

A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19

AKADEMIA OBRONY NARODOWEJ

CENTRUM INFORMATYKI

Do użytku służbowego

Egz. Nr2

MODEL EKSPERTOWEGO SYSTEMU WSPOMAGANIA DECYZJI SZEFA OBRONY PRZECIWLOTNICZEJ ZWIĄZKU TAKTYCZNEGO I DOWÓDCY PUŁKU PRZECIWLOTNICZEGO

CZEŚĆ III

System ekspertowy

Biblioteka Główna
Akademii Obrony Narodowej
S/3895 cz. 3



05-003895-002-0

WARSZAWA

68911



Do użytku służbowego



**MODEL EKSPERTOWEGO SYSTEMU WSPOMAGANIA DECYZJI SZEFA
OBRONY PRZECIWLOTNICZEJ ZWIĄZKU TAKTYCZNEGO I DOWÓDCY
PUŁKU PRZECIWLOTNICZEGO.**

Część III System ekspertowy.

Kierownik tematu: mjr mgr Grzegorz Kott
Kierownictwo naukowe: płk dr hab. inż. Czesław Flanek

Wykonawcy:

płk dr hab. inż. Czesław Flanek
mjr mgr Grzegorz Kott
Barbara Rogowiec

EKSPERT - OP

6 . 2 . 0 . 0

Wstęp - płk Cz.Flanek, mjr G.Kott

Rozdział 2 - mjr G.Kott

Rozdział 3

3.1.1. - płk Cz.Flanek

3.1.2., 3.1.3. - mjr G.Kott

Rozdział 4

4.1.1. - płk Cz.Flanek

4.1.2., 4.1.3. - mjr G.Kott

Edycja i redakcja całości - B.Rogowiec

Eksperti - konsultanci (pozyskiwanie wiedzy eksperckiej):

gen.bryg.dr inż. Stanisław Czepielik

płk dr inż. Ryszard Kuriata

płk dypl. Józef Gadzała

1. WSTĘP	2
2. PAKIET SZTUCZNEJ INTELIGENCJI SPHINX 2.2.....	3
2.1. SZKIELETOWY SYSTEM EKSPERTOWY PC-SHELL	4
2.1.1. Język reprezentacji wiedzy.....	10
2.1.2. Struktura języka opisu bazy wiedzy	12
2.1.3. Deklaracje źródeł wiedzy.....	14
2.1.3.1. Źródła typu kb.....	15
2.1.3.2. Źródła typu neural_net	16
2.1.3.3. Źródła typu metaphor i what_is	16
2.1.4. Blok faset	18
2.1.4.1. Fasety ask i single.....	18
2.1.4.2. Faseta unit.....	19
2.1.4.3. Faseta val	20
2.1.4.4. Faseta query	22
2.1.4.5. Faseta param	23
2.1.4.6. Faseta picture	24
2.1.4.7. Faseta sound	24
2.1.4.8. Faseta video	24
2.1.5. Blok opisu faktów	25
2.1.6. Blok opisu reguł	26
2.1.7. Blok control	28
2.2. KOMPUTEROWY SYSTEM WSPOMAGANIA INŻYNIERA WIEDZY CAKE.....	36
2.2.1. Właściwości aplikacji.....	36
2.2.2. Źródła wiedzy.....	37
2.2.3. Obiekty bazy wiedzy.....	38
2.2.4. Fasety atrybutów bazy wiedzy	38
2.2.5. Fakty bazy wiedzy	42
2.2.6. Reguły bazy wiedzy	43
2.2.7. Program aplikacji.....	45
3. ZASTOSOWANIE SYSTEMU EKSPERTOWEGO DO WSPOMAGANIA DECYZJI SZEFA OPL ZT I DOWÓDCY PLOT.....	46
3.1. OPTYMALIZACJA UGRUPOWANIA WOJSK OPL.....	46
3.1.1. Opis problemu	46
3.1.2. System ekspertowy UGRUPOWANIE	56
3.1.3. Zasady użytkowania systemu UGRUPOWANIE.....	67
3.2. WSPOMAGANIE WYPRACOWANIA DECYZJI OGNIOWEJ W PUŁKU PRZECIWLOTNICZYM....	76
3.2.1. Opis problemu	77
3.2.2. System ekspertowy DECYZJA.....	91
3.2.3. Zasady użytkowania systemu DECYZJA	100
4. BIBLIOGRAFIA	111

1. Wstęp

Niniejsze opracowanie stanowi opis zrealizowanych eksperymentalnych, laboratoryjnych systemów ekspertowych, wspomagających wybrane funkcje procesu decyzyjnego szefa obrony przeciwlotniczej związku taktycznego (OPL ZT) i dowódcy pułku przeciwlotniczego (pplot). Potencjalne obszary zastosowania takich systemów we wspomaganie dowodzenia obroną przeciwlotniczą są następujące:

- zastosowanie w procesie dowodzenia taktycznego i ogniowego szefa OPL ZT i dowódcy pplot bezpośrednio w wojskach OPL,
- zastosowanie w procesie kształcenia kadr dowódczych i sztabowych wojsk OPL w ramach Akademii Obrony Narodowej i Centrum Szkolenia Wojsk OPL.

W części pierwszej (wydanej w 1995 roku) opracowano projekt koncepcyjno - techniczny modelu ekspertowego systemu wspomaganie decyzji dowódcy pułku rakiet przeciwlotniczych (prplot) i szefa obrony przeciwlotniczej związku taktycznego. W pracy ujęto:

- zarys problematyki systemów ekspertowych,
- definicje procesów decyzyjnych w pracy szefa OPL ZT i dowódcy prplot,
- definicję uproszczonego modelu wspomaganie decyzji w obronie przeciwlotniczej ZT i koncepcji proponowanego rozwiązania (dedukcyjnej bazy danych).

Część druga (wydana w 1997 roku) była kontynuacją prac podjętych w części pierwszej. Zawierała formalny opis modelu fizycznego przetwarzania danych systemu wspomaganie dowodzenia operacyjno-taktycznego i ogniowego obroną przeciwlotniczą związku taktycznego. Użyty formalizm (Nowoczesna Analiza Strukturalna Yourdona/DeMarco) pozwolił:

- wyróżnić główne procesy (funkcje) realizowane przez system,
- określić dane niezbędne do realizacji każdej funkcji,
- określić informacje będące efektem każdej z funkcji.

Część trzecia (system ekspertowy) jest kontynuacją prac podjętych w części pierwszej i drugiej. Wykonana jest zgodnie z planem prac naukowo-badawczych na lata dziewięćdziesiąte (EKSPERT - OP, 6 . 2 . 0 . 0). Aplikacje eksperckie, będące efektem prac, są w dyspozycji Centrum Informatyki Akademii Obrony Narodowej.

2. Pakiet sztucznej inteligencji SPHINX 2.2

Pakiet SPHINX stanowi nowoczesne środowisko tworzenia, rozwoju i uruchamiania aplikacji, wykorzystujących mechanizmy sztucznej inteligencji. Pozwala na generowanie systemów ekspertowych oraz sieci neuronowych mogących, podczas rozwiązywania złożonych problemów, wzajemnie się uzupełniać. Jako produkt szkieletowy, niezależny dziedzinowo, nadaje się doskonale do wykorzystania w bardzo wielu obszarach zastosowań. Pakiet został uhonorowany nagrodą II stopnia podczas XI Międzynarodowych Targów Oprogramowania SOFTARG '97.

W skład pakietu wchodzi:

- szkieletowy system ekspertowy PC-SHELL,
- komputerowy system wspomagania inżynierii wiedzy CAKE,
- symulator sieci neuronowych NEURONIX .

Architektura pakietu SPHINX jest hybrydowa, integrująca system ekspertowy, system wspomagania inżyniera wiedzy oraz symulator sieci neuronowej. Połączenie to nie jest jedynie mechaniczne, lecz nastąpiła tu integracja zarówno na poziomie architektury systemu jak i języka reprezentacji wiedzy. Symulator sieci neuronowej występuje tu w podobnej roli jak źródło wiedzy w ramach architektury tablicowej. To podobieństwo wyraża się nie tylko w zakresie funkcjonalnym ale również formalnym, dotyczącym sposobu reprezentacji wiedzy. Korzyścią płynącą z takiego rozwiązania i oryginalną cechą pakietu SPHINX jest możliwość tworzenia hybrydowych aplikacji, gdzie odbywa się pełna kooperacja pomiędzy symulatorami sieci neuronowych a systemem ekspertowym.

System CAKE (ang. Computer-Aided Knowledge Engineering) został stworzony jako narzędzie wspomagające inżyniera wiedzy w procesie tworzenia i rozwijania aplikacji, uruchamianych z poziomu systemu ekspertowego PC-SHELL. Dostarczając rozbudowanych mechanizmów edycji i kontroli poprawności bazy wiedzy, system CAKE umożliwia szybkie i wygodne zaprojektowanie struktury bazy wiedzy oraz zdefiniowanie wszystkich jej elementów, zabezpieczając jednocześnie przed trudnymi do wychwycenia błędami. Użytkownik zwolniony jest z konieczności dokładnego zaznajamiania się ze składnią obowiązującego języka opisu wiedzy oraz korzystania z zewnętrznych edytorów tekstu w celu stworzenia aplikacji eksperckiej. Zastosowanie systemu kontroli uprawnień uniemożliwia osobom nieupoważnionym ingerencję w kod źródłowy aplikacji.

Do konstrukcji modelowych systemów ekspertowych UGRUPOWANIE i DECYZJA wykorzystane zostały podsystemy PC-SHELL oraz CAKE, stąd w niniejszym opracowaniu zostanie przedstawiona charakterystyka jedynie tych podsystemów.

2.1. Szkieletowy system ekspertowy PC-SHELL

System ekspertowy jest systemem informatycznym, wykorzystującym zapisaną wiedzę eksperta do rozwiązywania problemów, które normalnie wymagają ludzkiej inteligencji. System ekspertowy symuluje działanie eksperta w sposobie użytkownika - poprzez analizę odpowiedzi użytkownika na pytania kierowane do niego przez system. Właściwości, charakteryzujące systemy ekspertowe, są następujące:

- są narzędziem kodyfikacji wiedzy eksperckiej,
- mają zdolność rozwiązywania problemów specjalistycznych, w których dużą rolę odgrywa doświadczenie, a wiedza ekspercka jest dobrem rzadkim i kosztownym,
- zwiększają dostępność ekspertyzy,
- unifikują wnioskowanie - te same przesłanki determinują tą samą konkluzję,
- poziom ekspertyzy jest stabilny - jej jakość nie zależy od warunków zewnętrznych i czasu pracy systemu,
- jawna reprezentacja wiedzy w postaci zrozumiałej dla użytkownika końcowego,
- zdolność do objaśniania znalezionych przez system rozwiązań,
- możliwość przyrostowej budowy i pielęgnacji bazy wiedzy.

Moc programu ekspertowego w zakresie rozwiązywania danego problemu tkwi w zakodowanej w nim wiedzy, a nie w formalizmie i schematach wnioskowania, których ten program używa. Można to wyrazić w postaci stwierdzenia: im pełniejsza wiedza, tym szybciej uzyskuje się rozwiązanie. Tak więc problem z posiadaniem pełnej wiedzy tkwi w bazie wiedzy, a nie w sposobie realizacji procesu wnioskowania systemu ekspertowego. Oznacza to, że aby zbudować inteligentny program, należy go wyposażyć w dużą ilość dobrej jakości, specyficznej wiedzy o danym przedmiocie.

Ze względu na sposoby realizacji systemy ekspertowe możemy podzielić na dwie grupy:

- systemy dedykowane, tworzone od podstaw przez inżyniera wiedzy;
- systemy szkieletowe, będące gotowymi systemami ekspertowymi z pustą bazą wiedzy.

Jedną z najpoważniejszych barier w stosowaniu systemów szkieletowych w Polsce jest ich cena oraz brak lokalnej wersji językowej, która:

- uniemożliwia jawną reprezentację wiedzy w postaci zrozumiałej dla użytkownika końcowego,
- redukuje zdolność do objaśniania znalezionych przez system rozwiązań.

Ograniczenia te zostały zredukowane w szkieletowym systemie ekspertowym PC-SHELL .

Ogólnie system PC-SHELL charakteryzują następujące cechy:

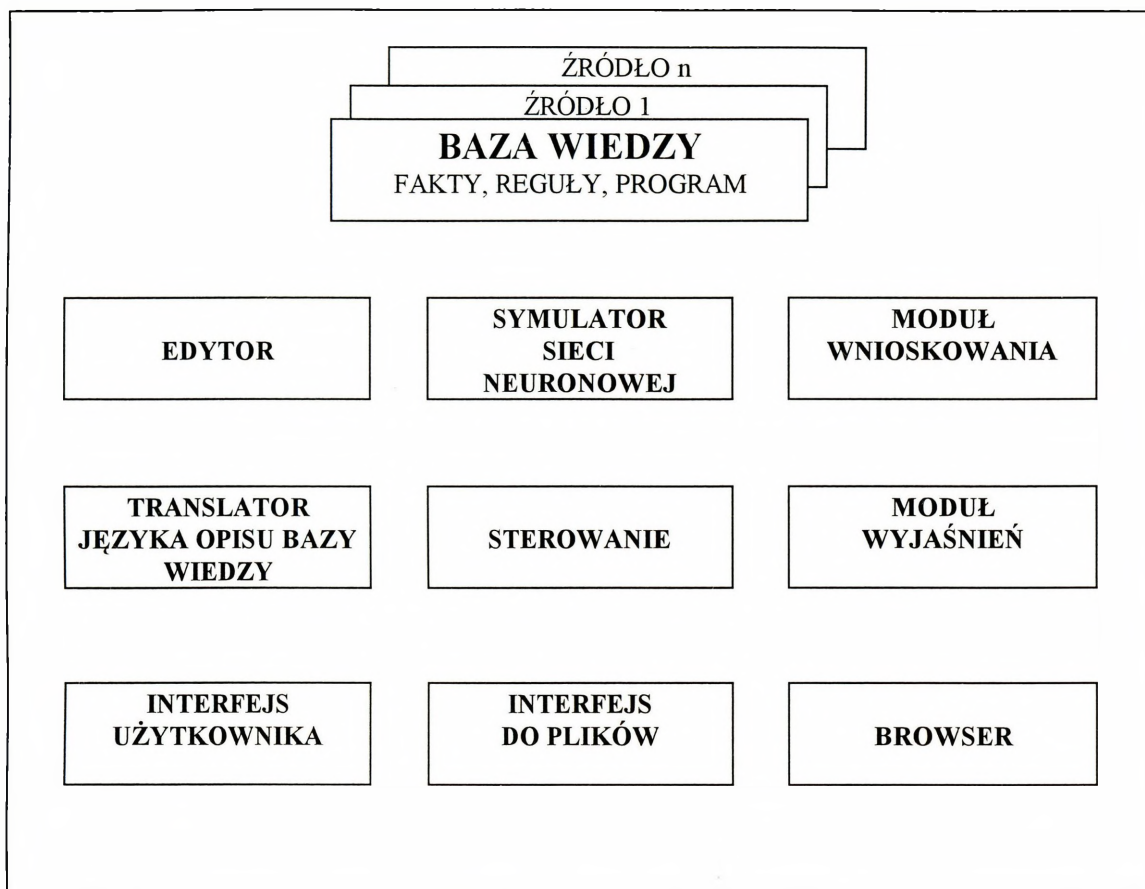
- 1) W zakresie struktury systemu:
 - elementy architektury tablicowej,
 - hybrydowość systemu.

- 2) W zakresie reprezentacji wiedzy:
 - deklaratywna reprezentacja wiedzy w formie reguł i faktów,
 - algorytmiczna (proceduralna) reprezentacja wiedzy w formie programu zawartego w bloku sterowania (control),
 - pełne rozdzielenie wiedzy eksperckiej i procedur sterowania,
 - wiedza o charakterze rozproszonym zawarta w sieci neuronowej,
 - wiedza ekspercka może być zawarta w kilku źródłach wiedzy.

- 3) W zakresie wyjaśnień:
 - wyjaśnienia typu JAK (HOW),
 - wyjaśnienia typu DLACZEGO (WHY),
 - wyjaśnienia typu CO TO JEST, dla pojęć zawartych w bazie wiedzy (WHAT IS),
 - METAFORY dotyczące reguł użytych w bazie wiedzy.

System PC-SHELL składa się z następujących elementów (rys. 1):

1. modułu sterującego,
2. translatora języka opisu bazy wiedzy,
3. modułu wnioskującego,
4. modułu wyjaśnień,
5. symulatora sieci neuronowych,
6. zewnętrznego edytora bazy wiedzy,
7. interfejsu użytkownika,
8. interfejsu do plików dyskowych.



Rys. 1. Architektura systemu PC-SHELL

Poniżej zostaną scharakteryzowane najważniejsze funkcjonalnie elementy architektury systemu PC-SHELL.

Moduł sterujący

Podstawowym zadaniem modułu sterującego jest koordynowanie wszelkich procesów realizowanych przez system PC-SHELL. Jednym z jego zadań jest komunikacja z użytkownikiem poprzez interfejs użytkownika. Interfejs ten pracuje w trybie tekstowym i opracowany został w taki sposób, by sugerował użytkownikowi określone warianty działania w danym kontekście, uwalniając go tym samym od konieczności ich pamiętania.

Bazy wiedzy, translator języka opisu bazy wiedzy

W odróżnieniu od klasycznych programów komputerowych wiedza zawarta w bazie wiedzy opisuje dziedzinę problemową bez podania szczegółowego sposobu rozwiązania danego problemu (algorytmu). Dzięki temu jest dużo bardziej czytelna, nawet dla osób nie będących specjalistami w dziedzinie systemów ekspertowych, ujmuje bowiem głównie merytoryczną stronę zagadnienia. Baza wiedzy przyjmuje na ogół postać pliku na dysku komputera. Przechowywana w nim wiedza jest zapisana za pomocą określonego języka

reprezentacji wiedzy, na który najczęściej składa się opis faktów (wiedza o charakterze faktograficznym), reguł stosowanych w procesie wnioskowania oraz w przypadku niektórych systemów - metareguł, opisujących strategię rozwiązywania danego problemu.

PC-SHELL jest systemem regułowym, stąd całość wiedzy o charakterze heurystycznym jest kodowana za pomocą reguł i faktów. Podstawową metodą reprezentacji wiedzy jest trójka (obiekt, atrybut, wartość). Regułowa reprezentacja wiedzy wykorzystuje formalne zapisy i twierdzenia logiki. Reguły zapisywane są w następującej postaci:

JEŻELI fakt₁ I fakt₂ I ... I fakt_n TO konkluzja.

Oznacza to, że jeżeli prawdziwe są przesłanki reguły (fakt₁, fakt₂, ... fakt_n) to prawdziwa jest także konkluzja. W związku z tym, iż przesłanki oraz konkluzje traktuje się jako wartości logiczne (prawda, fałsz), to reguły takie nazywa się predykatami, a ich przetwarzanie rachunkiem predykatów. Jedną z możliwych postaci wyrażen rachunku predykatów są tzw. klauzule Horna. Ich struktura jest zbliżona do generalnej postaci reguł wnioskowania z pewnymi dodatkowymi ograniczeniami wynikającymi z zasad rachunków predykatów - przesłanki muszą być w postaci koniunktywnej oraz dopuszczalna jest tylko jedną konkluzja. Jak już wcześniej wspomniano, w bazie wiedzy systemu, oprócz wiedzy nabywanej przez system, istnieje wiedza (metawiedza) opisująca sposób zachowania się systemu - np. w trakcie procesu wnioskowania. Wiedza taka jest najczęściej zapisywana w postaci reguł. Reguły te mogą być poddawane zmianom w trakcie pracy systemu - mogą być one zmieniane przez sam system.

System PC-SHELL ma architekturę tablicową. Istota systemów tablicowych opiera się na analogii ze sposobem, w jaki może współpracować kilku ekspertów rozwiązujących pewien problem. Każdy z nich włącza się do rozwiązania wspólnego problemu w zakresie jego kompetencji, a tworzone przez nich rozwiązania pośrednie zapisywane są na tablicy. W systemach tablicowych wiedza ekspercka nie jest zgromadzona w jednym miejscu (np. pliku), lecz rozproszona w postaci kilku wyodrębnionych fizycznie modułów, zwanych źródłami wiedzy (ang. knowledge sources). Poszczególne źródła wiedzy odpowiadają ekspertom z podanego przykładu. Ponadto zakłada się istnienie wspólnej, tj. dostępnej dla wszystkich źródeł, tablicy komunikacyjnej (ang. blackboard), będącej rodzajem pamięci roboczej. Każde źródło może posiadać własny mechanizm wnioskujący. W systemie PC-SHELL zaimplementowano niektóre elementy przedstawionego modelu tablicowego. W obecnej wersji system umożliwia podzielenie bazy wiedzy na mniejsze fragmenty, wyodrębnione tematycznie (problemowo). Narzędziem umożliwiającym załadowanie i uaktywnienie wybranych źródeł wiedzy są instrukcje getSource i solve. Nie zakłada się formalnie wyodrębnionych mechanizmów wnioskujących przypisanych do źródeł, ponieważ w przypadku komputerów sekwencyjnych nie ma to większego znaczenia. Natomiast dostępne w systemie PC-SHELL instrukcje inicjujące mechanizm wnioskujący pozwalają osiągnąć podobny efekt praktyczny. Rolę tablicy pełni tu część obszaru pamięci operacyjnej, w którym umieszczana jest baza wiedzy.

W systemie PC-SHELL wymaga się istnienia modułu głównego bazy wiedzy, który jako jedyny może być ładowany bezpośrednio w ramach opcji Plik. Wyłącznie moduł główny bazy wiedzy może zawierać opisy plików, faset oraz program. Źródła wiedzy mogą zawierać tylko reguły i/lub fakty. Moduł główny nie może zawierać reguł i faktów, pełniąc jedynie rolę sterującą oraz funkcję kontrolną.

Podstawową korzyścią z podzielenia bazy wiedzy na odrębne źródła jest możliwość rozwiązywania problemów dla których niezbędne są duże i bardzo duże bazy wiedzy. Zyskuje się przez to na szybkości działania systemu oraz na oszczędności pamięci operacyjnej. Rozwiązanie to ułatwia również proces utrzymywania bazy wiedzy, stanowiąc udogodnienie dla inżyniera wiedzy.

Zanim wiedza ekspercka zostanie wprowadzona do bazy wiedzy, musi być wcześniej odpowiednio sformalizowana (skodyfikowana). Do tego celu w systemie PC-SHELL wykorzystuje się tzw. język opisu bazy wiedzy. Wyrażony w tym języku opis bazy wiedzy może być utworzony i zapisany w pamięci dyskowej komputera za pomocą dowolnego edytora tekstowego pracującego w kodzie ASCII. Zadaniem translatora języka opisu bazy wiedzy jest czytanie pliku dyskowego zawierającego opis bazy wiedzy, tłumaczenie i wprowadzenie odpowiedniego kodu do bazy systemu PC-SHELL, mieszczącej się w całości w pamięci operacyjnej komputera.

Moduł wnioskowania

W odróżnieniu od konwencjonalnych systemów, systemy ekspertowe nie zawierają jawnego opisu sposobu rozwiązania danego problemu (algorytmu). To system ekspertowy, a ściślej jego część zwana modułem wnioskującym (ang. inference engine) rozwiązuje problem, wykorzystując wiedzę deklaratywną z bazy wiedzy. Moduł wnioskowania realizowany jest najczęściej w oparciu o zasady logiki formalnej.

Baza wiedzy stanowi tylko podstawę do wnioskowania w systemie ekspertowym. Wnioskowaniem zajmuje się moduł wnioskowania. Zadaniem modułu wnioskowania jest znalezienie przesłanek potwierdzających postawioną hipotezę. Moduł wnioskowania systemu PC-SHELL wykorzystuje wnioskowanie wstecz (ang. backward chaining). System zapewnia dwa tryby konsultacji:

- konwersacyjny,
- programowy, sterowany programem zawartym w bloku „control” bazy wiedzy.

Moduł wyjaśnień

Unikatową cechą systemów ekspertowych jest ich zdolność do automatycznego generowania różnego rodzaju objaśnień w trakcie konsultacji z użytkownikiem, w szczególności zaś wyjaśnień dotyczących sposobu rozwiązania problemu. Wyjaśnienia tworzone są przez tzw. moduł wyjaśniający (ang. explanation facility). System PC-SHELL dostarcza wyjaśnień:

- Jak? (ang. how) - odpowiadające na pytanie "W jaki sposób system wyprowadził dany zbiór konkluzji"; wyjaśnienia mają w tym przypadku charakter retrospektywny i pokazują logiczny wywód systemu.
- Dlaczego? (ang. why) - ten rodzaj wyjaśnień odpowiada na pytanie "dlaczego system zadał użytkownikowi dane pytanie". Wyjaśnienia te uzasadniają celowość pytania, poprzez pokazanie bieżącego kontekstu wnioskowania oraz tego jak odpowiedź na dane pytanie przyczyni się do rozwiązania problemu.
- Co to jest? (ang. what is) - wyjaśnienia te są tekstami objaśniającymi pojęcia zawarte w bazie wiedzy.

Wyjaśnienia są tak ważnym i specyficznym elementem technologii systemów ekspertowych, że można zaryzykować stwierdzenie, że system który jest ich pozbawiony nie jest systemem ekspertowym. Użytkownicy systemów ekspertowych często są zainteresowani sposobem rozwiązania problemu postawionego systemowi. Zdolność systemu ekspertowego do wyjaśnień, w tym zwłaszcza konkluzji i potwierdzanych hipotez, jest jedną z

najważniejszych cech tej klasy systemów, odróżniającą tę technologię od konwencjonalnych programów.

Wyjaśnienia udzielane przez system powinny przekonać użytkownika do poprawności zastosowanego rozumowania. Jednakże mogą one być podstawową do weryfikacji bazy wiedzy, jeśli użytkownik stwierdzi, że wyjaśnienia przedstawione przez system są nieprzekonujące. Wydaje się, że wyjaśnienia można również traktować jako narzędzie uruchomieniowe, ułatwiające lokalizowanie nieprawidłowości w bazie wiedzy, zwłaszcza w początkowych etapach jej tworzenia.

System PC-SHELL dostarcza retrospektywnych wyjaśnień (typu jak) rozumowania, które umożliwiło wyprowadzenie danego zbioru konkluzji lub potwierdzenie postawionej hipotezy. W czasie konsultacji system ekspertowy często pyta o obecność określonych symptomów (faktów). W takich sytuacjach użytkownik może mieć wątpliwości, czy zadane mu pytanie ma związek (lub jakiego rodzaju jest to związek) z rozwiązywanym problemem. W tym celu system PC-SHELL dostarcza wyjaśnień typu dlaczego (ang. "Why" explanations). W ramach tego typu wyjaśnień system pokazuje użytkownikowi jaka hipoteza jest rozważana oraz w jaki sposób odpowiedź na pytanie systemu dostarczy informacji niezbędnej do potwierdzenia bądź odrzucenia tej hipotezy. PC-SHELL dostarcza również wyjaśnień w formie tzw. metafor, będących objaśnieniami do reguł prezentowanych podczas wyjaśnień typu "How". Ponadto dostępne są również wyjaśnienia typu co to jest, będące tekstowymi objaśnieniami pojęć zawartych w bazie wiedzy. Należy podkreślić, że system PC-SHELL umożliwia kontrolowanie zakresu oraz głębokości wyjaśnień.

2.1.1. Język reprezentacji wiedzy

Język opisu bazy wiedzy systemu PC-SHELL służy do formalnego opisu wiedzy eksperckiej z określonej dziedziny. Językami tego typu posługują się przede wszystkim inżynierowie wiedzy. Ich podstawowym zadaniem jest pozyskanie wiedzy od specjalisty, stosując różne techniki wypracowane przez teorię i praktykę dziedziny systemów ekspertowych i zakodowanie jej za pomocą jakiegoś języka formalnego. Wydaje się, że prezentowany język opisu bazy wiedzy jest dostatecznie przejrzysty, by mógł być wykorzystywany również przez niespecjalistów.

System PC-SHELL jest systemem regułowym, stąd całość wiedzy o charakterze heurystycznym jest kodowana za pomocą reguł i faktów. Podstawową strukturą reprezentacji wiedzy jest tu trójka obiekt-atrybut-wartość (trójka OAW). Identyfikatory obiektów i atrybutów są symbolami rozpoczynającymi się od małej litery, po której może nastąpić dowolny ciąg znaków alfanumerycznych (liter i cyfr) oraz znaków `_`. Wartości atrybutów mogą być liczbami typu rzeczywistego, łańcuchami znakowymi lub reprezentowane przez zmienne. Liczby mogą być poprzedzone znakiem i zawierać kropkę dziesiętną (zakres wartości: od $3,4 \cdot 10^{-38}$ do $3,4 \cdot 10^{38}$). Łańcuchy znakowe są dowolnymi ciągami znaków zawartymi pomiędzy znakami cudzysłowu. W systemie wykorzystywane są również symbole, będące tu ciągami znaków, rozpoczynającymi się od małej litery, po której może nastąpić dowolny ciąg znaków złożony z liter i cyfr oraz znaku `_`. Nazwy zmiennych zbudowane są podobnie jak symbole, z tą różnicą, że muszą rozpoczynać się od dużej litery. Trójka obiekt-atrybut-wartość, w pełnej postaci, ma następującą składnię:

atrybut (obiekt) operator_relacji liczba lub
atrybut (obiekt) = łańcuch_znakowy lub
atrybut (obiekt) = zmienna

Identyfikator atrybutu (dla uproszczenia atrybut) jest jedynym obowiązkowym elementem trójki. Zarówno obiekt jak i wartość mogą być opuszczone. Wartość jest poprzedzona jednym z następujących operatorów relacji: "=", "<", ">", "<=", ">=" . W przypadku pojawienia się zmiennej jedynym dozwolonym operatorem relacji jest znak "=" .

Specjalnego wyjaśnienia wymaga rola operatorów zawierających znak =. System PC-SHELL wykorzystuje trzy postacie operatorów kwalifikowanych do tej kategorii:

1. =,
2. ==,
3. :=.

Ad. 1. Ten operator określa relację przyporządkowania określonej wartości do atrybutu: atrybut = wartość. Jest jednocześnie separatorem oddzielającym identyfikator atrybutu od jego wartości. Operator = wykorzystywany jest głównie w bloku reguł i faktów.

Przykład:

temperatura = 36.6
sytuacja_finansowa = "bardzo dobra"

Ad. 2. Dwuznak == oznacza test relacji równości, sprawdzając, czy wartości po lewej i prawej stronie tego operatora są takie same. Typowym kontekstem użycia są instrukcje if, while i for w bloku control bazy wiedzy.

Przykład:

```
if ( X == 10 )
    begin
        ciąg_instrukcji
    end;
while( Y==10 ) begin ciąg_instrukcji end;
```

Ad. 3. Dwuznak := jest operatorem przypisania wartości do zmiennych. Typowym kontekstem użycia jest instrukcja przypisania w bloku control.

Przykład:

```
//deklaracja zmiennej typu znakowego
char Zmienna2;
//deklaracje zmiennych o wartościach rzeczywistych
float Zmienna1, Zmienna3;
Zmienna1 := 123.45;
Zmienna2 := "łańcuch znaków";
Zmienna1 := Zmienna3;
```

Komentarze

W opisie bazy wiedzy można umieszczać komentarze. Tekst komentarza musi rozpoczynać się od dwuznaku '/' i zawierać w całości w jednym wierszu (zob. wcześniejszy przykład). Nie wolno używać komentarzy wewnątrz instrukcji języka programowania.

2.1.2. Struktura języka opisu bazy wiedzy

Opis bazy wiedzy w systemie PC-SHELL jest podzielony na pięć bloków:

1. blok opisu plików,
2. blok opisu faset,
3. blok opisu reguł,
4. blok opisu faktów,
5. blok sterowania.

Obowiązkowe jest wystąpienie przynajmniej jednego bloku - faktów lub reguł. Ogólną strukturę opisu bazy wiedzy przedstawiono na poniższym schemacie syntaktycznym. Nazwa_bazy_wiedzy jest dowolnym symbolem.

```
knowledge base nazwa_bazy_wiedzy
    sources
        opis_źródeł
    end;
    facets
        opis_faset
    end;
    rules
        opis_reguł
    end;
    facts
        opis_faktów
    end;
    control
        program
    end;
end;
```

Jak już wspomniano, system PC-SHELL ma możliwość korzystania z wielu źródeł wiedzy. Strukturę opisu źródła wiedzy przedstawia poniższy schemat syntaktyczny :

```
knowledge source nazwa_źródła_wiedzy
    facets
        opis_faset
    end;
    rules
        opis_reguł
    end;
    facts
        opis_faktów
    end;
end;
```

Jeśli system korzysta z architektury tablicowej (źródeł wiedzy w postaci eksperckich baz wiedzy), to musi wystąpić moduł główny bazy wiedzy (knowledge base). W takim przypadku moduł główny nie może zawierać opisu reguł i faktów. Całość wiedzy eksperckiej musi być umieszczona w źródłach. Moduł główny pełni wtedy rolę sterującą (blok control) i kontrolną. Z wymienionych bloków, musi wystąpić przynajmniej jeden: opis reguł lub faktów.

