

Rozdział 6

Moduł ekspercki

Mateusz Kalisch, Wojciech Moczulski, Piotr Przystałka, Anna Timofiejczuk, Piotr Tomasik, Dominik Wachla

6.1 Wstęp

Niniejszy rozdział przedstawia opis przeprowadzonych prac badawczych oraz wyniki badań jakie uzyskano podczas opracowania modułu szkieletowego systemu ekspertowego działającego w trybach online i offline, realizującego wnioskowanie diagnostyczne, nadzorującego przebieg procesów oraz wspomagającego użytkownika w podejmowaniu decyzji w sytuacjach kryzysowych. Celem pośrednim badań było opracowanie i weryfikacja sposobu wnioskowania, który umożliwi uwzględnienie niejednorodnej reprezentacji wiedzy pochodzącej z różnych źródeł oraz kontekst zdarzenia, którego dotyczy wiedza. W rozdziale przedstawiono podstawy teoretyczne wybranych metod i środków reprezentacji wiedzy, wybrane elementarne sposoby i algorytmy wnioskowania oraz omówiono zaproponowaną metodę wnioskowania złożonego. Ponadto przedstawiono projekt i implementację modułu szkieletowego systemu ekspertowego (w zakresie systemu działającego wg metod elementarnych oraz sposobu syntezy wnioskowań elementarnych). Omówiono również zagadnienia związane z projektem i implementacją słownika stwierdzeń.

6.2 Wybrane środki reprezentacji wiedzy oraz zastosowane metody wnioskowania

System ekspertowy (ang. expert system) jest programem komputerowym, który na podstawie specyficznej wiedzy z danej dziedziny (zgromadzonej w bazie wiedzy) może wyciągać wnioski podobnie do sposobu rozumowania człowieka [6.6, 6.23]. Można podać kilka ważnych cech odróżniających te systemy od tradycyjnych programów komputerowych. W systemach ekspertowych wiedza zapisywana jest w bazie wiedzy w sposób jawny umożliwiając jej interpretację i oddzielenie jej od procedur przetwarzania komputerowego. Wiedza używana

jest do rozwiązywania określonych problemów z zastosowaniem różnych metod wnioskowania. Systemy ekspertowe umożliwiają również wyjaśnianie rozwiązań problemów zaproponowanych przez system. Największą popularność w aplikacjach teoretycznych i praktycznych zdobyły systemy ekspertowe, w których przetwarzanie wiedzy bazuje głównie na metodach przetwarzania symboli. Bardzo ważną rolę w tym obszarze odgrywają szkieletowe systemy ekspertowe (ang. shell expert systems), które zawierają gotowy podsystem przetwarzania wiedzy, dzięki czemu zmniejszają pracochłonność podczas realizacji takiego systemu oraz ograniczają wymagania stawiane programistom, którzy zatrudniani są podczas prac rozwojowych i wdrożeniowych.

Należy w tym miejscu również zaznaczyć, że w krajowej literaturze z dziedziny diagnostyki technicznej funkcjonuje określenie systemów doradczych [6.9], które zostało wprowadzone w celu podkreślenia pomocniczej roli tego typu oprogramowania w podejmowaniu przez człowieka decyzji w zakresie działań technicznych. Jak zauważono w monografii [6.28] istotną rolę przypisano odbiorcy wyników działania tego systemu. Z kolei w literaturze anglosaskiej niezmiennie stosowana jest nazwa expert systems, która znajduje odbicie w źródle wiedzy systemu.

W niniejszym projekcie używane jest pojęcie systemu ekspertowego w szerszym sensie [6.28], gdzie system ten stosowany jest do rozwiązania jakiegoś problemu wymagającego rozumowania diagnostycznego lub/i dotyczącego nadzoru przebiegu procesów oraz wspomaganie użytkownika w podejmowaniu decyzji w sytuacjach kryzysowych. Dodatkowo ze względu na główne przeznaczenie systemu DISESOR polegające na wspomaganie podejmowania decyzji użytkowników systemów monitorowania procesów, urządzeń i zagrożeń zdecydowano się skoncentrować uwagę przede wszystkim na przypadku, w którym system ekspertowy będzie realizował wnioskowanie na podstawie danych pomiarowych. Obecne tendencje rozwoju szkieletowych systemów ekspertowych podążają w kierunku opracowania rozwiązania bardziej uniwersalnego, pozwalającego na wykorzystanie różnych elementarnych metod wnioskowania oraz różnych środków reprezentacji wiedzy [6.39]. Biorąc pod uwagę przytoczone argumenty zdecydowano, że szkieletowy system ekspertowy zostanie opracowany w taki sposób, aby mógł współdziałać z innymi modułami systemu DISESOR (głównie z modułem analitycznym i modułem prognostycznym) oraz, że zostaną wybrane takie środki reprezentacji i przetwarzania wiedzy [6.28], które umożliwią uwzględnienie niejednolitej reprezentacji wiedzy pochodzącej z różnych źródeł oraz kontekstu zdarzenia, którego dotyczy wiedza [6.35].

Wstępnie zdecydowano się wykorzystać trzy elementarne sposoby rozumowania, będące wynikiem badań podstawowych: wnioskowanie z zastosowaniem logiki dwuwartościowej [6.14], wnioskowanie posybilistyczne z zastosowaniem logiki rozmytej [6.40], wnioskowanie probabilistyczne z zastosowaniem rachunku prawdopodobieństwa [6.16]. Ponadto założono, że w procesie wnioskowania będzie uwzględniony kontekst [6.35] - np. warunki działania obiektu/procesu, historia eksploatacji, poprzednia klasa stanu itp. Jednym z kluczowych założeń z punktu widzenia implementacji całego systemu było założenie odnośnie za-

stosowania diagramów przepływu informacji bądź struktury w postaci drzewa umożliwiającego zapis procesu wnioskowania w formie scenariusza.

6.2.1 Wnioskowanie z uwzględnieniem kontekstu

W wielu systemach wiedza gromadzona jest w uporządkowany sposób. Jest to najczęściej stosowane przy regułowej reprezentacji wiedzy. Zarządzanie wiedzą w takich systemach ukierunkowane jest przede wszystkim na sposoby przeszukiwania baz w czasie wnioskowania i wyboru odpowiedniej reguły [6.6]. Podstawowym założeniem jest możliwość zastosowania algorytmów, które będą pozwalały na optymalny wybór lub aktualizację tylko tej części wiedzy, która jest związana z rozpatrywanym problemem. Do podejść, których celem jest tak rozumiane zarządzanie wiedzą, należą [6.35]:

- zastosowanie pakietów wiedzy (pakietów reguł), które gromadzą wiedzę dotyczącą określonych problemów,
- dodawanie klauzuli, które są sprawdzane podczas przeszukiwania bazy wiedzy (tzw. reguły kontekstowe).

Oba podejścia pozwalają na grupowanie reguł w zależności od tego, jakich warunków działania, jakich elementów lub jakich parametrów obiektu dotyczą. Zastosowanie takiego podejścia, przy odpowiednim sposobie wnioskowania, pozwala w pewnym sensie na ukierunkowanie procesu formułowania diagnozy, a więc podczas procesu wnioskowania uwzględniany jest pewien kontekst. Pojęcie kontekstu jest obecnie używane w wielu dziedzinach nauki [6.2, 6.27]. Definicję można podzielić na dwie grupy, z których pierwsza jest związana z podejściem kognitywistycznym [6.20, 6.26], a druga z podejściem inżynierskim [6.1, 6.2]. W pierwszym przypadku kontekst używany jest do modelowania oddziaływań (zależności) między obiektami rozumianymi jako zdarzenia. Te definicje kontekstu używane są między innymi w socjologii. W podejściu inżynierskim kontekst używany jest do reprezentacji procesu wnioskowania w przestrzeni o ograniczonej liczbie stanów, w których problem lub obiekt może się znaleźć. Opisane w dalszej części rozdziału koncepcje wykorzystania kontekstu w procesie łączenia klasyfikatorów opierają się na podejściu inżynierskim.

Kontekst jest rozumiany jako określona wiedza. Zastosowanie wiedzy kontekstowej w procesie wnioskowania przyrównuje się do zastosowania filtru definiującego w kolejnych etapach wnioskowania obszar wiedzy, które powinny być rozpatrywane. Wiedza kontekstowa rozumiana jest jako wiedza na temat historii zdarzeń i obecnego stanu rozpatrywanego obiektu lub procesu. Różnicą między kontekstem a wiedzą zapisaną bez informacji o kontekście, w jakim ją pozyskano jest to, że kontekst (wiedza kontekstowa) jest zawsze zależny od problemu. Charakterystyczne dla wiedzy kontekstowej jest to, że nie może być ona rozpatrywana bez kontekstu, który zawsze istnieje. W tym sensie każda wiedza jest wiedzą kontekstową. Jeżeli rozpatrywany jest jeden kontekst, to jest on najczęściej najbardziej ogólny, jaki można zdefiniować w określonych okolicznościach. Wiedza kontekstowa może dotyczyć dziedziny (np. wiedza dotycząca obiektów

i warunków), użytkowników (np. wiedza dotycząca celu działania użytkowników), środowiska (np. wiedza organizacyjna) oraz interakcji z innymi obiektami (np. historia zdarzeń).

Wnioskowanie z uwzględnieniem kontekstu można podzielić na kilka etapów [6.35]. Pierwszy z nich to identyfikacja kontekstu (wiedzy kontekstowej) [6.2]. Wiedza ta zawiera ogólne informacje na temat zdarzenia i problemu, co w określony sposób ogranicza przestrzeń rozwiązań. Kontekst jest identyfikowany (najczęściej wybierany ze zbioru zdefiniowanych kontekstów) w momencie wystąpienia określonego zdarzenia. Wiedza kontekstowa nie zawsze jest używana do znajdowania rozwiązania, ale ma na nie wpływ. Oznacza to, że wiedza ta na tym etapie nie jest jeszcze zorientowana na dany problem, a jedynie określa kontekst problemu. Kolejnym etapem jest tak zwana proceduralizacja, której wynikiem jest wiedza określana jako wiedza sproceduralizowana. Przez kontekst sproceduralizowany rozumie się wiedzę, która jest bezpośrednio związana z aktualnie rozpatrywanym zdarzeniem. Wiedza kontekstowa i wiedza sproceduralizowana zawarte są w zbiorze zawierającym całą dostępną wiedzę na dany temat. Ten zakres wiedzy nazywa się kontekstem zewnętrznym.

6.2.2 Drzewa decyzyjne, systemy rozmyte i sieci Bayesa

Wyróżnić można różne reprezentacje wiedzy deklaratywnej i proceduralnej. Najczęściej jednak stosowane są następujące środki [6.6, 6.28]: rachunek zdań i predykatów, stwierdzenia i sieci stwierdzeń, opisy słowne, wielowarstwowe schematy blokowe, reguły, ramy, sieci semantyczne, scenariusze, modele obliczeniowe, sieci neuronowe, sieci neuronowo-rozmyte, itp. W niniejszej pracy zdecydowano, że podstawowym elementem procesu wnioskowania będzie klasyfikator reprezentujący elementarny model wnioskowania (w szczególności uwzględnia się wnioskowanie klasyczne, wnioskowanie posybilistyczne, wnioskowanie probabilistyczne oraz wnioskowanie z uwzględnieniem kontekstu) zrealizowany w oparciu o adekwatny sposób reprezentacji wiedzy.

Drzewa decyzyjne

Drzewa decyzyjne są środkiem reprezentacji wiedzy w sposób deklaratywny oraz, w pewnym stopniu, w sposób proceduralny [6.28]. Ten środek reprezentacji wiedzy jest specjalnym rodzajem klasyfikatora, czyli struktury, która jest:

- bądź liściem wskazującym klasę,
- bądź węzłem decyzyjnym określającym pewien test do przeprowadzenia na wartości atrybutu (przypisanego do tego węzła), z którego wychodzi tyle krawędzi, ile jest możliwości wyników wspomnianego testu, a każda z tych krawędzi jest zakończona poddrzewem.

Stosowanie drzewa do klasyfikacji pewnego obiektu rozpoczyna się od korzenia tego drzewa i odbywa się aż do osiągnięcia liścia. O sposobie poruszania

się wzdłuż krawędzi drzewa decydują wyniki testów przeprowadzonych na wartościach atrybutów klasyfikowanego obiektu w poszczególnych węzłach decyzyjnych. Obiektowi zostaje przypisana taka klasa, która odpowiada osiągniętemu liściowi drzewa. Drzewo decyzyjne służy do reprezentacji wiedzy pewnej klasy C obiektów. Jest ona zdefiniowana poprzez:

- określenie jego struktury (która jest grafem – drzewem),
- określenie atrybutów przypisanych do węzłów nieterminalnych, czyli takich, które nie są węzłami końcowymi,
- określenie uporządkowanych list wartości, które przypisane są do poszczególnych krawędzi wychodzących z danego węzła nieterminalnego; uporządkowanie tych list jest istotne wtedy, gdy reprezentowane drzewo decyzyjne zostało poddane optymalizacji,
- określenie wartości atrybutu decyzyjnego (np. nazwy klasy), przypisanych do węzłów terminalnych (liści drzewa).

Węzłom można przyporządkować wagi, na podstawie których mogą być np. określane stopnie prawdziwości konkluzji, danej w postaci stwierdzenia „dla klasyfikowanego obiektu o cecha decyzyjna d ma wartość v ”.

