

Roman SIMIŃSKI
Uniwersytet Śląski, Instytut Informatyki

JEDNOSTKI DECYZYJNE JAKO NARZĘDZIE WIZUALIZACJI REGUŁOWEJ BAZY WIEDZY

Streszczenie. Artykuł przedstawia koncepcję wykorzystania jednostek decyzyjnych w zadaniu wizualizacji struktury regułowej bazy wiedzy. W artykule przedstawiono podstawowe metody wykorzystania grafów w zagadnieniu wizualizacji oraz opis koncepcji jednostek decyzyjnych. Przedyskutowano metody wykorzystania jednostek w systemie *kbBuilder* oraz przedstawiono przyszłe zadania, związane z implementacją nowych wersji oprogramowania.

Słowa kluczowe: regułowe bazy wiedzy, jednostki decyzyjne, wizualizacja

THE DECISION UNITS AS THE TOOL FOR RULE KNOWLEDGE BASE VISUALIZATION

Summary. The paper presents conception of visualization the structure of rule knowledge bases using decision units. The basic methods of utilization the graphs in rule base visualization task as well as the idea of using decision units have been presented in this paper. The methods of using decision units in *kbBuilder* system have been also discussed in this paper together with description of the future implementation task for the new software version.

Keywords: rule knowledge bases, decision units, visualization

1. Wprowadzenie

Systemy wykorzystujące regułowe bazy wiedzy oraz metody automatycznego wnioskowania są znanymi od wielu lat narzędziami rozwiązywania problemów niepodatnych na algorytmizację. Systemy takie przeżywają aktualnie swój cichy renesans, a jedną z przyczyn takiego stanu rzeczy jest intensywny rozwój metod eksploracji danych, który zaowocował dużą

łatwością automatycznego pozyskiwania reguł opisujących wiedzę odkrywaną w danych. Możliwość automatycznego pozyskiwania zweryfikowanych zbiorów reguł znacząco wpłynęła na niespektakularnie zauważalny wzrost zastosowań systemów z regułowymi bazami wiedzy, wpłynęła również na znaczący wzrost rozmiarów baz regułowych. Liczebności baz rzędu setek reguł są codziennością, liczebności rzędu tysięcy nie są rzadkością.

Tak duże bazy stanowią wyzwanie dla inżynierii wiedzy – modyfikacje, weryfikacja, pielęgnacja i rozwój takich baz stają się zagadnieniami złożonymi w sensie teoretycznym jak i kłopotliwymi w sensie implementacyjnym. Już dla baz o liczebności kilkudziesięciu reguł wizualizacja – nawet na współczesnych monitorach o dużym rozmiarze – jest problematyczna, a typowo stosowane metody skalowania nie zapewniają właściwego komfortu pracy inżyniera wiedzy. Istotą badań, których wycinek został zaprezentowany w tym artykule, jest poszukiwanie meta-reprezentacji dla regułowej bazy wiedzy, pozwalającej m.in. na skuteczną, efektywną i wygodną dla inżyniera wiedzy wizualizację struktury regułowej bazy wiedzy. W artykule tym wskazuje się jednostki decyzyjne jako metamodel regułowej bazy wiedzy, stawiając tezę, że zastosowanie jednostek decyzyjnych jako modelu regułowej bazy wiedzy umożliwi rozwiązanie problemu wizualizacji dużych baz wiedzy, pozwalając na opracowanie efektywnych i wygodnych dla użytkownika metod wizualnie zorientowanej inspekcji, weryfikacji i edycji baz regułowych.

Celem tego artykułu jest prezentacja aktualnego stanu prac związanych z wykorzystaniem jednostek decyzyjnych w inżynierii wiedzy, koncentrująca się na problemach wizualizacji i wizualnie zorientowanych inspekcji i edycji regułowych baz wiedzy. Z uwagi na ciągle rosnące rozmiary takich baz wiedzy problem ten stanowi interesujące wyzwanie zarówno w wymiarze badawczym, jak i implementacyjnym. Podstawą do sformułowania koncepcji jednostek decyzyjnych były doświadczenia zdobyte w trakcie prac badawczych, realizowanych w Instytucie Informatyki Uniwersytetu Śląskiego, wsparte doświadczeniami praktycznymi, wyniesionymi ze współudziału autora niniejszego artykułu w realizacji elementów systemu *PC-Shell* [5], będącego później składową pakietu Sphinx [6], w tym szczególnie prac nad systemem *CAKE* [12]. Istotną rolę odegrały również spostrzeżenia wynikające z analizy wyników prac realizowanych przez innych autorów lub we współpracy z nimi, np. baz wiedzy dla analiz finansowych [7], kredytowych, inwestycyjnych, marketingowych [15], diagnostyki zaburzeń depresyjnych [4] czy wspomagania diagnozowania w zakresie neurologii dziecięcej [18]. W opracowaniu koncepcji jednostek decyzyjnych ważną rolę odegrały również doświadczenia o charakterze praktycznym, związane z realizacją systemów *Infer* [8] oraz prototypowej wersji systemu *kbBuilder*, w którym zastosowano tę koncepcję [17].

