

Semantic Technologies, czyli Oracle i Web 3.0

Mikołaj Morzy
Instytut Informatyki Politechniki Poznańskiej
e-mail: Mikołaj.Morzy@put.poznan.pl

Abstrakt. Sieć semantyczna, zwana także modelem Web 3.0, to wizja Internetu, w którym dane są przechowywane, opisywane i powiązane w taki sposób, aby mogły być wykorzystane nie tylko przez ludzi, ale także przez maszyny (programy, pająki sieciowe czy inteligentnych agentów). Format danych powinien umożliwiać maszynom „rozumienie” danych w stopniu wystarczającym do tego, aby dane mogły podlegać automatycznej integracji, negocjacji, czy manipulacji. Na tę wizję Internetu jutra składa się wiele technologii, służących przede wszystkim do semantycznego wzbogacania danych, między innymi, XML jako wspólna składnia opisu danych, XML Schema jako język opisu typów danych i ich struktury, RDF jako sposób zapisywania metadanych o związkach między danymi, OWL do definicji wspólnych słowników, czy wreszcie SPARQL jako język zapytań. Opcja Oracle Spatial 11g serwera bazy danych Oracle 11g Enterprise Edition zawiera zaawansowane mechanizmy zarządzania danymi semantycznymi. Umożliwia ona, między innymi, wykorzystanie języków RDF, RDFS i OWL bezpośrednio w bazie danych, wzbogacanie danych relacyjnych o warstwę semantyki, wydawanie zapytań do danych relacyjnych przy wsparciu ontologii zdefiniowanych w OWL, pełny wachlarz operacji DML dla danych przechowywanych w RDF i OWL, wnioskowanie za pomocą silników RDF i OWL, a także wydawanie zapytań w języku zbliżonym do standardu SPARQL. Celem niniejszego artykułu jest zaznajomienie czytelnika z podstawami sieci semantycznych oraz szczegółowe przedstawienie rozwiązań wchodzących w skład pakietu Oracle Semantic Technologies.

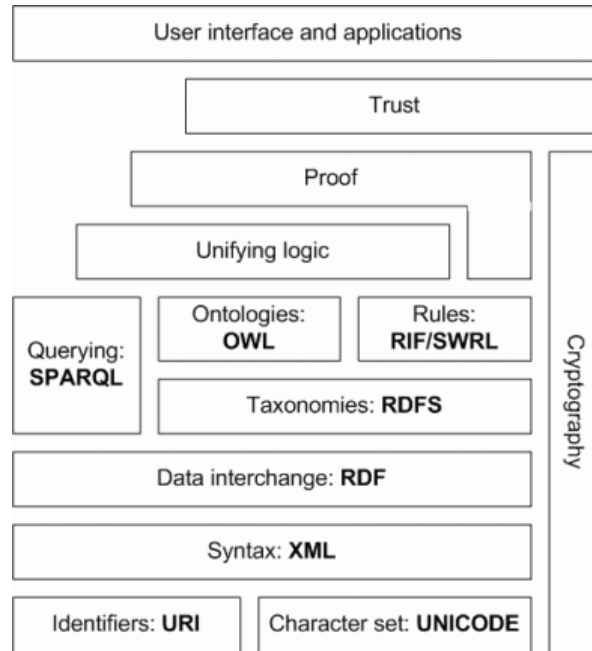
1. Wprowadzenie do sieci semantycznej

Sieć semantyczna (ang. *Semantic Web*), to pojęcie trudne do zdefiniowania ze względu na swoją objętość oraz mnogość znaczeń i interpretacji. Słownik wyrazów obcych W. Kopalińskiego tłumaczy termin „semantyka” jako „nauka o znaczeniu i zmianach znaczeń wyrazów, nauka o stosunkach między wyrażeniami, o stosunku wyrażenia do oznaczanych przedmiotów i stosunku wyrażenia do mówiącego podmiotu” (Kopaliński, 2007). Sieć semantyczna to sieć, w której poszczególne elementy posiadają swoje znaczenie, dostępne nie tylko ludziom, ale także maszynom. Sieć semantyczna to nie jest konkretny produkt, specyfikacja czy standard. To bardziej idea lub wizja, a nie technologia. Ostatecznym celem projektu sieci semantycznej jest udostępnienie wszystkich danych zamieszczonych w Internecie do przetwarzania: ludziom i maszynom. Łatwo się domyślić, że tak rozmyta i niejasna definicja pociąga za sobą wielość perspektyw spojrzenia na to, czym jest sieć semantyczna. Poniżej przedstawiono niepełną listę możliwych interpretacji (Passin, 2004)

- *dane czytelne dla maszyn:* „idea definiowania danych w Internecie i ich łączenia w taki sposób, że mogą być wykorzystywane także przez maszyny, nie tylko do celów wyświetlania, ale także automatyzacji, integracji i wielokrotnego wykorzystania w wielu różnych aplikacjach” (Herman, 2008)
- *inteligentni agenci:* „celem sieci semantycznej jest uczynienie aktualnego Internetu bardziej czytelnym dla maszyn w celu umożliwienia inteligentnym agentom pobierania i manipulowania informacjami” (Cost, Finin, & Joshi, 2002)
- *rozproszona baza danych:* „sieć semantyczna ma pełnić dla danych tę samą rolę, jaką HTML pełni dla danych tekstowych: ma dostarczać elastyczności umożliwiającej reprezentowanie dowolnej bazy danych oraz reguł logicznych oraz łączenie tych baz danych” (Berners-Lee, Karger, Stein, Swick, & Weitzner, 2000)
- *automatyczna infrastruktura:* „sieć semantyczna to nie aplikacja, to infrastruktura, która, jeśli zostanie poprawnie zaprojektowana, może stanowić istotny wkład do ewolucji ludzkiej wiedzy” (Berners-Lee, Hendler, & Lassila, The Semantic Web, 2001)

- *adnotacje*: „sieć semantyczna wzbogaca aktualną sieć o adnotacje zapisane w języku przetwarzalnym maszynowo, dodatkowo, adnotacje są ze sobą powiązane” (Euzenat, 2002)

Na sieć semantyczną składa się duża liczba różnych technologii: URI, XML, XSD, N3, Turtle, N-Triples, RDFS, OWL, WSDL, UDDI, SPARQL i wiele innych. Jest to hierarchiczna architektura wielowarstwowa, w której usługi każdej warstwy wymagają do działania obecności warstw niższych. Koncepcję warstw sieci semantycznej prezentuje Rysunek 1.



