

Roman SIMIŃSKI
Uniwersytet Śląski, Instytut Informatyki
Mateusz MANAJ
SoftGraf

WYBRANE ZAGADNIENIA REALIZACJI PODSYSTEMU EKSPERTOWEGO W APLIKACJI INTERNETOWEJ

Streszczenie. Niniejszy artykuł prezentuje wybrane praktyczne zagadnienia związane z realizacją podsystemu ekspertowego wbudowanego w aplikację internetową. Opisana została koncepcja takiego podsystemu, jego cele oraz metody wykorzystania w aplikacji. Przedstawione zostały funkcje edytora bazy wiedzy, przyczyny wykorzystania rozszerzonego opisu atrybutów i reguł. Następnie przedstawione zostały przykłady wykorzystania tych rozszerzeń podczas przygotowania do wnioskowania oraz jego realizacji.

Słowa kluczowe: regułowa baza wiedzy, wnioskowanie, aplikacja internetowa

IMPLEMENTATION OF EXPERT SUBSYSTEM IN THE WEB APPLICATION – SELECTED PRACTICAL ISSUES

Summary. This article describes the study of selected practical issues focused on the constructing expert system module build-in the web application. We describe a conception of knowledge based subsystem, its goal and method of utilization in whole system. We present functionality of knowledge base editor, we explain the reasons of the attributes' and rules' extended description usage. Next we present an examples of application proposed extensions during the inference preprocessing and processing stage.

Keywords: rule knowledge base, inference, web application

1. Wprowadzenie

Systemy ekspertowe są jedną z najbardziej utylitarnych form wykorzystania metod rozwiązywania problemów, zaliczanych do dziedziny sztucznej inteligencji. Systemy te wykorzystują bazy wiedzy oraz algorytmy wnioskowania do rozwiązywania zwykle wąsko określonych problemów, dla których trudno wskazać rozwiązanie o charakterze algorytmicznym. Właściwie zbudowany system ekspertowy stanowić może wartościowe narzędzie w zakresie wspomagania decyzji, w wielu zastosowaniach.

Realizacja systemu ekspertowego wymaga zarówno zaangażowania specjalistów z branży IT, jak i ekspertów dziedzinowych, a także osób mogących podjąć rolę inżynierów wiedzy. W sensie projektowo-logistycznym jest to przedsięwzięcie złożone i interdyscyplinarne, wymagające rozwiązania wielu nietrywialnych problemów.

Celem niniejszego artykułu jest przedstawienie raportu z realizacji projektu, w ramach którego powstał system ekspertowy dedykowany do wspomagania procesów decyzyjnych w branży budowlanej. System ten jest w istocie jednym z podsystemów rozbudowanego internetowego systemu wspomagającego planowanie, projektowanie oraz prowadzenie prac w zakresie budowy domów realizowanych przez nieinstytucjonalnych, prywatnych inwestorów. W dalszej części tego opracowania będzie on określany mianem podsystemu ekspertowego. Prace rozwojowe nad całym systemem ciągle trwają, również realizacja podsystemu ekspertowego ma charakter rozwojowy.

W trakcie ustalania koncepcji, opracowania projektu oraz realizacji wersji pilotażowej autorzy podsystemu ekspertowego zetknęli się z wieloma problemami, które wychodzą poza zakres typowych publikacji naukowych z zakresu systemów ekspertowych oraz inżynierii wiedzy. Zdaniem autorów prezentacja tych zagadnień praktycznych może być interesująca dla wszystkich tych, którzy planują realizację i wdrożenie systemu ekspertowego z wykorzystaniem technologii dedykowanych dla aplikacji internetowych. W istocie zatem, niniejsza publikacja nie ma charakteru ściśle naukowego, a raczej prezentuje nietrywialne zagadnienia związane z praktycznym wdrożeniem koncepcji oraz metod bazujących na wcześniejszych pracach badawczych [1, 2].

