

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

SCHEMAT KONGLOMERATOWEJ BAZY WIEDZY¹

Streszczenie. Artykuł opisuje przebieg prac nad rozwojem konglomeratowych baz wiedzy w kierunku opracowania metodyki dla zastosowań inżynierskich. Jednym z pierwszych działań jest podjęcie próby zdefiniowania schematu i instancji bazy. Taki podział jest konieczny, aby zapewnić środki techniczne do ustanowienia ograniczeń związanych z kontrolą przebiegu procesu projektowania bazy oraz jej późniejszego użytkowania.

Słowa kluczowe: bazy wiedzy, konglomeraty, *Semantic Web*, system wnioskujący, modularność

A SCHEMA FOR S-MODULAR KNOWLEDGE BASES

Summary. The paper presents advances of s-modular approach towards developing design methodology for engineers. One of the first proceedings is an attempt to define and differentiate a schema and an instance of a knowledge base. Such distinction is necessary to provide technical means for establishing constraints to control the process of knowledge base design, maintenance and deployment.

Keywords: knowledge bases, s-modules, *Semantic Web*, inference system

1. Wprowadzenie

Teorie tworzone podczas badań naukowych od wieków stanowią podstawę dla różnych metodyk stosowanych w inżynierii. Rolą twórcy metodyki inżynierskiej jest dostarczenie środków i metod, które pozwolą konstruować modele zgodne z daną teorią, ale ściśle odpowiadające danemu typowi sytuacji występujących w świecie rzeczywistym. Zadanie nie jest banalne. Byty teoretyczne mają bowiem to do siebie, że nie mają związku z rzeczywistością

¹ Praca częściowo finansowana z funduszy NCBiR, grant nr SP/I/1/77065/10, projekt SYNAT.

i wobec tego mogą opisywać dowolne sytuacje, w tym również takie, które z punktu widzenia istniejącego świata byłyby absurdalne.

Przykładem popularnej ostatnio teorii, dla której dotąd nie opracowano metodyki inżynierskiej, jest teoria rozproszonych (modularnych, kontekstowych) baz wiedzy. Uwzględnienie modularności baz wiedzy stało się konieczne, gdy podczas prac nad ontologiami w ramach inicjatywy *Semantic Web* stało się jasne, że jest to jedyna droga umożliwiająca racjonalizację wykorzystania posiadanych zasobów. Niemodularne bazy wiedzy okazują się drogie w tworzeniu, trudne w utrzymaniu i w niewielkim tylko stopniu zdadne do powtórnego użycia. Opracowano wiele teoretycznych modeli modularnych baz wiedzy. Jednym z nich jest zaproponowany i rozwijany przez grupę KMG (*Knowledge Management Group*), działającą w ramach Katedry Inżynierii Oprogramowania Wydziału ETI Politechniki Gdańskiej, model konglomeratowy.

Pomimo znacznego postępu prac nad formalną stroną baz wiedzy praktyczny aspekt ich użycia wciąż przedstawia wiele do życzenia. Dlaczego, mimo tak obiecującej formuły, środowiska inżynierskie wciąż wykazują brak zainteresowania omawianymi technikami sztucznej inteligencji? Przyczyn tego stanu rzeczy może być wiele. Jedną z bardziej prawdopodobnych jest fakt, że techniki te nie są wystarczająco „techniczne”, że wykazują wspomniane wady twórców teoretycznych: są zbyt elastyczne, zbyt oderwane od wymagań narzucanych przez realne problemy. W 2006 roku podjęto próbę stworzenia odpowiedniej metodyki w ramach projektu europejskiego NeOn (<http://www.neon-project.org>). Projekt zakończył się w 2010 roku. Opracowano dokumentację opisującą metodykę tworzenia ontologii i stworzono zestaw narzędzi o nazwie *NeOn Toolkit*. Jednakże szczegółowa analiza wyników tego projektu nie daje, w opinii autorów niniejszego artykułu, podstaw do wyciągnięcia pozytywnych wniosków. Podstawową wadą opracowanej metodyki jest to, że jedynie stworzono zestaw ogólnikowych wskazówek, w jaki sposób powinny postępować zespoły biorące udział w projektowaniu baz wiedzy. Nie dostarczono żadnych technicznych środków umożliwiających sprawowanie kontroli nad procesem wytwarzania ani też nad procesem późniejszej eksploatacji tych baz.

Celem prac podjętych przez grupę KMG jest opracowanie modelu bazy wiedzy, który umożliwi uzyskanie takiej kontroli. Silną inspiracją jest tutaj metodyka związana z tworzeniem relacyjnych baz danych. Otóż w przypadku tychże baz jednym ze środków technicznych, które umożliwiają sprawowanie kontroli nad działaniami uczestników dwóch etapów życia systemu (etapu powstawania i etapu eksploatacji), jest podział bazy na schemat i instancję. Wiąże się to z wyodrębnieniem roli projektanta, tworzącego schemat, i roli użytkownika, który „produkuje” instancje. To właśnie dzięki temu podziałowi istnieje możliwość odpowiedniego zorganizowania pracy osób tworzących bazę oraz zdefiniowania ograniczeń uniemożliwiających użytkownikowi podjęcie czynności niezgodnych z intencją autorów.

Zdefiniowanie, czym mógłby być schemat bazy wiedzy, okazało się zadaniem niełatwym. Bardzo pomocny w tych pracach okazał się wspomniany już model konglomeratowy. Dostarczył on matematycznych podstaw, na których można było oprzeć dalsze rozważania. W rezultacie, jako kolejny etap prac nad tym modelem, powstał formalnie opracowany metamodel konglomeratowej bazy wiedzy. Metamodel ten dodaje do już istniejących bytów kolejne, składające się na schemat bazy. Został on opisany matematycznie oraz ontologicznie. W ramach niniejszego artykułu zapoznamy Czytelnika jedynie z opisem matematycznym.

W rozdziale 2. przedstawiony jest opis wymagań stawianych przed opracowywanym metamodeliem na tle obecnych rozwiązań modularnych baz wiedzy. Rozdział 3. zawiera opis modelu konglomeratowego w dotychczasowej wersji. Rozdział 4. omawia matematyczny opis nowego metamodelu, a rozdział 5. — przykład zastosowania go do zaprojektowania prostej bazy wiedzy. W rozdziale 6. znajduje się podsumowanie.

