

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. Nr 2

MODEL EKSPERTOWEGO SYSTEMU WSPOMAGANIA DECYZJI DOWÓDCY ZWIĄZKU TAKTYCZNEGO

Projekt koncepcyjny

~~Biblioteka Główna
Akademii Obrony Narodowej
S/3940
05-003940-002-0~~

WARSZAWA

68927



AKADEMIA OBRONY NARODOWEJ

CENTRUM INFORMATYKI



MODEL EKSPERTOWEGO SYSTEMU WSPOMAGANIA DECYZJI DOWÓDCY ZWIĄZKU TAKTYCZNEGO

Projekt koncepcyjny systemu

EKSPERT -Z
6.5.0.0



WARSZAWA

1998

Praca została wykonana zgodnie z „Planem prac naukowo-badawczych na lata dziewięćdziesiąte” (6.5.0.0. - EKSPERT). Niniejsze opracowanie stanowi kontynuację prowadzonych prac nad rozwiązaniem w/w problemu. Całość tematu może być zakończona w 1999 roku. Przewiduje się, że powstanie eksperymentalny, laboratoryjny system ekspertowy wspomagający wybrane funkcje procesu decyzyjnego dowódcy związku taktycznego.

Potencjalne obszary zastosowania wyników pracy to:

- proces dydaktyczny na Wydziale Wojsk Lądowych AON;
- ćwiczenia dowódczo-sztabowe organizowanie przez Dowództwo Wojsk Lądowych.

Pracę naukowo-badawczą EKSPERT wykonał zespół pracowników naukowo-badawczych Centrum Informatyki AON.

Kierownictwo naukowe tematu - płk dr hab. inż. Czesław FLANEK -

współudział w opracowaniu rozdziału 4 i 5.

Płk dr inż. Ryszard WIELEBA - kierownik tematu - opracował rozdziały 2, 3,

współudział w rozdziałach 4, 5.

Płk dr inż. Maciej RATAJCZAK - współudział w opracowaniu rozdziału 3 i 4

Wstęp oraz Wnioski - opracował zespół.

Wprowadzenie

Temat pt. „Model ekspertowego systemu wspomagania decyzji dowódcy związku taktycznego” został wykonany zgodnie z „Planem prac naukowo-badawczych na lata dziewięćdziesiąte” (6.5.0.0. - EKSPERT). Niniejsze opracowanie stanowi kontynuację prowadzonych prac nad rozwiązaniem w/w problemu.

W części pierwszej (wydanej w 1997 roku) dokonano specyfikacji funkcjonalnej i technicznej poprzez:

- przedstawienie problematyki systemów ekspertowych,
- wyróżnienie głównych procesów (funkcji) realizowanych przez system,
- zdefiniowanie procesów decyzyjnych realizowanych na szczeblu ZT,
- przedstawienie ogólnej koncepcji proponowanego rozwiązania (ekspertowego wspomagania dowódcę ZT).

Część druga (projekt koncepcyjny) jest kontynuacją prac podjętych w części pierwszej.

Ze względu na złożoność problemu oraz trudności związane z permanentną reorganizacją SZ RP, jak również dostosowywanie procesu dowodzenia do procedur stosowanych i obowiązujących w NATO, całość tematu może być zakończona w 1999 roku. Przewiduje się, że powstanie eksperymentalny, laboratoryjny system ekspertowy wspomagający wybrane funkcje procesu decyzyjnego dowódcy związku taktycznego. Potencjalne obszary zastosowania takiego systemu to:

- proces dydaktyczny na Wydziale Wojsk Lądowych AON;
- ćwiczenia dowódczo-sztabowe organizowanie przez Dowództwo Wojsk Lądowych.

SPIS TREŚCI

1. WSTĘP	4
2. PAKIET SZTUCZNEJ INTELIGENCJI SPHINX 2.2	7
2.1 <i>SZKIELETOWY SYSTEM EKSPERTOWY PC-SHELL</i>	8
2.1.1 <i>Język reprezentacji wiedzy</i>	14
2.1.2 <i>Struktura języka opisu bazy wiedzy</i>	16
2.1.3 <i>Deklaracje źródeł wiedzy</i>	18
2.1.4 <i>Blok faset</i>	22
2.1.5 <i>Blok opisu faktów</i>	29
2.1.6 <i>Blok opisu reguł</i>	30
2.1.7 <i>Blok control</i>	32
2.2 <i>KOMPUTEROWY SYSTEM WSPOMAGANIA INŻYNIERA WIEDZY CAKE</i>	44
2.2.1 <i>Właściwości aplikacji</i>	45
2.2.2 <i>Źródła wiedzy</i>	46
2.2.3 <i>Obiekty bazy wiedzy</i>	47
2.2.4 <i>Fasety atrybutów bazy wiedzy</i>	47
2.2.5 <i>Fakty bazy wiedzy</i>	51
2.2.6 <i>Reguły bazy wiedzy</i>	52
2.2.7 <i>Program aplikacji</i>	54
3. ANALIZA PROCESU DECYZYJNEGO ORAZ IDENTYFIKACJA PROBLEMÓW DECYZYJNYCH	55
3.1 <i>ALGORYTM PRACY DOWÓDCY</i>	55
3.2 <i>PROBLEMY DECYZYJNE W PROCESIE PLANOWANIA DZIAŁAŃ</i>	60
3.3 <i>CZĄSTKOWE PROBLEMY DECYZYJNE PROCESU DOWODZENIA</i>	62
4. OCENA PODATNOŚCI FAZ PROCESU DOWODZENIA NA WSPOMAGANIE KOMPUTEROWE	66
5. KONCEPCJA EKSPERTOWEGO SYSTEMU WSPOMAGANIA DOWÓDCY ZT	85
5.1 <i>WYMAGANIA NA EKSPERTOWY SYSTEM WSPOMAGANIA DECYZJI</i>	85
5.2 <i>KONCEPCJA SYSTEMU EKSPERTOWEGO WSPOMAGANIA DECYZJI DOWÓDCY ZT</i>	93
6. WNIOSKI	102
7. LITERATURA	104

1. Wstęp

Ludzie, którzy zmuszeni są podejmować decyzje w dużym stresie mogą mieć trudności podjęcia właściwej decyzji z powodu niemożności przetworzenia olbrzymiej ilości informacji, w szczególności dotyczącej rozpoznania i identyfikacji rzeczywistej sytuacji. Odpowiednia ocena aktualnej sytuacji odgrywa olbrzymią rolę w procesie dowodzenia w trakcie działań bojowych, gdzie informacje powinny być efektywnie wykorzystane do kierowania środkami walki oraz w procesie podejmowania taktycznych decyzji. Systemy ekspertowe z założenia powinny wspierać podejmowanie decyzji poprzez wspomaganie procesu interpretowania i automatyzacji analizy lub oceny konkretnej sytuacji. Istotą ich funkcjonowania jest przyswajanie aktualnych danych z różnych źródeł, generowanie hipotez co do aktualnej sytuacji, sposobów postępowania w zaistniałej sytuacji i kierowanie ich do operatorów, planistów i innych decydentów.

Wydaje się, że najbardziej obiecującą dziedziną w zakresie wykorzystania technik sztucznej inteligencji w planowaniu działań bojowych jest integracja informacji pochodzących z różnych źródeł oraz mechanizmy interpretacji i doradztwa na różnych poziomach dowodzenia. W systemach sztucznej inteligencji prace skupiają się wokół budowy "naturalnego" języka, systemów ekspertowych i zaawansowanego dialogu człowiek-komputer. Systemy są implementowane dla pięciu kluczowych funkcji dowodzenia na poziomie walki grupowej :

- wymagania na siły uderzeniowe;
- ocena możliwości wykonania zadań;
- symulacja działań;
- planowanie operatywne działań;
- ocena strategii.

Każda z tych funkcji może być wykonywana ze wspomaganie komputerowym. Obok specjalistycznych symulatorów działań bojowych możliwe jest także wspomaganie za pomocą jednej z najbardziej rozwiniętych technik sztucznej inteligencji, czyli z wykorzystaniem systemów ekspertowych. Zarówno pierwsza technika jak i częściowo

druga, dają obiecujące wyniki o czym świadczą rezultaty prac badawczych z zakresu wspomaganie decyzji w działaniach bojowych prowadzonych na Wydziale Cybernetyki WAT, w Centrum Informatyki Sztabu Generalnego WP oraz Zarządzie Dowodzenia Sztabu Generalnego WP w ramach zautomatyzowanego systemu dowodzenia wojskami lądowymi „KOLORADO” oraz CI AON.

Według wielu specjalistów z zakresu sztucznej inteligencji system ekspertowy to specjalizowany system informatyczny, którego główną funkcją jest rozwiązywanie zadań o charakterze decyzyjnym w zakresie jednego lub wielu następujących obszarów: sterowanie procesami, testowanie obiektów technicznych, projektowanie, diagnozowanie, instruktaż, interpretowanie zjawisk, planowanie, prognozowanie, naprawianie obiektów itd. Do rozwiązywania tych zadań wykorzystywane są procedury wnioskowania, które wraz ze zgromadzoną wiedzą stanowią pewien model ekspertyzy. Wiedza systemu ekspertowego składa się z „faktów” oraz „heurystyk”. „Fakty” reprezentują informacje ogólnie dostępne i powszechnie akceptowane przez ekspertów z dziedziny, której ekspertyza dotyczy. „Heurystyki” natomiast charakteryzują proces oceny i rozwiązywania problemu przez konkretnego specjalistę np. jego domysły, przypuszczenia, intuicję, zdroworozsądkowe zasady.

System ekspertowy wspomaganie dowodzenia powinien spełniać bardzo duże wymagania, aby mógł skutecznie wspierać pracę dowódcy ZT. W kolejnych rozdziałach przedstawione zostaną zarówno analiza pracy dowódcy, zadania decyzyjne, których rozwiązywanie należy wspomagać oraz wymagania na system ekspertowy i koncepcję jego tworzenia.

W rozdziale drugim pracy przedstawiono krótką charakterystykę pakietu sztucznej inteligencji Sphinx 2.2, ponieważ przewiduje się że w środowisku tego systemu zostaną zrealizowane główne funkcje modelowego systemu ekspertowego wspomaganie dowódcy ZT.