2. Koncepcja systemu

Proponowane rozwiązanie jest elementem realizowanego systemu wspomagającego planowanie, projektowanie oraz realizację prac budowlanych. Grupa docelowa to inwestorzy indywidualni, zakłada się, że niekoniecznie będą oni dysponować wiedzą konieczną do zai-

nicjonowana i prowadzenia inwestycji polegającej na budowie bądź modernizacji domu. System z założenia miał być aplikacją internetową, o warstwie klienckiej działającej w środowisku przeglądarki WWW. System docelowo ma wspierać inwestorów przez uporządkowanie planowania przedsięwzięcia, oferowanie porad i podpowiedzi.

Istnieje wiele rozwiązań koncepcyjnie zbliżonych. Istotą tego opracowania nie jest analiza biznesowa oraz ocena celowości samego systemu, kluczowym jego elementem jest koncepcja oferowanego wsparcia oraz opracowania porad. Zakłada ona, że wszędzie tam, gdzie istnieją ściśle opracowane, poprawne merytorycznie oraz praktycznie sprawdzone wzorce, system będzie je implementował w sposób algorytmiczny, włączając odpowiednie procedury do warstwy biznesowej systemu. Jednak w branży budowlanej wiele problemów nie posiada jednoznacznie opracowanych metod postępowania, istnieje szereg wariantów rozwiązań, a także bardzo często rozwiązania są fragmentaryczne i niejednoznaczne. Koncepcja systemu zakładała, że dla takich problemów zastosowane zostanie podejście deklaratywne, wykorzystujące bazę wiedzy stworzoną przez ekspertów budowlanych oraz mechanizmy wnioskowania. Innymi słowy, elementem systemu ma być dedykowany, autorski podsystem ekspertowy, podejmujący próbę rozwiązania problemów, dla których fachowcy dziedzinowi nie potrafili wskazać algorytmizowalnego rozwiązania.

O realizacji wersji autorskiej zdecydowały dwa czynniki. Analiza istniejących rozwiązań [3, 7, 8] wykazała, że ze względu na specyfikę całego systemu, trudno będzie istniejące rozwiązania harmonijnie włączyć do systemu oraz głęboko je zintegrować. Drugim czynnikiem była chęć uniezależnienia się od rozwiązań zewnętrznych. Realizacja systemu jest przedsięwzięciem rozpisany na dłuższy okres czasu, wykorzystanie oprogramowania firm trzecich niesie za sobą spore ryzyko zaistnienia rozbieżności zarówno w sensie funkcjonalnym, technologicznym, jak i rozwojowym.

Podsystem ekspertowy ma być docelowo wykorzystywany w systemie jako inteligentny doradca, podejmujący próbę rozwiązywania problemów niepodatnych na algorytmizację. W aktualnej wersji podsystemu ekspertowego, w sensie ogólnym przyjęto typowe rozwiązania dla typowego systemu ekspertowego [9, 10]: regułową bazę wiedzy, klasyczne wnioskowanie w przód i wstecz, pozostawiając implementację zoptymalizowanych algorytmów na późniejszy czas [4, 5, 6].

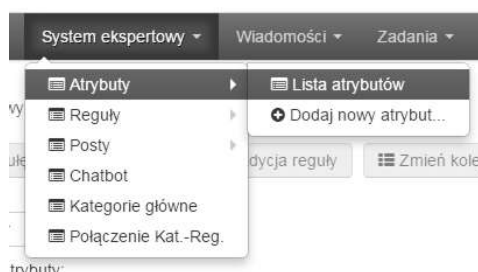
Jednak te typowe elementy systemu z bazą wiedzy zostały zrealizowane w niekoniecznie typowy sposób oraz zostały rozszerzone o istotne, zdaniem autorów, elementy praktyczne. Motywacją do ich wprowadzenia były specyficzne wymagania dziedzinowe dla podsystemu ekspertowego. Jednym z nich było wymaganie wykorzystania bazy regułowej i wnioskowania w nietypowy sposób – za pośrednictwem interfejsu zbliżonego do tzw. *chatbota*. Interfejs ten pozwala na wprowadzanie zapytań w języku naturalnym, inicjujących proces wnisko-

wania. Między innymi ta właśnie funkcja systemu wymagała opracowania specyficznych rozwiązań, które zostały przedstawione w dalszej części pracy.

