Budowa ontologii usług dla potrzeb wyszukiwania.

Boiński T.

10 października 2010

Streszczenie. Ontologie, dzięki zapewnieniu formalnego opisu przy zachowaniu czytelności dla człowieka, są coraz powszechniej stosowaną metodą opisu usług sieciowych. Dawniej postrzegane jako rozwiązanie problemu heterogeniczności sieci, obecnie poprzez mnogość organizacji je wytwarzających same ją wprowadzają. Pojawia się więc konieczność by aplikacje korzystające z usług sieciowych potrafiły automatycznie dostosować opis usług do swoich potrzeb. W publikacji zaprezentowano słownik WordNet i jego zastosowanie jako meta ontologię w łączeniu rozwiązań różnych dostawców. Zaproponowano algorytm oparty o słownik WordNet umożliwiający łączenie ontologii opisujących różne usługi w celu zapewnienia interoperacyjności rozwiązań dostępnych w sieci Internet.

Słowa kluczowe. ontologia, odwzorowywanie ontologii, opis usług sieciowych

1 Ontologie jako metoda opisu usług sieciowych

Usługi sieciowe charakteryzują się szeregiem parametrów takich jak format danych wejściowych i wyjściowych, lokalizacja, czas działania, koszt uruchomienia itp. Proces automatycznego doboru usług wymaga by opis ten był dostępny w jednolitej formie zrozumiałej dla komputera. Jedną z metod opisu usług sieciowych jest zastosowanie w tym celu ontologii. Rozwiązanie to cechuje się dużym poziomem formalizacji poprzez zgodność z logiką pierwszego rzędu na poziomie zapewniającym wnioskowalność [1]. Zachowana jest ponadto czytelność dla człowieka ułatwiając tym samym definiowane takiego opisu.

Utworzenie jednego wspólnego opisu dostępnych usług zachowuje jego spójność i uniezależnia środowisko wykonawcze od samego opisu. Wymaga jednak stałego rozwijania i utrzymywania tak utworzonej „książki telefonicznej". Pociąga to za sobą problemy z aktualnością danych. Każda modyfikacja usługi, dodanie nowej czy usunięcie już nieaktywnej wymaga aktualizacji rejestru usług. Taki centralny indeks staje się też wąskim gardłem. W sytuacji, gdy usług jest dużo nieustanne dostępy do ich opisów rodzą problemy wydajnościowe – zachodzi konieczność replikacji danych, synchronizacji serwerów itp.

Idealnym rozwiązaniem byłoby przechowywanie jedynie informacji o położeniu opisu usługi wyrażonego za pomocą formalnego języka i dostarczonego przez autora usługi. Dzięki temu po jednorazowej rejestracji możliwe jest aktualizowanie i rozwijanie usług bez konieczności każdorazowego aktualizowania

centralnego rejestru. Dobrym kandydatem na formalny język opisu są ontologie poprzez dobrze opisane struktury oraz istniejące standardy porządkujące język opisu.

Pomimo, że ontologie są postrzegane jako rozwiązanie dla heterogeniczności danych w Internecie, mnogość podmiotów będących dostawcami usług sama w sobie wprowadza heterogeniczność. Organizacja W3C podejmuje próby standaryzacji opisu danych za pomocą ontologii, jednak obecnie polega to jedynie na określaniu języka zapisu. Jednym z takich języków jest OWL [1], obecnie najpowszechniejszy język służący do opisu ontologii. Sytuację dodatkowo pogarsza brak ustalonych i powszechnie akceptowanych standardów budowy ontologii, takich jak np. nazewnictwo klas czy atrybutów.

Jednym z zaproponowanych wyjść z tej sytuacji jest tworzenie odwzorowań jednych ontologii w inne, czyli tworzenie powiązań między klasami, rolami oraz relacjami różnych ontologii bez ingerencji w ich konstrukcję. Niestety proces ręcznego tworzenia odwzorowań szybko rośnie do rangi problemu niemożliwego do wykonania ze względu na ilość istniejących rozwiązań. Ponadto każda zmiana jednej z ontologii wymagałaby modyfikacji wszystkich istniejących odwzorowań. Z kolei automatyzacja tego procesu napotyka na trudności ze względu na różnice w postrzeganiu świata przez różnych twórców. Komunikując się z usługą sieciową opisaną pewną ontologią, nie ma żadnych gwarancji, że świat systemu nawiązującego połączenie, jest opisany podobnie nawet pomimo zastosowania tego samego języka opisu świata. Wysoce prawdopodobne jest wręcz, że używa on zupełnie odmiennej terminologii, poziomu dokładności struktury semantycznej czy liczby i rodzaju atrybutów klas. System realizujący proces w sposób automatyczny musiałby być w stanie wywnioskować znaczenie elementów obu ontologii (swojej i usługi) a następnie wyszukać elementy najbardziej sobie odpowiadające. Pożądane jest również by system był w stanie łączyć elementy ontologii w jedną całość lub dokonywać ich faktoryzacji. W przeciwnym wypadku łączenie rozwiązań o różnym poziomie szczegółowości byłoby znacznie utrudnione.