Rozdział trzeci zawiera wnioski z analizy procesu decyzyjnego dowódcy ZT przedstawione w postaci zidentyfikowanych problemów decyzyjnych w procesie planowania działań oraz cząstkowe problemy decyzyjne procesu dowodzenia.

W rozdziale czwartym przedstawiono również zakres czynności, które powinny być objęte komputerowym wspomaganie planowania i wypracowania decyzji oraz zdefiniowano 9 zadań decyzyjnych podlegających wspomaganie w podejściu eksperckim.

W rozdziale piątym przedstawiono wymagania na ekspertowy system wspomaganie decyzji dowódcy ZT oraz jego ogólną koncepcję w środowisku pakietu sztucznej inteligencji Sphinx 2.2.

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 modułów modelowych systemu ekspertowego wspomagania dowódcy ZT, przewiduje się zastosowanie podsystemów PC-SHELL oraz CAKE, stąd w niniejszym opracowaniu zostanie przedstawiona charakterystyka oraz możliwości projektowe 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 tę 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 wyposażać 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:

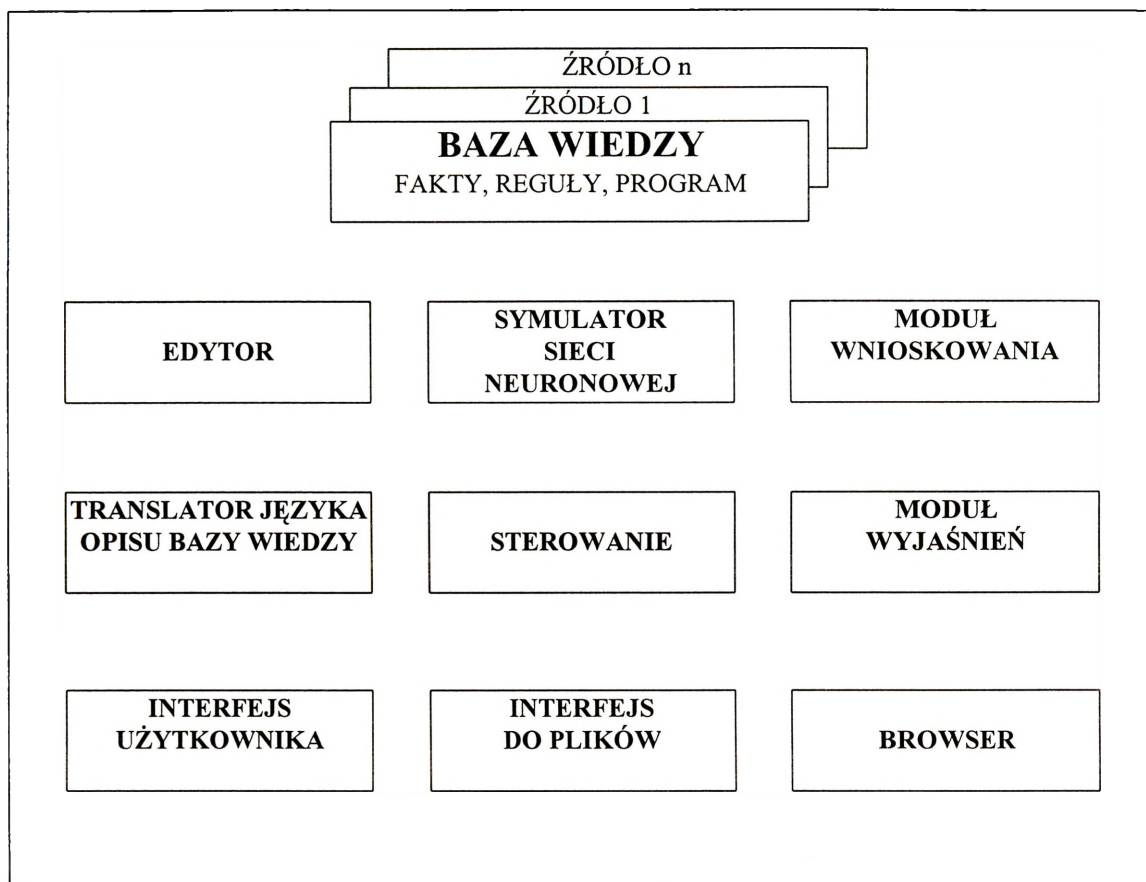
- I. W zakresie struktury systemu:
 - A. elementy architektury tablicowej,
 - B. hybrydowość systemu.

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

- III. W zakresie wyjaśnień:
 - A. wyjaśnienia typu JAK (HOW),
 - B. wyjaśnienia typu DLACZEGO (WHY),
 - C. wyjaśnienia typu CO TO JEST, dla pojęć zawartych w bazie wiedzy (WHAT IS),
 - D. 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 funkcjonalne 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, ujmując 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 wyrażenia 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 jedna 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";
Zmienna3 := 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_źródła_wiedzy
end;
```

Opis_źródła_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 :
    właściwość_1...właściwość_n ;
```