źródło: Wikipedia

Rys. 1. Stos technologii i warstw sieci semantycznej

U podstaw sieci semantycznej leży standard kodowania znaków (UNICODE) oraz ujednolicony sposób identyfikacji zasobów sieci semantycznej przy użyciu identyfikatorów URI (Group, 2001). Językiem opisu danych jest w sieci semantycznej język XML (W3C, Extensible Markup Language, 2008), umożliwiający definiowanie struktury dokumentów. Dodatkowo, wykorzystanie języka XSD (W3C, XML Schema Part 0: Primer Second Edition, 2004) umożliwia definiowanie schematów dokumentów i deklarowanie ograniczeń integralnościowych i referencyjnych. Warstwa wymiany i składowania danych wymaga wykorzystania modelu RDF (W3C, RDF/XML Syntax Specification, 2004). Jest to alternatywa dla przechowywania danych przy użyciu modelu relacyjnego, obiektowego czy semistrukturalnego. Podstawowym językiem zapytań dla danych składowanych w formacie RDF jest język SPARQL (W3C, SPARQL Query Language for RDF, 2008) i jego pochodne. Model RDF umożliwia przechowywanie danych atomowych, tzn. informacji o zasobach, ich własnościach i związkach między zasobami. Informacje dodatkowe, takie jak np. taksonomie zasobów, spójne słowniki (ontologie) czy własności związków są specyfikowane za pomocą języków RDFS (W3C, RDF Vocabulary Description Language 1.0: RDF Schema, 2004) oraz OWL (W3C, OWL Web Ontology Language Overview, 2004). Poza regułami wnioskowania wbudowanymi w RDFS i OWL, użytkownicy mają możliwość definiowania swoich własnych reguł, na podstawie których automaty wnioskujące mogą indukować wiedzę z bazy danych. Aktualnie, najpopularniejszy standard specyfikacji reguł użytkownika to SWRL (W3C, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004). Powyżej warstwy języka specyfikacji reguł sieć semantyczna nie doczekała się jeszcze uznanych standardów. Warstwy odpowiedzialne za mechanizmy wnioskowania i rodzaje logiki wykorzystywanej do wnioskowania (logiki pierwszego i drugiego rzędu, logiki temporalne, modalne, rozmyte, opisowe, itd.), sposoby rozwiązywania sprzeczności w trakcie wnioskowania, czy sposoby zapewniania

zaufania i wiarygodności usług sieci semantycznej są bardzo aktywnie rozwijane, ale nie wykrywały się jeszcze rozwiązania, które zostałyby powszechnie uznane za standardowe.

W niniejszym artykule skupiono się jedynie na podzbiorze technologii sieci semantycznej, które wchodzi w skład pakietu Oracle Semantic Technologies. W następnym rozdziale przedstawiono dwie kluczowe technologie: metodę opisu zasobów za pomocą języka RDF oraz pojęcia ontologii i wnioskowania z ontologii.

2. Podstawowe języki sieci semantycznej

2.1. RDF

RDF (ang. *Resource Description Framework*) (W3C, RDF Vocabulary Description Language 1.0: RDF Schema, 2004) to uniwersalny model reprezentacji danych. Umożliwia opisywanie większości dostępnych typów danych, opisuje strukturę zbiorów danych oraz związki pomiędzy poszczególnymi danymi. Charakterystyczną cechą RDF jest uderzająca prostota. RDF definiuje zasoby (ang. *resources*) i wyrażenia (ang. *statements*) formułowane na temat zasobów. Każde wyrażenie ma postać trójki (ang. *triple*) składającej się z podmiotu (ang. *subject*), predykatu (ang. *predicate*) i obiektu (ang. *object*). Czasami można napotkać też alternatywną nomenklaturę, w której predykat jest nazywany *własnością* (ang. *property*) a obiekt jest nazywany *wartością* (ang. *value*). Przykładowe trójki wyrażone w RDF mają następującą postać¹

```
('King', 'managerOf', 'Blake')
```

```
('Blake', 'worksIn', 'Sales')
```

```
('Ford', 'salary', 3000)
```

Wartością obiektu może być literał (np. łańcuch znaków lub liczba) lub inny zasób. Literały mogą być typowane, tzn. można wskazać, że dany łańcuch znaków powinien być interpretowany jako data, liczba czy też łańcuch znaków zgodny z określonym wzorcem. Podmiotem wyrażenia RDF musi być tylko i wyłącznie zasób, nie może być nim literał. Pewnym ograniczeniem RDF jest fakt, że konkretna trójka RDF nie jest traktowana jako zasób i nie posiada swojej własnej tożsamości. W związku z tym trójka RDF nie może być podmiotem wyrażenia RDF, innymi słowy, nie można (w łatwy sposób, bez uciekania się do reifikacji) specyfikować wyrażen opisujących inne wyrażenia. Taka możliwość byłaby pożądana gdyby istniała konieczność np. oceniania wiarygodności poszczególnych wyrażen, przypisywania im wagi, itp.

Podmioty wyrażen RDF są unikalnie identyfikowane za pomocą odnośników URI (ang. *Uniform Resource Identifier*) (Group, 2001). Odnośniki URI umożliwiają opisywanie i jednoznaczne identyfikowanie poszczególnych zasobów. Pewnym problemem jest nieczytelność odnośników URI, w praktyce wygodniejsze jest posługiwanie się etykietami do nazywania poszczególnych zasobów. Istnieje nawet standardowa cecha <http://www.w3.org/2000/01/rdf-schema#label> reprezentująca etykietę. Innym sposobem uproszczenia zapisu jest wykorzystanie *przestrzeni nazw* (ang. *namespaces*). Nie każdy zasób musi być identyfikowany za pomocą odnośnika URI, istnieją zasoby, których istnienie i tożsamość jest zdeterminowana przez związki z innymi zasobami (analogicznie do istnienia encji słabych w tradycyjnym modelu związków encji). Przykładowo, zasób reprezentujący pracownika Forda i identyfikowany przez URI <http://www.ploug.org.pl/emp/Ford> jest związany z zasobem reprezentującym adres domowy i posiadającym: nazwę ulicy, numer domu, kod pocztowy, itp. Sam zasób reprezentujący adres nie potrzebuje unikalnego identyfikatora, ponieważ jest całkowicie określony przez predykat <http://www.ploug.org.pl/emp/livesAt> wiążący adres z pracownikiem Ford. Takie zasoby nazywane są węzłami *anonimowymi* lub *pustymi* (ang. *anonymous node*, *blank node*, *b-node*). Kolekcja wyrażen RDF jest nazywana *składnicą RDF* (ang. *RDF store*), *bazą wiedzy* (ang. *knowledge base*), *zbiorem danych RDF* (ang. *RDF data store*), czy też po prostu bazą danych.