2.1.3. Deklaracje źródeł wiedzy

System PC-SHELL jest systemem hybrydowym o architekturze tablicowej. Oznacza to, że do rozwiązywania problemów może wykorzystywać wiele heterogenicznych źródeł wiedzy. W obecnej wersji systemu mogą to być: eksperckie bazy wiedzy, aplikacje oparte o sieci neuronowe oraz bazy danych z wyjaśnieniami tekstowymi. Schemat syntaktyczny deklaracji źródła wiedzy jest następujący:

```
sources
    opis_źródeł_wiedzy
end;
```

Opis_źródeł_wiedzy musi zawierać co najmniej z jeden opis, składający się z nazwy źródła oraz specyfikacji właściwości źródła. Nazwa źródła jest dowolną nazwą ustaloną przez inżyniera wiedzy, natomiast specyfikacje składają się z wyrażen zawierających słowa kluczowe systemu. Po nazwie źródła musi wystąpić znak `:`.

```
nazwa_źródła :
    właściwość1...właściwośćn ;
```

W obecnej wersji dostępne są następujące typy właściwości źródła: `type`, `file`. Pełny format tych opisu wymienionych właściwości pokazano na schemacie poniżej:

```
type kb | neural_net | metaphor | what_is
file łańcuch_znaków
```

Wyrażenie `type` służy do zadeklarowania typu źródła :

- `kb` - eksperckie bazy wiedzy,
- `neural_net` - sieci neuronowe,
- `metaphor` - bazy danych zawierające wyjaśnienia typu metafory,
- `what_is` - bazy danych zawierające wyjaśnienia typu co to jest.

Omówienie sposobów deklaracji źródeł powyższych typów znajduje się w następujących rozdziałach opracowania.

Wyrażenie `file` określa plik, w którym przechowywane jest źródło wiedzy. Łańcuch_znaków występujący po słowie `file` określa nazwę pliku i, jeśli to konieczne, ścieżkę dostępu. Należy dodać, że znaki `\` w ścieżce muszą być podwajane.

2.1.3.1. Źródła typu kb

Typowym źródłem w architekturze tablicowej jest baza wiedzy. W obecnej wersji PC-SHELL umożliwia wykorzystanie w jednej aplikacji do dziesięciu różnych baz wiedzy, ujętych w formie źródeł wiedzy. Formalną strukturę opisu źródła wiedzy przedstawiono w rozdziale dotyczącym deklaracji.

Nie jest dozwolone zawarcie w źródle wiedzy opisów plików, faset oraz bloku sterowania (programu). Moduł główny bazy wiedzy (oznaczony wyrażeniem knowledge base) nie może zawierać reguł i faktów. Natomiast wszystkie atrybuty używane w źródłach wiedzy muszą być zadeklarowane w bloku faset tego modułu.

Odwołanie do źródeł może nastąpić w programie (blok control) za pomocą instrukcji getSource i freeSource oraz solve. Nie jest możliwe ładowanie i uruchamianie źródeł jako samodzielnych baz wiedzy. Dozwolone jest natomiast wykorzystywanie tego samego źródła wiedzy przez wiele różnych aplikacji, które wykorzystują go do rozwiązania tego samego podproblemu.

Jeśli wszystkie źródła wiedzy typu kb znajdują się w tym samym katalogu i jest to bieżący katalog baz wiedzy, zadeklarowany w opcji Opcje | Katalogi, to wyrażenie file nie musi zawierać ścieżek do poszczególnych źródeł. W przeciwnym wypadku, tzn. gdy źródła znajdują się w innych katalogach niż bieżący katalog baz wiedzy, to nazwę pliku przechowującego źródło należy poprzedzić ścieżką do katalogu, w którym się dane źródło znajduje.

Przykład:

```
// Dotyczy teoretycznej aplikacji dla decyzji kredytowych
// Zakłada się, że ostateczna decyzja podejmowana jest po
// ustaleniu takich ocen jak: sytuacja finansowa klienta,
// zabezpieczenie kredytu, rentowność
sources
  decyzja_kredytowa :
    type kb
    file "c:\\aitech\\bw\\decyzja.zw" ;
  sytuacja_finansowa :
    type kb
    file "c:\\aitech\\bw\\sytfina.zw" ;

  zabezpieczenie :
    type kb
    file "c:\\aitech\\bw\\gwaranc.zw" ;
  rentownosc :
    type kb
    file "c:\\aitech\\bw\\rentwnsc.zw" ;
end;
```

2.1.3.2. Źródła typu neural_net

Podobnie jak w przypadku innych deklaracji źródeł wiedzy, deklaracje sieci neuronowych składają się ze zbioru oddzielnych deklaracji sieci neuronowych. Maksymalna liczba deklaracji wynosi 10.

Właściwość file odnosi się do pliku zawierającego definicję sieci neuronowej, utworzonego za pomocą systemu Neuronix. Pliki tego typu mają standardowo rozszerzenie "NPR". Pliki definiujące sieć zawierają informacje niezbędne do tego, by PC-SHELL mógł wygenerować odpowiedni symulator sieci neuronowej.

Wygenerowanie odpowiedniej sieci neuronowej następuje dynamicznie w trakcie pracy systemu, na podstawie informacji zawartych w pliku wskazanym przez wyrażenie file.

Przykład:

```
sources
-
  prognoza_finansowa :
    type neural_net
    file "c:\\aitech\\sieci\\prognoza.npr" ;
end;
```

2.1.3.3. Źródła typu metaphor i what_is

Deklaracje metaphor i what_is określają źródła w formie baz wyjaśnień tekstowych, utworzonych za pomocą programu dbMaker lub systemu CAKE. Pliki tego typu wykorzystywane są podczas wyjaśnień typu "What is" oraz objaśnień konkluzji. Deklaracja metaphor definiuje bazę metafor, również utworzoną za pomocą programu dbMaker lub systemu CAKE. Metafory mogą być wykorzystywane podczas wyjaśnień typu "How".

W obecnej wersji systemu PC-SHELL dozwolone jest zadeklarowanie co najwyżej jednego źródła typu metaphor i co najwyżej jednego źródła typu what_is.

Przykład:

```
sources
-
-
  metafory :
    type metaphor
    file "c:\\aitech\\bazy\\kredyt.dbm" ;
  coto :
    type what_is
    file "c:\\aitech\\bazy\\kredyt.dbw" ;
end;
```

Okna wyjaśnień typu Co to? (What is) są to tekstowe wyjaśnienia przygotowywane przez inżyniera wiedzy za pomocą programu dbMaker, objaśniające wybrane pojęcia z bazy wiedzy, użyte w konkluzji lub zapytaniu systemu. Dla przykładu, w systemie diagnostycznym, który ustalił konkretną usterkę, wyjaśnienia te mogą poza omówieniem samej usterki, określić sposób jej usunięcia.

Wyjaśnienia typu Jak? (How) są dostępne w oknie rozwiązań po zakończeniu procesu wnioskowania. Służą one udokumentowaniu i przedstawieniu użytkownikowi w jaki sposób system wyprowadził dany zbiór konkluzji (rozwiązań). Wyjaśnienia mają w tym wypadku charakter retrospektywny. W przypadku określenia przez inżyniera wiedzy w programie dbMaker dostępne są wyjaśnienia typu metafory dla wybranych reguł. Wyświetlenie metafory nastąpi po dwukrotnym naciśnięciu na regule lewego przycisku myszy lub po wybraniu reguły i naciśnięciu przycisku metafora.

Wyróżnienie faktu powoduje wyświetlenie małego okienka z dodatkowymi wyjaśnieniami na temat sposobu i źródła pozyskania faktu. Jeżeli dany fakt powstał w wyniku wnioskowania ze źródła wiedzy (użycie architektury tablicowej) wtedy dostępne są, po podwójnym naciśnięciu lewym klawiszem myszki, dalsze wyjaśnienia typu 'Jak ?'. Otwiera się wtedy identyczne okno wyjaśnień 'Jak ?' dla wybranego faktu

2.1.4. Blok faset

Fasetami określa się tu zbiór deklaracji odnoszących się do wybranych atrybutów. Blok faset zawiera wykaz wszystkich atrybutów używanych w bazie wiedzy, w tym również zawartych w źródłach wiedzy, wraz z przypisanymi do nich fasetami. Nie wszystkie atrybuty muszą być opisane fasetami, lecz wszystkie muszą być zadeklarowane w bloku faset.

Ogólną strukturę bloku opisu faset przedstawiono poniżej:

```
facets
  [ ask { yes | no } ]

  [ single { yes | no } ]

  atrybut1 [ deklaracje_faset1 ];
  -
  atrybutn [ deklaracje_fasetn ];
end;
```

Opis faset składa się z deklaracji globalnych ask, single oraz zbioru atrybutów i związanych z nimi faset. W systemie PC-SHELL dostępne są następujące rodzaje faset: ask, unit, val (values), single, query, param, picture, oraz faseta video. Spośród wymienionych, musi wystąpić przynajmniej jedna; kolejność deklaracji jest dowolna.

2.1.4.1. Fasety ask i single

Deklaracja ask określa, czy system może zadawać pytania o prawdziwość warunków reguł. Jeśli wybrana zostaje opcja "yes", to system będzie zadawał pytania o prawdziwość warunków, które nie mogą być potwierdzone w oparciu o wiedzę zawartą w bazie wiedzy w postaci faktów i reguł. Opcja "no" oznacza, że system nie może zadawać pytań użytkownikowi w celu potwierdzenia prawdziwości warunków. Może jedynie wykorzystywać wiedzę zawartą w bazie wiedzy. Od tych ogólnych reguł można określić wyjątki, stosując deklarację ask w deklaracji atrybutu. System przyjmuje domyślnie ask yes.

Zastosowanie fasety ask do definiowania atrybutu, określa czy system może stawiać pytania dotyczące danego atrybutu. Deklaracja umożliwia tworzenie wyjątków od globalnej deklaracji ask, która dotyczy wszystkich atrybutów w bazie wiedzy. Należy podkreślić, że system PC-SHELL zadaje pytania jedynie w sytuacji, gdy nie potrafi potwierdzić warunku reguły lub hipotezy wykorzystując fakty i reguły zawarte w bazie wiedzy.

Faseta single umożliwia zadeklarowanie, że w bazie wiedzy może wystąpić tylko jeden fakt zawierający atrybut, do którego odnosi się faset single. Ma to najczęściej miejsce w odniesieniu do atrybutów, których wartości wzajemnie się wykluczają. Dla przykładu, jeśli w bazie wiedzy występuje pewien fakt, np.: "temperatura_ciała=36.6" i atrybut "temperatura_ciała" opisano fasetą "single yes", to system przyjmie, że w bazie wiedzy nie ma innego faktu stwierdzającego np. że "temperatura_ciała=37.5". Jednakże, gdyby taki fakt się pojawił jako drugi w kolejności, to system będzie go ignorował.

W praktyce zastosowanie fasety single umożliwia zredukowanie liczby pytań stawianych przez system o wartość danego atrybutu, przez co jego działanie staje się bardziej "inteligentne".

Faseta single może pojawić się - w postaci deklaracji globalnej, przed opisami faset poszczególnych atrybutów (podobnie jak ask) i oznacza wtedy, że działaniem fasety single objęte są praktycznie wszystkie atrybuty używane w danej bazie wiedzy. Faseta single może być również użyta w odniesieniu do wybranych atrybutów. W ten sposób można definiować wyjątki, podobnie jak w przypadku fasety ask. Domyślnie przyjmuje się wartość "single no".

Przykład:

```
facets
    ask yes;
        single yes;

    atrybut_1:
        ask no
        single no;
    atrybut_2:
        single yes;
end;
```

2.1.4.2. Faseta unit

Faseta unit umożliwia zadeklarowanie jednostki miary, w której wyrażane są wartości danego atrybutu. Jednostka_miary jest w tym wypadku dowolnym tekstem, a ściślej łańcuchem znaków. Podczas wyświetlania informacji zawierającej dany atrybut (np. zapytania systemu, przeglądanie bazy wiedzy), dodatkowo będzie pojawiał się tekst zadeklarowany jako jednostka_miary.

Przykład:

```
facets
    wielkość_obrotów :
        unit "mln zł"
        ask yes
        query "Podaj wielkość obrotów:" ;
end;
```

2.1.4.3. Faseta val

Określa zbiór dopuszczalnych wartości danego atrybutu. Wartości mogą być liczbami rzeczywistymi (typ float) lub łańcuchami znakowymi. W obecnej wersji systemu do określenia dozwolonych lub niedozwolonych wartości służą następujące deklaracje związane z faseta val: oneof, someof, range, except. W przypadku, gdy wprowadzona do systemu wartość (np. w formie odpowiedzi użytkownika, dynamicznie dodawanego faktu, za pomocą instrukcji freadFacts) wykracza poza dopuszczalny zakres system automatycznie sygnalizuje błąd i nie pozwala na wprowadzenie takiego faktu do bazy wiedzy.

Ubočnym (zamierzonym) efektem działania faset oneof i someof może być automatycznie generowane okno zawierające zadeklarowane w fasetce wartości.

oneof

```
val oneof { wartość_1,..., wartość_n }
```

Faseta val oneof deklaruje dozwolony zbiór wartości atrybutu, z którym ta faseta jest związana. Wartości wartość_1,..., wartość_n mogą być wyłącznie liczbami lub łańcuchami znakowymi. Nie mogą w liście wartości wystąpić mieszane typy: liczby i łańcuchy znakowe.

Użycie tej fasety zakłada domyślnie wartość yes dla fasety single. Dlatego próba wprowadzenia kolejno dowolnych dwóch wartości z listy do bazy wiedzy zakończy się niepowodzeniem.

Zadeklarowany w tej fasetce zbiór wartości pojawia się w formie podpowiedzi, mających postać automatycznie generowanego okienka zawierającego zadeklarowaną listę wartości.

Użytkownik wybiera wartość z okna zgodnie z zasadami obowiązującymi w systemie Windows. Wybrana wartość zostaje automatycznie przypisana zmiennej (o ile występuje) lub uzgodniona z wartością stałą w aktywnym warunku reguły. W rezultacie fakt zawierający wybraną wartość jest wprowadzany do bazy wiedzy.

Przykład:

```
facets
// zapis poprawny
stan_wody :
    query "Określ stan wody w rzece:" ;
    val oneof { "niski", "średni", "wysoki" };
liczba_pomiarów :
    query "Podaj liczbę pomiarów"
    val oneof { 1, 2, 3, 4, 5 };
// zapis niepoprawny
temperatura_wody :
    query "Określ temperaturę wody"
    oneof { "niska", 15, "średnia", 16 };
end;
```

someof

```
val someof { wartość_1,..., wartość_n }
```

Podobnie jak faseta oneof, deklaracja someof określa zbiór dopuszczalnych wartości danego atrybutu. Różnica polega na tym, że faseta someof umożliwia wybranie kilku wartości z listy i umieszczenie ich w formie faktów w bazie wiedzy. Dlatego faseta ta zakłada użycie fasety single no dla danego atrybutu lub - o ile taka deklaracja nie wystąpiła - niejawnie ustala taką wartość tej fasety. Jawne użycie fasety single no łącznie z fasetą someof traktowane jest jako błąd.

Zastosowanie tej fasety może być odpowiednie dla tych atrybutów, które mają wartości niewykluczające się wzajemnie. Podobnie jak faseta oneof, faseta someof generuje podczas dialogu z użytkownikiem okno zawierające listę dopuszczalnych wartości. W tym przypadku jednak, użytkownik może wybrać jednocześnie kilka spośród nich.

range

```
val range przedział  
val range { przedział_1,..., przedział_n }
```

gdzie:

przedział jest przedziałem otwartym, oznaczany znakami: (), lub przedziałem domkniętym oznaczanym znakami: < >.

Faseta val range umożliwia deklarowanie wartości atrybutu w formie zbioru dopuszczalnych przedziałów. Wartości w tym przypadku mogą być wyłącznie liczbami. Nie jest dozwolone jednoczesne użycie którejs z wymienionych faset: oneof, someof, except. Dla oznaczenia liczbowej wartości minimalnej oraz maksymalnej w danej implementacji można używać odpowiednio symboli MIN oraz MAX.

Przykład:

```
facets  
  temperatura_ciała :  
    val range < 36, 42 >;  
  parametr_X:  
    val range { < -5, -1 >, < 1, 5 > };  
  parametr_Y:  
    val range { < MIN, 0 ), ( 0, MAX > };  
end;
```

except

```
val except przedział  
val except { przedział_1,..., przedział_n }
```

gdzie:

przedział jest przedziałem otwartym, oznaczanym znakami: (), lub przedziałem domkniętym oznaczanym znakami: < >.

Faseta ta określa zbiór dopuszczalnych wartości związanego z nią atrybutu, przez wyszczególnienie wartości niedozwolonych (w pewnym sensie odwrotnie do fasety range). Nie jest dozwolone jednocześnie użycie którejs z wymienionych faset: oneof, someof, range. Podobnie jak w przypadku fasety range, dla oznaczenia liczbowej wartości minimalnej oraz maksymalnej w danej implementacji, można używać odpowiednio symboli MIN oraz MAX.

Przykład:

```
facets  
  temperatura_ciała :  
    val except { < MIN, 36), ( 42, MAX > };  
  parametr_X:  
    val except { < MIN,-5 ),(-1, 1 ),( MAX,5 > };  
  parametr_Y:  
    val except { 0 };  
end;
```

2.1.4.4. Faseta query

```
query treść_zapytania | { lista_zapytań }
```

Jak już wspomniano system PC-SHELL generuje automatycznie pytania o prawdziwość określonych warunków aktywnych reguł lub o wartości atrybutów występujących w tych warunkach. Treść zapytań jest formułowana automatycznie, w sposób jednolity dla całej bazy wiedzy. Faseta query umożliwia zdefiniowanie przez użytkownika treści tych zapytań, w odniesieniu do wybranych atrybutów lub w odniesieniu do wybranych wartości atrybutu (druga postać definicji). W przypadku użycia tej fasety, system zadaje pytanie o treści zgodnej z określoną w faszecie, zamiast pytań automatycznie generowanych przez system.

Wystąpienie listy zapytań musi być powiązane z zadeklarowaniem listy wartości w postaci fasety val oneof. W takim przypadku każdej wartości musi odpowiadać jedno zapytanie na liście. Postać rozbudowana zapytań powoduje także inne zachowanie się okna konsultacji. Standardowo pojawienie się fasety val oneof powoduje, że w oknie konsultacji pojawia się lista tych wartości, natomiast w sytuacji zdefiniowania zapytań dla każdej wartości z osobna na liście możliwych odpowiedzi wystąpi lista możliwych odpowiedzi w postaci "tak / nie / nie wiem".

Przykład:

```
facets
  wielkość_obrotów :
    ask yes
    query "Podaj wielkość obrotów:" ;
  przychód :
    val oneof { "niski", "wysoki" }
    query { "Czy wartość przychodu jest niska ?",
            "Czy wartość przychodu jest wysoka ?" };

end;
```

2.1.4.5. Faseta param

```
param { zmienna1 = wartość1,..., zmiennaN = wartośćN }
```

Faseta param umożliwia zadeklarowanie tzw. zmiennych parametrycznych i przypisanie im wartości domyślnych. Zmienne takie mogą pojawiać się w bazie wiedzy np. w charakterze wartości progowych lub przedziałów wartości, z którymi porównywane są rzeczywiste wartości danego atrybutu. Dzięki temu mechanizmowi znacznie ułatwiona jest procedura parametryzacji baz wiedzy, która ma miejsce w niektórych zastosowaniach. Zmiana wartości zmiennych parametrycznych może nastąpić zarówno z poziomu programu zawartego w bloku control jak i w sposób interakcyjny za pomocą okna w opcji Narzędzia | Parametryzacja.

Wartości domyślne wartość1,..., wartośćN nie mogą być sprzeczne z deklaracjami typu: oneof, someof, range, except w fasecie val

Przykład:

```
facets
  ..
  // przykłady deklaracji poprawnych
  atrybut_1 :
    val range < 1, 3 >
    param { PARMIN = 2.1, PARMAX = 2.8 };
  atrybut_2 :
    val oneof { "alfa", "beta", "gamma" >
    param { PAR1 = "alfa", PAR2 = "beta" };

  // przykłady deklaracji niepoprawnej
  atrybut_3 :
    val range < 1, 3 >
    param { PARATR3 = -2, PARATR3 = 4 };
  atrybut_4 :
    val oneof { "alfa", "beta", "gamma" >
    param { PARATR4 = "omega", PARATR4 = "epsilon" };

end;
```

2.1.4.6. Faseta picture

picture NazwaPliku | { plik_1,..., plik_n }

Faseta picture umożliwia związanie atrybutów z rysunkami, np. w formie bitmap. W obecnej wersji rysunek jest automatycznie pokazywany, gdy pojawia się zapytanie dotyczące atrybutu, z którym związany jest rysunek oraz w przypadku pojawienia się rozwiązania do zapytania o atrybut ze zwiazanym rysunkiem. System umożliwia przypisanie rysunku bezpośrednio do atrybutu (postać pierwsza definicji) lub przypisuje każdej z wartości z osobna odrębny rysunek. Oczywiście druga postać wymaga wystąpienia w deklaracji atrybutu fasety val w postaci oneof. Ilość wartości determinuje ilość elementów listy nazw plików z rysunkami. Obecnie dopuszczane są jedynie rysunki w formacie bmp.

Przykład :

facets

obiekt :

```
val oneof { "p1", "m1", "m2" }  
picture { "p1.bmp", "m1.bmp", "m2.bmp" };
```

2.1.4.7.Faseta sound

sound NazwaPliku | { plik_1,..., plik_n }

Faseta sound umożliwia związanie atrybutów z dźwiękami (mową, muzyką itp.). Deklaracja jest identyczna jak fasety picture. Akceptowany format plików dźwiękowych to pliku typu wave (rozszerzenie .wav).

2.1.4.8. Faseta video

video NazwaPliku | { plik_1,..., plik_n }

Faseta video umożliwia związanie atrybutów z animacjami, np. zawartych w plikach w formacie avi lub mpg. System umożliwia przypisanie animacji bezpośrednio do atrybutu (postać pierwsza definicji) lub przypisuje każdej z wartości z osobna. Oczywiście druga postać wymaga wystąpienia w deklaracji atrybutu fasety val w postaci oneof. Ilość wartości determinuje ilość elementów listy nazw animacji.

Typy animacji zależne są od zainstalowanych standardowych sterowników odtwarzania tego typu plików. Wraz z systemem Sphinx dostarczana jest biblioteka Video for Windows umożliwiająca zainstalowanie sterowników do animacji typu .avi dla Windows 3.1 lub 3.11. Windows 95 posiada standardowo biblioteki do obsługi avi stąd odpada konieczność instalacji oprogramowania.

2.1.5. Blok opisu faktów

Blok opisu faktów umożliwia zapisanie wiedzy o charakterze faktograficznym, będącej zbiorem faktów na jakiś temat. Najczęściej będą to informacje względnie stałe, o charakterze parametrów dla bazy wiedzy. W praktyce ten blok nie musi wystąpić.

Opis faktów składa się ze zbioru faktów poprzedzonych słowem facts oraz zakończonych słowem end :

```
facts
      opis_faktów
end;
```

Podstawowym elementem faktów jest trójka obiekt - atrybut - wartość (trójka OAW), która może być poprzedzona znakiem negacji. W systemie PC-SHELL negacja jest oznaczona słowem not. W odróżnieniu od ogólnej składni trójki OAW fakty nie mogą zawierać zmiennych, a jedynym operatorem pomiędzy atrybutem i wartością jest znak =. Ilustrację dopuszczalnej składni faktów, bez zaznaczenia wymienionych ograniczeń, pokazano poniżej. Ilustracje poprawnie zbudowanych faktów zawarto w przykładzie :

Składnia faktów :

```
trójka_OAW;      lub
not trójka_OAW;
```

Przykład :

```
facts
      miesięczny_obrót_mld (firma_X) = 14.5;
      ocena_zdolności_kredytowej (klient 1) = "dobra";
      nie ma bóle mięśni;
      napięcie_progowe_MOSTE = 2.31;
      ocena_zabezpieczeń_kredytowych (firma_Y) = "bardzo dobra";
      rodzaj_zabezpieczeń = "papiery wartościowe";
      rodzaj_zabezpieczeń = "krajowe depozyty gotówkowe";
      rodzaj_zabezpieczeń = "hipoteka";
end;
```

Podczas ładowania bazy wiedzy fakty umieszczane są na początku (przed regułami). W procesie wnioskowania fakty pobierane są do uzgodnienia w kolejności zgodnej z ich pozycją w bazie wiedzy (dotyczy to również reguł).

2.1.6. Blok opisu reguł

Blok reguł pełni główną rolę z punktu widzenia reprezentacji wiedzy eksperckiej. Formalizm reguł jest dziedzinowo-niezależny i umożliwia kodowanie wiedzy praktycznie z każdej dziedziny.

Opis reguł składa się ze zbioru reguł poprzedzonych słowem `rules` oraz zakończonych słowem `end` :

```
rules
    opis_reguł
end;
```

Standardowo składnia reguł składa się z konkluzji, oraz części warunkowej. Konkluzja oraz część warunkowa oddzielone są słowem kluczowym `if`. Część warunkowa musi zawierać przynajmniej jeden warunek. Reguły systemu PC-SHELL dzielimy na proste i złożone. Reguły proste są odzwierciedleniem reguł Hornea czyli schematu:

konkluzja `if` warunek1 i warunek2 i ... i warunekN

Kolejne warunki oddzielone są od siebie przecinkiem `,` lub znakiem `&`.

W odróżnieniu od reguł prostych, reguły złożone zawierają warunki lub grupy reguł alternatywnych (operator logiczny `lub`). Warunki te oddzielone są za pomocą symbolu `|`, przy czym grupy warunków są ujmowane w nawiasy `(` oraz `)`.

```
[ numer_reguły : ] konkluzja1 if warunek_1 & warunek_2 &...& warunek_n;
[ numer_reguły : ] konkluzja2 if warunek_1 | (warunek_2 &...& warunek_n);
```

Należy zastrzec, że operacja łącznika logicznego `&` jest silniejsza od operacji alternatywy `lub` (ma wyższy priorytet).

System PC-SHELL udostępnia dwa rodzaje numeracji reguł: użytkownika (jawna) i automatyczną (niejawna). Numeracja użytkownika tworzona jest przez inżyniera wiedzy w opisie bazy wiedzy. Każda reguła powinna otrzymać numer, będący jej jednoznacznym identyfikatorem w obrębie całej bazy wiedzy, uwzględniając również ewentualne źródła wiedzy. Numery reguł muszą być liczbami z przedziału 0-9999. System zakłada, że jeśli pierwsza w kolejności reguła ma numer jawny, to pozostałe reguły muszą mieć również przypisane numery. I odwrotnie, jeśli pierwsza reguła nie ma numeru, to żadna z reguł w danej bazie wiedzy nie może mieć przypisanego przez użytkownika numeru. Złamanie którejś z tych zasad spowoduje błąd w czasie translacji bazy wiedzy.

Jeśli inżynier wiedzy nie nada jawnej numeracji regułom to system automatycznie przypisze wszystkim regułom w bazie wiedzy numery, zgodne z ich kolejnością w tekście źródłowym bazy wiedzy.

Zaleca się stosowanie jawnej numeracji. W przypadku aplikacji z bazami wiedzy ujętymi w formie źródeł wiedzy, jawna numeracja jest obowiązkowa.

Składnia konkluzji jest taka sama jak faktów, z tą różnicą, że w konkluzjach mogą pojawić się zmienne.

Część warunkowa reguł może zawierać trójki OAW, wyrażenia relacyjne oraz arytmetyczne. Warianty składni warunków reguł przedstawiono na poniższym schemacie:

trójka_OAW
not trójka_OAW
wyrażenia_relacyjne
instrukcja przypisania

Wyrażenia relacyjne zawierają dwa argumenty rozdzielone operatorami relacji: "=", "<=", ">=", "<", ">", "<>", przy czym operator "=" oznacza równość. Argumentami mogą być zmienne lub liczby. Zmienne muszą mieć wcześniej przypisaną wartość typu liczba. Jeżeli relacja jest spełniona, to wartością logiczną wyrażenia jest prawda, w przeciwnym razie wartością jest fałsz.

Instrukcja przypisania składa się ze zmiennej, dwuznaku := oraz następującego po nim wyrażenia arytmetycznego. Wyrażenie arytmetyczne może być liczbą, zmienną o przypisanej wcześniej wartości typu liczba lub wyrażeniem dwuargumentowym. W przypadku wyrażenia dwuargumentowego oba argumenty są rozdzielone dwuargumentowymi operatorami: -, +, *, /. Argumenty mogą być liczbami, zmiennymi o przypisanych wcześniej wartościach typu liczba lub funkcjami matematycznymi. Ilustrację poprawnych instrukcji przypisania zawartości pokazano na przykładzie poniżej :

X := 5.1
X1 := X2
X3 := 2 + 5
X4 := X5 - X6
X6 := 2 + sin(X7)
X8 := sin(X9) + cos(X10)

2.1.7. Blok control

Blok control służy do zdefiniowania programu systemu PC-SHELL. Program składa się ze zbioru instrukcji zawartych w bloku control. W ten sposób zachowana została zasada wyraźnego rozdzielania wiedzy eksperckiej oraz tzw. sterowania. Wydaje się, że rozwiązanie to powinno ułatwić budowę systemów ekspertowych działających w praktyce. Język programowania systemu PC-SHELL będzie doskonałony i rozwijany również w przyszłych wersjach tego systemu, z uwzględnieniem specyficznych wymagań stawianych przez system ekspertowy.

```
control
    program
end;
```

Program w systemie PC-SHELL składa się z dwóch części: deklaracji zmiennych oraz zbioru instrukcji :

```
[deklaracje zmiennych]
instukcje
```

Jeśli program nie wykorzystuje zmiennych, to deklaracje nie muszą oczywiście wystąpić. W obecnej wersji systemu dostępne są typy zmiennych int, long, float, double, char oraz tzw. zmienne systemowe. Zmienne typu int lub long mogą przyjmować wartości całkowite, zmienne typu float lub double wartości będące liczbami rzeczywistymi, natomiast zmienne typu char mogą przechowywać wartości będące znakami, symbolami lub łańcuchami znakowymi. Nazwy zmiennych muszą rozpoczynać się od dużej litery, po której może (choć nie musi) następować ciąg liter i/lub cyfr i/lub znaków "_". Ta sama nazwa nie może się pojawić w dwóch różnych deklaracjach typów.

Przykład poprawnych deklaracji zmiennych:

```
int Zmienna1;
char C1, C2, STR, Lancuch_znakowy;
float X, Y,Z, Srednia;
```

Zmienne systemowe nie są jawnie deklarowane a ich identyfikatory nie mogą się pojawić w deklaracjach zmiennych. W obecnej wersji systemu wprowadzono jedną zmienną tego typu o nazwie RETURN. Wartość tej zmiennej jest modyfikowana użyciem niektórych instrukcji programowania, pojawiając się jako "efekt uboczny" ich działania. Daje to możliwość specjalnej obsługi niektórych zdarzeń w systemie.

Typy numeryczne proste

Poza podstawowymi typami char oraz float (występującymi we wcześniejszych wersjach systemu PC-Shell), dostępne są również typy całkowite int, longint oraz rozszerzony typ zmiennoprzecinkowy double. Typy te odpowiadają typom zdefiniowanym w języku C. Dokładne dane typów numerycznych zestawiono w tabelicy poniżej.

Zakresy typów systemu PC-Shell

Typ	Ilość bajtów	Zakres
int	2	-32 768 .. 32 767
longint	4	-2 147 483 648 .. 2 147 483 648
float	4	$3.4 * 10^{-38}$.. $3.4 * 10^{38}$
double	8	$1.7 * 10^{-308}$.. $1.7 * 10^{308}$

W związku z wprowadzeniem nowych typów numerycznych, zdefiniowano następujące zasady obliczania wyrażeń arytmetycznych:

- wyrażenie numeryczne jest obliczane w zakresie takiego typu jak typ zmiennej po lewej stronie wyrażenia,
- po prawej stronie wyrażenia mogą wystąpić tylko elementy o typie zgodnym z typem wyrażenia lub typem podlegającym automatycznej konwersji,
- konwersja typów polega na automatycznym przypisaniu typu mniej dokładnego do typu bardziej dokładnego oraz na automatycznym przypisaniu typu całkowitego do typu zmiennoprzecinkowego,
- w systemie PC-SHELL znakiem oddzielającym część całkowitą od części ułamkowej (symbol dziesiętny) w liczbach zmiennoprzecinkowych jest kropka, dlatego w celu zapewnienia zgodności z innymi systemami wprowadzona jest dodatkowa instrukcja `c_ntos` służąca do konwertowania liczb na inne sposoby zapisu liczb (w szczególności polski z przecinkiem).

