

Jarosław BAŁ, Jolanta CYBULKA  
Politechnika Poznańska, Instytut Automatyki i Inżynierii Informatycznej

## **METODA PRZYROSTOWEGO TWORZENIA BAZY WIEDZY NA PODSTAWIE BAZY DANYCH, ONTOLOGII I WIEDZY EKSPERTA**

**Streszczenie.** Rozważamy narzędzie wspomagające budowanie bazy wiedzy. Punkt wyjścia stanowią: baza danych, ontologia odniesienia oraz niesformalizowana wiedza eksperta. Budujemy pierwotną bazę wiedzy odkrywając ontologię ze schematu bazy danych i specyfikując za pomocą relacji ontologicznych powiązania pomiędzy atrybutami bazy. W kolejnych krokach baza pierwotna jest wzbogacana o zapisywane do niej efekty wykonywanych zapytań.

**Słowa kluczowe:** tworzenie bazy wiedzy, ontologia, baza danych, wiedza eksperta

## **A KNOWLEDGE BASE INCREMENTAL CREATION USING A DATA BASE, ONTOLOGY AND EXPERT KNOWLEDGE**

**Summary.** We consider the tool supporting the creation of a knowledge base (KB). It uses as the input elements: a database, a referential ontology and the expert knowledge. We reveal the ontology from a database schema and we specify connections between database attributes by means of the ontological relations, forming the primary KB. It is then incrementally extended by the results of executed queries.

**Keywords:** knowledge base creation, ontology, database, expert knowledge

### **1. Wprowadzenie – scenariusz wykorzystania metody**

Współcześnie większość instytucji zarówno publicznych, jak i działających na rynku dysponuje znacznymi magazynami danych zgromadzonych w formie relacyjnym. Zatem, problemu nie stanowi zazwyczaj brak danych w formie przetwarzalnej informatycznie, lecz konieczność zapewnienia bezpiecznego, zunifikowanego i inteligentnego do nich dostępu. Zadanie może być realizowane na różne sposoby, w pracy proponujemy metodę budowania

bazy wiedzy na podstawie istniejącej bazy danych (istniejących baz danych), przy wykorzystaniu semantyki danych ujawnionej ze schematu bazy (schematów baz) w kontekście utworzonej wcześniej, odpowiednio bogatej semantycznie ontologii odniesienia. Ontologia zawiera warstwę reprezentującą semantykę dziedziny, z której pochodzą dane. By być użyteczna tworzona baza wiedzy musi uwzględniać wiedzę eksperta. Wiedzą tego rodzaju baza jest „nasycona” stopniowo przez zapisywanie doń faktów wywnioskowanych w efekcie jej eksploracji za pomocą zapytań.

Rozważmy następujący scenariusz postępowania.

Grupa współpracujących osób rozwiązuje problem polegający na przeszukiwaniu zgromadzonych danych w celu odnalezienia w nich śladów potwierdzających zajście w przeszłości określonych zdarzeń. Sposób przeszukiwania stanowi wiedzę eksperta, która, po sformalizowaniu, mogłaby posłużyć w przyszłości do oceny podobnych zjawisk. W tym celu wyposażymy grupę w narzędzie w formie odpowiedniej bazy wiedzy, której model początkowy zostanie wygenerowany półautomatycznie na podstawie bazy danych, ontologii odniesienia i asercji definiowanych przez inżyniera wiedzy, specyfikujących powiązania pomiędzy atrybutami bazy danych w terminach relacji pochodzących z ontologii. Dalszy rozwój bazy wiedzy odbywa się w trakcie jej użytkowania przez grupę. Polega on na dołączaniu faktów wywnioskowanych w trakcie przepytania bazy za pomocą przyjaznego języka zapytań, w którym wykorzystywane są pojęcia ontologiczne, a odpowiedzi uzyskuje się stosując wnioskowanie. Celem pracy jest zaprezentowanie metody budowy rozważanego systemu i narzędzia programistycznego ją realizującego.

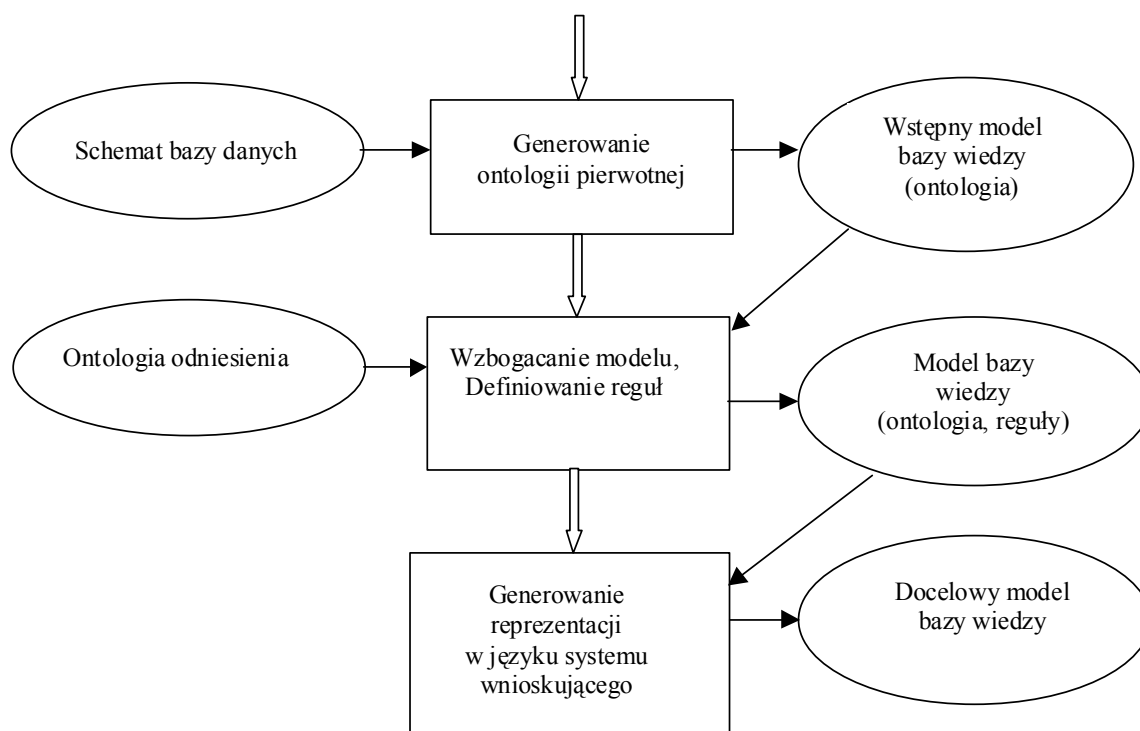
Prezentowana w pracy metoda wykazuje związki z nowym nurtem badawczym nazwanym OBDA (ang. *Ontology-Based Data Access*), w którym wskazuje się sposób dostępu do danych relacyjnych za pomocą języków zapytań bazujących na modelu konceptualnym dziedziny wyrażonym za pomocą ontologii [12]. Pośrednio, w naszej metodzie również uzyskujemy taki dostęp, jednakże jej zasadniczym celem jest tworzenie bazy wiedzy.

