

Marek Nahotko

Instytut Informacji Naukowej i Bibliotekoznawstwa
Uniwersytet Jagielloński

Semantyczny Web i jego ontologie

W końcu lat 90. jeden z głównych ideologów Internetu, Tim Berners-Lee, przedstawił nową wizję funkcjonowania tej sieci, nazywając ją Semantycznym Web (ang. Semantic Web). Semantyczny Web znany jest też pod nazwą Programowalnego Web (ang. Programmable Web) lub Web Wiedzy (ang. Knowledge Web). Podstawową ideą Semantycznego Web jest rozwój obecnie funkcjonującego Web z jego dokumentami hipertekstowymi w kierunku umożliwienia realizacji takich funkcji, jak: inteligentna nawigacja i wyszukiwanie informacji, zautomatyzowane wykorzystanie rozproszonych źródeł informacji oraz usługi przetwarzania wiedzy.

Kluczową rolę w tworzeniu semantycznego Web, szczególnie w zakresie technik reprezentacji wiedzy (dział sztucznej inteligencji), mogą odegrać sieci semantyczne, rachunek predykatów oraz **ontologie** (często nazywane też Modelem Domeny). Tym ostatnim należy poświęcić nieco uwagi. Umożliwiają one przetwarzanie i dystrybucję wiedzy pomiędzy programami w Web. Ontologie ogólnie definiowane są jako *reprezentacja dystrybuowanej konceptualizacji określonej domeny*^[1] czy też *formalna specyfikacja konceptualizacji*^[2]. Konceptualizacja w tych definicjach oznacza abstrakcyjny ogląd świata. Jest to para $\{D, R\}$, gdzie D jest opisywaną domeną, a R jest zestawem relacji określonych na D. Możemy więc stwierdzić, że konceptualizacja w terminologii ontologii oznacza metadane. Ontologie łączą ze sobą terminy ze słowników z jednostkami identyfikowanymi w trakcie konceptualizacji oraz udostępniają definicje służące uściśleniu znaczenia tych terminów. Pozwalają one na jednoznaczne rozumienie domeny, które to rozumienie może być przekazywane osobom i systemom aplikacyjnym. Ontologia jest logiczną teorią reprezentowaną przez intencjonalne znaczenie sformalizowanego słownika, tzn. jego ontologiczne zaangażowanie w określoną konceptualizację rzeczywistości. Intencjonalne modele języka używającego takiego słownika są ograniczone przez jego zaangażowanie ontologiczne. Ontologia pośrednio odzwierciedla to zaangażowanie (oraz konceptualizację będącą jego podstawą) przez przybliżanie modeli intencjonalnych. Ontologie utworzone zostały dla potrzeb sztucznej inteligencji. Wykorzystywane są w dystrybucji wiedzy w trakcie jej wielokrotnego użytkowania. Mogą one przedstawiać złożone relacje pomiędzy obiektami oraz zasady i twierdzenia nieuwzględniane w sieciach semantycznych. Ontologie opisujące wiedzę w określonej dziedzinie często są związane z systemami wiedzy.

Odrębnym zagadnieniem wiążącym się z tworzeniem ontologii jest decyzja o przyjęciu odpowiedniej syntaktyki. Ontologie powstają głównie na potrzeby WWW, więc syntaktyka języków ontologii musi być sformułowana przy użyciu istniejących standardów Web w zakresie reprezentacji informacji. Do tego celu może służyć XML i/lub RDF (ang. Resource Description Framework). RDF udostępnia środki pozwalające na dołączanie semantyki do dokumentu bez potrzeby określania struktury tego dokumentu. Sam RDF jest infrastrukturą pozwalającą na kodowanie, wymianę i wielokrotne wykorzystanie ustrukturalizowanych metadanych. W ten sposób można opisać obiekty, ich klasy i cechy. W odniesieniu do ontologii RDF dostarcza dwóch ważnych narzędzi: standardowej syntaktyki służącej tworzeniu ontologii oraz standardowego zestawu podstawowych pojęć, takich jak wystąpienie i podklasa relacji.

Ontologie wykorzystywane podczas tworzenia systemu są transformowane i tłumaczone na elementy systemu informacyjnego, co ogranicza koszty i zakres analizy konceptualnej. Jeżeli są one używane podczas funkcjonowania systemu, umożliwiają komunikację pomiędzy elementami oprogramowania (komunikacja sterowana ontologią).