Systemy rozmyte

Systemy rozmyte stosowane są wówczas, gdy wiedza jest nieprecyzyjna lub niepełna i równolegle istnieje potrzeba przetwarzania informacji o takim charakterze. Systemy rozmyte bazują na teorii zbiorów rozmytych zaproponowanej przez L. Zadeha w 1965 roku. Koncepcja ta jest ciągle rozwijana, co przejawia się szerokim spektrum publikacji związanych z tą tematyką w różnych dziedzinach nauki i techniki. Systemy, o których jest mowa, składają się z bazy wiedzy zapisanej w postaci rozmytych reguł warunkowych oraz z bloku wnioskowania wyposażonego w mechanizm wnioskowania rozmytego [6.33]. W zależności od zastosowania na wejścia systemów rozmytych podaje się wartości numeryczne lub wartości lingwistyczne. Gdy na wejście podawane są wartości numeryczne, niezbędne jest przeprowadzenie operacji rozmywania. Na wyjściu systemu rozmytego otrzymuje się wartości lingwistyczne. Jeżeli jest to konieczne, wartości numeryczne wyjścia uzyskuje się za pomocą operacji wyostrzania, dla zbioru rozmytego uzyskanego w procesie wnioskowania. Blok wnioskowania rozmytego realizowany jest na wiele sposobów z użyciem różnych operatorów stosowanych w logice rozmytej. Jego zadaniem jest stosowanie wiedzy zawartej w regułach w celu wypracowania konkluzji na podstawie przesłanek zbudowanych na wartościach danych. Opisane działania realizowane są zgodnie z metodami wnioskowania rozmytego.

Kluczowym zadaniem podczas tworzenia modeli rozmytych jest budowa bazy wiedzy systemu. Rozróżnia się trzy różne strategie pozyskania reguł *jeżeli-to*: metody bezpośrednie (źródłem reguł jest wiedza ekspercka), metody automatycznego generowania reguł na podstawie numerycznych danych oraz metody mieszane (wiedza ekspercka służy do określenia struktury oraz początkowych

wartości parametrów modelu, natomiast dane pomiarowe do strojenia modelu). Do najczęściej stosowanych systemów rozmytych możemy zaliczyć [6.24]: system Mamdaniego-Assilana, system rozmyty Takagi-Sugeno-Kanga, system rozmyty z parametrycznymi konkluzjami, system rozmyty Tsukamoto, system rozmyty Baldwina, systemy rozmyte typu-2 i inne. Podczas realizacji projektu zdecydowano się zastosować dwa podejścia:

1. System rozmyty Mamdaniego-Assilana [6.25], który opiera się na zbiorze rozmytych reguł warunkowych *jeżeli-to* w postaci kanonicznej:

$$R = \{R_i\}_{i=1}^I = \left\{ \text{jeśli} \left(\bigwedge_{n=1}^N X_n \text{ jest } A_{ni} \right) \text{ to } Y \text{ jest } B_i \right\}_{i=1}^I \quad (6.1)$$

gdzie I to liczba reguł, n to liczba zmiennych (atrybutów) wejściowych, X_i są wejściowymi zmiennymi lingwistycznymi, Y jest wyjściową zmienną lingwistyczną, A_{ni} reprezentuje wartości lingwistyczne dla i -tej reguły. W systemie tego typu zastosowano operację minimum jako t-normę modelując łącznik \wedge w przesłankach reguł oraz użyto koniunkcyjną interpretację reguł. Ponadto s-norma maksimum użyta została do agregacji wyników wnioskowania z poszczególnych reguł oraz metoda środka ciężkości do wyostrzenia wynikowego zbioru rozmytego.

2. System rozmyty Takagi-Sugeno-Kanga [6.34], którego baza wiedzy składa się z następujących reguł *jeżeli-to*:

$$R = \{R_i\}_{i=1}^I = \left\{ \text{jeśli} \left(\bigwedge_{n=1}^N x_{0n} \text{ jest } A_{ni} \right) \text{ to } y \text{ jest } f_i(\mathbf{x}_0) \right\}_{i=1}^I \quad (6.2)$$

gdzie x_{0n} oznacza singleton wejściowy, $\mathbf{x}_0 = [x_0, x_1, \dots, x_N]^T$, f_i jest funkcją w konkluzji i -tej reguły (z reguły wielomian stopnia pierwszego). Każda z reguł warunkowych prowadzi do wartości wyjścia typu numerycznego. Całkowite wyjście systemu wyznaczone jest jako średnia ważona wyjść y_i poszczególnych reguł (gdzie wagi wyznaczone są na podstawie stopni aktywacji poszczególnych reguł).

Sieci Bayesa

Siec bayesowska [6.4, 6.16] to acykliczny graf skierowany, w którym wierzchołki reprezentują zmienne (np. prędkość taśmy przenośnika, jego sposób sterowania lub stan techniczny, cechę obiektu itp.). Skierowane gałęzie grafu reprezentują natomiast informacyjną lub przyczynową zależność pomiędzy tymi zmiennymi. Zależność ta jest określona przez warunkowe prawdopodobieństwo, które jest przypisane do każdego połączenia pomiędzy węzłami w sieci. Zmienne reprezentowane przez wierzchołki przyjmują wartości dyskretne (np. TAK, NIE). Dla każdego wierzchołka W zdefiniowana jest tablica prawdopodobieństw warunkowych $\Pr(W|P_1, P_2, \dots)$ gdzie P_1, P_2, \dots są bezpośrednimi rodzicami

wierzchołka W . Dla wierzchołków reprezentujących przyczyny pierwotne (wierzchołki bez rodziców) prawdopodobieństwo warunkowe sprowadza się do prostych prawdopodobieństw.

Zazwyczaj utworzenie takiej sieci wymaga pomocy eksperta z dziedziny, której dotyczy tworzony model wnioskowania, w celu określenia właściwych zależności przyczynowych i prawdopodobieństw warunkowych. Możliwe jest również zastosowanie technik uczenia maszynowego (zarówno do doboru struktury sieci oraz jej parametrów swobodnych). Podstawowe kroki tworzenia i użycia sieci bayesowskiej (bez użycia danych trenujących) są następujące:

- zdefiniowanie zmiennych,
- zdefiniowanie połączeń pomiędzy zmiennymi,
- określenie prawdopodobieństw warunkowych i prawdopodobieństw *a priori*,
- wprowadzenie do sieci danych,
- uaktualnienie sieci,
- wyznaczenie prawdopodobieństw *a posteriori*.

6.2.3 Metody wnioskowania złożonego

Przyjęcie założenia, że podstawowym elementem procesu wnioskowania będzie klasyfikator reprezentujący elementarny model wnioskowania pozwala na wykorzystanie metod meta-uczenia do tworzenia zespołów klasyfikatorów (komitetów) odwzorowujących proces wnioskowania złożonego. Wyróżnić można wiele skutecznych metod meta-uczenia [6.21, 6.38], niemniej jednak w niniejszych badaniach skoncentrowano uwagę na podejściach takich jak metody głosowania, metody bootstrapowe Bagging i AdaBoost, metody bazujące na meta-klasyfikatorze oraz metody meta-uczenia z uwzględnieniem kontekstu.

Metody głosowania pozwalają na zastosowanie różnych klasyfikatorów. Wyróżnia się metody takie jak głosowanie jednomyślne, ważone lub większościowe. Głosowanie jednomyślne wymaga zgodnej decyzji wszystkich klasyfikatorów wchodzących w skład komitetu, co może w znacznym stopniu utrudnić klasyfikację, zwłaszcza w przypadku większej liczby dostępnych klas. Głosowanie większościowe polega na wybraniu klasy o największej liczbie głosów spośród wszystkich biorących udział w głosowaniu klasyfikatorów. W przypadku głosowania ważonego każdy z klasyfikatorów ma przypisaną wagę, która może na przykład odwzorować sprawność jego klasyfikacji.

Metoda Bagging polega na utworzeniu komitetu klasyfikatorów opartych na tym samym sposobie działania oraz różnym zestawie danych (tzw. próbek bootstrapowych) o liczebności równej liczebności danych źródłowych. Próbkę dla każdego z klasyfikatorów losowane są spośród danych źródłowych, przy czym każda z próbek ma takie samo prawdopodobieństwo wylosowania. Skutkiem takiego działania jest możliwość wielokrotnego powtarzania się tych samych próbek danych w pojedynczym zbiorze trenującym. Z reguły metoda ta poprawia wyniki klasyfikacji pod względem stabilności i dokładności oraz zmniejsza wariancję i przeciwdziała nadmiernemu dopasowaniu.

Kolejną grupą metod meta-uczenia stanowią metody Boostingu, do których zaliczana jest metoda AdaBoost. Ta metoda tworzenia komitetu klasyfikatorów pozwala na adaptacyjny wybór próbek zbioru trenującego, na podstawie błędów uzyskiwanych przez poprzednio wytrenowane klasyfikatory w komitecie. Każda z próbek w zbiorze trenującym ma przypisaną wagę, która określa jej istotność. Po nauczeniu pierwszego klasyfikatora na losowym zbiorze danych weryfikowana jest skuteczność klasyfikacji na tym samym zbiorze. Próbkę trenującą, dla której wynik działania klasyfikatora był błędny, w następnej iteracji mają wyższą wagę, co przekłada się na większe prawdopodobieństwo ich wylosowania. Dzięki temu kolejne klasyfikatory trenowane są z zastosowaniem zbioru danych, w którym występuje zwiększona liczba próbek, do tej pory źle rozpoznanych. Ponadto algorytm działający wg tej metody uwzględnia wartości wag dla poszczególnych klasyfikatorów w procesie końcowego wnioskowania (jak ma to miejsce w przypadku głosownia ważonego).

Jedną z często stosowanych metod meta-uczenia jest podejście bazujące na meta-klasyfikatorze. Podejście to polega na utworzeniu nadrzędnego klasyfikatora, którego wejściami są wyjścia innych bazowych klasyfikatorów oraz czasami atrybuty stanowiące wejście klasyfikatorów bazowych. Meta-klasyfikacja polega więc na równoległej pracy wielu często różnych klasyfikatorów (nierzadko utworzonych na podstawie innych zbiorów trenujących) oraz na przetworzeniu wyników wnioskowania bazowych klasyfikatorów za pomocą klasyfikatora nadrzędnego wyznaczającego ostateczną decyzję. Głównym celem takiego podejścia jest uzyskanie większej dokładności klasyfikacji w porównaniu z metodami bazującymi na pojedynczym klasyfikatorze.

Na potrzeby realizacji projektu opracowano trzy podstawowe schematy meta-uczenia z uwzględnieniem kontekstu. Każdy ze schematów stanowi osobną koncepcję wykorzystania kontekstu w procesie meta-uczenia. W niniejszym podrozdziale przedstawiono ideę działania schematów wnioskowania z uwzględnieniem kontekstu w zakresie diagnostyki uszkodzeń obiektów technicznych. Pierwsza koncepcja polega na traktowaniu cechy kontekstowej jako kolejnej zmiennej procesowej. W tym przypadku, podczas automatycznego budowania modelu w oparciu o dane trenujące, to od metody tworzenia klasyfikatora zależy w jaki sposób cecha kontekstowa wpłynie na wynik wnioskowania. Dane dostępne w repozytorium są wysyłane do dwóch klasyfikatorów, klasyfikatora kontekstu oraz głównego klasyfikatora, którego wynikiem jest końcowa decyzja w postaci sygnału diagnostycznego lub sygnatury uszkodzenia. Wynikiem działania klasyfikatora kontekstu jest dyskretna cecha kontekstowa, której wartość odpowiada jednemu ze znanych wariantów. Dopuszcza się możliwość użycia cechy kontekstowej znajdującej się bezpośrednio w bazie danych, w takim przypadku klasyfikator kontekstu nie jest wymagany. Zamiast pojedynczego klasyfikatora końcowego, możliwe jest zastosowanie komitetu klasyfikatorów, bazujących na jednej ze znanych metod łączenia klasyfikatorów lub fuzji ich wyjść.

Zasada przetwarzania danych wg drugiej koncepcji jest analogiczna do sposobu działania przełącznika. Strumień danych wejściowych (zarówno trenujących jak i nowych danych w trakcie właściwego procesu wnioskowania) jest prze-

kierowany pomiędzy klasyfikatory w zależności od aktualnie obowiązującego wariantu kontekstu. Takie podejście powoduje, że na etapie trenowania każdy klasyfikator otrzymuje inny zestaw danych, czego rezultatem jest zbiór różniących się pomiędzy sobą modeli bazujących na tym samym typie klasyfikatora. Nie ma konieczności sztucznego wprowadzania różnic w danych trenujących, aby wygenerować różniące się od siebie modele klasyfikatorów tego samego typu. Dopuszcza się również możliwość zastosowania zbioru klasyfikatorów różnego typu. Konieczność zastosowania wielu klasyfikatorów może wynikać z różnych charakterów danych powiązanych z różnymi wariantami kontekstu. Liczba klasyfikatorów musi odpowiadać liczbie wariantów kontekstu.

W trzeciej koncepcji wykorzystuje się wagi, które przypisane są do klasyfikatorów znajdujących się w komitecie. Komitec składa się z grupy klasyfikatorów. Do każdego klasyfikatora przypisana jest waga w postaci wartości liczbowej z zakresu od 0 do 1 określającej znaczenie klasyfikatora. Końcowa decyzja jest podejmowana na wzór ważonego głosowania większościowego, gdzie każdy z klasyfikatorów bazowych generuje etykietę klasy a etykieta o najwyższej sumie wag jest brana pod uwagę jako końcowa. Cecha kontekstowa podana na wejściu klasyfikatora ma wpływ na poziom istotności poszczególnych klasyfikatorów w komitecie, poprzez przypisanie do każdego z nich wagi. Szczególnym przypadkiem tej metody jest przełącznik, który dla każdego z wariantu cechy kontekstowej, do jednego z klasyfikatorów przypisuje wagę równą 1 a do pozostałych 0. W tym przypadku wynik tylko jednego z klasyfikatorów ma wpływ na rezultat całego komitetu. Jednak ten typ przełącznika różni się od koncepcji opisanej powyżej. W tym przypadku przełączaniu podlega jedynie końcowa decyzja klasyfikatorów natomiast każdy z klasyfikatorów otrzymuje na wejściu taki sam zestaw danych. To rozwiązanie wyklucza możliwość użycia klasyfikatorów tego samego typu o tych samych parametrach, ponieważ w większości przypadków modele zbudowane w oparciu o dane trenujące będą identyczne lub bardzo podobne. W celu zróżnicowania wyników działania klasyfikatorów wchodzących w skład komitetu, każdy z modeli powinien bazować na innym typie klasyfikatora lub bazować na tym samym typie ale korzystać z różnego zestawu parametrów lub powinno się zastosować metody modyfikujące dane wejściowe poprzez np. dobór różnych atrybutów (i/lub różny zestaw próbek zbioru trenującego) dla każdego z klasyfikatorów.