Artykuł rozpoczyna przegląd metod graficznej reprezentacji regułowych baz wiedzy, zawierający krótką dyskusję na temat wad i zalet prezentowanych rozwiązań. W kolejnej części

artykułu przedstawiona została koncepcja jednostek decyzyjnych, przy czym wyeksponowane zostały ich intuicyjny charakter oraz możliwości wizualizacyjne. Stosowne opisy formalne jednostek decyzyjnych zostały pominięte ze względu na ograniczony rozmiar niniejszego artykułu oraz nikłe znaczenie strony formalnej zagadnienia dla istoty tematu. Artykuł zamyka prezentacja wybranych elementów systemu wspomagania inżynierii wiedzy *kbBuilder*, wykorzystującego jednostki decyzyjne w zagadnieniach wizualizacji i wizualnie zorientowanych inspekcji i edycji regułowych baz wiedzy. Artykuł nie przewiduje prezentacji eksperymentów obliczeniowych, a jedynie krótką informację o wynikach testów akceptacyjnych zrealizowanego oprogramowania oraz jego porównanie do innych rozwiązań w tej grupie oprogramowania.

2. Regułowa baza wiedzy w postaci grafów – analiza możliwości

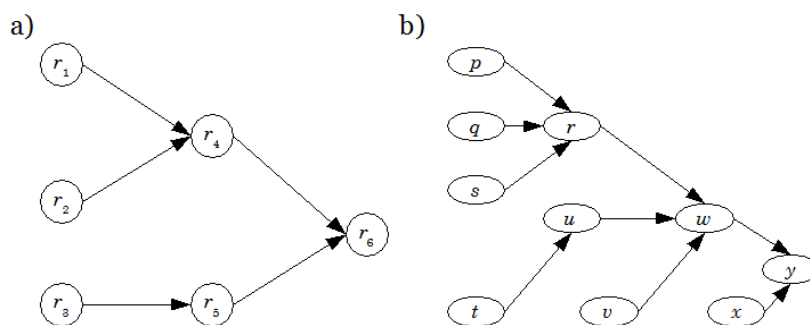
Organizacja struktury regułowej bazy wiedzy w postaci grafu jest naturalnym środkiem poszukiwania skutecznej metody wizualizacji takiej bazy. Wykorzystanie grafów może przyjąć różną formę; przedstawiony w dalszej części tego artykułu przykład ma za zadanie ilustrację toku rozumowania, który doprowadził do sformułowania koncepcji jednostek decyzyjnych, przedstawionej w następnym rozdziale. Dla uproszczenia rozważań oraz zachowania klarowności rysunków w przykładzie tym wykorzystano reguły przyjmujące postać implikacji w rachunku zdań. Zakłada się, że dana jest baza wiedzy składająca się z sześciu reguł:

$$\begin{aligned}r_1 &: p \wedge q \rightarrow r \\r_2 &: s \rightarrow r \\r_3 &: t \rightarrow u \\r_4 &: r \rightarrow w \\r_5 &: u \wedge v \rightarrow w \\r_6 &: w \wedge x \rightarrow y\end{aligned}$$

Rys. 1. Przykładowa baza wiedzy
Fig. 1. An example rule base

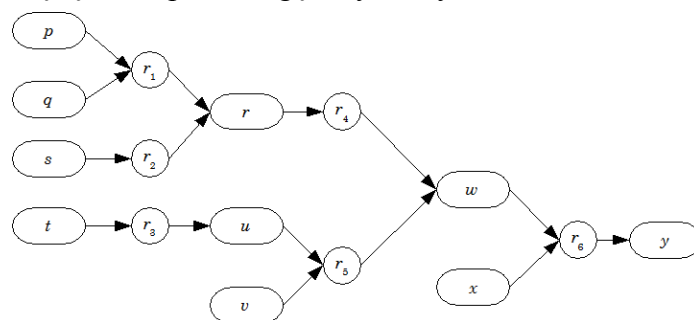
Na rys. 2a przedstawiono pierwszą możliwą formę prezentacji, zakładającą organizację bazy wiedzy w postaci grafu skierowanego, którego wierzchołki reprezentują reguły, a krawędzie występujące pomiędzy regułami – powiązania. Pomiędzy dwoma wierzchołkami takiego grafu wystąpi skierowana krawędź. Jeżeli konkluzja reguły, reprezentowanej przez pierwszy wierzchołek, występuje jako warunek w przesłance reguły, reprezentowanej przez wierzchołek drugi, diagram taki nazywany będzie roboczo *diagramem związków reguł*. Podstawową cechą takiej reprezentacji grafowej jest jej ogólność: korzystne jest – abstrahując od szczegółów – pokazanie związków występujących pomiędzy regułami; jednocześnie taki dia-

gram nie niesie żadnych dodatkowych informacji, pozwalających na stwierdzenie, jakie literały uczestniczą w tych związkach [11, 1, 3].