Dla obsługi poszczególnych poziomów ogólności stosowane są ontologie różnego rodzaju. Te najwyższego poziomu mogą np. opisywać najbardziej ogólne pojęcia, takie jak: przestrzeń, czas, materia, obiekt, wydarzenie, działanie itp. Ontologie domeny i zadań opisują słownik związany z określoną domeną (np. medycyna lub motoryzacja) lub też zadania lub działania (jak diagnostyka lub sprzedaż), przez uszczegółowianie terminów znajdujących się w ontologii najwyższego poziomu. Natomiast ontologie aplikacji opisują pojęcia zależne od określonej domeny lub zadania, które są często uszczegółowieniem kilku związanych ze sobą ontologii.

Innym celem tworzenia ontologii jest usprawnienie działania systemów wyszukiwania i reprezentacji informacji. Web jest doskonałym polem doświadczalnym dla zastosowań ontologii, począwszy od gromadzenia informacji, a skończywszy na jej wyszukiwaniu.

Można przedstawić kilka zasad i wskazówek dla projektowania ontologii:

- **Jasność.** Definicje powinny być obiektywne, wolne od kontekstów społecznych czy sprzętowych i przedstawione w języku naturalnym.
- **Spójność.** Definiowane pojęcia powinny być logicznie spójne.
- **Rozszerzalność.** Ontologia powinna przewidywać możliwość pojawienia się nowych zastosowań wspólnego słownika.
- **Minimalna tendencyjność kodowania.** Tendencyjność kodowania występuje wtedy, gdy sposób reprezentacji wybrany został wyłącznie z powodu łatwości notacji. Konceptualizacja powinna zostać określona na poziomie wiedzy, niezależnie od określonego sposobu kodowania na poziomie symboli.
- **Minimalne zaangażowanie ontologiczne.** Mówi się, że agent (programowy) jest zaangażowany w ontologii, gdy jego działania są zgodne z definicjami ontologii. Ontologia powinna posiadać minimalną ilość twierdzeń o modelowanym świecie.

Przykładem wykorzystania ontologii w Semantycznym Web są strony komercyjne, gdzie ontologie stosowane są do:

- umożliwienia opartej na urządzeniach technicznych komunikacji pomiędzy kupującym i sprzedającym;
- umożliwienia pionowej integracji rynków np. <http://www.verticalnet.com>^[3];
- przekazywania opisów możliwych do wielokrotnego wykorzystania pomiędzy różnymi rynkami.

Innym przykładem wykorzystania ontologii jest ich zastosowanie w wyszukiwarkach. Dzięki nim mogą one odejść od sposobu pracy opartego tylko na przeszukiwaniu słów kluczowych.

Ontologia zawiera hierarchiczny opis istotnych pojęć w domenie (is-a-hierarchy) oraz przedstawia zasadnicze właściwości każdego pojęcia przy pomocy mechanizmu atrybut-wartość. Dodatkowo możliwe jest wprowadzenie dalszych relacji pomiędzy pojęciami przy pomocy kolejnych wyrażeń logicznych. Ostatecznie jednostki w domenie są przydzielane do jednego lub kilku pojęć nadrzędnych, a poprzez nie związane z pojęciem zajmującym jeszcze wyższe miejsce w hierarchii. Zwykle przyjmuje się zasadę segmentacji kategorii, do której zakwalifikowano więcej niż 20-25 adresów. Kategorie te w wielu ontologiach obejmują od kilku do kilkudziesięciu tysięcy podkategorii, organizowanych na 5-10 poziomach hierarchii. Ich przykłady podane są w [4, s. 213-216]^[4].

Ontologie spełniają funkcje podobne do schematów baz danych. Różnią się jednak od nich kilkoma cechami:

- język służący definiowaniu ontologii jest bogatszy zarówno syntaktycznie, jak i semantycznie;
- ontologia musi posiadać szeroko uzgodnioną terminologię, ponieważ jednym z jej zadań jest wymiana informacji;
- ontologia zawiera wiedzę na temat domeny.

Ontologie tworzone są przy użyciu specjalizowanych języków, takich jak: Ontobroker, SHOE (Simple HTML Ontology Extensions - <http://www.cs.umd.edu/projects/plus/SHOE/>^[5]), OIL (Ontology Inference Layer lub Ontology Interface Layer - <http://www.ontoknowledge.org/oil/>^[6]) czy DAML (DARPA Agent Markup Language - <http://www.daml.org/>^[7]).