Przykład:

```
double D;
float F;
longint L;
int I;
...
D := F + L*I;      // wyrażenie poprawne, wykonywane na typie double
L := F + I;       // wyrażenie błędne, float nie może być przekształcony
                  // automatycznie do typu całkowitego
F := L*I + 10.4;  // wyrażenie poprawne,
F := D*1.3;      // wyrażenie błędne; double jest typem bardziej dokładnym niż
                  // float
```

Do konwersji pomiędzy typami zmiennoprzecinkowymi a całkowitymi służy instrukcja `ftoi`.

Tablice

Deklarowane zmienne mogą być zmiennymi prostymi lub zmiennymi indeksowymi (tablicami). Dozwolone jest deklarowanie i używanie tablic jedno i dwuwymiarowych. Wszystkie elementy tablicy są tego samego typu. Deklaracja tablicy określa poza typem i nazwą tablicy również jej wymiar i rozmiar. Tablice indeksowane są od 0, a nie od 1 jak np. w Fortranie lub PL/1.

Przykład deklaracji tablic:

```
float TAB1[10], TAB2[5,10];
int TAB3[2,2];
```

Typ złożony - rekordy

Rekord jest typem danych reprezentującym zdefiniowany przez użytkownika zbiór pól (zmiennych) dowolnego typu prostego. Rekord może zawierać dowolną ilość pól prostych, polami nie mogą być tablice, natomiast jest możliwe zdefiniowanie zmiennej rekordowej będącej tablicą rekordów określonego typu. Każdy rekord musi posiadać swoją unikatową nazwę określaną w momencie definiowania rekordu. Składnia definicji rekordu wygląda następująco :

```
record nazwa_rekordu
begin
    delaracja_pola1,
    ...
    delaracja_polan,
end opcjonalna_lista_zmiennych;
```

Przykład :

```
record Time
begin
    int Hour, Min, Sec;
end Czas;

record Dane_personalne
begin
    char Nazwisko, Imie;
    int Wiek;
    int Pensja;
end Tablica[10];
record Dane_personalne Rekord1;
```

Dostęp do pól rekordu uzyskuje się przy użyciu operatora '.' (kropka), nazywanego operatorem wyłuskania. Operator wyłuskania pola występuje bezpośrednio po nazwie zmiennej a przed nazwą pola. Poniżej przedstawiamy przykłady poprawnych wyłuskań pól :

```
Czas.Hour      := 12;  
Czas.Min       := 10;  
Czas.Sec       := 0;  
Czas.Min       := Czas.Min + 10;  
Tablica[0].Nazwisko := "Kowalski";  
Tablica[0].Imie   := "Jan";  
Rekord1.Nazwisko := Tablica[0].Nazwisko;
```

Zmienne parametryczne

Zmienne parametryczne są deklarowane wyłącznie w bloku faset. W bloku control można je wykorzystywać, w szczególności do zmiany bieżącej wartości, podobnie jak zwykłych zmiennych. Zasadniczym miejscem użycia zmiennych parametrycznych jest blok reguł, gdzie identyfikatory zmiennych parametrycznych muszą być poprzedzone znakiem '#'.

Funkcje użytkownika

Jednym z najważniejszych elementów współczesnych języków programowania strukturalnego są funkcje i procedury. Również język SPHINX udostępnia możliwość deklarowania funkcji przez użytkownika, które w rozumowaniu języka Pascal są procedurami. Funkcje muszą być zadeklarowane na początku bloku control przed jakąkolwiek inną instrukcją. Możliwe jest wcześniejsze zadeklarowanie nagłówek przed pełną definicją, co zapewnia możliwość wywołań rekurencyjnych i rekurencję pośrednią. Jedynymi elementami języka dopuszczalnymi przed funkcjami są deklaracje zmiennych oraz definicje rekordów, w takim przypadku są one traktowane jak zmienne globalne, dostępne również wewnątrz funkcji.

Deklarowanie funkcji

Deklarowanie funkcji polega na podaniu słowa kluczowego function po nim nazwy symbolicznej (rozpoczynającej się od małej litery) funkcji oraz ewentualnie listy argumentów, co stanowi tzw. nagłówek funkcji. Lista argumentów ujęta jest w nawiasy okrągłe '(' oraz ')', w przypadku jej braku nawiasy pomijamy. Definicja argumentu składa się z nazwy typu, który może być również typem rekordowym, następnie ewentualnego znaku referencji, potem nazwy zmiennej (z dużej litery), a później ewentualnego dwuznaku [] oznaczającego tablice deklarowanego typu. Kolejne argumenty oddzielone są od siebie znakiem przecinka.

```
function test( int X, int Y, char Tekst );  
function testTbl( int Pos[], char Tekst )  
function sqrt( double d, double &wynik );
```

Deklaracja funkcji jest zakończona znakiem średnika.

Definiowanie funkcji

Definicja funkcji rozpoczyna się od nagłówka funkcji, czyli podania jej nazwy i ewntualnie (o ile występują) listy argumentów. Następnie, wewnątrz bloku begin .. end definiujemy treść (ciało) funkcji. Należy tu zaznaczyć, że wszystkie zmienne zadeklarowane w nagłówku funkcji są lokalne, z wyjątkiem zmiennych referencyjnych. Wewnątrz każdej funkcji widoczne są zmienne globalne (zadeklarowana przed definicją funkcji w bloku głównym programu) oraz zmienna globalna RETURN służąca do przekazywania wyniku wykonywania funkcji i instrukcji. Wewnątrz ciała funkcji możemy używać dowolnych instrukcji, jak również wywołań funkcji zadeklarowanych przed nagłówkiem funkcji.

Zmienne referencyjne, wyróżniane symbolem '&', oznaczają powiązanie zmiennej podanej w momencie wywołania ze zmienną lokalną istniejącą w ciele funkcji. Jakakolwiek zmiana zawartości zmiennej referencyjnej jest uwidoczniona w zmiennej zewnętrznej. Szczególnie zalecane jest podawanie jako zmiennych referencyjnych dużych struktur tablicowych i rekordów jako, że wywołanie referencyjne jest szybsze, nie powoduje bowiem odkładania na stos dużej ilości zmiennych (argumentów).

// Przykład definicji funkcji rekurencyjnej

```
function silnia( long X, long &Result )  
begin  
  if ( X <= 1 )  
  begin  
    Result := 1;  
  end  
  else  
  begin  
    long L;  
    L := X - 1;  
    silnia( L, Result );  
    Result := Result * X;  
  end;  
end;
```

```
// Przykład wywołania funkcji silnia

function wykonajSilnie( long X )
begin
    long Wynik;
    char Łącuch;
    silnia( X, Wynik );
    sprintf( Łącuch, " %ld! = %ld", X, Wynik );
    messageBox( 0,0, "Silnia", Łącuch );
end;

// wykonanie poniższego wywołania powoduje obliczenie
// i wyświetlenie wyniku obliczenia wartości 10!

wykonajSilnie( 10 );
```

Konwersja argumentów

Typy argumentów podane w wywołaniach funkcji muszą zgadzać się z typami argumentów w definicjach i deklaracjach funkcji. Konwersje następują w przypadku podania bezpośrednio jako argumentów wywołania stałych w postaci liczb.

Tablice jako argumenty

Jeśli w definicji funkcji jako argumenty użyte są tablice, oznaczamy je przez podanie po nazwie zmiennej dwuznaku '[]', bez jawnego określenia rozmiaru tablicy. Zwiększa to znacznie elastyczność operowania na tablicach. Rzeczywisty rozmiar tablicy można uzyskać za pomocą instrukcji `arraySize`.

```
function zerujTablicę( int &Tablica[] )
begin
    int Rozmiar;
    int I1;
    arraySize( Tablica, 0, Rozmiar );

    while ( Rozmiar > 0 )
    begin
        Rozmiar := Rozmiar - 1;
        Tablica[Rozmiar] := 0;
    end;
end;
```

Zwracanie wartości przez funkcję

Funkcje definiowane przez użytkownika mogą zwracać wartości przez użycie zmiennych referencyjnych lub zmienną globalną RETURN, która zasadniczo służy do sygnalizowania prawidłowości wykonania instrukcji lub funkcji (1 prawidłowe zakończenia, 0 błąd). W funkcjach możliwe jest szybsze (i bardziej czytelne) przypisanie wartości tej zmiennej przez użycie słowa kluczowego return i podania po nim zwracanej wartości. Instrukcja ta powoduje natychmiastowe przerwanie wykonywania funkcji i przypisanie odpowiedniej wartości zmiennej RETURN.

```
function test( int X )
begin
    if ( X < 0 )
    begin
        return 0;
    end;
    // ...
    return 1;
end;

int Y;
Y := 1;
test( Y );
if ( RETURN == 1 )
begin
    messageBox( 0,0, "Wywołanie funkcji test", "prawidłowe" );
end
else
begin
    messageBox( 0,0, "Wywołanie funkcji test", "nieprawidłowe" );
end;
```

Funkcje dialogowe

Odrębnym rodzajem funkcji wykorzystywanych w systemie PC-SHELL są funkcje dialogowe. Służą one grupie instrukcji operujących na dialogach zewnętrznych i umożliwiają podpięcie pod przycisk znajdujący się w oknie dialogowym akcji, zdefiniowanej w funkcji. Instrukcja dlgBindButton przypisująca akcję przyciskowi, wymaga podania jedynie nazwy funkcji, natomiast postać jej argumentów powiązana jest z organizacją okna dialogowego i kolejnością powiązania kolejnych elementów. I tak, kolejne argumenty powinny odpowiadać kolejnym polom okna dialogowego (kolejność jest warunkowana kolejnością wywoływania instrukcji typu dlgBindXXX), typ kolejnego argumentu musi być taki sam jak typ odpowiadający polu (czyli taki jak w odpowiedniej instrukcji dlgBindXXXX). Ilość argumentów w funkcji może być mniejsza niż ilość pól (ale nie większa!). Oznacza to, że wewnątrz funkcji nie mamy dostępu do wszystkich pól.

Zainicjowanie przez użytkownika akcji - poprzez przyciśnięcie przycisku powoduje wywołanie przypisanej funkcji, gdzie kolejnym argumentom przypisywane są bieżące wartości pól dialogu. Zmiana wartości parametrów wewnątrz funkcji spowoduje po zakończeniu funkcji przepisanie nowych wartości do odpowiednich pól.

Uwaga: Pola statyczne nie posiadają odwzorowania do zmiennych, dlatego są one ignorowane w czasie wywoływania funkcji dialogowych i nie wolno ich uwzględniać na liście argumentów funkcji.

Przykład :

// Założenie:

// W bibliotece test.dll znajduje się definicja okna dialogowego

// o nazwie DIALOG_1, zawierającego pola :

// ID_NAZWISKO(101) - pole edycji na Nazwisko

// ID_IMIE(102) - pole edycji na Imię

// ID_PLECMEZ(103) - pole radiowe - Mężczyzna

// ID_PLECKOBIETA(104) - pole radiowe - Kobieta

// ID_CZYSC (200) - przycisk "Wyczyść"

// Funkcja czyści pola dialogowe po naciśnięciu przycisku

function wyczyśćPola(char &Nazwisko, char &Imie, int &Mez, int &Kob)

begin

 Nazwisko := "";

 Imie := "";

 Mez := 1; // domyślnie mężczyzna

 Kob := 0;

end;

....

int DLG, RET;

char Nazwisko, Imie;

int K,M;

dlgCreate(DLG, "test.dll", "DIALOG_1");

dlgBindEdit(DLG, 101, Nazwisko);

dlgBindEdit(DLG, 102, Imie);

dlgBindRadioButton(DLG, 103, M);

dlgBindRadioButton(DLG, 104, K);

dlgBindButton(DLG, 200, "wyczyśćPola");

Nazwisko := "Kowalski"; // domyślne wartości

Imie := "Jan";

M := 1;

K := 0;

dlgExecute(DLG, RET);

....

2.2. Komputerowy system wspomaganie inżyniera wiedzy CAKE

System CAKE (ang. Computer-Aided Knowledge Engineering), będący elementem pakietu SPHINX, został stworzony jako narzędzie wspomagające inżyniera wiedzy w procesie tworzenia i rozwijania aplikacji, uruchamianych z poziomu systemu ekspertowego PC-SHELL. Dostarczając rozbudowanych mechanizmów edycji i kontroli poprawności bazy wiedzy, system CAKE umożliwia szybkie i wygodne zaprojektowanie struktury bazy oraz zdefiniowanie wszystkich jej elementów, zabezpieczając jednocześnie przed trudnymi do wychwycenia błędami. Użytkownik zwolniony jest z konieczności dokładnego zaznajamiania się ze składnią obowiązującego języka opisu wiedzy oraz korzystania z zewnętrznych edytorów tekstu w celu stworzenia aplikacji eksperckiej. Zastosowanie systemu kontroli uprawnień uniemożliwia osobom nieupoważnionym ingerencję w kod źródłowy aplikacji.

2.2.1. Właściwości aplikacji

Okno właściwości aplikacji spełnia rolę okna głównego systemu CAKE. Umożliwia swobodne poruszanie się pomiędzy poszczególnymi składnikami aplikacji. Dostęp do żądanej informacji możliwy jest po wskazaniu jednej z "zakładek", reprezentujących konkretne elementy struktury:

- Opis - informacje ogólne na temat aplikacji
- Źródła - wykaz źródeł wiedzy przypisanych do bazy wiedzy
- Obiekty - lista obiektów zdefiniowanych w bazie wiedzy
- Fasety - zestawienie atrybutów i faset zdefiniowanych w bazie wiedzy
- Fakty - wykaz faktów zawartych w bazie wiedzy
- Reguły - wykaz reguł określonych w bazie wiedzy
- Program - tekst bloku sterującego aplikacją
- Uprawnienia - lista zarejestrowanych użytkowników aplikacji.

Informacje ogólne na temat aplikacji zawierają dane dotyczące bieżącej aplikacji:

- Nazwa aplikacji - pełna nazwa aplikacji określona przez inżyniera wiedzy
- Opis aplikacji - tekst o charakterze informacyjnym, mogący zawierać między innymi opis funkcjonalny aplikacji, dane o autorze, numer wersji (tekst może zawierać znaki odstępu, znaki interpunkcyjne, itp.)
- Nazwa pliku - nazwa oraz pełna ścieżka dostępu do pliku bazy wiedzy
- Data utworzenia - data i czas utworzenia aktualnie otwartej bazy
- Data ostatniej modyfikacji - data i czas ostatniej modyfikacji bazy
- System uprawnień ... - informacja o uaktywnieniu systemu kontroli dostępu do bazy wiedzy.

2.2.2. Źródła wiedzy

Informacje o źródłach wiedzy udostępniane są poprzez wybranie polecenia Źródła z opcji Edycja głównego menu systemu CAKE. W oknie zestawione są w postaci listy wszystkie źródła wiedzy przypisane do aktualnie otwartej bazy wiedzy. Obok typów poszczególnych źródeł (np. ekspercka baza wiedzy, definicja sieci neuronowej, itd.) podane są ich nazwy, określone przez inżyniera wiedzy na etapie tworzenia kolejnych źródeł. Aby rozszerzyć listę o nowe źródło wiedzy należy nacisnąć przycisk Dodaj, a następnie określić jego rodzaj, lokalizację oraz podać nazwę źródła. Po zakończeniu operacji lista zostanie uaktualniona. Naciśnięcie przycisku Zmień lub dwukrotne kliknięcie na nazwie wybranego źródła powoduje otwarcie okna właściwości źródła wiedzy, pozwalającego na modyfikację wszystkich elementów wchodzących w skład wskazanego źródła (np. dopisanie nowych reguł, faset, itd.). Należy pamiętać, że w przypadku źródeł typu definicja sieci neuronowej bezpośrednia edycja za pośrednictwem systemu CAKE nie jest możliwa. W celu usunięcia któregoś ze źródeł wystarczy zaznaczyć jego nazwę na liście i nacisnąć przycisk Usuń. Po potwierdzeniu zamiaru usunięcia źródła, zostanie ono wyłączone z bieżącej bazy wiedzy.

Okno właściwości eksperckiego źródła wiedzy umożliwia dostęp do wszystkich elementów zdefiniowanych w danym źródle wiedzy. Dostęp do żądanej informacji możliwy jest po wskazaniu jednej z "zakładek", reprezentujących konkretne składniki struktury źródła:

- Opis - informacje ogólne na temat źródła
- Obiekty - lista obiektów zdefiniowanych w źródle wiedzy
- Fasety - zestawienie atrybutów i faset zdefiniowanych w źródle wiedzy
- Fakty - wykaz faktów zawartych w źródle wiedzy
- Reguły - wykaz reguł określonych w źródle wiedzy.

W oknie opisu eksperckiego źródła wiedzy zamieszczone są ogólne informacje dotyczące bieżącego źródła wiedzy:

- Nazwa źródła - pełna nazwa źródła określona przez inżyniera wiedzy
- Opis źródła - tekst o charakterze informacyjnym, mogący zawierać między innymi określenie podproblemu objętego w źródle (tekst może zawierać znaki odstępu, znaki interpunkcyjne, itp.)
- Nazwa pliku - nazwa oraz pełna ścieżka dostępu do pliku źródła wiedzy
- Data utworzenia - data i czas utworzenia aktualnie otwartego źródła
- Data ostatniej modyfikacji - data i czas ostatniej modyfikacji źródła
- System uprawnień ... - informacja o uaktywnieniu systemu kontroli dostępu do bazy wiedzy.

Redefinicja obiektów, atrybutów i faset, faktów oraz reguł w źródle wiedzy, jest analogiczna do projektowania ich odpowiedników w bazie wiedzy. Stąd opis narzędzi programowych służących temu celowi został świadomie zredukowany i znajduje się w następnych rozdziałach, dotyczących bazy wiedzy.

2.2.3. Obiekty bazy wiedzy

Informacje o obiektach bazy wiedzy udostępniane są poprzez wybranie polecenia Obiekty opcji Edycja głównego menu systemu CAKE. W oknie obiektów bazy wiedzy przedstawiony jest wykaz obiektów zdefiniowanych w bieżącej bazie wiedzy. Jeżeli aplikacja nie zawiera źródeł wiedzy - na liście umieszczone zostaną nazwy wszystkich istniejących obiektów; w przeciwnym wypadku - w oknie pojawią się wyłącznie nazwy obiektów globalnych, zdefiniowanych w bazie (dostęp do pozostałych możliwy jest wyłącznie z poziomu źródeł wiedzy). Aby rozszerzyć bazę wiedzy o nowy obiekt, wystarczy nacisnąć przycisk Dodaj i, po ukazaniu się okna dialogowego, podać identyfikator-nazwę obiektu. Jeżeli zaistnieje konieczność zmiany identyfikatora - należy zaznaczyć żądany obiekt na liście i nacisnąć przycisk Zmień lub dwukrotnie kliknąć na danym identyfikatorze. Spowoduje to wyświetlenie okna, umożliwiającego modyfikację nazwy obiektu. Usunięcie obiektu z bazy wiedzy możliwe jest przez naciśnięcie przycisku Usuń - po zatwierdzeniu operacji wskazany obiekt zostanie skasowany. Należy jednak pamiętać, że usunięcie obiektu możliwe jest dopiero po usunięciu wszystkich reguł i faktów obejmujących dany obiekt (w przeciwnym wypadku próba jego usunięcia spowoduje wygenerowanie przez system ostrzeżenia).

2.2.4. Fasety atrybutów bazy wiedzy

Informacje o atrybutach i związanych z nimi fasetach bazy wiedzy udostępniane są poprzez wybranie polecenia Fasety opcji Edycja głównego menu systemu CAKE. W oknie wyszczególnione są nazwy atrybutów oraz aktualna postać faset globalnych bazy wiedzy.

Lista atrybutów

Na liście umieszczone są atrybuty, które zostały zdefiniowane w bieżącej bazie wiedzy. Jeżeli aplikacja zawiera źródła wiedzy - lista zawierać będzie wyłącznie nazwy atrybutów globalnych, zdefiniowanych w samej bazie (w tym przypadku informacje o pozostałych atrybutach dostępne są z poziomu źródeł wiedzy).

Zaznaczenie opcji Uporządkuj alfabetycznie umożliwia uszeregowanie identyfikatorów atrybutów w porządku alfabetycznym.

Aby utworzyć w bazie wiedzy nowy atrybut, należy nacisnąć przycisk Dodaj. Po określeniu typu atrybutu, na ekranie wyświetlone zostanie okno właściwości atrybutów, umożliwiające określenie dalszych szczegółów dotyczących nowo utworzonego atrybutu.

Przeglądanie oraz modyfikacja właściwości zdefiniowanych atrybutów możliwa jest po naciśnięciu przycisku Zmień lub dwukrotnym kliknięciu na nazwie wybranego atrybutu - na ekranie pojawi się okno właściwości, dające dostęp do szczegółowych danych na temat istniejących w bazie wiedzy atrybutów i pozwalające na wprowadzenie ewentualnych zmian.

W celu skasowania któregoś z atrybutów wystarczy wskazać jego identyfikator na liście i nacisnąć przycisk Usuń. Należy jednak pamiętać, że usunięcie atrybutu możliwe jest dopiero po usunięciu wszystkich reguł i faktów obejmujących dany atrybut (w przeciwnym wypadku próba jego usunięcia spowoduje wygenerowanie przez system ostrzeżenia).

Fasety globalne

Fasety globalne SINGLE i ASK określają domyślną postać tych faset dla atrybutów, którym nie zostały one jawnie przypisane. Zaznaczenie opcji SINGLE=YES jest tożsame z domyślnym ustaleniem fasety globalnej SINGLE. Zaznaczenie opcji ASK=YES jest tożsame z domyślnym ustaleniem fasety globalnej ASK. Ustawienia nie obejmują atrybutów lokalnych, zdefiniowanych w źródłach wiedzy.

Właściwości atrybutów

Okno właściwości umożliwia dostęp do szczegółowych informacji o atrybutach zdefiniowanych w bieżącej bazie lub źródle wiedzy. Dane atrybutów zebrane są w kilku grupach:

Ogólne - nazwa i typ atrybutu, wartości faset, zakres widoczności atrybutu. Wśród informacji zawartych w oknie znajdują się:

- * krótki opis typu atrybutu (umieszczony w górnej części okna)
- * nazwa atrybutu
- * definicje faset QUERY oraz UNIT (tylko w przypadku atrybutów posiadających wartość)
- * fasety ASK i SINGLE atrybutu.

Modyfikacja danych dotyczących atrybutów globalnych nie jest możliwa z poziomu źródeł wiedzy. Jeżeli aplikacja posiada źródła wiedzy, dla atrybutów o charakterze globalnym określa się dodatkowo ich "zakres widoczności", czyli listę źródeł, w których mogą występować odwołania do danego atrybutu (aktualna postać listy widoczna jest w dolnej części okna). Aby rozszerzyć listę źródeł należy nacisnąć przycisk **Dopisz źródło** do listy i, po ukazaniu się okna dialogowego, wskazać wybrane źródło wiedzy. Jeżeli atrybut ma być widoczny we wszystkich źródłach wiedzy należących do aplikacji, wystarczy nacisnąć przycisk **Dopisz wszystkie źródła**. W celu usunięcia źródła wiedzy z listy, należy wskazać jego nazwę i nacisnąć przycisk **Usuń źródło** z listy.

Wartości - lista dopuszczalnych wartości wyliczeniowych atrybutu. W oknie wyszczególnione są dopuszczalne wartości wyliczeniowe (numeryczne lub symboliczne) bieżącego atrybutu. Naciśnięcie przycisku **Uporządkuj listę wartości** umożliwia uporządkowanie wyświetlonych wartości (rosnąco lub zgodnie z porządkiem alfabetycznym). Aby dodać nową wartość atrybutu należy nacisnąć przycisk **Dodaj** - na ekranie pojawi się okno właściwości wartości, umożliwiające dokładne określenie wartości i, ewentualnie, przypisanie efektów dźwiękowych, rysunku lub sekwencji wideo. Jeżeli zachodzi potrzeba modyfikacji jednej z wartości, wystarczy wskazać wartość na liście i nacisnąć przycisk **Zmień** lub kliknąć dwukrotnie na pozycji danej wartości. Po wyświetleniu okna właściwości można przystąpić do wprowadzania wymaganych zmian. W celu usunięcia wybranej wartości należy nacisnąć przycisk **Usuń** - po potwierdzeniu zamiaru usunięcia wskazana wartość zostanie skasowana. Należy pamiętać, że wartości mogą być usuwane z bazy wiedzy pod warunkiem, że nie występują w żadnych regułach lub faktach. W przeciwnym przypadku, przy próbie usunięcia wartości system wyświetli ostrzeżenie. Dodawanie lub modyfikowanie wartości atrybutów globalnych nie jest możliwe z poziomu źródeł wiedzy.

Przedziały - przedziały dopuszczalnych wartości liczbowych atrybutu. W oknie wyszczególniono dozwolone lub niedozwolone przedziały wartości liczbowych atrybutu. Aby zdefiniować nowy przedział wartości należy nacisnąć przycisk Dodaj - na ekranie wyświetlone zostanie okno definicji przedziału, umożliwiające określenie krańców przedziału. W razie konieczności zmiany jednego z istniejących przedziałów, wystarczy zaznaczyć przedział na liście i nacisnąć przycisk Zmień lub dwukrotnie kliknąć na danym przedziale. Po pojawieniu się okna definicji można dokonać żądanych modyfikacji. Usuwanie wybranego przedziału odbywa się przez naciśnięcie przycisku Usuń. Po potwierdzeniu operacji usunięcia dany przedział zostanie skasowany. Dodawanie lub modyfikowanie przedziałów wartości atrybutów globalnych nie jest możliwe z poziomu źródeł wiedzy.

Podgląd tekstu - fragment tekstu źródłowego opisujący atrybut. W oknie podglądu wyświetlana jest definicja bieżącego atrybutu, przedstawiona w postaci fragmentu kodu źródłowego, zawierającego instrukcje języka opisu wiedzy PC-SHELL. Bezpośrednia edycja tekstu w oknie nie jest możliwa.

Grafika i dźwięk - efekty dźwiękowe, rysunki oraz sekwencje wideo przypisane do atrybutu. Każdemu zdefiniowanemu w bazie wiedzy atrybutowi można przypisać elementy o charakterze multimedialnym, takie jak dźwięk, rysunek czy sekwencja wideo, odtwarzane w trakcie procesu wnioskowania. Wybór pliku dźwiękowego, graficznego bądź wideo możliwy jest po naciśnięciu przycisku oznaczonego symbolem katalogu. Po pojawieniu się okna dialogowego należy wskazać nazwę żadanego pliku. Aby usunąć wybrany element wystarczy nacisnąć jeden z przycisków oznaczonych symbolem kosza na śmieci. Operacja dotyczy jedynie odwołania do pliku - sam plik nie jest usuwany z dysku. Przyciski znajdujące się z prawej strony umożliwiają odsłuchanie dźwięku, podgląd rysunku lub odtworzenie sekwencji wideo. Poniżej pól zawierających nazwy plików umieszczono opcje określające sposób prezentacji elementów multimedialnych:

- * Dopasuj rozmiary rysunku - zaznaczenie tej opcji powoduje automatyczne skalowanie rysunku tak, aby cały rysunek był widoczny w oknie podglądu
- * Centruj rysunek w poziomie - zaznaczenie tej opcji powoduje automatyczne umieszczanie rysunku w pozycji centralnej względem poziomej osi współrzędnych okna podglądu
- * Centruj rysunek w pionie - zaznaczenie tej opcji powoduje automatyczne umieszczanie rysunku w pozycji centralnej względem pionowej osi współrzędnych okna podglądu
- * Odtwórz dźwięk automatycznie - zaznaczenie tej opcji powoduje automatyczne odtwarzanie dźwięku w trakcie procesu wnioskowania.

Modyfikacja danych o elementach multimedialnych przypisanych do atrybutów globalnych nie jest możliwa z poziomu źródeł wiedzy.

Parametry - zmienne parametryczne opisujące atrybut. Zmienne parametryczne służą do przechowywania pewnych charakterystycznych wartości danego atrybutu (np. wartość domyślna, wartość minimalna, maksymalna), przy czym ich rodzaj nie jest w żaden sposób narzucony. W środkowej części okna znajduje się lista zmiennych parametrycznych, opisujących bieżący atrybut. Powyżej umieszczono pola edycyjne, umożliwiające wprowadzanie bądź modyfikację nazw oraz wartości zmiennych. Aby utworzyć nową zmienną parametryczną, należy nacisnąć przycisk Dodaj i wprowadzić

do pól edycyjnych nazwę zmiennej oraz jej wartość, a następnie zatwierdzić wprowadzone dane, naciskając przycisk Zatwierdź. W celu modyfikacji nazwy lub wartości wybranej zmiennej, wystarczy wskazać jej nazwę (identyfikator) na liście - w polach edycyjnych pojawią się aktualne dane wskazanej zmiennej - i wprowadzić wymagane poprawki, po czym zatwierdzić zmiany, naciskając przycisk Zatwierdź. Usunięcie wybranej zmiennej parametrycznej możliwe jest przez naciśnięcie przycisku Usuń. Po potwierdzeniu operacji usunięcia, wskazana zmienna zostanie skasowana. Naciśnięcie przycisku Kategorie zmiennych parametrycznych powoduje otwarcie okna parametryzacji, pozwalającego na definiowanie kategorii zmiennych parametrycznych, grupujących wartości zmiennych. Dodawanie lub modyfikowanie zmiennych parametrycznych dla atrybutów globalnych nie jest możliwe z poziomu źródeł wiedzy.

2.2.5. Fakty bazy wiedzy

Informacje o faktach zawartych w aktualnie otwartej bazie wiedzy udostępniane są poprzez wybranie polecenia Fakty opcji Edycja głównego menu systemu CAKE. Wyświetlone okno zawiera listę wszystkich faktów zawartych w aktualnie otwartej bazie wiedzy. Zestawienie faktów w oknie właściwości aplikacji jest możliwe tylko w przypadku aplikacji nie posiadających źródeł wiedzy. W przeciwnym wypadku wszystkie fakty mają charakter lokalny - dostęp do nich możliwy jest jedynie z poziomu poszczególnych źródeł wiedzy. Wprowadzenie nowego faktu do bieżącej bazy wiedzy odbywa się po naciśnięciu przycisku Dodaj. Na ekranie pojawi się okno edycji faktu, umożliwiające określenie postaci nowego faktu. Aby zmodyfikować postać istniejącego w bazie (umieszczonego na liście) faktu należy zaznaczyć żądany fakt i nacisnąć przycisk Zmień lub dwukrotnie kliknąć na danym fakcie - na ekranie wyświetlone zostanie okno edycji, pozwalające na wprowadzenie wymaganych zmian. W celu usunięcia wybranego faktu z bazy wiedzy wystarczy nacisnąć przycisk Usuń - po potwierdzeniu operacji usunięcia wskazany fakt zostanie skasowany.

Edycja faktu

Okno edycji faktu umożliwia określenie postaci faktu bazy wiedzy. Wszystkie fakty zawarte w bazie wiedzy mają postać trójki OAW. Ogólną postać faktu można przedstawić następująco:

[not] atrybut [(obiekt)] [= wartość].

Identyfikator obiektu jest składnikiem opcjonalnym. Dla atrybutów logicznych nie określa się wartości. Umieszczenie klauzuli not oznacza zanegowanie zdefiniowanego faktu. W górnej części okna widoczna jest aktualna postać faktu. Wśród poleceń pozwalających na zmianę wybranego składnika definicji faktu znajdują się:

- * Obiekt - wybór obiektu bazy wiedzy
- * Atrybut - wskazanie atrybutu
- * Wartość - przypisanie wartości atrybutu
- * Negacja - zanegowanie zdefiniowanego faktu.

Po zakończeniu edycji faktu należy nacisnąć przycisk OK, aby zatwierdzić jego nową postać; naciśnięcie przycisku Anuluj jest równoznaczne z zaniechaniem wprowadzania zmian w definicji faktu.

2.2.6. Reguły bazy wiedzy

Informacje o regułach bazy wiedzy udostępniane są poprzez wybranie polecenia Reguły opcji Edycja głównego menu systemu CAKE. W wyświetlonym oknie widoczne są zestawione w postaci listy, wszystkie reguły określone w bieżącej bazie wiedzy. Lista reguł wyświetlana jest w oknie właściwości aplikacji tylko wtedy, gdy aplikacja nie zawiera źródeł wiedzy. W przeciwnym wypadku wszystkie reguły mają charakter lokalny w obrębie poszczególnych źródeł wiedzy; dostęp do nich możliwy jest jedynie z poziomu źródeł. Definiowanie nowych reguł możliwe jest za pośrednictwem okna właściwości reguł. Aby dodać regułę do bazy wiedzy należy nacisnąć przycisk Dodaj, a następnie, po pojawieniu się okna edycji, określić postać nowej reguły. Jeżeli zachodzi konieczność modyfikacji reguły już istniejącej w bazie - wystarczy, po jej zaznaczeniu, nacisnąć przycisk Zmień bądź dwukrotnie kliknąć na danej regule. Na ekranie wyświetlone zostanie okno edycji, pozwalające na wprowadzenie wymaganych zmian. Usuwanie reguł odbywa się przez naciśnięcie przycisku Usuń. Po potwierdzeniu zamiaru usunięcia wskazana reguła zostanie skasowana. Po prawej stronie listy reguł znajduje się pasek narzędziowy umożliwiający realizację następujących operacji:

- Dopisanie nowej reguły będącej kopią reguły wskazanej. Umożliwia to przyspieszenie procesu tworzenia reguł w przypadku gdy kolejna reguła posiada zbliżoną strukturę do reguły już istniejącej.
- Określenie szczegółowych kryteriów poszukiwań reguł (np. występowanie określonej trójki OAW).
- Uporządkowanie listy reguł.