¹ Wszystkie przykłady bazują na popularnym schemacie użytkownika SCOTT z tabelami EMP, DEPT

Każdy schemat relacyjny może być z łatwością przetłumaczony do modelu RDF, a dane przechowywane w schemacie relacyjnym mogą być zaimportowane do składnicy RDF. Pojawia się zatem pytanie, jakie zalety i wady przejawia RDF w stosunku do tradycyjnego modelu relacyjnego. Po stronie wad zaliczyć należy bez wątpienia znacznie wolniejsze i mniej efektywne przetwarzanie danych, nieczytelną strukturę oraz nadmiarowość modelu. W przypadku danych relacyjnych własność (definicja atrybutu) jest specyfikowana tylko raz, w nagłówku tabeli. W modelu RDF każdy zasób (odpowiadający z grubsza krotce w tabeli) posiadający pewną własność musi deklarować posiadanie tej własności niezależnie. Jeśli sposób użycia danych jest strukturalny i dobrze zdefiniowany a schemat bazy danych nie podlega ewolucji, wówczas wykorzystanie modelu RDF mija się z celem. Jednakże, RDF posiada wiele istotnych zalet w stosunku do modelu relacyjnego:

- umożliwia integrację danych z wielu różnych schematów,
- pozwala na całkowicie elastyczne manipulowanie strukturą poszczególnych zasobów, ponieważ własności każdego zasobu są niezależne od pozostałych zasobów,
- ułatwia wielokrotne wykorzystanie tych samych danych w każdej aplikacji potrafiącej przetwarzać dane RDF,
- zezwala na wywodzenie nowych danych z istniejących danych na podstawie reguł wnioskowania zdefiniowanych w ontologii,
- pozwala wykorzystywać ontologie zewnętrzne w stosunku do zbioru danych,
- nie ogranicza danych do lokalnej bazy danych, ponieważ odnośniki URI definiujące zasoby mogą wskazywać na dowolne zasoby leżące poza lokalną składnicą RDF.

Gwałtowny wzrost popularności modelu RDF skutkował pojawieniem się, z jednej strony, powszechnie uznanych standardów, jak np. standard RDF/XML (W3C, RDF/XML Syntax Specification, 2004) reprezentacji danych przy użyciu języka XML, z drugiej strony, implementacją szerokiej gamy narzędzi do przetwarzania RDF, modelowania, importowania i eksportowania danych RDF oraz wnioskowania. Należy jednak pamiętać, że jest to nowa technologia, której wiele aspektów nie doczekało się jeszcze satysfakcjonującego rozwiązania. Przykładowo, kontrowersyjną kwestią pozostaje interpretacja odnośników URI. Jeśli odnośnik URI jest adresem URL, można śmiało zakładać, że rzeczywistym zasobem jest zasób wskazywany przez adres URL (strona WWW, dokument XML, itd.) W ogólności jednak URI nie wskazuje na żaden konkretny dokument, otwarcie w przeglądarce odnośnika <http://www.ploug.org.pl/emp/Ford> nie zwróci w odpowiedzi żadnego dokumentu. Nawet jeśli pod wskazywanym adresem URI znalazłby się jakikolwiek dokument, można tylko mieć nadzieję, że zawiera on szczegółowy opis definiowanego zasobu. Nie istnieje także żaden standardowy sposób odwoływania się do konceptów zdefiniowanych w jakimkolwiek dokumencie, ponieważ nie wiadomo, czy URI wskazuje na koncept, czy też na sam dokument. Dopóki nie wykrystalizują się pewne standardy interpretacji znaczenia odnośników URI, obowiązek interpretacji spoczywa na osobach wykorzystujących składnice RDF. Innym problemem wynikającym z używania RDF jest nieuniknione pojawianie się sprzeczności w danych. Ponieważ w modelu RDF każda trójka może być dowolnym wyrażeniem dotyczącym dowolnego zasobu, pojawienie się konfliktowych lub sprzecznych informacji jest tylko kwestią czasu, tym bardziej, że RDF funkcjonuje w otwartym i rozproszonym środowisku Sieci. Oczywiście wnioskowanie w obecności sprzeczności jest znacząco utrudnione i może często prowadzić do mylnych wniosków. W chwili obecnej RDF nie posiada żadnych mechanizmów radzenia sobie ze sprzecznościami (choć może je wykrywać). Wreszcie, RDF nie zawiera możliwości negowania wyrażen oraz nie umożliwia formułowania wyrażen dotyczących ogólnych klas zasobów (np. wyrażen dotyczących wszystkich pracowników).

2.2. RDFS i OWL

Ontologia, jako dyscyplina filozoficzna, to nauka o istnieniu. Ontologia identyfikuje byty, które faktycznie mogą istnieć, dostarcza metod opisu tych bytów oraz klasyfikuje je i wyznacza granice

dyskursu, określając te byty, o których można dyskutować w danym kontekście (Passin, 2004). W ostatnich latach termin „ontologia” został zawłaszczony przez informatykę. Zamiast o ontologii jako dyscyplinie zaczęto mówić o ontologiach, czyli formalnych definicjach zbiorów konceptów wykorzystywanych w danej dziedzinie. Istnieją zatem ontologie dla medycyny, chemii, ekonomii, podatków, czy leśnictwa. Wedle tego rozumienia na ontologię składają się: ustalony słownik słów wykorzystywanych do opisu konceptów, zbiór taksonomii, czyli hierarchii występujących w słowniku, a także dodatkowe informacje o cechach konceptów i związkach między konceptami. Zazwyczaj, częścią ontologii są informacje o strukturze (zasoby klasy `Employee` posiadają własności `age`, `income`, `job`), ograniczeniach związanych z własnościami (wartościami własności `age` są liczby całkowite), oraz o cechach poszczególnych własności (własność `isManagerOf` jest tranzytywna, własność `isAcquainted` jest zwrotna). Ontologie umożliwiają wzbogacanie oryginalnych składnic RDF o nowe fakty, które mogą być wywiedzione z istniejących faktów na podstawie reguł wnioskowania i własności zdefiniowanych w ontologiach. Dodatkowo, ontologie umożliwiają uspoźnienie danych pochodzących z różnych źródeł, ponieważ, jak już wspomniano, jednym z celów ontologii jest zdefiniowanie przestrzeni dyskursu i zdefiniowanie zbioru pojęć uznawanych za poprawne w danym kontekście.