W następnym punkcie omówimy proponowaną w pracy metodę. W trzecim punkcie zaprezentujemy rozwijane i stosowane od kilku lat środowisko biblioteczne SDL (ang. *Semantic Data Library*) [1, 2], które stanowi narzędzie implementacji metody. Punkt 4. zawiera charakterystykę zastosowanej ontologii odniesienia, a w punkcie 5. przedstawiamy przykład zastosowania metody do budowania konkretnej bazy wiedzy. Pracę kończy podsumowanie.

## 2. Opis metody

Na rys. 1 zilustrowano schematycznie proponowaną metodę. Po połączeniu się z bazą relacyjną następuje wczytanie jej schematu i na jego podstawie generuje się uproszczoną onto-

logię, wstępnie definiującą semantykę danych. Pojęcia ontologiczne odzwierciedlają nazwy tabel i atrybutów, zawarte w schemacie relacyjnym. W zależności od potrzeb semantyka ta może być wzbogacona o nowe pojęcia i relacje, o ile wczytamy odpowiednią ontologię odniesienia. Inżynier wiedzy może także dołączyć dodatkową wiedzę i wyrazić ją za pomocą reguł w języku SWRL (punkt 3.2). W ten sposób powstaje model konceptualny bazy wiedzy, do której następnie dane są „zasysane” z bazy relacyjnej. W procesie tym wykorzystuje się maszynę wnioskującą. Zastosowanie maszyny jest poprzedzone ujednocnieniem formatu modelu konceptualnego, polegającym na wygenerowaniu zapisu w języku maszyny wnioskującej.

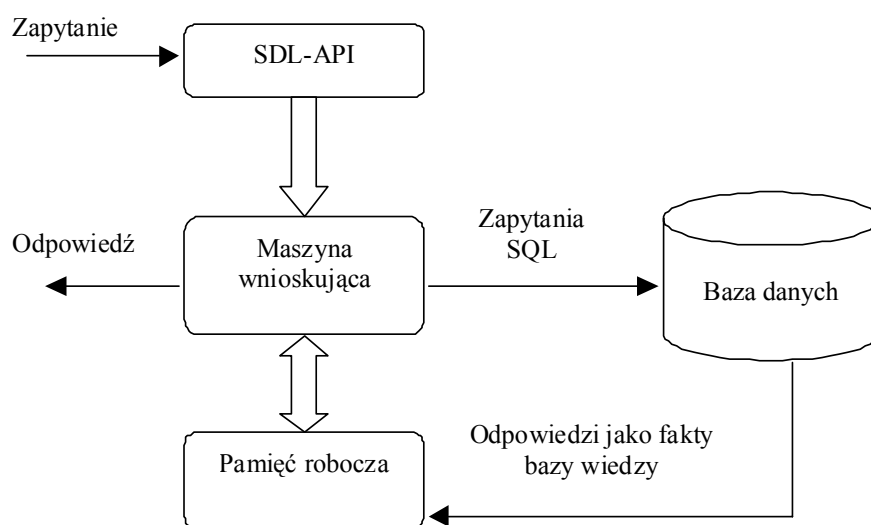


Rys. 1. Schemat metody tworzenia modelu bazy wiedzy

Fig. 1. Scheme of a knowledge base model creation method

Metodę przyrostowego wypełniania bazy wiedzy danymi relacyjnymi zilustrowano na rys. 2. W bazie wiedzy znajdują się tylko te dane, które zostaną uzyskane w odpowiedzi na zapytania skierowane pod adresem modułu SDL-API (punkt 3.1). Moduł ten wywołuje maszynę wnioskującą, która za pośrednictwem języka SQL współpracuje z systemem zarządzania bazą danych. Dane uzyskane w odpowiedzi na zapytanie składowane są w pamięci roboczej (pamięci operacyjnej), której zawartość może być zachowywana na dysku lub zerowana po zrealizowaniu zapytania. Zerowanie pamięci roboczej pozwala na utrzymywanie wiarygodności systemu, gdyż zapytania zadawane są zawsze do aktualnego stanu, w jakim znajduje się relacyjna baza danych. Ponadto, w regułach wnioskowania (o postaci klauzul Horna) nie jest stosowana negacja, co zapewnia monotoniczność wnioskowania. W pamięci roboczej zawarta jest także reprezentacja modelu bazy wiedzy, który zazwyczaj jest stały, może jednak pod-

legać zmianom. Inżynier wiedzy ma możliwość definiowania nowych reguł, a także pojęć i relacji ontologicznych, posługując się językiem systemu wnioskującego.



Rys. 2. Schemat wypełniania bazy wiedzy danymi relacyjnymi

Fig. 2. Scheme of population of a knowledge base with relational data

### 3. Charakterystyka środowiska SDL

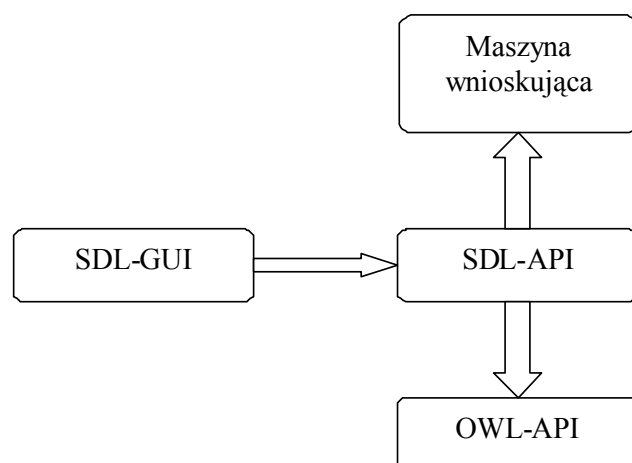
#### 3.1. Architektura systemu

Biblioteka SDL (ang. *Semantic Data Library*) jest narzędziem programistycznym, za którego pośrednictwem można specyfikować odniesienia danych z bazy relacyjnej do metadanych semantycznych, zawartych w ontologii referencyjnej. W ten sposób dane relacyjne zyskują jawnie wyspecyfikowaną semantykę, co daje początek opartej na danych relacyjnych bazy wiedzy. Dodatkowo, system umożliwia reprezentowanie wiedzy eksperta sformułowanej w postaci regułowej i dołączanej do bazy wiedzy. Baza ta może być następnie przetwarzana za pomocą zapytań, na które odpowiedzi uzyskuje się przez wnioskowanie.

Na rys. 3 przedstawiono architekturę modułową biblioteki SDL. System jest wyposażony w interfejs użytkownika SDL-GUI, za którego pomocą można wywoływać funkcje interfejsu aplikacyjnego SDL-API. Interfejs SDL-API komunikuje się z maszyną wnioskującą oraz interfejsem służącym do przetwarzania danych wyrażonych w języku OWL (punkt 3.2). Szczegóły realizacyjne systemu SDL omówiono w punkcie 3.3.