Drugim możliwym podejściem jest łączenie ontologii, w wyniku którego powstaje nowa ontologia zawierająca wszystkie elementy łączonych rozwiązań. Napotkane w tym przypadku problemy są bardzo podobne do wprowadzanych przez odwzorowywanie ontologii. Ponownie pojawia się problem dopasowania klas obu ontologii i zrewidowania ich wzajemnych zależności. W miejsce zbioru połączeń produktem jest tutaj nowa ontologia. Powoduje to jednak generowanie coraz większych i coraz bardziej skomplikowanych ontologii, które nie muszą reprezentować punkt widzenia wszystkich zainteresowanych środowisk.

Pomimo tych trudności wiele zespołów podjęło prace w kierunku zarówno odwzorowywania jak i łączenia. Wyróżnić można dwa główne podejścia do tego problemu. Jedno z nich zakłada, że procesowi tworzenia ontologii towarzyszy mniej lub bardziej świadoma myśl, iż w przyszłości będą one częścią większej całości. Podejście to zakłada istnienie pewnej meta ontologii, która będzie rozszerzana przez wszystkie inne ontologie. Wzajemne odwzorowanie dwu ontologii w takiej sytuacji sprowadzałoby się jedynie do stworzenia odwzorowań do meta ontologii. Podjęto kilka prób utworzenia jednej dużej meta ontologii takie jak SUMO [2] czy DOLCE [3]. Podejścia te są niestety jedynie propozycjami a nie zaakceptowanymi standardami. Nie jest również pewne czy w ogóle możliwe jest utworzenie jednej dużej ontologii, która byłaby w stanie zapewnić odpowiedni poziom szczegółowości nawet w obrębie jednej dziedziny problemowej.

MOST WIEDZY Downloaded from mostwiedzy.pl

Drugie podejście zakłada pełną lub częściową automatyzację procesu tworzenia odwzorowań pomiędzy dwiema ontologiami. Zakłada ono leksykalną i/lub semantyczną analizę obu ontologii w celu znalezienia najlepszego dopasowania klas obu ontologii. W dalszej części tej pracy zajmować będziemy się tylko tym podejściem.

2 Przegląd metod odwzorowywania i łączenia ontologii

Tematyka odwzorowania ontologii jest aktywnie badana przez wielu grup badawczych. Dotychczas powstało wiele rozwiązań, lecz większość z nich jest tylko częściowo zautomatyzowana. To użytkownikowi zazwyczaj pozostawia się ostateczną decyzje o kształcie odwzorowania. W tej części rozdziału omówione zostało kilka z wypracowanych metod rozwiązujących przedstawiony wcześniej problem.

Hovy [4] zaproponował zbiór algorytmów heurystycznych opierających się głównie o wyszukiwanie podobieństw lingwistycznych między konceptami oraz analizie leksykalnej opisów konceptów w języku naturalnym. Algorytm mapowania przebiega następująco:

1. Rozdzielenie elementów nazw konceptów
2. Porównywanie podciągów nazw konceptów
3. Porównywanie liczby i podobieństwa słów wspólnych w opisach konceptów w języku naturalnym

Wynikiem działania algorytmu są sugerowane powiązania pomiędzy ontologiami, które następnie muszą zostać skorygowane i zatwierdzone przez człowieka.

W swoim podejściu Euzenat i Volchev [5] analizują również strukturę ontologii zapisanych w języku OWL. W tej propozycji miarą rozwiązania jest średnia ważona pomiędzy podobieństwami takich elementów definicji konceptu jak typ czy etykieta konceptu, dziedzina, zakres oraz ograniczenia właściwości, taksonomia itp.

