

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

WEWNĘTRZNA REPREZENTACJA KONGLOMERATOWEJ BAZY WIEDZY W SYSTEMIE RKASEA

Streszczenie. Niniejszy artykuł prezentuje koncepcję wewnętrznej reprezentacji wiedzy ontologicznej zastosowaną w nowo opracowanym systemie zarządzania wiedzą ontologiczno-regulową RKASEA. Reprezentacja ta stanowi praktyczne wdrożenie opublikowanej wcześniej koncepcji modularyzacji bazy wiedzy polegającej na jej podziale na semantyczne jednostki – konglomeraty. Opracowana metoda reprezentacji stanowi zarazem rozwinięcie wprowadzonej wcześniej metody kartograficznej.

Słowa kluczowe: bazy wiedzy, logika opisowa, *Semantic Web*, system wnioskujący, wiedza, terminologia, zapytania terminologiczne

INTERNAL REPRESENTATION OF S-MODULAR KNOWLEDGE BASE IN RKASEA SYSTEM

Summary. This article presents a novel method of internal knowledge representation. The method has been exploited in newly created RKASEA system and is a practical implementation of the recently published concept of s-modular knowledge bases. The method is also a development of earlier idea of knowledge cartography.

Keywords: knowledge bases, description logics, *Semantic Web*, inference system, knowledge, terminology, terminological queries

1. Wstęp

Przed dwoma laty grupa KMG z Politechniki Gdańskiej stanęła przed zadaniem zaimplementowania nowego systemu zarządzania wiedzą ontologiczną wyrażoną za pomocą logiki opisowej. System ten miał pozwalać na obsługę bardziej ekspresywnych konstrukcji ontolo-

gicznych, a jednocześnie dawać użytkownikowi większą elastyczność w manipulowaniu poszczególnymi częściami bazy wiedzy.

Punktem startowym był tu system KASEA, który oferował znaczne możliwości, jeśli chodzi o wnioskowanie z baz wiedzy o znacznej liczbie osobników. KASEA wykorzystywała w tym celu metodę reprezentacji wiedzy nazwaną metodą kartograficzną [1]. Mimo wielu zalet tej metody, wprowadzało to do systemu szereg ograniczeń, m.in. skutkowało tym, że obsługa ontologii o dużej liczbie słabo powiązanych konceptów mogła być bardzo nieefektywna (ze względu na zbyt dużą długość sygnatury, patrz sekcja 3).

System KASEA oferował także użytkownikowi możliwość podziału bazy wiedzy na moduły zwane, według wykorzystywanej metody SIM [2], kontekstami. Możliwość tej metody były duże, jednak utworzony podział na moduły był stosunkowo statyczny, a jakkolwiek jego zmiana mocno utrudniona. Oznaczało to, że użytkownik musiał bardzo precyzyjnie określić podział na konteksty w przewidywaniu przyszłych zastosowań, co nie zawsze było proste.

Na przełomie lat 2008 i 2009 pojawiła się idea konglomeratowej bazy wiedzy [3]. U źródeł tej idei leży przekonanie, że zawartością bazy wiedzy można manipulować w sposób tak samo elastyczny, jak zawartością relacyjnej bazy danych (za pomocą zapytań SQL i algebry relacji). Dzięki opracowaniu formalizmu możliwe stało się precyzyjne określenie, jak rozumiemy jednostkę wiedzy oraz zorganizowania przetwarzania tych jednostek w sposób przypominający ten, który znamy z relacyjnych baz danych.

W modelu konglomeratowym kluczowe jest pojęcie jednostki wiedzy, którą operujemy, czyli konglomeratu. Konglomerat rozumiany jest ściśle semantycznie – mówiąc nieco nieformalnie, jako zbiór możliwości, które nie zostały wykluczone przez napływające do bazy wiedzy informacje. Z jednej strony ta postać konglomeratu umożliwia utworzenie zbioru działań na konglomeratach bardzo przypominającego algebrę relacyjną. Z drugiej strony jednak rodzi problemy związane z reprezentowaniem konglomeratów.

W niniejszym artykule zajmiemy się zagadnieniem reprezentacji konglomeratów. Pokażemy opracowane rozwinięcie metody kartograficznej, które pozwoliło na usunięcie znacznej części jej mankamentów, a także zaprezentujemy algorytmy, jakie zostały wykorzystane do realizacji przetwarzania konglomeratowej bazy wiedzy w zaimplementowanym przez grupę KMG nowym systemie zarządzania wiedzą RKASEA.

2. Algebra konglomeratów

Niniejsza sekcja prezentuje podstawy formalizmu konglomeratów. Więcej szczegółów na ten temat zainteresowany Czytelnik odnajdzie w pozycjach [3] i [4].

W modelu konglomeratowym przyjmuje się, że baza wiedzy jest zorganizowana w moduły (fragmenty), które mogą być przez użytkownika przetwarzane i łączone w celu otrzymania nowych jednostek wiedzy. Użytkownik dysponuje zaawansowanym aparatem pozwalającym na dostosowanie wybranych fragmentów bazy wiedzy do wymaganej postaci. Aparat ten obejmuje działania algebry konglomeratów, czyli działania na fragmentach bazy wiedzy wzorowane na operacjach algebry relacyjnej Codda [5].

Pojęcie konglomeratu, czyli podstawowej jednostki wiedzy, zdefiniowane jest w sposób ściśle semantyczny jako para (S, W) , gdzie S jest słownikiem (zbiorem terminów), a W zbiorem dopuszczalnych interpretacji (modeli konglomeratu), z których każda przypisuje fragment dziedziny zainteresowań Δ (jej podzbiór lub zbiór krotek elementów dziedziny) każdemu terminowi.

Pojęcie konglomeratu jest w zasadzie niezależne od wykorzystywanego języka, ale przyjmujemy tu, że do interakcji z bazą wiedzy wykorzystujemy pewien dialekt logiki opisowej¹ \mathcal{L} . Oznacza to, że terminy ze słownika S to przede wszystkim nazwy konceptów (które można rozumieć jako predykaty unarne; zbiór nazw konceptów w S oznaczamy $C(S)$) i ról (odpowiedników predykatów binarnych; ich nazwy oznaczamy $R(S)$).