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 następnych 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\\sytfm.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 faseta "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 Fasety 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 fasetą 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 jednoczesne 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 zada pytanie o treści zgodnej z określoną w fasetce, 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,..., zmienna_n = wartość_n }
```

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,..., wartość_n 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 rozdzielenia 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  
    deklaracja_pola,  
    ...  
    deklaracja_pola,  
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);

....

Instrukcje programowania języka SPHINX

Instrukcje dotyczące plików

- close
- fgetc
- fgetn
- fput (+2.15)
- freadFacts
- fsaveExplan
- fsaveSolution
- fseek (2.15)
- open
- readMat
- testEOF

Instrukcje sterujące wykonaniem programu

- break
- continue
- for
- exit
- function (2.2)
- goto
- if
- mainMenu
- menu
- return (2.2)
- run
- quit
- system
- while

Instrukcje inicjujące i sterujące procesem wnioskowania

- goal
- setSysText (2.2)
- solve
- solutionWin

Instrukcje operujące na bazie wiedzy

- addFact
- addFacts
- addSolution
- catchFact (2.15)

delFacts
delNewFacts
freeSource
getSource
saveExplan (2.15)
saveSolution
saveWhatIs (2.15)

Instrukcje do komunikacji z użytkownikiem

confirmBox
messageBox
neditBox
nsheetBox
seditBox
slistBox

Instrukcje konwersji

ascii
c_ntos (2.1)
ftoi (2.1)
makeOAV
ntos
precision
splitOAV
sprintf (2.15)
ston
strcat
strChange (2.15)

Instrukcje sterujące oknami

closeAppWindow
closeWindow (2.1)
createAppWindow
getActiveWindow (2.2)
getWindowType (2.2)
isAppRunning (2.1)
setAppWinTitle
setWindowPos (2.2)
showWindow (2.1)
vignette

Instrukcje symulatora sieci neuronowej

initNetwork (+2.2)
delNetwork (+2.2)

runNetwork (+2.2)

Instrukcje związane z parametryzacją baz wiedzy

changeCategory (2.1)
paramWindow (2.2)
setDefParamValue (2.1)

Instrukcje obsługujące dostęp do baz danych

sqlBind (2.1)
sqlDone (2.1)
sqlFetch (2.1)
sqlInit (2.1)
sqlQuery (2.1)
sqlTransact (2.1)

Instrukcje dostępu do plików .ini

getProfile (2.1)
writeProfile (2.1)

Instrukcje dotyczące dynamicznej wymiany danych

ddeConnect (2.1)
ddeDisconnect (2.1)
ddeExecute (2.1)
ddePoke (2.1)
ddeRequest (2.1)

Instrukcje dostępu do dialogów użytkownika (2.2)

dlgBindButton (2.2)
dlgBindCheckBox (2.2)
dlgBindComboBox (2.2)
dlgBindEdit (2.2)
dlgBindListBox (2.2)
dlgBindRadioButton (2.2)
dlgBindSpreadsheet (2.2)
dlgBindStatic (2.2)
dlgCreate (2.2)
dlgExecute (2.2)
setDialogPos

Instrukcje arkuszowe (2.2)

closeSheet

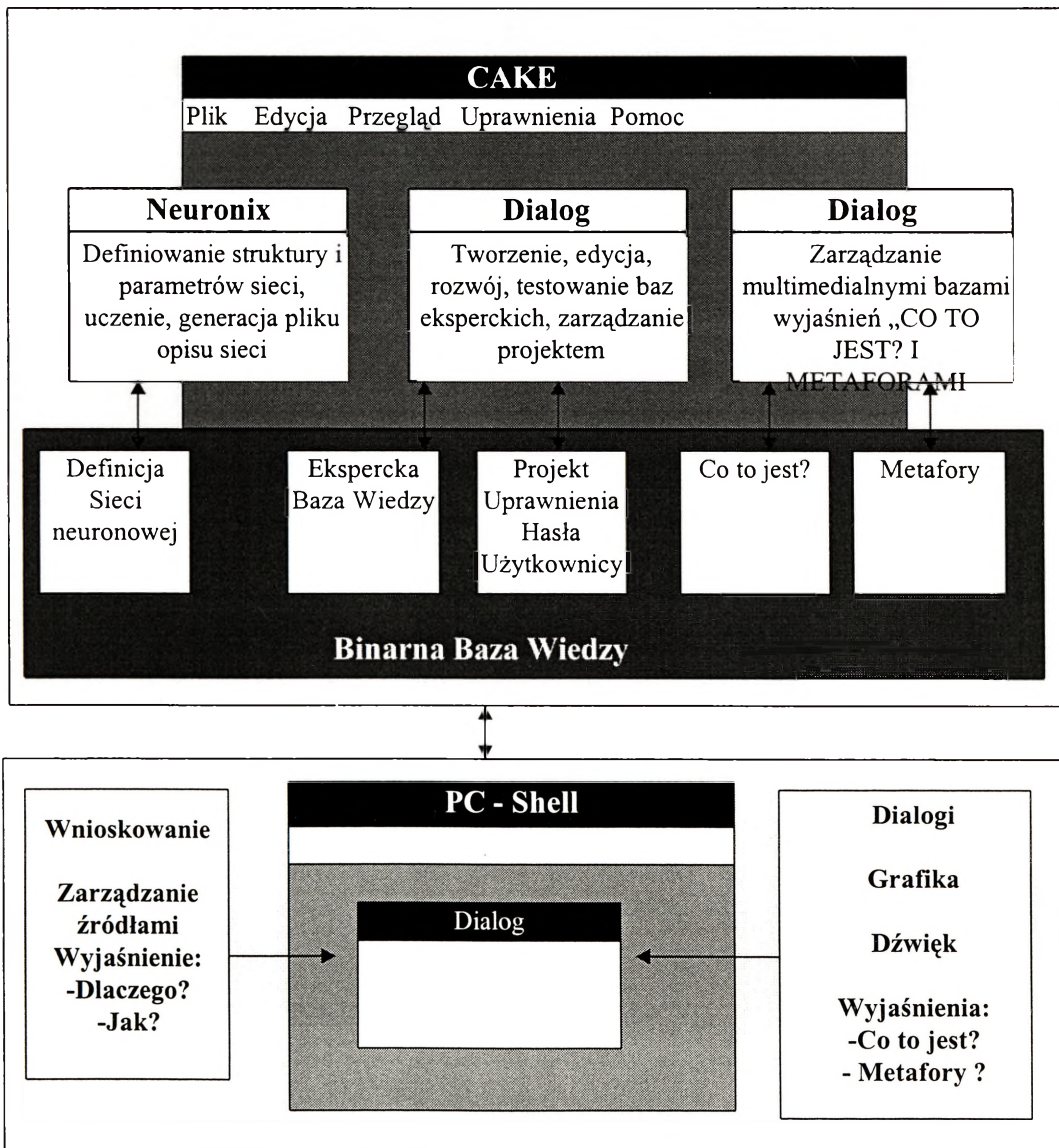
getSheetRange
getSheetValue
openSheet
readSheet
setSheetRange
setSheetValue
setWindowPos
showSheet
writeSheet

Inne instrukcje

arraySize (2.2)
fromClipboard (2.15)
getDate (2.2)
getTime (2.2)
random (2.1)
strLen (2.2)
toClipboard (2.15)
winHelp (2.2)

2.2 Komputerowy system wspomagania 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).

asety 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 żądanego 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 pojawiają 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. ANALIZA PROCESU DECYZYJNEGO ORAZ IDENTYFIKACJA PROBLEMÓW DECYZYJNYCH

3.1 Algorytm pracy dowódcy

Algorytm pracy dowódcy ZT (pionu operacyjnego) podczas przygotowania operacji (walki) będzie więc szeregiem ułożonych w logicznej kolejności czynności, realizowanych przez dowódcę od chwili otrzymania zadania po osiągnięcie gotowości do jego wykonania. Algorytm ten jest stały dla wszystkich szczebli dowodzenia. Różnice wynikające z operacyjnego rozmachu działań dotyczą głównie:

- kalkulacji czasowo-przestrzennych;
- skali problemów, które muszą być rozpatrzone podczas kolejnych etapów pracy.

Nie rzutuje to jednak w sposób zasadniczy na istotę kolejnych przedsięwzięć, które obejmują zwykle :

- analizę zadania;
- ocenę sytuacji;
- określenie zamiaru (koncepcji) działania;
- opracowanie decyzji;
- postawienie zadań;
- organizację współdziałania;
- kontrolę.

Z wymienionych powyżej przedsięwzięć pierwsze cztery wchodzi w zakres **planowania operacji (walki)**.

Analiza zadania jest procesem myślowym realizowanym przez dowódcę, zazwyczaj wspólnie z szefem sztabu. W trakcie analizy zadania dowódca szczegółowo rozpatruje następujące problemy:

- cel działania i zamiar przełożonego;
- istotę zadania, które otrzymał;
- miejsce i rolę podległych mu sił w ugrupowaniu operacyjnym (bojowym);
- zadania sąsiadów, wpływ ich działań na wykonanie własnego zadania, potrzeby w zakresie współdziałania z nimi.

W konsekwencji przeprowadzonej analizy zadania, dowódca powinien wyciągnąć wnioski dotyczące:

- celu przyszłych działań;
- przedsięwzięć wymagających natychmiastowej realizacji;
- potrzeb informacyjnych do określenia zamiaru (koncepcji) operacji (walki);
- organizacji pracy dowództwa.

Zazwyczaj równolegle z prowadzoną analizą zadania szef sztabu dokonuje **kalkulacji czasu** posiadanego na przygotowanie działań. Danymi wyjściowymi są tutaj:

- czas otrzymania zadania;
- termin gotowości do działania;
- inne ustalenia czasowe narzucone przez przełożonego.

Dokonując kalkulacji czasu określa się:

- ogólną ilość czasu posiadaną na przygotowanie operacji (walki);
- czas, który należy przeznaczyć do dyspozycji podwładnych;
- czas określenia zamiaru i opracowania decyzji;
- terminy postawienia zadań bojowych;
- czas na przedsięwzięcia związane z organizowaniem operacji (walki), doprowadzeniem wojsk do gotowości do działania i przygotowaniem terenu.

Po przeprowadzeniu analizy zadania i określeniu w kalkulacji czasu zasadniczych terminów, dowódca, w ramach informowania operacyjnego (taktycznego) zapoznaje najważniejsze osoby funkcyjne dowództwa z zadaniem, wnioskami z jego analizy oraz wydaje wytyczne dotyczące przygotowania danych do jego decyzji, organizacji rozpoznania oraz ewentualnej pracy w terenie. Po wydaniu przez dowódcę wytycznych, w komórkach funkcjonalnych SD sporządzane są **zarządzenia przygotowawcze**, które zawierają informacje dotyczące:

- ogólnego charakteru przyszłych działań;
- przedsięwzięć, które powinny zostać zrealizowane w określonych terminach;
- czasu i miejsca, w którym podwładnym postawione zostaną zadania bojowe;
- oraz inne dane zgodnie z wytycznymi dowódcy.

Zarządzenia te są następnie jak najszybciej wysyłane do poszczególnych elementów ugrupowania operacyjnego (bojowego) w celu umożliwienia rozpoczęcia przygotowywania się do działań na kolejnych szczeblach dowodzenia. W ramach **oceny sytuacji** ocenie podlegają:

- przeciwnik
- wojska własne
- warunki prowadzonych działań (infrastruktura terenu, warunki atmosferyczne, sytuacja elektroniczna i skażeń, inne czynniki mogące mieć wpływ na działania własne i przeciwnika.

W toku **oceny przeciwnika**, biorąc pod uwagę jego:

- skład;
- ukompletowanie;
- położenie wojsk;
- możliwość wykonania uderzeń wszelkimi siłami i środkami;
- system dowodzenia;
- dotychczasowe działania

określa się **prawdopodobny cel** jego przyszłych działań oraz **potencjalne sposoby działania w tych warunkach**.

Podczas **oceny wojsk własnych** rozważeniu podlega:

- skład;
- ukompletowanie;
- możliwości bojowe;
- możliwości zabezpieczenia logistycznego;
- morale wojsk własnych.

Na podstawie wniosków z tej oceny wypracowuje się kilka wariantów działania - możliwych sposobów wykonania zadania. Ich przygotowanie polega na zaprojektowaniu realnych w danej sytuacji sposobów wykonania zadania wraz z kalkulacjami operacyjno-taktycznymi i czasowo-przestrzennymi niezbędnymi do ich uzasadnienia. Autorami są oficerowie komórki planowania pionu operacyjnego. Dowódca może także opracować swój wariant(y) działania lub ograniczyć się do określenia wytycznych, na co zwrócić uwagę w przygotowywanych alternatywnych sposobach.

Wykreowane warianty zostają następnie skonfrontowane z potencjalnymi sposobami działania strony przeciwnej (których autorami są oficerowie komórki rozpoznania i WRE). W ten sposób, metodą symulacji rozwoju sytuacji, ujawnia się silne i słabe strony każdego wariantu. Rozwiązaniem idealnym jest skonfrontowanie każdego przygotowanego sposobu działania wojsk własnych z każdym potencjalnym działaniem

przeciwnika. Jeżeli jednak z braku czasu jest to niemożliwe, dowódca powinien określić, którą możliwość działania przeciwnika należy wziąć pod uwagę podczas konfrontacji. Po określeniu pozytywów i negatywów wariantów porównuje się je ze sobą, co pozwala na ustalenie optymalnego w danej sytuacji rozwiązania. Będzie to jednak tylko propozycja dla dowódcy, w żadnym wypadku nie naruszająca jego swobody wyboru, tym bardziej, że dowódca może mieć do porównania także swój, osobiście wypracowany wariant działania.

