARPI/arpi96-contents.php
2. Baader F. A., McGuiness D. L., Nardi D., Patel-Schneider P. F.: The Description Logic Handbook: Theory, implementation, and applications. Cambridge University Press, 2003.
3. Bechhofer S.: The DIG Description Logic Interface: DIG/1.1. University of Manchester, 2003.
4. Berners-Lee T., Hendler J., Lassila O.: .The Semantic Web. Scientific American, May 2001.
5. Borst P.: Construction of Engineering Ontologies for Knowledge Sharing and Reuse. PhD thesis, Tweente University, 1997.
6. Dym C. L., Levitt R. E.: Knowledge-based Systems in Engineering. McGraw-Hill, Inc., 1991.
7. Evaluation and Language resources Distribution Agency http://www.elda.fr.
8. GALEN http://www.openclinical.org/prj_galen.html.
9. Goczyła K., Grabowska T., Waloszek W., Zawadzki M.: The Cartographer Algorithm for Processing and Querying Description Logics Ontologies. Lectuire Notes in Computer Science, Vol. 3528 (2005), Ed. Springer Verlag, s. 163÷169.
10. Gruber T., Olsen G. R.: An Ontology for Engineering Mathematics. W: E. S. J. Doyle & P. Torasso, eds, 'KR94 Proceedings', Morgan Kaufmann, s. 258÷269, 1994.

11. Haarslev V., Möller R.: RACER User's Guide and Reference Manual. September 17, 2003, http://www.cs.concordia.ca/~haarslev/racer/racer-manual-1-7-7.pdf.
12. Horrocks, I. FaCT Reference Manual v1.6, August 1998, FaCT archive: http://www.cs-.man.ac.uk/~horrocks/FaCT.
13. KACTUS ontology library. http://hcs.science.uva.nl/projects/Kactus/toolkit/intro.html.
14. Linguistic Data Consortium www.ldc.upenn.edu.
15. Minsky M.: A Framework for Representing Knowledge. W: The Psychology of Computer Vision. Ed. P.H.Winston, McGraw-Hill, New York, 1975, s. 211÷277.
16. OWL Web Ontology Language Reference. www.w3.org/TR/owl-ref.
17. Pinto S., Gomez-Perez, Martins J.: Some Issues on Ontology Integration. W: Proceedings of the JCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, 1999.
18. The Rule Markup Initiative RuleML http://www.ruleml.org/.
19. Russel S. J., Norvig, P.: Artificial Intelligence. A modern Approach. Second Edition. Pearson Education International. 2003.
20. A Semantic Web Framework for Java, http://jena.sourceforge.net/.
21. Staab S., Studer R. (eds): Handbook on Ontologies. Springer Verlag, 2004.
22. W3C Semantic Web Activity. www.w3.org/2001/sw.
23. UMLS http://www.nlm.nih.gov/research/umls/.
24. Winston M., Chaffin R., Herrmann D.: A taxonomy of part-whole relationships. W: Cognitive Science, rozdz. 11, s. 417÷444, 1987.
25. WordNet http://wordnet.princeton.edu/.
26. RDF Primer http://www.w3.org/TR/rdf-primer/.

Recenzent: Prof. dr hab. inż. Stanisław Kozielski

Wpłynęło do Redakcji 10 czerwca 2006 r.

Abstract

One of the major and most challenging tasks for modern Information Technology is development of methods aimed at automatically acquiring and processing knowledge stored in the biggest information repository the man has ever created – the Internet. This knowledge is of miscellaneous nature, mainly due to the fact that it is stored in many languages and in numerous formats with different levels of structuring. The “Semantic Web initiative” strives to

achieve this goal by structuring the contents of Internet into publicly available and shared ontologies formulated in a commonly accepted, machine readable format.

In Section 2 of the paper we present some popular knowledge representation methods. First, rule-based systems are briefly discussed. Then the main focus is on ontological methods. One of the first ontological methods of knowledge representations were frame systems based on the Minsky's theory of frames. According to this theory, world consists of uniquely identifiable frames. Frames has their properties (slots) that are frames themselves. There are also constraints put on slots (facets) and other logical axioms. Frames can have instances. Frames theory strongly influenced modern software and knowledge engineering. Another, alternative approach are semantic networks. Expressiveness of semantic networks are similar to frames, although they put stress on graphical representation as well as tend to be more general (e.g. exceptions are possible). As a modern method of ontological knowledge representation, description logics (DLs) are presented. In a DL ontology, a domain is described as a set of concepts that are related to each other with binary relations (roles), and constrained via logical axioms. Decidable reasoning over DL ontologies is possible (which is not the case for frames) and there exist reasoners that proved to be efficient enough for practical applications.

DL ontologies are usually formulated in OWL-DL, a language promoted by W3C Consortium within the framework of Semantic Web initiative. In Section 3 we focus on the Semantic Web as a perspective global environment for publishing and exchanging knowledge throughout Internet. A main means for conveying knowledge are ontologies augmented by procedural rules where necessary. In this context, an ontology is meant as formal, explicit specification of shared conceptualization. An important role in the Semantic Web play programmatic agents that are responsible for searching for knowledge needed for a user in a specific domain.

Section 4 presents selected issues of ontology engineering. Firstly, it focuses on methods of ontology development. The main characteristics of these methods are specified, particularly:

- ontology integration as a process of creation a new ontology using existing ones,
- ontology merging as a process of unification existing ontologies to the one ontology, and
- ontology building as a process of creation the new ontology.

A few examples are given to present these methods. Moreover, the dependences between ontology engineering and software engineering are presented with special focus on the software development process and its influence on ontology development process. The Section 4 is concluded with the description of some existing ontologies that may be used for development of new ones or as a ready-to-use knowledge components of knowledge-based systems.

In Section 5 an alternative approach to knowledge acquisition and processing is presented. This approach is based on natural language processing. The idea consists in reading contents of the Web, interpreting their semantics and storing results in a global ontology. This aims at development of a semantic search engine (in the contrary with present syntactical search engines) that will be able to accept questions put by a user in a natural language and present answers also in natural language.

The paper is concluded by summing up perspectives of the Semantic Web.

Adresy

Krzysztof GOCZYŁA: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, ul. G. Narutowicza 11/12, 80-952 Gdańsk, kris@eti.pg.gda.pl
Teresa ZAWADZKA: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, ul. G. Narutowicza 11/12, 80-952 Gdańsk, tegra@eti.pg.gda.pl