3. Edycja bazy wiedzy

Baza wiedzy ma postać bazy regułowej – zakładamy, że regułowa baza wiedzy R jest skończonym, niepustym zbiorem m reguł: $\{r_1, r_2, \dots, r_m\}$, każda reguła $r \in R$ zapisana jest w postaci klauzuli Horna: $r: l_1 \wedge l_2 \wedge \dots \wedge l_n \rightarrow m$, gdzie $l_i, i: 1..n$ oznacza literal występujący w części warunkowej reguły r , a m to konkluzja reguły r . Każdy literal występujący w regule przyjmuje postać pary atrybut-wartość, wszystkie atrybuty tworzą zbiór atrybutów A , każdy atrybut $a \in A$ oraz dla każdego atrybutu określona jest jego dziedzina wartości V_a . W aktualnej wersji systemu zakłada się istnienie atrybutów symbolicznych i numerycznych ciągłych.

Tak określona postać bazy wiedzy jest typowa i nie odbiega od spotykanych w literaturze sposobów definiowania regułowych baz wiedzy. W rozważanej pracy istotnym zagadnieniem jest sposób fizycznej reprezentacji bazy regułowej, odpowiadający jednym z podstawowych wymagań – baza wiedzy ma być dostępna online, poprzez wygodny system wprowadzania i edycji atrybutów i reguł, pozwalający na pracę wielu inżynierów wiedzy lub ekspertów dziedzinowych. Zapewniony powinien zostać mechanizm pracy na jednej bazie, jednak z zapewnieniem identyfikacji autorów atrybutów oraz reguł.



Rys. 1. Podstawowe funkcje modułu edycji bazy wiedzy
Fig. 1. The knowledge base editor – primary functions

Tak postawione wymaganie w istocie wykluczyło typowe metody przechowywania baz regułowych – pliki tekstowe, w tym pliki XML, pliki arkuszy kalkulacyjnych. Dokonano dekompozycji regułowej bazy wiedzy na strukturę relacyjną, którą umieszczono w bazie danych rezydującej w warstwie serwerowej aplikacji. Funkcje serwera bazy danych dla całego systemu realizuje serwer MySQL. Dla zadania edycji regułowej bazy wiedzy utworzono system dedykowany dla inżynierów wiedzy i ekspertów dziedzinowych. Pozwala ona na tworzenie, edycję oraz przegląd list atrybutów, list reguł oraz, omówionych dalej postów, kategorii reguł oraz testowego chatbota. Wszystkie operacje realizowane w podsystemie edycji bazy

wiedzy odbywają się tylko dla zarejestrowanych użytkowników z odpowiednimi uprawnieniami, a każda istotna modyfikacja bazy jest rejestrowana.

3.1. Edycja atrybutów

Zapisanie struktury regułowej bazy wiedzy w postaci relacyjnej umożliwiło przechowywanie wielu dodatkowych informacji, zwykle niezapisywanych w typowych reprezentacjach regułowych. Z punktu widzenia inżynierii wiedzy można jednoznacznie zidentyfikować (w sensie zarejestrowanego użytkownika) twórcę atrybutu czy też reguły. Istotnym nowym elementem praktycznym jest m.in. wprowadzenie synonimów nazw atrybutów. Dla każdego atrybutu zaproponowano możliwość zdefiniowania obowiązkowej nazwy skrótowej, preferowanej nazwy opisowej oraz synonimów nazwy opisowej. Nazwa krótka może być skrótowa i potencjalnie zrozumiała tylko dla inżynierów wiedzy czy też ekspertów. Nazwa ta może mieć docelowo charakter wewnętrzny i – o ile to możliwe – nie musi być prezentowana użytkownikowi biznesowemu. Nazwa długa ma charakter opisowy i to właśnie ona ma być docelowo prezentowana w warstwie dialogowej systemu.