Określając zamiar dowódca zapoznaje się z przygotowanymi wariantami oraz ich uzasadnieniem i **dokonuje wyboru** najlepszego jego zdaniem sposobu działania, który ogłasza następnie jako swój zamiar (koncepcję). **W zamiarze dowódca określa:**

- jednoznaczny cel przyszłych działań;
- sposób wykonania zadania;
- sposób (miejsce) skupienia głównego wysiłku;
- ugrupowanie operacyjne (bojowe).

Tak sprecyzowany zamiar jest podstawą do dalszej pracy całego dowództwa podczas kontynuacji planowania. Od tego momentu działalność wszystkich komórek funkcjonalnych skupia się na zaplanowaniu działania każdego elementu ugrupowania operacyjnego (bojowego) pod kątem realizacji zamiaru dowódcy.

Po zapoznaniu z zamiarem zasadniczych osób funkcyjnych dowództwa, dowódca może wydać dodatkowe wytyczne dotyczące sposobu wykorzystania sił i środków rozpoznania i WRE, rodzajów wojsk, zabezpieczenia logistycznego, itp. dla umożliwienia skutecznej realizacji zamiaru.

W tym czasie komórka operacyjna sporządzić może, na podstawie zamiaru dowódcy **wstępne zarządzenie bojowe**. Celem jego sporządzania jest przyspieszenie rozpoczęcia procesu przygotowania walki na kolejnych szczeblach dowodzenia. Zarządzenie to może obejmować:

- wnioski z oceny przeciwnika;
- zadanie do wykonania;
- zadania realizowane na korzyść podwładnego przez przełożonego oraz zadania sąsiadów;
- czas gotowości do działania.

Następnie, **w ramach etapu opracowania decyzji** dowódca lub wyznaczona przez niego osoba funkcyjna (np. szef sztabu) wysłuchuje propozycji użycia sił i środków

rozpoznania, rodzajów wojsk, realizacji przedsięwzięć zabezpieczenia logistycznego, przygotowanych przez właściwe komórki funkcjonalne stanowiska dowodzenia. Na tym etapie sporządza się także zasadnicze dokumenty dowodzenia:

- plan operacji (walki) - komórka operacyjna;
- plany użycia rodzajów wojsk - komórka wsparcia ogniowego;
- plan rozpoznania i WRE - komórka rozpoznania i WRE;
- plan zabezpieczenia logistycznego - komórka logistyczna;
- rozkazy operacyjne (bojowe) - komórka operacyjna;
- zarządzenia rodzajów wojsk, zabezpieczenia logistycznego i inne - komórki wsparcia ogniowego i zabezpieczenia logistycznego.

Sporządzenie dokumentów dowodzenia w praktyce kończy okres planowania operacji (walki) na określony etap przyszłego działania. Następne etapy pracy wchodzą w zakres organizowania działań (doprowadzania zadań do wykonawców) oraz kontrolowania prawidłowości realizacji nakazanych przedsięwzięć.

Kolejny etap pracy - **postawienie zadań** - polega na przekazaniu podwładnym:

- co i w jakim celu;
- gdzie;
- kiedy należy wykonać.

Może on przyjąć różną formę. Zadania mogą być:

- postawione podwładnym ustnie po ich wezwaniu na SD;
- podobnie jak wyżej lecz na SD podwładnych;
- przesłane wykonawcom przez techniczne środki łączności;
- dostarczone w imieniu dowódcy przez oficerów kierunkowych (łącznikowych);
- postawione podwładnym podczas pracy w terenie przez dowódcę lub upoważnione do tego osoby.

Opisane powyżej etapy pracy dowódcy zawierają się w **procesie decyzyjnym**, który z punktu widzenia teorii dowodzenia obejmuje zwykle trzy fazy:

1. Przygotowanie (faza przeddecyzyjna), obejmujące:

- ◆ stworzenie (powstanie) sytuacji decyzyjnej;
- ◆ zbieranie i przetwarzanie informacji;
- ◆ kreację wariantów działania i określenie przewidywalnych ich skutków.

2. Wybór (faza podjęcia decyzji), obejmujący:

- ◆ sprecyzowanie kryteriów oceny;
- ◆ wartościowanie, czyli krytyczną ocenę wariantów według sprecyzowanych poprzednio kryteriów;
- ◆ decyzję, to znaczy wybór jednego z wariantów;

3. Dyspozycja (faza podecyzyjna), sprowadzająca się do :

- ◆ sprecyzowania i postawienia zadań;
- ◆ kontroli realizacji i wyników wprowadzenia decyzji w życie.

Ze względu na obszar zainteresowań, zatem i rozważań w pracy dotyczący przede wszystkim planowania walki, szczególnemu zainteresowaniu podlegać będą dalej zwłaszcza dwie pierwsze z wymienionych powyżej faz procesu decyzyjnego.

3.2 Problemy decyzyjne w procesie planowania działań

W trakcie realizacji procesu planowania pojawia się szereg problemów decyzyjnych zmuszających dowódcę do podjęcia decyzji. Problemy te mają zwykle charakter mieszany (to znaczy zarówno postulacyjny, optymalizacyjny jak i realizacyjny) co w znacznym stopniu rzutuje na sposób ich rozwiązywania. Zasadniczo typową dla struktury wojskowej formą decydowania jest jednoosobowość, która jest także jedną z zasad dowodzenia, co nie wyklucza jednak poważnego udziału szeregu osób i komórek dowództwa w całokształcie procesu podejmowania decyzji. Wręcz przeciwnie, udział tych ostatnich jest bardzo szeroki, szczególnie podczas rozwiązywania problemów (części problemów) realizacyjnych. W zależności od stylu pracy przyjętego przez dowódcę, może on podejmować osobiście wszystkie decyzje cząstkowe lub ograniczyć się do rozwiązania głównego problemu decyzyjnego, zwykle o charakterze postulacyjno-optymalizacyjnym, - czyli podjęcia zamiaru. Nie zmienia to jednak istoty rozpatrywanego zagadnienia.

Dowódca staje przed problemem (sytuacją) decyzyjnym w momencie otrzymania od przełożonego zadania do wykonania lub w przypadku takich zmian w rozwoju sytuacji, które w sposób zasadniczy zmieniają położenie w jakim się on (podległe mu siły) znajdują. Stanowi to nic innego jak stwierdzenie sytuacji decyzyjnej, a

w konsekwencji powoduje sprecyzowanie zakresu i czasu podjęcia decyzji. Etapem pracy dowódcy, w którym wymienione wydarzenia mają miejsce jest etap analizy zadania.

Etapem planowania, w którym między innymi te czynności są realizowane jest etap oceny sytuacji. Na tym etapie rozważa się przede wszystkim problemy typu postulacyjnego i optymalizującego. Głównymi problemami, które dowódca (dowództwo) musi rozwiązać są:

- jednoznaczne określenie celu przyszłego działania;
- wytworzenie wariantów działania, to znaczy ustalenie:
 - ◆ jak postępować (*jak nacierać, bronić się, przegrupowywać itp.*);
 - ◆ jak (gdzie) skupić główny wysiłek swoich działań (*na jakim kierunku, w jakim rejonie, na jakim działaniu, na działaniu jakiego elementu ugrupowania operacyjnego /bojowego/*);
 - ◆ jak ugrupować posiadane siły (*w jeden rzut, dwa rzuty ...*).

Kolejny etap - sprecyzowanie zamiaru, jest dla dowódcy zdecydowanie najważniejszy. Jak wspomniano polega on na dokonaniu wyboru pomiędzy dostępnymi wariantami działania - co oznacza, że jest on faktycznie podjęciem decyzji - czyli stwierdzeniem z pewnym stopniem ogólności *co, jak, kiedy i gdzie* ma zrealizować.

Jednakże, postać zamiaru określa jedynie ogólny zamysł przyszłego działania. Oznacza to dodatkową konieczność rozwiązania szeregu problemów cząstkowych o charakterze realizacyjnym, określających znacznie bardziej szczegółowo *kto i co* ma wykonać aby zrealizować myśl dowódcy wyrażoną w zamiarze. Problemy te rozwiązywane są w etapie opracowywania decyzji przez poszczególne komórki funkcjonalne SD.

W praktyce dowodzenia stosowanej w Siłach Zbrojnych RP propozycje rozwiązania tych cząstkowych problemów przedstawiane są dowódcy do zaakceptowania i dopiero po tym fakcie stają się częścią jego decyzji. Jest oczywiście dyskusyjne, czy ta formalna akceptacja jest konieczna, jeżeli wspomniane problemy rozwiązywane są przez specjalistów zgodnie z myślą przewodnią dowódcy zawartą w jego zamiarze. Trudno też wierzyć, że dowódca będzie przewyższał wiedzą każdego specjalistę, szczególnie w jego, dość wąskim, obszarze odpowiedzialności. Tak też jest zjawisko to rozpatrywane w armiach NATO - **dowódca podaje ogólny zarys rozwiązania problemu**, (podejmuje decyzję), specjaliści zajmują się szczegółami i formalna akceptacja dowódcy nie jest

niezbędna, zaś dowódca przyjmuje odpowiedzialność za całokształt działalności poprzez złożenie podpisu na rozkazie operacyjnym.

3.3 Częstkowe problemy decyzyjne procesu dowodzenia

Częstkowe problemy decyzyjne w procesie dowodzenia wymagają :

- precyzyjnych zadań dla ogólnowojskowych elementów ugrupowania operacyjnego (bojowego):
 - ◆ ich rozmieszczenia, stanu, stopnia ukończenia i gotowości bojowej,
 - ◆ realnych możliwości bojowych;
 - ◆ ugrupowania bojowego;
 - ◆ zadań bojowych;
- sposobu wykorzystania WRiA:
 - ◆ zadań realizowanych przez siły WRiA przełożonego i sąsiadów;
 - ◆ obiektów uderzeń WRiA ze wskazaniem określonych środków ogniowych;
 - ◆ pasów bezpieczeństwa;
 - ◆ stanu, możliwości ogniowych i podziału WRiA;
 - ◆ przydziału i podziału środków materiałowych WRiA;
 - ◆ norm zużycia środków materiałowych i organizacji ich dowozu;