Semantyczna definicja konglomeratu pozwoliła na zdefiniowanie w ramach omawianego modelu zestawu działań na konglomeratach bardzo przypominającego algebrę relacyjną. Operacje te obejmują projekcję (π ; zawężającą słownik), przemianowanie (ρ ; zamieniające znaczenie nazw w słowniku) selekcję (σ ; ograniczającą ilość interpretacji²), sumę, przecięcie i różnicę konglomeratów (\cap , \cup , $-$; są to operacje binarne na zbiorach interpretacji).

Semantyczna definicja konglomeratu, a także dobór działań algebry, pozwalają na *sprawne* (w znaczeniu opisanym w [7]; ang. *robust*) wykonywanie podstawowych (patrz [7]) czynności zmiany słownika modułu, rozszerzenia słownika i złączenia modułów, zaś w rzeczywistości stanowi naturalne teoriozbiorowe domknięcie minimalnego zestawu operacji pozwalającego na wykonanie tych działań. Dodatkowo, jak pokazano w [4], algebra konglomeratów jest wariantem algebry cylindrycznej [8], a prawa tej ostatniej algebry, a także prawa algebry relacji Codda obowiązują również dla algebry konglomeratów.

Operacje algebry konglomeratów mogą posłużyć do wyprodukowania nowego konglomeratu z istniejących, według życzeń i potrzeb użytkownika. Na bazie tej algebry utworzono język KQL, służący do wydawania poleceń bazie wiedzy i pozwalający użytkownikowi na wykorzystanie działań algebraicznych w podobny sposób, w jaki czyni się to w SQL-u (więcej na temat języka KQL i jego związków z SQL-em piszemy w publikacji [9]). Uzyskuje się dzięki temu elastyczność przetwarzania wiedzy porównywalną z elastycz-

¹ Zakładamy tutaj znajomość podstaw logiki opisowej przez Czytelnika. Więcej informacji na jej temat można znaleźć w wielu opracowaniach, m.in. w [6].

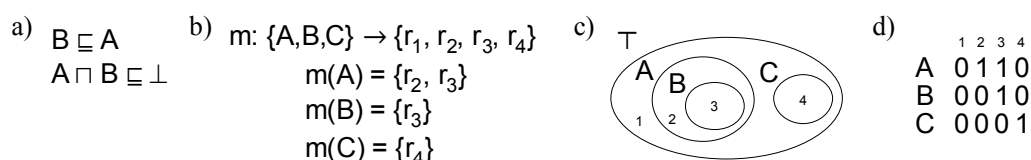
nością przetwarzania danych oferowaną przez silniki relacyjne, a możliwości użycia raz utworzonej bazy wiedzy mogą wyjść znacznie poza zakres przewidywanych zastosowań.

3. Metoda kartograficzna

Ze względu na semantyczną strukturę konglomeratu naturalny wydaje się wybór takiej metody jego reprezentacji, która skupia się na opisywaniu przestrzeni dostępnych interpretacji. Taką metodą jest *kartografia wiedzy*, której podstawy przypominamy w niniejszej sekcji. Metoda ta posiada jednak pewne niedostatki, które sprawiają, że wymagała ona rozwinięcia, co jest przedmiotem dyskusji podsumowującej tę część artykułu.

Metoda kartograficzna powstała w 2004 roku, inspirowana oryginalnie potrzebami projektu PIPS. Projekt ten (finansowany w ramach 6. Programu Ramowego UE) zakładał przechowywanie dużych ilości semantycznie opisanych informacji na tematy związane ze zdrowiem i zdrowym stylem życia.

Metoda kartograficzna służy do reprezentowania ontologii zapisanych w logice opisowej. Organizuje ona koncepty w tzw. *mapy konceptów*. Mapa konceptów przyporządkowuje każdemu konceptowi pewną liczbę *regionów*, czyli, formalnie rzecz ujmując, jest odwzorowaniem $m: C(S) \rightarrow S_R$, gdzie S to pewien narzucony słownik, a S_R to zbiór regionów. Zależności pomiędzy zbiorami regionów (nakładanie się, zawieranie, rozłączność) odpowiadają dokładnie zależnościom pomiędzy zakresami reprezentowanych na mapie konceptów. Mapę można graficznie przedstawiać w postaci diagramu Venna, co zilustrowano na rys. 1.



Rys. 1. Przykład mapy konceptów (b) reprezentującej terminologię (a) zapisanej graficznie (c) i binarnie (d)

Fig. 1. An example of a map of concepts (b) representing terminology (a) in the graphical (c) and in the binary notation (d)

Dzięki metodzie kartograficznej można w stosunkowo prosty i intuicyjny sposób przedstawić skomplikowane zależności pomiędzy grupami konceptów. W ramach prac nad metodą utworzono szereg algorytmów służących m.in. do tworzenia mapy konceptów na podstawie aksjomatów (zdań) logiki opisowej. Dzięki umieszczaniu na mapie konceptów postaci $\exists R.C$ możliwe było także odwzorowanie na mapie ontologicznych (wprowadzonych przez użytkownika) zależności pomiędzy konceptami i rolami.

² Warto wspomnieć, że selekcja jest jedynym operatorem, w którym wykorzystujemy zdania języka \mathcal{L} . Zdania te stanowią odpowiednik θ -ograniczeń z algebry relacyjnej.

Jedno z rozszerzeń metody kartograficznej polegało na wprowadzeniu obsługi tzw. *dziedzin konkretnych* (patrz także [10]). Dziedzinami konkretnymi określa się w logice opisowej przestrzenie wartości atrybutów zawierających wartości proste, np. numeryczne i tekstowe. Rozszerzenie metody kartograficznej polegało na wprowadzeniu dodatkowej mapy dla każdej z obsługiwanych dziedzin. Miejsce konceptów zajęły tam tzw. koncepty konkretne, czyli zakresy wartości odpowiednich dla danej dziedziny (np. zakresy postaci $[x, y]$ dla dziedziny liczb całkowitych). Atrybuty obsługiwane były w sposób podobny jak role, z zastrzeżeniem, że ich dziedziny i zakresy przyporządkowane były odpowiednim mapom.