W systemie SDL przeprowadzane są wnioskowania wykorzystujące dane z bazy relacyjnej oraz metadane (pojęcia, reguły) zdefiniowane w ontologii. W bieżącej implementacji można wnioskować o hierarchii pojęć w ontologii oraz na bazie zdefiniowanych reguł. W systemie zaimplementowano metody realizacji zapytań wykorzystujące wnioskowanie hybrydowe [1, 2], na które składają się mechanizmy wnioskowania wstecz oraz w przód. Zapy-

tania formułowane są z wykorzystaniem pojęć ontologicznych. Możliwe jest nakładanie różnego rodzaju warunków na zmienne występujące w zapytaniach (z użyciem relacji =, <, >, itp.). Zapytania zadawane są w języku systemu wnioskującego Jess (punkt 3.2), trwają prace nad wersją graficzną języka zapytań, w której pytania przyjmują formę skierowanych grafów. Dzięki takiemu podejściu proces przepytывania bazy wiedzy może być realizowany przez użytkownika niebędącego profesjonalistą z zakresu informatyki.



Rys. 3. Modułarna architektura biblioteki SDL  
Fig. 3. Modular architecture of the SDL library

### 3.2. Języki, metody i narzędzia realizacji systemu

Realizując system SDL wykorzystano następujące języki reprezentacji wiedzy:

- OWL (ang. Web Ontology Language) [4], trójwarstwowy (OWL-Lite, OWL-DL, OWL-Full) język wyrażania ontologii, który wykorzystano do reprezentowania modelu koncepcyjnego wiedzy.
- SWRL (ang. Semantic Web Rule Language) [7], wykorzystany do formalizowania wiedzy eksperta; SWRL umożliwia definiowanie reguł w postaci klauzul Horna, które mogą operować na pojęciach ontologicznych; wydedukowany za pomocą reguły fakt (nowa dana) może być dodany do bazy wiedzy, z kolei, nowe dane powodują stosowanie nowych reguł (w przypadku sterowania reguł danymi); język SWRL można wzbogacać o tzw. funkcje wbudowane, które wyrażają warunki w języku SWRLB (ang. SWRL Built-ins) [7], na przykład  $?x > ?y$  ( $?x$  i  $?y$  oznaczają zmienne).

System SDL, współpracuje z dwoma narzędziami, którymi są:

- napisana w Javie maszyna wnioskująca Jess (ang. *Java Expert System Shell*) [9], w której zastosowano algorytm Rete [3]; narzędzie służy do przeprowadzania wnioskowania w przód i wstecz; Jess jest udostępniany nieodpłatnie do celów badawczych i edukacyjnych na okres dwóch lat; w SDL wykorzystano wersję 7.1p1.

- Jena2 [8], darmowy system służący do budowania aplikacji w paradygmacie sieci semantycznej; w bibliotece SDL wykorzystywane są funkcje interfejsu API Jena2 przeznaczone do wykonywania operacji na plikach w języku OWL (tworzenie, wczytywanie).

### 3.3. Charakterystyka funkcji systemu SDL

Poszczególne moduły biblioteki SDL zbudowano posiłkując się językami, metodami i narzędziami zaprezentowanymi w punkcie 3.2. Maszyną wnioskującą jest system Jess. Jako OWL-API wykorzystano aplikację Jena2, natomiast SDL-API oraz SDL-GUI to metody oraz funkcje zaimplementowane w języku Java. SDL-API jest interfejsem programisty, za którego pośrednictwem biblioteka SDL udostępnia swoje funkcje. Wykorzystywanie SDL-API ułatwia interfejs graficzny SDL-GUI. Do funkcji dostępnych z poziomu interfejsu graficznego należą:

- 1) odczytywanie i wyświetlanie schematu relacyjnej bazy danych w postaci drzewa zawierającego tabele z kolumnami oraz widoki,
- 2) generowanie ontologii w języku OWL, na podstawie schematu relacyjnej bazy danych,
- 3) przypisywanie pojęć ontologicznych kolumnom tabel w relacyjnej bazie danych,
- 4) transformowanie ontologii OWL na skrypt w języku Jess służący do wnioskowania w przód,
- 5) transformowanie ontologii OWL na skrypt w języku Jess służący do wnioskowania wstecz.

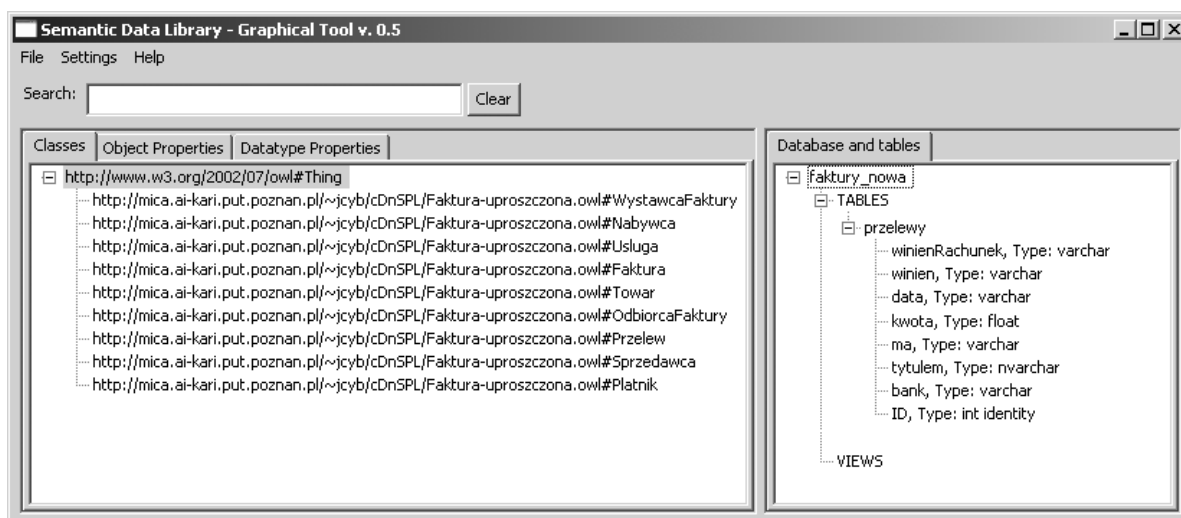
Do innych funkcji realizowanych przez bibliotekę SDL z poziomu interfejsu programistycznego należą:

- a) generowanie zapytań w języku Jess na podstawie reguł SWRL,
- b) sterowanie maszyną Jess z poziomu języka Java,
- c) zarządzanie procesem wnioskowania w przód,
- d) zarządzanie procesem wnioskowania wstecz,
- e) zarządzanie procesem wnioskowania hybrydowego oraz
- f) realizowanie zapytań zawierających pojęcia ontologiczne.