 - ◆ zadań WRiA w poszczególnych fazach operacji (walki);
- sposobu wykorzystania WOPL:
 - ◆ możliwości oddziaływania środków napadu powietrznego przeciwnika;
 - ◆ organizacji rozpoznania środków napadu powietrznego;
 - ◆ składu, gotowości i możliwości bojowych sił WOPL;
 - ◆ zadań w zakresie WOPL realizowanych przez przełożonego i sąsiadów;
 - ◆ własnych zadań w zakresie WOPL;
 - ◆ sposobów zwalczania desantów;
 - ◆ organizacji powiadamiania i działania wojsk podczas odpierania uderzeń środków napadu powietrznego;
 - ◆ zabezpieczenia materiałowego sił WOPL;

- ◆ terminu gotowości systemu WOPL;
- sposobu wykorzystania Wojsk Inżynieryjnych:
 - ◆ zadań zabezpieczenia inżynieryjnego;
 - ◆ terminów ich wykonania;
 - ◆ zadań realizowanych przez przełożonego w tym zakresie;
 - ◆ stanu etatowych sił i środków inżynieryjnych;
 - ◆ możliwości wykonania zadań posiadanymi siłami i środkami;
 - ◆ wykorzystania przydzielonych sił i środków inżynieryjnych;
 - ◆ podział sił i środków inżynieryjnych;
- sposobu wykorzystania Wojsk Obrony Przeciwchemicznej:
 - ◆ możliwości użycia przez przeciwnika BMR;
 - ◆ wpływu skażeń na działania wojsk;
 - ◆ elementu (elementów) ugrupowania operacyjnego (bojowego) lub rejonu, na osłonie którego należy skupić główny wysiłek;
 - ◆ przedsięwzięć ochronno-obronnych, które muszą być wykonane w wojskach;
 - ◆ stanu i możliwości sił obrony przeciwchemicznej;
 - ◆ sposobu użycia sił obrony przeciwchemicznej;
 - ◆ zadań realizowanych przez przełożonego;
 - ◆ sposobów prowadzenia rozpoznania skażeń;
 - ◆ zadań grup awaryjno-ratowniczych;
 - ◆ składu i rozmieszczenia sił do likwidacji skutków skażeń;
- sposobu wykorzystania sił i środków Rozpoznania i WRE:
 - ◆ celu rozpoznania;
 - ◆ zadań rozpoznania i kolejności ich wykonania;
 - ◆ rodzajów informacji , sposobów i czasu ich pozyskania;
 - ◆ terminu gotowości systemu rozpoznania;
 - ◆ celu WRE;
 - ◆ zadań WRE i kolejności ich wykonania;
 - ◆ terminu gotowości systemu WRE;
- sposobu wykorzystania przydzielonego (jedynie w wyjątkowych wypadkach, w warunkach normalnych na szczeblu ZT nie stosuje się) wysiłku lotnictwa:

- ◆ stanu i miejsc bazowania lotnictwa wydzielonego do działania na korzyść;
- ◆ możliwości bojowych lotnictwa w zakresie zwalczania typowych obiektów na polu walki;
- ◆ podziału przydzielonego wysiłku;
- ◆ udziału lotnictwa w rażeniu ogniowym przeciwnika;
- sposobu i zakresu ewentualnego współdziałania z jednostkami Marynarki Wojennej:
 - ◆ sił MW działających w określonym obszarze morskim;
 - ◆ potrzeb i możliwości współdziałania;
- sposobu realizacji zabezpieczenia logistycznego:
 - ◆ dróg dowozu i ewakuacji;
 - ◆ sposobu rozmieszczenia i przesunięć urządzeń tyłowych;
 - ◆ ilości zapasów materiałowych;
 - ◆ czasu i sposobu zaopatrywania w środki materiałowe;
 - ◆ sił i środków oraz sposobu ochrony i obrony tyłów;
 - ◆ wykorzystania jednostek obrony terytorialnej na korzyść logistyki;
 - ◆ wykorzystania infrastruktury lokalnej na potrzeby logistyki;
- sposobu wykorzystania sił Obrony Terytorialnej (OT):
 - ◆ stanu, możliwości bojowych i rozmieszczenia wojsk OT;
 - ◆ zadań, które mogą realizować;
- sposobu wykorzystania i zakresu współdziałania z siłami i środkami układu pozamilitarnego:
 - ◆ stanu zasobów miejscowych oraz urządzeń infrastruktury;
 - ◆ możliwego zakresu ich wykorzystania;
- sposobu organizacji dowodzenia i łączności:
 - ◆ podziału personelu na poszczególne SD;
 - ◆ miejsc rozmieszczenia i dróg przemieszczenia głównych i zapasowych stanowisk dowodzenia;
 - ◆ czasu pracy poszczególnych SD w określonych położeniach;
 - ◆ czasu i sposobu wykorzystania doraźnie organizowanych stanowisk (punktów) dowodzenia;

- ◆ wyznaczenia zastępców;
 - ◆ stanu i możliwości sił i środków łączności;
 - ◆ zadań łączności;
 - ◆ rozwinięcia i przesunięcia węzłów łączności;
 - ◆ ochrony systemu łączności przed oddziaływaniem przeciwnika;
- sposobu i zakresu organizacji współdziałania:
- ◆ celu współdziałania;
 - ◆ etapów współdziałania;
 - ◆ realizacji zadań przez poszczególne elementy ugrupowania bojowego w poszczególnych etapach.

4. OCENA PODATNOŚCI FAZ PROCESU DOWODZENIA NA WSPOMAGANIE KOMPUSEROWE

Podstawowe wymagania dotyczące zadań komputerowego wspomaganie procesu dowodzenia powinny być ukierunkowane na realizację czynności dowódcy podczas:

- analizy zadania;
- oceny sytuacji;
- określenia zamiaru (koncepcji) walki;
- opracowania decyzji.

W pionie operacyjnym zakres czynności objętych komputerowym wspomaganie planowania i wypracowania decyzji powinien być następujący :

- a) **w ramach analizy zadania** : przedstawienie zadania w formie graficznej na tle aktualnego położenia wojsk, specyfikacja przedsięwzięć do natychmiastowej realizacji, opracowanie propozycji kalkulacji czasu, opracowanie propozycji planu pracy dowództwa (sztabu);
- b) **w ramach oceny sytuacji**: potencjalne sposoby działania przeciwnika w aktualnym składzie, oraz przy istniejących uwarunkowaniach terenowych i pogodowych, możliwości wykonania uderzeń przez przeciwnika (prawdopodobne kierunki, dopuszczalne terminy, użyte siły i ich ugrupowanie), wskazanie newralgicznych punktów w systemie obrony przeciwnika, prognoza ogniowego oddziaływania przeciwnika i jego przewidywanych skutków, prawdopodobna dyslokacja podstawowych elementów systemu dowodzenia przeciwnika, opracowanie zestawienia priorytetowych celów w ugrupowaniu przeciwnika, propozycje organizacji marszu wojsk własnych (ilość dróg, ilość i skład kolumn, tempo marszu), propozycje (warianty) składu i ugrupowania sił własnych do natarcia, propozycje ugrupowań do obrony z uwzględnieniem warunków terenowych i panujących warunków atmosferycznych;
- c) **podczas określenia (wypracowywania) zamiaru (koncepcji) działania**: konfrontacja możliwych wariantów działania wojsk własnych i przeciwnika, zobrazowanie (w formie graficznej lub tabelarycznej) wyników ww. symulacji, zobrazowanie zamiaru dowódcy;
- d) **podczas opracowania decyzji**: zobrazowanie propozycji użycia sił i środków rozpoznania i RWE, wsparcia ogniowego (rodzajów wojsk) i realizacji zabezpieczenia

logistycznego, opracowanie (generowanie) sformalizowanych dokumentów bojowych w formie graficznej i tekstowej.

Z powyższej analizy wynika, że podstawowym problemem decyzyjnym jest decyzja dowódcy ZT, dotycząca użycia sił i środków bojowych, jakimi dysponuje lub chce użyć w nadchodzącej walce. Inne czynności procesu planowania i podejmowania decyzji są z praktycznego punktu widzenia czynnościami towarzyszącymi. Ściśle zależą od sformułowanego zadania dotyczącego działań bojowych lecz nie powinny mieć na istotnego wpływu. W przypadku, gdy mamy do czynienia z niemożliwością realizacji postawionego zadania (wynikałoby to z braku informacji o stanie wojsk i środków) możemy traktować to jako konieczność kolejnego postawienia zadań przez dowódcę.

Wynika stąd, że we wstępnej fazie rozważań nad problemami wspomaganie i podejmowanie decyzji o działaniach bojowych należy zająć się modelowaniem samego planu walki wojsk. Mając ustabilizowany pogląd na to czy proponowany model planowania walki i realizacji samej walki jest poprawny, możemy w kolejnym kroku modelowania zająć się przedstawieniem modelu rozpoznania, wsparcia i logistycznego zabezpieczenia działań bojowych wojsk lądowych. Jest to metoda, jaką preferuje się we wszystkich znanych w literaturze i opisanych przypadkach konstrukcji modelu procesu podejmowania decyzji związanej z działaniami militarnymi.

Opierając się o przedstawioną w rozdziale poprzednim analizę procesu planowania i podejmowania decyzji zakłada się, że pewnym przybliżeniem wspomaganie w systemie eksperckim może być poniższe, zestawienie zadań decyzyjnych. W przedstawionym zestawieniu zwraca się uwagę szczególnie na te elementy, które wydają się mało precyzyjne i ich ustalenie polega na pewnym doświadczeniu i intuicji dowódców oraz ich sztabów. Cząstkowe zadania decyzyjne, które mogą być rozwiązane metodą dokładnych kalkulacji np. poprzez rozwiązanie zadań optymalizacji stanowią również przedmiot dyskusji w podejściu eksperckim ze względu na rozważanie ich reprezentacji w bazie modeli, będącej częścią składową bazy wiedzy. Do zadań decyzyjnych podlegających wspomaganie w podejściu eksperckim można zaliczyć:

1. Postawienie precyzyjnych zadań dla ogólnowojskowych elementów ugrupowania bojowego ZT dotyczących:

- ich rozmieszczenia, stanu, stopnia ukończenia i gotowości bojowej;
- realnych możliwości bojowych;
- ugrupowania bojowego;
- zadań bojowych.

Mając następujące informacje:

- otrzymane zadanie (to, gdzie, kiedy, w jakim celu wykonać);
- o posiadanych wojskach własnych (stopień ukończenia, morale, rozmieszczenie, gotowość bojowa);