Rysunek poniżej przedstawia przykład małej i prostej ontologii (wyrażonej w języku OIL - zob.^[8]).

class-def zwierzę	%zwierzęta są klasą
class-def roślina	%rośliny są klasą
subclass-of NOT zwierzę	%wyłączenie ze zwierząt
class-def drzewo	
subclass-of roślina	%drzewa są rodzajami roślin
class-def gałąź	
slot-constraint is-part-of has-value drzewo	%gałęzie są częścią drzew
class-def liść	
slot-constraint is-part-of has-value gałąź	%liście są częścią gałęzi
class-def definicja drapieżników	%drapieżniki są zwierzętami
subclass-of zwierzę	
slot-constraint zjada value-type zwierzę	%które zjada tylko inne zwierzęta
class-def definicja roślinożernych	%roślinożerne są zwierzętami
subclass-of zwierzę	
slot-constraint zjada value-type roślina	%które zjada tylko rośliny lub części roślin
OR (slot-constraint is-part-of has-value roślina)	
class-def żyrafa	%żyrafy są zwierzętami

subclass-of zwierzę	
slot-constraint <i>zjada</i> value-type liść	%oraz jedzą liście
class-def lew	
subclass-of zwierzę	%lwy też są zwierzętami
slot-constraint <i>zjada</i> value-type roślinożerne	%ale zjadają one roślinożerne
class-def rośliny jadalne	%rośliny jadalne to rośliny zjadane
subclass-of roślina	%przez drapieżników i roślinożerców
slot-constraint <i>zjadany</i> has-value roślinożerny, drapieżnik	

Rys. 1. Przykład ontologii.

Ontologia zawiera listę definicji klas ("class-def") i cech ("slot-def"); te ostatnie zostały w powyższym przykładzie pominięte. Definicja klasy łączy nazwę klasy z jej opisem, zawierając następujące elementy (z których każdy jest opcjonalny):

- Definicja **typu** - może on być zarówno podstawowy, jak i definiowalny; podstawowe klasy pozwalają na tworzenie niezbędnych warunków dla przynależności do klasy.
- **Subclass-of** - lista jednego lub kilku wyrażeń klasy (class-expressions - zob. niżej). Klasa zdefiniowana w definicji klasy musi być podklasą każdego wyrażenia klasy na liście.
- **Slot-constraint** - lista zero lub więcej ograniczeń cech (zob. poniżej). Definiowana klasa w definicji klasy musi być podklasą każdego slot-constraint na liście (zauważmy, że slot-constraint definiuje klasę).

Class-expression może być nazwą klasy, ograniczeniem cech lub arbitralnie utworzonym zestawem połączeń boole'owskich class-expressions.

Slot-constraint jest listą jednego lub więcej ograniczeń stosowanych do cechy. Cecha jest relacją binarną (tzn. jej wystąpieniami są pary jednostek), ale slot-constraint to faktyczne definicje klasy - jego wystąpieniami są te jednostki, które spełniają ograniczenia.

Slot-constraint zawiera następujące główne składniki:

- **Name** - nazwa cechy (ciąg znaków).
- **Has-value** - lista jednego lub kilku wyrażeń klasy. Każdy przypadek klasy definiowanej przez ograniczenie cechy (slot-constraint) musi być połączony poprzez relacje cechy z wystąpieniem każdej class-expression na liście.
- **Value-type** - lista jednej lub kilku class-expressions. Jeżeli wystąpienie klasy definiowanej przez slot-constraint jest powiązane poprzez relację cechy z pewną jednostką x, to x musi być wystąpieniem każdego class-expression na liście.

Ontologia OIL jest strukturą składającą się z kilku części, które także mogą posiadać własną strukturę, niektóre z nich są opcjonalne, a inne powtarzalne. Dla opisu ontologii w OIL wyróżnia się trzy poziomy:

- Poziom obiektu, na którym opisywane są konkretne wystąpienia ontologii.
- Pierwszy poziom meta, na którym umieszczane są rzeczywiste definicje ontologiczne. Tu definiuje się terminologię, która może być uprzedmiotowiona na poziomie obiektu. Jest to zasadniczy poziom OIL (nazywany także definicją ontologii) będący narzędziem służącym tworzeniu ustrukturalizowanych słowników o dobrze zdefiniowanej semantyce.
- Drugi poziom meta (tzn. poziom meta-meta), nazywany też kontenerem ontologii, służy opisowi cech ontologii, takich jak: autor, nazwa, przedmiot itp. Dla wyrażania metadanych ontologii często używa się Dublin Core Metadata Element Set (<http://www.dublincore.org/>^[9]).

Do tworzenia kontenera ontologii zaadaptowano zestaw elementów Dublin Core (<http://ebib.oss.wroc.pl/standard/dc.html>^[10]). Pomimo tego, że w DC każdy element jest opcjonalny i powtarzalny, w OIL wprowadzono w tym zakresie pewne ograniczenia. W poniższym wykazie^[11] elementy obligatoryjne oznaczono znakiem "+". Elementy oznaczone "*" są opcjonalne.