Podstawowym językiem specyfikacji ontologii jest RDFS (ang. *RDF Schema*) (W3C, *RDF Vocabulary Description Language 1.0: RDF Schema*, 2004). Należy tu zaznaczyć, że wybór nazwy jest wyjątkowo nieszczęśliwy. Nazwa *RDF Schema* sugeruje, że związek między RDF i RDFS jest podobny do związku między XML i XML Schema, nie jest to jednak prawdą. O ile XML Schema definiuje poprawną strukturę dokumentów XML, o tyle RDFS służy do specyfikacji poprawnego słownika, który można wykorzystywać w danym kontekście. RDFS definiuje najważniejsze elementy służące do specyfikowania ontologii (konkretne przykłady użycia poszczególnych konstruktów zostaną przedstawione w dalszej części artykułu):

- klasy: `rdfs:Resource`, `rdfs:Class`, `rdfs:Literal`, `rdf:Property`, `rdf:Statement`
- własności: `rdf:type`, `rdfs:subClassOf`, `rdfs:subProperty`
- ograniczenia: `rdfs:domain`, `rdfs:range`
- kontenery: `rdf:Bag`, `rdf:Seq`, `rdf:Alt`, `rdfs:Container`
- własności pomocnicze: `rdfs:seeAlso`, `rdfs:isDefinedBy`, `rdfs:comment`, `rdfs:label`

Powyższy zestaw konstruktów nie jest wystarczający aby poprawnie zamodelować bardziej złożone zależności między klasami i własnościami, np. możliwość specyfikowania liczności związków (pracownik może pracować w co najwyżej dwóch zespołach) lub określania związków wymaganych (każdy pracownik musi być przypisany do co najmniej jednego projektu). W odpowiedzi na oczywiste niedoskonałości języka RDFS powstała rodzina języków OWL (ang. *Web Ontology Language*) (W3C, *OWL Web Ontology Language Overview*, 2004). OWL występuje w trzech wersjach: OWL Lite, OWL DL i OWL Full. OWL Lite jest najmniejszym zbiorem konstruktów, o najmniejszej mocy wyrażania, lecz jest wspierany przez największą liczbę silników wnioskowania. OWL DL wspiera to podzbiór konstruktów zgodnych z logiką opisową (ang. *descriptive logics*), pozwala na specyfikowanie bardziej złożonych zależności a jednocześnie pozostaje w sferze obliczalności. OWL Full, czyli pełny język OWL, zawiera najbogatszy zestaw konstruktów, ale wnioskowanie na podstawie niektórych z tych konstruktów może być bardzo kosztowne lub wręcz niemożliwe. Wszystkie wersje języka OWL zawierają w sobie, jako podzbiór, cały język RDFS. Nie sposób tutaj wymienić wszystkich konstruktów języka OWL, więc poniżej przedstawiono tylko kilka przykładowych, których znaczenie powinno być oczywiste: `owl:DataRange`, `owl:sameAs`, `owl:SymmetricProperty`, `owl:equivalentClass`, `owl:minCardinality`, itd. Przypadki użycia wybranych konstruktów zostaną przedstawione w dalszej części artykułu.

3. Oracle Semantic Technologies

Oracle Semantic Technologies (OST) to ogólna nazwa zbioru narzędzi umożliwiających przetwarzanie danych semantycznych bezpośrednio wewnątrz systemu baz danych Oracle. OST jest częścią opcji Spatial bazy danych Oracle 11g Enterprise Edition, ale do działania wymaga również zainstalowanych opcji Partitioning i Advanced Compression. Najważniejszą częścią OST jest repozytorium RDF umożliwiające składowanie trójek RDF natywnie wewnątrz bazy danych. Do przechowywania większości metadanych o składnicach RDF, ontologiach, indeksach i regułach służy schemat użytkownika MDSYS. OST zapewnia użytkownikom następujące korzyści:

- możliwość składowania, ładowania i przetwarzania grafów RDF bezpośrednio w silniku bazy danych,
- wsparcie dla wnioskowania za pomocą języka OWL, RDFS oraz reguł wnioskowania definiowanych przez użytkownika,
- implementację dużej części standardu W3C SPARQL języka zapytań do danych semantycznych pod postacią funkcji `SEM_MATCH()`
- możliwość wzbogacania danych relacyjnych za pomocą ontologii zdefiniowanych w RDF/OWL i integrację ontologii z tradycyjnym językiem SQL poprzez funkcje `SEM_RELATED()` i `SEM_DISTANCE()`,
- obsługę dowolnie wielkich składnic RDF liczących ponad miliard trójek RDF i zapewnienie wydajności poprzez użycie technik kompresji i partycjonowania,
- otwarcie danych semantycznych na potrzeby aplikacji zewnętrznych poprzez dostarczenie interfejsów programistycznych do składnicy RDF (Jena Adaptor API),
- wykorzystanie wszystkich zalet systemu zarządzania bazą danych: skalowalności, dostępności, bezpieczeństwa i wydajności.

W przypadku wielu aplikacji dane semantyczne są pobierane z zewnętrznych repozytoriów i ładowane do bazy danych Oracle. OST umożliwia opracowanie elastycznej i wydajnej procedury importowania danych poprzez wsparcie trzech trybów importowania. Najszybszym trybem importowania jest *ładowanie hurtowe* (ang. *bulk loading*), polegające na załadunku danych z pliku tekstowego zgodnego z formatem N-Triple (N-Triples) do tabeli tymczasowej, a następnie zastosowanie procedury `SEM_APIS.BULK_LOAD_FROM_STAGING_TABLE()` do załadunku do docelowej tabeli. Ograniczeniem tego rozwiązania jest fakt, że procedura ładowania hurtowego nie potrafi obsługiwać literałów dłuższych niż 4 KB. Drugi tryb importowania polega na wykorzystaniu interfejsu programistycznego Java do załadunku danych. Także i w tym przypadku dane źródłowe muszą być zgodne z formatem N-Triple. Wreszcie, dane semantyczne mogą być załadowane za pomocą tradycyjnych operacji DML (INSERT, UPDATE) wykorzystujących do tworzenia trójek RDF konstruktor typu `SDO_RDF_TRIPLE_S`.