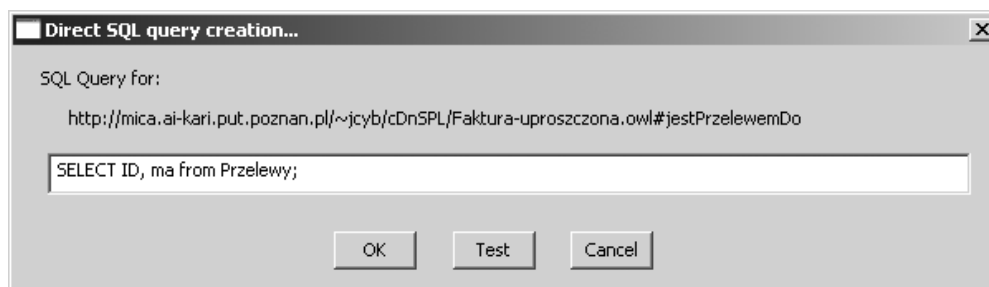
Na rys. 4. przedstawiono przykładowe okno obrazujące interfejs graficzny systemu. W lewym panelu widnieje zestawienie pojęć przykładowej ontologii odniesienia (omówiono ją pokrótce w punkcie 4.), a w prawym – elementy schematu relacyjnej bazy danych.

Najprostszą ontologię można wygenerować automatycznie ze schematu bazy danych, wtedy następuje równoczesne przyporządkowanie tabelom i atrybutom tabel bytów z ontologii. Na rys. 4. pokazano „zewnątrzną” ontologię wczytaną przez OWL-API/SDL-API. Mając taką ontologię, inżynier wiedzy dokonuje przyporządkowania elementom schematu bazy relacyjnej bytów ontologicznych, na przykład tabeli *Przelewy* odpowiada pojęcie *Przelew*. W bazie relacyjnej dane agregowane w tabelach opatrzone są nazwami (nazwy atry-

butów). W ontologii reprezentujemy powiązania pomiędzy atrybutem będącym kluczem głównym i innym atrybutem tabeli, stosując relacje binarne (typu *datatype property*). Relacje takie zachodzą pomiędzy obiektami klasy odpowiadającej tabeli a danymi typu zapisywanego w kolumnie. Na przykład, relacji *jestPrzelewemDo* zachodzącej pomiędzy przykładem pojęcia *Przelew* i daną typu *string*, reprezentującą nazwę właściciela rachunku, odpowiada powiązanie łączące atrybut *ID* i atrybut *ma* w tabeli *Przelewy*. Opisaną odpowiedniość inżynier wiedzy specyfikuje za pomocą zapytania SQL. Zapytanie odpowiadające rozważanemu przykładowi zilustrowano na rys. 5.



Rys. 4. Interfejs graficzny biblioteki SDL  
Fig. 4. Graphical User Interface of SDL library



Rys. 5. Przykładowe odwzorowanie relacji z ontologii na relacje wiążące atrybuty  
Fig. 5. Example mapping between ontology relations and relations between database attributes

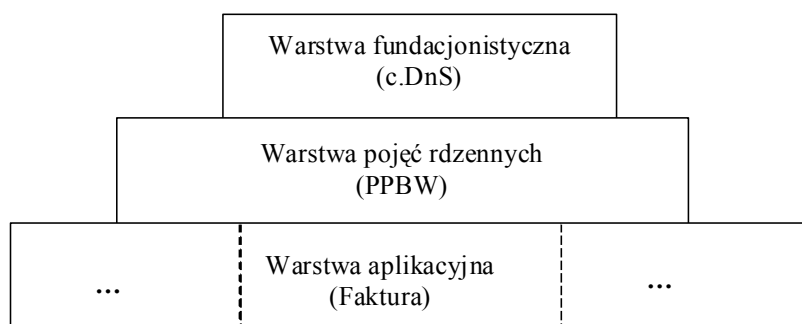
## 4. Ontologia odniesienia

Jednym z trzech elementów wykorzystywanych w opisywanej metodzie jest ontologia odniesienia. W stosunku do niej formułujemy następujące wymagania:

- a) reprezentowanie kilku warstw semantycznych, a mianowicie: najogólniejszych kategorii pojęć, pojęć ogólnych (rdzennych) wskazanej dziedziny oraz pojęć szczegółowych dziedziny i aplikacji,

b) istnienie wersji sformalizowanej w języku o odpowiedniej sile wyrazu, możliwym do przetworzenia w systemie.

Spełniając wymagania, zbudowano strukturę warstwową zobrazowaną na rys. 6. Warstwę najwyższą stanowi ontologia fundacjonistyczna *konstruktywnych deskrypcji i sytuacji c.DnS* [5, 6], która jest specjalizowana przez ontologię pojęć rdzennych w dziedzinie procesów przestępczych i procedur ich wykrywania *PPBW* [10], a ta z kolei przez ontologie aplikacyjne, wśród których znajduje się ontologia dwóch typów dokumentów handlowo-finansowych, a mianowicie faktur i przelewów [11]. Wszystkie rozważane warstwy ontologii zaimplementowano w języku OWL.



Rys. 6. Struktura warstwową ontologii odniesienia  
Fig. 6. The layered structure of the reference ontology

Podejście, w którym dla pojęć szczegółowych, bezpośrednio odpowiadających semantyce danych z bazy relacyjnej, można wskazać pojęcia je subsumujące, stanowi znakomitą podstawę procesu integrowania danych pochodzących z różnych źródeł. Dobór odpowiedniej ontologii fundacjonistycznej jest jedną z ważniejszych decyzji podejmowanych w trakcie tworzenia ontologii. Scharakteryzujemy pokrótce ontologię *c.DnS*.

W rozważanym paradygmacie modeluje się dwie składowe: ontologię bytów bazowych (tego, „co istnieje realnie”) i ontologicznie zreifikowane środki (nazywane *bytami schematycznymi*) do opisywania bytów bazowych z punktu widzenia pewnych bytów zwanych *agentami postrzegającymi*. Uzyskujemy w ten sposób pewien rodzaj kontekstualizacji bytów ontologicznych. Byty bazowe, które są klasyfikowane (opisywane) przez byty schematyczne traktuje się jako *ugruntowane ontologicznie*.

Ontologia *c.DnS* ma następujące zalety:

1. Ontologia ma dużą siłę wyrazu i łatwo ją zastosować do modelowania różnych dziedzin.
2. Dobrze nadaje się do modelowania dziedzin, w których większość bytów ma charakter niefizyczny.
3. Za pomocą rozważanej ontologii można modelować wiedzę epistemiczną.
4. Uproszczona wersja ontologii wyrażona w języku OWL jest udostępniana publicznie [5].

Modelowanie w paradygmacie *c.DnS* polega w ogólności na zdefiniowaniu relacji (1). Na bazie struktury (1) definiuje się binarne (lub trójargumentowe, z argumentem czasowym) projekcje, specyfikujące powiązania pomiędzy parami pojęć.