Rys. 2. Dwie proste, grafowe reprezentacje bazy wiedzy
Fig. 2. Two simple graph representations of rule base

Wady nadmiernej ogólności nie ma diagram przedstawiony na rys. 2b, prezentujący omawianą bazę wiedzy z uwzględnieniem jedynie literałów reguł – przykładowa reprezentacja przybiera postać grafu skierowanego, którego wierzchołki reprezentują literały występujące w warunkach i konkluzjach reguł, a krawędzie powiązania pomiędzy tymi literałami określone są regułami; graf taki zwany będzie dalej *diagramem związków literałów* [2]. Taka reprezentacja pokazuje szczegółowe zależności pomiędzy literałami, jasno wskazując zależności pomiędzy opisywanymi przez nie elementami dziedziny problemu, gubi się jednak informacje o regułach. W sensie ogólnym diagram związków literałów może być identyczny z drzewem decyzyjnym, a po oznakowaniu odpowiednich par krawędzi – z drzewami AND-OR. Należy jednak pamiętać, że omawiane diagramy mają modelować zawartość dowolnej bazy regułowej, również takiej, która zawiera anomalie – np. zależności cykliczne, a w takim przypadku otrzymany diagram nie będzie drzewem. Zatem reprezentacje bazujące na strukturach drzewiastych nie mogą stanowić uniwersalnego, ogólnego modelu dowolnej bazy regułowej i nie będą brane pod uwagę w tym artykule.



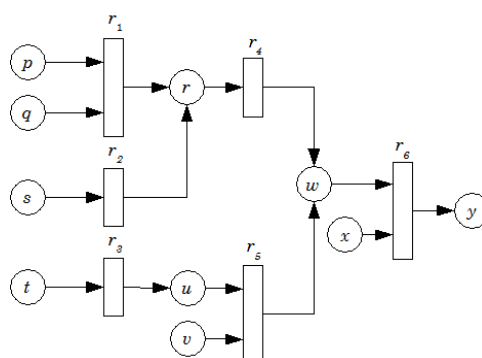
Rys. 3. Diagram struktury reguł
Fig. 3. The rules structure diagram

Istnieje możliwość połączenia idei obu rozważanych diagramów; efektem będzie diagram nazywany tu *diagramem struktury reguł*. Przyjmuje on postać dwudzielnego grafu skierowanego, którego wierzchołki reprezentują odpowiednio reguły lub literały. Zorientowanie tego grafu może być dwojakie – od konkluzji do przesłanek dla *wnioskowania regresywnego* lub od

przesłanek do konkluzji dla *wnioskowania progresywnego*. Rysunek 3 prezentuje diagram struktury reguł dla rozważanej, przykładowej bazy wiedzy.

Reprezentacja bazy wiedzy w postaci diagramu struktury reguł rzeczywiście dostarcza takich informacji jak diagramy związku reguł i diagram związków literałów jednocześnie. Wydaje się, że diagram taki jest idealnym środkiem graficznej reprezentacji bazy wiedzy. Pozwala on na klarowną prezentację zależności ogólnych (zorientowanie na analizę powiązań pomiędzy wierzchołkami reguł) oraz zależności szczegółowych (zorientowanie na analizę powiązań pomiędzy wierzchołkami literałów). Diagramy tego typu mogą być wykorzystane w modelowaniu baz wiedzy, ich weryfikacji i wizualizacji [2]. Niestety diagram ten ma istotną wadę – już w bazach liczących kilkanaście reguł staje się nieczytelny. Przedstawienie bazy wiedzy, liczącej kilkadziesiąt reguł, na typowym monitorze komputerowym w postaci czytelnej dla użytkownika niestety w realnych warunkach nie jest możliwe, co zresztą potwierdza praca [2]. Jednak z punktu widzenia inżynierii wiedzy diagram struktury reguł jest interesujący i obiecujący, zatem celowe wydaje się znalezienie reprezentacji stanowiącej jego uogólnienie.

Diagram struktur reguł ma również inną cechę – w sensie swej struktury jest siecią Petriego [10]. W pracach [13, 14] zaprezentowano koncepcje reprezentacji i wykorzystania sieci Petriego w modelowaniu baz wiedzy – rys. 4 prezentuje sieć Petriego dla przykładowej bazy wiedzy, gdzie literały są odpowiednio *miejscami*, a reguły *tranzycjami*. Wnioski związane z wykorzystaniem sieci Petriego w modelowaniu regułowych baz wiedzy zawierają cytowane wcześniej prace.



Rys. 4. Baza wiedzy jako sieć Petriego

Fig. 4. The rule base as the Petri net

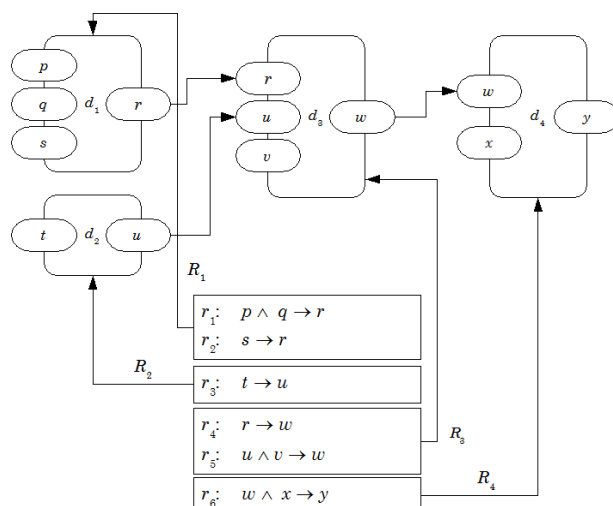
3. Baza wiedzy w postaci sieci jednostek decyzyjnych