OST wspiera wnioskowanie i wywodzenie nowych faktów na bazie zgromadzonej wiedzy przy użyciu ontologii i reguł wnioskowania. Użytkownik może definiować swoje własne reguły wnioskowania, może także polegać na regułach zaszytych w językach RDFS i OWL. Jak już wspomniano wcześniej, w przypadku języka OWL nieuniknione jest zawarcie kompromisu między siłą wyrażania a obliczalnością. OST wspiera trzy podzbiory reguł wnioskowania, które powinny zaspokoić potrzeby większości aplikacji. Najprostszy podzbiór, zwany RDFS++, to oryginalny zbiór konstruktorów zawartych w języku RDFS oraz dwa dodatkowe konstrukty: `owl:sameAs` (wskazuje, że dwa zasoby są tym samym, mimo że posiadają różne identyfikatory URI) oraz `owl:inverseFunctionalProperty` (własność jest odwrotnie funkcyjna jeśli obiekt własności jednoznacznie definiuje podmiot, np. własność `isManagerOf` jest odwrotnie funkcyjna, ponieważ dla danego zasobu *y* nie mogą istnieć dwa podmioty x_1 i x_2 takie, że spełnione są jednocześnie $\langle x_1, isManagerOf, y \rangle$ i $\langle x_2, isManagerOf, y \rangle$). Drugi podzbiór to OWLSIF, czyli słownik OWL wzbogacony o semantykę konstruktu IF – ten podzbiór jest implementacją słownika

pD* autorstwa Hermana Horsta (Horst, 2005). Najbogatszy podzbiór o największej mocy wyrażania to OWLPrime, obejmujący konstrukty do definiowania ograniczeń integralnościowych, konstrukty do klasyfikacji cech związków, konstrukty do określania ekwiwalencji zasobów i klas, itd. Rzecz jasna, wybór konkretnego podzbioru ma bezpośredni wpływ na przetwarzanie danych, ponieważ im większa moc wyrażania danego podzbioru semantycznego, tym więcej dodatkowej wiedzy może zostać wywnioskowanej.

Najważniejsze pojęcia składające się na OST to sieć semantyczna, model, baza reguł oraz indeksy regułowe. W terminologii OST sieć semantyczna oznacza graf powstały z wszystkich trójek RDF, w którym podmioty i obiekty tworzą węzły grafu, a własności tworzą etykietowane krawędzie skierowane w grafie. Częścią sieci semantycznej są także puste węzły (*b-nodes*). Utworzenie sieci semantycznej jest pierwszym krokiem koniecznym do załadowania danych semantycznych do bazy danych. W momencie utworzenia sieci semantycznej w schemacie użytkownika MDSYS tworzone są podstawowe obiekty repozytorium a także wskazywana jest przestrzeń tabel, w której będą przechowywane wszystkie modele. Pod pojęciem modelu kryje się zbiór trójek RDF przechowywanych w kolumnie wskazanej tabeli. Typem danych dla danych semantycznych jest wbudowany typ obiektowy SDO_RDF_TRIPLE_S, w momencie tworzenia modelu użytkownik wskazuje tabelę źródłową i właściwą kolumnę w tej tabeli. Wynikiem utworzenia modelu jest utworzenie perspektywy w schemacie MDSYS, nadanie właścicielowi uprawnień potrzebnych do pracy z modelem (w tym uprawnień DML do tabel systemowych przechowujących model) oraz uaktualnienie repozytorium (perspektywa MDSYS.RDF_MODEL\$). Baza reguł to obiekt przechowujący reguły wnioskowania. Jak już wcześniej wspomniano, poza wykorzystaniem predefiniowanych zbiorów reguł (RDFS++, OWLSIF, OWLPrime) użytkownik może definiować własne reguły w postaci obiektów o strukturze *poprzednik* → *następnik*. W momencie utworzenia bazy reguł w schemacie MDSYS tworzona jest perspektywa pokazująca zdefiniowane przez użytkownika reguły oraz nadawane są wszystkie konieczne uprawnienia. W celu przyspieszenia wnioskowania użytkownik może utworzyć indeks regułowy. Indeks regułowy to obiekt przechowujący wszystkie fakty wywnioskowane z bazy wiedzy za pomocą określonego zbioru reguł wnioskujących. Utworzenie takiego indeksu może być bardzo kosztowną operacją, w szczególności jeśli baza wiedzy jest duża i zastosowano zbiór reguł o dużej mocy wnioskowania. Obecność indeksów regułowych zdecydowanie przyspiesza wykonywanie zapytań do bazy wiedzy.

4. Przykład przetwarzania danych semantycznych

W bieżącej sekcji przedstawiono poglądowy przykład wykorzystania Oracle Semantic Technologies do przetwarzania danych semantycznych. W celu zwiększenia przejrzystości prezentacji jako przykładowy zbiór wykorzystano powszechnie znany zbiór danych o pracownikach i departamentach, standardowo tworzony w schemacie użytkownika SCOTT.

Jak już wspomniano wcześniej, pierwsze kroki obejmują utworzenie sieci semantycznej, tabeli do przechowywania danych semantycznych oraz modelu danych semantycznych. Kroki te zostały zilustrowane poniżej:

```
SQL> EXECUTE SEM_APIS.CREATE_SEM_NETWORK('rdf_tblspace');  
SQL> CREATE TABLE emp_rdf (id NUMBER, triple SDO_RDF_TRIPLE_S);  
SQL> EXECUTE SEM_APIS.CREATE_RDF_MODEL('Employees', 'emp_rdf', 'triple');
```