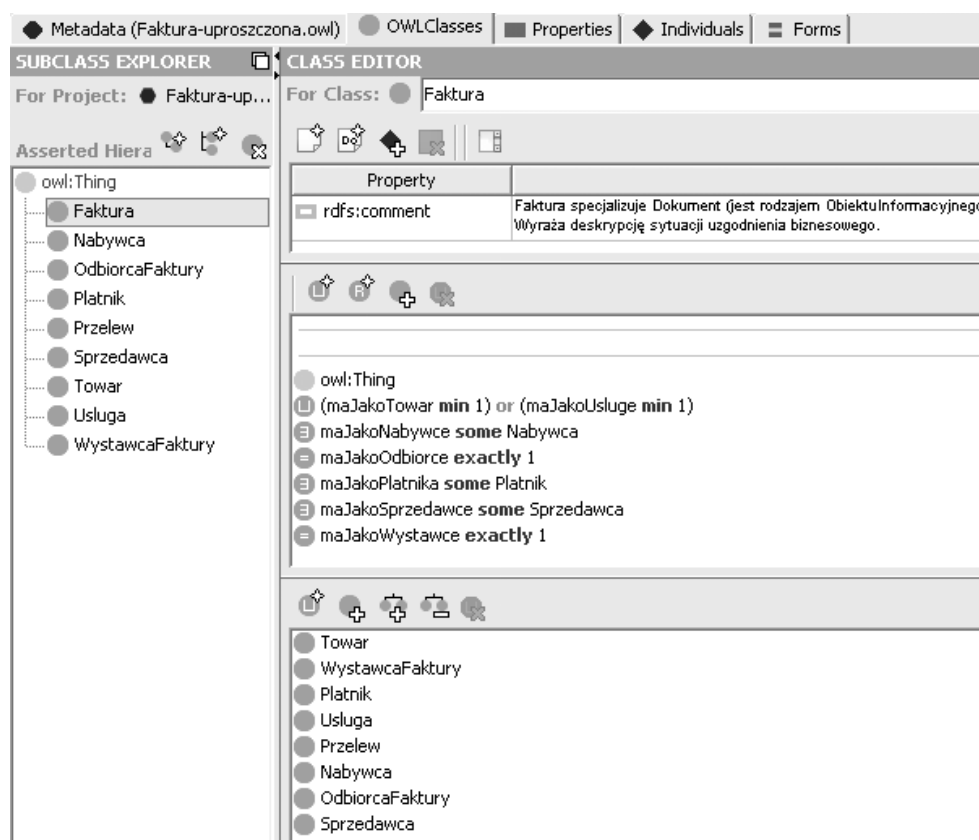


$$c.DnS = \langle S, E, A, D, C, K, I \rangle \quad (1)$$

Zmienna  $S$  oznacza *sytuację* będącą rodzajem bytu schematycznego, który reifikuje ontologicznie pewien stan rzeczy. Sytuacja jest konstytuowana (za pomocą formalnej relacji ustanawiania) przez *ugruntowane byty*  $E$  pochodzące z ontologii bytów bazowych. Sytuacja  $S$  jest zlokalizowana w *przedziale czasowym*  $T_S$  (jest on rodzajem *konceptu*  $C$ , który służy do klasyfikowania bytów bazowych). Sytuacja jest postrzegana przez pewnego *agenta społecznego*  $A$  (na którego rzecz działa pewien *agent fizyczny*, z obszaru bytów bazowych), który w czasie  $T_D$  tworzy jej *deskrypcję*  $D$ , spełnianą przez sytuację  $S$ . Deskrypcja może być *przyjmowana* lub *współdzielona* przez *kolektyw*  $K$  agentów społecznych. Deskrypcja  $D$  jest wyrażana przez *obiekt informacyjny*  $I$ , który może mieć różne reprezentacje fizyczne (byty bazowe).

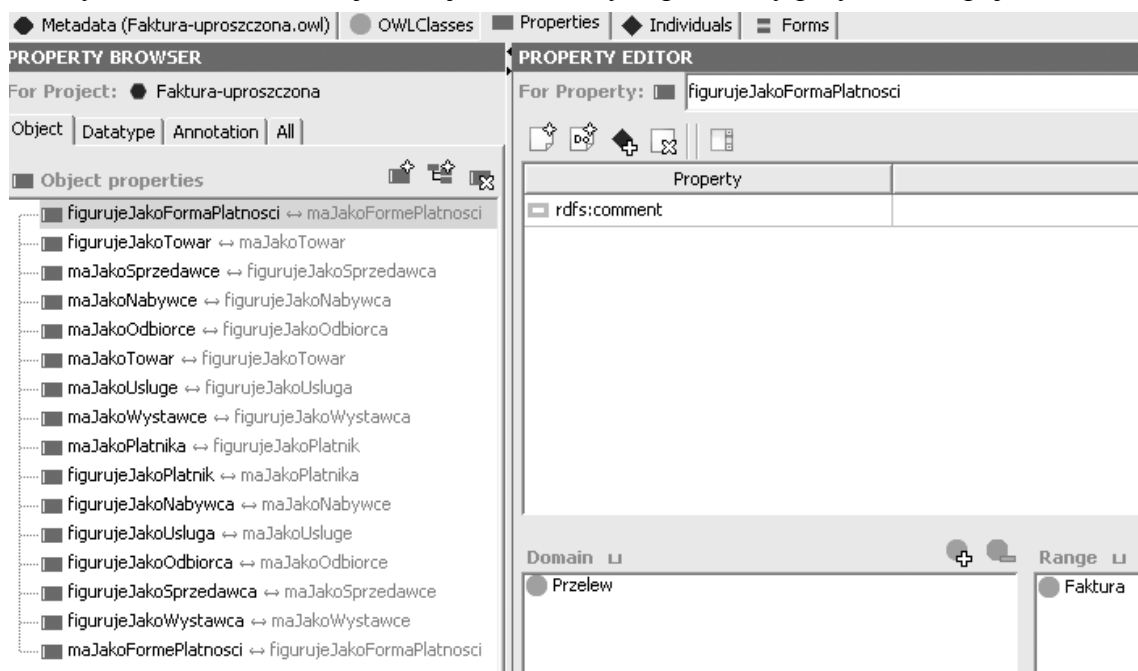
Pojęcia omówionej warstwy ontologii mogą być uszczegółowione przez pojęcia warstwy rdzennej (np. PPBW). Na przykład, możemy rozważać pewną sytuację biznesową, w której ramach toczy się proces przestępczy. Elementem tej sytuacji są: uzgodnienie handlowe dotyczące transakcji kupna-sprzedaży, operacja finansowa przekazania kwoty pieniędzy, wydanie, transport i odbiór towaru lub wykonanie usługi. Agentem śledzącym sytuację jest prokurator lub policjant, który tworzy deskrypcję tej sytuacji w postaci rozpoznanego planu procesu przestępczego (wersję śledczą, a może ich być kilka). Elementem tego planu jest rozważane uzgodnienie handlowe oraz operacja finansowa (które również toczą się zgodnie z pewnym subplanem). Deskrypcję uzgodnienia handlowego stanowi dokument handlowy (rodzaj obiektu informacyjnego) w postaci faktury, a deskrypcję bankowego transferu pieniędzy opisują wyciągi z rachunków bankowych. W punkcie 5. prezentujemy przykład tworzenia konkretnej aplikacji (bazy wiedzy) z zastosowaniem reguł analitycznych wykorzystujących dokumenty handlowe i finansowe. Zatem, w dalszej części zaprezentujemy przykładową ontologię aplikacyjną, reprezentującą semantykę faktur i przelewów bankowych [11].