Na przykładową bazę wiedzy można spojrzeć w odmienny sposób. Niech reguły zostaną zgrupowane według literału występującego w konkluzjach reguł – do danej grupy wchodzić będą tylko te reguły, które mają ten sam literał w konkluzji. Załóżmy również, że daną grupę

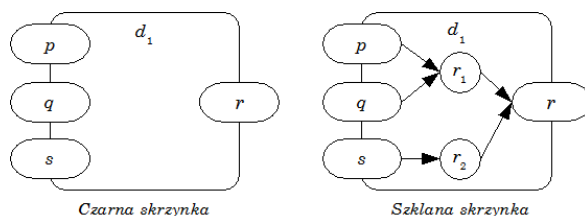
reguł zamknijemy wewnątrz pewnego obiektu, zwanego dalej *jednostką decyzyjną*. Przykładowa baza wiedzy zawiera w konkluzjach cztery różne literały: r , u , w , y , zatem powstaną cztery jednostki decyzyjne, oznaczone odpowiednio d_1 , d_2 , d_3 i d_4 . W każdej jednostce można wyodrębnić trzy elementy:

- zbiór reguł mający wspólną konkluzję;
- wspólny literał występujący w konkluzjach tych reguł;
- zbiór literałów występujących we wszystkich warunkach wymienionych reguł.

Do jednostki d_1 zostanie przypisany zbiór reguł $R_1 = \{r_1, r_2\}$, literał r występujący w konkluzjach tych reguł oraz zbiór literałów $\{p, q, s\}$, występujących w przesłankach reguł ze zbioru R_1 . W analogiczny sposób możemy zbudować pozostałe trzy jednostki decyzyjne: d_2 , d_3 i d_4 oraz przypisać im zbiory reguł, literał występujący w konkluzji oraz zbiór literałów występujących w przesłankach reguł, co ilustruje rys. 5, przedstawiający *diagram jednostek decyzyjnych*.



Rys. 5. Diagram jednostek decyzyjnych
Fig. 5. The decision units diagram



Rys. 6. Dwa spojrzenia na jednostkę decyzyjną
Fig. 6. Two-way interpretation of the decision unit

Diagram jednostek decyzyjnych zawiera informacje o powiązaniach występujących pomiędzy literałami, tak samo jak w przypadku diagramu związków literałów i diagramu struktury reguł. Jednak zestaw tych informacji został ograniczony jedynie do tych powiązań, które występują pomiędzy literałami z części warunkowych i konkluzji reguł niewchodzących

w skład danej jednostki decyzyjnej. Informacje o przynależności literałów do danej reguły (dostępne jawnie na diagramie struktury reguły) są ukryte wewnątrz każdej jednostki decyzyjnej – w takim przypadku jednostka spełnia rolę tzw. czarnej skrzynki. Istnieje możliwość ujawnienia tych zależności – wtedy jednostka przybiera postać tzw. szklanej skrzynki, co prezentuje rys. 6.

W widoku szklanej skrzynki reguły należące do danej jednostki decyzyjnej mogą być w trakcie wizualizacji prezentowane w wielu różnych formach. Można prezentować *tekstową postać* reguł, rozwinąć diagram do postaci prezentującej *strukturę reguły*, poszukiwać analogii do *bramek logicznych and/or* lub prezentować reguły w postaci *tablicy decyzyjnej*. Są to przykładowe formy prezentacji, reguły jednostki decyzyjnej mogą być również przedstawione w postaci sieci Petriego.

Jednostki decyzyjne dzielą regułową bazę wiedzy na hierarchicznie zorganizowane podgrupy reguł. Reguły z takiej podgrupy – mające ten sam literał w konkluzji – są przypisywane do określonej jednostki decyzyjnej. W kontekście systemów ekspertowych konkluzje reguł najczęściej reprezentują pewną *decyzję*¹. Zatem reguły przypisane do jednostki decyzyjnej określają sposób wypracowywania pewnej *elementarnej decyzji*, reprezentowanej właśnie przez literał występujący w konkluzjach reguł jednostki. Taka jest motywacja nadania nazwy *jednostki decyzyjnej* wyodrębnionemu fragmentowi bazy wiedzy.

Analogia pomiędzy jednostką decyzyjną a układem bramek and/or (rys. 7c) odpowiada koncepcjom *reguły złożonej*, które przewidują wykorzystanie w przesłance zarówno funktora *and*, jak i *or*. Reguły złożone nie są powszechnie wykorzystywane (wykorzystuje je np. system PC-Shell [6]), zamiast nich postuluje się wykorzystywanie osobnych reguł hornowskich. Jednak w wielu zastosowaniach praktycznych stosowanie reguł złożonych jest naturalne i uzasadnione, przykładem niech będzie baza wiedzy opisywana w [15]. W rozpatrywanym przykładzie para reguł:

$$\begin{aligned} r_1 &: p \wedge q \rightarrow r \\ r_2 &: s \rightarrow r \end{aligned}$$

przypisana do jednostki decyzyjnej d_1 w istocie stanowi złożoną regułę r_{12} o postaci:

$$r_{12} : (p \wedge q) \vee s \rightarrow r$$

Jednostka decyzyjna reprezentuje zatem *regułę złożoną* nawet wtedy, gdy pierwotna reprezentacja wiedzy nie przewiduje takiej możliwości.