Baza wiedzy zawiera informacje o pracownikach i departamentach. Dla departamentów przechowywana jest ich lokalizacja (nazwa została wykorzystana jako część identyfikatora URI). Dla pracowników przechowywana jest płaca oraz informacje o tym, w którym departamencie dany pracownik pracuje, czy jest przełożonym innych pracowników, czy kieruje jakimś departamentem. Dla uproszczenia, nazwiska pracowników zostały wykorzystane jako fragmenty identyfikatorów URI. Przykładowe fakty umieszczone w bazie wiedzy są następujące:

```

-- King jest menadżerem Blake'a
SQL> INSERT INTO emp_rdf VALUES (1, SDO_RDF_TRIPLE_S('Employees',
    'http://www.ploug.org.pl/emp/King',
    'http://www.ploug.org.pl/emp/managerOf',
    'http://www.ploug.org.pl/emp/Blake'));

-- Blake jest menadżerem Turnera
SQL> INSERT INTO emp_rdf VALUES (2, SDO_RDF_TRIPLE_S('Employees',
    'http://www.ploug.org.pl/emp/Blake',
    'http://www.ploug.org.pl/emp/managerOf',
    'http://www.ploug.org.pl/emp/Turner'));

-- Clark pracuje w departamencie Accounting
SQL> INSERT INTO emp_rdf VALUES (3, SDO_RDF_TRIPLE_S('Employees',
    'http://www.ploug.org.pl/emp/Clark',
    'http://www.ploug.org.pl/emp/worksIn',
    'http://www.ploug.org.pl/emp/Accounting'));

-- Allen pracuje jako Salesman
SQL> INSERT INTO emp_rdf VALUES (4, SDO_RDF_TRIPLE_S('Employees',
    'http://www.ploug.org.pl/emp/Allen',
    'http://www.w3.org/1999/02/22-rdf-syntax-ns#type',
    'http://www.ploug.org.pl/emp/Salesman'));

-- departament Accounting znajduje się w Nowym Jorku
SQL> INSERT INTO emp_rdf VALUES (5, SDO_RDF_TRIPLE_S('Employees',
    'http://www.ploug.org.pl/emp/Accounting',
    'http://www.ploug.org.pl/emp/locatedIn',
    'http://www.ploug.org.pl/emp/NewYork'));

-- Ford zarabia 3000
SQL> INSERT INTO emp_rdf VALUES (6, SDO_RDF_TRIPLE_S('Employees',
    'http://www.ploug.org.pl/emp/Ford',
    'http://www.ploug.org.pl/emp/salary',
    '"3000"^^xsd:decimal'));

-- Smith zarabia 8000
SQL> INSERT INTO emp_rdf VALUES (7, SDO_RDF_TRIPLE_S('Employees',
    'http://www.ploug.org.pl/emp/Smith',
    'http://www.ploug.org.pl/emp/pay',
    '"8000"^^xsd:decimal'));

```

Do tak skonstruowanej bazy wiedzy możemy wydawać zapytania przy wykorzystaniu języka SPARQL. Oracle Semantic Technologies implementuje większą część standardu SPARQL poprzez funkcję tablicową `SEM_MATCH()` umożliwiającą specyfikowanie wzorców grafowych. Przykładowe zapytanie znajdujące wszystkich pracowników pracujących na etacie 'Clerk' ma następującą postać:

```

SQL> SELECT m
      2 FROM TABLE( SEM_MATCH('( ?m rdf:type :Clerk)', SEM_MODELS('Employees'),
      3      null, SEM_ALIASES(SEM_ALIAS('', 'http://www.ploug.org.pl/emp/'), null));

```

M

```

-----
http://www.ploug.org.pl/emp/Smith
http://www.ploug.org.pl/emp/Miller
http://www.ploug.org.pl/emp/Adams

```

Póki co, wynik nie jest oszałamiający. Pamiętać jednak należy, że podstawową zaletą modelu semantycznego jest możliwość automatycznego wnioskowania na podstawie zgromadzonej wiedzy.

Proste zapytanie o podwładnych pracownika, który jest zatrudniony na etacie 'President', zwraca oczekiwany wynik:

```
SQL> SELECT n
FROM TABLE( SEM_MATCH('( ?m :managerOf ?n) (?m rdf:type :President)',
    SEM_MODELS( 'Employees' ), null, SEM_ALIASES(
        SEM_ALIAS( '', 'http://www.ploug.org.pl/emp/' )), null));
N
-----
http://www.ploug.org.pl/emp/Jones
http://www.ploug.org.pl/emp/Blake
http://www.ploug.org.pl/emp/Clark
```

ale wystarczy tylko wskazać, że cecha managerOf jest tranzytywna:

```
SQL> INSERT INTO emp_rdf VALUES (8, SDO_RDF_TRIPLE_S('Employees',
    'http://www.ploug.org.pl/emp/managerOf',
    'rdf:type',
    'owl:TransitiveProperty'));
```

i włączyć wnioskowanie, aby wynik stał się zupełnie inny:

```
SQL> SELECT n
FROM TABLE( SEM_MATCH('( ?m :managerOf ?n) (?m rdf:type :President)',
    SEM_MODELS( 'Employees' ), SDO_RDF_RULEBASES( 'OWLPRIME' ),
    SEM_ALIASES( SEM_ALIAS( '', 'http://www.ploug.org.pl/emp/' ), null));
N
-----
http://www.ploug.org.pl/emp/Blake
http://www.ploug.org.pl/emp/Jones
http://www.ploug.org.pl/emp/Clark
http://www.ploug.org.pl/emp/Smith
http://www.ploug.org.pl/emp/Turner
http://www.ploug.org.pl/emp/Martin
http://www.ploug.org.pl/emp/Miller
http://www.ploug.org.pl/emp/Allen
http://www.ploug.org.pl/emp/Adams
http://www.ploug.org.pl/emp/Ward
http://www.ploug.org.pl/emp/Scott
http://www.ploug.org.pl/emp/James
http://www.ploug.org.pl/emp/Ford
```