Metoda kartograficzna wykazała wiele zalet. Przy swojej stosunkowej intuicyjności okazała się także praktycznie realizowalna, rozszerzalna i dająca duże możliwości obsługi baz wiedzy zawierających duże liczby osobników.

Jednakże metoda kartograficzna w swojej oryginalnej postaci wprowadzała wiele ograniczeń. Opracowane algorytmy budowy mapy konceptów nie pozwalały na dynamiczną przebudowę mapy w trakcie pracy, co oznaczało, że użytkownik musiał z góry wprowadzić całą część terminologiczną ontologii i nie mógł jej później aktualizować. Dodatkowo kartograficzna reprezentacja wiedzy zastosowana bezpośrednio polegała na przechowywaniu tzw. *sygnatur* konceptów i osobników, czyli ciągów binarnych o długości równej wyróżnionej na mapie liczbie regionów. Oznaczało to, że stosunkowo łatwo było utworzyć ontologię niereprezentowalną w praktyce. Przykładowo, ontologia postaci $\{C_i \sqsubseteq \top\}$, $i \in [1 \dots 50]$ generowała sygnatury o długości 2^{50} bitów, co sprawiało, że metoda przestawała być użyteczna w praktyce.

4. Grafowo-kartograficzna reprezentacja konglomeratów

4.1. Założenia metody reprezentacji

Zawarta w poprzedniej sekcji dyskusja cech metody kartograficznej pozwala na wyciągnięcie wniosku, iż pomimo dużej atrakcyjności metody (w szczególności skupieniu się na reprezentowaniu semantycznego aspektu ontologii), jej podstawowa wersja nie jest dobrze dostosowana do reprezentacji konglomeratów. W szczególności założenie o niezmienności terminologii stoi w sprzeczności z ideą wykorzystania konglomeratów jako dynamicznie przekształcanych fragmentów ontologii, których postać dostosowywana jest do specyficznych potrzeb użytkownika. Wniosek ten skłonił autorów niniejszego artykułu do podjęcia próby opracowania nowej, rozszerzonej wersji metody kartograficznej posiadającej następujące cechy:

- możliwość dynamicznej aktualizacji konglomeratu; algorytmy budowy struktury reprezentacyjnej powinny być dostosowane do możliwości częstej zmiany również terminologicznej części bazy wiedzy;
- możliwość wykonania operacji algebraicznych na konglomeratach; zestaw algorytmów powinien być rozszerzony o algorytmy pozwalające na uzyskanie konglomeratów będących wynikiem podstawowych operacji algebraicznych;
- większe możliwości przechowywania informacji o luźno powiązanych grupach terminów; wykonanie zapytań KQL może prowadzić do złamania (obowiązującego dla oryginalnej wersji metody kartograficznej) założenia o tym, że większe fragmenty terminologii to taksonomie (rozumiane jako hierarchie konceptów rozłącznych [11]). Z tego względu rozwinięcie metody powinno oferować większe możliwości praktyczne w zakresie przechowywania zależności między luźniej powiązаныmi konceptami;
- zwiększona ekspresywność zależności ontologicznych; założenia nowego projektu pociągały za sobą konieczność zwiększenia ekspresywności wykorzystywanego dialektu logiki opisowej (oryginalna metoda kartograficzna przeznaczona była dla dialektu \mathcal{ALC}).

4.2. Podstawy grafowo-kartograficznej reprezentacji wiedzy

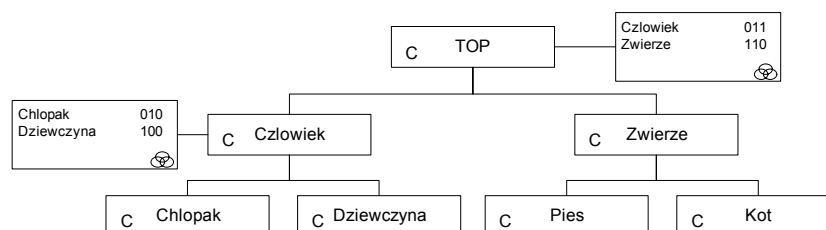
W niniejszej pracy przedstawiamy nową metodę reprezentacji wiedzy, dostosowaną do przechowywania informacji potrzebnych w konglomeratowej bazie wiedzy. Koncentrujemy się tu na wiedzy terminologicznej, czyli zależnościach pomiędzy konceptami i rolami. Ze względu na dodatkowe założenia projektu, takie jak konieczność obsługi reguł, reprezentacja wiedzy na temat osobników, czyli asercyjnej części ontologii, została także poddana znacznej rewizji. Przyjęto tutaj podejście oparte na koncepcji tzw. źródeł osobników, której szczegóły przedstawiamy w innej pracy [12].

Założenia dotyczące rozwoju metody kartograficznej wymienione w poprzednim punkcie zrealizowano poprzez rozwinięcie idei tzw. *sygnatur hierarchicznych*, zarysowanej w pracy [11]. Idea ta polegała na utworzeniu pewnych obszarów na mapie konceptów, które nie są rozwijane w pełną strukturę regionów. W nowo opracowanej metodzie reprezentacji wiedzy koncepcja wyewoluowała w spojrzenie hierarchiczno-aspektowe, gdzie dodatkowo nie są rozwijane (bez wyraźnej potrzeby) obszary odpowiadające wspólnym częściom zakresów niepowiązanych konceptów.

Zasadniczą formę reprezentacji konglomeratu stanowią drzewa odzwierciedlające hierarchię podrzędności konceptów, ale także innych terminów (atrybutów, ról, konceptów konkretnych). W szczególnych przypadkach, kiedy użytkownik wyrazi zależności pomiędzy nierozłącznymi konceptami, drzewo przekształca się w graf. Dodatkowo dla wyrażenia

bardziej skomplikowanych powiązań między konceptami wykorzystywane są częściowe mapy konceptów w postaci takiej samej jak w oryginalnej wersji metody.