Dla każdej jednostki decyzyjnej można zdefiniować *zbiór warunków*, którego elementami są zbiory literałów występujących w przesłankach reguł takiej jednostki. Dla jednostki d_1 ma

¹ Charakter oraz dziedzinowy kontekst takiej *decyzji* mogą być bardzo różne. W systemach *wspomagania decyzji* pojęcie to rzeczywiście bezpośrednio koresponduje z sugestią systemu, która ma wspomóc podjęcie pewnej decyzji. W systemach *diagnostycznych* decyzja oznacza np. sugestię ustalenia przyczyny wykrytej usterki czy też wskazanie do podjęcia akcji naprawczej, a w systemach *klasyfikujących* zwykle wskazanie do zaliczenia obiektu do pewnej klasy.

on następującą postać: $\{ \{ p, q \}, \{ s \} \}$. Suma elementów takiego zbioru będzie tworzyła *zbiór punktów wejściowych* jednostki decyzyjnej; w przypadku d_1 ma on postać: $\{ p, q, s \}$ oraz jest reprezentowany graficznie owalnymi elementami z wpisanymi do nich literałami, umieszczonymi z lewej strony jednostki decyzyjnej. Analogicznie, literał występujący w konkluzji wszystkich reguł jednostki będzie *punktem wyjściowym* jednostki decyzyjnej. Dla jednostki d_1 jest nim literał r , reprezentowany owalnym symbolem graficznym, umieszczanym z prawej strony symbolu jednostki decyzyjnej. W aktualnie prezentowanych rozważaniach punkt wyjściowy będzie zawsze zbiorem jednoelementowym.

W przypadku gdy pomiędzy literałami z konkluzji oraz warunkami reguł występują powiązania, jednostki decyzyjne hierarchicznie tworzą zorganizowaną strukturę, która może przyjąć postać sieci. Struktura sieci jednostek decyzyjnych jest opisana powiązaniem pomiędzy *punktami wyjściowymi* a *punktami wejściowymi* jednostek. Dla jednostek d_1, d_2, d_3, d_4 powiązania takie występują pomiędzy punktami reprezentującymi literały r, u i w . Możliwe jest wyznaczenie takich powiązań jedynie na podstawie znajomości punktów wyjściowych i wejściowych jednostek. Punkty wejściowe i wyjściowe poszczególnych jednostek decyzyjnych przedstawiono w tabeli 1.

Tabela 1

Punkty wejściowe i wyjściowe przykładowych jednostek

Jednostka	Punkty wejściowe	Punkty wyjściowe
d_1	$\{p, q, s\}$	$\{r\}$
d_2	$\{t\}$	$\{u\}$
d_3	$\{r, u, v\}$	$\{w\}$
d_4	$\{w, x\}$	$\{y\}$

Pojedyncza jednostka decyzyjna może być rozpatrywana jako środek wypracowywania przez system decyzji elementarnej, sieć jednostek decyzyjnych może być traktowana jako globalny model decyzyjny, reprezentowany przez dany system. Opisuje sposób wypracowywania założonych celów wnioskowania systemu. Z uwagi na prostotę koncepcji jednostek decyzyjnych, łatwość graficznej prezentacji oraz intuicyjność, jednostki decyzyjne stanowią – zdaniem autora – interesującą propozycję modelu regułowej bazy wiedzy, będącego nadbudową tej reprezentacji.

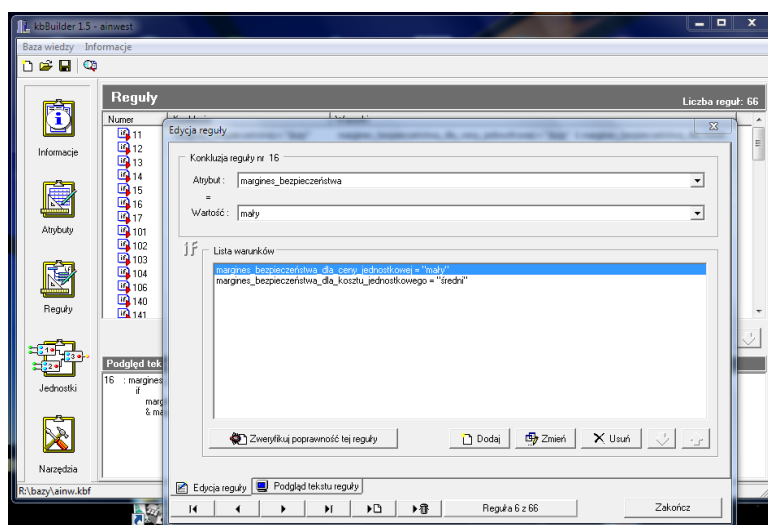
W niniejszym artykule pracy nie przedstawiono formalnego modelu jednostek decyzyjnych, model ten szczegółowo został opisany w pracach [16, 17]. Formalny opis nie jest istotny dla omawianego w artykule zagadnienia wizualizacji struktury regułowej bazy wiedzy w ujęciu praktycznym.