Innym przykładem obrazującym siłę danych semantycznych jest możliwość określenia cechy jako relacji zwrotnej. Przykładowo, baza wiedzy zawiera informacje o tym, którzy pracownicy się ze sobą znają w postaci predykatu knows. Poniżej przedstawiono przykładowy fakt, deklarację zwrotności relacji (jeśli x zna y to y zna x) oraz zapytanie odczytujące pary pracowników, którzy się znają.

```
-- Allen zna Turnera
SQL> INSERT INTO emp_rdf VALUES (9, SDO_RDF_TRIPLE_S('Employees',
    'http://www.ploug.org.pl/emp/Allen',
    'http://www.ploug.org.pl/emp/knows',
    'http://www.ploug.org.pl/emp/Turner'));
```

```
-- knows jest cechą
SQL> INSERT INTO emp_rdf VALUES (10, SDO_RDF_TRIPLE_S('Employees',
    'http://www.ploug.org.pl/emp/knows',
    'http://www.w3.org/1999/02/22-rdf-syntax-ns#type',
    'http://www.w3.org/1999/02/22-rdf-syntax-ns#Property'));
```

```
-- knows jest cechą symetryczną
SQL> INSERT INTO emp_rdf VALUES (11, SDO_RDF_TRIPLE_S('Employees',
  'http://www.ploug.org.pl/emp/knows',
  'rdf:type',
  'owl:SymmetricProperty'));

-- znajdź pary pracowników postaci X zna Y
SQL> SELECT m,n
FROM TABLE( SEM_MATCH('( ?m :knows ?n)',
  SEM_MODELS('Employees'), SDO_RDF_RULEBASES('OWLPRIME'),
  SEM_ALIASES(SEM_ALIAS('', 'http://www.ploug.org.pl/emp/'), null));
M
N
-----
http://www.ploug.org.pl/emp/Allen    http://www.ploug.org.pl/emp/Turner
http://www.ploug.org.pl/emp/Turner   http://www.ploug.org.pl/emp/Allen
```

Kolejny przykład pokazuje wykorzystanie ontologii do uspoźnienia danych. W przykładowej bazie wiedzy informacja o płacy pracownika jest przedstawiona za pomocą własności <http://www.ploug.org.pl/emp/salary>, ale pracownik Smith posiada płacę reprezentowaną za pomocą własności <http://www.ploug.org.pl/emp/pay>. Taka rozbieżność może z łatwością wynikać np. na skutek importu danych z zewnętrznego źródła, niespójności w terminologii, itp. Tradycyjna baza danych nie potrafiłaby poprawnie uwzględnić pracownika Smith w zapytaniu dotyczącym atrybutu *salary* ponieważ krótka opisująca Smitha posiadałaby inny schemat (zamiast atrybutu *salary* występowałby atrybut *pay*). W semantycznej bazie danych wystarczy wzbogacić ontologię o informację, że w rzeczywistości *salary* i *pay* to to samo:

```
SQL> INSERT INTO emp_rdf VALUES (12, SDO_RDF_TRIPLE_S('Employees',
  'http://www.ploug.org.pl/emp/salary',
  'owl:equivalentProperty',
  'http://www.ploug.org.pl/emp/pay'));

```

aby zapytanie o pracowników zarabiających powyżej \$3000 dawało pożądane rezultaty:

```
SQL> SELECT e, s
2  FROM TABLE( SEM_MATCH('( ?e :salary ?s)',
3    SEM_MODELS('Employees'), SDO_RDF_RULEBASES('OWLPRIME'),
4    SEM_ALIASES(SEM_ALIAS('', 'http://www.ploug.org.pl/emp/'),
5      null, null))
6  WHERE s > 3000;

E
-----
http://www.ploug.org.pl/emp/King      5000
http://www.ploug.org.pl/emp/Lansky    8000
http://www.ploug.org.pl/emp/Smith     8000
S
```

Oracle Semantic Technologies nie ogranicza użytkownika do predefiniowanego zbioru reguł wnioskowania zapisanego w konkretnym języku RDFS/OWL. Użytkownik może z łatwością definiować swoje własne reguły i wykorzystywać je do wnioskowania. Poniższy przykład ilustruje utworzenie bazy reguł i zdefiniowanie reguły definiującej cechę *WorksTogetherWith* jako konsekwencji tego, że dwoje pracowników pracuje w tym samym departamencie.

```
SQL> EXECUTE SEM_APIS.CREATE_RULEBASE('EmpRules');

SQL> INSERT INTO mdsys.semr_EmpRules VALUES ('WorksTogetherWithRule',
  '(?x :worksIn ?z) (?y :worksIn ?z)', null,
  '(?x :worksTogetherWith ?y)',
  SEM_ALIASES(SEM_ALIAS('', 'http://www.ploug.org.pl/emp/')));

```

```
SQL> SELECT x,y
2 FROM TABLE( SEM_MATCH('( ?x :worksTogetherWith ?y)',
3 SEM_MODELS('Employees'), SDO_RDF_RULEBASES('RDFS','EmpRules'),
4 SEM_ALIASES(SEM_ALIAS('','http://www.ploug.org.pl/emp/')), null,null))
5 WHERE (x != y);
```

X	Y
http://www.ploug.org.pl/emp/Smith	http://www.ploug.org.pl/emp/Jones
http://www.ploug.org.pl/emp/Smith	http://www.ploug.org.pl/emp/Adams
http://www.ploug.org.pl/emp/Smith	http://www.ploug.org.pl/emp/Scott
http://www.ploug.org.pl/emp/Smith	http://www.ploug.org.pl/emp/Ford
http://www.ploug.org.pl/emp/Jones	http://www.ploug.org.pl/emp/Smith
http://www.ploug.org.pl/emp/Jones	http://www.ploug.org.pl/emp/Adams
...	

Na koniec pokazano, w jaki sposób dane semantyczne mogą posłużyć do wzbogacenia tradycyjnych danych relacyjnych. W poniższym przykładzie zdefiniowano ontologię zawierającą prostą taksonomię lokalizacji departamentów: miasta Nowy Jork i Boston leżą w regionie Wschód, podczas gdy miasta Chicago i Dallas leżą w regionie Zachód. Korzeniem taksonomii jest region US obejmujący oba regiony.

```
-- Nowy Jork leży w regionie Wschód
SQL> INSERT INTO emp_rdf VALUES (13, SDO_RDF_TRIPLE_S('Employees',
'http://www.ploug.org.pl/emp/NewYork',
'http://www.w3.org/2000/01/rdf-schema#subClassOf',
'http://www.ploug.org.pl/emp/East'));

-- region Wschód należy do regionu US
SQL> INSERT INTO emp_rdf VALUES (14, SDO_RDF_TRIPLE_S('Employees',
'http://www.ploug.org.pl/emp/East',
'http://www.w3.org/2000/01/rdf-schema#subClassOf',
'http://www.ploug.org.pl/emp/US'));

...
```

Tak definiowana ontologia może być wykorzystana do znalezienia wszystkich pracowników pracujących w departamentach należących do regionu Zachód (należy zwrócić uwagę, że w klauzuli FROM występują zwykłe tabele z danymi relacyjnymi a wzbogacenie o ontologię odbywa się tylko poprzez wykorzystanie funkcji SEM_RELATED()).

```
SQL> SELECT ename, dname, loc FROM emp NATURAL JOIN dept
2 WHERE SEM_RELATED(loc,
3 '<http://www.w3.org/2000/01/rdf-schema#subClassOf>',
4 '<http://www.ploug.org.pl/emp/West>',
5 SEM_MODELS('Employees'), SEM_RULEBASES('OWLPRIME')) = 1;
```