6.3 Opis głównych funkcjonalności modułu

W ramach badań opracowano koncepcję i projekt modułu szkieletowego systemu ekspertowego z zastosowaniem wybranych diagramów notacji UML. Podczas opracowania modułu szkieletowego systemu ekspertowego przyjęto kilka głównych założeń, które są zgodne z celem oraz z planowanymi rezultatami projektu. W trakcie realizacji badań dokonano szczegółowego przeglądu bibliotek oprogramowania ułatwiającego implementację podstawowych środków reprezentacji wiedzy oraz wybranych i opracowanych mechanizmów wnioskowania. W przypadku wnioskowania z zastosowaniem logiki dwuwartościowej rozpa-

trywano następujące opcje: tablice decyzyjne z wykorzystaniem silników Open Rules [6.30], Drools [6.11], OpenL Tablets [6.29]; wnioskowanie bazujące na regułach z wykorzystaniem silników JScheme [6.19], Clojure [6.15], TuProlog [6.36], Roolie [6.32], JRuleEngine [6.18]; drzewa decyzyjne z zastosowaniem silników QuickDT [6.31], BigML [6.5], RapidMiner [6.3]. Na podstawie przeprowadzonej analizy zdecydowano, że wnioskowanie z zastosowaniem logiki klasycznej będzie wykorzystywać algorytm drzew decyzyjnych zaimplementowany w oprogramowaniu RapidMiner. Reprezentacja procesu wnioskowania w tej formie jest czytelna i łatwa do interpretacji przez użytkownika. Wprowadzenie dodatkowo mechanizmu ręcznej edycji drzewa (dodawanie/usuwanie/zmiana parametrów węzłów) pozwala użytkownikowi na budowanie procesu wnioskowania w oparciu o reguły ostre (składające się z wartości ciągłych oraz nominalnych).

W przypadku silników wnioskowania posybilistycznego rozpatrywano następujące biblioteki i aplikacje: FuzzyTech [6.13], LFLC 2000 [6.22], jFuzzyLogic [6.17], FuzzyLite [6.12]. Po porównaniu wybranych narzędzi zdecydowano się na wykorzystanie biblioteki FuzzyLite, która w znacznym stopniu rozszerza domyślną specyfikację języka FCL (ang. Fuzzy Control Language). Liczba dodanych elementów jest większa niż w przypadku biblioteki jFuzzyLogic. Biblioteka nie udostępnia domyślnego interfejsu użytkownika pozwalającego na tworzenie oraz edycję systemów rozmytych, co wymaga opracowania edytora do interfejsu modułu systemu ekspertowego. Biblioteka FuzzyLite w odróżnieniu od komercyjnych narzędzi jest darmowa i udostępniona na licencji LGPL v3.0.

Przeprowadzono również przegląd bibliotek i aplikacji umożliwiających realizację silnika wnioskowania probabilistycznego. Podczas analizy brano pod uwagę następujące czynniki: dostępność źródeł, dostępność bibliotek API, dostępność aplikacji w postaci binarnej, dostępność graficznego interfejsu użytkownika, możliwość strojenia parametrów sieci i doboru struktury sieci, przykłady i inne narzędzia. W wyniku porównania zdecydowano się na biblioteki udostępnione w ramach systemu Weka [6.37]. Interfejs tego systemu został wyposażony w graficzny edytor sieci przekonań, pozwalający użytkownikowi na tworzenie sieci w oparciu na wiedzy eksperta, bez konieczności posiadania sklasyfikowanych danych trenujących. Jednocześnie aplikacja posiada zbiór algorytmów pozwalających na budowę sieci oraz wyznaczenie wartości prawdopodobieństw warunkowych w sposób automatyczny przy pomocy danych. Aplikacja Weka została napisana w języku JAVA co czyni ją w pełni kompatybilną z oprogramowaniem RapidMiner. Została również opracowana wtyczka integrująca kluczowe biblioteki algorytmów oprogramowania Weka z aplikacją RapidMiner. Dzięki tym narzędziom istnieje możliwość integracji wybranych algorytmów i części interfejsu oprogramowania Weka w programie RapidMiner i opracowanie nowych operatorów bazujących na algorytmach zawartych w aplikacji Weka.

W przypadku łączenia wyników elementarnych sposobów wnioskowania w pierwszej kolejności rozpatrywano możliwość wykorzystania gotowych narzędzi dostępnych w oprogramowaniu bazowym. Aplikacja RapidMiner została wyposażona w zbiór operatorów udostępniających funkcje meta-uczenia, między innymi za pomocą metody głosowania, metody Bagging, AdaBoost i metody

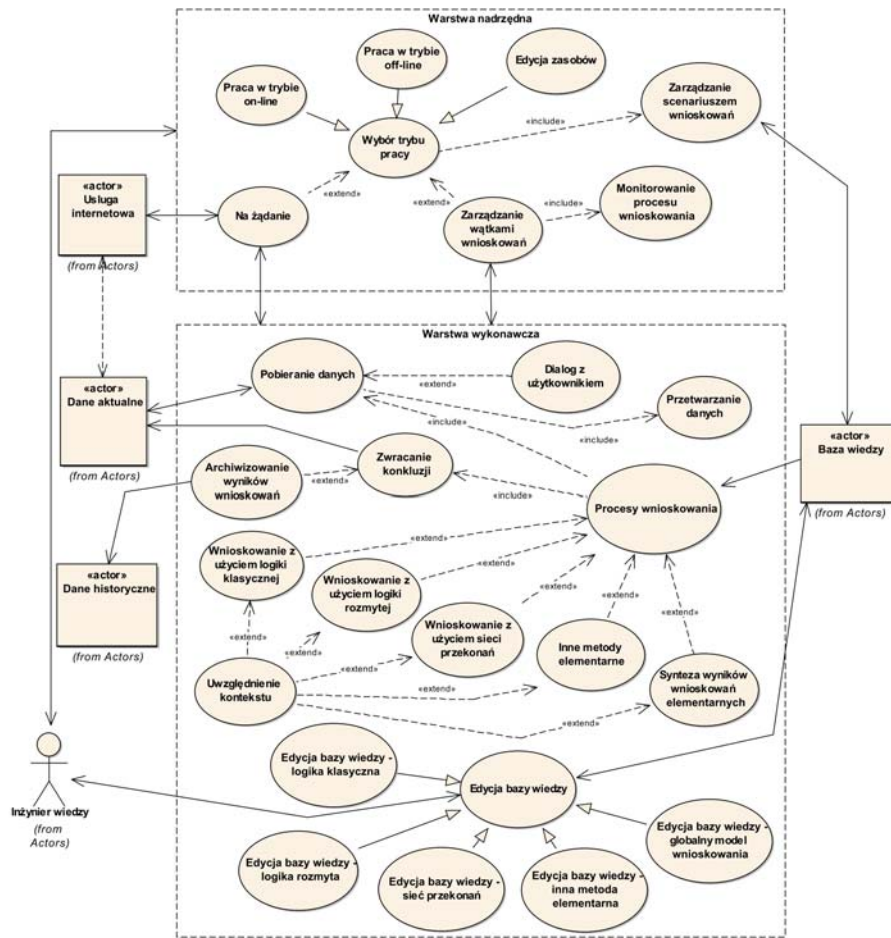
bazującej na meta-klasyfikatorze. W celu realizacji szkieletowego systemu ekspertowego zdecydowano, że dodatkowo zaimplementowane zostaną algorytmy meta-uczenia z uwzględnieniem kontekstu i zostaną one dodane do środowiska aplikacji RapidMiner jako zaawansowane operatory.

Główne funkcjonalności modułu szkieletowego systemu ekspertowego przedstawiono na rys. 6.1 w formie diagramu przypadków użycia. Jak można zauważyć, przyjęto architekturę dwuwarstwową. Warstwa nadrzędna pełni rolę i funkcję koordynacyjną w celu zarządzania pracą modułu ekspertowego. Możliwa jest praca modułu w trybie online (dane przekazywane są do systemu w czasie monitorowania obiektu), w trybie offline (do modułu przekazywane są dane historyczne zgromadzone w repozytorium) oraz w trybie edycji (zmieniane są zasoby repozytorium np. aktualizowana jest baza wiedzy).

Podczas pracy systemu w trybie online możliwe jest również użycie modułu (silnika wnioskowań) w taki sposób, że dane przekazywane są do modułu za pomocą usługi internetowej co dodatkowo pozwala na zwiększenie uniwersalności rozwiązania pod względem różnorodności źródeł danych. Ta funkcjonalność systemu reprezentowana jest przez przypadek „Na żądanie”, który bezpośrednio ma dostęp do funkcji warstwy wykonawczej. Niezależnie od trybu pracy zawsze niezbędne jest odpowiednie przygotowanie schematu/scenariusza wnioskowania („Zarządzanie scenariuszem wnioskowań”). Ponadto, wybór trybu pracy pozwala na uruchomienie wątków wnioskowań zgodnie z przygotowanym scenariuszem/schematem („Zarządzanie wątkami wnioskowań”). Podczas wykonywania wątków w warstwie wykonawczej inżynier wiedzy ma możliwość monitorowania postępów wnioskowania za pomocą funkcji warstwy nadrzędnej („Monitorowanie procesu wnioskowania”).

Działania w warstwie wykonawczej uzależnione są od wybranego trybu pracy. Użytkownik modułu ma możliwość edycji zasobów bazy wiedzy reprezentowanej w różnorodny sposób („Edycja bazy wiedzy”). Zgodnie z założeniami przyjęto, że dostępne będą trzy podstawowe mechanizmy wnioskowania tzn. wnioskowanie klasyczne z zastosowaniem logiki dwuwartościowej (w oparciu o drzewa decyzyjne), wnioskowanie posybilistyczne z zastosowaniem logiki rozmytej (w oparciu o systemy rozmyte) oraz wnioskowanie probabilistyczne z zastosowaniem rachunku prawdopodobieństwa (w oparciu o sieci Bayesa). Ponadto, dla każdego elementarnego mechanizmu wnioskowania możliwe jest uwzględnienie kontekstu. Warstwa wykonawcza umożliwia edycję (tzn. tworzenie, zapis, odczyt, modyfikację) baz wiedzy za pomocą zbioru funkcji do edycji drzewa decyzyjnego, edycji systemu rozmytego oraz edycji sieci Bayesa.

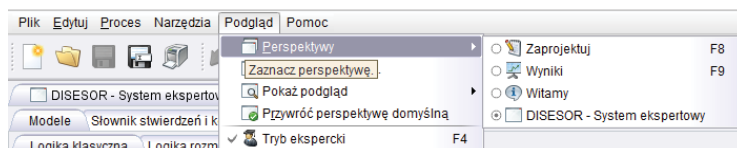
W projekcie modułu systemu ekspertowego przewiduje się również możliwość dalszej rozbudowy systemu o inne mechanizmy wnioskowania i niezbędne dla nich sposoby reprezentacji wiedzy (np. poprzez wykorzystanie gotowych mechanizmów zaimplementowanych w wybranym środowisku bazowym). Główny proces wnioskowania umożliwi również łączenie wyników wnioskowań elementarnych (z zastosowaniem odpowiednich metod). Z tego względu zbiór funkcji do edycji bazy wiedzy jest rozszerzony o funkcjonalność pozwalającą na zarządzanie globalnym modelem wnioskowania. W tym miejscu należy również zaznaczyć, że



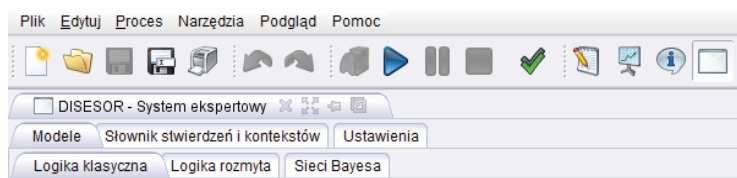
Rys. 6.1. Diagram przypadków użycia modułu szkieletowego systemu eksperckiego

w zbiorze funkcji do edycji bazy wiedzy przewidziano również możliwość zarządzania słownikami stwierdzeń. Proces wnioskowania zwraca konkluzje (elementarne i/lub będące wynikiem syntezy) w postaci stwierdzeń, które archiwizowane są w repozytorium danych. Warstwa wykonawcza odpowiedzialna jest również za pobieranie i przetwarzanie danych, na podstawie których realizowany jest proces wnioskowania. Przewiduje się również możliwość działania systemu ekspertowego w sposób klasyczny, gdzie system ten prowadzi dialog z użytkownikiem końcowym.

Moduł ekspercki systemu DISESOR został oparty o jądro aplikacji RapidMiner, która jest wyposażona w menadżer okien pozwalający na modyfikację wielkości i orientacji poszczególnych elementów interfejsu w głównym oknie aplikacji i zapisanie konfiguracji jako perspektywy. Perspektywa może też stanowić zupełnie nowy fragment interfejsu. Przełączanie pomiędzy perspektywami może odbywać się za pomocą odpowiednich pozycji w menu głównym programu lub poprzez przyciski umieszczone w pasku narzędziowym. Perspektywa modułu szkieletowego systemu ekspertowego, dodana do aplikacji RapidMiner (rys. 6.2), zawiera wszystkie najważniejsze elementy interfejsu pozwalające na obsługę modułu. Wszystkie elementy modułu wymagające interfejsu zostały pogrupowane w perspektywie za pomocą zakładek (rys. 6.3). Do głównej zakładki o nazwie „DISESOR – System ekspertowy” zostały dodane trzy zakładki: „Modele”, „Słownik stwierdzeń i kontekstów”, „Ustawienia”. W zakładce „Modele” umieszczono edytory trzech głównych typów modeli wnioskowania (drzewo decyzyjne, system rozmyty, sieć Bayesa). W zakładce „Słownik stwierdzeń i kontekstów” zamieszczono edytor służący do utworzenia i edycji słownika stwierdzeń. W ostatniej zakładce zawarto parametry działania systemu ekspertowego.