2. Czym powinien być schemat bazy wiedzy?

2.1. Podział na schemat i instancje

Jak zostało powiedziane we wstępie, główną rolą metodyki inżynierskiej jest dostarczenie środków i metod umożliwiających sprawowanie kontroli nad działaniami uczestników kolejnych etapów życia systemu. Ogólnie mówiąc, chodzi o zdefiniowanie ról, w jakich występują ci uczestnicy i nałożenie ograniczeń uniemożliwiających im działania, które stoją w sprzeczności z wyznaczonymi celami. Zdefiniowanie ról staje się możliwe dopiero wówczas, gdy wprowadzony zostanie podział bazy na *schemat* i *instancję*.

W przypadku baz wiedzy opartych na ontologiach można powiedzieć, że omawiany podział polega na wydzieleniu części terminologicznej i części zawierającej opis faktów. Część zawierająca opis faktów jest „wymierna”. Chcąc przeprowadzić wnioskowanie, użytkownik łączy z częścią terminologiczną swój opis faktów, zadaje następnie interesujące go zapytania, na koniec usuwa opis faktów. Ogólnie można powiedzieć, że każdy z przypadków ma swój własny opis faktów.

Jednak czy taki model eksploatacyjny odpowiada potrzebom inżynierii? Usuwanie pozornie niepotrzebnych opisów faktów kłóci się z inżynierskim rozumieniem pojęcia bazy wiedzy. Przecież baza powinna gromadzić wprowadzoną informację. Tego jednak nie da się zrobić, ponieważ w przypadku standardowych, niemodularnych baz wiedzy jak dotąd nie określono, w jaki sposób łączyć ze sobą informacje zawarte w różnych opisach faktów; jak mają one na siebie oddziaływać, w jaki sposób należy utrzymać spójność, w sytuacji gdy, z powodu zmiany punktu widzenia nowe fakty mogą zaprzeczać starym? W końcu nikt też nie określił, co zrobić, gdy terminologia przestaje być aktualna.

2.2. Schemat w modularnych bazach wiedzy

Przedstawione pytania znajdują się w centrum uwagi badaczy, którzy zajmują się bazami modularnymi. W tym przypadku modułami są nie tylko różne zbiory faktów, ale też różne terminologie, ujmujące opis dziedziny problemu z różnych punktów widzenia. W ramach tego kierunku badań odnotowano w ostatnich latach znaczne postępy. Uwaga naukowców była skoncentrowana na zaproponowaniu sposobu, w jaki informacja zawarta w jednym module powinna wpływać na informację zawartą w innych modułach, oraz na sposobie opisanego oddziaływania.

Dla ilustracji posłużmy się przykładem najlepiej obecnie dopracowanej techniki modularnych baz wiedzy, a mianowicie techniki zwanej DDL (ang. *Distributed Description Logic*) ([1]), która wprowadza pojęcie modularności dla ontologii wyrażonych w logice opisowej (ang. *Description Logic*)². W ramach tej techniki przepływ informacji z modułu źródłowego do modułu docelowego jest realizowany za pośrednictwem reguł zdefiniowanych przez użytkownika, tzw. reguł pomostowych (ang. *bridge rules*) i reguł odpowiedniości osobników (ang. *individual correspondences*). Dla ontologii źródłowej O_i i ontologii docelowej O_j reguły łączące przyjmują formę:

$$\begin{array}{lll} i:C & \xrightarrow{\sqsubseteq} & j:D & \text{reguła wstawiająca (ang. } \textit{into rule})} \\ i:C & \xrightarrow{\sqsupseteq} & j:D & \text{reguła wchłaniająca (ang. } \textit{onto rule})} \end{array}$$

oraz:

$$i:a \quad \longmapsto \quad j:b \quad \text{reguła odpowiedniości osobników}$$

Reguła wstawiająca deklaruje, że osobniki będące wystąpieniami konceptu C z i -tego modułu odpowiadają jedynie osobnikom będącym wystąpieniami konceptu D w j -tym module. Z kolei reguła wchłaniająca działa odwrotnie: osobniki, które są wystąpieniami konceptu D w j , mogą odpowiadać jedynie osobnikom będącym wystąpieniami konceptu C w module i . W dalszej części rozdziału oba rodzaje reguł będą nazywane regułami łączącymi.

Widać wyraźnie, że taki rodzaj łączenia modułów wiąże między sobą semantycznie ich konkretne zawartości. Zmiana ich zawartości powoduje konieczność przedefiniowania reguł łączących. Sprawia to, że reguły o takiej postaci nie mogą być częścią schematu, gdyż nie są elementem stałym — zmieniają się wraz ze zmianą zawartości instancji.

Przytoczony przykład jest reprezentatywny dla naszych obecnych rozważań. Inne techniki modularne definiują bazę wiedzy w bardzo podobny sposób: jako zamknięte zbiory zdań w jakiś sposób ze sobą powiązane. Nie uwzględniają one przyszłej dynamiki bazy i nie dają nam żadnych wskazówek, co mogłoby być schematem, a co instancją.

² Autorzy zakładają tutaj, że czytelnik posiada pewną znajomość podstaw logiki opisowej. Więcej informacji na jej temat można znaleźć w wielu opracowaniach, m.in. w [2].

2.3. Schemat a semantyka bazy

Uwzględnienie dynamiki bazy wiedzy wiąże się z umiejętnością udzielenia odpowiedzi na wiele pytań, w tym m.in. na pytania następujące:

- W jaki sposób można dołączyć nowy moduł?
- Z którymi modułami i w jaki sposób można go połączyć zależnościami międzymodułowymi?
- Które moduły są nieusuwalne, a które można usunąć?
- Które moduły można modyfikować, a które muszą pozostać niezmiennie?
- Które z istniejących opisów zależności międzymodułowych można usunąć lub zmienić?
- W jaki sposób dodawać opisy nowych zależności?

Przedstawione pytania są oczywiście bardzo ważne, ale uwzględniają jedynie aspekty techniczne. Dużo ważniejsze jest to, aby te aspekty w jakiś sposób korespondowały z semantycznymi właściwościami elementów bazy. Innymi słowy, powinno być wiadomo, jaki wpływ na semantykę projektowanego modelu ma podjęcie danej decyzji projektowej. Do postawionych pytań należy zatem dodać następujące:

- Reprezentacją czego jest dodawany moduł?
- Co oznacza takie a nie inne połączenie go z innymi modułami?
- Czym jest dodanie lub usunięcie modułu?
- Co oznacza dodanie lub usunięcie związku?
- Dlaczego niektóre moduły (związki) można usunąć, a innych nie?
- Dlaczego zawartość niektórych modułów można zmieniać, a inne muszą pozostać nienaruszalne?