W aktualnej wersji systemu wykorzystywana jest jeszcze nazwa krótka, jednak doświadczenia praktyczne ekspertów tworzących reguły wykazało, że dla nich często bardzo skrócone nazwy są wygodne i zrozumiałe, lecz używane przez nich skróty są niezrozumiałe dla użytkowników biznesowych. Synonimy dla atrybutów wprowadzono przewidując niejednoznaczność pytań wprowadzanych przez użytkowników w interfejsie chatbota. To samo pojęcie może być określane w różny sposób – przykładowo terminy *bojler*, *ogrzewacz wody*, *podgrzewacz wody* mogą być różnymi określeniami tego samego obiektu. Zastosowanie nazwy krótkiej, opisowej oraz synonimów pozwala na późniejszą, elastyczniejszą identyfikację atrybutów przy analizie zapytania kierowanego do chatbota. Przykład okna edycji atrybutu z synonimami przedstawia rys. 2.

Dla każdego atrybutu symbolicznego definiuje się zbiór jego wartości, z każdym atrybutem można powiązać dodatkowe właściwości, m.in.:

- treść zapytania, która jest wykorzystywana w trakcie wnioskowania, w momencie ustalenia wartości atrybutu,
- krótki opis dla użytkownika biznesowego,
- wewnętrzny opis widoczny tylko dla inżynierów wiedzy i ekspertów,
- treść artykułu (dokument html z bazy artykułów systemu) tematycznie powiązanego z atrybutem,
- przydział atrybutu do kategorii atrybutów.

Rys. 2. Podstawowe właściwości atrybutu
Fig. 2. Basic attribute's properties

Wszystkie te dodatkowe właściwości atrybutu mają charakter praktyczny i są związane z chęcią zapewnienia możliwości uzyskiwania dodatkowych objaśnień dla atrybutów oznaczających pojęcia nieznanne użytkownikom systemu. Nazwa opisowa to pierwszy element, krótko objaśniający znaczenie atrybutu, drugi element to artykuł, który może być rozbudowanym dokumentem html. Przykład okna edycji dodatkowych właściwości atrybutu przedstawia rys. 3.

Rys. 3. Dodatkowe właściwości atrybutu
Fig. 3. Additional attribute's properties

Praca nad bazą wiedzy systemu realizowana jest przez wielu inżynierów/ekspertów współdzielących opisy atrybutów. Zgodnie z przyjętą metodyką pracy opracowali oni rozbudowany słownik atrybutów. Aby ułatwić wyszukiwanie oraz poprawić możliwości kontroli spójności na poziomie pojęciowym, atrybuty przypisywane są do kategorii, wprowadzono również mechanizmy szybkiego filtrowania według dowolnej właściwości, co ilustruje rys. 4.

ID	Uwagi	Nazwa	Typ	Synonimy
	All	prz	All	
3025	OK	Przewód kominowy	Symboliczny	
3026	OK	Przewód kominowy spełnia normy	Symboliczny	<ul style="list-style-type: none"> • komin kom • komin kom

Rys. 4. Edycja, wyszukiwanie, filtracja atrybutów

Fig. 4. Attributes – edit, search, filtering

3.2. Edycja reguł

Edycja reguł jest kluczowym elementem podsystemu zarządzania bazą wiedzy. Poszczególni autorzy mają podgląd wszystkich reguł, jednak edytować mogą tylko swoje. Zapewniono mechanizmy filtracji reguł i wyszukiwania ze względu na wykorzystywane atrybuty. Możliwy jest podgląd skrótowy – sama konkluzja, oraz szczegółowy – pełna struktura reguły. Możliwości wyszukiwania, filtrowania i przeglądu reguł przedstawia rys. 5.

Atrybut	Operator	Wartość	Konkluzja ?
Zbudowanie kominka możliwe	==	tak	✓
Przewód kominowy	==	Istnieje	✗
Dopływ świeżego powietrza	==	zapewniony	✗
Przewód kominowy spełnia normy	==	tak	✗