Okno właściwości reguł

Okno właściwości pozwala na przeglądanie oraz modyfikację szczegółowych danych na temat reguł, zdefiniowanych w aktualnej bazie lub źródle wiedzy. Dostęp do określonego typu informacji możliwy jest po wskazaniu jednej z "zakładek, umieszczonych w górnej części okna:

Struktura - dokładna definicja bieżącej reguły. W oknie wyszczególnione są wszystkie elementy struktury bieżącej reguły oraz jej numer identyfikacyjny.

Konkluzja, wchodząca w skład reguły, przybiera zawsze postać trójki atrybut-obiekt-wartość (OAW). Aby określić lub zmodyfikować zawarte w konkluzji elementy należy nacisnąć, znajdujący się obok pola konkluzji, przycisk <<Wstaw (lub <<Modyfikuj). Na ekranie wyświetlone zostanie okno właściwości trójki OAW, umożliwiające edycję poszczególnych elementów konkluzji.

Lista warunków obejmuje wszystkie przesłanki prowadzące do konkluzji bieżącej reguły wnioskowania. W przypadku istnienia kilku warunków, mogą one być łączone przy użyciu operatorów logicznych (i, lub) oraz zagnieżdżane przy pomocy nawiasów. Określenie nowego warunku możliwe jest po naciśnięciu przycisku Dodaj. Na ekranie pojawia się okna dialogowe, pozwalające określić jego rodzaj, a następnie szczegółową postać. Aby zmodyfikować wybrany warunek, wystarczy wskazać go na liście i nacisnąć przycisk Zmień lub dwukrotnie kliknąć na wybranym warunku. Po pojawieniu się okna edycyjnego można wprowadzić żądane zmiany. W celu usunięcia warunku z listy, należy nacisnąć przycisk Usuń. Po zatwierdzeniu operacji usuwania wybrany warunek zostanie skasowany. Po prawej stronie listy warunków znajdują się przyciski pozwalające na wstawienie lub usunięcie nawiasów oraz opcje zmiany operatora logicznego dla aktualnie zaznaczonego warunku. Naciśnięcie przycisku Kolejność... umożliwia zmianę kolejności występowania warunków na liście. (pociąga to jednak za sobą konieczność ponownego wstawienia nawiasów grupujących poszczególne warunki).

Podgląd tekstu - fragment kodu źródłowego opisujący regułę. W oknie podglądu wyświetlana jest definicja bieżącej reguły, przedstawiona w postaci fragmentu kodu źródłowego, zawierającego instrukcje języka opisu wiedzy PC-SHELL. Bezpośrednia edycja tekstu w oknie nie jest możliwa.

Notatki - adnotacje dotyczące reguły. Okno notatek służy do wprowadzania adnotacji dotyczących bieżącej reguły. Notatki mają charakter wyłącznie informacyjny i nie wpływają w żaden sposób na przebieg procesu wnioskowania.

Metafora - słowne objaśnienie bieżącej reguły. Metafory spełniają funkcję objaśniającą treść danej reguły, będąc uzupełnieniem (komentarzem) jej formalnego zapisu. Oprócz tego, metafory mogą zawierać informacje na temat zastosowanych metod rozwiązania problemu lub odwołania do literatury poświęconej danemu zagadnieniu.

2.2.7. Program aplikacji

Aktualna postać bloku sterowania aplikacji udostępniana jest poprzez wybranie polecenia Program opcji Edycja głównego menu systemu CAKE. W wyświetlonym oknie widoczna jest aktualna postać bloku sterowania bieżącej aplikacji. Aby rozpocząć jego edycję należy nacisnąć przycisk Edycja bloku sterowania lub kliknąć dwukrotnie w obszarze okna podglądu tekstu. Jeżeli aplikacja nie zawiera bloku sterującego, na ekranie pojawi się okno wyboru szablonu, umożliwiając wybór jednego z gotowych wzorców tekstu źródłowego bloku, który w następnej kolejności wystarczy zmodyfikować, dostosowując do konkretnych potrzeb. Jeżeli blok sterujący został już wcześniej utworzony, uruchomiony zostanie Edytor bloku sterowania, pozwalający na dalszą edycję tekstu źródłowego bloku sterującego aplikacji. Informacja 'Blok sterowania w trakcie edycji' wyświetlona w oknie podglądu tekstu oznacza, że Edytor bloku sterowania został już uruchomiony. W celu wznowienia edycji wystarczy nacisnąć przycisk Edycja bloku sterowania.

3. Zastosowanie systemu ekspertowego do wspomagania decyzji szefa OPL ZT i dowódcy pplot

3.1. Optymalizacja ugrupowania wojsk OPL

3.1.1. Opis problemu

Jednym z podstawowych problemów rozwiązywanych podczas planowania obrony przeciwlotniczej jest określenie najkorzystniejszego wariantu ugrupowania wojsk OPL tak na szczeblach taktycznych jak i operacyjnych. Zwykle pod pojęciem najkorzystniejszego (optymalnego) wariantu ugrupowania rozumie się takie ugrupowanie które spełniając wymagania operacyjno-taktyczne zapewnia wojskom i obiektom maksymalną efektywność osłony przed środkami napadu powietrznego przeciwnika.

Metody wyboru najkorzystniejszego ugrupowania wojsk OPL oparte są zwykle o analizę i porównanie możliwych do przyjęcia wariantów ugrupowania wojsk OPL w świetle najbardziej prawdopodobnych wariantów działania ŚNP przeciwnika. Dokonanie pełnego przeglądu i porównanie możliwych do przyjęcia wariantów ugrupowania wojsk OPL jest skomplikowane przynajmniej z dwóch powodów. Po pierwsze - istnieją poważne trudności w prognozowaniu najbardziej prawdopodobnych wariantów działania ŚNP przeciwnika, po drugie - uciążliwość i czasochłonność określenia efektywności poszczególnych wariantów ugrupowania, porównanie ich oraz wybór wariantu najkorzystniejszego w świetle przyjętego kryterium jest na tyle duża, że bywa praktycznie nieopłacalna.

Istotną pomoc w tym zakresie przynoszą modele matematyczne, algorytmy i programy (baza metod) systemu MIKRO - OPL. Pozwalają one w stosunkowo krótkim czasie uzyskać dane do przyjęcia optymalnego wariantu ugrupowania wojsk OPL.

Dla określenia optymalnego wariantu ugrupowania wojsk OPL bardzo ważne jest przyjęcie odpowiedniej funkcji kryterium, która z jednej strony zawierałaby cel działania przeciwnika powietrznego, z drugiej zaś cel działania systemu OPL.

Dla przeciwnika powietrznego planującego uderzenie na wojska i obiekty muszą zajść dwa zdarzenia, aby mógł on wykonać zadanie:

1. zdarzenie polegające na tym, że środki napadu powietrznego przeciwnika przenikną przez system OPL (pokonają system OPL, dolecą do osłanianych obiektów),
2. zdarzenie polegające na tym, że środki napadu powietrznego zniszczą obiekty.

Prawdopodobieństwo wykonania zadania przez ŚNP przeciwnika można wyliczyć korzystając ze wzoru:

$$Q = Q1 * Q2$$

gdzie:

$Q1$ - prawdopodobieństwo zdarzenia 1.,

$Q2$ - prawdopodobieństwo zdarzenia 2.

Należy się liczyć z tym, że przeciwnik będzie dążył do maksymalizowania wartości $Q1$ i $Q2$. Przez zwiększenie: stopnia skuteczności poszczególnych sposobów działania, dokładność celowania, skuteczność działania środków rażenia - będzie się starał wybrać taki wariant działania, który zagwarantuje mu osiągnięcie maksymalnych wartości $Q1$ i $Q2$.

Strona organizująca obronę przeciwlotniczą wojsk i obiektów będzie usiłować sprowadzić wartość $Q1$ i $Q2$, a tym samym Q do minimum. Ze względu na ograniczoną ilość sił środków OPL wydzielonych do osłony wojsk i obiektów oraz ograniczone możliwości bojowe sprzętu wartości:

$$P1 = 1 - Q1$$

$$P2 = 1 - Q2$$

można zmienić jedynie w pewnych granicach określonych nierównościami:

$$0 \leq P1 \leq P1_{MAX} \leq 1$$

$$0 \leq P2 \leq P2_{MAX} \leq 1$$

Należy zauważyć, że strona organizująca obronę przeciwlotniczą będzie mieć większy wpływ na zmianę wartości $Q1$. Pewne jest, że prawdopodobieństwo Q będzie nie większe niż mniejsza z dwóch liczb $Q1$ i $Q2$ tzn.

$$Q \leq \min(Q1, Q2)$$

Strona organizująca obronę przeciwlotniczą powinna dążyć do minimalizacji wartości $Q1$.

Z powyższego wynika, że cele obu stron, tzn. przeciwnika powietrznego i strony organizującej obronę przeciwlotniczą, są zdecydowanie przeciwstawne. Taką sytuację można utożsamić z grą o sumie zerowej (grą ściśle konkurencyjną), w której funkcją wypłaty jest prawdopodobieństwo odparcia napadu przeciwnika powietrznego wyrażone wzorem:

$$P = 1 - (1 - P1) * (1 - P2)$$

gdzie:

$P1$ - prawdopodobieństwo tego, że ŚNP przeciwnika nie pokonają systemu OPL (nie przenikną do obiektów),

$P2$ - prawdopodobieństwo tego, że ŚNP przeciwnika które przeniknęły przez system OPL nie zniszczą obiektów.

Funkcja ta w dostatecznym stopniu odzwierciedla zainteresowania oraz cele działania obu stron.

Dla j-tego wariantu nalotu ŚNP przeciwnika ($j = 1, 2, \dots, J$) i k-tego wariantu ugrupowania wojsk OPL ($k = 1, 2, \dots, K$) wzór na prawdopodobieństwo odparcia napadu przeciwnika powietrznego przyjmuje postać:

$$P_{jk} = 1 - (1 - P1_{jk}) * (1 - P2_{jk})$$

Prawdopodobieństwo $P1_{j,k}$ można obliczyć ze wzoru:

$$P1_{jk} = \frac{M_{jk}}{N_{jk}}$$

gdzie:

M_{jk} - wartość przeciętna liczby zniszczonych ŚNP przeciwnika przy j-tym wariantcie jego działania i k-tym wariantcie ugrupowania,
 N_{jk} - ilość ŚNP przeciwnika biorących udział w j-tym wariantcie nalotu i k-tym wariantcie ugrupowania.

Wielkość M_{jk} określa się ze wzorów:

$$M_{jk} = \sum_{n=1}^{N_j} [1 - \prod_{i=1}^{I_k} (1 - p_{in})]$$

lub przy uwzględnieniu ważności celów powietrznych:

$$M_{jk} = \sum_{n=1}^{N_j} a_n [1 - \prod_{i=1}^{I_k} (1 - p_{in})]$$

gdzie:

n - numer celu powietrznego,
 i - numer pododdziału (środka) OPL
 p_{in} - prawdopodobieństwo zniszczenia celu powietrznego o numerze n przez pododdział (środek) OPL o numerze i,
 a_n - ważność celu,
 N_j - ilość ŚNP przeciwnika biorących udział w nalocie w j-tym wariantcie działania,
 I_k - ilość pododdziałów (środków) OPL zwalczających cele powietrzne w k-tym wariantcie ugrupowania.

Prawdopodobieństwo $P1_{jk}$ odpowiada średniemu prawdopodobieństwu zniszczenia każdego środka napadu powietrznego przeciwnika biorącego udział w j-tym wariantcie nalotu i k-tym wariantcie ugrupowania wojsk OPL.

Prawdopodobieństwo $P2_{jk}$ można obliczyć ze wzoru:

$$P2_{jk} = \frac{1}{C_c} \sum_{c=1}^{C_c} P_{cjk} * b_c$$

gdzie:

c - numer obiektu ($c = 1, 2, \dots, C_c$),
 b_c - ważność operacyjna obiektu o numerze c,
 P_{cjk} - prawdopodobieństwo zniszczenia obiektu o numerze c przy j-tym wariantcie działań ŚNP przeciwnika i k-tym wariantcie ugrupowania wojsk OPL.

Prawdopodobieństwo P_{cjk} można określić z zależności:

$$P_{cjk} = 1 - \prod_{n=1}^{N_c} \{1 - z_{cn} * p_{cn} * \prod_{i=1}^{I_c} (1 - y_{in} * p_{cin})\}$$

gdzie:

N_c - ilość ŚNP atakujących obiekt o numerze c ,
 p_{cn} - prawdopodobieństwo zniszczenia obiektu c przez ŚNP o numerze n ,
 p_{cin} - prawdopodobieństwo zniszczenia przez i -ty środek OPL ŚNP o numerze n ,
atakującego obiekt c ,

$$z_{cn} = \begin{cases} 1 & \text{gdy ŚNP o numerze } n \text{ atakuje obiekt } c; \\ 0 & \text{w przeciwnym przypadku.} \end{cases}$$

$$y_{in} = \begin{cases} 1 & \text{gdy środek OPL o numerze } i \text{ zwalcza ŚNP o numerze } n; \\ 0 & \text{w przeciwnym przypadku.} \end{cases}$$

Dążeniem strony organizującej obronę przeciwlotniczą jest doprowadzenie wyrażenia:

$$\sum_{c=1}^{C_c} P_{cjk} * b_c$$

do wartości maksymalnej. Można to uzyskać przez przyjęcie takiego ugrupowania wojsk OPL, które by koncentrowało główny wysiłek obrony na obiektach o dużej ważności operacyjnej.

Dla przeciwnika celem jest sprowadzenie wyrażenia:

$$\sum_{c=1}^{C_c} P_{cjk} * b_c$$

do minimum, poprzez wykonanie uderzeń na obiekty o dużej ważności operacyjnej. Wynika to bezpośrednio ze wzoru na wyliczenie prawdopodobieństwa P_{2jk} . Zmniejszanie P_{cjk} w tych składnikach sumy, w których liczby b_c są duże obniża wartość P_{2jk} i odwrotnie, każde zwiększenie P_{cjk} przy dużych współczynnikach b_c podwyższa wartość P_{2jk} .

Dla $j = 1, 2, \dots, J$ najbardziej prawdopodobnych wariantów działania ŚNP przeciwnika oraz $k = 1, 2, \dots, K$ możliwych do przyjęcia, w określonej sytuacji operacyjnej, wariantów ugrupowania wojsk OPL można skonstruować macierz gry $[P_{jk}]$, w której funkcją wypłaty jest prawdopodobieństwo odparcia napadu przeciwnika powietrznego P_{jk} . Jest to postać normalna gry ściśle konkurencyjnej. Wartości prawdopodobieństwa P_{jk} można uzyskać przy pomocy systemu MIKRO-OPL.

Rozwinięta postać macierzy $[P_{jk}]$ przedstawia się następująco:

		Warianty działania ŚNP przeciwnika							
		1	2	j	J
1		P_{11}	P_{21}	P_{j1}	P_{J1}
2		P_{12}	P_{22}	P_{j2}	P_{J2}
...	
...	
k	Warianty ugrupowania wojsk OPL	P_{1k}	P_{2k}	P_{jk}	P_{Jk}
...	
...	
K		P_{1K}	P_{2K}	P_{jK}	P_{JK}

Dążenie przeciwnika powietrznego do minimalizacji P_{jk} można zapisać:

$$\min_{j=1,2,\dots,J} P_{jk} = \min_{j=1,2,\dots,J} [1 - (1 - P_{1jk}) * (1 - P_{2jk})]$$

Dla ustalonego k powyższe wyrażenie przedstawia minimalną wartość prawdopodobieństwa odparcia napadu przeciwnika powietrznego ze względu na możliwe sposoby jego działania. Strona organizująca obronę przeciwlotniczą wojsk i obiektów powinna wybrać ten wariant ugrupowania, dla którego spełniony jest warunek:

$$V_1 = \max_{k=1,2,\dots,K} \min_{j=1,2,\dots,J} P_{jk} = \max_{k=1,2,\dots,K} \min_{j=1,2,\dots,J} [1 - (1 - P_{1jk}) * (1 - P_{2jk})]$$

Wielkość V_1 jest dolną wartością gry o macierzy $[P_{kj}]$ czyli zagwarantowaną minimalną wygraną strony organizującej obronę przeciwlotniczą.

Przeciwnik powietrzny, posiadając pewne informacje, może zastosować taki wariant działania, przy którym prawdopodobieństwo odparcia napadu powietrznego będzie:

$$V_2 = \min_{j=1,2,\dots,J} \max_{k=1,2,\dots,K} P_{jk} = \min_{j=1,2,\dots,J} \max_{k=1,2,\dots,K} [1 - (1 - P_{1jk}) * (1 - P_{2jk})]$$

Wielkość V_2 jest górną wartością gry o macierzy $[P_{kj}]$, czyli zagwarantowaną minimalną przegraną przeciwnika powietrznego.

Teoria gier macierzowych pozwala stwierdzić, że:

$$1. V_1 \leq V_2, \text{ czyli } \max_{k=1,2,\dots,K} \min_{j=1,2,\dots,J} P_{jk} \leq \min_{j=1,2,\dots,J} \max_{k=1,2,\dots,K} P_{jk}$$

2. Warunkiem koniecznym i wystarczającym na to, aby

$$\max_{k=1,2,\dots,K} \min_{j=1,2,\dots,J} P_{jk} = \min_{j=1,2,\dots,J} \max_{k=1,2,\dots,K} P_{jk}$$

jest, aby macierz miała punkt siodłowy.

3. Jeżeli punkt (j^*, k^*) jest punktem siodłowym macierzy $[P_{jk}]$, to

$$P_{j^* k^*} = \max_{k=1,2,\dots,K} \min_{j=1,2,\dots,J} P_{jk} = \min_{j=1,2,\dots,J} \max_{k=1,2,\dots,K} P_{jk}$$

W takim przypadku $P_{j^* k^*}$ będzie wartością gry, k^* będzie numerem optymalnego wariantu ugrupowania wojsk OPL, zaś j^* będzie numerem optymalnego wariantu działania ŚNP przeciwnika.

Jeżeli $V_1 < V_2$, to gra nie posiada rozwiązania w zakresie tzw. strategii czystych i rozwiązania należy szukać w zakresie strategii mieszanych. Rozwiązanie to można otrzymać stosując dowolną ze znanych metod rozwiązywania gier macierzowych (np. metodę graficzną, programowania liniowego, metodę podmacierzy lub metodę von Neumanna).

Strategią mieszaną strony organizującej obronę przeciwlotniczą i przeciwnika powietrznego definiują zbiory:

$$S_K = \{x_k: x_k \geq 0, \sum_{k=1}^K x_k = 1\}$$

$$S_J = \{y_j: y_j \geq 0, \sum_{j=1}^J y_j = 1\}$$

gdzie:

S_K - zbiór wariantów ugrupowania strony organizującej obronę przeciwlotniczą,

S_J - zbiór wariantów działania przeciwnika powietrznego,

x_k - częstość stosowania wariantu ugrupowania k ,

y_j - częstość stosowania wariantu j działania ŚNP przeciwnika.

Jeżeli rozwiązanie gry uzyskuje się w zakresie strategii mieszanych to oczekiwana wygrana strony organizującej obronę wyraża się wzorem:

$$P(X, Y) = \sum_{k=1}^K \sum_{j=1}^J P_{jk} * x_k * y_j$$

gdzie:

$X = \{x_1, x_2, \dots, x_k, \dots, x_K\}$ - zbiór strategii strony organizującej obronę,

$Y = \{y_1, y_2, \dots, y_j, \dots, y_J\}$ - zbiór strategii przeciwnika powietrznego.

Jeżeli dla pewnych $x^* \in S_K$ i $y^* \in S_J$ spełniony jest warunek:

$$P(x, y^*) \leq P(x^*, y^*) \leq P(x^*, y)$$

to strategię x^* i y^* są **optymalnymi strategiami mieszanymi** a $P(x^*, y^*)$ jest **wartością gry**.

Dolną wartość gry określa się z zależności:

$$V_1 = \max_{x^k \in S_K} \min_{y^j \in S_J} \sum_{k=1}^K \sum_{j=1}^J P_{jk} * x_k * y_j$$

zaś górną wartość gry z zależności:

$$V_1 = \min_{y^j \in S_J} \max_{x^k \in S_K} \sum_{k=1}^K \sum_{j=1}^J P_{jk} * x_k * y_j$$

Po wyznaczeniu optymalnych strategii (x^*, y^*) oraz wartości gry $P(x^*, y^*)$ należy ze względów praktycznych brać pod uwagę te warianty ugrupowania i te warianty działania ŚNP przeciwnika dla których częstości stosowania (x_k, y_j) osiągają największe wartości.

Poniżej przedstawiono dwa przykłady wyznaczania optymalnego wariantu ugrupowania wojsk OPL, wykorzystując do określenia prawdopodobieństwa odparcia napadu powietrznego przeciwnika (funkcja wypłaty w grze) programy systemu MIKRO-OPL.

Przykład 1:

Macierz gry $[P_{jk}]$ przedstawia się następująco:

		Warianty działania ŚNP przeciwnika (j)				
		1	2	3	4	5
Warianty ugrupowania wojsk OPL (k)	1	0,29	0,38	0,42	0,19	0,31
	2	0,32	0,36	0,22	0,27	0,42
	3	0,34	0,41	0,21	0,29	0,37
	4	0,41	0,39	0,41	0,38	0,40
	5	0,28	0,16	0,29	0,38	0,33

$$V_1 = \max_{k=1,2,\dots,K} \min_{j=1,2,\dots,J} P_{jk} = 0,38$$

$$V_2 = \min_{j=1,2,\dots,J} \max_{k=1,2,\dots,K} P_{jk} = 0,38$$

$$V_1 = V_2 = 0,38$$

Ta gra posiada punkt siodłowy ($k = 4, j = 4$). W takim przypadku czwarty wariant ugrupowania ($k=4$) jest wariantem optymalnym dla strony organizującej obronę przeciwlotniczą, a wariant działania ŚNP przeciwnika ($j = 4$) jest optymalnym wariantem działania przeciwnika powietrznego.

Wariant ugrupowania ($k = 4$) gwarantuje prawdopodobieństwo odparcia napadu przeciwnika powietrznego nie mniejsze niż 0,38 bez względu na przyjęty wariant działania ŚNP przeciwnika.

Przykład 2:

Macierz gry $[P_{jk}]$, po odrzuceniu zdecydowanie niekorzystnych wariantów ugrupowania wojsk OPL i działania ŚNP przeciwnika, przedstawia się następująco:

		Warianty działania ŚNP przeciwnika (j)		
		1	2	3
Warianty ugrupowania wojsk OPL (k)	1	0,52	0,41	0,13
	2	0,31	0,43	0,38
	3	0,51	0,37	0,51

$$V_1 = \max_{k=1,2,\dots,K} \min_{j=1,2,\dots,J} P_{jk} = 0,37$$

$$V_2 = \min_{j=1,2,\dots,J} \max_{k=1,2,\dots,K} P_{jk} = 0,43$$

$$V_1 \neq V_2$$

Gra nie posiada punktu siodłowego. W takim przypadku poszukuje się rozwiązania w zakresie strategii mieszanych. Rozwiązując tę grę korzysta się następujących danych:

$$X = \{ x_1 = 0,10; \quad x_2 = 0,50; \quad x_3 = 0,40 \}$$

$$Y = \{ y_1 = 0,16; \quad y_2 = 0,76; \quad y_3 = 0,08 \}$$

Wielkości X i Y są częstościami stosowania poszczególnych wariantów ugrupowania wojsk OPL i działania ŚNP przeciwnika.

Z praktycznego punktu widzenia, ze względu na dużą liczbę wariantów ugrupowania oraz ograniczone możliwości manewrowe sprzętu, nie ma sensu stosowanie wszystkich wariantów ugrupowania wojsk OPL zgodnie z wyliczonymi częstościami. Proponuje się w takich przypadkach przyjęcie założonego progu wzrostu prawdopodobieństwa odparcia napadu powietrznego przez stosowanie strategii mieszanych - oznaczanego przez ε .

- Gdy $|V_1 - V_2| \leq \varepsilon$ to należy pozostać przy tym wariantie ugrupowania wojsk OPL, który gwarantuje prawdopodobieństwo odparcia napadu powietrznego nie mniejsze niż V_1 . Stosowanie manewru zgodnie wyliczonymi częstościami pozwala w tym przypadku jedynie nieznacznie zwiększyć prawdopodobieństwo odparcia napadu powietrznego, nieproporcjonalnie mało w stosunku do zużycia sił i środków na wykonanie manewru.
- Gdy $|V_1 - V_2| > \varepsilon$ to opłacalne jest (jeżeli sytuacja na to pozwala) zaangażowanie odpowiednich sił i środków na wykonanie manewru zgodnie z wyliczonymi częstościami. Pozwoli to znacznie zwiększyć prawdopodobieństwo odparcia napadu przeciwnika powietrznego.

W przykładzie 2, jeżeli przyjmiemy $\varepsilon = 0,10$ to rozwiązanie jest następujące:

- za optymalny wariant ugrupowania wojsk OPL przyjąć wariant trzeci,
- prawdopodobieństwo odparcia nalotu ŚNP wyniesie 0,37.

Jeżeli przyjmiemy $\varepsilon = 0,02$ to rozwiązanie jest następujące:

- stosować manewr z częstościami $\{ x_1=0,10; x_2=0,50; x_3=0,40 \}$.
- prawdopodobieństwo odparcia nalotu ŚNP wyniesie 0,405.

Optymalny wariant ugrupowania wojsk OPL uzyskany w wyniku rozwiązania gry należy każdorazowo przeanalizować z punktu widzenia:

- warunków dowodzenia,
- możliwości manewru,
- zaopatrzenia,
- maskowania,
- obrony przed bronią masowego rażenia.

3.1.2. System ekspertowy UGRUPOWANIE

Poniżej przedstawiono źródłową postać aplikacji UGRUPOWANIE:

```
//-----  
// Opis aplikacji:  
// Określenie optymalnego wariantu ugrupowania wojsk OPL  
//-----  
knowledge base ugr_opl  
  
sources  
co_to_jest:  
  type what_is_file  
  file "ugr_opl.dbw";  
metafory:  
  type metaphors  
  file "ugr_opl.dbm";  
baza_regulowa_001:  
  type kb  
  file "ugr_opl.zw";  
end; // sources  
  
facets  
  
ask yes;  
single yes;  
  
v1 :  
  ask no  
  single yes  
  val range  
  < min, max >;  
v2 :  
  ask no  
  single yes  
  val range  
  < min, max >;  
v :  
  ask no  
  single yes  
  val range  
  < min, max >;  
oplm1 :  
  ask no  
  single yes  
  val range  
  < min, max >;  
oplm2 :  
  ask no  
  single yes  
  val range  
  < min, max >;
```

```

opl3 :
  ask  no
  single yes
  val range
    < min, max >;
opl4 :
  ask  no
  single yes
  val range
    < min, max >;
opl5 :
  ask  no
  single yes
  val range
    < min, max >;
snpx1 :
  ask  no
  single yes
  val range
    < min, max >;
snpx2 :
  ask  no
  single yes
  val range
    < min, max >;
snpx3 :
  ask  no
  single yes
  val range
    < min, max >;
snpx4 :
  ask  no
  single yes
  val range
    < min, max >;
snpx5 :
  ask  no
  single yes
  val range
    < min, max >;
strategia_gry :
  ask  yes
  single yes
  val oneof
  {
    "CZYSTA",
    "MIESZANA"
  };
ma_punkt_siodlowy :
  ask  no
  single yes
  val range
    < min, max >;
optymalne_ugrupowanie_OPL :
  ask  no
  single yes
  val range
    < min, max >;

```

```

prog_wzrostu_prawdopodobienstwa :
  query " Podaj próg wzrostu prawdopodobieństwa odparcia napadu powietrznego:"
  ask  yes
  single yes
  val range
    < 0, 1 >;
nie_stosowac_manewru :
  ask  no
  single yes;
stosowac_manewr :
  ask  no
  single yes;

end; // facets

```

rules

```

  optymalne_ugrupowanie_OPL = 1 if
    ma_punkt_siodlowy = X,
    X == 1,
    strategia_gry = "CZYSTA";
  optymalne_ugrupowanie_OPL = 2 if
    ma_punkt_siodlowy = X,
    X == 2,
    strategia_gry = "CZYSTA";
  optymalne_ugrupowanie_OPL = 3 if
    ma_punkt_siodlowy = X,
    X == 3,
    strategia_gry = "CZYSTA";
  optymalne_ugrupowanie_OPL = 4 if
    ma_punkt_siodlowy = X,
    X == 4,
    strategia_gry = "CZYSTA";
  optymalne_ugrupowanie_OPL = 5 if
    ma_punkt_siodlowy = X,
    X == 5,
    strategia_gry = "CZYSTA";

  strategia_gry = "CZYSTA" if
    ma_punkt_siodlowy > 0;

  strategia_gry = "MIESZANA" if
    ma_punkt_siodlowy = 0;

  ma_punkt_siodlowy = 1 if
    oplm1 = X,
    snpx1 = Y,
    X == Y;
  ma_punkt_siodlowy = 1 if
    oplm1 = X,
    snpx2 = Y,
    X == Y;
  ma_punkt_siodlowy = 1 if
    oplm1 = X,
    snpx3 = Y,
    X == Y;

```

ma_punkt_siodlowy = 1 if
 oplm1 = X,
 snpx4 = Y,
 X == Y;
 ma_punkt_siodlowy = 1 if
 oplm1 = X,
 snpx5 = Y,
 X == Y;

 ma_punkt_siodlowy = 2 if
 oplm2 = X,
 snpx1 = Y,
 X == Y;
 ma_punkt_siodlowy = 2 if
 oplm2 = X,
 snpx2 = Y,
 X == Y;
 ma_punkt_siodlowy = 2 if
 oplm2 = X,
 snpx3 = Y,
 X == Y;
 ma_punkt_siodlowy = 2 if
 oplm2 = X,
 snpx4 = Y,
 X == Y;
 ma_punkt_siodlowy = 2 if
 oplm2 = X,
 snpx5 = Y,
 X == Y;
 ma_punkt_siodlowy = 3 if
 oplm3 = X,
 snpx1 = Y,
 X == Y;
 ma_punkt_siodlowy = 3 if
 oplm3 = X,
 snpx2 = Y,
 X == Y;
 ma_punkt_siodlowy = 3 if
 oplm3 = X,
 snpx3 = Y,
 X == Y;
 ma_punkt_siodlowy = 3 if
 oplm3 = X,
 snpx4 = Y,
 X == Y;
 ma_punkt_siodlowy = 3 if
 oplm3 = X,
 snpx5 = Y,
 X == Y;
 ma_punkt_siodlowy = 4 if
 oplm4 = X,
 snpx1 = A,
 snpx2 = B,
 snpx3 = C,
 snpx4 = D,
 snpx5 = E,
 (X==A|X==B|X==C|X==D|X==E);
 ma_punkt_siodlowy = 4 if

```

    oplm4 = X,
    snpx2 = Y,
    X == Y;
ma_punkt_siodlowy = 4 if
    oplm4 = X,
    snpx3 = Y,
    X == Y;
ma_punkt_siodlowy = 4 if
    oplm4 = X,
    snpx4 = Y,
    X == Y;
ma_punkt_siodlowy = 4 if
    oplm4 = X,
    snpx5 = Y,
    X == Y;
ma_punkt_siodlowy = 5 if
    oplm5 = X,
    snpx1 = Y,
    X == Y;
ma_punkt_siodlowy = 5 if
    oplm5 = X,
    snpx2 = Y,
    X == Y;
ma_punkt_siodlowy = 5 if
    oplm5 = X,
    snpx3 = Y,
    X == Y;
ma_punkt_siodlowy = 5 if
    oplm5 = X,
    snpx4 = Y,
    X == Y;
ma_punkt_siodlowy = 5 if
    oplm5 = X,
    snpx5 = Y,
    X == Y;
optymalne_ugrupowanie_OPL = 1 if
    strategia_gry = "MIESZANA",
    nie_stosowac_manewru,
    oplm1 = X,
    v1 = Y,
    X >= Y;
optymalne_ugrupowanie_OPL = 2 if
    strategia_gry = "MIESZANA",
    nie_stosowac_manewru,
    oplm2 = X,
    v1 = Y,
    X >= Y;
optymalne_ugrupowanie_OPL = 3 if
    strategia_gry = "MIESZANA",
    nie_stosowac_manewru,
    oplm3 = X,
    v1 = Y,
    X >= Y;
optymalne_ugrupowanie_OPL = 4 if
    strategia_gry = "MIESZANA",
    nie_stosowac_manewru,
    oplm4 = X,
    v1 = Y,