Przedstawione spojrzenie na wymagania stawiane metodykom inżynierskim sprawia, iż model proponowany przez omówione w punkcie 2.2 techniki modularnych baz wiedzy nie jest odpowiedni. Konieczne jest takie rozwiązanie, które obejmuje pewną algebrę, pozwalającą opisać rolę modułów w sieci ich powiązań na wyższym poziomie abstrakcji niż ten związany z regułami łączącymi. Takim rozwiązaniem może być zaprezentowany w następnym rozdziale model konglomeratowy.

3. Model konglomeratowy

3.1. Algebra konglomeratów

W niniejszej sekcji zaprezentowany zostanie w skrócie formalizm modelu konglomeratowego. Szczegółowo został on przedstawiony w [3, 4, 5].

Teoretyczny model konglomeratowy definiuje bazę wiedzy jako dowolny podzbiór zbioru wszystkich możliwych konglomeratów \mathcal{K} . Element tego zbioru, czyli konglomerat, jest definiowany w następujący sposób:

Definicja 3.1. (konglomerat)

Konglomeratem k nazywamy parę (\mathbf{S}, \mathbf{W}) składającą się ze słownika \mathbf{S} i zbioru \mathbf{W} , zawierającego dopuszczalne interpretacje terminów z \mathbf{S} . Kiedy chcemy się odwołać do jednego z elementów pary, używamy notacji odpowiednio $\mathbf{S}(k)$ oraz $\mathbf{W}(k)$. Każdą interpretację z klasy \mathbf{W} nazywamy *modelem* konglomeratu k . ■

Terminy ze słownika \mathbf{S} to, zgodnie z założeniami logiki opisowej, nazwy konceptów, ról oraz nazwy indywidualne. Interpretacje definiowane są jako para $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, gdzie $\Delta^{\mathcal{I}}$ jest dziedziną, a $\cdot^{\mathcal{I}}$ funkcją przypisującą terminom z \mathbf{S} ich zakresy: osobnikom elementy $\Delta^{\mathcal{I}}$, konceptom podzbiory $\Delta^{\mathcal{I}}$, a rolom podzbiory $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. W dalszej części artykułu wykorzystujemy również pojęcie *projekcji* interpretacji. \mathcal{I}' jest projekcją interpretacji \mathcal{I} słownika \mathbf{S} do słownika \mathbf{S}' (co zapisujemy jako $\mathcal{I}' = \mathcal{I}|\mathbf{S}'$) wtwg. \mathcal{I}' jest interpretacją słownika \mathbf{S}' , dla której spełnione są następujące warunki: $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$ oraz $X^{\mathcal{I}'} = X^{\mathcal{I}}$ dla każdego $X \in \mathbf{S}'$.

Semantyczny opis konglomeratu pozwolił na zdefiniowanie w ramach modelu algebry konglomeratów, czyli zestawu działań bardzo przypominającego algebrę relacyjną.

Projekcja konglomeratu k , oznaczana symbolem $\pi_{\mathbf{S}}$, $\mathbf{S} \subseteq \mathbf{S}(k)$, tworzy nowy konglomerat o słowniku \mathbf{S} i zbiorze interpretacji utworzonym przez projekcję zawartości $\mathbf{W}(k)$ do \mathbf{S} :

$$\pi_{\mathbf{S}}(k) = (\mathbf{S}, \{\mathcal{I}|\mathbf{S}: \mathcal{I} \in \mathbf{W}(k)\}) \quad (1)$$

Przemianowanie, oznaczane symbolem ρ_{γ} , pozwala na zmianę nazw w słowniku danego konglomeratu z użyciem *funkcji przemianowującej* $\gamma: \mathcal{N} \rightarrow \mathcal{N}$ (\mathcal{N} to zbiór wszystkich nazw). Przez $\gamma(\mathcal{I})$, gdzie \mathcal{I} jest interpretacją słownika \mathbf{S} , oznaczamy interpretację \mathcal{I}' , taką że $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}$ oraz $\gamma(X)^{\mathcal{I}'} = X^{\mathcal{I}}$ dla każdego $X \in \mathbf{S}$:

$$\rho_{\gamma}(k) = (\gamma(\mathbf{S}(k)), \gamma(\mathbf{W}(k))) \quad (2)$$

Selekcja σ_{α} ogranicza zbiór interpretacji konglomeratu do tych, które spełniają zadane zdanie³ α :

$$\sigma_{\alpha}(k) = (\mathbf{S}(k), \{\mathcal{I} \in \mathbf{W}(k): \mathcal{I} \models \alpha\}) \quad (3)$$

Unia (\cup), *przecięcie* (\cap) oraz *różnica* ($-$) konglomeratów to działania, które dokonują odpowiednio sumowania, przecięcia i odjęcia zbiorów dopuszczalnych modeli dwóch konglomeratów (zakładamy, że $\mathbf{S}(k_1) = \mathbf{S}(k_2)$):

$$k_1 \cup k_2 = (\mathbf{S}(k_1), \mathbf{W}(k_1) \cup \mathbf{W}(k_2)) \quad (4)$$

$$k_1 \cap k_2 = (\mathbf{S}(k_1), \mathbf{W}(k_1) \cap \mathbf{W}(k_2)) \quad (5)$$

³ Selekcja jest jedynym operatorem, w którym wykorzystujemy zdania języka \mathcal{L} . Zdania te stanowią odpowiednik θ -ograniczeń z algebry relacyjnej.

$$k_1 - k_2 = (\mathbf{S}(k_1), \mathbf{W}(k_1) - \mathbf{W}(k_2)) \quad (6)$$

W [7] udowodniono, że algebra relacji Codda [6] jest wariantem algebry cylindrycznej [8]. W [4] udowodniono to samo w stosunku do algebry konglomeratów. Dzięki temu przejmie ona zarówno prawa algebry cylindrycznej, jak i prawa algebry relacji Codda.