Rys. 5. Lista reguł – podstawowe operacje, szczegółowy opis reguły

Fig. 5. Rules list – basic action, detailed rules description

Edycja pojedynczej reguły przedstawiona jest na rys. 6. Korzystając z listy atrybutów, możliwe jest szybkie i wygodne definiowanie pojedynczej konkluzji oraz przesłanki o dowolnej liczbie warunków. Dla atrybutów numerycznych ciągłych możliwe jest definiowanie

deskryptorów zawierających operatory relacyjne. Z racji ograniczonego rozmiaru niniejszego opracowania pominięto prezentację możliwości tworzenia opisu dla reguły, opisu dla inżyniera wiedzy oraz przypisywania do reguły tematycznie związanego artykułu oraz przydziału reguły do kategorii tematycznych. W dolnej części rys. 6 występuje jednak nowy, ciekawy, lecz niepozorny element. Jest to pole tekstowe, w które można określać, coś, co zostało umownie nazwane „rezultatem konkluzji”. Jest tekst, w którym inżynier wiedzy lub ekspert określić może słowny, krótki, lecz przyjazny dla użytkownika opis znaczenia reguły. Tekst ten można interpretować jako rozbudowane słowo kluczowe, określające znaczenie, funkcję lub przeznaczenie reguły, bądź pewnej grupy reguł. Informacja ta wraz z informacjami związanymi z atrybutami (nazwa opisowa, synonimy) wykorzystywana jest przy wnioskowaniu realizowanym w ramach chatbota.

Rys. 6. Edytor pojedynczej reguły
Fig. 6. A single rule editor

4. Podsystem ekspertowy w działaniu

W realizowanym projekcie przewidziano wykorzystanie podsystemu ekspertowego w dwóch trybach. W pierwszym trybie podsystem ekspertowy jest uruchamiany niejawnie przez inne moduły aplikacji, dzieje się to w ściśle określonych momentach, uzgodnionych z ekspertami dziedzinowymi. W zależności od kontekstu oraz specyfiki aktualnie realizowanej operacji, wykonywane jest klasyczne wnioskowanie wstecz lub w przód. Wykorzystywane algorytmy nie odbiegają od wersji literaturowych. Różnica tkwi w selekcji reguł, realizacji z wykorzystaniem języka SQL. Wnioskowanie jest ukryte przed użytkownikiem, jego rezultat albo wpływa na przebieg części algorytmicznej, albo skutkuje informacjami dla użytkownika (porady, wskazówki) wykorzystującymi elementy opisowe, omówione przy okazji definiowania atrybutów i reguł. Ten scenariusz wstępnie zaimplementowano w module planowa-

nia inwestycji, ma on zostać również wdrożony w aktualnie realizowanym module projektowym. Pierwsze doświadczenia i praktyczne eksperymenty potwierdzają właściwe działanie zaimplementowanych mechanizmów, nie zaobserwowano problemów wydajnościowych. Eksperymenty wskazują jednak jednoznacznie, że wiedza dziedzinowa zapisana w bazie regułowej wymaga jeszcze wielu poprawek.

W drugim trybie podsystem ekspertowy ma brać udział w rozwiązywaniu problemów użytkownika systemu, opisanych zapytaniem wpisywanym do e-inspektora, który przypomina swoim wyglądem chatbota. Należy zwrócić wyraźnie uwagę na to, że celem nie było stworzenie sprawnie konwersującego chatbota, a jedynie próba wykorzystania bazy wiedzy i wnioskowania w konwencji zbliżonej właśnie to rozwiązań tej klasy. Przykładowe zapytanie może mieć postać przedstawioną na rys. 7.

Rys. 7. Zapytanie dla chatbota

Fig. 7. The chatbot query

Pytanie kierowane do e-inspektora może być sformułowane w dowolny sposób. W pierwszym podejściu założono, że zapytanie zostanie przetworzone z wykorzystaniem przetwarzania języka naturalnego, wykorzystania mechanizmu analizy morfologicznej. Należy zwrócić uwagę, że treść zapytania może mieć charakter faktów (np. „Mam problem brak prądu”) lub hipotezy („Kiedy można zbudować kominek?”). Moduł obsługi e-inspektora posługuje się oryginalnym autorskim oprogramowaniem poszukującym reguł pasujących do przetworzonego zapytania, wykorzystując podobieństwo zapytania do sygnatur reguł bazujących na nazwach atrybutów (krótkiej, opisowej), ich synonimach, nazwach wartości. Z racji ograniczonych ram tego opracowania element ten nie może być omówiony szerzej, problematyce tej poświęcona jest osobna publikacja.