Na rys. 7. przedstawiono zestaw pojęć ontologicznych. Każde z nich jest subsumowane przez pojęcie warstwy wyższej.



Rys. 7. Zestawienie pojęć ontologii dotyczącej faktur i przelewów  
 Fig. 7. The concepts of the invoice and bank account transfers ontology

Na rys. 8. zawarto definicję relacji zachodzących pomiędzy przykładami pojęć.



Rys. 8. Relacje typu *object properties* ontologii faktur i przelewów Relacje typu *object properties* ontologii faktur i przelewów  
 Fig. 8. The object properties of the invoice and bank account transfers ontology

Na rys. 9 zobrazowano relacje zachodzące pomiędzy przykładami pojęć i danymi wskazanymi typów.



Rys. 9. Relacje typu *datatype properties* ontologii faktur i przelewów

Fig. 9. The datatype properties of the invoice and bank account transfers ontology

## 5. Przykład zastosowania metody

Zilustrujemy opracowaną metodę przez podanie sposobu tworzenia za pośrednictwem biblioteki SDL, bazy wiedzy wspomagającej analizowanie przelewów bankowych. Analiza przelewów jest jednym z elementów procedury wykrywania przestępstw gospodarczych. Przez zadawanie zapytań reprezentujących metody analityczne, a odnoszących się do danych zgromadzonych w bazie relacyjnej, następuje gromadzenie wiedzy w bazie wiedzy.

W prezentowanym przykładzie wykorzystano relacyjną bazę danych dotyczącą przelewów bankowych. Baza ta zawiera opis ponad 6500 przelewów dokonywanych pomiędzy czterystoma firmami. Zawiera ona dane „wybielone” wykorzystane w rzeczywistym śledztwie. Na schemat bazy danych składa się jedna tabela *Przelewy* zawierająca osiem kolumn. Przedstawiono go w tabeli 1.

Tabela 1

Schemat relacji *Przelewy*

ID	winienRachunek	winien	data	kwota	ma	tytułem	bank
----	----------------	--------	------	-------	----	---------	------

Poszczególne kolumny oznaczają:

- *ID* jest numerem identyfikującym przelew w bazie danych (kluczem głównym),
- *winienRachunek*, to numer rachunku, z którego zostały przelane pieniądze,
- *winien* oznacza nazwę właściciela rachunku, z którego zostały przelane pieniądze,
- *data* jest datą dokonania przelewu,
- *kwota*, to kwota przelewu,

- *ma* jest nazwą właściciela rachunku, na który zostały przelane pieniądze,
- *tytułem* określa przyczynę dokonania przelewu,
- *bank* jest nazwą banku lub oddziału banku, z którego dokonano przelewu.

Pierwszym krokiem opracowanej metody jest wygenerowanie ontologii pierwotnej na podstawie schematu relacyjnej bazy danych. W tym celu wykorzystuje się funkcję SDL-GUI – *Generate Simple OWL Ontology...* Ontologia jest generowana zgodnie z zasadą: nazwa pojęcia (klasy) to nazwa tabeli bazy, a na podstawie nazw kolumn (atrybutów) generowane są nazwy właściwości typu *datatypeProperty*, poprzedzone nazwą tabeli i znakiem myślnika. Wygenerowana ontologia pierwotna będzie więc zawierać jedną klasę *Przelewy* oraz osiem właściwości: *Przelewy-ID*, *Przelewy-wininenRachunek*, *Przelewy-winien* itd. Tak prosta ontologia nie zawsze jest użyteczna (zwłaszcza jako baza do integrowania danych pochodzących z różnych źródeł), dlatego w systemie udostępniono funkcję wzbogacania ontologii pierwotnej lub wręcz zastępowania jej ontologią odniesienia o większej sile wyrazu. W przykładzie zastosowano ontologię omówioną w punkcie 4.

W kolejnym kroku wygenerowano docelowy model bazy wiedzy w języku systemu wnioskującego Jess. Utworzone w ten sposób skrypty (do wnioskowania wstecz oraz w przód) w języku Jess wczytano do systemu wnioskowania hybrydowego realizowanego przez bibliotekę SDL. Krok ten kończy budowanie modelu konceptualnego przykładowej bazy wiedzy.

Ostatni etap polega na wypełnianiu bazy wiedzy danymi pochodzącymi z bazy relacyjnej. Ekspert-analityk sformułował trzy zapytania. Założył, że odpowiedzi na nie będą pożądanymi faktami, wywnioskowanymi z danych relacyjnych, zapisywanymi następnie w bazie wiedzy. Zapytania sformułowano w wersji graficznej języka zapytań i przedstawiono je na rysunkach 10, 11 i 12.

Pierwsze (rys. 10) z zapytań testuje istnienie jednokierunkowego łańcucha różnych firm (stała TRAWLOLLEX i zmienne  $?fi$ ,  $i = 2, \dots, 6$ ) przelewających pieniądze (numery rachunków w łańcuchu są różne). Długość łańcucha ustawiono arbitralnie na sześć. Dаты przelewów muszą być uszeregowane rosnąco (zmienne  $?di$ ,  $i = 1, \dots, 5$ ), natomiast kwoty (zmienne  $?ki$ ,  $i = 1, \dots, 5$ ) na kolejnych przelewach są równe lub większe od poprzednich. Znalezienie takiego łańcucha obrazuje realną sieć pośredników finansowych (zmienne  $?fi$ ,  $i=2, \dots, 6$ ), przelewy mogą być zapłatą za ten sam towar, lecz nie muszą.