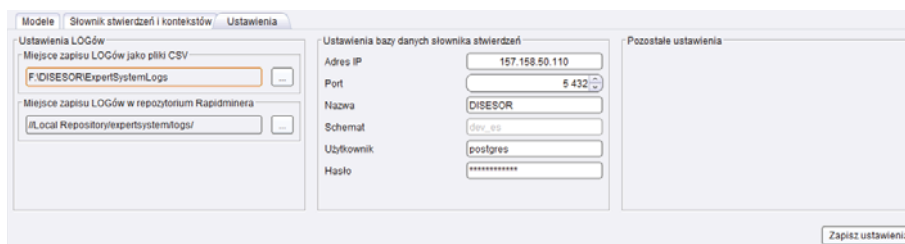
Rys. 6.2. Widok menu zmiany perspektywy w systemie DISESOR



Rys. 6.3. Widok zakładek w perspektywie modułu systemu ekspertowego

6.3.1 Zakładka ustawień modułu

W ustawieniach zawarto parametry związane z pracą szkieletowego systemu ekspertowego. Zakładka została podzielona na trzy sekcje: „Ustawienia LOGów”, „Ustawienia bazy danych słownika stwierdzeń” oraz „Pozostałe ustawienia” (rys. 6.4). Sekcja „Ustawienia LOGów” pozwala na określenie lokalizacji plików dziennika, które mogą być zapisywane w sposób automatyczny w trakcie realizacji procesu wnioskowania. Możliwe jest zapisanie plików dziennika w dowolnej lokalnej lokalizacji na dyskach twardych urządzenia, na którym realizowany jest proces. Drugą lokalizacją jest domyślne repozytorium systemu DISESOR. W sekcji „Ustawienia bazy danych” słownika stwierdzeń umieszczono parametry związane z dostępem do bazy danych słownika stwierdzeń. Wymagane jest podanie adresu IP, portu, nazwy bazy danych, nazwy użytkownika oraz hasła. Do poprawnego działania modułu, baza danych musi być umieszczona w schemacie o nazwie *dev.es*. Trzecia sekcja pozostaje pusta, pojawia się w niej inne opcje niezwiązane z dwoma poprzednimi sekcjami.

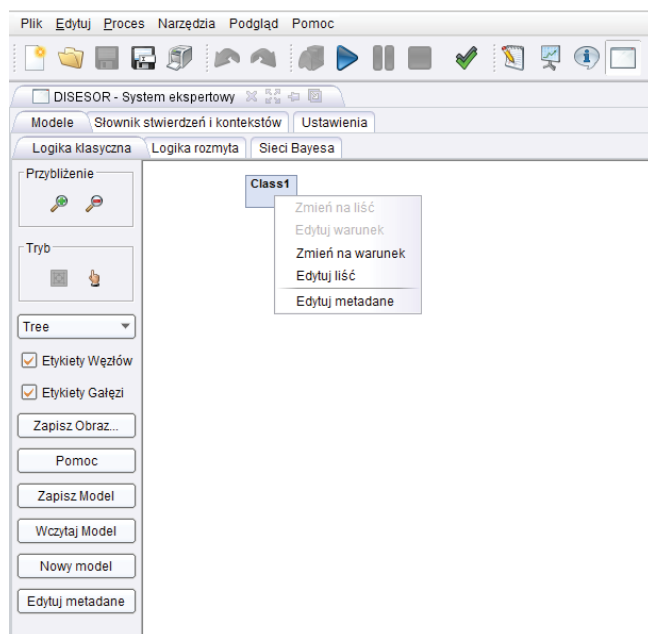


Rys. 6.4. Zakładka ustawień szkieletowego systemu ekspertowego

6.3.2 Edytor drzewa decyzyjnego

Moduł ekspercki wykorzystuje trzy elementarne sposoby wnioskowania tzn. wnioskowanie z zastosowaniem drzewa decyzyjnego, wnioskowanie z zastosowaniem systemu rozmytego i wnioskowanie z zastosowaniem sieci Bayesa. Aby użytkownik systemu (np. inżynier wiedzy) miał możliwość zbudowania każdego z tych modeli, w szczególności bazując na wiedzy eksperckiej, opracowano intuicyjne edytory każdego z wybranych typów modeli i udostępniono je w perspektywie szkieletowego systemu ekspertowego. Jak już wspomniano, do reprezentacji modelu opartego na logice klasycznej wybrano drzewo decyzyjne. To rozwiązanie niesie ze sobą wiele zalet, jak np. możliwość skorzystania z algorytm automatycznego budowania drzewa z wykorzystaniem danych oraz prostotę graficznej reprezentacji warunków i konkluzji.

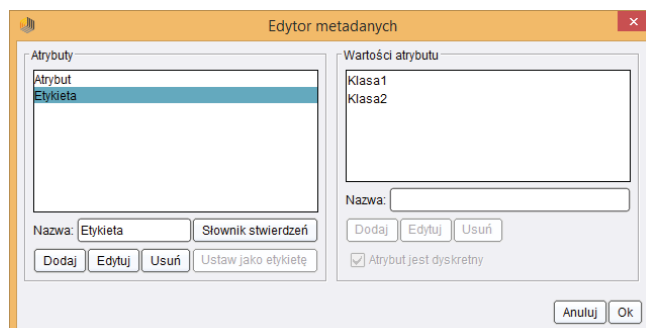
Domyślny panel wizualizacji drzewa decyzyjnego dostępny w aplikacji RapidMiner nie pozwala na modyfikowanie struktury modelu. Konieczne było zmodyfikowanie okna tak, aby użytkownik miał możliwość zmiany struktury drzewa



Rys. 6.5. Widok okna edytora drzewa decyzyjnego wraz z dostępnymi opcjami

poprzez edycję i usuwanie warunków oraz liści. W przypadku tworzenia nowego modelu od podstaw, użytkownik ma do dyspozycji pojedynczy liść z domyślną klasą. W oknie edytora istnieje możliwość edycji drzewa poprzez wciśnięcie prawego klawisza myszy na wybranym węźle. W zależności od typu węzła (warunek lub liść) wyświetli się menu kontekstowe, w którym dostępne będą jedynie opcje związane z wybranym typem węzła (rys. 6.5). Z lewej strony okna edytora znajduje się zbiór opcji pozwalających na modyfikowanie sposobu wyświetlania drzewa oraz zapis i odczyt modelu z pliku. Ważnym zbiorem danych w trakcie tworzenia drzewa są metadane, które zawierają informację o dostępnych atrybutach i ich wartościach. Zalecane jest uzupełnienie metadanych o wszystkie atrybuty oraz wartości przed rozpoczęciem budowania drzewa, ponieważ późniejsza modyfikacja metadanych może w dużym stopniu wpłynąć na strukturę opracowanego wcześniej drzewa. Okno edycji metadanych można otworzyć za pomocą przycisku znajdującego się z lewej strony okna lub ostatniej pozycji menu kontekstowego (rys. 6.5).

Okno edytora metadanych zostało podzielone na dwie sekcje (rys. 6.6). Lewa strona okna dotyczy listy atrybutów, natomiast prawa strona listy wartości atrybutów dyskretnych. Jeżeli okno edytora metadanych zostało otworzone dla nowego pustego drzewa, dostępne będą dwa atrybuty o nazwach „Atrybut” i „Etykieta” widoczne na liście z lewej strony. Element listy podświetlony kolorem niebieskim jest atrybutem decyzyjnym. Na liście musi znajdować się atrybut de-



Rys. 6.6. Okno edytora metadanych

czyzyjny i musi być to atrybut dyskretny, który nie może zostać usunięty z listy, dopóki inny atrybut nie zostanie wybrany jako atrybut decyzyjny.



Rys. 6.7. Okno dialogowe wyświetlające listę wszystkich stwierdzeń zapisanych w słowniku stwierdzeń i kontekstów

Nowy atrybut zostanie dodany tylko wtedy, gdy jego nazwa nie jest pusta, na liście nie istnieje atrybut o tej samej nazwie i jest to nazwa poprawna (tj. o właściwej składni). Błędna nazwa (o niewłaściwej składni) jest sygnalizowana czerwonym kolorem tła pola edycyjnego nazwy atrybutu. W przypadku atrybutu decyzyjnego (oraz dowolnego atrybutu dyskretnego) istnieje możliwość dodania nowego atrybutu i jego wartości automatycznie ze słownika stwierdzeń za pomocą przycisku „Słownik stwierdzeń”. Po wybraniu tej opcji wyświetla się okno z listą wszystkich stwierdzeń zapisanych w bazie danych (rys. 6.7). Po zaakceptowaniu wyboru, do listy atrybutów zostanie dodany nowy atrybut, którego nazwa jest etykietą stwierdzenia. Automatycznie zostaną dodane również wartości atrybutu, których nazwy odpowiadają etykietom wariantów wybranego stwierdzenia.

Prawa strona okna edytora metadanych (rys. 6.6) dotyczy wartości atrybutów dyskretnych. Po wybraniu elementów na liście znajdującej się z lewej strony okna atrybutu dyskretnego elementy tej sekcji okna stają się aktywne. Po wybraniu innego atrybutu z listy umieszczonej z lewej strony, na liście z prawej strony okna wyświetli się lista wartości przypisanych do aktualnie zaznaczonego

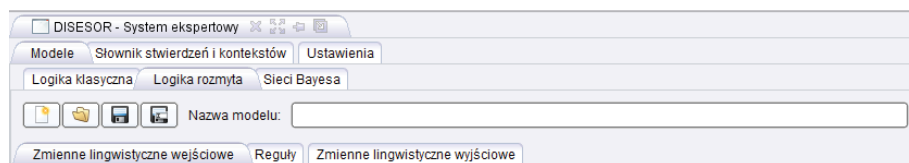
atrybutu. Podobnie jak w przypadku nazwy atrybutu, nazwa wartości również musi spełniać kryteria poprawności opisane wcześniej dla nazwy atrybutu. Pole "Atrybut jest dyskretny" widoczne w prawym dolnym rogu okna, pozwala na określenie czy wybrany atrybut jest dyskretny czy ciągły. Wybór jest możliwy tylko w przypadku atrybutów, które nie są oznaczone jako decyzyjne, atrybut decyzyjny musi być atrybutem dyskretnym.

Jedną z możliwości w trakcie edycji drzewa jest zmiana parametrów liścia. Liść drzewa odpowiada konkluzji uzyskanej w procesie wnioskowania i musi zawierać informację o końcowym wyborze wartości atrybutu decyzyjnego w procesie wnioskowania. Należy pamiętać, że modyfikowanie wartości atrybutu decyzyjnego może wpłynąć na kształt drzewa, dlatego zalecane jest przygotowanie kompletnej listy wartości przed rozpoczęciem budowania drzewa. Główną metodą wyboru wartości końcowej decyzji w pojedynczym liściu jest przypisanie do każdej z dostępnych wartości wag za pomocą odpowiedniego pola. Wartość o najwyższej przypisanej wadze będzie traktowana jako wynik wnioskowania przypisany do określonego liścia. Na podstawie przypisanych wag obliczany jest również współczynnik pewności o decyzji zwracany przez klasyfikator razem z wynikiem klasyfikacji. Inną operacją możliwą do wykonania na liściu, jest jego przekształcenie w warunek. Zmiana jest możliwa po wybraniu opcji „Zmień na warunek” w menu kontekstowym wybranego węzła. Można utworzyć nowy atrybut, który automatycznie zostanie dodany do metadanych lub wybrać atrybut zapisany już w metadanych. Dowolny warunek występujący w drzewie może być zamieniony na liść poprzez wybranie opcji „Zmień na liść” z menu kontekstowego węzła. Okno widoczne po wyborze tej opcji jest identyczne z oknem edycji parametrów liścia. Sposób jego obsługi jest również taki sam. Po zaakceptowaniu zmian warunek jest zastępowany liściem, a wszystkie jego dalsze węzły są usuwane. Ostatnią opcją pozwalającą na zmodyfikowanie drzewa jest edycja istniejącego warunku. Okno edycji można otworzyć wybierając opcję „Edytuj warunek” z menu kontekstowego węzła. Okno edycji warunku jest rozwinięciem okna zmiany liścia na warunek.

6.3.3 Edytor systemu rozmytego

Panel edytora systemu rozmytego został podzielony na trzy zakładki odpowiadające kolejno: edytorowi zmiennych wejściowych, edytorowi reguł oraz edytorowi zmiennych wyjściowych (rys. 6.8). Na pasku zadań zakładki znajdują się przyciski pozwalające użytkownikowi na zapisywanie oraz otwieranie wcześniej utworzonych modeli. Edytor modeli rozmytych obsługuje dwa typy plików, model logiki rozmytej kompatybilny z aplikacją RapidMiner oraz pliki modelu zapisane w postaci skryptu FCL. Język FCL został opracowany w celu usystematyzowania zapisu modelu logiki rozmytej. Za pomocą skryptu możliwe jest zdefiniowanie zmiennych lingwistycznych wejściowych, wyjściowych oraz reguł.

Bazowa dokumentacja języka FCL ogranicza się do podstawowych słów kluczowych, które redukują listę rodzajów funkcji przynależności do funkcji trójkątnych i składających się z listy punktów. Biblioteka użyta jako silnik obliczeń dla



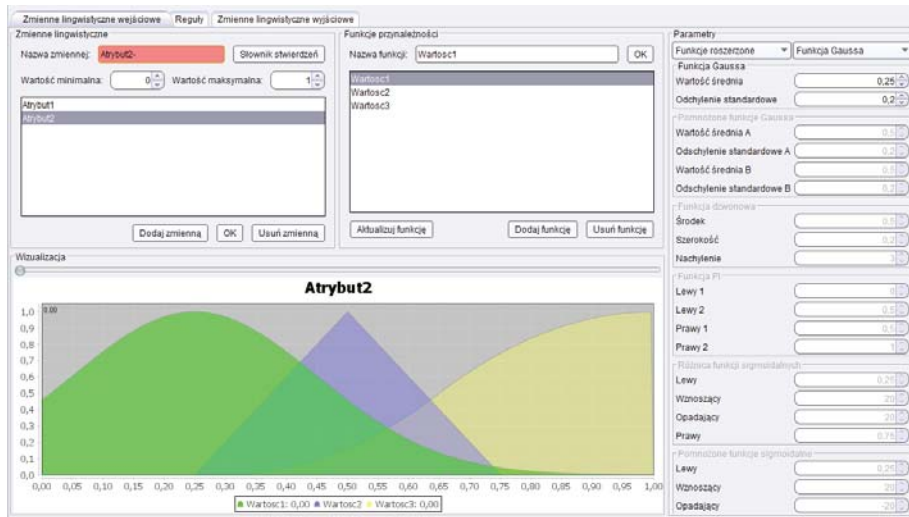
Rys. 6.8. Główne opcje zakładki edytora systemu rozmytego

modelu bazującego na logice rozmytej rozszerza funkcjonalność języka dodając inne rodzaje funkcji przynależności jak np. funkcja Gaussa oraz sigmoidalna.