4. Jednostki decyzyjne we wspomaganii inżynierii wiedzy

Jednostki decyzyjne zostały wykorzystane w wielu problemach z zakresu inżynierii wiedzy. Zastosowano je jako narzędzie weryfikacji regułowych baz wiedzy [16] oraz jako środek modelowania takich baz [17]; zależności modelowane przez sieć jednostek decyzyjnych zostały również wykorzystane w optymalizacji procesów wnioskowania [8, 9]. Podstawowym celem tego artykułu jest prezentacja wybitnie użytecznego charakteru jednostek decyzyjnych – wykorzystania ich jako intuicyjnego środka wizualizacji struktury bazy regułowej.

W ramach realizowanych prac został zaimplementowany system *kbBuilder*, stanowiący poligon doświadczalny dla różnych zastosowań jednostek decyzyjnych. System ten był opisywany wcześniej [16, 17], w tym artykule autor systemu pragnie zwrócić uwagę na dwa aspekty wyróżniające opracowaną metodę reprezentacji wiedzy oraz sam system; są to:

1. zdolność do graficznego prezentowania struktury złożonych baz wiedzy o dużej liczbie reguł,
2. możliwość interaktywnego weryfikowania opracowanych reguł z wykorzystaniem przyjaznego podsystemu komunikacji z użytkownikiem.



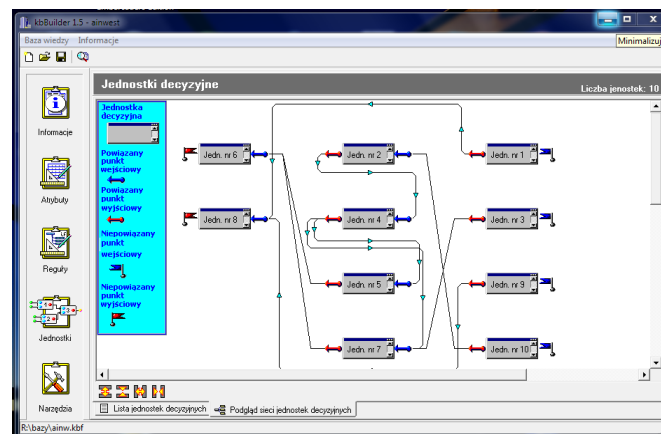
Rys. 7. Lista reguł i edycja pojedynczej reguły

Fig. 7. The rules list and rule editor

Systemy narzędziowe, wspomagające realizację baz wiedzy systemów ekspertowych, nie oferują rozbudowanych mechanizmów graficznej inspekcji czy edycji bazy wiedzy. Zawartość baz wiedzy jest prezentowana zwykle w postaci kodu źródłowego bazy reguł, zapisanego zgodnie z wybranym formalizmem, lub w postaci edytowalnej listy reguł dla systemów mających dedykowany interfejs. Graficznie zorientowana edycja baz regułowych aktualnie jest rozwiązaniem rzadko spotykanym. W aktualnej postaci system *kbBuilder* oferuje możliwość graficznej inspekcji bazy regułowej; aktualnie jest rozwijany w stronę w pełni graficznie zorientowanego edytora bazy wiedzy. Obecnie wykorzystywana wersja pozwala na edycję reguł

przez dedykowane okna dialogowe (rys. 7). System prezentuje bazę wiedzy jako sieć jednostek decyzyjnych w sposób przedstawiony na rys. 8. Należy zwrócić uwagę, że sieć jednostek, pokazana na rys. 8, przedstawia bazę liczącą blisko 700 reguł.

System pozwala na weryfikację wprowadzonych reguł w ramach jednostki decyzyjnej. Możliwa jest zarówno statyczna detekcja anomalii, której wyniki zaprezentowano na rys. 9, jak i testy przeprowadzane interakcyjnie przez użytkownika systemu (rys. 10). Prezentowany podsystem spełnia rolę debugera dla baz wiedzy – pozwala śledzić poszczególne kroki wnioskowania w ramach jednostki. Proces ten może odbywać się w trybie wcześniej opisywanej „czarnej skrzynki” – reguły biorące udział we wnioskowaniu nie są ukrywane przed użytkownikiem systemu. Rysunek 10 prezentuje pracę w trybie „szklanej skrzynki” – użytkownik widzi uaktywniane reguły, może dokonać ich inspekcji i modyfikacji. Wyniki prowadzonych prac weryfikacyjnych mogą zostać wydrukowane lub zapisane w postaci raportów tekstowych.