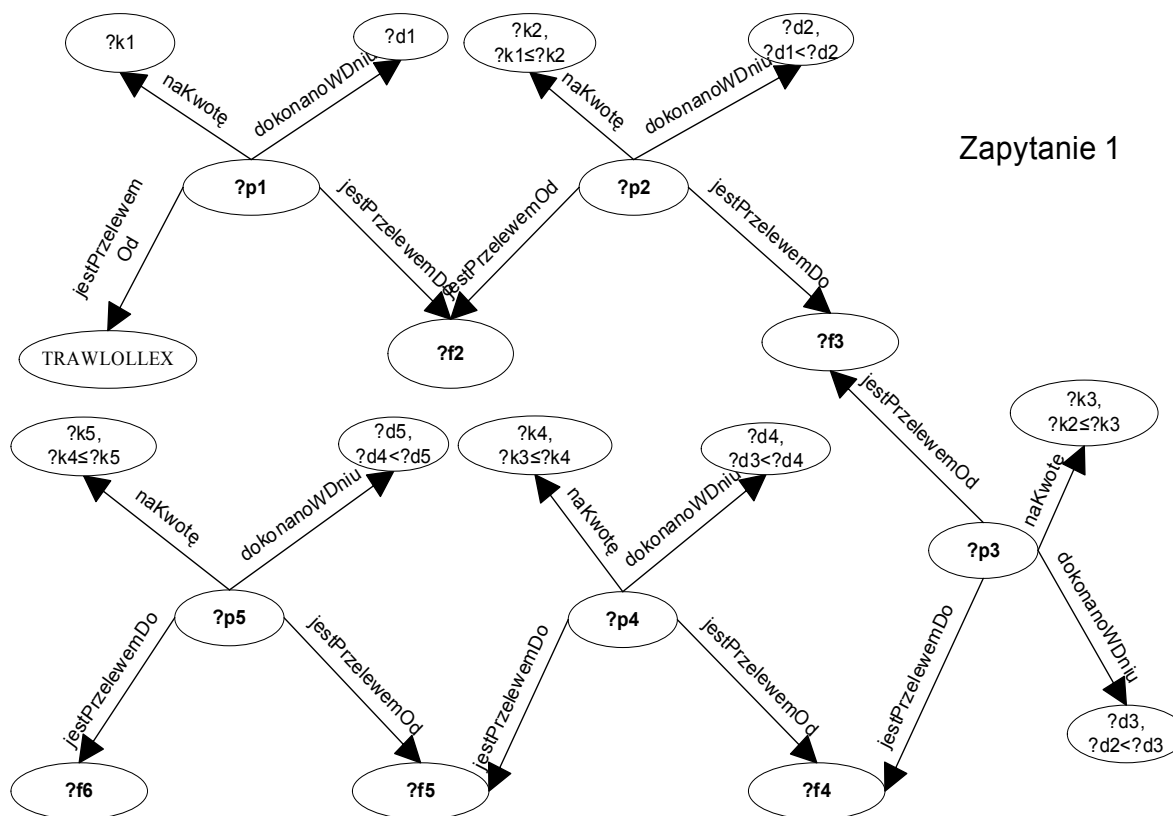
Drugie zapytanie (rys. 11) dotyczy zamkniętego łańcucha rachunków bankowych firm, w którym pieniądze najpierw wypływają z firmy TRAWLOLLEX, a następnie przez sieć pośredników wracają do niej. Znalezienie takiego łańcucha może sugerować popełnienie przestępstwa karno-skarbowego, czyli karuzeli VAT. Dаты przelewów muszą być uszeregowane rosnąco, natomiast kolejne kwoty muszą być mniejsze lub równe (malejące kwoty oznaczają prowizję dla firm uczestniczących w procederze przestępczym).

Trzecie zapytanie (rys. 12) dotyczy wymiany pieniędzy pomiędzy dwiema firmami na dokładnie tę samą kwotę. W tym przypadku w całej bazie danych wyszukujemy zbiory 2 dowolnych firm, które wzajemnie do siebie przelewały te same kwoty, przy czym daty przelewów muszą być różne. Znalezienie takich przelewów sugeruje fikcyjny przepływ towaru.

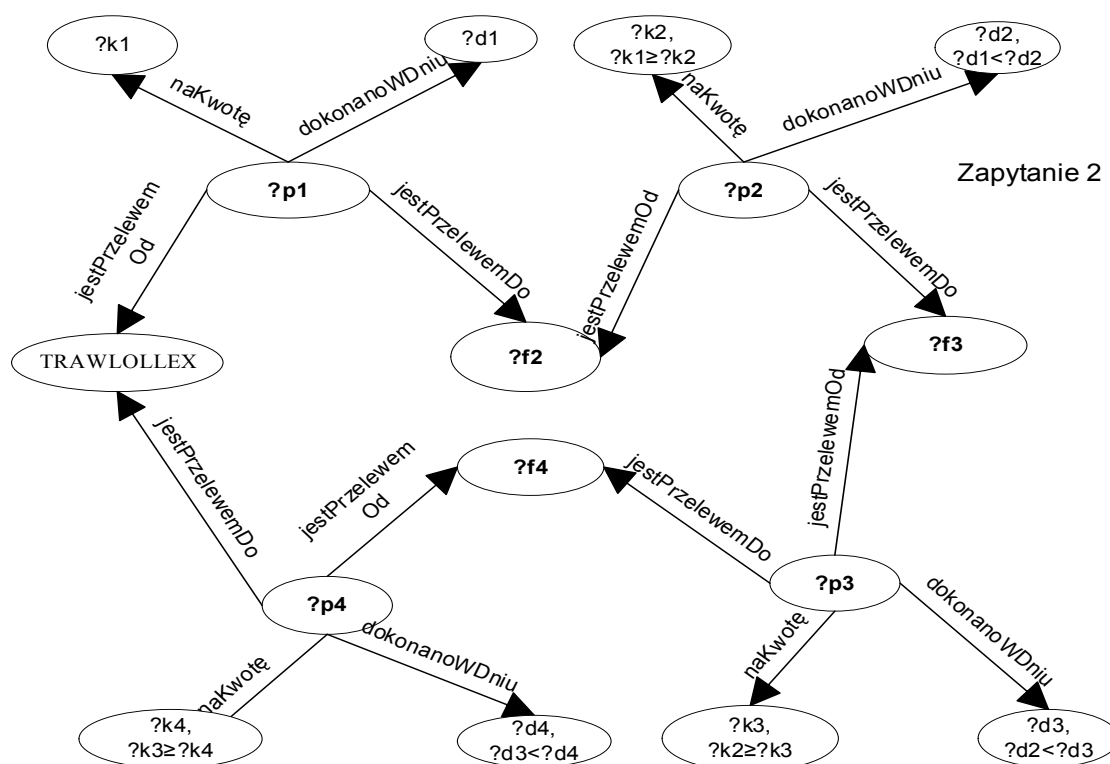
Czasy realizacji zapytań zestawiono w tabeli 2.

Tabela 2

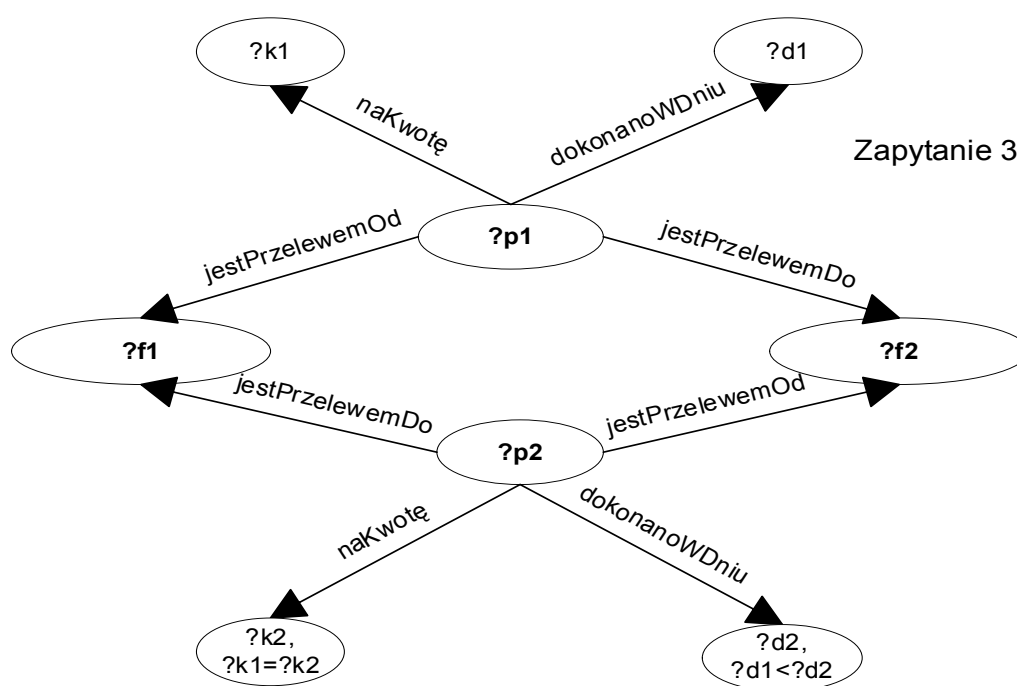
Czasy realizacji przykładowych zapytań		
Zapytanie	SDL [ms]	Liczba wyników
1	67 515	15 464
2	51 375	711
3	179 297	244