ENAME	DNAME	LOC
SMITH	RESEARCH	<http://www.ploug.org.pl/emp/Dallas>
ALLEN	SALES	<http://www.ploug.org.pl/emp/Chicago>
WARD	SALES	<http://www.ploug.org.pl/emp/Chicago>
JONES	RESEARCH	<http://www.ploug.org.pl/emp/Dallas>
MARTIN	SALES	<http://www.ploug.org.pl/emp/Chicago>
BLAKE	SALES	<http://www.ploug.org.pl/emp/Chicago>
SCOTT	RESEARCH	<http://www.ploug.org.pl/emp/Dallas>
TURNER	SALES	<http://www.ploug.org.pl/emp/Chicago>
ADAMS	RESEARCH	<http://www.ploug.org.pl/emp/Dallas>

JAMES	SALES	< http://www.ploug.org.pl/emp/Chicago >
FORD	RESEARCH	< http://www.ploug.org.pl/emp/Dallas >

11 rows selected.

Powyższy przykład pokazuje, że Oracle Semantic Technologies nie zmuszają do przepisania wszystkich danych do postaci semantycznej. Wzbogacenie danych relacyjnych o prostą semantykę (np. taksonomię pojęć i bytów) umożliwia pisanie znacznie bardziej zaawansowanych i elastycznych zapytań przy minimalnej ingerencji w dane źródłowe (w przykładzie konieczne było powiązanie poszczególnych departamentów z zasobami ontologii poprzez odnośniki URI).

5. Uwagi końcowe

Technologie semantyczne nie są już domeną kilku wizjonerów z W3C. Semantyczny model danych, który ma wielkie szanse zrewolucjonizować sposób przetwarzania informacji w sposób podobny do rewolucji wprowadzonej przez model relacyjny w latach 70-tych XX wieku, staje się powoli częścią komercyjnych rozwiązań. Baza danych Oracle 11g jest jednym z pierwszych dużych systemów wspierających dane semantyczne w całej rozciągłości. Oczywiście, patrząc na stos technologii semantycznych, wciąż jeszcze jest to początek drogi, aspekty związane z zaufaniem, reputacją, zaawansowaną logiką wywodzenia wiedzy, czy inteligentnymi agentami nadal są technologią zbyt niedojrzałą do praktycznych zastosowań. Należy jednak pamiętać, że sieć semantyczna rozwija się oddolnie i każda kolejna warstwa zdobywa akceptację po rozprzestrzenieniu się warstwy niższej. Dziś nie sposób zaprzeczyć, że standardy takie jak UNICODE, XML czy XML Schema stały się nieodłącznymi elementami krajobrazu informatycznego. W chwili obecnej obserwuje się coraz powszechniejszą akceptację modelu semantycznego jako modelu składowania danych, dotyczy to przede wszystkim języków RDF i RDFS. Kolejnym krokiem będzie upowszechnienie się ontologii jako uniwersalnych sposobów opisu modelowanych fragmentów świata rzeczywistego. Stąd, niedaleko już do porywającej wizji Tim Berners-Lee w której Internet jest gigantyczną składnicą całej wiedzy ludzkości, a maszyny i ludzie mogą z tej wiedzy swobodnie korzystać w celu automatyzacji wykonywania postawionych im zadań. Oracle Semantic Technologies to narzędzie przybliżające tę wizję.

Bibliografia

- Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). The Semantic Web. *Scientific American*.
- Berners-Lee, T., Karger, D. R., Stein, L. A., Swick, R. R., & Weitzner, D. J. (2000, February 4). *PROPOSAL: Semantic Web Development*. Pobrano z lokalizacji W3C: <http://www.w3.org/2000/01/sw/DevelopmentProposal>
- Cost, S., Finin, T., & Joshi, A. (2002, February). A Case Study in the Semantic Web and DAML+OIL. *IEEE Intellignet Systems*, pp. 40-47.
- Euzenat, J. (2002). An infrastructure for formally ensuring interoperability in a heterogeneous semantic web. W I. F. Cruzetal, *The emerging semantic web* (strony 245-261). IOS Press.
- Group, W. U. (2001, September). *URIs, URLs, and URNs: Clarifications and Recommendations 1.0*. Retrieved from W3C: <http://www.w3.org/TR/uri-clarification/>
- Herman, I. (2008, October). *Semantic Web Activity Statement*. Retrieved from W3C: <http://www.w3.org/2001/sw/Activity>
- Horst, H. (2005). Completeness, decidability and complexity of entailment for RDF Schema and an intensional variant of OWL. *Journal of Web Semantics*, strony 79-115.
- Kopaliński, W. (2007). *Słownik wyrazów obcych i zwrotów obcojęzycznych*. Rytm.
- N-Triples*. (brak daty). Pobrano z lokalizacji W3C: <http://www.w3.org/2001/sw/RDFCore/ntriples/>
- Passin, T. B. (2004). *Explorer's Guide to the Semantic Web*. Greenwich, CT: Manning Publications Co.

W3C. (2008, November). *Extensible Markup Language*. Pobrano z lokalizacji W3C: <http://www.w3.org/TR/xml/>

W3C. (2004, February). *OWL Web Ontology Language Overview*. Pobrano z lokalizacji W3C: <http://www.w3.org/TR/owl-features/>

W3C. (2004, February). *RDF Vocabulary Description Language 1.0: RDF Schema*. Pobrano z lokalizacji W3C: <http://www.w3.org/TR/rdf-schema/>

W3C. (2004, February). *RDF/XML Syntax Specification*. Pobrano z lokalizacji W3C: <http://www.w3.org/TR/rdf-syntax-grammar/>

W3C. (2008, January). *SPARQL Query Language for RDF*. Pobrano z lokalizacji W3C: <http://www.w3.org/TR/rdf-sparql-query/>

W3C. (2004, May). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. Pobrano z lokalizacji W3C: <http://www.w3.org/Submission/SWRL/>

W3C. (2004, October). *XML Schema Part 0: Primer Second Edition*. Pobrano z lokalizacji W3C: <http://www.w3.org/TR/xmlschema-0/>