Innym możliwym podejściem jest to reprezentowane przez GLUE [8]. System ten do poszukiwania odwzorowań stosuje techniki uczenia się maszyn. Różne algorytmy uczące analizują informacje zawarte w instancjach konceptów oraz w taksonomii ontologii. Wyniki te są następnie scalane za pomocą technik probabilistycznych. System ten najlepiej sprawuje się dla ontologii posiadających dużą liczbę instancji oraz gdy atrybuty wypełnione są tekstowymi etykietami a nie odnośnikami do innych instancji.

Jako jedni z niewielu w swoich rozważaniach Zhao i Halang [9] zauważają konieczność istnienia pewnego podobieństwa między odwzorowywanymi ontologiami. Autorzy odwzorowanie oraz podobieństwo ontologii opierają o teorię zbiorów przybliżonych [10] oraz teorię przestrzeni konceptualnej [11], w wyniku czego powstało rozwiązanie rozbudowane matematycznie i bardzo silne z formalnego punktu widzenia.

Połączenie wielu reguł podobieństwa zaproponowali Ehrig i Sure [12]. Tezą ich pracy jest stwierdzenie, iż zestawienie wielu, wprowadzonych ręcznie przez

MOST WIEDZY Downloaded from mostwiedzy.pl

człowieka reguł podobieństwa konceptów da lepsze odwzorowanie jednej ontologii w drugą niż zastosowanie prostych warunków. Rozwiązanie to zapewnia praktycznie automatyczne odwzorowanie oraz łączenie ontologii na podstawie jednorazowo wprowadzonego przez ekspertów zestawu reguł.

Powyżej przedstawiono jedynie kilka reprezentatywnych podejść do zagadnienia łączenia i odwzorowywania ontologii. Pomimo dużej ich różnorodności, istniejące obecnie rozwiązania są niewystarczające. Brakuje algorytmów oceny podobieństwa ontologii a priori a także algorytmów, które mogły by być z powodzeniem zastosowane przez różnego rodzaju systemy komputerowe dla dowolnej pary ontologii. Złożoność i różnorodność języka naturalnego, stosowanego do opisu poszczególnych elementów ontologii, pozostaje nadal nierozwiązanym problemem zwłaszcza, gdy przystępujemy do analizy rozwiązań utworzonych w dwu lub więcej językach. W przypadku ontologii monolingwistycznych rozwiązania takie jak WordNet [13] pozwalają na sprawne określanie relacji między zastosowanymi opisami konceptów.

3 Model ontologii

Autor proponuje przyjęcie definicji ontologii wynikającej z połączenia podejścia Hovy'ego [4] oraz Euzenata i Volcheva [5]. Ontologia rozumiana jest jako zbiór klas i bytów powiązanych ze sobą relacjami dziedziczenia i przynależności. W skład ontologi wchodzą zatem następujące elementy:

- klasa - jest to koncept agregujący elementy świata rzeczywistego, posiadający wspólny zbiór atrybutów (ale o nieokreślonej wartości) zdefiniowanych w tej klasie, np. klasa algorytmów szyfrujących opartych o algorytm RSA. Przyjęto również występowanie specjalnej klasy THING będącej klasą, po której dziedziczą wszystkie inne klasy, oraz do której przynależą wszystkie byty w ontologii. Dzięki temu analizowany graf jest zawsze grafem spójnym, odpowiada owl:Thing z języka OWL,

- byt - reprezentuje konkretny element świata rzeczywistego posiadający skonkretyzowane wartości parametrów, np. algorytm szyfrujący oparty o algorytm RSA posiadający klucze o długości 1024 bity,

- komentarz - jest to dodatkowy opis słowny przypisany do klasy lub bytu niosący dodatkową informację o znaczeniu danego konceptu,

- relacja dziedziczenia - relacja ta może zachodzić pomiędzy dwiema klasami, umożliwia agregacje klas w bardziej ogólne koncepty. Mówimy, że klasa X dziedziczy po klasie Y, jeżeli każdy element klasy X jest również elementem klasy Y. Klasa dziedzicząca posiada wszystkie właściwości zdefiniowane w klasie dziedziczonej, jest to też relacja przechodnia, np. klasa algorytmów szyfrujących opartych o algorytm RSA dziedziczy po klasie algorytmów szyfrujących asymetrycznych,

- relacja przynależności - relacja ta może zachodzić pomiędzy bytem a klasą, umożliwia powiązanie bytu z klasą definiująca właściwości tego bytu, np. algorytm szyfrujący oparty o algorytm RSA posiadający klucze o długości 1024 bity przynależy do klasy algorytmów szyfrujących opartych o algorytm RSA.