Dla opracowywanej reprezentacji stworzono notację graficzną. Przykład grafu konceptów zapisanego w tej notacji jest pokazany na rysunku 2.



Rys. 2. Przykład grafu konceptów. Węzły oznaczone są prostokątami z literą C, dołączone mapy prostokątami z trzema kołami, krawędzie symbolizują relację bycia dzieckiem

Fig. 2. An example of a graph of concepts. Rectangles with "C" denote nodes, edges symbolize the child relationship, maps attached to nodes are denoted with three circles

Głównym elementem reprezentacji jest *węzeł terminu* (konceptu, roli lub atrybutu), z którym kojarzemy:

- nazwę terminu lub zbiór nazw terminów równoważnych (o równych zakresach),
- zbiór węzłów dzieci (bezpośrednio potomnych węzłów terminów),
- słownik terminów zawartych w węzłach potomnych i danych w węźle,
- zbiór map bitowych (odpowiednik map konceptów, ale nazwę tę rozszerzamy na struktury pokazujące zależności między innymi terminami – rolami i atrybutami).

Fakt, że węzeł w_2 jest dzieckiem węzła w_1 , oznacza, że terminy zawarte w węźle w_2 są podrzędne (ich zakresy zawierają się) względem tych zawartych w węźle w_1 . Fakt ten w notacji graficznej zaznaczamy, rysując krawędź pomiędzy węzłami. Relacja bycia węzłem potomnym jest domknięciem relacji bycia dzieckiem i, oczywiście, również oznacza podrzędność odpowiednich terminów.

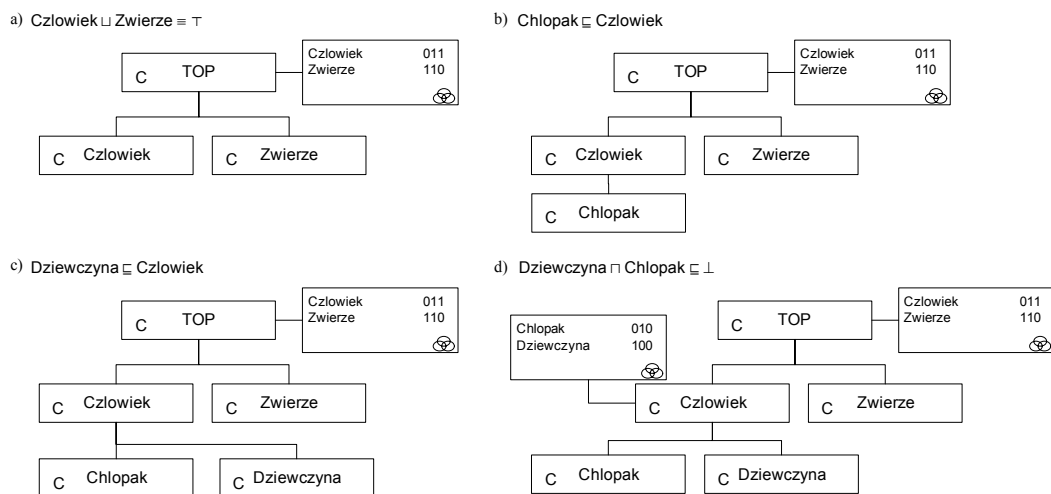
Konglomerat reprezentowany jest poprzez kilka powiązanych ze sobą grafów terminów. Głównym grafem jest *graf konceptów* reprezentujący zależności w podstawowej dziedzinie konglomeratu (Δ). Pozostałe grafy to *graf ról* opisujący powiązania pomiędzy rolami (w ramach dziedziny $\Delta \times \Delta$), *grafy konceptów konkretnych* (dla dziedzin konkretnych Δ^D_i) oraz *grafy atrybutów* (w ramach dziedzin $\Delta \times \Delta^D_i$).

Korzeniem grafu terminów jest zawsze węzeł reprezentujący całą dziedzinę (np. dla grafu konceptów koncept uniwersalny i ewentualnie inne koncepty mu równoważne). Pozostałe węzły są wobec nich potomne. Operacje na grafie konceptów (i dowolnym innym grafie terminów) polegają na przekształcaniu go m.in. poprzez wprowadzanie nowych krawędzi i nowych map.

4.3. Algorytmy dla grafowo-kartograficznej reprezentacji wiedzy

Dla grafów terminów opracowano szereg algorytmów pozwalających na wykonanie na nich podstawowych operacji, takich jak: dodanie aksjomatu (algebraiczna operacja selekcji), usunięcie terminu (projekcja), przecięcie grafów. Ponadto dokonano adaptacji pewnych dodatkowych mechanizmów pochodzących z metody kartograficznej i pozwalających na zachowanie spójności pomiędzy grafem konceptów a grafami ról i atrybutów. Część spośród opracowanych algorytmów przedstawiamy w niniejszej sekcji artykułu; ze względu na ograniczenie miejsca są one przeważnie naszkicowane. O ile nie zaznaczono inaczej, algorytmy dla grafów terminów pokazane są w wersji dla grafów konceptów, ich przetworzenie dla innych grafów jest zazwyczaj bezpośrednie.

Na początek należy zauważyć, że reprezentacja kartograficzna i reprezentacja grafowo-kartograficzna są częściowo zamienne. W ramach pojedynczego grafu terminów możliwe jest przechodzenie od jednej do drugiej formy reprezentacji. Odpowiednie algorytmy (przedstawione niżej) używają pomocniczych pojęć *mapy częściowej* $m|S$ i *mapy znaczącej*. Intuicyjnie³ mapa częściowa, to mapa, z której usunięto terminy spoza S , natomiast mapa znacząca to mapa, której nie da się przedstawić jako złożenie map częściowych⁴, tj. mapa faktycznie przedstawiająca zależności między wszystkimi opisywanymi konceptami.