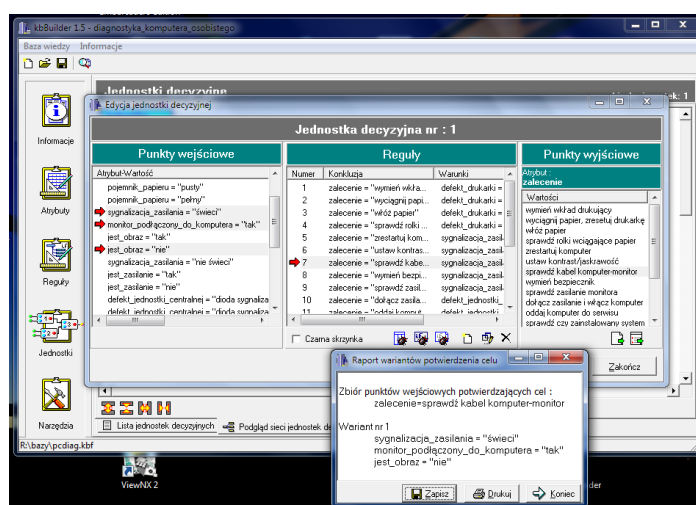


Rys. 8. Sieć jednostek decyzyjnych
Fig. 8. The decision units net

Numer	Kondycja	Warunki	Atrybut/zalecenie
1	zalecenie = "wymień wkł...	defekt_drukarki =	Wartości
2	zalecenie = "wyciągnij papk...	defekt_drukarki =	wymień wkład drukarki
3	zalecenie = "włóż papier"	defekt_drukarki =	wyciągnij papier, zresetuj drukarkę
4	zalecenie = "sprawdź rolki"	defekt_drukarki =	włóż papier
5	zalecenie = "zresetuj kom."	sygnalizacja_zasil	sprawdź rolki wciągające papier
6	zalecenie = "ustaw kontras..."	sygnalizacja_zasil	zresetuj komputer
7	zalecenie = "sprawdź kabł..."	sygnalizacja_zasil	ustaw kontras/rozstrój
8	zalecenie = "wymień bezpi..."	sygnalizacja_zasil	sprawdź kabel komputer
9	zalecenie = "sprawdź zasil..."	sygnalizacja_zasil	wymień bezpiecznik

Reguły dla których wykryto duplikaty	Typ nadmiarowości
10 : zalecenie = "dłóżcz zasilanie i włóż komputer"	Duplikaty
12 : zalecenie = "dłóżcz zasilanie i włóż komputer"	

Rys. 9. Weryfikacja jednostki decyzyjnej – wykryte duplikaty reguł
Fig. 9. Verification of decision unit – duplicated rule detection



Rys. 11. Lista reguł i edycja pojedynczej reguły
Fig. 11. The rules list and rule editor

5. Podsumowanie i wnioski końcowe

Prace związane z rozwojem jednostek decyzyjnych zostały ukierunkowane na zagadnienia o charakterze badawczym, związane z modelowaniem, weryfikacją systemów oraz optymalizacją wnioskowania. Aspekt badawczy zdominował aspekt implementacyjny; okazało się, że pierwotnie opracowana i oprogramowana forma wizualnej reprezentacji bazy wiedzy w systemie *kbBuilder* nie wytrzymuje próby czasu. Co prawda użytkownicy aktualnie testujący odświeżone wersje oprogramowania pozytywnie ocenili rozwijającą się koncepcję jednostek decyzyjnych, ich intuicyjność, użyteczność oraz nowe możliwości merytoryczne. Jednak wskazano wiele niedoskonałości implementacyjnych oraz niezgodności z aktualnie obowiązującymi standardami w zakresie oprogramowania wykorzystującego wizualnie zorientowaną edycję z wykorzystaniem grafów. Jednak należy zwrócić uwagę na fakt, iż obecnie system *kbBuilder* jest jednym z niewielu systemów tej klasy w skali światowej. Przegląd aktualnego stanu oprogramowania, wspomagającego prace inżyniera wiedzy, bądź konstruowania dziedzinowych baz wiedzy wskazuje, że wyraźnie lepszych funkcjonalnie narzędzi jest niewiele. System *kbBuilder* w aktualnej wersji rozwojowej stanowi interesującą bazę doświadczalną dla dalszych prac, które zostały zainicjowane w 2011 roku. Ich celem jest realizacja biblioteki *kbCore* oraz nowej wersji systemu *kbBuilder*.

Biblioteka *kbCore* jest obiektową biblioteką umożliwiającą programowanie własnych systemów z bazą wiedzy – reguły są zapisywane w plikach XML, biblioteka dostarcza metod ich odczytu, reprezentowania w pamięci operacyjnej, oferuje algorytmy wnioskowania. Biblioteka, zaimplementowana w standardowym C++, docelowo ma być przenośnym kodem dostępnym na zasadzie wolnego oprogramowania. Do tworzenia baz reguł w postaci XML można

wykorzystać zwykły edytor tekstowy, jednak właściwym rozwiązaniem będzie wykorzystanie nowej wersji systemu *kbBuilder*, który zachowa wszystkie pozytywnie oceniane funkcje oraz wzbogaci się o nową, w pełni wizualną wersję wizualnie zorientowanego edytora regułowych baz wiedzy. Biorąc pod uwagę obecny stan oprogramowania do realizacji systemów z bazą wiedzy, można się spodziewać, że biblioteka *kbCore* i system *kbBuilder* będą interesującą propozycją dla inżynierów wiedzy oraz grona osób zainteresowanych rozwijaniem systemów z bazą wiedzy.