Rys. 10. Reprezentacja graficzna pierwszego zapytania  
Fig. 10. Graphical representation of the first query



Rys. 11. Reprezentacja graficzna drugiego zapytania  
 Fig. 11. Graphical representation of the second query



Rys. 12. Reprezentacja graficzna trzeciego zapytania  
 Fig. 12. Graphical representation of the third query

Istotnym elementem realizacji zapytań jest wczytywanie danych pochodzących z relacyjnej bazy danych. W wielu przypadkach proces ten zabiera od 50 do 90% czasu potrzebnego na udzielenie odpowiedzi. Dlatego też, mimo że powyższe wyniki są akceptowalne, to planuje się przeprowadzenie badań nad opracowaniem bardziej efektywnych metod odpowiedzialnych za dostęp do relacyjnej bazy danych. Równolegle trwają prace nad sformalizowaniem metody wnioskowania hybrydowego, którego efekt pozwoli jeszcze efektywniej zarządzać procesem realizacji zapytań. Wydajność biblioteki SDL [2] sugeruje, że istnieją klasy zapytań, w których jej efektywność jest porównywalna z efektywnością języka SQL i jednocześnie tworzenie zapytań jest o wiele prostsze. Klasy te możliwe będą do określenia dopiero wtedy, gdy zostanie opracowany formalny model realizacji zapytań oraz procesów wnioskowania.

## 6. Podsumowanie

W pracy przedstawiono zastosowanie biblioteki SDL do przyrostowego tworzenia bazy wiedzy, przy wykorzystaniu bazy danych, ontologii odniesienia i wiedzy eksperta, dotyczącej rozwiązywania problemów z użyciem zgromadzonych danych. Wykonane eksperymenty wskazują potrzebę oszacowania klasy zapytań odczytujących dane z bazy danych, tak by był to proces efektywny. Do wykonania pozostaje także sformalizowanie metody i oszacowanie jej złożoności obliczeniowej dla wskazanej klasy zapytań. Zastosowany sposób budowania ontologii odniesienia okazał się odpowiedni zarówno ze względu na jego uniwersalność, jak i możliwość łatwego dołączania szczegółowych siatek pojęć.

Pracę wykonano w ramach programu 0014/R/2/T00/06/02 MNiSzW, Polska Platforma Bezpieczeństwa Wewnętrznego oraz grantu DS-45-083/2009 finansowanego przez Politechnikę Poznańską.

## BIBLIOGRAFIA

1. Bąk J., Jędrzejek C.: Wnioskowanie hybrydowe w relacyjnej bazie danych wykorzystujące podejście semantyczne. WKŁ, Rozdział 27, Warszawa 2008, s. 333÷348.
2. Bąk J., Jędrzejek C., Querying relational databases using ontology, rules and Jess reasoning engine. *Studies in Automation and Information Technology*, Vol. 33, 2008, s. 25÷44.
3. Charles F., Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. *Artificial Intelligence*, 19, 1982, s. 17÷37.

4. Dean M., Schreiber G. (eds): OWL Web Ontology Language Reference. <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>
5. Gangemi A.: The OWL version of the Constructive Descriptions and Situations ontology. <http://www.loa-cnr.it/ontologies/collInt.owl>.
6. Gangemi A., Lehmann J., Catenacci C.: Norms and plans as unification criteria for social collectives. W: Materiały Dagstuhl Seminar 07122, Normative Multi-agent Systems, G. Boella, L. v. d. Torre, H. Verhagen (Eds.), Vol. II, 2007, s. 48÷87. <http://drops.dagstuhl.de/opus/volltexte/2007/910>.
7. Horrocks I. et al.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>.
8. Jena2, <http://jena.sourceforge.net/>.
9. Jess, <http://jessrules.com/>.
10. Ontologia procesów przestępczych i procedur ich wykrywania, <http://mica.ai-kari.put.poznan.pl/~jcyb/PPBW.owl> (dostępne 31.01.2009).
11. Ontologia faktur i przelewów, <http://mica.ai-kari.put.poznan.pl/~jcyb/Faktura-uproszczona.owl> (dostępne 31.01.2009).
12. Poggi A. et al.: Linking Data to Ontologies, Journal on Data Semantics Vol. X, 2008, s. 133÷173.

Recenzent: Dr hab. inż. Krzysztof Goczyla, prof. Pol. Gdańskiej

Wpłynęło do Redakcji 31 stycznia 2009 r.

## Abstract

We show how to incrementally create a knowledge base on the basis of the relational database, the suitable reference ontology and the expert knowledge, which concerns the problem solving methods that use the relational data. Our method contains several steps. In the first one the database schema is processed in order to obtain a primary conceptual model expressed in the form of an OWL ontology. This ontology may be updated by a knowledge engineer on the basis of a suitable reference ontology. The requirements for such an ontology are that it should be expressive enough and formalised. We built such an ontological structure [10, 11] by ourselves, and formalised it in the OWL language [4]. The updated ontology may be additionally extended by rules written in SWRL [7], and then translated into the language of a Jess [9] inference machine. The ontological concepts and relations are mapped on the relational tables and attributes. This step ends the creation of a knowledge base conceptual



model. Next, the base is populated by the relational data by means of inferences done according to queries, in which the ontological entities are used. Additionally, we invented the graphical language to express the queries in order to make the language available to non-professionals. The method is implemented with the use of the SDL library [1, 2] that was also developed by us. The class of efficiently processed queries should be established. Also, the method as a whole needs formalisation and the computation of its complexity.

### **Adresy**

Jarosław BAŁ: Politechnika Poznańska, Instytut Automatyki i Inżynierii Informatycznej,  
Pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Polska, jaroslaw.bak@put.poznan.pl.

Jolanta CYBULKA: Politechnika Poznańska, Instytut Automatyki i Inżynierii Informatycznej,  
Pl. M. Skłodowskiej-Curie 5, 60-965 Poznań, Polska.