MOST WIEDZY Downloaded from mostwiedzy.pl

W celu zachowania siły wyrazu wprowadzono dodatkowe relacje:

- tożsamości - oznaczającą, że dwie klasy lub dwa byty są identyczne,
- rozłączności - oznaczającą, że dwie klasy lub dwa byty są różne,
- unii - relacja zachodząca tylko pomiędzy klasami, mówiąca, że klasa będąca wynikiem unii jest złączeniem klas wchodzących w jej skład, czyli element x należy do unii klas jeżeli należy do przynajmniej jednej z klas tworzących unię,
- przecięcia - relacja zachodząca tylko pomiędzy klasami, mówiąca, że klasa będąca wynikiem przecięcia jest częścią wspólną klas wchodzących w jej skład, czyli element x należy do przecięcia klas jeżeli należy do każdej z klas tworzących unię.

Ontologie zgodne z powyższym modelem zapisane zostaną w języku OWL, a zawarte w nich dane tekstowe (nazwy klas i bytów, komentarze) będą w języku angielskim.

4 WordNet jako meta-ontologia

WordNet jest leksykalną bazą danych utworzoną i rozwijaną oryginalnie dla języka angielskiego. W odróżnieniu od tradycyjnych słowników operuje on na znaczeniach słów (tzw. *synset*) a nie na samych słowach. Synset oznacza tutaj grupę słów będących synonimami. Różne znaczenia danego słowa zawsze ujęte są w różnych synsetach. Synsety są pomiędzy sobą powiązane relacjami, których typ uzależniony jest od rodzaju słowa. WordNet wyróżnia rzeczowniki, czasowniki, przymiotniki i przysłówki. Dzięki temu stworzono sieć koncepcji powiązanych ze sobą znaczeniowo.

W WordNet ujęto następujące relacje:

- rzeczowniki:
  - nadrzędność (hypernim) - Y jest hypernimem X jeżeli każdy X jest typu Y,
  - podrzędność (hyponim) - Y jest hyponimem X jeżeli każdy Y jest typu X,
  - typy skoordynowane (ang. *coordinate terms*) - Y jest skoordynowany z X jeżeli X i Y mają wspólny hypernim,
  - holonim - Y jest holonimem X jeżeli X jest częścią Y,
  - meronim - Y jest meronimem X jeżeli Y jest częścią X,
- czasowniki:
  - nadrzędność (hypernim) - Y jest hypernimem X jeżeli czynność X jest typu Y,
  - troponim - Y troponimem X jeżeli czynność Y jest szczególnym przypadkiem czynności X,
  - zawieranie - czasownik Y zawiera się w X jeżeli wykonując czynność X musimy wykonać czynność Y,

MOST WIEDZY Downloaded from mostwiedzy.pl

  - typy skoordynowane (ang. *coordinate terms*) - czasowniki współdzielące hypernim,

- przymiotniki:
  - relacyjność,
  - podobieństwo,
  - imiesłowy,

- przysłówki:
  - wspólny rdzeń przymiotnikowy.

Dzięki takiej strukturze WordNet z powodzeniem może być wykorzystany do analizy języka naturalnego. Dzięki implementacji relacji pomiędzy słowami możliwa jest nie tylko analiza leksykalna ale również i semantyczna. O sile tego rozwiązania świadczy mnogość implementacji algorytmów określających podobieństwo słów [14][15] czy określania znaczenia słów [16][17].

Struktura powiązań pomiędzy słowami zdefiniowana w słowniku WordNet umożliwia wiązanie ze sobą pojęć pomimo zastosowania odmiennej bazy leksykalnej. Zdolność taka jest niezastąpiona przy porównywaniu konceptów opisujących różne spojrzenie na ten sam element świata rzeczywistego, zwłaszcza w połączeniu z obserwacją Zhao i Halanga o konieczności istnienia pewnego podobieństwa między łączonymi ontologiami. Zaobserwowano jednak, że trakcie procesu łączenia i odwzorowywania ontologii istotne jest nie tyle podobieństwa samych ontologii co ich elementów [19]. Można sobie wyobrazić dwie ontologie o zupełnie różnym poziomie szczegółowości lub pokrywające się tematycznie tylko w pewnym stopniu. Wyszukując zbiory elementów podobnych można przeprowadzić proces ich integracji.