BIBLIOGRAFIA

1. Chang C. L., Combs J. B., Stachowitz R. A.: A report on the Expert Systems Validation Associate (EVA). *Expert Systems with Applications*, 1990, s. 217÷230.
2. Coenen F., Bench-Capon T.: *Maintenance of Knowledge-Based Systems*. Academic Press, London 1993.
3. Grossner C., Chander P. G., Preece A.: On The Structure of Rule Based Expert Systems. Distributed Artificial Intelligence Group Technical Report – DAI-0592-0013, Computer Science Dept., Concordia University, Canada 1992.
4. Kielan K., Simiński R.: SALOMON – system ekspertowy do diagnostyki zaburzeń afektywnych. Materiały II Krajowej Konferencji TIM'97 – „Techniki informatyczne w medycynie”, Ustroń 1997, s. 167÷174.
5. Michalik K.: Sphinx 2.2 – dokumentacja pakietu. AITECH, Katowice 1998.
6. Michalik K., Simiński R.: The Hybrid Architecture Of The AI Software Package Sphinx. *Proceedings of International Conference: Colloquia in Artificial Intelligence CAI'98*, 1998, s. 210÷219.
7. Michalik K., Twardowski Z.: Financial Analysis Using a Hybrid Expert Systems. *Materiały Międzynarodowej Konferencji ECAP'94, Workshop AI in Finance and Business*, Amsterdam 1994.
8. Nowak A., Simiński R.: Wybrane zagadnienia implementacji wieloplatformowego modułu wnioskowania wstecz Infer v 2.0 dla systemu z regułową reprezentacją wiedzy – schemat bazy wiedzy i budowa struktur danych. *Materiały V Konferencji Naukowej „Inżynieria wiedzy i systemy ekspertowe”*, Wrocław 2003, s. 70÷77.
9. Nowak A., Simiński R., Wakulicz-Deja A.: Inference algorithms for hierarchical knowledge bases [in:] Kłopotek M. A., Przepiórkowski A., Wierchoń S. T., Trojanowski K. (eds.): *Recent Advances in Intelligent Information Systems*. Academic Publishing House EXIT 2009, s. 567÷578.

10. Rozenberg G., Thiagarajan P. S.: Petri nets: basic notations, structure, behavior. Current Trends in Concurrency. Lecture Notes in Computer Science, Vol. 224, Springer-Verlag, Berlin 1986, s. 585÷668.
11. Rushby J., Crow J.: Evaluation of an expert system for fault detection, isolation, and recovery in the manned maneuvering unit. NASA Contractor Report CR-187466, SRI International, Menlo Park CA 1990.
12. Simiński R.: System CAKE jako narzędzie realizacji medycznego systemu ekspertowego. Materiały konferencyjne III Konferencji TIM'98 – „Techniki informatyczne w medycynie”, Ustroń 1998, s. 39÷48.
13. Simiński R.: Petri Nets and Matrix Representation of Rule Knowledge Base for Verification Task. Advances in Soft Computing, Intelligent Information Processing and Web Mining, Springer-Verlag, 2005, s. 571÷576.
14. Simiński R., Janus S.: Wizualizacja wnioskowania w regułowych bazach wiedzy z wykorzystaniem sieci Petriego. Studia Informatica, Vol. 32, No. 2A(96), Wydawnictwo Politechniki Śląskiej, Gliwice 2011, s. 459÷472.
15. Simiński R., Kurowska J.: Wykorzystanie systemów ekspertowych we wspomaganii decyzji marketingowych. Zeszyty Naukowe Wyższej Szkoły Humanitas, s. Zarządzanie, z. 1, Oficyna Wydawnicza Humanitas, Sosnowiec 2008, s. 113÷124.
16. Simiński R., Wakulicz-Deja A.: Decision units as a tool for rule base modeling and verification. Advances in Soft Computing, Information Processing and Web Mining, Springer Verlag, 2003, s. 553÷556.
17. Simiński R., Wakulicz-Deja A.: Application of Decision Units in Knowledge Engineering. Lecture Notes in Artificial Intelligence, Vol. 3066, Rough Sets and Current Trends in Computing, Springer Verlag, 2004, s. 721÷726.
18. Wakulicz-Deja A., Paszek P.: Optimization on Decision Problems on Medical Knowledge Bases. Materiały Międzynarodowej Konferencji Intelligent Informations Systems VI, 1997, s. 204÷210.

Wpłynęło do Redakcji 11 stycznia 2012 r.

Abstract

The paper presents the conception of visualization the structure of large rule knowledge bases using decision units. The basic methods of utilization the graphs in rule base visualization task have been analyzed and the idea and an example of using decision units have been

presented. The methods of using decision units in *kbBuilder* system have been also discussed in this paper together with description of the future implementation task for the new software version – short discourse has been included in the summary to this paper on foreseen directions of development of new features of *kbBuilder* system.

Adres

Roman SIMIŃSKI: Uniwersytet Śląski, Instytut Informatyki, ul. Będzińska 39,
41-200 Sosnowiec, Polska, roman.siminski@us.edu.pl.