W przypadku odnalezienia reguł pasujących do zapytania następuje próba ocenyżądanego trybu wnioskowania (opisana w osobnej publikacji) – w przypadku wnioskowania wstecz następuje proces potwierdzenia warunków reguły wybranej ze zbioru reguł pasujących. Ilustruje to rys. 8, uaktywniona jest reguła przedstawiona n rys. 5 i 6. W przypadku potwierdzenia wszystkich warunków, użytkownik otrzymuje odpowiedź przedstawioną na rys. 9. W opracowaniu jest mechanizm objaśnień.



Rys. 8. Wnioskowanie wstecz
Fig. 8. The backward inference

Zrealizowany mechanizm wyszukiwania reguł pasujących do zapytania jest mocno uzależniony od zawartości bazy wiedzy, szczególnie od opisowych elementów, związanych z atrybutami i regułami. Aktualnie tylko część bazy jest właściwie przygotowana, i dla tej bazy zaproponowany mechanizm działa zadowalająco. Dla pozostałej części bazy wykryto dużą podatność na znajdowanie reguł bardzo odległych od semantyki zapytania. Postanowiono opracować dodatkowy mechanizm selekcji reguł bazujący na „rezultatach konkluzji” (podrozdział 3.2) oraz kategoriach reguł.



Rys. 9. Potwierdzenie celu wnioskowania
Fig. 9. The inference goal confirmation

Mechanizm wykorzystujący teksty opisujące znaczenie konkluzji reguły użyty został w znanym już autouzupełnianiu w trakcie pisania zapytania. Ilustruje to rys. 10. Użytkownik może wybrać temat porady wśród wyświetlanych dynamicznie podpowiedzi. Mechanizm ten działa skutecznie, użytkownik wybierając zadany temat na pewno uzyska wsparcie – odpowiednia reguła lub reguły powinny istnieć. Po wybraniu przez użytkownika odpowiedniego tematu uruchamiane jest zawsze wnioskowanie wstecz, odbywające się w sposób klasyczny. Mechanizm działa sprawnie, choć oczywiście, jeżeli autor reguły nie uzupełni odpowiednich informacji, lub zdefiniuje nieadekwatny tekst, otrzymana podpowiedź może być daleka od oczekiwanej.

Zadaj pytanie

Kiedy

ZAPYTAJ →

Co to jest wieniec żelbetowy i kiedy należy go stosować?

Czy planowane jest wykonywanie urządzeń wodnych z wyłączeniem przypadków, kiedy pozwolenie wodno-prawne nie jest wymagane?

Kiedy możliwe jest zbudowanie kominka?

Kiedy można rozpocząć budowę po dokonaniu skutecznego zgłoszenia?

Kiedy stosować kominek z otwartym paleniskiem i kasetą

menty/piwnica

je o wykopach i ich rodzajach,

ch wykonania czy opłacalności

y/podtogi

Rys. 10. Autouzupełnianie regułowe
Fig. 10. The rule oriented autocomplete

Kolejna możliwość zawężenia tematyki pytań bazuje na kategoriach reguł, które budowane są przez ekspertów (rys. 1). Kategorie są zorganizowane hierarchicznie, z każdą kategorią można wiązać reguły do niej pasujące. Lista kategorii wyświetlana jest w oknie e-inspektora (rys. 7), użytkownik może wybrać interesującą kategorię, następnie bazując na rezultatach konkluzji (analogicznie jak przy autouzupełnianiu) użytkownik może uaktywnić wnioskowanie wstecz dla konkretnego zagadnienia – w bazie istnieje przynajmniej jedna reguła dla tego zagadnienia, inaczej tekst rezultatu konkluzji nie byłby wyświetlony.