3.2. Wyrażenia algebry konglomeratów

Operacje algebry konglomeratów pozwalają produkować dowolne konglomeraty z już istniejących. Dzięki temu możliwe jest dowolne zagłębianie ich wewnątrz innych operacji, czyli tworzenie *wyrażeń konglomeratowych* o dowolnej złożoności.

Definicja 3.2. (wyrażenie algebry konglomeratów)

Wyrażeniem algebry konglomeratów dla dziedziny D nazywamy formułę postaci:

$$F, G ::= F \cup G \mid F \cap G \mid F - G \mid \pi_{\mathbf{S}}(F) \mid \rho_{\gamma}(F) \mid \sigma_{\alpha}(F) \mid d,$$

przy czym \mathbf{S} , γ , α i d oznaczają odpowiednio dowolny słownik, funkcję przemianowującą, zdanie i element dziedziny D . ■

Najprostszy wybór dziedziny D to wskazanie pewnego dozwolonego zbioru konglomeratów. Tylko w takim przypadku możemy mówić o wartości wyrażenia algebry konglomeratów, które rozumiane jest jako wynik wykonania poszczególnych działań.

W dalszej części artykułu wykorzystane zostaną jednak także inne dziedziny, jak np. zbiór *zmiennych konglomeratowych*. Zmienna konglomeratowa może być nazwą wybraną ze zbioru dozwolonych nazw zmiennych (zbiór ten określa konkretna implementacja), która w procesie konkretyzacji musi podlegać wartościowaniu w uniwersum konglomeratów \mathcal{K} . Wartość dowolnej zmiennej v oznaczamy przez $|v|$.

Możliwości formułowania wyrażeń algebry i produkowania dowolnych konglomeratów pozwoliły na zdefiniowanie języka *KQL* (patrz [9]), którego zastosowanie w ramach konglomeratowej bazy wiedzy jest analogiczne do zastosowania języka SQL w ramach relacyjnej bazy danych. Obecny stan rozwoju metod konglomeratowych pozwala zatem na poprawną realizację obsługi danej instancji bazy wiedzy, jednakże nie daje możliwości projektowania baz wiedzy w tym sensie, jaki opisano w punkcie 2.3. Dlatego też dalsze prace grupy KMG skoncentrowały się na opracowaniu metod i środków służących temu zadaniu. W tym celu konieczne było rozwinięcie modelu przez wprowadzenie pojęć równości, większości i mniejszości konglomeratów.

W celu precyzyjnego zdefiniowania tych pojęć przyjmujemy pewne dodatkowe założenie: uznajemy, że każdy słownik i każdy zbiór modeli każdego konglomeratu są *zredukowane*. Zredukowany słownik $[\mathbf{S}(k)]$ powstaje z $\mathbf{S}(k)$ przez usunięcie terminów, których możliwe interpretacje nie są niczym ograniczone (tj. takich terminów X , dla których $k = \pi_{\mathbf{S}(k)}(\pi_{\mathbf{S}(k) - \{X\}}(k))$). Z kolei zredukowany zbiór interpretacji, oznaczany $[\mathbf{W}(k)]$, powstaje

z $\mathbf{W}(k)$ przez projekcję do słownika zredukowanego. *Zredukowany konglomerat* $[k]$, składa się zatem ze zredukowanego słownika i zredukowanego zbioru interpretacji.

Założenie to ułatwia porównywanie konglomeratów, gdyż pozwala na zignorowanie nieistotnych różnic słowników: za konglomerat większy (analogicznie mniejszy oraz równy) uznajemy konglomerat o większym zbiorze interpretacji ($k_1 > k_2 \Leftrightarrow \mathbf{W}(k_1) \supseteq \mathbf{W}(k_2)$). Inną korzyścią jest łatwość zdefiniowania konglomeratu pustego: konglomerat $k_0 = (\emptyset, \emptyset)$ jest konglomeratem pustym i istnieje taki tylko jeden w uniwersum \mathcal{K} .

Opierając się na pojęciach równości, mniejszości i większości konglomeratów, definiujemy *wyrażenia predykatywne*: równanie i nierówność konglomeratów.

Definicja 3.3. (równanie i nierówność konglomeratowa)

Równaniem algebry konglomeratów dla dziedziny D nazywamy formułę postaci $F = G$, a *nierównością algebry konglomeratów* – formułę postaci $F > G$, $F < G$, $F \geq G$ lub $F \leq G$, gdzie F i G są wyrażeniami algebry dla dziedziny D . ■

Jeśli $D \subseteq \mathcal{K}$, to fakt, że wyrażenie predykatywne jest spełnione, wynika z istnienia odpowiedniej relacji (równości bądź nierówności) między zredukowanymi konglomeratami będącymi wartościami wyrażen F i G . Dla wszelkich pozostałych dziedzin fakt spełniania należy zdefiniować.

Wyrażenia konglomeratowe oraz wyrażenia predykatywne są istotnymi elementami stworzonego systemu definiowania ograniczeń, który opisujemy w następnym podrozdziale.

4. Konglomeratowa baza wiedzy

Dotychczas konglomeratowa baza wiedzy (KBW) była definiowana jako ustalony i skończony zbiór konglomeratów. Jest oczywiste, że taka definicja uniemożliwiała jakąkolwiek kontrolę jej zawartości. Aby tę kontrolę umożliwić, należy wprowadzić do definicji bazy wiedzy wiele nowych elementów. W podrozdziale tym na nowo zdefiniujemy, czym jest konglomeratowa baza wiedzy, uwzględniając jej aspekty statyczny (schemat) i dynamiczny (instancję). Tym samym zostanie zaproponowany pewien metamodel tej bazy. Na kształt proponowanego metamodelu wpływa analiza możliwych rodzajów zmian. Wyróżniamy trzy takie rodzaje:

1. Zmiana zawartości konglomeratu, a w tym:
 - a) zmiany słownikowe, czyli dodanie do modułu nowych terminów ogólnych lub indywidualnych (zakładamy, że ze względu na postulat zachowania monotoniczności słowników nie można zmniejszać)
 - b) zmiany wpływające na semantykę znajdujących się w słowniku terminów ogólnych, czyli dodanie nowych zdań terminologicznych (aksjomatów),

- c) dodanie nowych faktów o osobnikach, których nazwy znajdują się w słowniku.
2. Dodawanie lub usuwanie konglomeratów.
3. Dodawanie lub usuwanie związków między konglomeratami.