```

```

    X >= Y;
    optymalne_ugrupowanie_OPL = 5 if
    strategia_gry = "MIESZANA",
    nie_stosowac_manewru,
    oplm5 = X,
    v1 = Y,
    X >= Y;

    optymalne_ugrupowanie_OPL = 0 if
    strategia_gry = "MIESZANA",
    stosowac_manewr;

    nie_stosowac_manewru if
    v = X,
    prog_wzrostu_prawdopodobienstwa = Y,
    X <= Y;

    stosowac_manewr if
    v = X,
    prog_wzrostu_prawdopodobienstwa = Y,
    X > Y;

```

end;

control

```

function minmax(float &TABLICA[], float &Wm[], float &Wx[], float &Km[], float &Kx[], float
&V1, float &V2 );
function minmax(float &TABLICA[], float &Wm[], float &Wx[], float &Km[], float &Kx[], float
&V1, float &V2 )
begin
    int I, J, IW, IK;
    float B_max, B_min;

    arraySize(TABLICA, 1, IW);
    arraySize(TABLICA, 1, IK);
    // messageBox(0,0,"aaa","bbb");

    IW := IW - 1;
    IK := IK - 1;
    V1 := 0;
    V2 := 1;
    for I := 0 to IW step 1
    begin
        B_min := TABLICA[I,1];
        B_max := TABLICA[I,1];
        for J := 0 to IK step 1
        begin
            if (TABLICA[I,J] > 0)
            begin
                if (TABLICA[I,J] < B_min)
                begin
                    B_min := TABLICA[I,J];
                end;
                if (TABLICA[I,J] > B_max)
                begin
                    B_max := TABLICA[I,J];
                end;
            end;
        end;
    end;
end;

```

```

        end;
    end;
    end;
    Wm[I] := B_min;
    Wx[I] := B_max;
    if (V1 < B_min)
    begin
        if (B_min > 0)
        begin
            V1 := B_min;
        end;
    end;
end;

for J := 0 to IK step 1
begin
    B_min := TABLICA[1,J];
    B_max := TABLICA[1,J];
    for I := 0 to IW step 1
    begin
        if (TABLICA[I,J] > 0)
        begin
            if (TABLICA[I,J] < B_min)
            begin
                B_min := TABLICA[I,J];
            end;
            if (TABLICA[I,J] > B_max)
            begin
                B_max := TABLICA[I,J];
            end;
        end;
    end;
    Km[J] := B_min;
    Kx[J] := B_max;
    if (V2 > B_max)
    begin
        if (B_max > 0)
        begin
            V2 := B_max;
        end;
    end;
end;
end;

int II, JJ, NN, W1, W2, K1, K2, ILW, ILK, Odp;
float RR;
char TEKST, CC, DD;

char TytW[5], Tyt[3];
float TP[5,5], Tpp[5];
float W_min[5], W_max[5], K_min[5], K_max[5], V1, V2, V;
char Par[2], Tpar[3];
float WarPar[2];

char Arkusz, DopArk[2], Info1[17], Info2[7];

char F1[5], F2[5], F3[5];
run;

```

```

// parametry konsultacji
Par[0] := "działania ŚNP przeciwnika";
Par[1] := "ugrupowania wojsk OPL";
Tpar[0] := "Parametry konsultacji.";
Tpar[1] := "Liczba wariantów.";
Tpar[2] := "Wartość";
WarPar[0] := 5;
WarPar[1] := 5;

// dialog pozyskania prawdopodobieństw odparcia przeciwnika powietrznego
DopArk[0] := "arkusz1.xls";
DopArk[1] := "arkusz2.xls";
Arkusze := DopArk[0];
TytW[0] := "Ugrupowanie 1";
TytW[1] := "Ugrupowanie 2";
TytW[2] := "Ugrupowanie 3";
TytW[3] := "Ugrupowanie 4";
TytW[4] := "Ugrupowanie 5";

// objaśnienia
Info1[0] := " ";
Info1[1] := "Dla j=1,2,...,J najbardziej prawdopodobnych wariantów";
Info1[2] := "działania ŚNP przeciwnika oraz k=1,2,...,K możliwych do";
Info1[3] := "przyjęcia, w określonej sytuacji operacyjnej, wariantów";
Info1[4] := "ugrupowania wojsk OPL można skonstruować macierz gry [Pjk], ";
Info1[5] := "w której funkcją wypłaty jest prawdopodobieństwo odparcia";
Info1[6] := "napadu przeciwnika powietrznego Pjk.";
Info1[7] := "Jest to postać normalna gry ściśle konkurencyjnej.";
Info1[8] := "Wartości prawdopodobieństwa Pjk można uzyskać przy pomocy";
Info1[9] := "systemu MIKRO-OPL.";
Info1[10] := "1. ZBIÓR FAKTÓW BAZY WIEDZY TWORZONY JEST NA PODSTAWIE
MACIERZY GRY.";
Info1[11] := "2. POSZUKIWANE JEST ROZWIĄZANIE W ZAKRESIE STRATEGII CZYSTYCH.";
Info1[12] := "3. BRAK ROZWIĄZANIA W ZAKRESIE STRATEGII CZYSTYCH DETERMINUJE
";
Info1[13] := " KONIECZNOŚĆ ZADEKLAROWANIA PROGU WZROSTU
PRAWDOPODOBIENSTWA ";
Info1[14] := " PRZY STOSOWANIU STRATEGII MIESZANYCH.";
Info1[15] := "4. PROPONOWANY JEST WARIANT UGRUPOWANIA LUB SUGEROWANE
JEST";
Info1[16] := " STOSOWANIE MANEWRU Z ODPOWIEDNIMI CZĘSTOŚCIAMI.";

// O systemie
Info2[0] := "System pozwala skonsultować decyzję dotyczącą wyboru wariantu";
Info2[1] := "ugrupowania wojsk OPL w zależności od przewidywanych wariantów";
Info2[2] := "działania ŚNP przeciwnika.";
Info2[3] := "Podstawą teoretyczną działania systemu jest utożsamienie sytuacji";
Info2[4] := "przeciwstawności celów stron z grą ściśle konkurencyjną.";
Info2[5] := "Funkcją wypłaty w grze jest prawdopodobieństwo odparcia napadu";
Info2[6] := "przeciwnika powietrznego.";

createAppWindow;
vignette("UGRUPOWANIE", "\nEkperymentalny, laboratoryjny system ekspertowy\n \n
\nOPTYMALIZACJA UGRUPOWANIA WOJSK OPL\n \n \nAutor: Grzegorz Kott CI AON", "Copyright
©1998-99 Centrum Informatyki AKADEMII OBRONY NARODOWEJ");
setAppWinTitle("UGRUPOWANIE - optymalizacja ugrupowania wojsk OPL");
precision(4,2);

```

menu "Główne &opcje"

1. "&Konsultacja"
2. "&Parametry konsultacji"
3. "&Objaśnienia"
4. "O &systemie"
5. "&Zakończenie konsultacji"

case 1:

```
// messageBox( 0, 0, "Arkusz", Arkusz );
openSheet("Konsultacja",Arkusz);
showSheet("Konsultacja",0);

ftoi( WarPar[0], JJ);
ILW := JJ;
JJ := JJ - 1;

// wypełnienie tablicy - arkusza
for II := 0 to JJ step 1
begin
    W1 := 3;
    W2 := 2 + ILW;
    K1 := II+3;
    K2 := K1;
    getSheetRange("Konsultacja", "", W1, K1, W2, K2, Tpp);
    CC := "Działanie ŚNP przeciwnika - WARIANT ";
    NN := II + 1;
    ntos(NN, DD);
    strcat(CC, DD);
    Tyt[0] := CC;
    Tyt[1] := "Wojska OPL:";
    Tyt[2] := "Pjk";
    nsheetBox( 0, 0, ILW, 0, Tyt, TytW, Tpp );
    setSheetRange("Konsultacja", "", W1, K1, W2, K2, Tpp);
end;

closeSheet("Konsultacja");
openSheet("Konsultacja", Arkusz );
showSheet("Konsultacja", 1 );

// wypełnienie tablicy zawartoscia arkusza
for II := 0 to JJ step 1
begin
    W1 := 3;
    W2 := 2 + ILW;
    K1 := II+3;
    K2 := K1;
    getSheetRange("Konsultacja", "", W1, K1, W2, K2, Tpp);
    for NN := 0 to JJ step 1
    begin
        // TP[II,NN] := Tpp[NN];
        TP[NN,II] := Tpp[NN];
    end;
end;

// analiza danych
```

```

// sprintf( TEKST, " 1==%f, 2==%f, 3==%f, 4==%f, 5==%f", TP[0,0], TP[0,1], TP[0,2], TP[0,3],
    TP[0,4]);
// MessageBox( 0, 0, "PIERWSZY WIERSZ", TEKST );

minmax(TP,W_min,W_max,K_min,K_max,V1,V2);

// sprintf( TEKST, "WIERSZ_MIN 1==%f, 2==%f, 3==%f, 4==%f, 5==%f", W_min[0], W_min[1],
    W_min[2], W_min[3], W_min[4]);
// MessageBox( 0, 0, "MIN PO WIERSZACH", TEKST );
// sprintf( TEKST, "WIERSZ_MAX 1==%f, 2==%f, 3==%f, 4==%f, 5==%f", W_max[0], W_max[1],
    W_max[2], W_max[3], W_max[4]);
// MessageBox( 0, 0, "MAX PO WIERSZACH", TEKST );
// sprintf( TEKST, "KOLUMNA_MIN 1==%f, 2==%f, 3==%f, 4==%f, 5==%f", K_min[0],
    K_min[1], K_min[2], K_min[3], K_min[4]);
// MessageBox( 0, 0, "MIN PO KOLUMNACH", TEKST );
// sprintf( TEKST, "KOLUMNA_MAX 1==%f, 2==%f, 3==%f, 4==%f, 5==%f", K_max[0],
    K_max[1], K_max[2], K_max[3], K_max[4]);
// MessageBox( 0, 0, "MAX PO KOLUMNACH", TEKST );
// sprintf( TEKST, "V1==%f, V2==%f", V1, V2);
// MessageBox( 0, 0, "V1/V2", TEKST );

```

```

// dodanie faktów będących wynikiem analizy danych

```

```

JJ := ILW;
JJ := JJ - 1;
for II := 0 to JJ step 1
begin
    NN := II + 1;
    sprintf( F1[II], "opl%0i",NN);
    sprintf( F3[II], "oplmm%0i",NN);
end;
addFacts(ILW,_,F1,W_min);

```

```

for II := 0 to JJ step 1
begin
    NN := II + 1;
    sprintf( F2[II], "snp%0i",NN);
end;
addFacts(ILW,_,F2,K_max);

```

```

addFact(, ma_punkt_siodlowy, 0);
addFact(, v1, V1);
addFact(, v2, V2);
V := V1 - V2;
V := abs(V);
addFact(, v, V);

```

```

goal("optymalne_ugrupowanie_OPL=X");

```

```

delNewFacts;

```

```

case 2:

```

```

nsheetBox( 0, 0, 2, 0, Tpar, Par, WarPar );
if ((WarPar[0]==5) & (WarPar[1]==5))
begin
    // MessageBox( 0, 0, "WarPar[0]", "5" );
    Arkusz := DopArk[0];

```

```
end
else
begin
    // messageBox( 0, 0, "WarPar[0]", "inne" );
    Arkusz := DopArk[1];
end;

case 3:
    slistBox( 0, 0, "Objaśnienia", " ", Info1, 17, Odp );

case 4:
    slistBox( 0, 0, "O systemie", " ", Info2, 7, Odp );

case 5:
    exit;

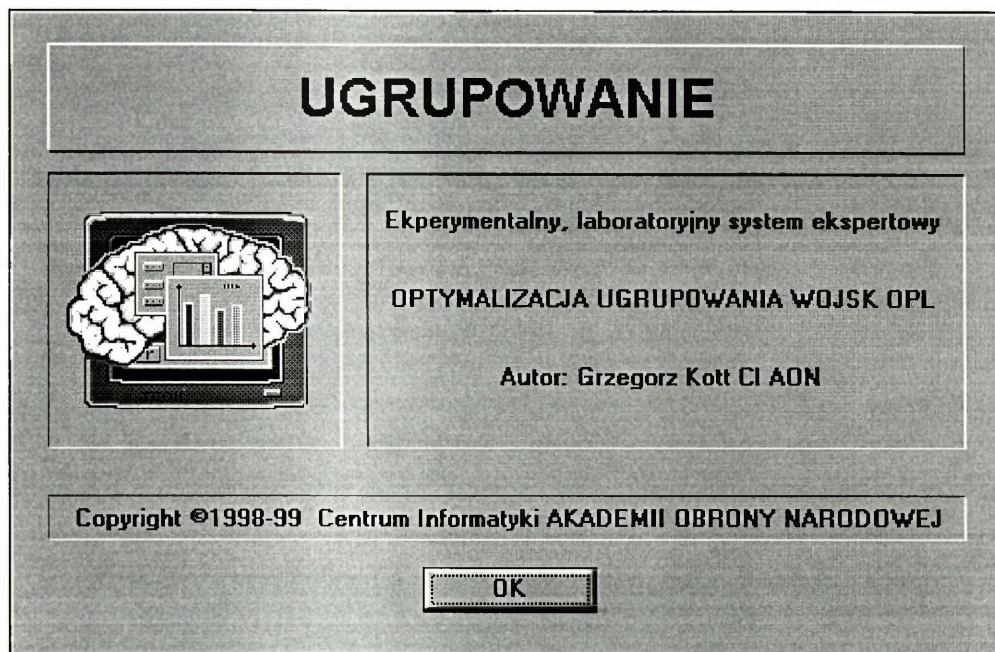
end;

end;

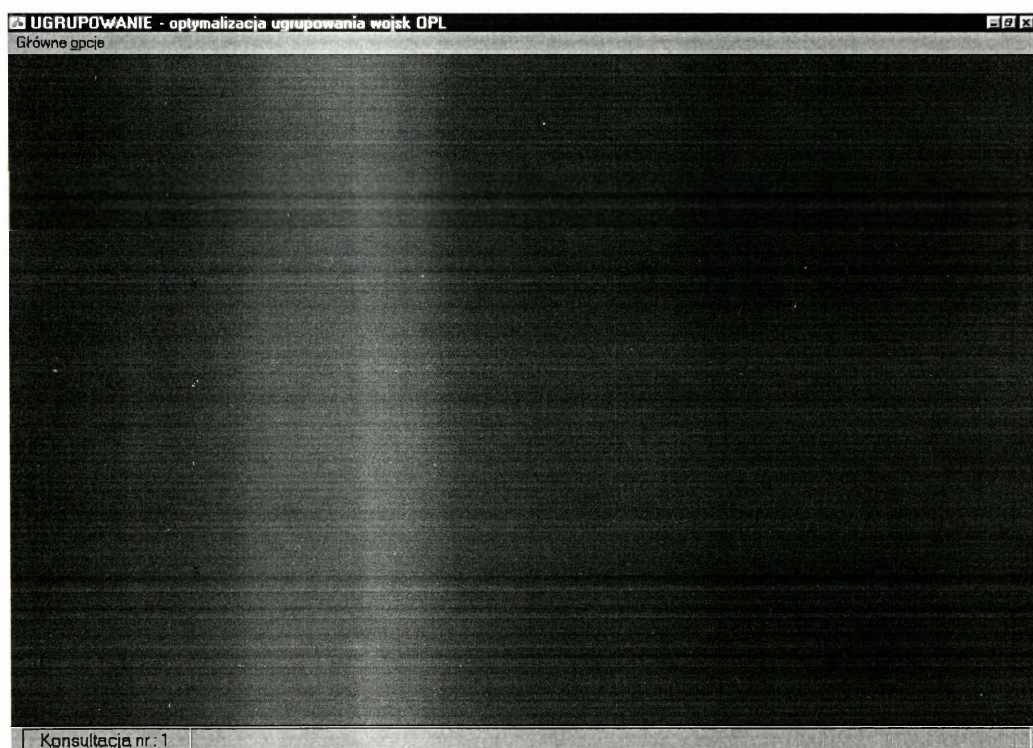
end; // knowledge base
```

3.1.3. Zasady użytkowania systemu UGRUPOWANIE

Uruchomienie systemu ugrupowanie następuje z poziomu systemu PC-SHELL poprzez otwarcie bazy wiedzy zapisanej w pliku ugr_opl.bw. Następuje automatyczna retranslacja aplikacji i jej uruchomienie. Gotowość aplikacji do współpracy z użytkownikiem sygnalizowana jest wyświetleniem poniższych komunikatów:



Naciśnięcie przycisku OK powoduje przejście systemu ekspertowego w tryb konsultacji.



Główne opcje systemu, udostępniane po wybraniu hasła Główne opcje z paska menu, to:

- Konsultacja - umożliwia skonsultowanie z systemem wyboru optymalnego wariantu ugrupowania wojsk OPL na podstawie zdefiniowanych i ocenionych wariantów ugrupowania i wariantów działania ŚNP przeciwnika.
- Parametry konsultacji - umożliwia zdefiniowanie parametrów konsultacji - ilości rozpatrywanych wariantów ugrupowania wojsk OPL i ilości wariantów działania ŚNP przeciwnika.
- Objaśnienia - wyświetla informacje dotyczące zasad działania systemu.
- O systemie - wyświetla informacje o przeznaczeniu systemu.
- Zakończenie konsultacji - powoduje zakończenie działania systemu UGRUPOWANIE i powrót do systemu PC-SHELL.

Najpierw opisane zostaną efekty działania opcji: Parametry konsultacji, Objaśnienia i O systemie. Opcja Konsultacja opisana będzie jako ostatnia - w opisie tym przedstawione zostaną wyniki działania systemu UGRUPOWANIE i ich interpretacja.

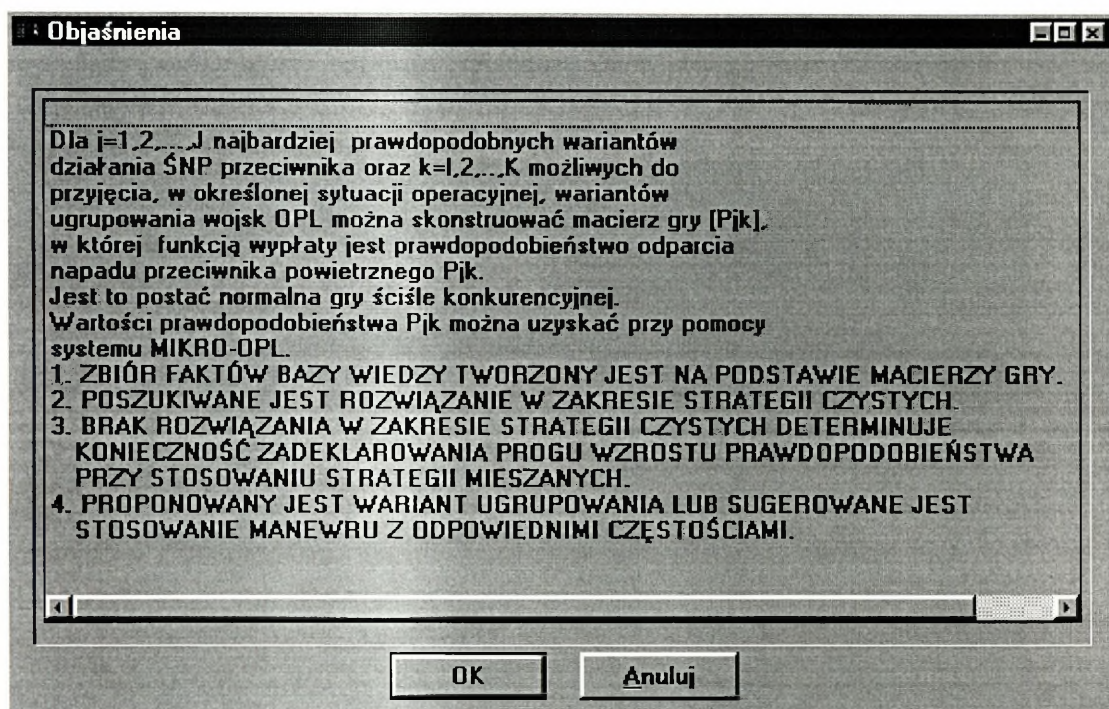
Wybór opcji **Parametry konsultacji** umożliwia zdefiniowanie parametrów konsultacji. Użytkownik powinien podać ilości rozpatrywanych wariantów ugrupowania wojsk OPL i ilości wariantów działania ŚNP przeciwnika. Inicjalnie system zakłada rozpatrywanie pięciu wariantów dla każdej ze stron. Ewentualne zmiany należy wpisać w odpowiednią rubrykę kolumny Wartość. Ustalenie parametrów konsultacji należy zakończyć naciśnięciem przycisku OK.

The screenshot shows a dialog box titled "Parametry konsultacji". At the top left, there is a text input field containing the number "5". Below this is a table with two columns: "Liczba wariantów:" and "Wartość". The table contains two rows of data:

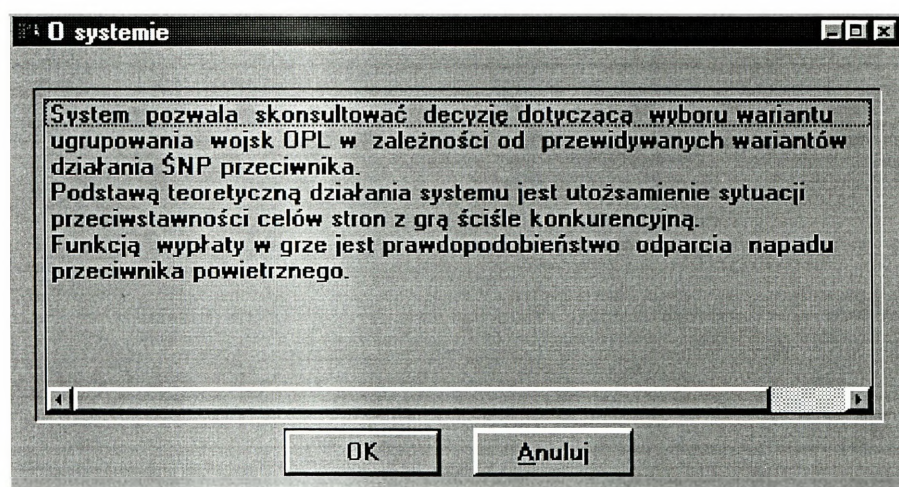
	Liczba wariantów:	Wartość
1	działania ŚNP przeciwnika	5
2	ugrupowania wojsk OPL	5

To the right of the table are two buttons: "OK" and "Anuluj".

Wybór opcji **Objaśnienia** spowoduje wyświetlenie informacji, dotyczących zasad działania systemu UGRUPOWANIE. Naciśnięcie przycisku Anuluj umożliwi powrót do głównego menu systemu.



Wybór opcji **O systemie** spowoduje wyświetlenie informacji, dotyczących przeznaczenia systemu UGRUPOWANIE. Naciśnięcie przycisku Anuluj umożliwi powrót do głównego menu systemu.



Wybór opcji **Konsultacja** umożliwi skonsultowanie z systemem wyboru optymalnego wariantu ugrupowania wojsk OPL na podstawie zdefiniowanych i ocenionych wariantów ugrupowania i wariantów działania ŚNP przeciwnika.

Podstawową czynnością jest wypełnienie arkusza, reprezentującego macierz gry P_{jk} . W zależności od zdefiniowanych parametrów konsultacji wyświetlane są kolejne formularze, umożliwiające zdefiniowanie elementów macierzy.

Dla przykładu 1 formularze mają postać:

The screenshot shows a window titled "Działanie ŚNP przeciwnika - WARIANT 1". At the top left, there is a text input field containing the value "0,29". To the right of this field are two buttons: "OK" and "Anuluj". Below the input field is a table with the following data:

	Wojska OPL:	P_{jk}
1	Ugrupowanie 1	0,29
2	Ugrupowanie 2	0,32
3	Ugrupowanie 3	0,34
4	Ugrupowanie 4	0,41
5	Ugrupowanie 5	0,28

zaś dla przykładu 2 formularze są postaci:

The screenshot shows a window titled "Działanie ŚNP przeciwnika - WARIANT 1". At the top left, there is a text input field containing the value "0,52". To the right of this field are two buttons: "OK" and "Anuluj". Below the input field is a table with the following data:

	Wojska OPL:	P_{jk}
1	Ugrupowanie 1	0,52
2	Ugrupowanie 2	0,31
3	Ugrupowanie 3	0,51

Wynikiem wypełnienia formularzy jest stworzenie arkusza, reprezentującego macierz gry P_{jk} . Dla przykładu 1 przyjmuje ona postać:

		Warianty działania ŚNP przeciwnika				
		1	2	3	4	5
Warianty ugrupowania wojsk OPL	1	0,29	0,38	0,42	0,19	0,31
	2	0,32	0,36	0,22	0,27	0,42
	3	0,34	0,41	0,21	0,29	0,37
	4	0,41	0,39	0,41	0,38	0,40
	5	0,28	0,16	0,29	0,38	0,33

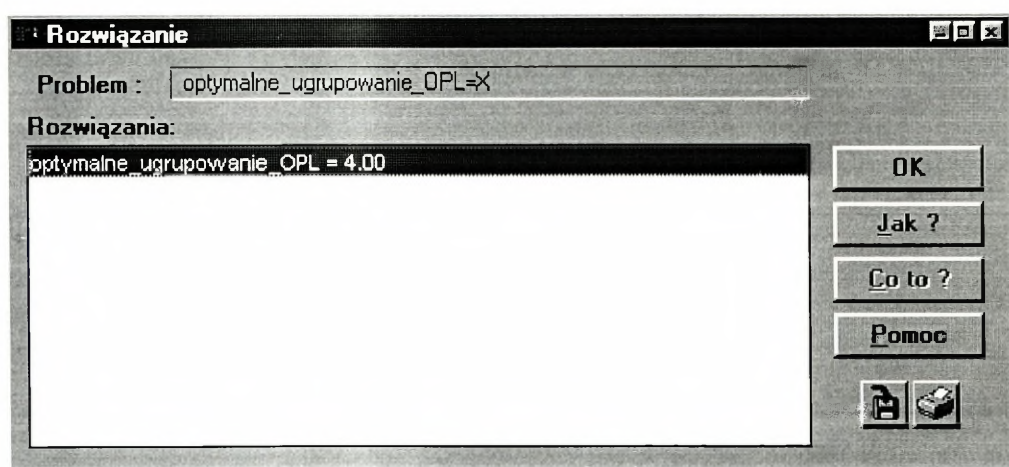
Dla przykładu 2:

		Warianty działania ŚNP przeciwnika		
		1	2	3
Warianty ugrupowania wojsk OPL	1	0,52	0,41	0,13
	2	0,31	0,43	0,38
	3	0,51	0,37	0,51

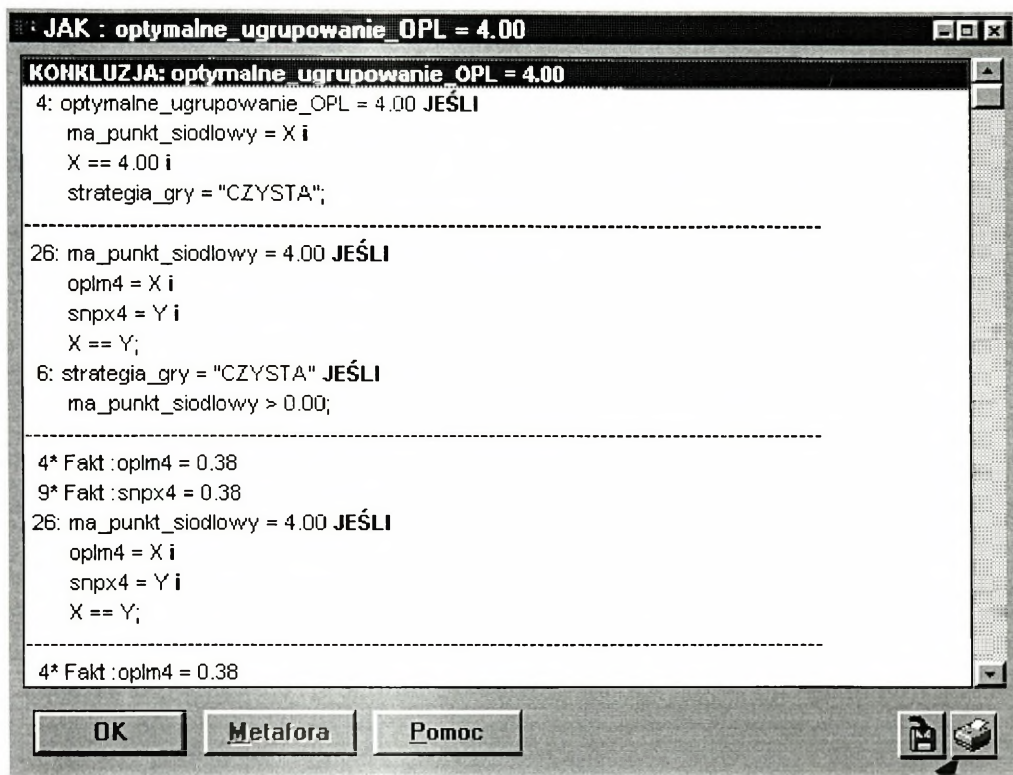
Modyfikacje macierzy prawdopodobieństw są zapamiętywane, stanowiąc podstawę definicji faktów bieżącej konsultacji i podstawę do modyfikacji macierzy prawdopodobieństw kolejnych konsultacji z systemem UGRUPOWANIE.

Zakończenie modyfikacji i zatwierdzenie zmian w ostatnim formularzu wariantu inicjuje proces wnioskowania systemu.