5 Zastosowanie słownika WordNet przy odwzorowywaniu ontologii

Opisane w poprzedniej części właściwości słownika WordNet umożliwiają opracowanie mechanizmu odwzorowywania i łączenia ontologii. Zaobserwowano ponadto, że wiele ontologii pojęcia reprezentujące koncepty wyraża w postaci rzeczowników a większość informacji zawartych w ontologii wyrażona jest za pomocą klas i relacji dziedziczenia zachodzących pomiędzy nimi. Dziedzinowość z kolei zwiększa prawdopodobieństwo występowania wspólnej bazy koncepcyjnej służącej do opisu danego wycinka świata [19]. Dzięki temu proponowane podejście, przede wszystkim na podstawie związków między słowami zdefiniowanymi w słowniku WordNet, ale także dzięki leksykalnej i strukturalnej analizie łączonych ontologii, umożliwia dynamiczne tworzenie zintegrowanej ontologii łączącej wiedzę zawartą w ontologiach wejściowych.

W niniejszej pracy zaproponowano algorytm w postaci pseudokodu odwzorowywania ontologii (Alg. 1). Jest to algorytm rekurencyjny o pesymistycznej złożoności obliczeniowej wynoszącej $O(mn)$, gdzie n to rozmiar, czyli liczba klas i bytów, ontologii wejściowej A, m to rozmiar ontologii wejściowej B. Rozpoczynając pracę algorytm pobiera na wejście dwie ontologie wejściowe A i B (linie 42 oraz 43). Wyjściem algorytmu jest ontologia wynikowa C, tworzona w trakcie

MOST WIEDZY Downloaded from mostwiedzy.pl

jego pracy (linia 44) oraz zwracana po jego zakończeniu (linia 49). Bazując na założeniu, że w ontologiach opisanych w języku OWL istnieje zawsze wspólny korzeń owl:Thing jednakowy dla wszystkich konceptów pobieramy go dla każdej z ontologii (linie 45-47) i przeprowadzamy operację łączenia poddrzew węzła owl:Thing ontologii A z węzłem owl:Thing ontologii B, a całość zapisujemy jako poddrzewo węzła owl:Thing ontologii C (linia 48). Algorytm opiera się o dwie funkcje:

- łączącą poddrzewa wskazanych węzłów (linie 2-22),
- umieszczającą węzeł jednej ontologii w drzewie drugiej ontologii tak, by jego bezpośrednim rodzicem był węzeł zawierający element uogólniający a potomkami węzły bardziej szczegółowe – o ile takie istnieją (linie 23-42).

Obie te funkcje są bardzo podobne w działaniu, szczegółowo omówiona zostanie pierwsza z nich. Jako parametry wejściowe przyjmowane są trzy wartości:

- węzeł spinający poddrzewo ontologii A,
- węzeł spinający poddrzewo ontologii B,
- węzeł ontologii C, do której ma być podpięty wynik złączenia drzew będących pozostałymi parametrami funkcji.

Algorytm porównuje elementy na tym samym poziomie w strukturze drzewa na zasadzie każdy z każdym (pętle zaczynające się w liniach 3 i 4). W zależności od wyniku tego porównania wykonywana jest jedna z następujących akcji:

1. w przypadku gdy pojęcia są identyczne (linie 6-10) tworzone jest pojęcie wspólne (linia 7), dodawane do ontologii wyjściowej (linia 8) a następnie poddrzewa wyznaczone przez te dwa węzły są ze sobą łączone poprzez rekurencyjne wywołanie funkcji (linia 9),
2. w przypadku, gdy pojęcia są całkowicie rozłączne (linie 10-13) są one wraz ze swoimi poddrzewami w całości przepisywane do ontologii wynikowej,
3. w przypadku, gdy obecnie porównywany węzeł z ontologii A jest bardziej ogólny w swoim znaczeniu niż węzeł ontologii B (linie 13-16) to węzeł ontologii A jest dodawany do ontologii wynikowej (linia 14) a węzeł ontologii B jest lokowany w drzewie o korzeniu będącym analizowanym węzłem ontologii A (linia 15). Lokowanie to odbywa się za pomocą bliźniaczej funkcji opisanej poniżej,
4. w przypadku, gdy obecnie porównywany węzeł z ontologii A jest bardziej szczegółowy w swoim znaczeniu niż węzeł ontologii B (linie 16-19), operacje wykonywane są analogiczne do punktu 3, jednak tym razem do ontologii wyjściowej dopisywany jest węzeł ontologii B a lokowany węzeł ontologii A.