5. Podsumowanie

Celem niniejszej pracy było przedstawienie raportu zawierającego wybrane praktyczne zagadnienia realizacji projektu, w ramach którego powstał ekspertowy podsistem dedykowany do wspomagania procesów decyzyjnych w aplikacji internetowej dla branży budowlanej. W trakcie realizacji projektu napotkano wiele praktycznych problemów. Jednym z nich był problem związany z wymianą informacji pomiędzy podsystemem a użytkownikiem. Atrybutowo zorientowana reprezentacja regułowa okazała się niezrozumiała dla biznesowych użytkowników systemu, wymagała ona swoistego opakowania. Opakowanie to przyjęło formę dodatkowych właściwości atrybutów i reguł, które są prezentowane w warstwie dialogowej systemu. Te dodatkowe właściwości, definiowane przez inżynierów wiedzy i ekspertów, wykorzystywane są do kontekstowego poszukiwania reguł w trakcie wnioskowania w obrębie modułu e-inspektora. Zrealizowana pilotażowa wersja wykazała, iż problemem nie była implementacja zarządzania bazą wiedzy oraz wnioskowania, a przygotowanie mechanizmów, pozwalających zrealizować oprogramowanie zorientowane na użytkownika. Wpisuje się to w jednoznacznie dominujący aktualnie trend *użyteczności aplikacji internetowych*.

Przedstawione opracowanie dalekie jest od kompletności, w trakcie realizacji podsystemu ekspertowego napotkano wiele innych problemów, stanowiących zarówno wyzwanie o cha-

rakterze inżynierskim, jak i naukowym. Ciągłe interesującym i otwartym zagadnieniem jest rozwój regułowo zorientowanego e-inspektora, którego specyfika każe poszukiwać niesza-blonowych modyfikacji istniejących algorytmów wnioskowania, wpisujących się w koncepcję wnioskowania mieszanego. Prace związane z tą tematyką będą kontynuowane i będą przedmiotem kolejnych publikacji.

BIBLIOGRAFIA

1. Simiński R.: Extraction of Rules Dependencies for Optimization of Backward Inference Algorithm. [w:] Kozielski S., Mrozek D., Kasprowski P., Małysiak-Mrozek B., Kostrzewa D. (red.): Beyond Databases, Architectures, and Structures, Communications in Computer and Information Science, Springer International Publishing, 2014, s. 191÷200.
2. Nowak-Brzezinska A., Siminski R.: Knowledge mining approach for optimization of inference processes in rule knowledge bases. LNCS, Vol. 7567, Springer Verlag, 2012, s. 534÷537.
3. Browne P.: JBoss Drools Business Rules. Packt Publishing, 2009.
4. Forgy C. L.: On the efficient implementation of production systems. Carnegie-Mellon University, 1979.
5. Forgy C.: Rete: A Fast Algorithm for the Many Patterns/Many Objects Match Problem. Artificial Intelligence, Vol. 19, 1982, s. 17÷37.
6. Miranker D. P.: TREAT: A Better Match Algorithm for AI Production Systems; Long Version. University of Texas, 1987, s. 78÷85.
7. Nalepa G., Ligeza A., Kaczor K.: Overview of Knowledge Formalization with XTT2 Rules. Rule-Based Reasoning, Programming, and Applications, LNCS, Vol. 6826, Springer Verlag, 2011, s. 329÷336.
8. Browne P.: JBoss Drools Business Rules. Packt Publishing, 2009.
9. Russell S., Norvig P.: Artificial Intelligence: A Modern Approach. 2nd Edition. Prentice Hall, 2003.
10. Luger G. F.: Artificial Intelligence. Addison Wesley, 2000.

Abstract

This article describes the study of selected practical issues focused on the constructing expert system module build-in the web application. We describe a conception of knowledge

based subsystem, its goal and method of utilization in whole system. We present functionality of knowledge base editor, we explain the reasons of the attributes' and rules' extended description usage. Next we present an examples of application proposed extensions during the inference preprocessing and processing stage. We illustrate discussed issues through the use of screen shoots from prototype of the real-world system. Finally, we describe three-way methods of processing natural language query for rule based e-assistant. We briefly describe the morphological approach for searching rules matching to the query and we introduce the application of additional rules information and the usage of hierarchically organized rules categories.

Adresy

Roman SIMIŃSKI: University of Silesia, Institute of Computr Science,
ul. Będzińska 39, 41-200 Sosnowiec, Poland, roman.siminski@us.edu.pl.

Mateusz MANAJ: Softgraf, ul. Plebiscytowa 51G, 43-190 MIKOŁÓW, Poland,
mmanaj@softgraf.pl.