Dla przykładu 1 efektem działania systemu jest wskazanie czwartego wariantu ugrupowania jako optymalnego:



Naciśnięcie przycisku Jak? pozwoli prześledzić proces wnioskowania prowadzący do uzyskania powyższej konkluzji:



Efekt wnioskowania można wydrukować, uzyskując raport systemu:

RAPORT SYSTEMU PC-SHELL 2.2
 KONSULTACJA NR : 86 SPORZĄDZONA DNIA : 30.10.1998 O GODZ : 23:24
 WYJAŚNIENIA 'JAK'

KONKLUZJA: optymalne_ugrupowanie_OPL = 4.00

4: optymalne_ugrupowanie_OPL = 4.00 JEŚLI
 ma_punkt_siodlowy = X i
 X == 4.00 i
 strategia_gry = "CZYSTA";

26: ma_punkt_siodlowy = 4.00 JEŚLI
 opl4 = X i
 snpx4 = Y i
 X == Y;

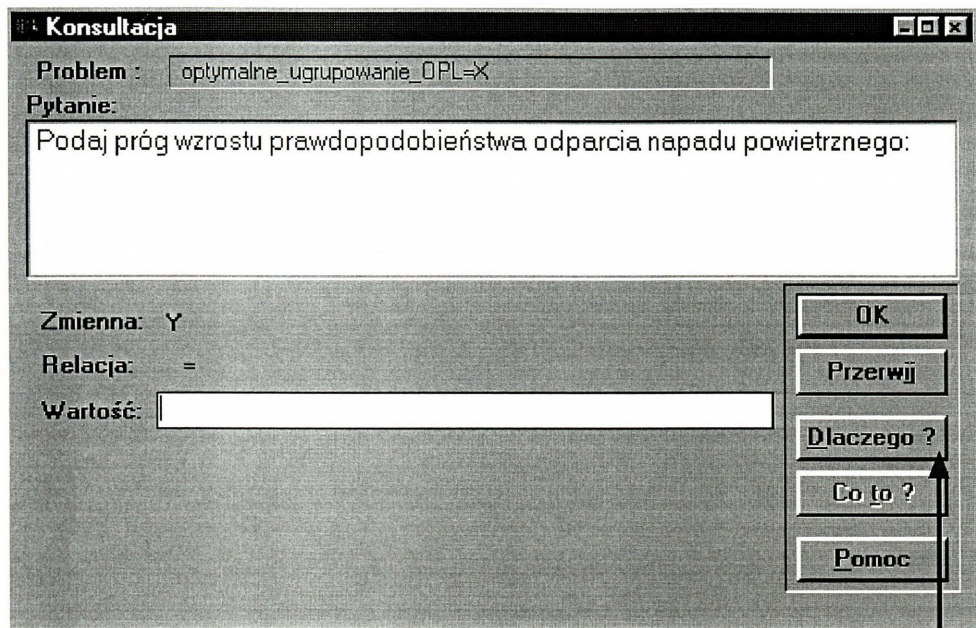
6: strategia_gry = "CZYSTA" JEŚLI
 ma_punkt_siodlowy > 0.00;

4* Fakt : opl4 = 0.38
 9* Fakt : snpx4 = 0.38

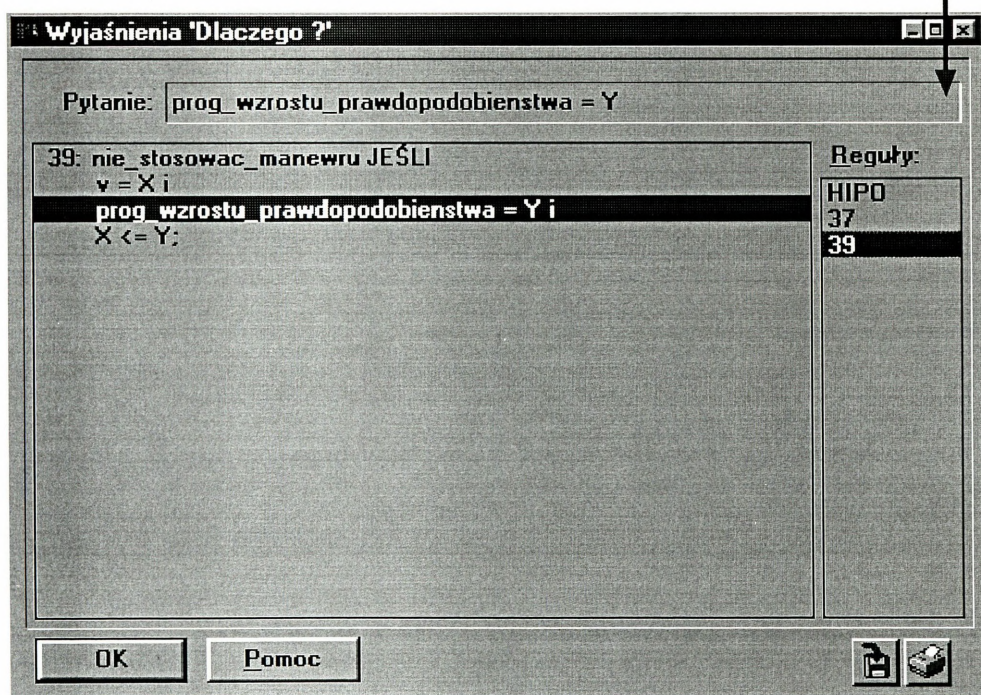
26: ma_punkt_siodlowy = 4.00 JEŚLI
 opl4 = X i
 snpx4 = Y i
 X == Y;

4* Fakt : opl4 = 0.38
 9* Fakt : snpx4 = 0.38

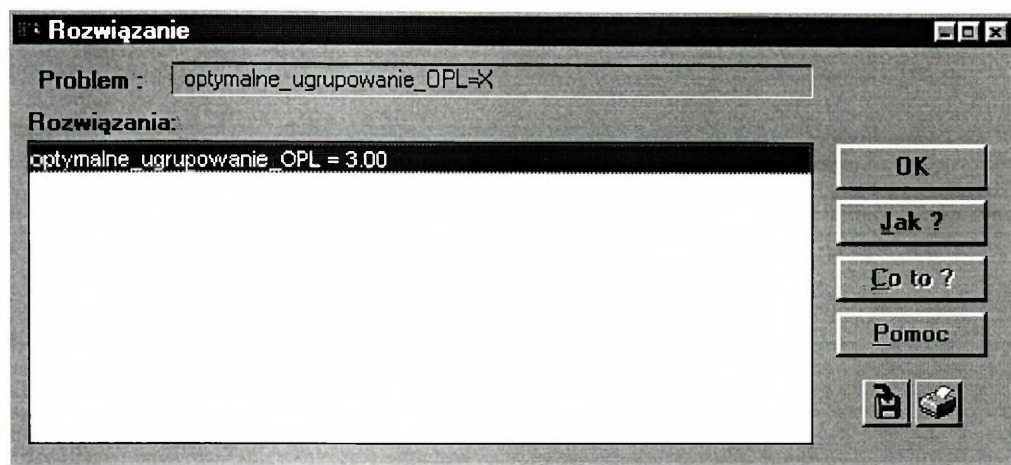
Dane zawarte w przykładzie 2 nie pozwalają na uzyskanie rozwiązania w zakresie strategii czystych. Użytkownik musi zdecydować jaki zakłada próg wzrostu prawdopodobieństwa odparcia napadu powietrznego:



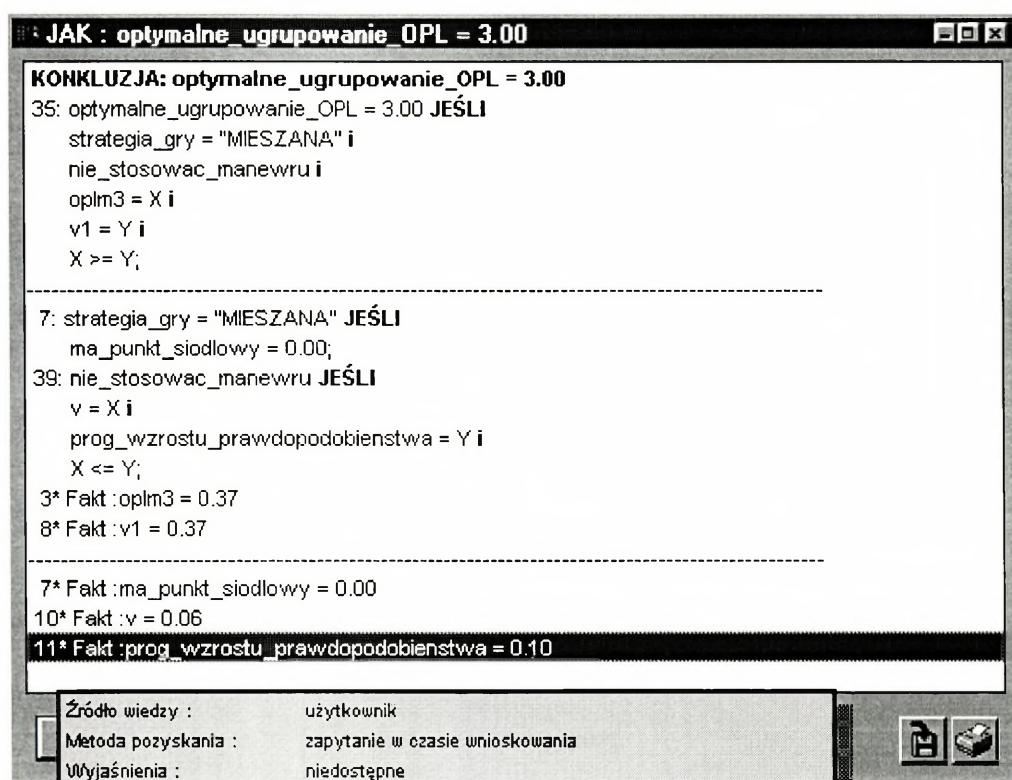
Kontekst zadanego pytania można uzyskać po naciśnięciu przycisku Dlaczego?:



Wpisanie jako wartości progowej liczby 0.10 pociąga za sobą uzyskanie konkluzji - wskazanie trzeciego wariantu ugrupowania jako optymalnego:

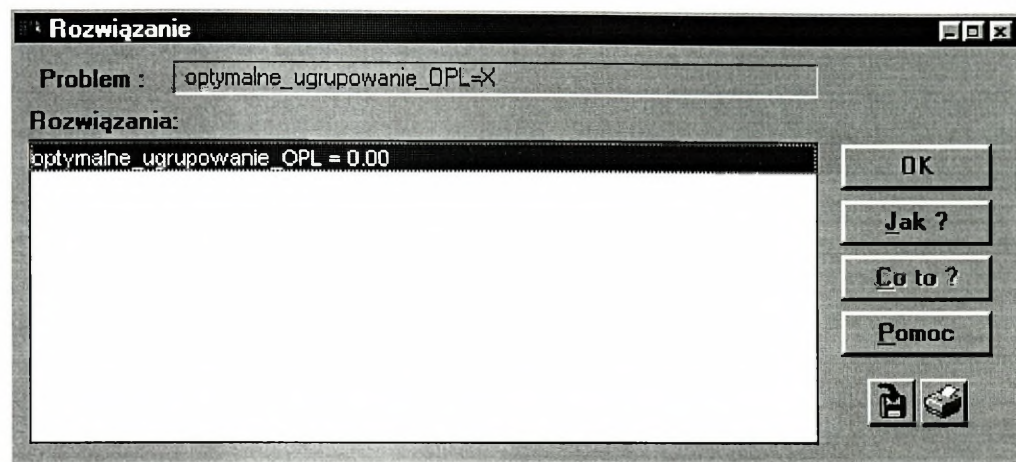


Proces wnioskowania prowadzący do uzyskania powyższej konkluzji jest następujący:

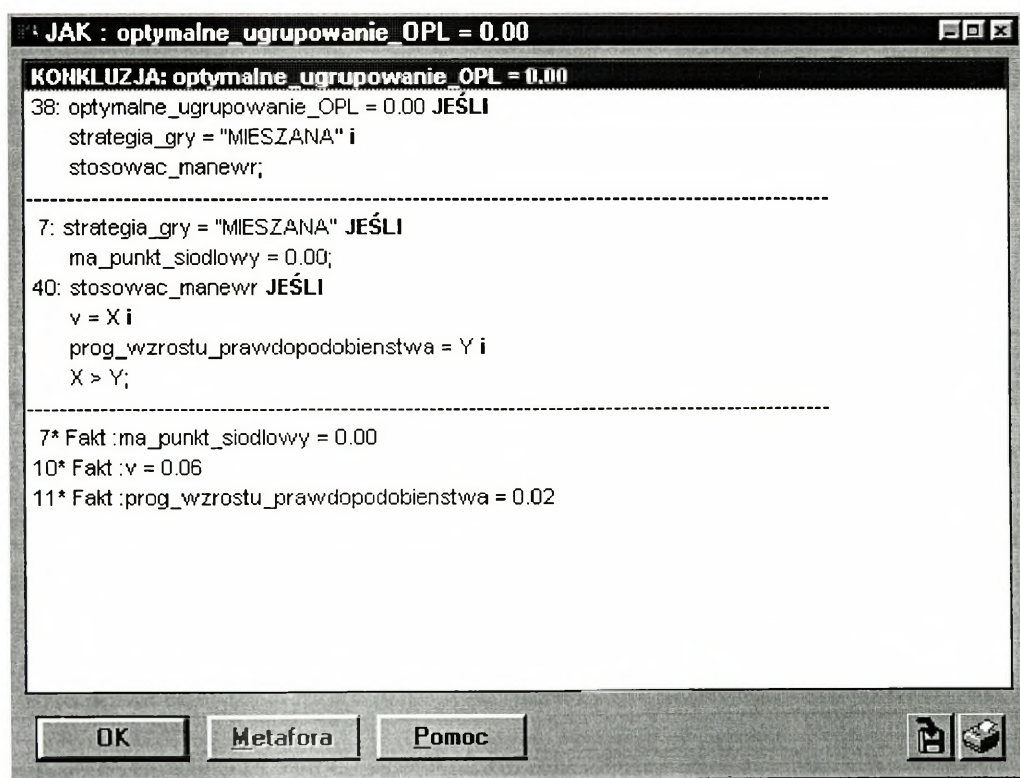


Wskazując na fakt w oknie wyjaśnień Jak? użytkownik może uzyskać informacje o źródle wiedzy, z którego pochodzi fakt oraz metodzie jego pozyskania. Fakt prog_wzrostu_prawdopodobienstwa został pozyskany zapytaniem w procesie wnioskowania.

Wpisanie jako wartości progowej liczby 0.02 pociąga za sobą uzyskanie konkluzji - brak wskazania wariantu ugrupowania jako optymalnego:



Uzyskaną konkluzję należy utożsamić z koniecznością stosowania manewru zgodnie z zadanymi częstościami. Proces wnioskowania prowadzący do uzyskania powyższej konkluzji jest następujący:



3.2. Wspomaganie wypracowania decyzji ogniowej w pułku przeciwlotniczym

Wypracowanie decyzji ogniowej w pułku przeciwlotniczym (pplot) jest jednym z wielu elementów procesu kierowania ogniem w trakcie odpierania nalotów środków napadu powietrznego (ŚNP). Kierowanie ogniem rozpoczyna się w momencie otrzymania danych o ŚNP od przełożonego lub po samodzielnym wykryciu ŚNP za pomocą stacji radiolokacyjnych będących na wyposażeniu pułku. Dane o ŚNP otrzymywane od przełożonego mogą być uzupełnione o decyzje nakazujące, które środki (cele) mają być zwalczane w pierwszej kolejności. Dynamiczny charakter działań bojowych oraz szybko zmieniające się warunki powietrznego pola walki sprawiają, że decyzje przełożonego i wynikające z nich zadania ogniowe do niszczenia ŚNP nie zawsze mogą określać szczegółowe działania. Dlatego też szczególnego znaczenia nabiera fakt samodzielnego wypracowywania decyzji ogniowej przez dowódcę pplot. Aby dowódca mógł wypracować i podjąć decyzję musi on zebrać oraz przeanalizować wiele danych o ŚNP przeciwnika, o działaniach prowadzonych względem tych celów przez sąsiadów oraz inne rodzaje wojsk (np. lotnictwo) oraz dane o stanie sił i środków własnych wojsk a także o działaniach prowadzonych przez własne środki oraz inne rodzaje wojsk. Ma to na celu wypracowanie takiej decyzji ogniowej aby zadać przeciwnikowi maksymalnie straty przy minimalnych stratach własnych. Działalność środków ogniowych nie może powodować strat wśród wojsk własnych (sąsiadów, lotnictwa). Złożoność tego zadania rośnie wraz ze wzrostem liczby sił zaangażowanych w walkę zarówno po stronie własnej jak i przeciwnika. Bez automatyzacji rozwiązanie tego zadania na współczesnym polu walki jest niemożliwe. Zautomatyzowanie tego procesu nie może z kolei ograniczyć wpływu dowódcy na jego zamiar oraz prowadzenie walki.

W trakcie kierowania środkami walki pułku należy:

- odbierać z nadrzędnego szczebla dowodzenia wskazania celów powietrznych z nakazem zwalczania a także informacji o tym, że cel jest zwalczany przez lotnictwo myśliwskie (LM), wojska raketowe OP lub inne środki obrony przeciwlotniczej,
- odbierać z nadrzędnego szczebla dowodzenia ograniczenia dotyczące prowadzenia walki z ŚNP np.: nakazy (zakazy) prowadzenia ognia w określonym obszarze odpowiedzialności,
- odbierać z podległych elementów systemu meldunki o położeniu, stanie gotowości i działalności bojowej,
- odbierać rozkazy, komendy, sygnały dowodzenia, dane dotyczące ugrupowania bojowego wojsk OPL i obszaru odpowiedzialności,
- redagować, w oparciu o meldunki od podwładnych oraz dane własne, meldunki zbiorcze o stanie, położeniu, gotowości i działalności bojowej własnych wojsk i automatycznie je wysyłać,
- redagować i przysyłać podwładnym rozkazy cyfrowe i komendy,
- wypracowywać decyzje zwalczania celów powietrznych oraz zainicjować automatyczną transmisję tych decyzji do podległych baterii i dywizjonów artylerii przeciwlotniczej. Automatycznie wypracowywana decyzja musi uwzględniać zamiar dowódcy, który za pomocą odpowiednich dyrektyw powinien mieć możliwość kształtowania ostatecznej decyzji podziału celów,
- wykorzystywać mechanizm selekcji celów - cele nie spełniające kryterium selekcji nie będą przez moduł podziału rozpatrywane.

3.2.1. Opis problemu

Proces WYPRACOWYWANIA DECYZJI OGNIOWEJ obejmuje następujące czynności:

- wybór celów do zwalczania na podstawie:
 - * analizy położenia trasy lotu ŚNP w stosunku do stref rażenia aktywnych środków rażenia (prognozowania trasy przelotu ŚNP),
 - * analizy położenia celów względem rubieży ostatecznego postawienia zadania (decyzja o zdecentralizowanym wyborze celów do zniszczenia),
- określenie kolejności i wymaganej intensywności zwalczania celów poprzez:
 - * szeregowanie celów poprzez nadawanie im priorytetów określających wymaganą kolejność zwalczania,
 - * optymalizację przydziału celów do zwalczania (względem liczby celów i wartości oczekiwanej strat zadanych ŚNP, mierzonej przez wyrażony im potencjał bojowy),
 - * określenie wymaganej intensywności zwalczania celów (miarą intensywności oddziaływania jest prawdopodobieństwo skutecznego zniszczenia celu),
- określenie wykonawców, którym ma być przekazana decyzja dotycząca przydziału celów do zwalczania na podstawie:
 - * oceny czasu reakcji pododdziału (zestawu), na który składają się:
 - czas niezbędny na powzięcie decyzji i postawienie zadania ogniowego,
 - czas niezbędny na powzięcie decyzji o przejściu do GB 1,
 - czas przekazania sygnału (komendy),
 - czas reakcji zestawu (pododdziału), zależny od stopnia gotowości bojowej i czasu bezpośredniego przygotowania strzelania;
 - * wyznaczonych rubieży postawienia zadań ogniowych,
- określenie sposobów zwalczania celów (podział celów: scentralizowany, zdecentralizowany lub mieszany) - implikowanych:
 - * odległością wykrycia celu,
 - * liczbą celów,
 - * stosowaniem zakłóceń,
 - * manewrami celu,
 - * wysokością celów,
 - * parametrem kursowym celu,
 - * prawdopodobieństwem zniszczenia celu,
 - * stratami w podsystemie dowodzenia.

W wyniku realizacji powyższych przedsięwzięć proces pozwoli określić:

- liczbę środków rażenia bplot (brplot, baplot) gotowych do prowadzenia ognia i możliwości użycia ich do odparcia nalotu,
- możliwości ześrodkowania ognia do niszczenia najważniejszych grup celów i podziału ognia dla jednoczesnego lub kolejnego rażenia kilku celów powietrznych,
- liczbę rakiet posiadanych na wyrzutniach i czasu koniecznego dla ich uzupełnienia,
- możliwości postawienia zadań ogniowych dla pododdziałów.

Efektom działania procesu w danej sytuacji powietrznej może być decyzja o odparciu nalotu ŚNP, obejmująca:

- wybór celów do zniszczenia; podział celów między pododdziały raketowe, pododdziały plot i PRWB (ześrodkowanie lub podział ognia do celów),
- określenie celów do kolejnego ostrzelania,
- wyznaczenie zużycia rakiet i amunicji plot do niszczenia wyznaczonych celów,
- wybór sposobu postawienia zadań ogniowych wykonawcom po uzgodnieniu zadań LM i wojsk raketowych OP oraz sąsiednich oddziałów OPL.

Proces WYPRACOWYWANIA DECYZJI OGNIOWEJ obejmuje następujące podprocesy:

AKTUALIZACJA STANU ŚRODKÓW OGNIOWYCH

Podproces AKTUALIZACJA STANU ŚRODKÓW OGNIOWYCH na podstawie danych opisujących podległe środki ogniowe określa czy dany środek może ostrzeliwać cel powietrzny. Wynikiem pracy tego podprocesu jest wyliczenie wartości dla każdego środka. Wartość równa zero oznacza, że środek spełnia wszystkie wymagania do tego aby móc prowadzić aktywną walkę z celem. Każda inna wartość określa przyczynę dlaczego dany środek nie będzie uwzględniany w procesie przydziału celów.

W celu uproszczenia analizy w trakcie wypracowywania decyzji pojęcie środek ogniowy w dalszej części pracy będzie utożsamiane z pojęciem kanału celowania¹.

Środki ogniowe, które w rzeczywistości mają możliwość zwalczania kilku celów w jednym cyklu strzelania² będą traktowane jako zbiór kanałów celowania, które mają takie same parametry taktyczno-techniczne oraz to samo miejsce dyslokacji.

W systemie czasu rzeczywistego podproces AKTUALIZACJA STANU ŚRODKÓW OGNIOWYCH określa także, na podstawie czasu systemowego, czy środki, które rozpoczęły walkę z wcześniej wskazanym celem już ją zakończyły. W związku z tym porównywany jest czas jaki upłynął od chwili wskazania celu z czasem cyklu strzelania środka. Czas cyklu strzelania umieszczony jest w tablicy danych stałych opisującej parametry taktyczno-techniczne środków. Jeżeli środek zakończył walkę z celem, zerowana jest tablica zawierająca decyzję podziału dla danego środka. W tym przypadku w dalszym procesie podziału w miarę możliwości zostanie przydzielony następny cel do zwalczania.

¹ **Kanał celowania** jest to środek ogniowy, który w danym cyklu strzelania może prowadzić samodzielnie skuteczną walkę z pojedynczym ŚNP.

² **Cykl strzelania** danego środka ogniowego (kanału celowania) jest to czas liczony od chwili otrzymania rozkazu zwalczania celu aż do oceny wyników strzelania do wskazanego celu. Czas ten jest określany w parametrach taktyczno-technicznych danego typu środka ogniowego.

WSKAŹNIKI EFEKTYWNOŚCI STRZELANIA

Najbardziej skomplikowanym, wykonującym najwięcej obliczeń jest podproces WSKAŹNIKI EFEKTYWNOŚCI STRZELANIA. Na podstawie danych o celach i danych o stanie aktywnych środków ogniowych ocenia on możliwości zwalczania celów tzn. wylicza dla każdej pary cel - środek ogniowy, liczbę - wskaźnik efektywności strzelania według metod i wzorów opisanych poniżej. Im większa wartość wskaźnika efektywności tym skuteczniejsze będzie ostrzelanie wybranego celu. Wynikiem działania podprocesu jest macierz wskaźników efektywności strzelania (tablica WSKAŹNIKI):

	śo ₁	śo ₂	śo _i	śo _m
cel ₁	a _{1,1}	a _{2,1}		a _{i,1}		a _{m,1}
cel ₂	a _{1,2}	a _{2,2}		a _{i,2}		a _{m,2}
.....						
cel _j	a _{1,j}	a _{2,j}		a _{i,j}		a _{m,j}
.....						
cel _n	a _{1,n}	a _{2,n}		a _{i,n}		a _{m,n}

gdzie:

- m - maksymalna ilość środków ogniowych,
- n - maksymalna ilość zwalczanych celów (ŚNP),
- i - numer środka ogniowego (kanału celowania) - z zakresu od 1 do m,
- j - numer celu - z zakresu od 1 do n,
- a_{i,j} - wskaźnik efektywności strzelania dla i-tego środka i j-tego celu.

Nieprawidłowo (błędnie merytorycznie) skonstruowany wskaźnik może spowodować, że cele będą przydzielane niezgodnie z założeniami taktycznymi. Dlatego bardzo ważne jest, aby wskaźnik efektywności odpowiednio uwzględniał parametry i współrzędne położenia celu.

Wskaźnik efektywności uwzględnia :

1. Prawdopodobieństwo zniszczenia celu przez środek ogniowy w strefie rażenia,
2. Parametr kursowy celu względem środka oraz błąd wyznaczenia kursu celu. Wielkości te są wyrażane przez prawdopodobieństwo wejścia celu w strefę rażenia środka (P2).
3. Czas dolotu celu do bliższej rubieży wskazywania celów.
4. Priorytet celu.

Postać wskaźnika efektywności jest następująca :

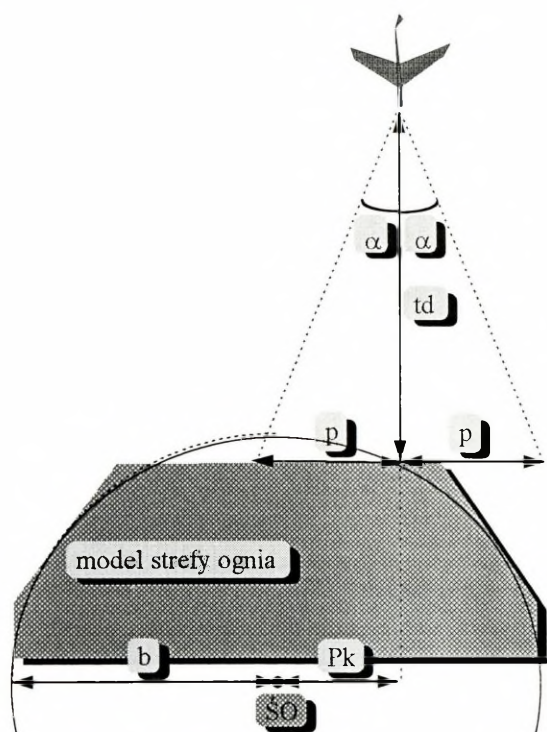
$$a_{ij} = (1 + PRC_i) * (P1_j * P2_{ij} * WSK_{ij}) \quad (1)$$

gdzie:

- i - numer celu,
- j - numer środka,
- $P1_j$ - prawdopodobieństwo zniszczenia celu przez j-ty środek w strefie rażenia,
- $P2_{ij}$ - prawdopodobieństwo wejścia celu w strefę rażenia środka,
- WSK_{ij} - wskaźnik uwzględniający czas dolotu celu,
- PRC_i - priorytet i-tego celu.

Prawdopodobieństwo $P1_j$ jest określone przez parametry taktyczno - techniczne środka ogniowego. W przyjętym modelu środka ogniowego zakłada się, że prawdopodobieństwo to jest jednakowe w całej strefie ognia.

Zakłada się, że punkt wejścia celu w strefę ognia środka ogniowego wynikający z aktualnych parametrów lotu nie jest punktem lecz odcinkiem. Punkt ten został „rozmyty” ponieważ parametry lotu celu tj. prędkość, kurs, wysokość nie są stałe lecz ulegają ciągłym zmianom i wyliczony punkt jest tylko jednym z wielu możliwych. Wszystkie możliwe punkty wejścia celu w strefę ognia tworzą więc odcinek p (rys. 2).



ŚO - miejsce stania środka ogniowego,

b - maksymalny parametr kursowy środka ogniowego,

Pk - parametr kursowy celu,

p - rozrzut parametru kursowego w chwili wejścia celu w strefę rażenia,

α - kąt rozrzutu parametru kursowego.

Rys. 2. Możliwe punkty (odcinek) wejścia celu w strefę ognia

Rozrzut parametru kursowego (p) w chwili wejścia celu w strefę zależy od :

- czasu dolutu celu do dalszej rubieży rażenia,
- prędkości celu,
- zmian wyznaczenia kursu celu wynikających z dyskretnego charakteru tego procesu i jest wyrażony za pomocą następującej zależności:

$$p = td * V_c * a \quad (2)$$

gdzie:

td - czas dolutu do dalszej rubieży strefy ognia.

V_c - prędkość celu,

a - współczynnik kąta rozrzutu - wartość (0.05 - 0.3),

Konsekwencją powyższej zależności jest definicja prawdopodobieństwa wejścia celu w strefę rażenia środka - P_{2ij} :

$$P_{2ij} = \begin{cases} 0 & \text{dla } P_k > p + b, \\ 1 & \text{dla } p < b \text{ i } P_k < b - p, \\ \frac{b}{p} & \text{dla } p > b \text{ i } P_k < p - b, \\ 0.5 + \frac{(b - P_k)}{2p} & \text{dla pozostałych przypadków,} \end{cases} \quad (3)$$

gdzie:

b - maksymalny parametr kursowy środka ogniowego,

P_k - parametr kursowy celu,

p - rozrzut parametru kursowego w chwili wejścia celu w strefę rażenia,

Jeżeli cel będzie się znajdował w strefie rażenia wtedy prawdopodobieństwo P_{2ij} będzie przyjmowało wartość 1.

Wskaźnik WSK zapewnia dwukrotne zwiększenie się wskaźnika efektywności gdy czas dolutu jest większy niż czas cyklu strzelania środka. Zastosowanie tego wskaźnika powoduje, że wskaźnik efektywności strzelania pozwala określić czy opłacalne jest zwalczanie celu bliższego, czy może lepiej poczekać na inny, który jest dalej ale jest celem o wyższym priorytecie. Wskaźnik WSK ma następującą postać :

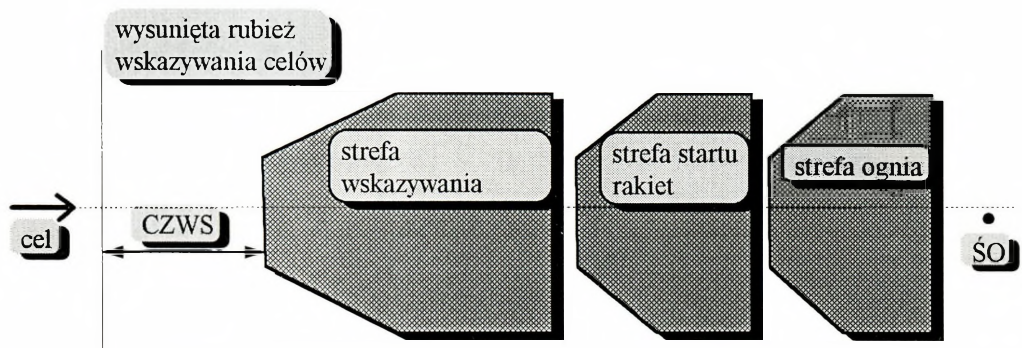
$$WSK_{ij} = \frac{TC_j^2}{tb_{ij}^2 + TC_j^2} \quad (4)$$

gdzie:

TC_j - czas cyklu strzelania środka,

tb_{ij} - czas dolutu do bliższej rubieży wskazywania celu.

Bliższa rubież wskazywania to rubież na której należy wskazać cel wykonawcom, aby mogli oni go wykryć, śledzić i zestrzelić na bliższej granicy strefy ognia. Jest to ostateczny moment na wskazanie celu.



Rys. 3. Zobrazowanie wysuniętej rubieży wskazywania celów.

Strefa ognia - jest to strefa w której mogą być zestrzelone cele. Kształt oraz wielkość rzeczywistej strefy jest zależna od typu środka ogniowego.

Strefa startu rakiet - jeżeli samolot (ŚNP) znajduje się w tej strefie to muszą wystartować rakiety aby został on strącony w strefie ognia. W module podziału celów pplot strefa startu rakiet nie jest wyznaczana.

Strefa wskazywania celów - jeżeli samolot (ŚNP) znajduje się w tej strefie to musi on być wskazany podległym pododdziałom aby one zdążyły go przechwycić i ostrzelać. Z pojęciem strefy wskazywania celów kojarzy się pojęcie dalszej i bliższej rubieży wskazywania celów.

Dalszą rubież wskazywania celów wylicza się według wzoru 5:

$$drw_{ij} = Dds_{ij} + V_i * \left(\frac{Dds_{ij}}{V_r} + Tps_j \right) \quad (5)$$

gdzie:

- i - numer celu,
- j - numer środka,
- drw_{ij} - dalsza rubież wskazywania i-tego celu do zwalczania przez j-ty środek,
- Dds_{ij} - dalsza rubież strefy ognia j-tego środka dla i-tego celu,
- V_i - prędkość i-tego celu,
- V_r - prędkość lotu rakiety j-tego środka,
- Tps_j - czas przygotowania strzelania j-tego środka,

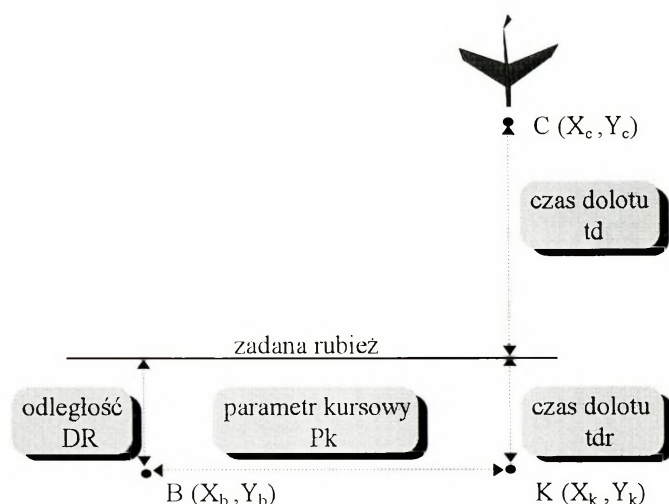
Bliższą rubież wskazywania celów wylicza się według wzoru 6:

$$brw_{ij} = Dbs_{ij} + V_i * \left(\frac{Dbs_{ij}}{V_r} + Tps_j \right) \quad (6)$$

gdzie:

- i - numer celu,
- j - numer środka,
- brw_{ij} - bliższa rubież wskazywania i-tego celu do zwalczania przez j-ty środek,
- Dbs_{ij} - bliższa rubież strefy ognia j-tego środka dla i-tego celu,
- V_i - prędkość i-tego celu,
- V_r - prędkość lotu rakiety j-tego środka,
- Tps_j - czas przygotowania strzelania j-tego środka,

Kształt i wielkość strefy ognia jest dla danego środka ogniowego stała, natomiast kształt i wielkość stref wskazywania celów oraz startu rakiet jest zależna od parametrów strefy ognia oraz od parametrów lotu celu tzn. od jego prędkości, wysokości i kursu. Wyznaczenie czasu dolotu do zadanej rubieży przedstawia rys.4.