- o potencjalnym przeciwniku (ukończenie, położenie, gotowość bojowa);
- o prawdopodobnym charakterze działań;
- ograniczenia terenowe, meteorologiczne, taktyczne i logistyczne

należy wyznaczyć sposób postępowania dla różnych form działań:

- **działania zaczepne:**

- jak nacierać - całością sił / częścią sił;
- jaką częścią sił;
- w całym pasie działania / na wybranym odcinku pass działania;
- na którym odcinku;
- na wybranym kierunku;
- na jakim kierunku;
- na jaką głębokość;
- jaki element przeciwnika rozbić;
- który element przeciwnika rozbić;
- czy wprowadzić do walki drugi rzut;
- kiedy wprowadzić do walki drugi rzut;
- na jakim kierunku i z jakiej rubieży wprowadzić drugi rzut;
- jak odpierać kontrataki przeciwnika:
 - * z miejsca;

- * z marszu;
- czy nacierać od czoła;
- czy nacierać na skrzydło:
 - * na które skrzydło;
 - * czy na obydwie skrzydła;
- **działania obronne:**
 - jak się bronić:
 - * obroną stałą
 - * obroną manewrową;
 - * na której rubieży wybrać przedni skraj obrony;
 - * ile wybrać kolejnych rubieży obrony;
 - * gdzie wybrać kolejne rubieże obrony;
 - * gdzie wybrać rubieże ryglowe;
 - * czy wykonywać zwroty zaczepne;
 - * w jakim kierunku wykonywać zwroty zaczepne;
 - * z jakich rubieży;
 - * jakimi siłami wykonywać zwroty zaczepne;
- **jak się przegrupowywać /przemieszczać:**
 - po których drogach;
 - po ilu drogach;
 - jedną kolumną;
 - wieloma kolumnami;
 - iloma kolumnami;
 - którymi kolumnami na których drogach;
 - jakim rodzajem transportu;
 - na własnych środkach;
 - transportem kolejowym;
 - transportem powietrznym;
 - transportem morskim;
 - transportem kombinowanym;
- **główny wysiłek:**
 - jak (gdzie, na kim, na jakim działaniu) skupić główny wysiłek:

- * na jakim kierunku;
 - * w jakim rejonie;
 - * na którym elemencie ugrupowania;
 - * pierwszym rzucie;
 - * drugim rzucie (odwodzie);
 - * lewym / prawym skrzydle / centrum;
 - * na jakim działaniu;
- **ugrupowanie:**
- jak ugrupować swoje siły:
 - * jeden rzut z odwodem;
 - * w dwa rzuty;
 - * w dwa rzuty z odwodem;
 - * w trzy rzuty;
 - * jak ugrupować siły w kolumnie /kolumnach marszowych.

2. Sposób wykorzystania sił i środków rozpoznania i WRE oraz oddziaływania psychologicznego dotyczący:

- celu rozpoznania;
- zadań rozpoznania i kolejności ich wykonania;
- rodzajów informacji , sposobów i czasu ich pozyskania;
- terminu gotowości systemu rozpoznania;
- celu WRE;
- zadań WRE i kolejności ich wykonania;
- terminu gotowości systemu WRE.

Mając następujące informacje:

- zadanie;
- informacje o przeciwniku otrzymane od przełożonego;
- ilość sił i środków rozpoznania i WRE etatowych, przydzielonych i działających na korzyść;
- aktualne położenie systemu rozpoznania;
- morale wojsk własnych i przypuszczalne przeciwnika (z punktu widzenia działań psychologicznych);

- ograniczenia meteorologiczne, terenowe, taktyczne (zastrzeżone pasy), techniczne (pasma częstotliwości), zasobowe;
- stosunek miejscowej ludności

należy wyznaczyć:

- jakie obiekty rozpoznać;
- ile obiektów rozpoznać;
- jakimi siłami;
- jak ugrupować siły rozpoznania i WRE w poszczególnych etapach działania;
- jak współdziałać z ogólnowojskowymi elementami rozpoznawczymi i elementami rozpoznawczymi rodzajów wojsk;
- na jakie obiekty oddziaływać WRE;
- jakimi środkami;
- jak wykorzystać rozpoznanie lotnicze;
- na jakich działaniach skupić główny wysiłek:
 - utrzymanie (podniesienie) morale wojsk własnych;
 - obniżenie morale przeciwnika.

3. Sposobu wykorzystania WRiA:

- zadań realizowanych przez siły WRiA przełożonego i sąsiadów;
- obiektów uderzeń WRiA ze wskazaniem określonych środków ogniowych;
- pasów bezpieczeństwa;
- stanu, możliwości ogniowych i podziału WRiA;
- przydziału i podziału środków materiałowych WRiA;
- norm zużycia środków materiałowych i organizacji ich dowozu;
- zadań WRiA w poszczególnych fazach operacji i (walki).

Mając informacje:

- zadanie;
- wytyczne od przełożonego pod względem funkcjonalnym;
- zamiar dowódcy;
- ilość posiadanych środków ogniowych;

- ilość przydzielonej amunicji;
- ograniczenia taktyczne, techniczne, meteorologiczne

należy wyznaczyć:

- ile obiektów porazić;
- które obiekty porazić;
- jakimi środkami;
- jakim stopniem porażenia;
- ile ogni wykonać;
- jakie ogni wykonać;
- ile wybrać stanowisk ogniowych / rubieży ogniowych;
- gdzie je wybrać (w stosunku do ugrupowania wojsk własnych);
- w których etapach działań;

obrona:

- podczas podejścia i rozwinięcia przeciwnika;
- podczas walki o przedni skraj obrony /o utrzymanie głównego pasa obrony;
- podczas działań w głębi własnego ugrupowania;
- podczas zabezpieczania zwrotów zaczepnych;

natarcie:

- podczas podejścia i rozwinięcia wojsk;
- podczas przekazywania przedniego skraju obrony przeciwnika;
- podczas działań w głębi jego ugrupowania;
- podczas odpierania zwrotów zaczepnych;
- jakie elementy ugrupowania operacyjnego /bojowego wzmocnić poprzez przydział artylerii;
- jakimi siłami je wzmocnić;
- kiedy (w jakich etapach działania).

4. Sposób wykorzystania przydzielonego wysiłku lotnictwa, dotyczący:

- stanu i miejsc bazowania lotnictwa wydzielonego do działania na korzyść;
- możliwości bojowych lotnictwa w zakresie zwalczania typowych obiektów na polu walki;
- podziału przydzielonego wysiłku;
- udziału lotnictwa w rażeniu ogniowym przeciwnika.

Mając informacje:

- zadanie;
- zamiar;
- ilość przydzielonych samolotów;
- możliwy czas ich wykorzystania;
- czas reakcji na wezwanie z pola walki;
- rodzaj dysponowanych statków powietrznych;
- stopień obezwładnienia obrony przeciwlotniczej przeciwnika;
- kto panuje w powietrzu;
- ilość i rodzaj opłacalnych celów;
- odległość celów od rejonów bazowania;
- możliwości posiadanych statków powietrznych;
- zapasy środków materiałowych w tym specjalistycznych środków rażenia;
- warunki atmosferyczne

należy wyznaczyć:

- na jakie obiekty przeciwnika wykonać uderzenia;
- kiedy - w jakim etapie działań;
- jakim wysiłkiem;
- jakie środki rażenia użyć;
- jak zorganizować współdziałanie z własnymi środkami przeciwlotniczymi i LWŁąd.

5. Sposób wykorzystania wojsk inżynieryjnych dotyczący:

- zadań zabezpieczenia inżynieryjnego;
- terminów ich wykonania;
- zadań realizowanych przez przełożonego w tym zakresie;
- stanu etatowych sił i środków inżynieryjnych;
- możliwości wykonania zadań posiadanymi siłami i środkami;
- wykorzystania przydzielonych sił i środków inżynieryjnych;
- podziału sił i środków inżynieryjnych.

Mając informacje:

- zadanie;
- wytyczne otrzymane w pionie funkcjonalnym;
- zamiar dowódcy;
- stan sił i środków inżynieryjnych:
 - specjalistycznych;
 - w strukturach ogólnowojskowych;
- ocena przeciwnika pod względem inżynieryjnym;
- ilość etatowych i przydzielonych elementów rozpoznania inżynieryjnego;
- możliwości rozpoznawcze tych elementów;
- możliwości (wydajność) maszyn inżynieryjnych;
- warunki terenowe;
- warunki atmosferyczne;
- dane taktyczno-techniczne inżynieryjnych środków rażenia;
- możliwości środków rozminowania;
- ilość środków przeprowowych

należy wyznaczyć:

- jakie zadania rozpoznania inżynieryjnego wykonają inżynieryjne elementy rozpoznawcze a jakie ogólnowojskowe elementy rozpoznawcze;
- jakie rejony trzeba szczególnie rozpoznać pod względem inżynieryjnym;

- jakie ukrycia wykonać;
- ile ukryć wykonać;
- gdzie/ dla kogo wykonać te ukrycia;
- jakimi siłami:
 - specjalistycznymi;
 - ogólnowojskowymi;
- gdzie postawić zapory minowe;
- ile km zapór postawić;
- jakich zapór:
 - przeciwpancernych;
 - przeciwpiechotnych;
 - mieszanych;
- jakimi siłami je postawić;
- jakim sposobem:
 - ręcznym; mechanicznym; narzutowym;
 - kombinowanym;
- gdzie wykonać niszczenia;
- jakie obiekty niszczyć;
- które drogi utrzymywać;
- na jakich odcinkach;
- jakimi siłami;
- gdzie wykonać przejścia w zaporach:
 - własnych;
 - przeciwnika;
- ile przejść;
- jakimi siłami (kim);
- jakim sposobem;
- gdzie zorganizować przeprawy;
- ile przepraw zorganizować;
- jakich przepraw:
 - mostowych;
 - desantowych;

- promowych;
- po dnie;
- jakie obiekty pozorne zbudować;
- ile obiektów;
- gdzie;
- jakimi środkami:
 - specjalistycznymi;
 - ogólnowojskowymi;
- którymi ;
- jakie siły wydzielić do likwidacji skutków uderzeń przeciwnika;
- jakie elementy ugrupowania wzmocnić siłami inżynieryjnymi;
- jakimi siłami;
- na jakie etapy podzielić działania.

6. Sposób wykorzystania wojsk OPL, dotyczący:

- możliwości oddziaływania środków napadu powietrznego przeciwnika;
- organizacji rozpoznania środków napadu powietrznego;
- składu, gotowości i możliwości bojowych sił OPL;
- zadań w zakresie OPL realizowanych przez przełożonego i sąsiadów;
- własnych zadań w zakresie OPL;
- sposobów zwalczania desantów;
- organizacji powiadamiania i działania wojsk podczas odpierania uderzeń środków napadu powietrznego;