Rys. 3. Przykładowe kilkukrotne wykonanie operacji selekcji

Fig. 3. An exemplary execution of several consecutive selections

Graf-do-mapy (wejście: graf konceptów G ; wyjście: mapa m)

for każdy węzeł w grafie G **do**

Dołóż do mapy m wszystkie mapy z G

Dołóż do mapy m zależności podrzędności z terminami z węzłów-dzieci

next

³ Formalnie dla $m: C(S) \rightarrow S_R$ mapa częściowa dla $S \subseteq C(S)$ to odwzorowanie $m|S: S \rightarrow S_R'$ takie, że $\forall C \in S: m|S(C) = \mathbf{D}(m(C))$, zaś \mathbf{D} to operacja usunięcia duplikatów — dwa regiony r_1 i r_2 to duplikaty, gdy należą dokładnie do tych samych konceptów, tj. $\{C \in S: r_1 \in m(C)\} = \{C \in S: r_2 \in m(C)\}$.

⁴ Czyli nie istnieją dwie mapy częściowe $m|S_1$ i $m|S_2$, $S_1, S_2 \subset S$ takie, że ich złożenie daje w wyniku oryginalną mapę m .


```

Mapa-do-grafu (wejście: mapa  $m$ ; wyjście: graf konceptów  $G$ )
  Utwórz węzeł dla konceptu uniwersalnego i dla wszystkich konceptów z  $\mathbf{C}(\mathbf{S})$ 
  for każdy niespełnialny koncept  $C$  ze zbioru nazw  $\mathbf{C}(\mathbf{S})$  do
    Podłącz węzeł  $C$  jako dziecko do węzła  $T$ 
    Umieść mapę  $m\{C\}$  w węźle  $T$ 
  next

  for każda para spełnialnych konceptów  $C_1, C_2$  ze zbioru nazw  $\mathbf{C}(\mathbf{S}) \cup \{T\}$  do
    if  $m\{C_1, C_2\}$  wyraża podrzędność  $C_1$  względem  $C_2$  then
      Podłącz węzeł  $C_1$  jako dziecko do węzła  $C_2$ 
    end if
  next

  Usun zależności przechodnie z grafu  $G$ 
  for każdy podzbiór  $S$  słownika  $\mathbf{C}(\mathbf{S})$ , taki że  $|S| > 1$  do
    Utwórz częściową mapę  $m|S$ 
    if wszystkie koncepty z  $S$  są spełnialne i mapa  $m|S$  jest znacząca then
      Umieść mapę  $m|S$  w węzłach  $G$  o najmniejszym słowniku zawierającym  $S$ 
    end if
  next

```

W powyższych algorytmach zakładamy, że operacja usuwania zależności przechodnich dodatkowo sprowadza węzły pozostające w cyklu skierowanym do jednego węzła, a operacja umieszczenia mapy $m: S \rightarrow R$ w węźle w łączy ją (za pomocą operacji złożenia) ze wszystkimi znajdującymi się w w mapami $m': S' \rightarrow R'$ takimi, że $S \cap S' \neq \emptyset$.

Naszkicowane algorytmy teoretycznie pozwalają na bezpośrednie wykorzystanie mechanizmów (np. algorytmów budowy mapy konceptów) opracowanych dla metody kartograficznej. Jednak takie postępowanie byłoby wysoce nieefektywne i nie pozwoliłoby na spełnienie wyznaczonych założeń (między innymi o łatwości zmiany terminologii). Z tego względu dla reprezentacji grafowo-konglomeratowej utworzono dedykowany zestaw algorytmów, wykorzystujących jednak często podobny do użytego w powyższych procedurach schemat postępowania.

Jednym z najważniejszych działań na grafie terminów jest dodanie aksjomatu, czyli algebraiczna operacja selekcji. Algorytm dla tej operacji przedstawiono poniżej:

```

Dodaj-aksjomat (wejście: graf konceptów  $G$ , aksjomat  $\alpha$ ;
                wyjście: uaktualniony graf konceptów  $G$ )
 $S_\alpha :=$  zbiór konceptów wykorzystanych w  $\alpha$ 
  Utwórz mapę  $m$  dla aksjomatu  $\alpha$ 
  Dodaj węzły dla konceptów niezawartych w  $G$ 
  for każdy niespełnialny według  $m$  koncept  $C$  z  $S_\alpha$  do
    Podłącz węzeł  $C$  jako dziecko do węzła  $T$ 
    Umieść mapę  $m\{C\}$  w węźle  $T$ 
  next

  for każda para spełnialnych konceptów  $C_1, C_2$  ze zbioru nazw  $S_\alpha \cup \{T\}$  do
    if  $m\{C_1, C_2\}$  wyraża podrzędność  $C_1$  względem  $C_2$  then
      Podłącz węzeł  $C_1$  jako dziecko do węzła  $C_2$ 
    end if
  next

  Usun zależności przechodnie z grafu  $G'$ 
  for każdy podzbiór  $S$  słownika  $S_\alpha$ , taki że  $|S| > 1$  do
    Utwórz częściową mapę  $m|S$ 
    if wszystkie koncepty z  $S$  są spełnialne i mapa  $m|S$  jest znacząca then
      Umieść mapę  $m|S$  w węzłach  $G$  o najmniejszym słowniku zawierającym  $S$ 
    end if
  next
  (*) Sprawdź spełnialność konceptów podrzędnych względem  $S$ 