Rys. 4. Graficzne przedstawienie czasu dolotu celu do zadanej rubieży.

Czas dolotu celu do zadanej rubieży ogniowego wyznacza się według poniższych wzorów :

$$td = tk - tdr \quad (7)$$

$$tk = \frac{V_x |(X_B - X_C)| + V_y |(Y_B - Y_C)|}{(V_x^2) + (V_y^2)} \quad (8)$$

$$tdr = \frac{DR}{V} \quad (9)$$

$$X_K = X_C + V_x * tk \quad (10)$$

$$Y_K = Y_C + V_y * tk \quad (11)$$

$$Pk = \sqrt{(X_B - X_K)^2 + (Y_B - Y_K)^2} \quad (12)$$

Oznaczenia stosowane na rysunku i w powyższych wzorach :

V_x, V_y - składowe prędkości celu,

X_B, Y_B - miejsce stania baterii,

X_C, Y_C - położenie celu,

tk - czas dolotu celu do punktu X_K, Y_K ,

Pk - parametr kursowy,

DR - odległość rubieży od środka ogniowego,

V - prędkość celu.

Jednym z narzędzi umożliwiających dowódcy pułku aktywne wpływanie na decyzję podziału jest priorytet celów. Wartość priorytetu ma istotny wpływ na wartość wskaźnika efektywności strzelania. W związku z tym zmieniając priorytet celu dowódca zmienia wartość wskaźnika efektywności a tym samym zmienia się automatycznie wyznaczana decyzja podziału. Priorytet jest zwiększany także automatycznie dla celów wskazanych do zwalczania przez przełożonego (maksymalna wartość priorytetu) lub dla celów wskazywanych przez dowódcę.

Priorytet celu wyznacza się na podstawie następujących danych:

1. przynależności celu,
2. typu celu powietrznego,
3. wskaźnika działań względem danego celu,
4. rozkazów z nadrzędnego stanowiska dowodzenia,
5. rozkazów dowódcy pułku.

Dowódca może za pomocą odpowiedniej dyrektywy nadać celowi dowolną wartość priorytetu.

Cel, aby był uwzględniony w procesie automatycznego podziału, spełniać musi następujące warunki :

1. Prędkość celu musi być mniejsza od maksymalnej i większa od minimalnej prędkości przy której środek ogniowy może go zniszczyć.
2. Wysokość celu musi być mniejsza od maksymalnej i większa od minimalnej wysokości przy której środek może go zniszczyć.
3. Z charakterystyki lotu celu wynika, że wejdzie on w strefę rażenia.
4. Cel nie przekroczył bliższej rubieży postawienia zadania ogniowego.

Jeżeli cel nie spełnia jednego z powyższych warunków wtedy jego priorytet przyjmuje się równy zeru.

Priorytet celu (PRC) przyjmuje następujące wartości:

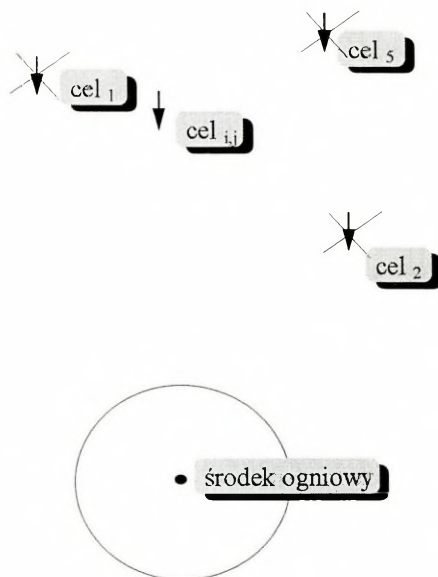
- 2 - cel wskazany przez przełożonego pplot lub z priorytetem przez dowódcę pplot,
- 1 - cel wskazany przez dowódcę pplot lub cel wskazany automatycznie,
- 0 - cel zwalczany przez LM, samolot własny lub cel zwalczany przez inne środki OPL.

DECYZJA DOWÓDCY

W procesie kierowania ogniem dowódca musi mieć możliwość prowadzenia walki zgodnie ze swoją koncepcją. Algorytm automatycznego kierowania ogniem środków obrony przeciwlotniczej wyznacza cyklicznie w ustalonym okresie np. co 10 sekund propozycję podziału celów. Musi on zapewnić, aby każda decyzja dowódcy była zrealizowana nawet jeśli nie pokrywa się ona z propozycją podziału celów jaką algorytm sam automatycznie wypracował. Należy dodać w tym miejscu, że dowódca swoją decyzją może spowodować, że podział celów może być z matematycznego punktu widzenia nieoptymalny. Decyzje dowódcy nakazujące zwalczanie określonego celu przez wskazany środek będą umieszczane za pomocą podprocesu DECYZJA DOWÓDCY w tablicy DYREKTYWY. Do jednego środka ogniowego dowódca może przydzielić ręcznie tylko jeden cel. Zawsze wykonywana jest ostatnia decyzja dowódcy. Podproces DECYZJA DOWÓDCY nadaje wskaźnikowi efektywności wybranego celu maksymalną wartość oraz zeruje pozostałe wskaźniki efektywności z tablicy DYREKTYWY przydzielonego celu odnoszące się do innych środków a także wskaźniki dla danego środka odnoszące się do innych celów. Odpowiada to pominięciu przy podejmowaniu decyzji niektórych nie wybranych celów (patrz rys. 5, 6). W ten sposób zapewnia się, że podproces WSKAZYWANIE CELÓW, wykorzystujący algorytm optymalizacji matematycznej przydzieli „cele dowódcy” do wybranych przez niego środków.

	$\acute{s}o_1$	$\acute{s}o_2$	$\acute{s}o_j$	$\acute{s}o_m$
cel ₁	a _{1,1}	a _{2,1}		0		a _{m,1}
cel ₂	a _{1,2}	a _{2,2}		0		a _{m,2}
.....						
cel _j	0	0		a _{max}		0
.....						
cel _n	a _{1,n}	a _{2,n}		0		a _{m,n}

Rys. 5. Zobrazowanie sposobu realizacji ręcznego przydziału celu



Rys.6. Zobrazowanie zasady realizacji ręcznego przydziału celu do środka ogniowego

POWIELANIE

Zadaniem podprocesu POWIELANIE jest zwielokrotnienie wiersza tablicy wskaźników efektywności (**tablica WSKAŹNIKI**) dla tych celów, które powinny być wskazywane kilku środkom. Pozwala to „oszukać” algorytm optymalizacyjny i spowoduje, że będzie on widział jeden cel jako kilka. Cele w tablicy **WSKAŹNIKI** będą zwielokrotniane w zależności od priorytetu celu i od jego składu.

	śo ₁	śo ₂	śo _i	śo _m
cel ₁	a _{1,1}	a _{2,1}		a _{i,1}		a _{m,1}
cel ₂	a _{1,2}	a _{2,2}		a _{i,2}		a _{m,2}
cel ₂	a _{1,2}	a _{2,2}		a _{i,2}		a _{m,2}
.....						
cel _j	a _{1,j}	a _{2,j}		a _{i,j}		a _{m,j}
.....						
cel _n	a _{1,n}	a _{2,n}		a _{i,n}		a _{m,n}

Rys. 1. Zobrazowanie sposobu realizacji przydziału jednego celu do kilku środków ogniowych w zależności od składu i priorytetu celu.

Dla określenia do ilu kanałów celowania cel ma być wskazany wykorzystywane są dane zawarte w poniższych tabelach, których zawartość musi być ustalona z ekspertami od prowadzenia działań bojowych.

bez ograniczenia zużycia rakiet

	priorytet 0	priorytet 1	priorytet 2	priorytet 3	priorytet 4
cel pojedynczy	0	1	1	1	1
cel grupowy	0	1	2	3	3

z ograniczeniem zużycia rakiet

	priorytet 0	priorytet 1	priorytet 2	priorytet 3	priorytet 4
cel pojedynczy	0	1	1	1	1
cel grupowy	0	1	2	2	2

Tabele te określają na podstawie ograniczeń na zużycie rakiet, priorytetu oraz składu celów, liczbę maksymalnych wskazań celu do kanałów celowania.

Cel, który ma być zwalczany przez kilka środków ogniowych powinien być wskazywany do kilku kanałów celowania nie w jednym, najbliższym lecz w kolejnych cyklach podziału po jednym wskazaniu. Zapobiega to zablokowaniu wszystkich wolnych środków do obsługi jednego celu. Jeżeli po wskazaniu celu do jednego kanału celowania, w kolejnych cyklach wskazywania są nadal wolne środki, które nie przystąpiły do walki z innymi, nowo wykrytymi celami wtedy wybrany cel wskazywany jest do kolejnego środka. Taka konstrukcja metody podziału preferuje rozdzielanie ognia i zwalczanie większej liczby celów, niż zwalczanie tylko jednego celu, chociażby najważniejszego. W tym miejscu należy nadmienić, iż dowódca obiektu ma w każdej chwili możliwość ręcznego przydzielenia celu a zatem może doprowadzić do wskazania jednego celu wszystkim środkom. Czyni to jednak w sposób świadomy.

WSKAZYWANIE CELÓW

Podproces WSKAZYWANIE CELÓW wykorzystuje algorytm podziału celów opisany dalej do wyznaczenia optymalnej decyzji podziału celów. Decyzja ta umieszczona będzie w **tablicy DECYZJE**. Podproces WSKAZYWANIE CELÓW na podstawie tej tablicy po odpowiednim zakodowaniu decyzji podziału utworzy kolejkę depe³ do wysłania dla podległych środków ogniowych.

Na podstawie oceny wyników strzelania oraz odebranych od podwładnych meldunków i aktualnej decyzji podziału określone będą meldunki, które następnie będą zakodowane i wysłane do przełożonego. Meldunki te będą określały aktualny stan walki z celami wskazanymi przez przełożonego. Podobnie zostaną określone i wysłane do sąsiednich systemów informacje o stanie walki ze ŚNP.

ALGORYTM PODZIAŁU CELÓW

Podział celów jest to przyporządkowanie celów do zwalczania poszczególnym środkiem ogniowym. W przypadku kilku środków ogniowych (kanałów celowania) oraz kilku (kilkunastu) celów wariantów podziału może być kilkadziesiąt. W każdym z tych przypadków efekty walki mogą być bardzo różne. Celem procesu WYPRACOWYWANIA DECYZJI OGNIOWEJ jest wypracowanie optymalnej decyzji podziału tzn. takiej aby straty przeciwnika były jak największe.

Podczas wyboru optymalnej decyzji podziału są uwzględniane:

1. priorytety celów,
2. parametry lotu celu względem miejsca dyslokacji poszczególnych środków,
3. rozkazy zwalczania konkretnych celów,
4. informacje o tym, że cel jest zwalczany przez LM lub inne środki OPL,
5. ograniczenia automatycznego zwalczania do zadanej strefy odpowiedzialności i do zadanego przedziału wysokości,
6. zakazy zwalczania konkretnych celów,
7. meldunki o stanie gotowości, o położeniu, o stanie rakiet gotowych do ostrzału oraz o działalności bojowej poszczególnych środków.

Na podstawie danych o celach i danych o stanie aktywnych środków ogniowych podproces WSKAŹNIKI wylicza możliwości zwalczania celów. W wyniku otrzymujemy macierz wskaźników efektywności strzelania. Im większa jest wartość wskaźnika efektywności strzelania tym większe jest prawdopodobieństwo na skuteczne zwalczenie celu. Zbiór wskaźników dla wszystkich par środek - cel stanowi tablicę wskaźników, która jest daną wejściową dla modułu optymalizacji decyzji - do określenia optymalnej decyzji podziału.

³ Depesza jest to część informacyjna słów danych przesyłanych między zautomatyzowanymi obiektami dowodzenia i kierowania środkami walki.

Istotą podziału celów jest wybór optymalnej decyzji podziału z uwzględnieniem wszystkich wolnych środków ogniowych mogących ostrzeliwać cele.

Proces WYPRACOWYWANIA DECYZJI OGNIOWEJ ma wytworzyć decyzję podziału, która będzie następnie wysłana do podwładnych. Proces ten wywoływany jest co określony kwant czasu (np. 10 sek.). Im ten czas jest krótszy tym kierowanie ogniem jest dokładniejsze lecz wymagania na sprzęt są wtedy większe. W rzeczywistym systemie musi tu być ustalony rozsądny kompromis.

Podział celów jest wypracowywany w automatycznym lub półautomatycznym reżimie pracy. W reżimie automatycznym są uwzględniane cele znajdujące się w sektorze lub pasie odpowiedzialności. Jeśli nie będzie celów w sektorze (pasie), wtedy będą uwzględniane cele spoza sektora. W reżimie półautomatycznym będą zwalczane tylko cele wskazane przez dowódcę pułku (za pomocą dyrektyw) lub jego przełożonego (za pomocą depeesz⁴).

Założenia algorytmu podziału celów:

1. Oprogramowanie ma zapewnić kierowanie ogniem pułku plot.,
2. Podział celów ma przydzielać po jednym celu dla każdego kanału celowania na jeden cykl strzelania.
3. Podział celów ma analizować możliwości zwalczania n celów.
4. Cele przydzielane ręcznie przez dowódcę, nie są objęte kontrolą automatyczną i powinny być wyraźnie zaznaczone.
5. Podział celów musi udostępniać wszystkie dane obrazujące aktualny stan walki z celami wskazanymi do zwalczania.
6. Podział celów powinien w szerokim zakresie wspomagać dowódcę przy ręcznym przydzielaniu celów.

Podział celów w pplot wyznacza decyzję podziału na pojedyncze środki ogniowe (kanały celowania). Ponieważ zakłada się, że podległe pododdziały będą wyposażone w zautomatyzowane wozy dowodzenia, dlatego do podległych baterii wysyłane będą tylko numery celów, które mają być zwalczane przez środki ogniowe wchodzące w skład poszczególnych baterii bez numerów środków, które mają zwalczać te cele.

Decyzje podziału celów dla poszczególnych środków ogniowych będą wyznaczane w baterijnym obiekcie (wozie) dowodzenia, który kieruje nimi bezpośrednio.

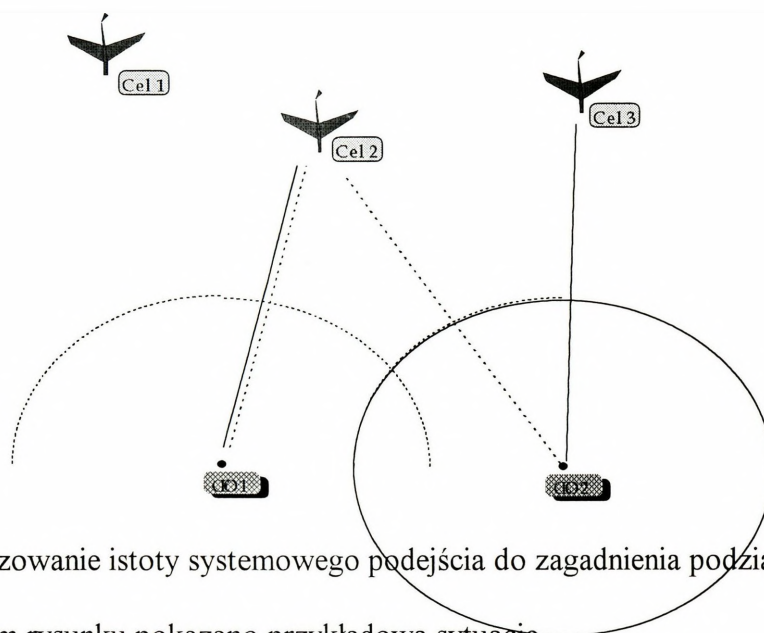
Cele, których numery będą odbierane w baterii od pplot będą traktowane priorytetowo w przeciwieństwie do pozostałych celów śledzonych samodzielnie przez baterie. Z tego wynika, że cele nie wskazane przez pplot będą zwalczane przez baterię tylko wtedy, gdy obsłuży ona (zwalczy) cele wskazane przez przełożonego. Oczywiście dowódca baterii może za pomocą ręcznego przydziału celów spowodować zwalczanie celów w inny sposób ale czyni to na swoją odpowiedzialność.

Ten sposób podejścia do systemowego zwalczania celów powoduje, że pplot staje się koordynatorem w kierowaniu ogniem podległych mu aktywnych środków ogniowych.

Odpowiadając na pytanie: Czy podziałem celów nie mogłyby się zajmować tylko bateryjne obiekty dowodzenia? Odpowiedź brzmi nie, ponieważ suma optymalnych bateryjnych podziałów celów nie jest optymalnym pułkowym podziałem celów. Ten systemowy sposób podejścia do zagadnienia podziału celów będzie zapobiegał sytuacjom, kiedy dwie baterie zwalczałyby ten sam cel, natomiast inne cele nie byłyby zwalczane. Ponadto dowódca pułku

⁴ Wymiana danych bazuje na depepszach opisanych w dokumencie opracowanym przez Szefostwo WOPL pt. „Zestaw depepsz informacyjnych przekazywanych w PZSyD ZENIT” WARSZAWA 1995.

może samodzielnie, według swojej koncepcji walki, nierównomiernie wykorzystywać poszczególne baterie mając na uwadze walkę z ŚNP w dłuższej perspektywie czasowej.



Rys. 8. Zobrazowanie istoty systemowego podejścia do zagadnienia podziału celów

Na powyższym rysunku pokazano przykładową sytuację.

Gdyby środki ogniowe (baterie) samodzielnie podejmowałyby decyzję o zwalczaniu celów oba środki zwalczyłyby cel 2 (przydział celów zaznaczony liniami przerywanymi).

Podział celów w pułku zaznaczony jest liniami ciągłymi - cel 2 zwalczany jest przez środek 1 natomiast cel 3 zwalczany jest przez środek 2. Wskazania z pułku do baterii (środka) byłyby następujące: cel 2 jako priorytetowy wskazany byłby do środka 1, cel 3 jako priorytetowy wskazany byłby do środka 2.

Po uwzględnieniu priorytetów podział celu w obu bateriach (środkach) pokryłby się z decyzją wyznaczoną przez pplot. Efektem takiego postępowania jest fakt, że zamiast ostrzeliwania jednego celu dwoma środkami, ostrzelano dwa cele.

POSTAĆ ZADANIA PODZIAŁU CELÓW

Dla zadanej macierzy $A=[a_{ij}]_{M \times N}$ wyznaczyć :

$$x^* \in \Omega \text{ aby } Q(A, x^*) = \max_{x \in \Omega} Q(A, x)$$

gdzie:

- A - macierz wskaźników efektywności,
- M - ilość środków,
- N - ilość celów do podziału,
- a_{ij} - wskaźnik efektywności strzelania (wskaźnik ten można interpretować również jako wskaźnik strat poniesionych przez j-ty cel w wyniku oddziaływań i-tego środka ogniowego),
- x^* - optymalna decyzja podziału celów,
- x - dopuszczalna decyzja podziału celów,
- Ω - zbiór rozwiązań dopuszczalnych,
- Q - globalny wskaźnik efektywności (strat przeciwnika).

Globalny wskaźnik efektywności równa się sumie wskaźników efektywności strzelania poszczególnych środków do przydzielonych celów. Wskaźnik ten ma następującą postać :

$$Q(A, x) = \sum_{i=1}^M \sum_{j=1}^N a_{ij} * x_{ij}$$

gdzie:

- a_{ij} - element decyzji podziału celów - wskaźnik efektywności strzelania,
- $x_{ij} = 0$ - gdy j-ty cel nie jest przydzielony do i-tego środka,
- 1 - gdy j-ty cel jest przydzielony do i-tego środka (może być przydzielony tylko jeden cel).

Dla wyznaczenia optymalnej decyzji podziału celów należy wykorzystać metodę programowania matematycznego. Zadaniem tej metody będzie ustalenie na podstawie tablicy zawierającej wskaźniki efektywności strzelania optymalnego podziału celów.

Stopień przydatności każdego środka do zwalczania określonego celu ocenia funkcja przydatności. Funkcja ta przyporządkowuje każdej parze środek - cel nieujemną liczbę rzeczywistą zwaną wskaźnikiem efektywności strzelania. Im większa wartość wskaźnika efektywności tym skuteczniejsze będzie ostrzelanie wybranego celu. Postać wskaźnika jest opisana we wcześniejszej części opracowania. Zbiór wskaźników dla wszystkich par środek - cel tworzy macierz wskaźników efektywności. Macierz ta jest daną wejściową na której bazuje metoda programowania matematycznego, która ma wyznaczyć dla każdego środka cele do zniszczenia w taki sposób, aby straty zadane przeciwnikowi były maksymalne.

3.2.2. System ekspertowy DECYZJA

Poniżej przedstawiono źródłową postać aplikacji DECYZJA:

```
//-----  
//  
// Źródłowa postać aplikacji zapisanej w pliku:  
// d:\sfinks22\cattbw\w_dec_og.bbw  
//  
//-----  
  
knowledge base w_dec_og  
  
sources  
  
baza_regulowa_001:  
  type kb  
  file "W_DEC_og.zw";  
  
co_to_jest:  
  type what_is_file  
  file "coto001.dbw";  
  
metafory:  
  type metaphors  
  file "meta001.dbm";  
  
end; // sources  
  
facets  
  
ask yes;  
single yes;  
  
bliższa_rubież_strefy_ognia_środka_dla_celu :  
  ask yes  
  single yes  
  query "Podaj bliższą rubież strefy ognia środka ogniowego dla celu (km):"  
  unit "km"  
  val range  
    ( 0, max >;  
  
czas_cyklu_strzelania_środka :  
  ask yes  
  single yes  
  query "Podaj czas cyklu strzelania środka ogniowego (sek):"  
  unit "sek"  
  val range  
    ( 0, max >;  
  
czas_dolotu_do_bliższej_rubieży_wskazywania_celu :  
  ask yes  
  single yes  
  unit "sek"  
  val range  
    ( 0, max >;
```

czas_dolotu_do_dalszej_rubiezy_strefy_ognia :

```
ask yes
single yes
query "Podaj czas dolotu do dalszej rubiezy strefy ognia:"
unit "sek"
val range
< 0, max >;
```

czas_przygotowania_strzelania_środka :

```
ask yes
single yes
query "Podaj czas przygotowania strzelania środka ogniowego (sek):"
unit "sek"
val range
< 0, max >;
```

dalsza_rubież_strefy_ognia_środka_dla_celu :

```
ask yes
single yes
query "Podaj dalszą rubież strefy ognia środka dla celu (km):"
unit "km"
picture "d:\sfinks22\cattbw\r5.bmp"
val range
( 0, max >;
```

dalsza_rubież_wskazywania_celów_do_zwalczania :

```
ask yes
single yes
val range
( 0, max >;
```

maksymalny_parametr_kursowy_środka_ogniowego :

```
ask yes
single yes
query "Podaj maksymalny parametr kursowy środka ogniowego (km):"
unit "km"
val range
( 0, max >;
```

parametr_kursowy_celu :

```
ask yes
single yes
query "Podaj parametr kursowy celu:"
unit "km"
val range
< 0, max >;
```

prawdopodobieństwo_zniszczenia_celu_w_strefie_rażenia :

```
ask yes
single yes
query "Podaj prawdopodobieństwo zniszczenia celu przez środek w strefie rażenia:"
unit "parametry taktyczno-techniczne środka"
val range
< 0, 1 >;
```

prawdopodobieństwo_wejścia_celu_w_strefę_rażenia_środka :

```
ask yes
single yes
val range
  < 0, 1 >;
```

prędkość_celu :

```
ask yes
single yes
query "Podaj prędkość celu (m/sek):"
unit "m/sek"
val range
  ( 0, max >;
```

prędkość_lotu_rakiety_środka :

```
ask yes
single yes
query "Podaj prędkość lotu rakiety danego środka ogniowego (m/sek):"
unit "m/sek"
val range
  ( 0, max >;
```

priorytet_celu :

```
ask yes
single yes
query "Podaj priorytet celu:"
val oneof
{
  0,
  1,
  2,
  3,
  4
};
```

rozrzut_parametru_kursowego :

```
ask yes
single yes
unit "km"
val range
  < 0, max >;
```

wskaźnik_czasu_dolotu :

```
ask yes
single yes
val range
  < 0, max >;
```

wskaźnik_efektywności_strzelania :

```
ask yes
single yes
val range
  < 0, 1 >;
```

współczynnik_kąta_rozrzutu_pk :

```
ask yes
single yes
query "Podaj współczynnik kąta rozrzutu parametru kursowego celu - wartość <0.05, 0.3>:"
unit "wartość <0.05, 0.3>"
picture "d:\sfinks22\cattbw\lr2.bmp"
val range
< 0.05, 0.3 >;
```

bliższa_rubież_wskazywania_celów_do_zwalczenia :

```
ask yes
single yes
val range
< 0, max >;
```

bliższa_rubież_czas_dolotu :

```
ask yes
single yes
val range
< 0, max >;
```

dalsza_rubież_czas_dolotu :

```
ask yes
single yes
val range
< 0, max >;
```

k_czas_dolotu :

```
ask yes
single yes
val range
< 0, max >;
```

składowa_X_prędkości_celu :

```
ask yes
single yes
query "Podaj wartość składowej X prędkości celu (m/sek):"
val range
< 0, max >;
```

składowa_Y_prędkości_celu :

```
ask yes
single yes
query "Podaj wartość składowej Y prędkości celu (m/sek):"
val range
< 0, max >;
```

wsp_X_celu :

```
ask yes
single yes
query "Podaj współrzędną X celu (km):"
val range
< 0, max >;
```

wsp_X_środk_a_ogniowego :

```
ask yes
single yes
val range
```

< 0, max >;

wsp_Y_celu :

ask yes

single yes

query "Podaj współrzędną Y celu (km):"

val range

< 0, max >;

wsp_Y_środka_ogniowego :

ask yes

single yes

val range

< 0, max >;

x_k :

ask yes

single yes

val range

< 0, max >;

y_k :

ask yes

single yes

val range

< 0, max >;

end; // facets

rules

1001: wskaźnik_efektywności_strzelania = X if

prawdopodobieństwo_zniszczenia_celu_w_strefie_rażenia = P1,

prawdopodobieństwo_wejścia_celu_w_strefę_rażenia_środka = P2,

wskaźnik_czasu_dolotu = WSK,

priorytet_celu = PRC,

A := P1 * P2,

B := A * WSK,

C := 1 + PRC,

X := B * C;

1003: prawdopodobieństwo_wejścia_celu_w_strefę_rażenia_środka = 0 if

parametr_kursowy_celu = PK,

maksymalny_parametr_kursowy_środka_ogniowego = B,

rozzut_parametru_kursowego = P,

A := P + B,

PK > A;

- 1004: $\text{prawdopodobieństwo_wejścia_celu_w_strefę_rażenia_środką} = 1$ if
 $\text{parametr_kursowy_celu} = PK,$
 $\text{maksymalny_parametr_kursowy_środką_ogniowego} = B,$
 $\text{rozrzut_parametru_kursowego} = P,$
 $A := B - P,$
 $PK < A,$
 $P < B;$
- 1005: $\text{prawdopodobieństwo_wejścia_celu_w_strefę_rażenia_środką} = X$ if
 $\text{parametr_kursowy_celu} = PK,$
 $\text{maksymalny_parametr_kursowy_środką_ogniowego} = B,$
 $\text{rozrzut_parametru_kursowego} = P,$
 $X := P / B,$
 $A := B - P,$
 $PK < A,$
 $P > B;$
- 1006: $\text{prawdopodobieństwo_wejścia_celu_w_strefę_rażenia_środką} = Y$ if
 $\text{parametr_kursowy_celu} = PK,$
 $\text{maksymalny_parametr_kursowy_środką_ogniowego} = B,$
 $\text{rozrzut_parametru_kursowego} = P,$
 $A := B - PK,$
 $D := 2 * P,$
 $C := A / D,$
 $S := P + B,$
 $Q := B - P,$
 $R := \text{abs}(Q),$
 $PK \leq S,$
 $PK \geq R,$
 $Y := 0.5 + C;$
- 1010: $\text{rozrzut_parametru_kursowego} = X$ if
 $\text{czas_dolotu_do_dalszej_rubieży_strefy_ognia} = TD,$
 $\text{prędkość_celu} = VC,$
 $\text{współczynnik_kąta_rozrzutu_pk} = ALFA,$
 $A := TD * VC,$
 $Y := A * ALFA,$
 $X := Y / 1000;$
- 1020: $\text{wskaźnik_czasu_dolotu} = X$ if
 $\text{czas_cyklu_strzelania_środką} = TC,$
 $\text{czas_dolotu_do_bliższej_rubieży_wskazywania_celu} = TB,$
 $A := TC * TC,$
 $B := TB * TB,$
 $C := A + B,$
 $X := A / C;$
- 1030: $\text{dalsza_rubież_wskazywania_celów_do_zwalczania} = X$ if
 $\text{dalsza_rubież_strefy_ognia_środką_dla_celu} = DDS,$
 $\text{prędkość_celu} = VC,$
 $\text{prędkość_lotu_rakiety_środką} = VR,$
 $\text{czas_przygotowania_strzelania_środką} = TPS,$
 $A := DDS * 1000,$
 $B := A / VR,$
 $C := B + TPS,$
 $D := C * VC,$
 $E := A + D,$
 $X := E / 1000;$

1040: bliższa_rubież_wskazywania_celów_do_zwalczenia = X if
 bliższa_rubież_strefy_ognia_środka_dla_celu = DBS,
 prędkość_celu = VC,
 prędkość_lotu_rakiety_środka = VR,
 czas_przygotowania_strzelania_środka = TPS,
 A := DBS * 1000,
 B := A / VR,
 C := B + TPS,
 D := C * VC,
 E := A + D,
 X := E / 1000;

1050: czas_dolotu_do_bliższej_rubieży_wskazywania_celu = X if
 k_czas_dolotu = TK,
 bliższa_rubież_czas_dolotu = TBR,
 X := TK - TBR;

1060: k_czas_dolotu = X if
 składowa_X_prędkości_celu = VX,
 składowa_Y_prędkości_celu = VY,
 wsp_X_środka_ogniowego = XS,
 wsp_Y_środka_ogniowego = YS,
 wsp_X_celu = XC,
 wsp_Y_celu = YC,
 XXS := 1000 * XS,
 YYS := 1000 * YS,
 XXC := 1000 * XC,
 YYC := 1000 * YC,
 A := XXS - XXC,
 B := abs(A),
 C := VX * B,
 D := YYS - YYC,
 E := abs(D),
 F := VY * E,
 G := C + F,
 H := VX * VX,
 I := VY * VY,
 J := H + I,
 X := G / J;

1070: bliższa_rubież_czas_dolotu = X if
 bliższa_rubież_wskazywania_celów_do_zwalczenia = BRW,
 prędkość_celu = VC,
 A := 1000 * BRW,
 X := A / VC;

1080: czas_dolotu_do_dalszej_rubieży_strefy_ognia = X if
 k_czas_dolotu = TK,
 dalsza_rubież_czas_dolotu = TDR,
 X := TK - TDR;

1090: dalsza_rubież_czas_dolotu = X if
 dalsza_rubież_strefy_ognia_środka_dla_celu = DDS,
 prędkość_celu = VC,
 A := 1000 * DDS,
 X := A / VC;

```

1100: parametr_kursowy_celu = X if
    x_k = XK,
    y_k = YK,
    wsp_X_środk_ogniowego = XS,
    wsp_Y_środk_ogniowego = YS,
    A := XS - XK,
    B := A * A,
    C := YS - YK,
    D := C * C,
    E := B + D,
    X := sqrt(E);

```

```

1110: x_k = X if
    wsp_X_celu = XC,
    skladowa_X_prędkości_celu = VX,
    k_czas_dolotu = TK,
    A := 1000 * XC,
    B := VX * TK,
    C := A + B,
    X := C / 1000;

```

```

1120: y_k = X if
    wsp_Y_celu = YC,
    skladowa_Y_prędkości_celu = VY,
    k_czas_dolotu = TK,
    A := 1000 * YC,
    B := VY * TK,
    C := A + B,
    X := C / 1000;

```

```
end; // rules
```

```
control
```

```

run;
createAppWindow;
vignette("DECYZJA OGNIOWA", "\nEkperymentalny, laboratoryjny system ekspertowy\n\n\nLICZENIE
WSKAZNIKA EFEKTYWNOŚCI SRODKA OGNIOWEGO\n\n\nAutor: Grzegorz Kott CI AON",
"Copyright ©1998-99 Centrum Informatyki AKADEMII OBRONY NARODOWEJ");
setAppWinTitle("WSKAZNIK EFEKTYWNOŚCI SRODKA OGNIOWEGO");
precision(4,2);