Druga z funkcji w swoim działaniu jest bardzo podobna. Zamiast łączyć dwa drzewa znajduje najlepsze położenie dla wskazanego węzła jednej ontologii w poddrzewie drugiej ontologii. Różni się rodzajem oraz liczbą pobieranych parametrów wejściowych:

- węzeł, który ma być ulokowany w poddrzewie,

MOST WIEDZY Downloaded from mostwiedzy.pl

Algorytm 1 Pseudokod prezentujący pracę algorytmu łączenia ontologii

```
1: program OntologyMerger {
2: function combineSubTrees(Node root_A, Node root_B, Node root_C) {
3: for all child_A : root_A.getChildren() do
4:   for all child_B : root_B.getChildren() do
5:     int result = compareConcepts(child_A, child_B);
6:     if result == EQUAL then
7:       Node node = combineConcepts(child_A, child_B);
8:       root_C.addChild(node);
9:       combineSubTrees(child_A, child_B, node);
10:    else if result == DISJOINT then
11:      root_C.addChildWithSubTree(child_A);
12:      root_C.addChildWithSubTree(child_B);
13:    else if result == A_MORE_GENERAL_THAN_B then
14:      root_C.addChild(child_A);
15:      placeNodeInSubTree(child_B, child_A);
16:    else if else if (result == B_MORE_GENERAL_THAN_A then
17:      root_C.addChild(child_B);
18:      placeNodeInSubTree(child_A, child_B);
19:    end if
20:  end for
21: end for
22: }
23: function placeNodeInSubTree(Node node, Node root) {
24: for all child : root.getChildren() do
25:   int result = compareConcepts(node, child);
26:   if result == EQUAL then
27:     Node node = combineConcepts(node, child);
28:     root_C.addChild(node);
29:     combineSubTrees(node, child);
30:   else if result == DISJOINT then
31:     root_C.addChildWithSubTree(child);
32:     root_C.addChildWithSubTree(node);
33:   else if result == NODE_MORE_GENERAL_THAN_CHILD then
34:     root_C.addChild(node);
35:     placeNodeInSubTree(child, node);
36:   else if result == CHILD_MORE_GENERAL_THAN_NODE then
37:     root_C.addChild(child);
38:     placeNodeInSubTree(node, child);
39:   end if
40: end for
41: }
42: input OWLOntology Ontology_A;
43: input OWLOntology Ontology_B;
44: output OWLOntology Ontology_C;
45: Node root_A = Ontology_A.getOWLThing();
46: Node root_B = Ontology_B.getOWLThing();
47: Node root_C = Ontology_C.getOWLThing();
48: combineSubTrees(root_A, root_B, root_C);
49: return  Ontology_C;
50: }
```


MOST WIEDZY Downloaded from mostwiedzy.pl

- węzeł spinający drzewo, w którym ulokowany ma być węzeł będący pierwszym parametrem funkcji.

Kluczowe znaczenie ma tutaj procedura porównywania elementów ontologii (compareConcepts(child_A, child_B)). Mogą one być porównywane na trzy sposoby: semantycznego, leksykalnego oraz strukturalnego. Funkcja ta w pierwszej kolejności bada podobieństwo semantyczne pomiędzy konceptami pozwalające szczegółowo określić rodzaj zależności pomiędzy pojęciami. W przypadku, gdy badanie podobieństwa semantyczne nie powiedzie się, możliwe jest rozstrzygnięcie identyczności lub rozłączności konceptów na postawie równania:

$$P = w_1 P_l + w_2 P_s + w_3 P_o$$

gdzie:
P_l - podobieństwo leksykalne etykiet porównywanych elementów,
P_s - podobieństwo strukturalne wynikające z podobieństwa typów i liczności relacji w jakich znajdują się porównywane elementów,
P_o - podobieństwo leksykalne komentarzy przypisanych do porównywanych elementów,
w_i - wagi nadane poszczególnym podobieństwom dobrane na podstawie przeprowadzonych badań, wyjściowo przyjmują one wartości: $w_1 = 0,7$; $w_2 = 0,25$; $w_3 = 0.05$.

Wartość prawdopodobieństwa P jest liczbą z przedziału $< 0; 1 >$. Poziom p podobieństwa stanowiący o podobieństwie elementów nie jest obecnie znany. Wyjściowo przyjmuje się, że wartość podobieństwa $p = 0.7$ oznacza elementy identyczne. Dla $p < 0.7$ badane koncepty są uznawane za różne. Wartość ta wynika z badań innych badaczy, podobną wartość przyjmuje np. narzędzie Falcon-AO [18]. Przeprowadzone badania pozwolą na dokładne określenie tej wartości.