```

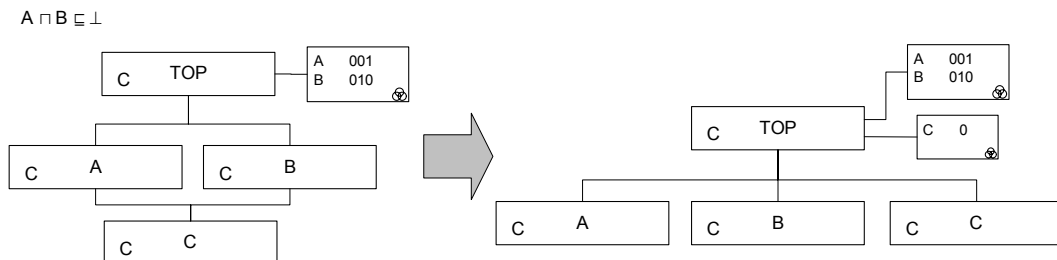
```

end if
next

```

Jak widać, algorytm dokładania aksjomatu jest bardzo podobny do opisanego wcześniej algorytmu przekształcającego mapę konceptów w graf konceptów. Jest to zrozumiałe, gdyż tak naprawdę przyjęto tu założenie, że „dodajemy” informacje zawarte w pewnej mapie do istniejącego grafu G . Działanie algorytmu dla przykładowej serii aksjomatów zilustrowano na rysunku 3. Różnicą w stosunku do procedury *Mapa-do-grafu* jest mechanizm sprawdzania spełnialności konceptów podrzędnych względem wszystkich terminów objętych pewną zależnością opisywaną mapą (wiersz oznaczony (*)). Mechanizm ten jest potrzebny ze względu na fakt, że wprowadzona zależność może oddziaływać na te koncepty, co zilustrowano na przykładzie z rysunku 4. Wprowadzenie informacji o rozłączności konceptów A, B spowodowało niespełnialność konceptu C i, w efekcie, uniezależnienie go od pozostałych konceptów.

Opracowany zestaw algorytmów pozwala na wykonanie operacji projekcji (usunięcia terminu) oraz przecięcia grafów terminów. Z uwagi na brak miejsca nie przytaczamy tu dokładnej postaci algorytmów. Usunięcie terminu polega na usunięciu terminu z węzła lub, jeśli był to jedyny termin w węźle, całego węzła. W tym ostatnim przypadku musi nastąpić przepisanie węzłów dzieci do węzłów rodziców usuniętego węzła i odpowiednia korekta map. Z kolei w operacji przecięcia dokonuje się podobnych działań, jak w przypadku operacji selekcji, ale źródłem dodatkowych węzłów, krawędzi i map jest drugi graf terminów.



Rys. 4. Przykład operacji sprawdzania spełnialności konceptu podrzędnego

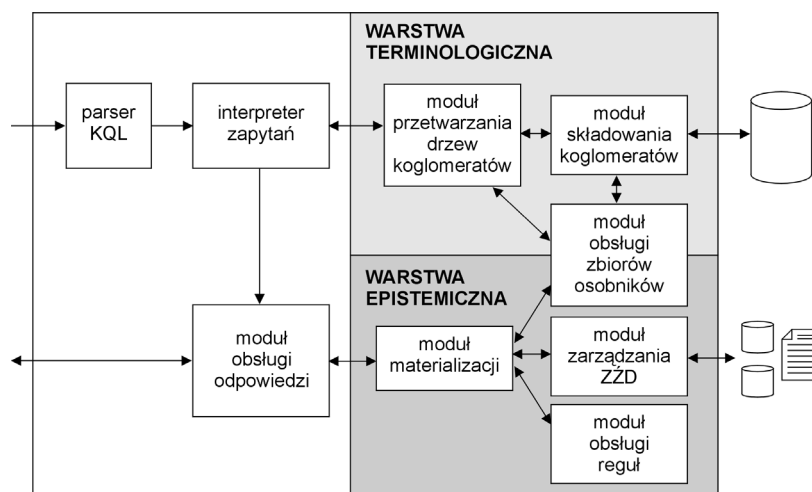
Fig. 4. An example of operation of satisfiability check for a subsumed concept

W ogólnym przypadku informacje na temat wzajemnych zależności w ramach pewnego zbioru terminów S wydobywamy w postaci mapy binarnej (jeśli interesują nas zależności pomiędzy konceptami złożonymi, trzeba je najpierw nazwać, np. wprowadzając aksjomat postaci $A \equiv C$). Mapa ta tworzona jest na podstawie podgrafu grafu G , zawierającego węzły z terminami S i wszystkich przodków. Z tego podgrafu usuwamy terminy spoza G , a następnie wynik zamieniamy na mapę według opisaney wcześniej procedury. Należy zaznaczyć, że w niektórych przypadkach algorytm postępowania można znacznie uprościć,

np. zapytanie o podrzędność dwóch konceptów atomowych można zrealizować przez prostą analizę zależności potomności węzłów grafu konceptów⁵.

Na uwagę zasługuje także kwestia współpracy różnych grafów terminów w ramach reprezentacji pojedynczego konglomeratu. Zagadnienie to zostało potraktowane podobnie jak w oryginalnej metodzie kartograficznej. Tutaj omówimy tylko koordynację zawartości grafów konceptów i ról (dla grafów konceptów i atrybutów proces ten przebiega podobnie). Do grafu konceptów wprowadzane są wszystkie koncepty postaci $\exists R.C$ wykorzystane przez użytkownika w aksjomatach⁶. Koncepty te obsługiwane są tak jak koncepty atomowe. Jednocześnie w drzewie ról tworzone są węzły dla roli R i dla specjalnej roli $R|C$, która obejmuje ten fragment zakresu roli R , dla którego dopełnieniem roli są wystąpienia konceptu C (czyli, bardziej formalnie, $(R|C)^T = \{(a, b) \in R^T : b \in C^T\}$). Utrzymywana dodatkowo jest informacja, że $\exists R.C$ jest dziedziną roli $R|C$. Podobnie jak w przypadku oryginalnej metody kartograficznej stosuje się operację tzw. *postprocessingu*, która ma na celu uzupełnienie zależności pomiędzy konceptami postaci $\exists R.C$ (przykładowo $\exists R.A \sqsubseteq \exists R.B$, jeśli $A \sqsubseteq B$). Warto zaznaczyć, że w sposób naturalny w zastosowanej reprezentacji obsługiwane są hierarchie ról (ekspresywność oznaczana literą \mathcal{H}), ponadto w grafach ról wprowadzane są także role odwrotne (\mathcal{I}).

4.4. Praktyczna realizacja w systemie RKASEA



Rys. 5. Struktura systemu RKASEA

Fig. 5. The structure of RKASEA system

⁵ Należy także zauważyć, że dyskusja ta ma charakter poglądowy, gdyż w systemie RKASEA zapytania terminologiczne zadaje się poprzez wykorzystanie języka KQL [9] i tzw. metaontologii [13]. Sprawia to, że zapytanie terminologiczne staje się zapytaniem o zależności pomiędzy osobnikami.