W zakładce edytora zmiennych lingwistycznych wejściowych można wyróżnić cztery podstawowe obszary związane ze zmiennymi lingwistycznymi, funkcjami przynależności, parametrami funkcji oraz wizualizacji (rys. 6.9). W sekcji zmiennych lingwistycznych użytkownik ma możliwość dodawania nowych zmiennych, których nazwy odpowiadają nazwom atrybutów w tabeli danych systemu DISESOR. Podobnie jak w przypadku edytora drzewa decyzyjnego, błędna nazwa będzie oznaczona czerwonym kolorem tła pola edycyjnego. Istnieje możliwość dodania zmiennej wraz z wartościami bezpośrednio ze słownika stwierżeń. Mechanizm ten działa dokładnie tak samo jak w przypadku drzewa decyzyjnego, nowo wyświetlone okno zawiera listę wszystkich stwierżeń a wybrane przez użytkownika stwierdzenie jest dodawane do listy zmiennych lingwistycznych. Sekcja „Parametry” zawiera listę dostępnych typów funkcji wraz z ich parametrami. Funkcje zostały podzielone na cztery grupy: funkcje podstawowe, rozszerzone, krawędziowe oraz pozostałe. Zmiana grupy za pomocą listy rozwijanej umieszczonej w lewym górnym rogu sekcji modyfikuje listę funkcji i parametrów. Lista rozwijana znajdująca się w prawym górnym rogu sekcji pozwala na wybranie typu funkcji dostępnej w aktualnie wybranej grupie. Wybranie funkcji przynależności z listy w sekcji „Funkcje przynależności” automatycznie modyfikuje wygląd sekcji „Parametry” poprzez wyświetlenie odpowiedniej grupy funkcji, typu samej funkcji oraz wypełnienie pól parametrów wartościami domyślnymi przypisanymi do wybranej funkcji przynależności.

Oprócz standardowych pól parametrów w postaci nazwy parametru i wartości liczbowej, niektóre funkcje zostały wyposażone w inne elementy interfejsu. Dotyczy to funkcji dyskretnej, gdzie użytkownik określa dokładnie punkty, przez które przechodzi linia funkcji. Kolejne punkty są zapisywane w osobnych wierszach w nawiasach. Po zapisaniu wszystkich punktów, należy wcisnąć przycisk „Aktualizuj” znajdujący się pod polem edycyjnym w celu zweryfikowania poprawności wpisanych punktów. Tylko poprawnie zapisane wiersze będą brane pod uwagę, wiersze które nie będą mogły być poprawnie rozkodowane przez algorytm są ignorowane. Inną funkcją wyposażoną w niestandardowy interfejs edycji parametrów jest „Funkcja liniowa”. Parametry tej funkcji dotyczą współczynników, przez które są mnożone wartości zmiennych wejściowych.

Ostatnim fragmentem zakładki jest sekcja wizualizacji. W tej sekcji wyświetlane są funkcje przynależności przypisane do wybranej zmiennej lingwistycznej. Kolorem zielonym wyświetlana jest funkcja zaznaczona na liście funkcji przyna-



Rys. 6.9. Okno edytora zmiennych lingwistycznych wejściowych

leżności. Suwak znajdujący się w górnej części sekcji pozwala na zweryfikowanie odpowiedzi wszystkich funkcji przynależności na sztucznie zadaną wartość zmiennej lingwistycznej. Przesuwając suwak, na wykresie przesuwają się również pionowe linie wraz z największą wartością funkcji przynależności. W ramce legendy (pod wykresem) obok nazwy każdej z funkcji wyświetla się wartość funkcji przynależności dla każdej z nich.

Edytor reguł umożliwia zdefiniowanie reguł z uwzględnieniem zmiennych lingwistycznych wejściowych i wyjściowych (rys. 6.10). Budowanie reguł jest możliwe tylko w sytuacji, gdy istnieje przynajmniej jedna zmienna lingwistyczna wejściowa i wyjściowa.

Warunek	Warunek	Warunek	Konkluzja	Konkluzja
Atrybut1 JEST Wartosc1 AND	Atrybut2 JEST Wartosc1		STWempok JEST Nie	STWemptoohigh JEST Tak
Atrybut1 JEST Wartosc1 AND	Atrybut2 JEST Wartosc2		STWempok JEST Nie	STWemptoohigh JEST Tak
Atrybut1 JEST Wartosc1	Atrybut2 JEST Wartosc3		STWempok JEST Nie	STWemptoohigh JEST Tak
Atrybut1 JEST Wartosc2 AND	Atrybut2 JEST Wartosc1		STWempok JEST Tak	STWemptoohigh JEST Nie
Atrybut1 JEST Wartosc2 AND	Atrybut2 JEST Wartosc2		STWempok JEST Tak	STWemptoohigh JEST Nie
Atrybut1 JEST Wartosc2 AND	Atrybut2 JEST Wartosc3		STWempok JEST Nie	STWemptoohigh JEST Nie
Atrybut1 JEST Wartosc3 AND	Atrybut2 JEST Wartosc1		STWempok JEST Nie	
Atrybut1 JEST Wartosc3 AND	Atrybut2 JEST Wartosc2		STWempok JEST Tak	
Atrybut1 JEST Wartosc3 AND	Atrybut2 JEST Wartosc3		STWempok JEST Nie	

(Atrybut1) jest (Wartosc3) | |

Aktywny

T-Norma Minimum | Minimum | Lub Maximum

Rys. 6.10. Widok okna zakładki reguł modelu logiki rozmytej

Okno zakładki reguł podzielono na dwie części. Lewa połowa dotyczy warunków a prawa konkluzji. Warunki należy dodać poprzez zaznaczenie pustej komórki tabeli umieszczonej z lewej strony i wybranie w dolnej części sekcji "Warunki" nazwy zmiennej lingwistycznej wyjściowej za pomocą lewej listy rozwijanej oraz wybranie funkcji przynależności na liście rozwijanej z prawej strony sekcji. Po wybraniu nazw należy wcisnąć przycisk „Aktualizuj regułę”, co spowoduje dodanie nowej reguły. Po dodaniu nowej reguły w nowym wierszu, w sekcji „Konkluzje” automatycznie zostaną dodane wszystkie konkluzje. Liczba konkluzji w wierszu zależy od liczby zmiennych lingwistycznych wyjściowych. Jeżeli nie wszystkie konkluzje w określonej regule powinny być brane pod uwagę, nieistotne konkluzje należy wyłączyć odznaczając pole obok nazwy „Aktywny” w lewym dolnym rogu sekcji. W ten sposób wybrana konkluzja zostanie oznaczona czarnym kolorem i nie będzie brana pod uwagę w trakcie wnioskowania. Po dodaniu nowego wiersza, do każdej z wygenerowanych konkluzji przypisana jest pierwsza dostępna funkcja przynależności. Aby przypisać do konkluzji inną funkcję należy zaznaczyć odpowiednią komórkę tabeli oraz wybrać docelową funkcję z listy rozwijanej znajdującej się w prawym dolnym rogu sekcji „Konkluzje”.

W sekcji dotyczącej warunków możliwe jest dodawanie reguł oraz warunków. Jeżeli w pojedynczym wierszu znajduje się więcej warunków, istotne jest aby określić pomiędzy nimi operator logiczny (domyślnie dodawany jest I/AND). Możliwe jest również zdefiniowanie nawiasów w przypadku bardziej złożonych warunków poprzez zaznaczenie pól (i) umieszczonych z lewej i prawej strony list rozwijanych zmiennych lingwistycznych i funkcji przynależności w dolnej części sekcji. Ostatnia lista rozwijana umieszczona w prawym dolnym rogu sekcji pozwala na zmianę typu operatora logicznego pomiędzy warunkami. Zaznaczony warunek można usunąć za pomocą przycisku „Usuń regułę”. Reguły, w których występują błędy są zaznaczone czerwonym tłem. Błąd może wystąpić np. w przypadku niedomknięcia wszystkich nawiasów w regule. Przycisk „Generuj reguły” usuwa wszystkie aktualnie zdefiniowane reguły i wypełnia tabelę regułami z wszystkimi możliwymi kombinacjami warunków. Na samym dole zakładki znajdują się parametry związane ze sposobem analitycznej realizacji sumy i różnicy logicznej pomiędzy warunkami oraz t -normy.

Zmienna decyzyjna w modelu logiki rozmytej może być utworzona w oparciu o dwa podejścia. Pierwsze z podejść wymaga zastosowania pojedynczej zmiennej lingwistycznej wyjściowej wraz z szeregiem przypisanych funkcji przynależności. Nazwa zmiennej lingwistycznej odpowiada nazwie atrybutu decyzyjnego a nazwy funkcji przynależności nazwom wartości dyskretnej jakie może przyjmować atrybut. Wartości przynależności dla każdej funkcji odpowiadają wartością stopnia pewności decyzji. W drugim podejściu należy dodać tyle zmiennych lingwistycznych ile jest wartości dyskretnej atrybutu. Nazwa każdej zmiennej musi zgadzać się z nazwą wartości atrybutu. Nazwa atrybutu musi być określona osobno w odpowiednim polu edycyjnym. Wartość stopnia pewności o decyzji wynika z wartości funkcji defuzyfikacji każdej ze zmiennych. Nazwa zmiennej, która zwróci największą wartość jest traktowana jako etykieta końcowej decyzji.

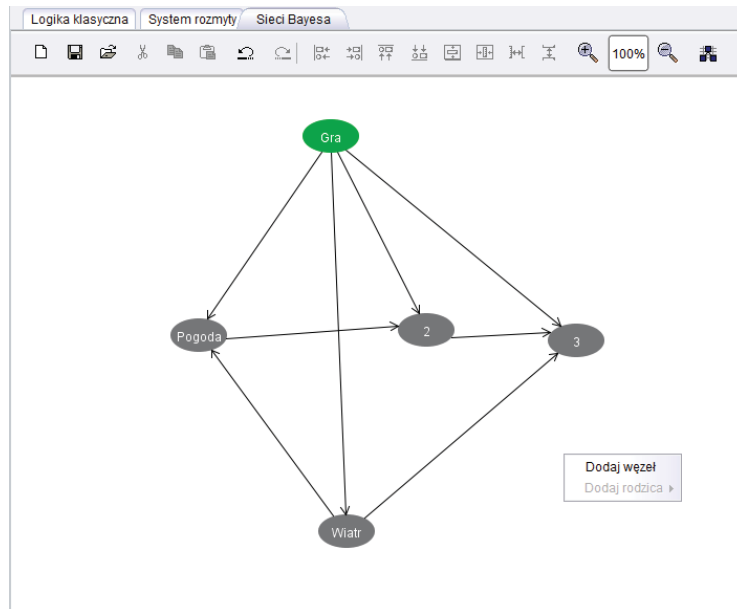
Interfejs zmiennych lingwistycznych wyjściowych (rys. 6.11) jest zbliżony do interfejsu i sposobu działania edytora zmiennych lingwistycznych wejściowych. Dostępne są dodatkowe parametry w sekcji dot. zmiennych lingwistycznych. Pole edycyjne „Nazwa atrybutu” dotyczy nazwy atrybutu decyzyjnego w zbiorze danych. W przypadku pierwszej metody realizacji wnioskowania, kiedy dostępna jest tylko jedna zmienna lingwistyczna, nazwa atrybutu jest taka sama jak nazwa zmiennej lingwistycznej. Dla drugiej metody jest dostępna większa liczba zmiennych lingwistycznych i wymagane jest określenie nazwy atrybutu osobno. W dolnej części sekcji konieczne jest określenie metody defuzyfikacji i agregacji wyniku. Wartość domyślna zmiennej lingwistycznej jest zwracana w sytuacji, gdy żadna z reguł nie jest spełniona. Możliwe jest zadanie stałej wartości lub flagi NaN.



Rys. 6.11. Okno zakładki zmiennych lingwistycznych wyjściowych

6.3.4 Edytor sieci Bayesa

Edytor sieci Bayesa jest zmodyfikowaną wersją edytora dostępnego w aplikacji Weka. Pasek narzędzi umieszczony w górnej części zakładki pozwala na zapisanie oraz wczytanie modelu z pliku (rys. 6.12). Model sieci może być zapisany do pliku w formacie zgodnym z aplikacją Weka jak i systemem DISE-SOR. Kolejne opcje dotyczą struktury oraz sposobu wyświetlania i organizacji węzłów w edytorze (kopiowanie węzłów, rozmieszczanie węzłów, przybliżanie widoku). Aby dodać nowy węzeł do edytora, należy wcisnąć prawy klawisz myszy nad pustym fragmentem okna edytora a następnie wybrać opcję „Dodaj węzeł” (rys. 6.13).



Rys. 6.12. Okno zakładki edytora sieci Bayesa

The "Dodaj węzeł" dialog box contains the following fields and controls:

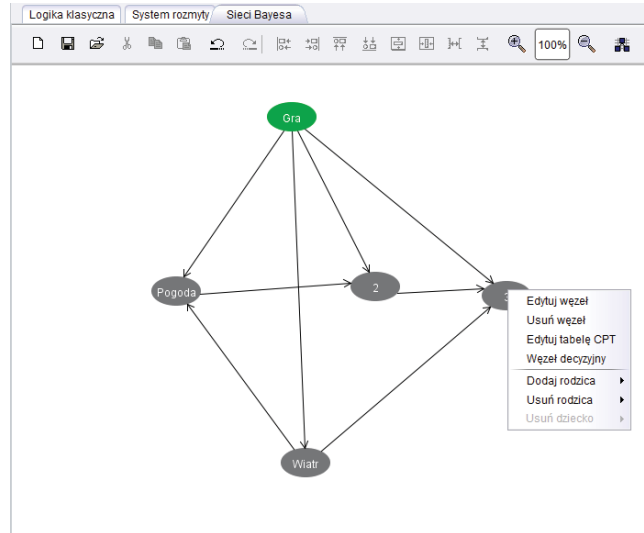
- Text field: "Nazwa atrybutu:" with value "Atrybut1".
- Text area: "Wartosc1" and "Wartosc2".
- Checkboxes: "Atrybut nominalny" and "All".
- Text field: "Nazwa wartości:" with value "Wartosc2".
- Buttons: "Dodaj", "Edytuj", "Usuń", "Anuluj", "OK", and "Edytuj wartości".
- Text: "Słownik stwierdzeń" next to the "Nazwa wartości:" field.