- zabezpieczenia materiałowego sił OPL;
- terminu gotowości systemu OPL.

Mając informacje:

- zadanie;
- wytyczne przekazane w pionie funkcjonalnym;
- zamiar dowódcy;
- stan sił i środków OPL;
- informacje o przeciwniku powietrznym;
- ile samoloto- i śmigłowcowylotów;

- jakie typy;
- na jakie obiekty;
- z jakich kierunków;
- sytuacja w powietrzu;
- kto panuje w powietrzu.
- ilość posiadanych środków OPL;
- zasięg i pułap rażenia posiadanych środków;
- ilość amunicji:
 - ogólnie;
 - podział na poszczególne etapy działania;
- ilość stacji radiolokacyjnych;
- parametry stacji radiolokacyjnych;
- charakterystyka terenu;
- zakaz prowadzenia ognia określony przez przełożonego:
 - do celów na określonej wysokości;
 - w określonym czasie;
 - w określonych korytarzach powietrznych

należy wyznaczyć:

- jak zorganizować dowodzenie / kierowanie ogniem;
 - scentralizowane;
 - zdecentralizowane;
 - na jakim szczeblu;
 - w jakich etapach działań;
- gdzie skupić główny wysiłek OPL:
 - w jakim etapie walki;
 - na jakim elemencie ugrupowania:
 - * I rzut;
 - * II rzut;
 - * SD;
 - * logistyka;
- na zwalczaniu jakich środków napadu powietrznego przeciwnika:
 - samoloty;

- śmigłowce;
- jak zorganizować rozpoznanie przestrzeni powietrznej;
- z kim współdziałać:
 - lotnictwo myśliwskie;
 - siły Obrony Powietrznej.

7. Sposób wykorzystania wojsk obrony przeciwchemicznej, dotyczący:

- możliwości użycia przez przeciwnika BMR;
- wpływu skażeń na działania wojsk;
- elementu (elementów) ugrupowania operacyjnego (bojowego) lub rejonu, na osłonie którego należy skupić główny wysiłek;
- przedsięwzięć ochronno-obronnych, które muszą być wykonane w wojskach;
- stanu i możliwości sił obrony przeciwchemicznej;
- sposobu użycia sił obrony przeciwchemicznej;
- zadań realizowanych przez przełożonego;
- sposobów prowadzenia rozpoznania skażeń;
- zadań grup awaryjno-ratowniczych;
- składu i rozmieszczenia sił do likwidacji skutków skażeń.

Mając informacje:

- zadanie;
- informacje otrzymane w pionie funkcjonalnym;
- zamiar dowódcy;
- siły i środki wojsk obrony przeciwchemicznej;
- stan;
- wyposażenie;
- wyszkolenie;
- ukompletowanie;
- warunki atmosferyczne;
- informacje o przeciwniku pod względem zagrożenia użycia broni masowego rażenia;

- możliwości sprzętu (wydajność urządzeń odkażających i innych);
- warunki atmosferyczne;
- temperaturę powietrza;
- siłę i kierunek wiatru ;
- ciśnienie powietrza;
- charakterystykę terenu

należy wyznaczyć:

- jak zorganizować rozpoznanie skażeń:
 - siłami autonomicznymi;
 - siłami specjalistycznymi;
- czy wykorzystać do tego celu śmigłowce:
 - specjalistyczne;
 - przystosowane;
- czy prowadzić zabiegi specjalne /sanitarne;
- jak metodą je prowadzić;
- autonomiczną (w oparciu o środki sił ogólnowojskowych);
- kompleksową (w oparciu o specjalistyczne siły opchem. i inne dostępne środki);

8. Sposób organizacji dowodzenia i łączności, dotyczący:

- podziału personelu na poszczególne SD;
- miejsc rozmieszczenia i dróg przemieszczenia głównych i zapasowych stanowisk dowodzenia;
- czasu pracy poszczególnych SD w określonych położeniach;
- czasu i sposobu wykorzystania doraźnie organizowanych stanowisk (punktów) dowodzenia;
- wyznaczenia zastępców,
- stanu i możliwości sił i środków łączności;
- zadań łączności;
- rozwinięcia i przesunięcia wozów łączności;
- ochrony systemu łączności przed oddziaływaniem przeciwnika.

Mając informacje:

- zadanie własnego szczebla dowodzenia (rozkaz);
- zarządzenie przekazane w pionie funkcjonalnym- dane z wytycznych
- łączności wyższego szczebla;
- zamiar dowódcy;
- osoby funkcyjne (ilość, rozmieszczenie na SD, informacyjne relacje pomiędzy mmi);
- stan sił i środków łączności;
- rozmieszczenie stanowisk dowodzenia w terenie i planowane przemieszczanie ich w kolejnych etapach działań;
- ogólne (instrukcyjne) zasady organizacji łączności;
- dane taktyczno -techniczne środków łączności;
- ograniczenia narzucone przez przełożonego np. zakresy częstotliwości, zakazy pracy w określonym czasie, nakazy pracy;
- warunki meteorologiczne;
- warunki terenowe;
- parametry istniejącej sieci telekomunikacyjnej

należy wyznaczyć:

- komu i z kim zapewnić łączność;
- ile relacji;
- jakie relacje;
- jakimi środkami;
- jakimi siłami;
- czy i gdzie musi być łączność utajniona;
- gdzie może / musi być łączność:
 - radiowa ,
 - radioliniowa ,
 - przewodowa;
 - inna;
- jakie siły i środki wydzielić na:
 - SD;
 - ZSD:

- TSD;
- inne punkty dowodzenia;
- jak wykorzystać istniejącą infrastrukturę telekomunikacyjną;
- które jej elementy wykorzystać;
- jak ochronić system łączności przed rozpoznaniem i oddziaływaniem radioelektronicznym przeciwnika;
- w jaki sposób przemieszczać system łączności podczas kolejnych etapów działań;
- komu dostarczyć dane łączności;
- kiedy muszą być dostarczone;
- kiedy je zmienić;

9. Sposób realizacji zabezpieczenia logistycznego, dotyczący:

- dróg dowozu i ewakuacji;
- sposobu rozmieszczenia i przesunięć urządzeń tyłowych;
- ilości zapasów materiałowych;
- czasu i sposobu zaopatrywania w środki materiałowe;
- sił i środków oraz sposobu ochrony i obrony tyłów;
- wykorzystania jednostek obrony terytorialnej na korzyść logistyki;
- wykorzystania infrastruktury lokalnej na potrzeby logistyki.

Mając informacje:

- zadanie;
- wytyczne przełożonego w pionie funkcjonalnym;
- zamiar dowódcy;
- stan etatowych i przydzielonych sił i środków logistyki;
- stan posiadanych zapasów materiałowych;
- stan i możliwe wykorzystanie infrastruktury miejscowej;
- tabelaryczne (zakładane) wartości zużycia środków i strat sanitarnych;
- ilość przydzielonych na działania:
 - jednostek ognia;
 - jednostek napełnienia;
 - racji żywnościowych;

- innych środków materiałowych;
- charakterystykę sieci komunikacyjnej w obszarze działań;
- ilość i zdolności przewozowe posiadany środków transportu i ewakuacji;
- wydajność środków remontowych;
- ilość i pojemność szpitali polowych;
- charakterystykę miejscowej infrastruktury logistycznej.

należy wyznaczyć:

zaopatrzenie:

- jakie środki materiałowe trzeba dostarczyć;
- ile których środków;
- komu je dostarczyć;
- kiedy muszą być dostarczone;
- jakimi środkami transportu;
- czyimi środkami:
 - własnymi;
 - podwładnego;
 - przełożonego;
 - sposobem kombinowanym;
- jakimi drogami;
- skąd można je pobrać;
- gdzie rozmieścić własne składy zaopatrzenia;
- które drogi wykorzystać jako główne i zapasowe drogi dowozu i ewakuacji;
- jak ochronić instalacje logistyczne i transport;

zabezpieczenie techniczne:

- co ewakuować;
- skąd;
- jakimi środkami;
- jakimi drogami;
- dokąd;

- do kiedy;
- co zrobić ze środkami których nie można wyewakuować;
- gdzie rozmieścić swoje instalacje e remontowe;
- z których stacjonarnych instalacji skorzystać;
- co wyremontować;
- do kiedy;
- jakimi siłami;
- co zrobić ze środkami których nie można wyremontować;

pomoc medyczna:

- jakich rannych ewakuować (ze względu na jakie wskazania);
- dokąd;
- jakimi środkami;
- gdzie rozmieścić instalacje pomocy medycznej;
- z których szpitali stacjonarnych skorzystać;
- jak skupić główny wysiłek;
- na jakim elemencie ugrupowania;
- na jakim rodzaju działań logistycznych;
- w jakim etapie operacji (walki).

Wyróżnione powyżej 9 problemów decyzyjnych może być zarówno wspomagane w jednym systemie eksperckim o dość skomplikowanej strukturze, gdzie pojawia się reprezentacja wiedzy z wymienionych dziedzin lub może być rozstrzygane w odrębnych systemach dziedzinowych. Szereg szczegółowych decyzji o charakterze ilościowym powinno być wyznaczane i wspomagane poprzez rozwiązanie zadań optymalizacyjnych pod warunkiem precyzyjnego opisu ograniczeń na poszczególne zmienne decyzyjne.

Przedstawione powyżej informacje w dużej mierze mają charakter ograniczeń. Tam, gdzie występuje niepełna bądź niepewna informacja wskazane jest zastosowanie heurystyk, opracowanych w drodze pozyskiwania wiedzy przez inżyniera wiedzy w oparciu o ustalenia z ekspertem dziedzinowym (np. specjalistę rodzaju wojsk). Heurystyki te mogą być zapisywane w postaci reguł decyzyjnych lub innych form reprezentacji

wiedzy, na przykład ocena możliwości pokonania terenu (przegrupowania wojsk) może odbywać się w oparciu o algorytmy znajdowania dróg w terenie zdigitalizowanym w postaci mapy cyfrowej np. reprezentowanym za pomocą specyficznej sieci neuronowej, zwanej komórkową, w której wyróżnione punkty terenowe przedstawiane bądź jako osobne neurony, a możliwe przejścia jako połączenia sieci neuronowej. Wzorce dróg pomiędzy wyróżnionymi punktami zapamiętywane są w postaci wyróżnionych wag poszczególnych neuronów. Problem ten może być również rozwiązany z zastosowaniem systemów informacji przestrzennej (GIS).