⁶ Przy czym należy zaznaczyć, że dzięki możliwości dynamicznej zmiany terminologii ograniczenie dotyczące używania konceptów $\exists R.C$ w zapytaniach użytkownika zostało zniesione.

Opisane wcześniej algorytmy zostały zaimplementowane w ramach nowego systemu zarządzania wiedzą RKASEA. System ten daje użytkownikowi możliwości stworzenia i obsługi konglomeratowych baz wiedzy. Językiem, którym posługuje się użytkownik, jest KQL (*Knowledge Query Language*) [9], którego trzon ma bezpośrednie przełożenie na operacje algebry konglomeratów.

Strukturę systemu RKaSeA przedstawiono na rysunku 5. Kwestią reprezentacji konglomeratów i przetwarzania ich części terminologicznej zajmuje się przede wszystkim moduł przetwarzania drzewa konglomeratów nazwany (od pierwszych liter opisu) *eMPeDoKles*. Moduł ten współpracuje z pozostałą częścią systemu, odbierając polecenia od interpretera zapytań, przekazując informacje na temat terminologii do epistemicznej warstwy systemu (zajmującej się obsługą asercyjnej części bazy wiedzy) oraz współpracując z modułem składowania konglomeratów w celu utrwalenia struktur w relacyjnej bazie danych.

W praktycznej realizacji opisanej reprezentacji wykorzystano szereg optymalizacji. Poszczególne węzły grafów terminów są indeksowane względem nazwy. Do operowania mapami w pamięci operacyjnej wykorzystano bibliotekę obsługi drzew OBDD, co pozwoliło m.in. na usprawnienie tworzenia mapy częściowej i procesu badania, czy jest ona znacząca. System skonstruowano tak, że część dotycząca obsługi drzew OBDD jest wymienna, co pozwala na stosowanie różnych dostępnych rozwiązań.

Utworzony system poddano weryfikacji poprzez ładowanie ontologii (opracowano tutaj szereg przypadków testowych) o pewnych typowych strukturach trudnych do obsługi przez oryginalną metodę kartograficzną. Jedne z najważniejszych to:

- wyznaczona liczba n konceptów niepowiązanych ze sobą żadnymi zależnościami: $\{C_i \sqsubseteq \top\}, i \in [1..n]$,
- nierozłączna hierarchia n konceptów: koncepty połączone są zależnościami dziedziczenia, ale nie ma między nimi zadeklarowanych rozłączności.

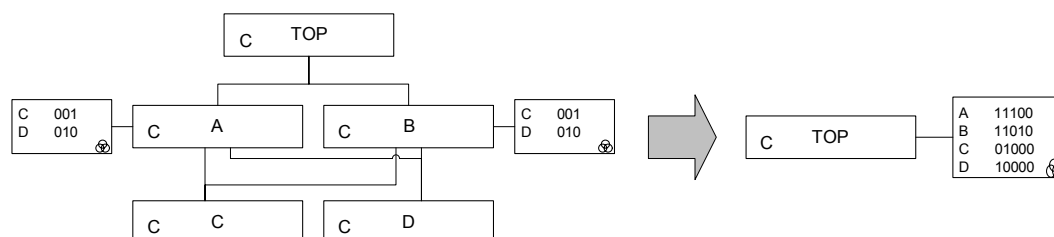
Czas ładowania wymienionych struktur dla $n = 50$ był bardzo krótki (rzędu kilku sekund), m.in. dlatego, że specyfika wykorzystanej reprezentacji sprawia, że ich załadowanie wymaga operacji wyłącznie na grafie. Obie wymienione ontologie były niemożliwe do przetworzenia przez oryginalną metodę kartograficzną ze względu na długość sygnatury.

Wstępne testy wskazują też, że czas ładowania rzeczywistych ontologii nie odbiega znacznie od oferowanego przez system KASEA, co wydaje się wskazywać na znaczny potencjał opracowanej metody i przyjętych heurystyk, gdyż ontologia ładowana jest tutaj na zasadzie dynamicznej aktualizacji terminologii.

5. Podsumowanie

W niniejszym artykule przedstawiono grafowo-kartograficzną metodę reprezentacji wiedzy ontologicznej. Metoda ta stanowi rozwinięcie opracowanej wcześniej kartografii wiedzy i usuwa szereg jej mankamentów, pozwalając m.in. na znacznie prostszą dynamiczną aktualizację terminologii (możliwość wprowadzania nowych aksjomatów), przechowywanie szerszej gamy struktur konceptów i wykorzystanie dowolnych konceptów postaci $\exists R.C$ w zapytaniach użytkownika. Poprzez dekompozycję pełnej mapy konceptów na szereg map binarnych udało się osiągnąć skrócenie sygnatur przy jednoczesnym zachowaniu możliwości zapamiętywania złożonych zależności między terminami oferowanych przez metodę kartograficzną. Reprezentacja grafowo-kartograficzną została uzupełniona szeregiem algorytmów pozwalających na wykonanie najważniejszych działań algebry konglomeratów. Opracowane metody posłużyły do praktycznej realizacji systemu zarządzania wiedzą RKASEA, do zapisu i przetwarzania konglomeratowych bazy wiedzy.

Opracowane mechanizmy wydają się także mieć duży potencjał rozwojowy. Jednym z mankamentów metody jest fakt, że w obecnie przyjętej wersji algorytmów mapy binarne przedstawiające zależności między pewnymi terminami S umieszczane są we wszystkich węzłach nadrzędnych o minimalnym słowniku zawierającym S . W specyficznych przypadkach może to prowadzić do dublowania informacji. Rozwiązaniem tutaj może być przechowywanie większych fragmentów konglomeratu w postaci mapy i dynamiczny dobór tych fragmentów (ideę tę zilustrowano na rys. 6). Innym aktualnie badanym kierunkiem rozwoju jest zwiększenie liczby obsługiwanych operacji algebry konglomeratów, gdyż operacje unii i różnicy nie są w chwili obecnej przetwarzane. Wiąże się to z problemem reprezentowalności ich wyników szerzej opisanym w pracy [15]. W [15] zaproponowano także rozwiązanie, którego bezpośrednia adaptacja polegałaby na duplikowaniu grafów terminów (np. wykonanie unii skutkowałaby powstaniem dwóch grafów konceptów, dwóch grafów ról itp.). Metoda kartograficzna wydaje się jednak mieć większy potencjał zapamiętywania alternatywnych zestawów interpretacji, co w chwili obecnej jest przedmiotem dalszych badań.