Zmiana zawartości konglomeratu polega na „dopisaniu” nowego aksjomatu do którego z konglomeratów. „Dopisanie” jest ujęte w cudzysłów, gdyż musimy pamiętać, że konglomeraty nie składają się ze zdań. Dopisanie zdania α musi więc polegać na wykonaniu selekcji σ_α . W wyniku selekcji otrzymujemy nowy zbiór interpretacji, a więc nowy konglomerat z uniwersum \mathcal{K} . W konsekwencji takiej zmiany baza wiedzy będzie obejmowała nowy zbiór konglomeratów. Lepiej więc postrzegać KBW nie jako zbiór konglomeratów, lecz jako skończony zbiór zmiennych konglomeratowych. Opisywana zmiana nie spowoduje zmiany tego zbioru, lecz jedynie zmianę wartościowania jednej ze zmiennych.

Przyjęcie, że baza jest zbiorem zmiennych konglomeratowych, pozwala nam zdefiniować nowy typ elementów, umożliwiających kontrolę nad zmianami zawartości konglomeratów.

Definicja 4.1. (sprzęg)

Niech V będzie zbiorem zmiennych konglomeratowych. *Sprzęgiem nad zbiorem V* nazywamy wyrażenie predykatywne (równanie lub nierówność konglomeratową) dla dziedziny V . ■

Sprzęg s jest spełniony, gdy spełnione jest wyrażenie predykatywne otrzymane w wyniku zamiany wszystkich zmiennych w s na ich wartości (czyli konglomeraty). Przyjmujemy, że w bazie wiedzy akceptowane są tylko takie zmiany wartościowania zmiennych konglomeratowych, dla których wszystkie sprzęgi są prawdziwe (spełnione).

Drugi rodzaj zmian, czyli **dodanie lub usunięcie konglomeratu lub związku między konglomeratami**, jest dużo trudniejszy do kontrolowania, lecz to właśnie on jest najistotniejszy z punktu widzenia modularnej bazy wiedzy. Dodanie lub usunięcie konglomeratu wiąże się ze zmianą zbioru zmiennych konglomeratowych. To z kolei wpływa na konieczność modyfikacji zbioru sprzęgów. Jednak zmiany tego zbioru również nie mogą być dowolne. Konieczne zatem staje się wprowadzenie nowych elementów: typów konglomeratów i typów sprzęgów. Ich zadaniem jest wydzielenie grup konglomeratów i sprzęgów ze względu na ich właściwości i uzyskanie kontroli nad dotyczącymi ich zmianami.

Typy konglomeratów stanowią strukturę uporządkowaną relacją dziedziczenia. Strukturę tę nazywamy *hierarchią typów konglomeratowych*.

Definicja 4.2. (hierarchia typów konglomeratowych)

Hierarchia typów konglomeratowych to para (T_K, \triangleleft) , T_K nazywamy zbiorem *typów zmiennych konglomeratowych*, a \triangleleft *relacją dziedziczenia*, będącą ostrym porządkiem częściowym. W przypadku gdy $t_1 \triangleleft t_2$ ($t_1, t_2 \in T_K$), mówimy, że t_2 *dziedziczy od t_1* . ■

Zdefiniowana hierarchia może być nałożona na zbiór zmiennych konglomeratowych.

Definicja 4.3. (ustalenie typów)

Niech V_K będzie skończonym zbiorem zmiennych konglomeratowych. *Ustaleniem typów przez nałożenie hierarchii typów* (T_K, \triangleleft) dla zbioru V_K nazywamy stworzenie funkcji f_{in} , nazywanej *funkcją konkretyzacji* z T_K do $\mathcal{P}(V_K)$ (zbioru potęgowego V_K). Funkcja f_{in} uwzględnia przechodniość relacji \triangleleft w taki sposób, że $\forall t_1, t_2 \in T_K \forall v \in V_K ((t_1 \triangleleft t_2 \wedge v \in f_{in}(t_2)) \rightarrow v \in f_{in}(t_1))$. Zmienną przyporządkowaną typowi $t \in T_K$ (tj. należąca do $f_{in}(t)$) nazywamy zmienną typu t . Funkcję f_{in} nazywamy wynikiem ustalenia typów. ■

Należy zauważyć, że typy konglomeratowe nie są typami konglomeratów, lecz jedynie zmiennych konglomeratowych. Dwie zmienne różnych typów mogą w szczególnych przypadkach mieć tę samą wartość. Z kolei do grupowania sprzęgów służą typy sprzęgów.

Definicja 4.4. (typ sprzęgu)

Niech V_K będzie skończonym zbiorem zmiennych konglomeratowych, dla którego ustalono typy, tworząc strukturę $(T_K, \triangleleft, f_{in})$. *Typem sprzęgu nad zbiorem V_K* nazywamy wyrażenie predykatywne (równanie lub nierówność konglomeratową) dla dziedziny slotów L . *Slotem* $l \in L$ nazywamy parę (v, t) , gdzie $t \in T_K$, a $v \in V_K$ lub v jest zmienną, której zakresem wartościowania jest $f_{in}(t)$. ■

Dany sprzęg s jest typu t_s wtwg. jest on wynikiem przekształcenia t_s ; polegającego na zastąpieniu wszystkich slotów (v, t) przez v , gdy $v \in V_K$, lub przez $|v|$ w przeciwnym przypadku. Typy sprzęgów pozwalają doprecyzować definicję typów konglomeratowych.

Definicja 4.5. (ograniczenia dla typu konglomeratowego)

Niech T_S będzie zbiorem typów sprzęgów. Ograniczenia dla typu konglomeratowego nad zbiorem typów sprzęgów T_S mogą przyjmować następujące formy:

Ograniczeniem liczebnościowym dla danego typu konglomeratowego t nazywamy trójkę (t_s, l, N) , gdzie: $t_s \in T_S$, l jest wyróżnionym slotem, a N jest podzbiorem liczb naturalnych.

Ograniczeniem edycyjnym dla danego typu konglomeratowego t nazywamy wartość E ze zbioru $\{terminologia, fakty, nietwórz\}$. ■

Dysponując opisanymi elementami, możemy już sformułować definicje schematu i instancji KBW. Schemat jest ustaloną częścią bazy wiedzy. Przyjmujemy, że jeśli ulegnie on zmianie, będziemy mieli do czynienia z nową bazą.