6 Miary podobieństwa elementów ontologii

Zaproponowano 4 miary podobieństwa elementów ontologii – podobieństwo semantyczne jako miarę główną oraz leksykalne oparte o nazwę konceptu, leksykalne oparte o komentarz przypisany do konceptu oraz strukturalne. Poniżej przedstawiono wskazane miary:

1. Podobieństwo semantyczne klas i bytów – jest bazową formą podobieństwa pomiędzy konceptami. Do porównywania elementów ontologii używany jest semantyczny słownik WordNet. Dzięki zdefiniowanym w nim relacjom nadrzędności i podrzędności pomiędzy słowami możliwe jest określenie wzajemnego zawierania się znaczeń porównywanych konceptów. Porównując dwie klasy ze sobą na podstawie semantycznego podobieństwa opisujących je etykiet można określić ich wzajemną relację - czy są sobie równoważne, rozłączne lub jedna z nich zawiera się logicznie w drugiej, czyli ustalić hierarchię dziedziczenia klas i przynależności bytów opisanych danymi konceptami.

2. Podobieństwo leksykalne klas i bytów – w przypadku, gdy określenie podobieństwa semantycznego jest niemożliwe algorytm wspiera swoje działanie za pomocą podobieństwa leksykalnego. Porównywane są etykiety słów i na

MOST WIEDZY Downloaded from mostwiedzy.pl

podstawie skali podobieństwa, reprezentowanej przez liczbę z przedziału od 0 do 1, określone zostanie podobieństwo słów. W tym celu stosowane będą algorytmy JCn [14] oraz Lin [15]. Charakteryzują się one korelacją zbliżoną do granicznej, wynoszącej 0,885, uzyskując odpowiednio 0,828 oraz 0,834. Za pomocą tych algorytmów możliwe jest sprawne wyznaczenie podobieństwa leksykalnego porównywanych elementów. Przeprowadzone badania pozwolą zweryfikować poprawność ich działania a także określić, który z nich lepiej zachowuje się przy operowaniu na zwrotach wykorzystywanych w ontologiach z dziedziny bezpieczeństwa.

3. Podobieństwo leksykalne komentarzy do klas i bytów – Kolejnym atrybutem pozwalającym określić podobieństwo dwóch elementów ontologii jest tożsamość komentarza jednego z elementów z komentarzem drugiego. Podobieństwo to opisuje się wzorem:

$$P_o = a_1 * P_{op} + a_2 * 0.5 * (P_{z1} + P_{z2})$$

gdzie:
P_{op} - podobieństwo ciągów tekstowych będących komentarzem do analizowanych konceptów,
P_{z1} - podobieństwo etykiety pierwszego elementu (klasy bądź bytu) do komentarza drugiego elementu,
P_{z2} - podobieństwo etykiety drugiego elementu do komentarza pierwszego elementu,
a_i - wagi poszczególnych podobieństw, wyjściowo wynoszą one $a_1 = a_2 = 0.5$.

Wagi poszczególnych składowych tego podobieństwa będą obiektem przeprowadzonych badań. Wyjściowo składowym tego podobieństwa nadano identyczne wagi. Należy tutaj zauważyć, że obecne algorytmy wyznaczające znaczenie słów mają niską jakość. Stąd należy to podobieństwo traktować uzupełniająco jako niosące dodatkową informację, lecz o znaczeniu drugorzędnym.

4. Podobieństwo strukturalne klas i bytów – opisuje w jakim stopniu badane elementy są osadzone wśród innych je otaczających. Istotny tutaj jest rodzaj, kierunek oraz ilość relacji w jakich uczestniczą oba elementy. Podobieństwo te można wyrazić wzorem:

$$P_s = \frac{\sum_i min(r_i)}{\sum_i max(r_i)}$$

gdzie: r_i - ilość wystąpień i-tej relacji.

Podobieństwo te gra szczególną rolę przy rozwiązywaniu konfliktów. Większe podobieństwo strukturalne umożliwia rozróżnienie elementów podobnych leksykalnie na podstawie związków z innymi elementami ontologii czy też posiadanych atrybutach.