- **+Tytuł** - nazwa ontologii;
- **+Twórca** - nazwa agenta (osoby, grupy osób, oprogramowania) - twórcy ontologii;
- ***Opis rzeczowy** - słowa kluczowe lub symbole klasyfikacji;
- **Opis** - tekst w języku naturalnym opisujący treść ontologii;
- ***Wydawca** - definiuje jednostkę odpowiedzialną za udostępnienie źródła;
- ***Współtwórca** - nazwa agenta (osoby, grupy osób, oprogramowania) pomagającego w tworzeniu ontologii;
- ***Data** - data utworzenia, modyfikacji lub udostępnienia ontologii;
- **+Typ** - rodzaj źródła. Wartością standardową jest ontologia;
- ***Format** - cyfrowa materializacja źródła;
- **+Identyfikator** - URI ontologii;
- ***Źródło** - opcjonalny odsyłacz (URI) do źródeł, z którego pochodzi ontologia;
- **+Język** - język ontologii. Predefiniowaną i wymaganą wartością jest "OIL";
- ***Relacja** - wykaz odesłań do innych ontologii OIL;

- ***Własność** - informacja o prawach własności dotyczących ontologii.

W niniejszej pracy przedstawione zostały ontologie jako wykonywalne, formalne konceptualizacje będące reprezentacją porozumienia osiągniętego przez grupy ludzi. Pozwalają one na budowanie aplikacji opartych na przetwarzaniu wiedzy w Semantycznym Web. Ich teoria i praktyka jest obecnie dopiero tworzona, stąd do rozwiązania pozostaje jeszcze wiele kwestii ich dotyczących. Z jednej strony wielu specjalistów wątpi, czy możliwa jest realizacja tych idei i czy przyniosą one praktyczne korzyści, z drugiej strony w USA wydaje się na badania związane z ontologiami dziesiątki milionów dolarów (np. w Defense Advanced Research Projects Agency - DARPA), mając nadzieję, że wpłyną one w istotny sposób na Web i ułatwią dostęp do jego zasobów.

Przypisy

- [1] HODGE, Gail. Taxonomy of Knowledge Organization Sources/Systems. In *Networked Knowledge Organisation Systems/Services NKOS* [on-line]. [dostęp 29 września 2003]. Dostępny w World Wide Web: http://nkos.slis.kent.edu/KOS_taxonomy.htm.
- [2] GRUBER, Thomas. A translation Approach to Portable Ontology Specification. *Knowledge Acquisition*, 1993, Vol. 5.
- [3] *Verticalnet* [on-line]. [dostęp 29 września 2003]. Dostępny w World Wide Web: <http://www.verticalnet.com/>.
- [4] SOSIŃSKA-KALATA, Barbara. *Klasyfikacja : struktury organizacji wiedzy, piśmiennictwa i zasobów informacyjnych*. Warszawa: Wydaw. SBP, 2002. ISBN 83-87629-87-1.
- [5] SHOE. In *Department of Computer Science* [on-line]. [dostęp 29 września 2003]. Dostępny w World Wide Web: <http://www.cs.umd.edu/projects/plus/SHOE/>.
- [6] OIL. In *On-To-Knowledge* [on-line]. [dostęp 29 września 2003]. Dostępny w World Wide Web: <http://www.ontoknowledge.org/oil/>.
- [7] DAML. In *The DARPA Agent Markup Language (DAML)* [on-line]. [dostęp 29 września 2003]. Dostępny w World Wide Web: <http://www.daml.org/>.
- [8] DECKER, Stefan. The Semantic Web - on the respective Roles of XML and RDF. In *Stanford University Database Group* [on-line]. [dostęp 29 września 2003]. Dostępny w World Wide Web: <http://www-db.stanford.edu/%7Estefan/paper/2000/xmlrdf.pdf>.
- [9] *Dublin Core Metadata Initiative* [on-line]. [dostęp 29 września 2003]. Dostępny w World Wide Web: <http://www.dublincore.org/>.
- [10] Dublin Core Metadata Initiative. In *EBIB Elektroniczny Biuletyn Informacyjny Bibliotekarzy* [on-line]. [dostęp 29 września 2003]. Dostępny w World Wide Web: <http://ebib.oss.wroc.pl/standard/dc.html>.
- [11] HORROCKS, Ian. The Ontology Interface Layer OIL. In *On-To-Knowledge* [on-line]. [dostęp 29 września 2003]. Dostępny w World Wide Web: <http://www.ontoknowledge.org/oil/TR/oil.long.html>.