Definicja 4.6. (schemat konglomeratowej bazy wiedzy)

Schematem konglomeratowej bazy wiedzy jest struktura $(K_S, V_{KS}, (T_K, \triangleleft), f_{inS}, T_S, O, f_o, w_{KS})$, gdzie:

- K_S — podzbiór \mathcal{K} ,
- V_{KS} — zbiór zmiennych konglomeratowych,
- (T_K, \triangleleft) — hierarchia typów konglomeratowych,
- f_{inS} — wynik nałożenia (T_K, \triangleleft) na V_{KS} ,
- T_S — zbiór typów sprzęgów,

- O — zbiór ograniczeń dla typów konglomeratów nad T_S ,
- f_o — funkcja przypisująca każdemu typowi z T_K podzbiór O ; f_o uwzględnia przechodność relacji \triangleleft , przypisując każdemu typowi także ograniczenia wszystkich typów dziedziczonych w ten sposób, że gdy $t_1 \triangleleft t_2$, to $f_o(t_1) \subseteq f_o(t_2)$,
- w_{KS} — funkcja częściowa $V_{KS} \rightarrow K_S$.

Wszystkie przedstawione zbiory są zbiorami skończonymi. ■

Istnienie elementów K_S i V_{KS} uwzględnia możliwość, że część instancji bazy wiedzy może zostać ustalona już na etapie projektowania. Możemy więc mówić o zmiennych konglomeratowych i konglomeratach *predefiniowanych*. Ich obecność w schemacie sprawia, że nie mogą one zostać usunięte ani zmienione.

Z kolei instancja jest pewnym chwilowym stanem bazy wiedzy. Zmiana stanu, wywołana nowymi wpisami, jest związana z pojawieniem się nowej instancji.

Definicja 4.7. (instancja konglomeratowej bazy wiedzy)

Instancją schematu konglomeratowej bazy wiedzy $\Sigma = (K_S, V_{KS}, (T_K, \triangleleft), f_{inS}, T_S, O, f_o, w_{KS})$ jest struktura $(\Sigma, (K, V_K, S, w_K), f_{in}, f_{sin})$, gdzie:

- K — podzbiór \mathcal{K} ,
- V_K — zbiór zmiennych konglomeratowych,
- S — zbiór sprzęgów,
- w_K — funkcja $V_K \rightarrow K$,
- f_{in} — wynik nałożenia (T_K, \triangleleft) na V_K ,
- f_{sin} — funkcja $S \rightarrow T_S$,

przy czym muszą być spełnione wszystkie *warunki dopasowania schematu*, czyli:

- *warunek utrzymania stałych*: $K \supseteq K_S$,
- *warunek utrzymania zmiennych*: $V_K \supseteq V_{KS}$,
- *warunek utrzymania wartości stałych*: $w_K \supseteq w_{KS}$,
- *warunek właściwego typowania konglomeratów*: $f_{in} \supseteq f_{inS}$,
- *warunek spełniania sprzęgów*: wszystkie sprzęgi ze zbioru S muszą być spełnione,
- *warunek właściwego typowania sprzęgów*: dla każdego $s \in S$: s musi być typu $f_{sin}(s)$,
- *warunek utrzymania ograniczeń*: dla każdego typu t , dla każdej zmiennej konglomeratowej v z $f_{in}(t)$ muszą być spełnione wszystkie ograniczenia ze zbioru $f_o(t)$, czyli:
 - ograniczenie liczebnościowe (t_k, l, N) jest spełnione, gdy liczba sprzęgów ze zbioru S , którym funkcja f_{sin} przypisuje typ t , a które powstają w wyniku zamiany l na v , mieści się w N ,
 - spełnione jest ograniczenie edycyjne E .

■

Ścisłe matematyczne wyrażenie znaczenia ograniczenia edycyjnego E jest trudne i pochłonęłoby tu zbyt wiele miejsca. W dużym uproszczeniu można powiedzieć, że gdy $E = \text{nietwórz}$, to nie wolno tworzyć żadnych nowych konglomeratów danego typu (czyli $f_{in}(t) = f_{ins}(t)$), gdy $E = \text{terminologia}$, to nie wolno „dopisywać” do konglomeratu żadnych zdań terminologicznych i, analogicznie, gdy $E = \text{fakty}$, to nie wolno „dopisywać” do konglomeratu żadnych nowych faktów.

Przedstawiony matematyczny model konglomeratowej bazy wiedzy został opisany w postaci ontologii. Część asercyjna (opis faktów) tej ontologii zawiera opis jednej instancji KBW, przy czym zawartość konglomeratów jest reprezentowana zdaniowo przy wykorzystaniu pojęcia \mathcal{L} -(2)- \mathbf{D} -reprezentowalności (patrz [5]), czyli każdemu konglomeratowi odpowiada zbiór ontologii.

5. Przykład zastosowania schematu

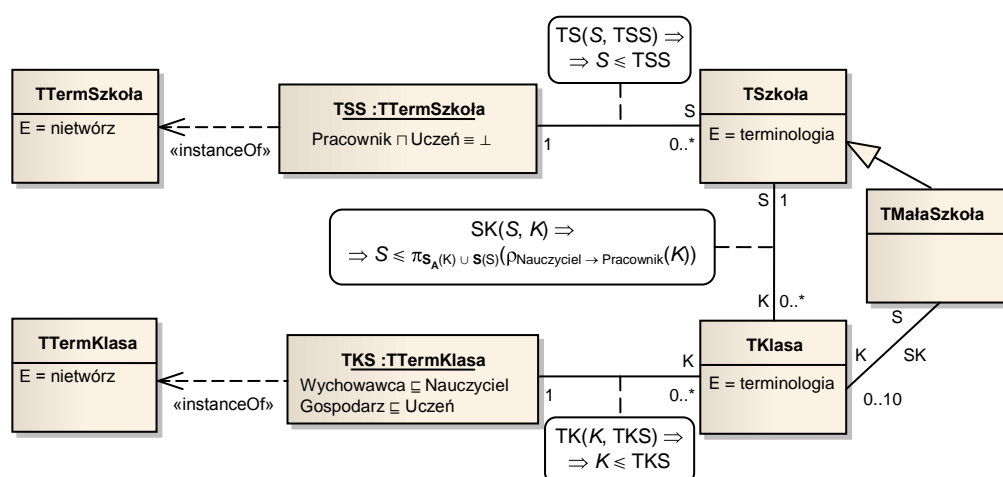
W niniejszym rozdziale zaprezentowano przykładowe wykorzystanie schematu konglomeratowej bazy wiedzy w postaci wprowadzonej w podrozdziale 4.2. Rozpatrywany tu przykład dotyczy bazy przechowującej informacje na temat szkół, klas, uczniów i nauczycieli.