MOST WIEDZY Downloaded from mostwiedzy.pl

7 Podsumowanie

System komputerowy oparty o ontologie nie może sprawnie współpracować z innymi systemami bez wiedzy o konstrukcji ontologii tych systemów. Integracja ontologii opisujących różne systemy umożliwi ich interoperacyjność bez konieczności lub tylko przy nieznacznej modyfikacji ich konstrukcji - za pomocą pojęć znanych jednemu z systemów możliwe będzie odniesienie się do pojęć znanych drugiemu systemowi.

Łączenie ontologii, tworząc w wyniku nową zawierającą wszystkie elementy łączonych rozwiązań, ma na celu zapewnienie interoperacyjności wspomnianych systemów. Obecnie dowolność reprezentacji świata rzeczywistego w systemach komputerowych powoduje, że systemy różnych dostawców napotykają problemy w wymianie informacji. Proces ręcznego tworzenia odwzorowań szybko rośnie do rangi problemu niemożliwego do wykonania ze względu na ilość istniejących rozwiązań oraz ich zmienność. Automatyzacja tego procesu, pomimo napotykanych trudności, wynikających z różnic w postrzeganiu świata przez rożnych twórców, umożliwia współpracę systemów opartych o usługi działającym w dynamicznym otoczeniu bez konieczności modyfikacji systemów i usług. Zaprezentowany algorytm umożliwia dynamiczne uzgadnianie mechanizmów współpracy pomiędzy systemami i usługami umożliwiając tym samym dynamiczną kompozycję usług w skomplikowane aplikacje.

Literatura

[1] W3C Recommendation, OWL Web Ontology Language Guide, http://www.w3.org/TR/owl-guide/, 2004

[2] Niles, I., Pease, A., 2001, Towards a Standard Upper Ontology, Teknowledge Corporation, USA

[3] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., 2003, Sweetening WordNet with Dolce, American Association for Artificial Intelligence, USA

[4] Hovy, E., 1998, Combining and standardizing largescale, practical ontologies for machine translation and other uses, In The First International Conference on Language Resources and Evaluation (LREC), pp. 535–542, Granada, Spain

[5] Euzenat, J., Valtchev, P., 2004, Similarity-based ontology alignment in OWL-Lite, In The 16th European Conference on Artificial Intelligence (ECAI-04), Valencia, Spain

[6] Noy, N., 2004, Semantic Integration: A Survey Of Ontology-Based Approaches, Stanford Medical Infomatics, Stanford University, USA

[7] Kalfoglou, Y., Schorlemmer, M., 2003, Ontology mapping: the state of the art, The Knowledge Engeneering Review, 18(1):1–31

[8] Doan, A., Madhavan, J., Domingos, P., Halevy, A., Learning to map between ontologies on the semantic web. In The Eleventh International WWW Conference, Hawaii, US, 2002.

MOST WIEDZY Downloaded from mostwiedzy.pl

[9] Zhao, Y., Halang, W., Rough Concept Lattice based Ontology Similarity Measure, FernUniversitaet, 58084 Hagen, Germany

[10] Pawlak, Z., Rough Sets, International Journal of Information and Computer Science, 1982, pp. 341–356

[11] Ganter, B., Wille, R., Formal Concept Analysis: Mathematical Foundations, Springer-Verlag, New York, 1999

[12] Ehrig, M., Sure, Y., Ontology mapping - an integrated approach, Proceedings of the First European Semantic Web Symposium , volume 3053 of LNCS , pp. 76–91, Heraklion, Greece, MAY 2004. Springer Verlag.

[13] Fellbaum, Ch., WordNet: An Electronic Lexical Database, Cambridge, Massachussets, USA, The MIT Press, 1998

[14] Jiang J. and Conrath D., Semantic similarity based on corpus statistics and lexical taxonomy, Proceedings of International Conference Research on Computational Linguistics (ROCLING X), 1997, Taiwan.

[15] Lin D., An information-theoretic definition of similarity, Proceedings of the 15th International Conference on Machine Learning, 1998

[16] Banerjee S., Adapting the Lesk Algorithm for Word Sense Disambiguation to WordNet, 2002

[17] Banerjee S. and Pedersen T., An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet, Computational Linguistics and Intelligent Text Processing, pp. 117–171, Springer, 2002

[18] Jian N., Hu W., Cheng G., Qu Y., Falcon-AO: Aligning ontologies with falcon, In Integrating Ontologies Workshop Proceedings, Citeseer, 2005

[19] Boiński T., Orłowski P., Szpryngier P., Krawczyk H., Influence and selection of basic concepts on ontology design, International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Valencia, Spain, 2010 (w publikacji)

MOST WIEDZY Downloaded from mostwiedzy.pl