Rys. 6.13. Okno pozwalające na dodanie nowego węzła

Nazwa każdego węzła musi być taka sama jak nazwa wybranego atrybutu, ponieważ każdy węzeł musi być związany z atrybutem. Zbyt długie nazwy węzłów są zastępowane w widoku edytora numerem porządkowym. W celu dodania nowego węzła należy podać nazwę atrybutu, który jest jednocześnie nazwą węzła, oraz jego wartości (rys. 6.13). Sieć Bayesa zaimplementowana w oprogramowaniu Weka może współpracować jedynie ze zmiennymi dyskretnymi, dlatego w przypadku atrybutu ciągłego należy wykonać dyskretyzację zmiennej. W tym celu należy określić progi, pomiędzy którymi powstają przedziały definiujące wartości dyskretne. Wybór typu zmiennej następuje za pomocą zaznaczenia lub odznaczenia pola „Atrybut nominalny”. Zmiana zaznaczenia modyfikuje nieznacznie wygląd interfejsu dając dostęp tylko do tych opcji, które są związane z wybranym typem. Dla atrybutu dyskretnego dostępne jest pole edycyjne „Nazwa wartości” oraz przyciski „Dodaj”, „Edytuj” i „Usuń”. Podobnie jak w przypadku poprzednich edytorów, w tym przypadku również istnieje możliwość wyświetlenia okna z listą zapisanych stwierdzeń. W tej sytuacji nazwa atrybutu i lista wartości zostanie wypełniona automatycznie etykietą stwierdzenia oraz etykietami wariantów. W przypadku atrybutu ciągłego, należy określić wartości progowe algorytmu dyskretyzacji poprzez wciśnięcie przycisku „Edytuj wartości”.

Wciśnięcie prawego klawisza myszy w chwili, kiedy kursor znajduje się nad węzłem w edytorze spowoduje wyświetlenie menu kontekstowego edycji węzła. Menu zostało podzielone na dwie części. Górna część dotyczy edycji wybranego węzła, dolna natomiast pozwala na zarządzanie relacjami pomiędzy węzłami (rys. 6.14). Opcja „Edytuj węzeł” wyświetla okno edycji węzła, które wygląda i funkcjonuje tak samo jak okno dodania nowego węzła do edytora. Opcja „Usuń węzeł” usuwa wybrany węzeł z okna edytora. Element „Edytuj tabelę CPT” wyświetla okno edytora wartości prawdopodobieństwa dla wybranego węzła.

W oknie edytora prawdopodobieństw zamieszczona jest tabela prawdopodobieństw, w której kolumny dotyczą wartości przypisanych do wybranego węzła a wiersze odnoszą się do wszystkich kombinacji wartości węzłów znajdujących się wyżej w strukturze i będących bezpośrednio połączonymi z wybranym węzłem. Suma wartości w wierszach musi być równa 1. Wartość prawdopodobieństwa w ostatniej kolumnie może być obliczana automatycznie przez algorytm. W tym celu po zakończeniu edycji wartości w poprzednich kolumnach należy zaznaczyć ostatnią komórkę tabeli. Jeden z węzłów musi być związany nazwą z atrybutem decyzyjnym. Dodatkowo w tworzonym modelu klasyfikatora opartego na sieci Bayesa należy określić, który węzeł jest węzłem decyzyjnym. Do tego celu służy opcja „Węzeł decyzyjny” widoczna w menu kontekstowym węzła (rys. 6.14). Wybranie tej opcji spowoduje zmianę koloru węzła na zielony. W przypadku nieoznaczenia węzła decyzyjnego i próbie zapisu modelu do pliku, wyświetlony zostanie komunikat informujący o konieczności wybrania węzła decyzyjnego. W drugiej części menu kontekstowego zawarte są opcje pozwalające na budowanie zależności pomiędzy węzłami (rys. 6.14). Najważniejszą opcją jest „Dodaj rodzica”, która pozwala na wybranie węzła nadrzędnego z listy wszystkich dostępnych węzłów. Pozostałe dwie opcje służą do usuwania relacji pomiędzy wybranymi węzłami.

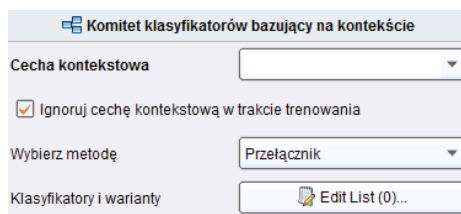


Rys. 6.14. Okno edytora sieci Bayesa z widocznym menu kontekstowym wybranego węzła

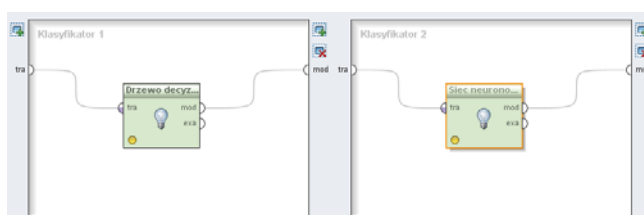
6.3.5 Model wnioskowania z uwzględnieniem kontekstu

Moduł szkieletowego systemu ekspertowego posiada nowy operator, który pozwala na tworzenie komitetów klasyfikatorów uwzględniających kontekst. Sposób działania operatora jest zgodny z konwencją przyjętą przez autorów bazowego oprogramowania. Dodatkowy operator jest tzw. super-operatorem i pozwala na umieszczanie w jego „wnętrzu” innych operatorów, w tym przypadku klasyfikatorów. W operatorze zaimplementowano algorytmy bazujące na koncepcjach klasyfikacji z wykorzystaniem cechy kontekstowej. Za pomocą parametrów dostępnych w oknie konfiguracji operatora, użytkownik może zdecydować o sposobie wykorzystania cechy kontekstowej. W samym operatorze istnieje możliwość zastosowania różnych klasyfikatorów, które zostaną wykorzystane w celu uzyskania końcowej decyzji. Wyjściem operatora jest pojedynczy model, który zawiera wszystkie modele klasyfikatorów wchodzących w skład komitetu. Powstały model bazujący na kontekście jest również kompatybilny z modelami domyślnie budowanymi w systemie DISESOR. Model taki może zostać zapisany w dowolnym miejscu repozytorium systemu i wykorzystany w późniejszym właściwym procesie wnioskowania poprzez zastosowanie standardowych funkcji oraz operatorów systemu DISESOR.

Na rys. 6.15 pokazano widok panelu konfiguracji operatora. Do jego poprawnego działania wymagana jest cecha kontekstowa reprezentowana za pomocą dyskretnego atrybutu znajdującego się w zbiorze danych trenujących. Tego typu atrybut należy wybrać z listy dostępnej w panelu konfiguracji operatora lub wpisać jego nazwę ręcznie, jeżeli nie jest widoczny na liście. Atrybut będący cechą kontekstową może wchodzić w skład zbiorów danych trenujących klasyfikatory

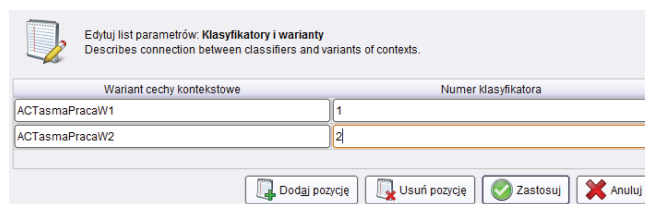


Rys. 6.15. Panel parametrów operatora komitetu klasyfikatorów bazujących na kontekście



Rys. 6.16. Widok okna procesu z dwoma typami klasyfikatorów

wewnątrz operatora lub może zostać wykluczony z tego procesu za pomocą opcji Ignoruj cechę kontekstową w trakcie trenowania. W celu zbudowania komitetu składającego się z większej liczby klasyfikatorów, należy dodać do widoku podprocesu operator "Komitet klasyfikatorów bazujący na kontekście" (rys. 6.16) nowe obszary robocze za pomocą ikon ze znakiem plusa. W każdym z obszarów należy umieścić klasyfikator a następnie w oknie dialogowym, które wyświetli się po wybraniu opcji „Klasyfikatory i warianty” przypisać do każdego z obszarów wariant cechy kontekstowej (rys. 6.17). W przypadku wariantu, który nie został przypisany do żadnego z obszarów, wariant automatycznie zostanie przypisany do pierwszego obszaru.



Rys. 6.17. Okno konfiguracji powiązań pomiędzy klasyfikatorami i wariantami cechy kontekstowej

6.4 Słownik stwierdzeń

W systemach ekspertowych do komunikacji z użytkownikiem stosowane są stwierdzenia [6.6]. Stwierdzenie jest wypowiedzią orzekającą o zaobserwowanych faktach lub reprezentującą określoną opinię daną w postaci zdania oznajmującego czyli zdania z orzeczeniem, któremu przypisywana jest wartość logiczna (0 lub 1) lub stopień przekonania (prawdziwości) $\langle 0; 1 \rangle$ o słuszności tego stwierdzenia [6.7, 6.10]. Przykładami stwierdzeń są następujące zdania [6.6, 6.7, 6.10]:

Jabłko jest zielone. (6.3)

Temperatura panevki łożyska promieniowego jest niższa od 40 °C. (6.4)

Temperatura panevki łożyska promieniowego jest wyższa lub równa 40 °C. (6.5)

Od strony formalnej, stwierdzenie s może być reprezentowane za pomocą następującej czwórki [6.7–6.9]:

$$s = \langle o, a, v, p \rangle \quad (6.6)$$

gdzie: o – to obiekt, a – atrybut (cecha) przysługująca obiektowi o , v – wartość atrybutu (cechy) oraz p – stopień prawdziwości lub stopień przekonania (w przypadku stwierdzenia definiowanego przez eksperta) o słuszności stwierdzenia.

Stwierdzenie można również zapisać w postaci [6.7]:

$$s = \langle c, p \rangle \quad (6.7)$$

gdzie c – jest treścią stwierdzenia, czyli przedstawioną w języku naturalnym wypowiedzią o tym, że obiektowi o przysługuje atrybut (cecha) a o wartości v .

Wśród stwierdzeń można wyróżnić stwierdzenia proste (jednowariantowe) oraz stwierdzenia złożone (wielowariantowe) [6.7]. Stosując zapis (6.7) stwierdzenia n -wariantowe można przedstawić w postaci [6.7]:

$$s = \langle \underline{\mathbf{c}}, \underline{\mathbf{p}} \rangle = \langle c_{1:n}, b_{1:n} \rangle, \quad (6.8)$$

gdzie treść stwierdzenia złożonego $\underline{\mathbf{c}}$ jest n -elementowym wektorem (listą) wariantów treści stwierdzenia, a $\underline{\mathbf{p}}$ jest n -elementowym wektorem wartości stopnia prawdziwości kolejnych wariantów jego treści. W podobny sposób zapis stwierdzenia wg (6.6) można rozszerzyć na stwierdzenie wielowariantowe:

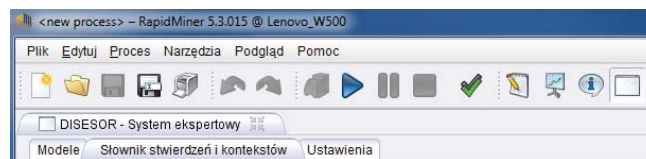
$$s = \langle o, a, \underline{\mathbf{v}}, \underline{\mathbf{p}} \rangle = \langle o, a, v_{1:n}, b_{1:n} \rangle. \quad (6.9)$$

Z analizy treści wielu stwierdzeń wynika, że treść stwierdzenia ze względu na syntaktykę (składnia, gramatyka) języka naturalnego w jakim jest wyrażona nie może być bezpośrednio wyprowadzona na podstawie trójki $\langle o, a, v \rangle$. Z drugiej strony, w treści stwierdzenia można wyróżnić fragmenty stanowiące

opis jednego z ww. elementów. Zatem, każdemu elementowi trójki $\langle o, a, v \rangle$ można przypisać adekwatny fragment treści stwierdzenia wyrażonego w języku naturalnym. Fragmenty te po złożeniu w odpowiedniej kolejności w całość utworzą treść stwierdzenia, poprawną pod kątem syntaktyki stosowanego języka naturalnego.

Przykładowo w stwierdzeniu (6.3) obiektem o jest *jabłko* gdzie fragment treści tego stwierdzenia odnoszący się do tego obiektu to wyraz „*Jabłko*”. Atrybut a opisujący obiekt o jest w tym stwierdzeniu niejawny i jest nim *kolor* jabłka. W takim przypadku, fragment treści stwierdzenia odnoszący się do atrybutu a jest zbiorem pustym. Jednocześnie wartość v atrybutu jest znana i jest to wartość lingwistyczna *zielony*. Odpowiadającą tej wartości fragment treści stwierdzenia to wyraz „*zielone*”. Ostatnim elementem kompletującym całą treść stwierdzenia jest wyrażenie określające charakter posiadania przez obiekt o określonej wartości v atrybutu a [6.9]. W rozpatrywanym przypadku jest to czasownik *być* występujący w formie „*jest*”.