Przyjmujemy przy tym następujące założenia:

- szkoły opisywane są z punktu widzenia ekonomicznego zarządzania instytucją; są tam informacje o uczniach, ale nauczyciele traktowani są po prostu jako pracownicy (1),
- klasy opisywane są w znacznie większych szczegółach, zapamiętywane są m.in. informacje o gospodarzu i wychowawcy klasy (2),
- terminologia opisu szkół i klas zmienia się bardzo rzadko bądź wcale (3a), liczba szkół i klas, a także informacje o nich, mogą się natomiast zmieniać bardzo szybko (3b).

Zgodnie z tymi założeniami można przygotować schemat bazy konglomeratowej. Schemat taki pokazano w postaci graficznej na rysunku 1. Zastosowano tutaj zaadaptowany wariant notacji UML: symbole klas reprezentują typy zmiennych konglomeratowych, symbole obiektów — zmienne konglomeratowe, asocjacje — typy sprzęgów. Ograniczenia na typy zostały pokazane w formie ograniczeń liczebności po jednej ze stron asocjacji (ograniczenia liczebnościowe) oraz w postaci dodatkowych informacji naniesionych na symbol klasy (ograniczenia edycyjne i kreacyjne). Dokładna postać wyrażeń algebraicznych została podana przy odpowiednich typach sprzęgów w zaokrąglonych ramkach.

Schemat definiuje pięć typów zmiennych konglomeratowych: $TTermSzkoła$, $TTermKlasa$, $TSzkoła$, $TKlasa$ oraz $TMałaSzkoła$. Hierarchia dziedziczenia obejmuje jedynie dziedziczenie między $TMałaSzkoła$ a $TSzkoła$ (tj. relacja $\triangleleft = (TSzkoła, TMałaSzkoła)$).



Rys. 1. Graficzna reprezentacja schematu przykładowej konglomeratowej bazy wiedzy
 Fig. 1. Graphical representation of the exemplary schema of s-modular knowledge base

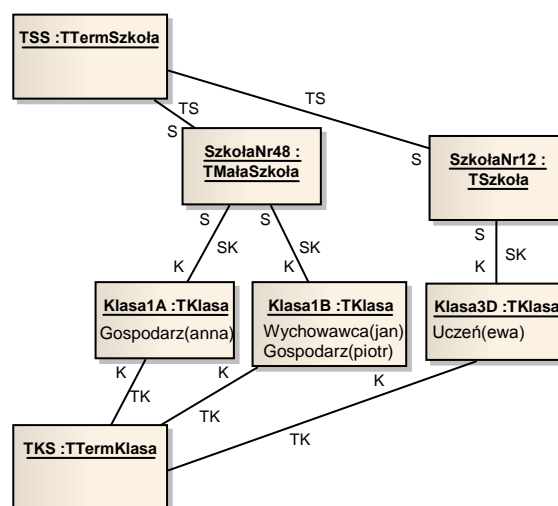
Typy *TTermSzkoła* i *TTermKlasa* mają charakter pomocniczy, przypisane jest im ograniczenie kreacyjne, co oznacza, że użytkownik bazy nie będzie mógł tworzyć konglomeratów tych typów. Każdy z tych typów ma jedno predefiniowane wystąpienie, tj. predefiniowaną zmienną konglomeratową, konkretyzującą dany typ. Wystąpienia te to odpowiednio *TSS* i *TKS*. Mają one przypisaną zawartość opisującą prostą terminologię dla szkół (zmienna *TSS*; zawiera ona aksjomat mówiący o rozłączności conceptów *Uczeń* i *Pracownik* — założenie 1) i dla klas (*TKS*; zawiera ona aksjomaty, mówiące, że każdy *Wychowawca* jest *Nauczycielem*, a *Gospodarz* — *Uczniem* — założenie 2); terminologie te zdecydowano się umieścić w predefiniowanych konglomeratach ze względu na założenie 3a.

Dwa zasadnicze typy zmiennych konglomeratowych to *TSzkoła* i *TKlasa*. Użytkownik może tworzyć dowolną liczbę zmiennych tych typów. Zawartość każdej zmiennej konglomeratowej typu *TSzkoła* opisuje pojedynczą szkołę, a typu *TKlasa* — pojedynczą klasę. Jednak konglomeraty te będą mogły zawierać jedynie asercje opisujące aktualną wiedzę na temat szkół i klas (ze względu na ograniczenie edycyjne). Typ *TMałaSzkoła* dziedziczy od typu *TSzkoła*, jednak wprowadza dodatkowe ograniczenie na liczbę klas.

W schemacie zawarto trzy typy sprzęgów. Typ *TS* opisuje sprzęg pomiędzy zmienną konglomeratową typu *TSzkoła* a zmienną *TSS*, zawierającą terminologię. Odpowiednie ograniczenia liczebnościowe wymuszają, by każda ze zmiennych typu *TSzkoła* (i *TMałaSzkoła*) była związana sprzęgiem typu *TS* z *TSS*. Działanie tego sprzęgu podlega ograniczeniu zbioru interpretacji konglomeratu przypisanego zmiennej typu *TSzkoła*, tak aby uwzględniał on odpowiednie zależności terminologiczne z *TSS* (rozłączność conceptów *Uczeń* i *Pracownik*). Analogiczny typ sprzęgu *TK* i analogiczne ograniczenie wprowadzono dla typu *TKlasa*.

Ostatni typ sprzęgu, *SK*, opisuje zależność między zmienną konglomeratową typu *TSzkoła* a zmienną typu *TKlasa*. Według ograniczeń każda klasa musi należeć do jakiejś szkoły (tj. być związana jednym sprzęgiem typu *SK* ze zmienną typu *TSzkoła*). Typ sprzęgu jest wy-

rażeniem algebraicznym, zapewniającym że informacje na temat nauczycieli zostaną przeniesione do konglomeratu reprezentującego szkołę jako informacje o pracownikach (przemianowanie $\sigma_{Nauczyciel \rightarrow Pracownik}$), informacje o uczniach zostaną zaś przeniesione bezpośrednio. Projekcja zapewnia, że do słownika konglomeratu opisującego szkołę nie zostaną przeniesione terminy specyficzne dla klas (np. *Wychowawca*). Dodatkowe ograniczenie liczebnościowe sprawia, że liczba klas przypisana małej szkole nie może przekraczać dziesięciu.