Rys. 6. Ilustracja idei przechowywania większych fragmentów konglomeratu w postaci mapy
 Fig. 6. Illustration of the idea of holding larger fragments of conglomerations as binary maps

BIBLIOGRAFIA

1. Goczyła K., Waloszek W., Zawadzka T., Zawadzki M.: The Knowledge Cartography – a New Approach to Reasoning over Description Logics Ontologies. W: SOFSEM 2006: Theory and Practice of Computer Science, red. J. Wiedermann, J. Stuller, G. Tel, J. Pokorný, M. Bielikova, Berlin Heidelberg, Springer-Verlag, 2006 (LNCS 3831), s. 293÷302.
2. Goczyła K., Waloszek A., Waloszek W.: Hierarchiczny podział przestrzeni ontologii na konteksty. Bazy danych. Nowe technologie, red. Kozielski S., Małysiak B., Kasprowski P., Mrozek D., WKŁ, 2007, s. 247÷260.
3. Goczyła K., Waloszek A., Waloszek W.: Algebra konglomeratów jako narzędzie opisu problemów przetwarzania ontologii, *Studia Informatica*, Vol. 30, No. 2A, 2009, s. 141÷156.
4. Goczyła K., Waloszek A., Waloszek W.: Algebra of ontology modules for semantic agents. ICCCI 2009, Wrocław, Polska, Springer-Verlag 2009, s. 492÷503.
5. Codd E.F.: Relational Completeness of Data Base Sublanguages. *Database Systems*, 1979, t. 6, s. 65÷98.
6. Baader F.A., McGuinness D.L., Nardi D., Patel-Schneider P.F. (red.): *The Description Logic Handbook: Theory, implementation, and applications*. Cambridge University Press, 2003.
7. Konev B., Lutz C., Walther D., Wolter F.: Formal Properties of Modularisation. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, red. Parent C., Spaccapietra S., Stuckenschmidt H., Springer, 2009.
8. Henkin L., Monk J.D., Tarski A.: *Cylindric Algebras pt. 1*, *Studies in Logic and the Foundations of Mathematics*, Vol. 64, North-Holland, Amsterdam, 1971.
9. Goczyła K., Piotrowski P., Waloszek A., Waloszek W., Zawadzka T.: Język KQL jako realizacja idei języka SQL dla bazy wiedzy (praca zgłoszona do publikacji).
10. Goczyła K., Waloszek A., Waloszek W.: Wnioskowanie z dziedzin konkretnych w podejściu kartograficznym. Bazy danych. Nowe technologie: Architektura, metody formalne i zaawansowana analiza danych, red. Kozielski S., Małysiak B., Kasprowski P., Mrozek D., WKŁ, 2007, s. 261÷272.
11. Waloszek W.: *Metody strukturalnej analizy ontologii opartych na logice opisowej*. Praca doktorska, Gdańsk, 2008.
12. Goczyła K., Waloszek A., Waloszek W., Zawadzka T.: Wnioskowanie z różnych źródeł osobników w systemie RKaSeA (praca zgłoszona do publikacji).
13. Goczyła K., Waloszek A., Waloszek W., Zawadzka T.: Ideologiczny i praktyczny model metaontologii (praca zgłoszona do publikacji).

Recenzenci: Prof. dr hab. inż. Stanisław Kozielski
Prof. dr hab. inż. Andrzej Świerniak

Wpłynęło do Redakcji 31 stycznia 2010 r.

Abstract

The paper presents and discusses a novel method for representing ontological knowledge expressed in Description Logics (a branch of logic utilized in OWL-DL language employed in Semantic Web initiative). The method may be treated as a vastly revised version of the cartographic method and provides practical means of implementing the theoretical concept of s-modules (semantically modularized knowledge base).

The theory of s-modules has been recently developed and described. To briefly summarize the theory: each s-module is a fragment of a knowledge base which is described by a set of allowed interpretations. These fragments can be combined with use of algebraic operations, which resemble those used in the relational algebra. This gives the user a wide range of methods for manipulating of acquired knowledge.

The presented method allows for representing s-modules (theoretical entities consisting of infinite numbers of interpretations) in the computer memory and for executing transformations of data structures in order to perform algebraic operations. The knowledge being manipulated is stored in the form of recursive trees forming a subsumption hierarchy in which more complex interrelationships between concepts and roles are expressed by *maps of knowledge*.

The notion of maps has been originally used in knowledge cartography. In the initial version of the theory the whole ontology was represented by one map. This could lead to complexity problems, especially in ontologies in which very little has been said about connections between meanings of introduced terms. The new method can therefore be perceived as a developed version of knowledge cartography, in which only a part of ontological knowledge is described by maps, and the number and scope of the maps can be dynamically adjusted to users' needs. In this way many shortcomings of original knowledge cartography have been overcome.

Finally, it has to be mentioned that the method has been deployed in the recently developed knowledge management system RKaSeA. The paper describes and discusses the practical experiences with the method gained as a result of development of the system.

Adres

Krzysztof GOCZYŁA: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Katedra Inżynierii Oprogramowania, ul. G. Narutowicza 11/12, 80-233 Gdańsk, Polska, kris@eti.pg.gda.pl.

Aleksander WALOSZEK: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Katedra Inżynierii Oprogramowania, ul. G. Narutowicza 11/12, 80-233 Gdańsk, Polska, alwal@eti.pg.gda.pl.

Wojciech WALOSZEK: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Katedra Inżynierii Oprogramowania, ul. G. Narutowicza 11/12, 80-233 Gdańsk, Polska, wowal@eti.pg.gda.pl.

Teresa ZAWADZKA: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Katedra Inżynierii Oprogramowania, ul. G. Narutowicza 11/12, 80-233 Gdańsk, Polska, tegra@eti.pg.gda.pl.