Interfejs użytkownika słownika stwierdzeń został opracowany jako moduł systemu DISESOR, który ściśle współpracuje z modułem eksperckim. Główne okno modułu jest widoczne w panelu systemu ekspertowego w postaci perspektywy o nazwie „Słownik stwierdzeń i kontekstów” (rys. 6.18). Ze względu na wymaganą funkcjonalność słownika został on podzielony na sześć zakładek: SŁOWNIK STWIERDZEŃ; Dodaj/Usuń OBIEKT; Dodaj/Usuń/Edytuj ATRYBUT; Dodaj/Usuń WARTOŚĆ ATRYBUTU; Dodaj/Usuń STWIERDZENIE; Dodaj/Usuń WARIANT STWIERDZENIA. Dostęp do wszystkich zakładek posiadają tylko użytkownicy mający nadane uprawnienia do korzystania ze szkieletowego systemu ekspertowego DISESOR. Są to użytkownicy dodani przez administratora systemu do tabeli AUTOR w bazie danych słownika stwierdzeń. Użytkownicy nieposiadający uprawnień mają jedynie dostęp do pierwszej zakładki słownika stwierdzeń i mogą jedynie przeglądać jego zawartość.



Rys. 6.18. Zakładka słownika stwierdzeń w perspektywie modułu eksperckiego systemu DISESOR

Zakładka Słownik stwierdzeń i kontekstów stanowi przeglądarkę zawartości słownika stwierdzeń, z której mogą skorzystać użytkownicy posiadający i nieposiadający uprawnienia do korzystania z wszystkich funkcjonalności słownika stwierdzeń.

W tym oknie aplikacji zostało umieszczonych osiem tabel bezpośrednio odnoszących się do tabel bazy słownika stwierdzeń (rys. 6.19):

Rys. 6.19. Zakładka SŁOWNIK STWIERDZEŃ

1. AUTOR – tabela zawierająca informacje o użytkownikach posiadających uprawnienia do korzystania ze słownika stwierdzeń. Są to użytkownicy, którzy mają dostęp do wszystkich zakładek słownika stwierdzeń oraz mogą dodawać, usuwać oraz edytować jego zawartość w przewidzianym zakresie.
2. OBIEKT – tabela zawierająca obiekty zdefiniowane w słowniku stwierdzeń.
3. ATRYBUT – tabela zawierająca atrybuty zdefiniowane w słowniku stwierdzeń.
4. WARTOŚĆ ATRYBUTU – tabela zawierająca wartości atrybutów zdefiniowane w słowniku stwierdzeń.
5. STWIERDZENIE – tabela zawierająca stwierdzenia zdefiniowane w słowniku stwierdzeń.
6. WARIANT STWIERDZENIA – tabela zawierająca warianty stwierdzeń zdefiniowane w słowniku stwierdzeń.
7. TEKST – tabela zawierająca fragmenty tekstów składających się na treść stwierdzenia.
8. SYNONIM – tabela zawierająca synonimy wybranych słów występujących w tekście reprezentującym treść stwierdzenia.

Przeglądanie zawartości słownika stwierdzeń można prowadzić na dwa sposoby:

1. Poprzez tworzenie ścieżki przeglądania – wybierając lewym przyciskiem myszy dostępne pola w tabelach (podczas ładowania słownika stwierdzeń zapelniona jest tabela użytkowników AUTOR), powodujemy dynamiczne zmiany w tabelach powiązanych wg. schematów przedstawionych poniżej. W trakcie takiego przeglądania słownika stwierdzeń użytkownik jest prowadzony po pewnej

wyznaczonej ścieżce. Głównym zamierzeniem takiego schematu było przeglądanie słownika względem korzenia słownika, którym jest autor (użytkownik z uprawnieniami).

2. Poprzez swobodny wybór elementu słownika – naciskając prawy przycisk myszy w obrębie dowolnej tabeli powodujemy wyświetlenie w niej wszystkich pól zapisanych dla danego elementu słownika w odpowiedniej tabeli w bazie danych słownika stwierżeń. W dalszej kolejności jeżeli chcemy dowiedzieć się jakie elementy słownika są powiązane z danym elementem w wybranej tabeli postępujemy podobnie wybierając ten element lewym przyciskiem myszy, powodując dynamiczną zmianę w tabelach powiązanych.

Zakładka Dodaj/Usuń OBIEKT pozwala na definiowanie nowych obiektów zapisywanych w słowniku stwierżeń oraz usuwanie istniejących obiektów ze słownika stwierżeń. Elementy zakładki zawierają informację o cechach opisujących tworzony obiekt. Elementy zakładki okna wspomagają użytkownika w określeniu cech tworzonego obiektu. Dodanie nowego obiektu do słownika stwierżeń spowoduje automatyczne odświeżenie drzewa obiektów oraz tabeli obiektów. Użytkownik może usuwać obiekty, które sam utworzył i które nie posiadają obiektów podrzędnych oraz nie zostały jeszcze powiązane z atrybutami. Żeby usunąć obiekt, który został już powiązany z atrybutem/atributami należy w pierwszej kolejności usunąć te powiązania. Nie będzie to jednak możliwe jeżeli inny użytkownik powiązał dany obiekt z utworzonym przez siebie atrybutem/atributami.

Zakładka Dodaj/Usuń/Edytuj ATRYBUT pozwala na definiowanie nowych atrybutów zapisywanych w słowniku stwierżeń oraz edycję i usuwanie istniejących atrybutów ze słownika stwierżeń. Elementy zakładki zawierają informację o cechach opisujących tworzony atrybut. Elementy zakładki wspomagają użytkownika w określeniu cech tworzonego atrybutu. Użytkownik może usuwać atrybuty, które sam utworzył oraz które nie zostały jeszcze powiązane z obiektami. Żeby usunąć atrybut, który został już powiązany z obiektem/obiektami należy w pierwszej kolejności usunąć te powiązania. Nie będzie to jednak możliwe jeżeli inny użytkownik powiązał dany atrybut z utworzonym przez siebie obiektem/obiektami.

Zakładka Dodaj/Usuń WARTOŚĆ ATRYBUTU pozwala na definiowanie nowych wartości atrybutów zapisywanych w słowniku stwierżeń. Elementy zakładki okna zawierają informację o cechach opisujących tworzoną wartość atrybutu. Elementy zakładki wspomagają użytkownika w określeniu cech tworzonej wartości atrybutu. Użytkownik może usuwać wartości atrybutu, które sam utworzył oraz dla których nie zostały zdefiniowane jeszcze warianty stwierżenia. Żeby usunąć wartość atrybut, dla której został zdefiniowany wariant stwierżenia należy w pierwszej kolejności usunąć ten wariant stwierżenia. Nie będzie to jednak możliwe jeżeli inny użytkownik utworzył wariant stwierżenia dla danej wartości atrybutu.

Zakładka Dodaj/Usuń STWIERDZENIE pozwala na definiowanie nowych stwierżeń zapisywanych w słowniku stwierżeń. Elementy zakładki okna zawierają informację o cechach opisujących tworzone stwierżenie. Elementy zakładki

wspomagają użytkownika w określeniu cech tworzonego stwierdzenia. Użytkownik może usuwać stwierdzenia, które sam utworzył oraz dla których nie zostały zdefiniowane jeszcze warianty stwierdzenia. Żeby usunąć stwierdzenie, dla którego został zdefiniowany wariant stwierdzenia należy w pierwszej kolejności usunąć ten wariant stwierdzenia. Nie będzie to jednak możliwe jeżeli inny użytkownik utworzył wariant stwierdzenia dla danego stwierdzenia.

Zakładka Dodaj/Usuń WARIANT STWIERDZENIA pozwala na definiowanie nowych wariantów stwierdzeń zapisywanych w słowniku stwierdzeń. Elementy zakładki zawierają informację o cechach opisujących tworzone warianty stwierdzenia. Elementy zakładki wspomagają użytkownika w określeniu cech tworzonego wariantu stwierdzenia. Użytkownik, może usuwać warianty stwierdzeń, które sam utworzył.

Tabele TEKST i SYNONIM służą do definiowania i przechowywania tekstów opisujących obiekty, atrybuty i wartości atrybutów oraz łącznik w wariacie stwierdzenia, które są wykorzystywane w budowie treści stwierdzenia wynikającej z przyjętego schematu dla danego wariantu stwierdzenia. Tabela TEKST może służyć do bezpośredniego określenia opisu np. obiektu lub utworzenia klasy głównej dla danego tekstu, której elementy są definiowane w tabeli SYNONIM. Można np. utworzyć w tabeli TEKST opisy obiektu jakim jest kombajn jako: KOMBAJN, kombajn, kombajnu, maszyny, Maszyna itp. lub utworzyć w tabeli TEKST np. klasę KOMBAJN i następnie w tabeli SYNONIM dodać do niej elementy: kombajn, kombajnu, Maszyny. Takie postępowanie pozwala na uporządkowanie tych informacji opisowych co może być korzystne dla innych użytkowników wybierających tekst opisu dla swojego obiektu. Ponieważ dane opisowe wyświetlane w tabelach TEKST i SYNONIM dla wszystkich elementów systemu są przechowywane w jednej tabeli w bazie danych słownika stwierdzeń to uproszczone jest przeszukiwanie tej tabeli pod względem poszukiwania zadanego tekstu – służy do tego pole szukaj gdzie wpisując dany tekst dynamicznie są wyszukiwane teksty zawierające wpisywaną frazę.

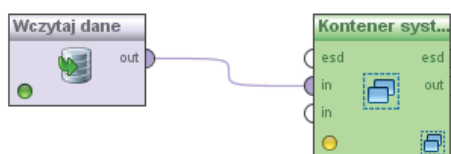
Integracja słownika ze szkieletowym systemem ekspertowym

Słownik stwierdzeń został zintegrowany ze szkieletowym systemem ekspertowym na trzech poziomach. Pierwszy został opisany powyżej i dotyczy modułu słownika stwierdzeń pozwalającego na dodawanie i edycję stwierdzeń w bazie danych. Drugi poziom integracji dotyczy edytorów trzech podstawowych modeli wnioskovania. Trzeci poziom integracji jest realizowany na etapie wykonywania procesów wnioskovania. W systemie DISESOR, podobnie jak ma to miejsce w oprogramowaniu RapidMiner, istnieje określona struktura danych opierająca się na tabeli, w której kolumny są określane mianem atrybutów a w wierszach umieszczane są kolejne rekordy danych. Każdy atrybut odpowiada innej zmiennej, która może reprezentować np. zmienną procesową. Dane przechowywane w tabeli mogą być reprezentowane przez wartości ciągłe (zmiennoprzecinkowe lub całkowite) lub dyskretne (ciąg znaków). Dostępne są też typy danych repre-

zentujące np. datę, godzinę czy typy specjalne jak np. pozycja rekordu na liście. Nazwa kolumny w tabeli odpowiada nazwie atrybutu. W przypadku atrybutu dyskretnego, wartości widoczne w komórkach przyjmują jeden z dostępnych wariantów.

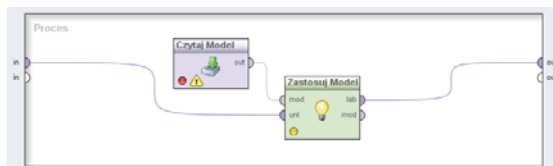
Każdy edytor modeli wnioskowania dostępny w module eksperckim, pozwala na zbudowanie modelu od podstaw lub edycję już istniejącego. Zakłada się, że w większości przypadków wyjście modelu będzie połączone z jednym ze stwierdzeń zapisanych w słowniku. W takim przypadku nazwa atrybutu wynikowego musi odpowiadać etykietce stwierdzenia a wynikiem działania modelu w trakcie wnioskowania jest wybór jednego z wariantów stwierdzenia. Etykiety wariantów stwierdzenia również muszą być określone w trakcie budowania modelu. W celu zminimalizowania ryzyka wystąpienia błędu, który mógłby skutkować nieprawidłowym działaniem systemu ekspertowego, dodano do każdego z edytorów okno dialogowe, które łączy się automatycznie ze słownikiem stwierdzeń, wyświetla wszystkie dostępne stwierdzenia i pozwala na wybór jednego z nich. Po potwierdzeniu wyboru etykiety stwierdzenia oraz wariantów automatycznie dodawane są do modelu.

W przypadku budowania klasyfikatora w oparciu o dane, należy zwrócić szczególną uwagę, aby dane trenujące zostały prawidłowo przygotowane a nazwa atrybutu klasy oraz jej warianty pokrywały się z etykietami odpowiedniego stwierdzenia w słowniku. Wynik działania dowolnego klasyfikatora może być związany z konkretnym stwierdzeniem i wariantem. W tym przypadku nazwa atrybutu decyzyjnego jest etykietą stwierdzenia, a wartość atrybutu jest etykietą wariantu. Końcowa decyzja będąca wynikiem działania systemu ekspertowego jest prezentowana użytkownikowi w postaci komunikatu tekstowego związanego z określonym stwierdzeniem oraz wariantem. W celu automatyzacji procesu łączenia wyniku wnioskowania, które zawierają etykiety wariantów stwierdzeń z treścią stwierdzenia, zaimplementowano kontener systemu DISE-SOR (rys. 6.20), który pozwala na budowanie podprocesów w jego „wnętrzu”.



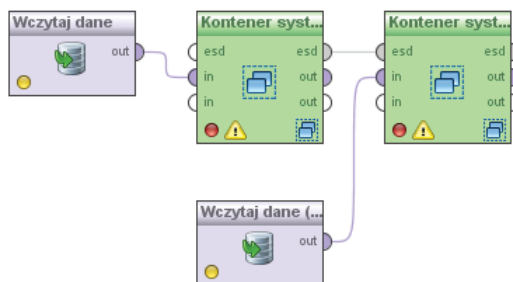
Rys. 6.20. Fragment schematu wnioskowania z zastosowaniem kontenera

Zaletą takiego rozwiązania jest możliwość dokładnego śledzenia zagnieżdżonego fragmentu procesu (rys. 6.21). Kontener śledzi dane wejściowe oraz wyjściowe operatora, którego zadaniem jest zastosowanie utworzonego modelu. Jeżeli kontener rozpozna nazwę atrybutu wynikowego jako etykietę jednego ze stwier-



Rys. 6.21. Fragment procesu wnioskowania umieszczony wewnątrz kontenera

dzeń, rozpocznie przypisywanie treści stwierdzenia do każdego z rekordów na podstawie etykiet wariantów. Treści są zapisywane w strukturze danych o roboczej nazwie *esd*, która została opracowana na potrzeby implementacji systemu ekspertowego. Dane te mogą być przekazywane pomiędzy kontenerami, w ten sposób struktura danych jest rozszerzana o nowe wyniki wnioskowania z kolejnych modeli wykorzystanych w ramach jednego procesu wnioskowania (rys. 6.22).