Rys. 2. Graficzna reprezentacja instancji przykładowej konglomeratowej bazy wiedzy
Fig. 2. Graphical representation of the exemplary instance of s-modular knowledge base

Przykładową instancję, zgodną z podanym schematem, zilustrowano na rys. 2. Oczywiście użytkownik może dodawać asercje do istniejących opisów szkół i klas, może też dodawać nowe szkoły i klasy, jednak schemat wymusza, aby każda nowo dodawana zmienna konglomeratowa typu *TSzkoła* była związana z *TSS*, a zmienna typu *TKlasa* z *TKS* oraz z jedną zmienną typu *TSzkoła*. Ze względu na te sprzęgi schemat wymusza także konieczność wyprowadzania dodatkowych wniosków: przykładowo ze zbioru modeli konglomeratu *SzkołaNr48* musi wynikać, że *jan* jest *Pracownikiem*, a *anna* i *piotr* są *Uczniami*. Warto podkreślić, że wnioskowanie nie obejmuje typów zmiennych konglomeratowych: np. nie możemy wnioskować, że *SzkołaNr12* jest małą szkołą – przypisanie typu zależy od decyzji użytkownika.

6. Podsumowanie i przyszłe prace

Niniejszy artykuł prezentuje efekty prac nad zmianą definicji bazy wiedzy, która w rezultacie umożliwiłaby stworzenie środków technicznych, pozwalających na sprawowanie kontroli nad kolejnymi etapami życia bazy. Przedstawiono nowy metamodel konglomeratowej bazy wiedzy, który uwzględnia jej podział na schemat i instancję. Zdefiniowano także, co

należy rozumieć przez aktualizacje bazy oraz jakim ograniczeniom mogą one podlegać. Umożliwiło to rozdzielenie ról projektantów i użytkowników, a co za tym idzie – otworzyło drogę do opracowania metodyki, która lepiej realizowałaby wymagania związane z praktyką inżynierską.

Opisane rezultaty ustanawiają bazę, na której można oprzeć dalsze prace. Ich daleko-
siężnym celem jest dostarczenie odpowiednich systemów, które będą potrafiły wesprzeć pro-
jektantów w dziele tworzenia schematu bazy, a w trakcie eksploatacji zapewnią spójność in-
stancji ze schematem. W krótkiej perspektywie łatwo określić, jakie konkretnie zadania nale-
ży podjąć w pierwszej kolejności. Są to: (1) utworzenie metody zapisu schematu bazy, (2)
uwzględnienie faktu istnienia schematu i instancji w języku KQL, (3) stworzenie algorytmów
wnioskujących z konglomeratowych baz wiedzy oraz (4) stworzenie odpowiednich narzędzi
wnioskujących. Niektóre z tych zadań zostały już zapoczątkowane, m.in.: stworzenie metaon-
tologii służącej opisowi schematu konglomeratowej bazy wiedzy (<http://knot260.eti.pg.gda.pl/ontologies/congloschema/>), opracowanie wtyczek do systemu Protege obsługujących
KBW (<http://protege.stan-ford.edu/>), prace nad nową wersją systemu wnioskującego.

BIBLIOGRAFIA

1. Borgida A., Serafini L.: Distributed Description Logics: Assimilating Information from Peer Sources. *Journal on Data Semantics*, 2003, s. 153-184.
2. Baader F. A., McGuinness D. L., Nardi D., Patel-Schneider P. F. (eds.): *The Description Logic Handbook: Theory, implementation, and applications*. Cambridge University Press, 2003.
3. Goczyła K., Waloszek A., Waloszek W.: Algebra konglomeratów jako narzędzie opisu problemów przetwarzania ontologii. *Studia Informatica*, Vol. 30, No. 2A, Gliwice 2009, s. 141÷156.
4. Goczyła K., Waloszek A., Waloszek W.: A Semantic Algebra for Modularized Description Logics Knowledge Bases. *Proc. of DL Workshop 2009, CEUR-WS*, Vol. 477, 2009.
5. Goczyła K., Waloszek A., Waloszek W.: Algebra of ontology modules for semantic agent. *Computational collective intelligence: semantic web, social networks and multiagent systems*, LNCS Vol. 5796, Springer, 2009, s. 492÷503.
6. Codd E. F.: Relational Completeness of Data Base Sublanguages. *Database Systems*, Vol. 6, 1979, s. 65÷98.
7. Imielinski T., Lipski W. J.: The Relational Model of Data and Cylindric Algebras. *J. Comput. Syst. Sci.*, Vol. 28, No. 1, 1984, s. 80÷102.

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. Studia Informatica, Vol. 31, No. 2A, Gliwice 2010, s. 47÷61.

Wpłynęło do Redakcji 15 stycznia 2012 r.

Abstract

The article discusses the problem of extending the s-modular approach by creating an outline of the process of designing and creating s-modular knowledge bases. One of the main obstacles to this task is the fact that no precise definition has been provided for the s-modular knowledge base and no distinction has been made between a schema and an instance of such a base.

The last problem seems to be prevalent for all the popular methods of ontology modularization. Although several definitions of a knowledge base itself have been proposed, to the best of the knowledge of the authors, no extensive work has been done to analyze dynamic changes of such a base, and consequently to identify its invariant part.

In the paper we undertake the aforementioned issue and try to formulate a list of questions that should be answered in order to define a schema in the proper way. On the basis of the list we propose a new definition of an s-modular knowledge base with a schema that allows for adding and deleting modules while maintaining the design assumptions for described structure.

The definitions introduced in the paper rely heavily on the s-module algebra. The connections between modules in a base are described as *couplers*, which basically are algebraic expressions binding together contents of several modules. The schema in turn introduces the notion of *module types* and *coupler types* that form categories for specific s-modules containing distinguished fragments of domain knowledge, and describing typical and required relationships between modules.

In the concluding part of the paper we present a brief scenario of use of the introduced notions. We show how design assumptions and requirements may be transferred into a schema, and how the schema influences future dynamics of a knowledge base.

Adresy

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.