```

```

addFact(_, wsp_X_środk_ogniowego, 0);
addFact(_, wsp_Y_środk_ogniowego, 0);

```

```

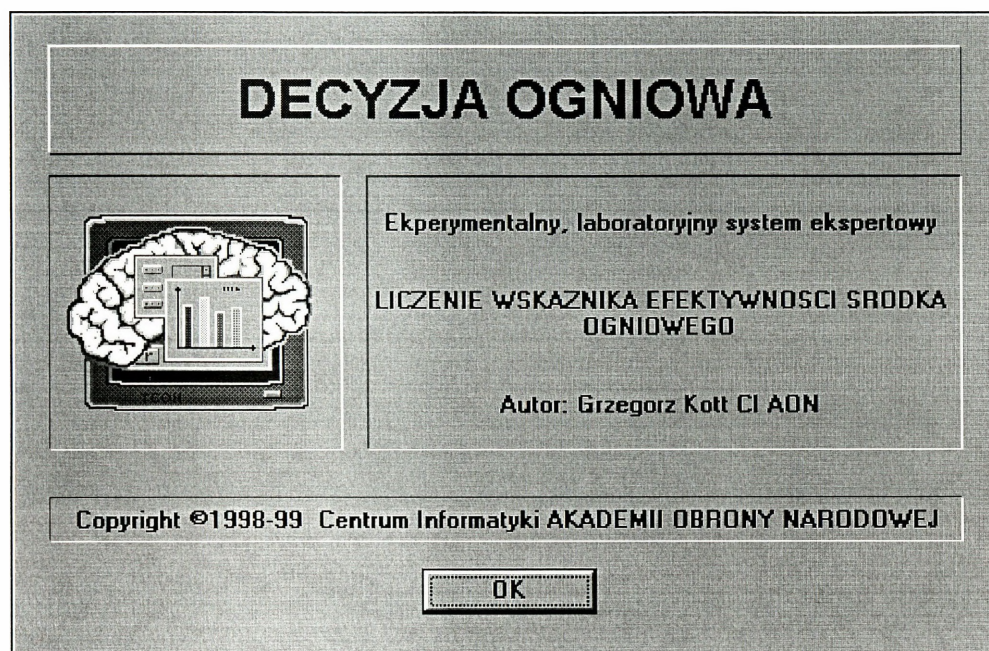
goal("wskaźnik_efektywności_strzelania=X");
// goal("prawdopodobieństwo_wejścia_celu_w_strefę_rażenia_środk = X");
// goal("rozzrut_parametru_kursowego = X");
// goal("wskaźnik_czasu_dolotu = X");
// goal("dalsza_rubież_wskazywania_celów_do_zwalczania = X");
// goal("bliższa_rubież_wskazywania_celów_do_zwalczania = X");
// goal("czas_dolotu_do_bliższej_rubieży_wskazywania_celu = X");

```

```
// goal("k_czas_dolotu = X");  
// goal("blizsza_rubiez_czas_dolotu = X");  
// goal("czas_dolotu_do_dalszej_rubiezy_strefy_ognia = X");  
// goal("parametr_kursowy_celu = X");  
  
    delNewFacts;  
end;  
  
end; // knowledge base
```

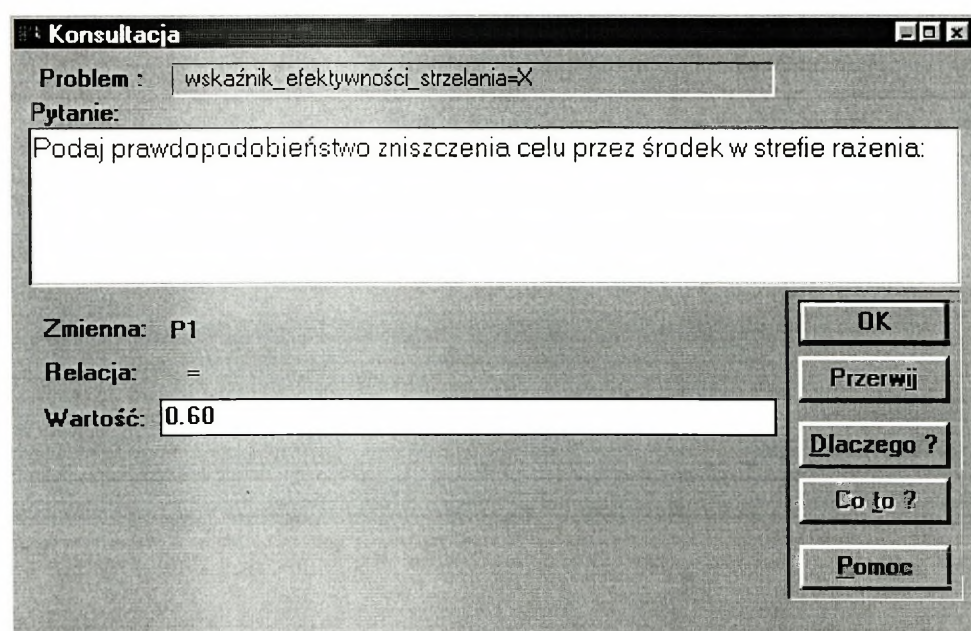
3.2.3. Zasady użytkowania systemu DECYZJA

Uruchomienie systemu DECYZJA następuje z poziomu systemu PC-SHELL poprzez otwarcie bazy wiedzy zapisanej w pliku w_dec_og.bw. Następuje automatyczna retranslacja aplikacji i jej uruchomienie. Gotowość aplikacji do współpracy z użytkownikiem sygnalizowana jest wyświetleniem poniższych komunikatów:

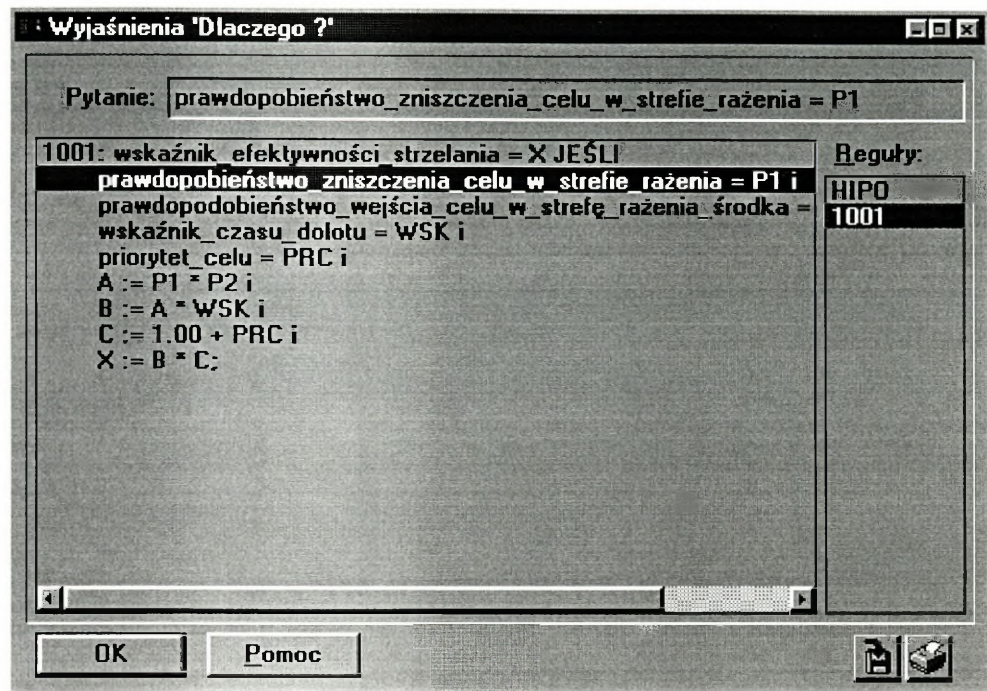


Naciśnięcie przycisku OK powoduje przejście systemu ekspertowego w tryb konsultacji.

System zadaje użytkownikowi pytania, tworząc na podstawie udzielonych odpowiedzi nowe fakty. Fakty te stanowią podstawę wnioskowania.

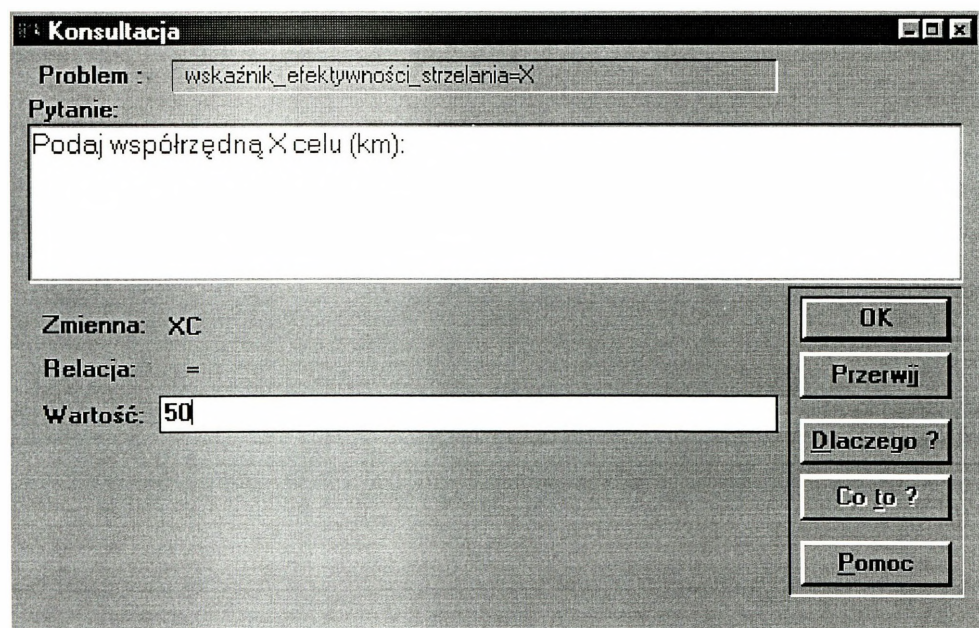


Użytkownik może uzyskać kontekst zadanego pytania, naciskając przycisk Dlaczego?. Efektem będzie wyświetlenie okna wyjaśnień wraz z zawartą w nim regułą, która spowodowała żądanie podania faktu:



Oprócz polecenie podania prawdopodobieństwa zniszczenia celu w strefie rażenia aplikacja zażąda dodatkowo:

- podania współrzędnej X celu:



- podania wartości składowej X prędkości celu:

The screenshot shows a dialog box titled "Konsultacja". At the top, the "Problem" field contains the text "wskaźnik_efektywności_strzelania=X". Below it, the "Pytanie:" label is followed by a large text area containing the question "Podaj wartość składowej X prędkości celu (m/sek):". In the lower-left section, the "Zmienna:" field is set to "VX", the "Relacja:" field is set to "=", and the "Wartość:" field contains the number "300". On the right side of the dialog, there is a vertical stack of five buttons: "OK", "Przerwij", "Dlaczego?", "Co to?", and "Pomoc".

- podania wartości składowej Y prędkości celu:

The screenshot shows a dialog box titled "Konsultacja". At the top, the "Problem" field contains the text "wskaźnik_efektywności_strzelania=X". Below it, the "Pytanie:" label is followed by a large text area containing the question "Podaj wartość składowej Y prędkości celu (m/sek):". In the lower-left section, the "Zmienna:" field is set to "VY", the "Relacja:" field is set to "=", and the "Wartość:" field contains the number "0". On the right side of the dialog, there is a vertical stack of five buttons: "OK", "Przerwij", "Dlaczego?", "Co to?", and "Pomoc".

- podania współrzędnej (umownej) Y celu:

Konsultacja

Problem : wskaźnik_efektywności_strzelania=X

Pytanie:
Podaj współrzędną Y celu (km):

Zmienna: YC
Relacja: =
Wartość: 6

OK
Przerwij
Dlaczego ?
Co to ?
Pomoc

- podania maksymalnego parametru kursowego środka ogniowego:

Konsultacja

Problem : wskaźnik_efektywności_strzelania=X

Pytanie:
Podaj maksymalny parametr kursowy środka ogniowego (km):

Zmienna: B
Relacja: =
Wartość: 100

OK
Przerwij
Dlaczego ?
Co to ?
Pomoc

- określenia dalszej rubieży strefy ognia środka ogniowego:

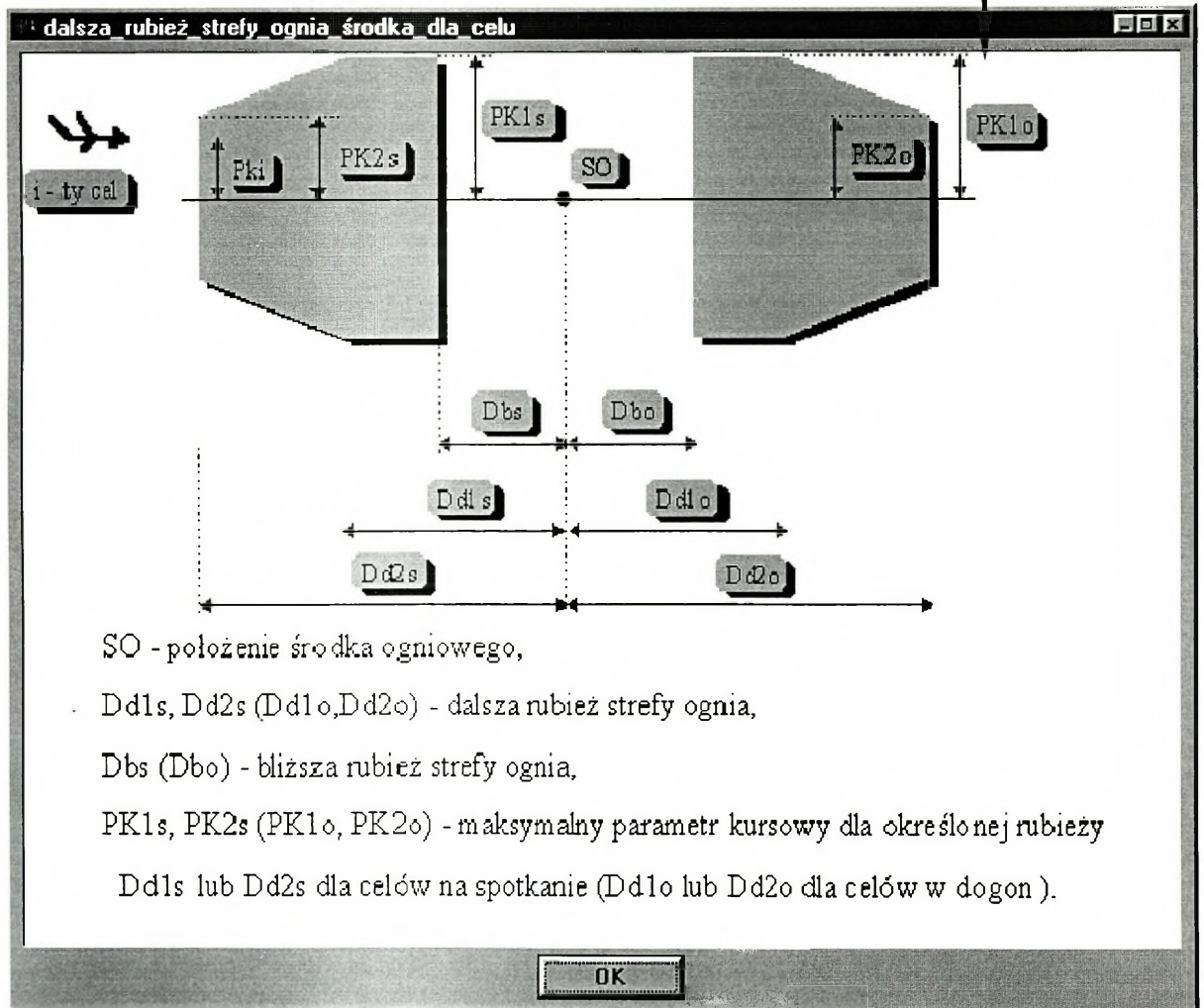
Konsultacja

Problem :

Pytanie:

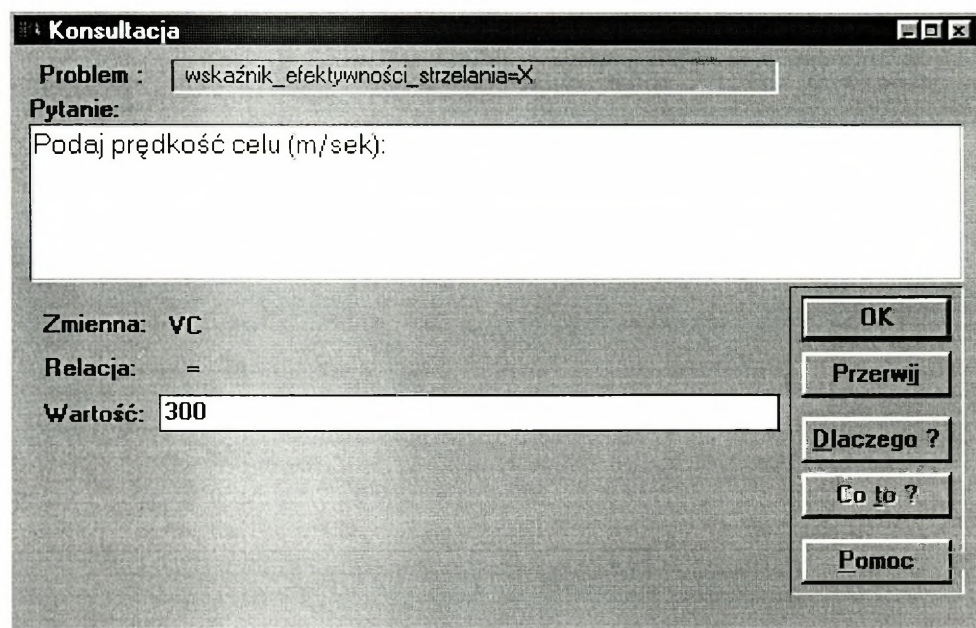
Zmienna: DDS
 Relacja: =
 Wartość:

OK
 Przewij
 Dlaczego?
 Co to?
 Pomoc



Jeżeli w prawym, górnym rogu okna uzgadniania faktu pojawi się ikona obrazu, to z faktem związany jest schemat, wyjaśniający zadane pytanie.

Następnie system uzgadnia kolejne fakty - prędkość celu:



Konsultacja

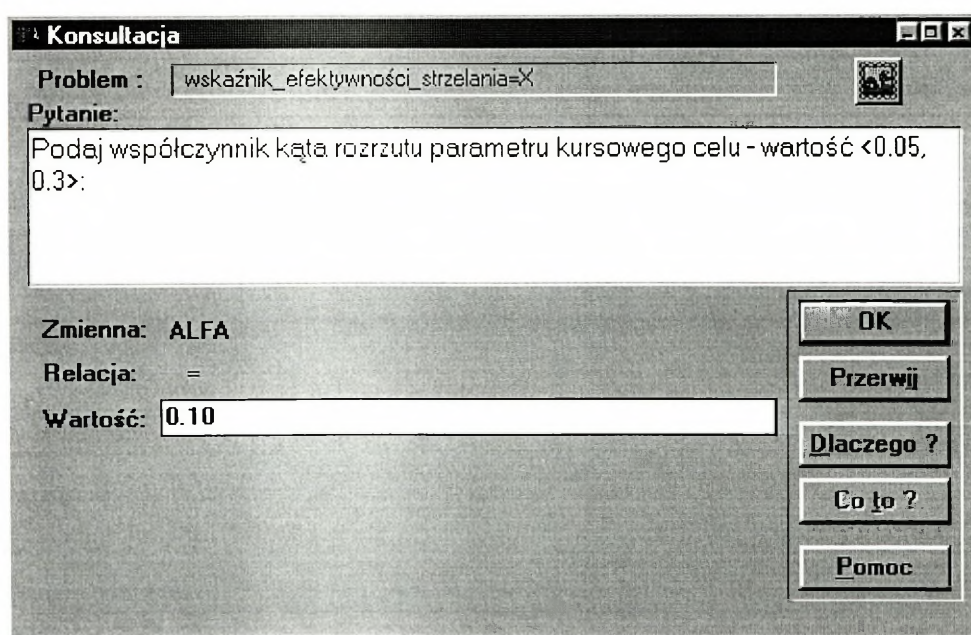
Problem : wskaźnik_efektywności_strzelania=X

Pytanie:
Podaj prędkość celu (m/sek):

Zmienna: VC
Relacja: =
Wartość: 300

OK
Przerwij
Dlaczego ?
Co to ?
Pomoc

- współczynnik kąta rozrzutu parametru kursowego:



Konsultacja

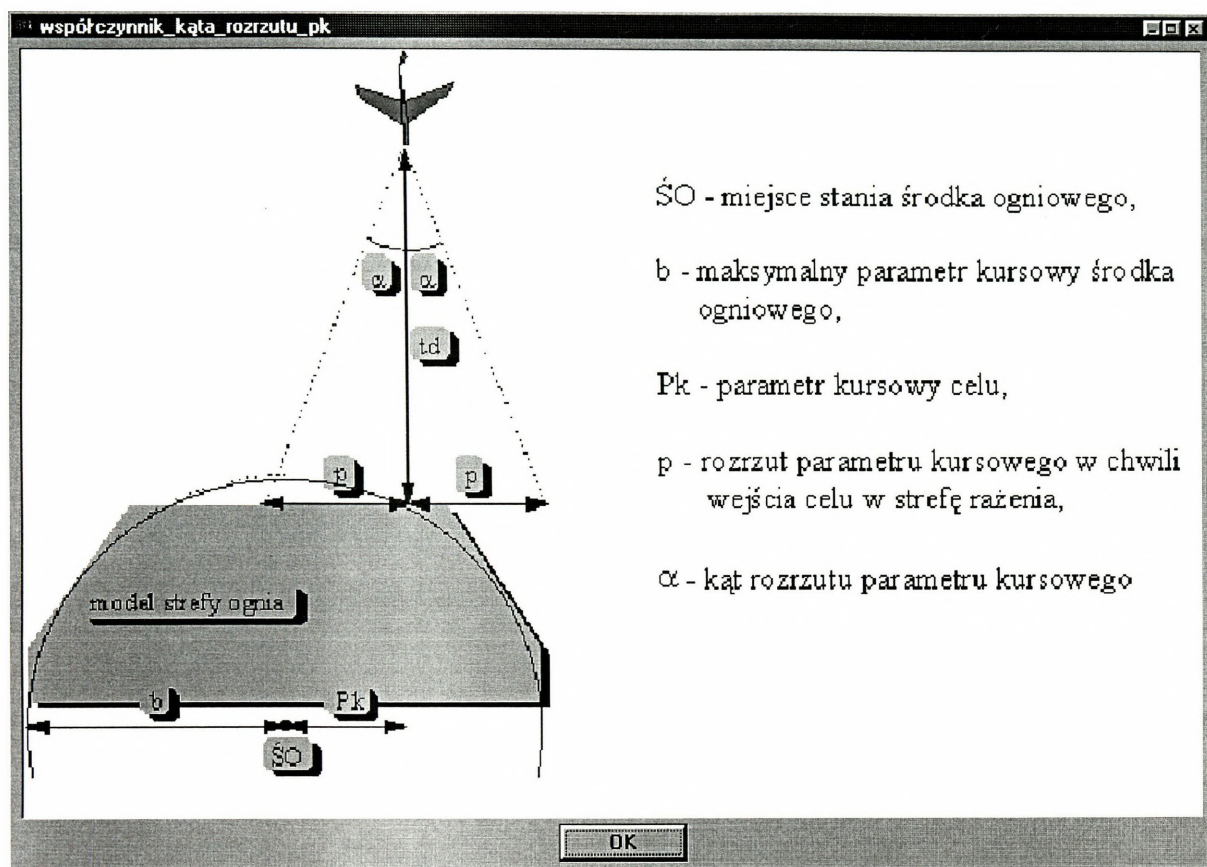
Problem : wskaźnik_efektywności_strzelania=X

Pytanie:
Podaj współczynnik kąta rozrzutu parametru kursowego celu - wartość <0.05, 0.3>:

Zmienna: ALFA
Relacja: =
Wartość: 0.10

OK
Przerwij
Dlaczego ?
Co to ?
Pomoc

z pojęciem którego związany jest poniższy schemat:



- czas cyklu strzelania środka ogniowego:

Konsultacja

Problem : wskaźnik_efektywności_strzelania=X

Pytanie:
 Podaj czas cyklu strzelania środka ogniowego (sek):

Zmienna: TC
 Relacja: =
 Wartość: 55

OK
 Przerwij
 Dlaczego ?
 Co to ?
 Pomoc

- bliższą rubież strefy ognia środka ogniowego:

Konsultacja

Problem : wskaźnik_efektywności_strzelania=X

Pytanie:
Podaj bliższą rubież strefy ognia środka ogniowego dla celu (km):

Zmienna: DBS

Relacja: =

Wartość: 1.50

OK
Przerwij
Dlaczego ?
Co to ?
Pomoc

- prędkość rakiety opisywanego środka ogniowego:

Konsultacja

Problem : wskaźnik_efektywności_strzelania=X

Pytanie:
Podaj prędkość lotu rakiety danego środka ogniowego (m/sek):

Zmienna: VR

Relacja: =

Wartość: 540

OK
Przerwij
Dlaczego ?
Co to ?
Pomoc

- czas przygotowania strzelania środka ogniowego:

Konsultacja

Problem : wskaźnik_efektywności_strzelania=X

Pytanie:
Podaj czas przygotowania strzelania środka ogniowego (sek):

Zmienna: TPS
Relacja: =
Wartość: 70

OK
Przerwij
Dlaczego ?
Co to ?
Pomoc

- oraz wybiera priorytet celu:

Konsultacja

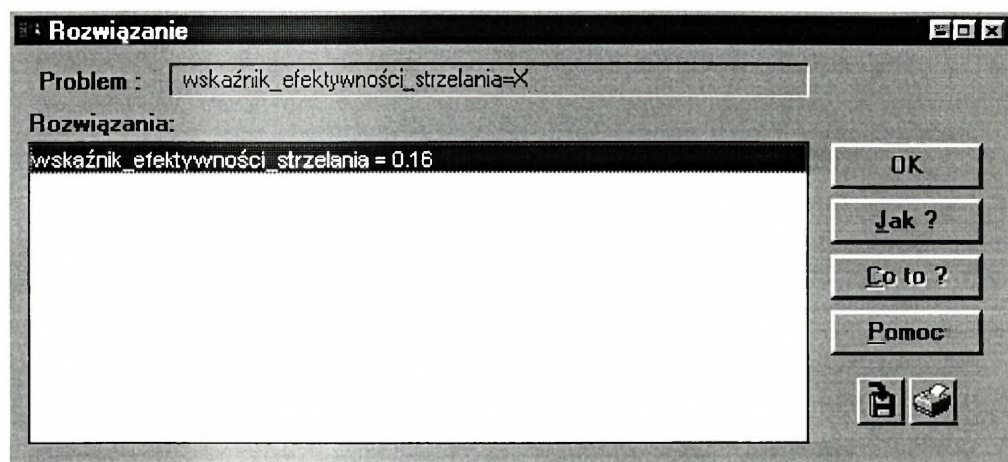
Problem : wskaźnik_efektywności_strzelania=X

Pytanie:
Podaj priorytet celu:

0.000000
1.000000
2.000000
3.000000
4.000000

OK
Przerwij
Dlaczego ?
Co to ?
Pomoc

W efekcie uzyskuje konkluzję - wyliczony wskaźnik efektywności strzelania dla pary środków ogniowych - cel:



Raport z wnioskowania, z racji lokalności generowanych faktów pośrednich, ogranicza się do potwierdzenia uzyskanego wyniku:

RAPORT SYSTEMU PC-SHELL 2.2: ROZWIĄZANIA SPORZĄDZONY: 2.11.1998

Hipoteza : wskaźnik_efektywności_strzelania=X

TAK: hipoteza potwierdzona

Rozwiązanie nr 1 : wskaźnik_efektywności_strzelania = 0.16

Koniec listy rozwiązań

Poniżej przedstawiono wyniki pośrednie uzgadniania reguł, uzyskane podczas wnioskowania:

wskaźnik_efektywności_strzelania :	aij	0,16
prawdopodobieństwo_zniszczenia_celu_w_strefie_rażenia :	P1	0,60
prawdopodobieństwo_wejścia_celu_w_strefę_rażenia_środką :	P2	0,48
wskaźnik_czasu_dolotu :	WSK	0,28
priorytet_celu :	PRC	1,00
prawdopodobieństwo_wejścia_celu_w_strefę_rażenia_środką :	P2	0,48
parametr_kursowy_celu :	Pk	100,18
rozrzut_parametru_kursowego :	p.	4,00
maksymalny_parametr_kursowy_środką_ogniowego :	b	100,00
wskaźnik_czasu_dolotu :	WSK	0,28
czas_cyklu_strzelania_środką :	TC	55,00
czas_dolotu_do_bliższej_rubieży_wskazywania_celu :	tb	88,89

dalsza_rubież_wskazywania_celów_do_zwalczania :	drw	36,56
dalsza_rubież_strefy_ognia_środka_dla_celu :	Dds	10,00
prędkość_celu :	Vc	300,00
prędkość_lotu_rakiety_środka :	Vr	540,00
czas_przygotowania_strzelania_środka :	Tps	70,00
blizsza_rubież_wskazywania_celów_do_zwalczania :	brw	23,33
blizsza_rubież_strefy_ognia_środka_dla_celu :	DbS	1,50
prędkość_celu :	Vc	300,00
prędkość_lotu_rakiety_środka :	Vr	540,00
czas_przygotowania_strzelania_środka :	Tps	70,00
rozrzut_parametru_kursowego :	p.	4,00
czas_dolotu_do_dalszej_rubieży_strefy_ognia :	td	133,33
współczynnik_kąta_rozrzutu_pk :	ALFA	0,10
prędkość_celu :	Vc	300,00
czas_dolotu_do_bliższej_rubieży_wskazywania_celu :	tb	88,89
k_czas_dolotu :	tk	166,67
składowa_X_prędkości_celu :	Vx	300,00
składowa_Y_prędkości_celu :	Vy	0,00
wsp_X_środka_ogniowego :	Xs	0,00
wsp_Y_środka_ogniowego :	Ys	0,00
wsp_X_celu :	Xc	50,00
wsp_Y_celu :	Yc	6,00
blizsza_rubiez_czas_dolotu :	tbr	77,78
blizsza_rubież_wskazywania_celów_do_zwalczania :	brw	23,33
prędkość_celu :	Vc	300,00
czas_dolotu_do_dalszej_rubieży_strefy_ognia :	td	133,33
k_czas_dolotu :	tk	166,67
składowa_X_prędkości_celu :	Vx	300,00
składowa_Y_prędkości_celu :	Vy	0,00
wsp_X_środka_ogniowego :	Xs	0,00
wsp_Y_środka_ogniowego :	Ys	0,00
wsp_X_celu :	Xc	50,00
wsp_Y_celu :	Yc	6,00
dalsza_rubiez_czas_dolotu :	tdr	33,33
dalsza_rubież_strefy_ognia_środka_dla_celu :	Dds	10,00
prędkość_celu :	Vc	300,00
parametr_kursowy_celu :	Pk	100,18
X_k :	Xk	100,00
Y_k :	Yk	6,00
k_czas_dolotu :	tk	166,67
składowa_X_prędkości_celu :	Vx	300,00
składowa_Y_prędkości_celu :	Vy	0,00
wsp_X_celu :	Xc	50,00
wsp_Y_celu :	Yc	6,00
wsp_X_środka_ogniowego :	Xs	0,00
wsp_Y_środka_ogniowego :	Ys	0,00

4. Bibliografia

- [1] Flanek Cz.: Wspomaganie dowodzenia w systemie OPL wojsk operacyjnych. Rozprawa habilitacyjna, AON 1991
- [2] Flanek C., Kuriata R., Gadzała J., Kott G. Model ekspertowego systemu wspomagania decyzji dowódcy prplot i szefa OPL ZT. AON 1995
- [3] Kacprzyk D.: Definicja danych wejściowych i wyników procesu wypracowania decyzji ogniowej pplot. Praca studyjna, AON 1998
- [4] Kott G.: Model ekspertowego systemu wspomagania decyzji szefa obrony przeciwlotniczej związku taktycznego (szefa opl zt) i dowódcy pułku przeciwlotniczego (d-cy pplot). Część II Specyfikacja funkcjonalna. AON, 1997
- [5] Michalik K.: PC-Shell 2.1 Podręcznik użytkownika. AITECH, Katowice 1997
- [6] Michalik K.: PC-Shell 2.1 Przewodnik inżyniera wiedzy. AITECH, Katowice 1997
- [7] Mulawka J.J.: Systemy ekspertowe. WNT, Warszawa 1997
- [8] Owen G.: Teoria gier. PWN, Warszawa 1975
- [9] Rutkowska D., Piliński M., Rutkowski L.: Sieci neuronowe, algorytmy genetyczne i systemy rozmyte. PWN, Warszawa 1997