Rys. 6.22. Przykład zastosowania dwóch połączonych ze sobą kontenerów

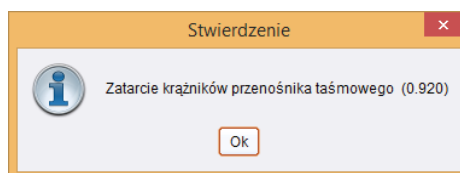
Istnieje możliwość zastosowania większej liczby kontenerów, w których użyto pojedyncze modele lub pojedynczego kontenera zawierającego cały proces. Wszystkie dane zapisane w strukturze *esd* (etykiety stwierdzeń i wariantów, treści stwierdzeń, opcjonalnie również dane wejściowe modeli) odpowiadają całemu procesowi i mogą zostać zapisane bezpośrednio w repozytorium danych w postaci plików *.log* lub przekształcone do struktury danych zgodnej z oprogramowaniem RapidMiner (rys. 6.23).

Wynik działania ostatniego modelu jest jednocześnie wynikiem działania całego procesu. Bezwzględnie wymagane jest, aby atrybut decyzyjny tego modelu był powiązany ze stwierdzeniem, ponieważ rezultat jego działania docelowo ma być prezentowany na ekranie dyspozytora (rys. 6.24).

Stwierdzenie kontekstowe nie jest bezpośrednio prezentowane dyspozytorowi, ale może mieć istotny wpływ na sposób realizacji procesu. Analogicznie do stwierdzenia, stwierdzenie kontekstowe również jest reprezentowane przez etykie-

Row No.	1_INPUT_wartosc	1_OUTPUT_(Tree): Class	1_OUTPUT_(Tree): Confidence	1_OUTPUT_(Tree): Message
1	72	STWtempok	1	Temperatura silnika jest prawidłowa
2	72	STWtempok	1	Temperatura silnika jest prawidłowa
3	72	STWtempok	1	Temperatura silnika jest prawidłowa
4	72	STWtempok	1	Temperatura silnika jest prawidłowa
5	72	STWtempok	1	Temperatura silnika jest prawidłowa
6	72	STWtempok	1	Temperatura silnika jest prawidłowa
7	72	STWtempok	1	Temperatura silnika jest prawidłowa
8	72	STWtempok	1	Temperatura silnika jest prawidłowa
9	72	STWtempok	1	Temperatura silnika jest prawidłowa
10	204	STWtempoohigh	1	Temperatura silnika jest zbyt wysoka
11	204	STWtempoohigh	1	Temperatura silnika jest zbyt wysoka
12	204	STWtempoohigh	1	Temperatura silnika jest zbyt wysoka
13	204	STWtempoohigh	1	Temperatura silnika jest zbyt wysoka
14	204	STWtempoohigh	1	Temperatura silnika jest zbyt wysoka
15	204	STWtempoohigh	1	Temperatura silnika jest zbyt wysoka
16	204	STWtempoohigh	1	Temperatura silnika jest zbyt wysoka
17	72	STWtempok	1	Temperatura silnika jest prawidłowa
18	72	STWtempok	1	Temperatura silnika jest prawidłowa
19	72	STWtempok	1	Temperatura silnika jest prawidłowa

Rys. 6.23. Dane zwracane przez kontener w formie tabelarycznej



Rys. 6.24. Przykład komunikatu wygenerowanego przez moduł ekspercki przy użyciu zdefiniowanego stwierdzenia

tę stwierdzenia i etykiety wariantów. Etykieta stwierdzenia kontekstowego jest zapisana w nazwie atrybutu, a etykiety wariantów są widoczne w komórkach tabeli jako wartości atrybutu. Stwierdzenie kontekstowe może być użyte jako cecha podstawowa, czyli jako jeden z atrybutów będący wejściem klasyfikatora wchodzącego w skład procesu wnioskowania. Korzystając z odpowiedniego operatora można również wykorzystać dodatkowe mechanizmy fuzji klasyfikatorów, które są sterowane za pomocą cechy kontekstowej.

6.5 Podsumowanie

Celem badań zaprezentowanych w tym rozdziale było opracowanie szkieletowego systemu ekspertowego działającego w trybach online oraz offline, realizującego wnioskowania diagnostyczne, nadzorującego przebieg procesów oraz wspomagającego użytkownika w podejmowaniu decyzji w sytuacjach kryzysowych. Utworzono projekt systemu informatycznego oraz zaimplementowano moduł szkieletowego systemu ekspertowego w postaci wtyczki aplikacji RapidMiner. Utworzone oprogramowanie przetestowano zgodnie z dobrą praktyką tworzenia oprogramowania (testy jednostkowe) oraz z zastosowaniem danych symulacyjnych pozyskanych w efekcie realizacji czynnego numerycznego eksperymentu diagnostycznego. Uzyskane rozwiązanie charakteryzuje się następującymi cechami:

- baza wiedzy szkieletowego systemu ekspertowego ma strukturę umożliwiającą gromadzenie wiedzy reprezentowanej w różnej formie oraz pozyskanej z różnych źródeł,
- struktura bazy wiedzy oraz zastosowane jądro aplikacji RapidMiner umożliwia wielowątkowe i wieloaspektowe wnioskowanie diagnostyczne w trybach online oraz offline,
- moduł wyposażony jest w różne elementarne mechanizmy wnioskowania (wnioskowanie z wykorzystaniem edytowalnych drzew decyzyjnych, wnioskowanie z wykorzystaniem systemu rozmytego, wnioskowanie z wykorzystaniem sieci Bayesa, wnioskowanie z uwzględnieniem kontekstu) oraz w algorytmy syntezy wyników wnioskowań elementarnych (również z uwzględnieniem kontekstu),
- opracowano różne sposoby wyznaczenia diagnozy wypadkowej dla przypadku procesów wnioskowania prowadzonych wieloma ścieżkami - z użyciem metody głosowania większościowego, wyboru klasyfikatora z uwzględnieniem kontekstu, oraz z użyciem meta-klasyfikatora,
- moduł wyposażony jest w intuicyjny edytor baz wiedzy oraz słownik stwierdzeń umożliwiający użytkownikowi końcowemu czytelną prezentację wyników wnioskowania z wykorzystaniem języka naturalnego,
- dzięki wykorzystaniu wybranych metod reprezentacji wiedzy i wnioskowania możliwe jest tworzenie tzw. wielomodeli diagnostycznych lub stosowanie określonego środka reprezentacji wiedzy w zależności od klasy monitorowanych obiektów,
- szkieletowy system ekspertowy zintegrowany jest z repozytorium danych, a przez to również z pozostałymi modułami systemu DISESOR,
- moduł szkieletowego systemu ekspertowego zaimplementowany jest z wykorzystaniem bibliotek oprogramowania RapidMiner z zastosowaniem języka JAVA.

Literatura

- [6.1] V. Akman. Contesti in intelligenza artificiale: una fugace rassegna (context in artificial intelligence: A fleeting overview). *La Svolta Contestuale. McGraw-Hill*, s. 147–166, 2002.
- [6.2] V. Akman, M. Surav. Steps toward formalizing context. *AI Magazine*, 17(3):55–72, 1996.
- [6.3] H. C. Akthar F. RapidMiner 5, Operator Reference. Strona domowa oprogramowania (dn. 05.12.2014): www.rapid-i.com, 2012.
- [6.4] M. Bednarski, W. Cholewa, W. Frid. Identification of sensitivities in bayesian networks. *Engineering Applications of Artificial Intelligence*, 17(4):327–335, 2004.
- [6.5] BigML. Machine learning REST API. Strona domowa oprogramowania (dn. 05.12.2014): www.bigml.com, 2014.
- [6.6] W. Cholewa. Expert systems in technical diagnostics. J. Korbicz, Z. Kowalczyk, J. Kościelny, W. Cholewa, redaktorzy, *Fault Diagnosis. Models, Artificial Intelligence, Applications*, s. 591–631. Springer Berlin Heidelberg, 2004.

- [6.7] W. Cholewa. Multimodal statement networks for diagnostic applications. *Proceedings of ISMA2010 – International Conference on Noise and Vibration Engineering*, s. 817–830, Leuven, Belgium, Sep 20-22, 2010.
- [6.8] W. Cholewa, i in. *Szkieletowy system doradczy DIADYN*, vol. 137 serii *Zeszyty*. Politechnika Śląska, Katedra Podstaw Konstrukcji Maszyn, Gliwice, 2008.
- [6.9] W. Cholewa, W. Pedrycz. *Systemy doradcze*. Skrypty uczelniane, nr 1447. Politechnika Śląska, Gliwice, 1987.
- [6.10] W. Cholewa, T. Rogala, P. Chrzanowski, M. Amarowicz. Statement networks development environment REx. P. Jędrzejowicz, N. Nguyen, K. Hoang, redaktorzy, *Computational Collective Intelligence. Technologies and Applications*, vol. 6923 serii *Lecture Notes in Computer Science*, s. 30–39. Springer Berlin Heidelberg, 2011.
- [6.11] DRools. Business Rules Management System (BRMS) solution. Strona domowa oprogramowania (dn. 31.12.2014): www.jboss.org/drools/, 2014.
- [6.12] FuzzyLite. Free and open-source fuzzy logic control library programmed in c++ for multiple platforms. Strona domowa oprogramowania (dn. 21.05.2014): www.fuzzylite.com, 2014.
- [6.13] FuzzyTech. Software development tools for fuzzy logic and neural-fuzzy solutions. Strona domowa oprogramowania (dn. 21.05.2014): <http://www.fuzzytech.com/>, 2014.
- [6.14] A. Grzegorzcyk. *Zarys logiki matematycznej*. PWN, Warszawa, 1981.
- [6.15] R. Hickey. The Clojure Programming Language. Strona domowa oprogramowania (dn. 31.12.2014): www.clojure.org, 2014.
- [6.16] F. V. Jensen, T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer Publishing Company, Incorporated, wyd. 2nd, 2007.
- [6.17] jFuzzyLogic. The most complete fuzzy logic library in java. Strona domowa oprogramowania (dn. 21.05.2014): <http://jfuzzylogic.sourceforge.net/html/index.html>, 2014.
- [6.18] jRuleEngine. Java Rule Engine API - JSR-94. Strona domowa oprogramowania (dn. 05.12.2014): www.jruleengine.sourceforge.net, 2014.
- [6.19] P. N. Ken Anderson, Tim Hickey. JScheme. Strona domowa oprogramowania (dn. 31.12.2014): www.jscheme.sourceforge.net, 2014.
- [6.20] J. H. Kotelly. Context logic. fundamental concepts notations, and derived notions. *Journal of Formal Logic*, 11(4):431–446, 1970.
- [6.21] L. Kuncheva. *Combining pattern classifiers: Methods and algorithms*. New Jersey: Wiley-Interscience, 2004.
- [6.22] LFLC2000. Special software package for design and performance of fuzzy logic controllers. Strona domowa oprogramowania (dn. 21.05.2014): <http://irafm.osu.cz/en/c100-software/>, 2014.
- [6.23] J. Liebowitz. *The Handbook of Applied Expert Systems*. CRC Press, Inc., Boca Raton, FL, USA, wyd. 1st, 1997.
- [6.24] J. Łęski. *Systemy neuronowo-rozmyte*. Wydawnictwo Naukowo-Techniczne, 2008.
- [6.25] E. MAMDANI, S. ASSILIAN. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Hum.-Comput. Stud.*, 51(2):135–147, Sier. 1999.

- [6.26] J. McCarthy. Generality in artificial intelligence. *Commun. ACM*, 30(12):1030–1035, Gru. 1987.
- [6.27] M. Mikalsen, A. Kofod-Petersen. Representing and reasoning about context in a mobile environment. *REVUE D'INTELLIGENCE ARTIFICIELLE (RIA)*, 19:479–498, 2005.
- [6.28] W. Moczulski. *Diagnostyka techniczna. Metody pozyskiwania wiedzy*. Wydawnictwo Politechniki Śląskiej, Gliwice, 2002.
- [6.29] OpenL Tablets. Easy Business Rules. Strona domowa oprogramowania (dn. 31.12.2014): www.openl-tablets.sourceforge.net, 2014.
- [6.30] OpenRules. Business Rules and Decision Management System. Strona domowa oprogramowania (dn. 21.05.2014): www.openrules.com, 2014.
- [6.31] QuickDT. A fast and easy to use decision tree learner in java. Strona domowa oprogramowania (dn. 05.12.2014): www.github.com/sanity/quickdt, 2014.
- [6.32] Roolie. A simple Java Rule Engine. Strona domowa oprogramowania (dn. 05.11.2014): www.roolie.sourceforge.net, 2014.
- [6.33] L. Rutkowski. *Metody i techniki sztucznej inteligencji*. Wydawnictwo Naukowe PWN, Warszawa, 2005.
- [6.34] M. Sugeno. *Industrial Applications of Fuzzy Control*. Elsevier Science Inc., New York, NY, USA, 1985.
- [6.35] A. Timofiejczuk. *Metodyka wnioskowania z uwzględnieniem kontekstu w diagnostyce maszyn*. Wydawnictwo Politechniki Śląskiej, Gliwice, 2011.
- [6.36] TuProlog. Light-weight Prolog system for distributed applications. Strona domowa oprogramowania (dn. 11.09.2014): www.apice.unibo.it/xwiki/bin/view/Tuprolog/, 2014.
- [6.37] Weka. Data mining software in java. Strona domowa oprogramowania (dn. 13.01.2015): <http://www.cs.waikato.ac.nz/ml/weka/>, 2015.
- [6.38] M. Woźniak. *Metody fuzji informacji dla komputerowych systemów rozpoznawania*. OWPW, Warszawa, 2006.
- [6.39] J. Yang, C. Ye, X. Zhang. An expert system shell for fault diagnosis. *Robotica*, 19(6):669–674, 10 2001.
- [6.40] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.*, 100:9–34, Kwi. 1999.