Należy ocenić, że wszystkie przedstawione problemy decyzyjne dadzą się wspomagać w systemie eksperckim, jednakże, aby ten proces był efektywny to powinno się zbudować system współpracujących ze sobą podsystemów eksperckich, na poszczególnych szczeblach dowodzenia oraz dotyczących wyróżnionych zadań decyzyjnych. Powinna charakteryzować je wspólna platforma informacyjna w postaci dostępnych baz danych oraz baz wiedzy z odpowiednimi, określonymi dla szczebla dowodzenia uprawnieniami.

Utworzenie specjalizowanych narzędzi wydobywania wiedzy i jej uogólniania z systemu baz danych o wojskach własnych i przeciwnika pozwoli stopniowo opracować bazy wiedzy, dostosowane do wyróżnionych zadań decyzyjnych.

5. KONCEPCJA EKSPERTOWEGO SYSTEMU WSPOMAGANIA DOWÓDCY ZT.

Sięgające początku lat osiemdziesiątych badania nad zastosowaniem komputerowego wspomaganie procesów podejmowania decyzji zaowocowały wprężeniem komputerów do rozwiązywania problemów i podejmowania decyzji w wielu dziedzinach. Obszar stosowania komputerowego wspomaganie decyzji jest bardzo rozległy, a jedną z dziedzin jego stosowania są działania bojowe. Poniższa propozycja próbą określenia sformalizowanych ram dla ekspertowego rozwiązywania zagadnień z zakresu działań bojowych.

5.1 Wymagania na ekspertowy system wspomaganie decyzji

Ogólne wymagania użytkownika

Istnieje duże zapotrzebowanie na konstrukcję komputerowego "narzędzia" do wspomaganie decyzji w planowaniu i prowadzeniu działań bojowych. Takie narzędzie powinno pozwalać na:

- łatwy opis sytuacji konfliktowej poprzez wprowadzanie:
 - ◆ stron konfliktu według pewnego schematu;
 - ◆ ich zadań (również przypuszczalnych);
 - ◆ zestawu posunięć (opisanych w czasie);
- identyfikację działań operacyjnych lub taktycznych w oparciu o wprowadzony opis sytuacji;
- wprowadzanie własnych sugestii eksperta-dowódcy co do możliwych scenariuszy wprowadzonej sytuacji konfliktowej;
- definiowanie własnych kryteriów oceny posunięć walczących stron;
- uzyskiwanie w trybie interaktywnym odpowiedzi na pytania dotyczące stron konfliktu w różnych przekrojach informacyjnych;
- pokazywanie rozwoju sytuacji konfliktowej (monitorowanie) wraz z wariantami jej rozwiązania;
- zapamiętywanie historii działań taktycznych i poszerzanie bazy informacyjnej o przebiegu tego typu konfliktu (uzupełnianie wiedzy o konfliktach), graficzną reprezentację elementów modelu;

- przygotowanie raportów z prowadzonej analizy sytuacji decyzyjnych;
- prowadzenie treningów osób podejmujących decyzję w operacji i walce.

Przedstawiany wyżej zestaw ogólnych wymagań na system wspomagania decyzji może być poszerzony, co nakłada na projektanta systemu konieczność przygotowania rozwiązań elastycznych. Ze względu na konieczność obiektywizacji przedstawionych zjawisk realizacja postawionego zadania może narażać na pewne trudności. Należy zwrócić uwagę na następujące zagadnienia:

- niepewność informacji co do stanu stron konfliktu, ich możliwych posunięć i preferencji;
- dostępność informacji;
- różne regulaminy walki i strategie postępowania (inny system wartości) stron konfliktu, indywidualizm dowódców stron walczących;
- konieczność uchwycenia tendencji, zależności o prawidłowości w celu uogólnienia modelowanej sytuacji konfliktowej lub w celu prognozowania rozwoju dalszych działań;
- dynamika operacji lub walki;
- możliwość weryfikacji proponowanych decyzyjnych odpowiedzi.

Wymagania można podzielić na następujące grupy:

- wymagania co do ogólności systemu wspomagania dowodzenia,
 - zakres wspomagania decyzji (planowanie działań bojowych w jak najszerszym zakresie),
 - objętość systemu wspomagania (jakie rodzaje podległych wojsk będą uwzględnione w systemie);
- wymagania co do jakości wspomagania dowodzenia;
 - łatwość komunikacyjna;
 - rozbudowany interfejs;
 - możliwości tłumaczenia decyzji;
 - możliwość określania alternatywnych rozwiązań,
 - śledzenie obiektów;
 - prezentacja jedynie potrzebnych informacji;
 - planowanie działań w trakcie posiadania bardzo ograniczonych źródeł informacji.

Wymagania co do ogólności systemu wspomagania dowodzenia.

Stworzenie systemu, który będzie wspomagał wszystkie podległe jednostki dowódcy ZT w każdym rodzaju przedsięwzięć, jaki one podejmują jest bardzo trudne. Po pierwsze taki system byłby zbyt skomplikowany, co wiązałoby się z czasem jego konstrukcji. Niektóre działania jednostek są rutynowe i odbywają się według utartych reguł, stąd ich wspomaganie nie jest celowe. System wspomagania dowodzenia powinien zajmować się:

- problemami przegrupowań jednostek w czasie działań,
- uzupełniania zapasów, rozmieszczania jednostek w terenie (tworzenia ugrupowania bojowego),
- planowania ruchu jednostek, przewidywania strat,
- planowania innych ważnych przedsięwzięć.

Zakres wspomagania decyzji powinien więc dotyczyć planowania działań bojowych w jak najszerszym zakresie. Ważne jest nie tylko to, co dana jednostka robi w danej chwili, ale także to, co zrobi w przypadku, gdy sytuacja rozwinię się nie po myśli dowódcy. Z tym problemem musi radzić sobie system ekspertowy, który ma być uznany za przydatny. Dąży się do tego, aby taki system obliczał straty, jakich dowódcy powinni się spodziewać w trakcie działań wojennych. Wyznaczanie ewentualnych strat nie musi być jednak dokładne, zależy ono bowiem od wielu czynników oraz od uproszczeń modelowych przyjętych w celu oszacowania strat.

Wspomaganie w takim systemie obejmie nie tylko komunikację z dowódcą ugrupowania bojowego, ale także dotyczyć będzie wyznaczania zadań podległych jednostek, planowania ugrupowania bojowego, planowania działań całego ugrupowania, wyznaczania możliwych strat w czasie walki, wspomagania rozpoznania przeciwnika.

Objętość systemu wspomagania (jaki rodzaje podległych wojsk będą uwzględnione w systemie) ma więc kluczowy wpływ na stopień komplikacji danego systemu, natomiast samo wspomaganie różnych rodzajów wojsk, może być realizowane przez połączenie osobnych systemów ekspertowych wspomagających wybrany fragment procesu dowodzenia. Przykładem takiego podejścia są systemy Wielkiej Brytanii, które powstają głównie przez łączenie systemów mniejszych, o węższym zakresie działania.

Wymagania co do jakości wspomaganie dowodzenia.

Dowódca związku taktycznego nie ma zwykle czasu na długotrwałe analizowanie sytuacji na polu walki. Nie ma więc czasu na analizowanie wszystkich danych napływających do niego, nie może śledzić sam całej sytuacji na ekranie monitora. Powinien mieć możliwość oglądania tylko najważniejszych elementów na najważniejszych odcinkach walki. Z tym wiążą się właśnie wymagania co do jakości wspomaganie dowodzenia.

Łatwość komunikacyjna to przede wszystkim, możliwie prosty interfejs łączący dowódcę z systemem wspomaganie decyzji. Interfejs ten, w zależności od jakości systemu może być graficzny lub graficzno-głosowy. Pierwszy z nich jest stosowany w większości systemów wspomaganie dowodzenia, ponieważ obraz jest niejako naturalnie związany z przedstawianiem sytuacji na polu walki. Zastępuje on mapę terenu z naniesioną sytuacją, dzięki czemu niesie w sobie tyle samo informacji co i mapa, łatwo jest go natomiast zmienić. Najbardziej zaawansowane systemy wspomaganie decyzji (zwłaszcza systemy US Navy) posiadają dodatkowo interfejs głosowy dla dowódcy. Dzięki niemu dowódca wydaje komendy głosem, bez potrzeby marnowania czasu na wprowadzanie danych do komputera.

Bardzo istotną cechą dobrego systemu wspomaganie decyzji jest rozbudowany interfejs, który posiada możliwość:

- tłumaczenia (uzasadniania) podjętych decyzji,
- możliwość określania alternatywnych rozwiązań,
- śledzenia obiektów, prezentacji jedynie potrzebnych informacji,
- planowania walki w trakcie posiadania bardzo ograniczonych źródeł informacji.

Każda z tych cech ma kluczowe znaczenie i każda z nich jest zwykle wymagana przez dowódcę.

Tłumaczenie podjętych decyzji jest możliwe w każdym profesjonalnym systemie wojskowym. Dowódca często nie jest pewien, dlaczego system podjął taką, a nie inną decyzję. System ekspertowy powinien mu ją więc objaśnić. W wojskowych systemach ekspertowych oprócz bazy wiedzy występuje także element zwany pierwotną bazą wiedzy, bazą wiedzy ekspertów itp. Jest to wiedza w postaci reguł (zwykle są to reguły), które

podane przez ekspertów z danej dziedziny są następnie zamieniane na taką reprezentację, wiedzy, jaka jest zastosowana w wojskowym systemie ekspertowym w jego właściwej bazie wiedzy. Baza pierwotna nie podlega jednak zniszczeniu, lecz jest zachowywana, gdyż posługując się jej regułami, tłumaczy się dowódcy podjęte przez system decyzje. Taka baza jest tworzona w przypadku posiadania reprezentacji wiedzy w postaci sieci semantycznych, ram itp. Reprezentacja ta jest bowiem trudna do rozszyfrowania przez osobę nie znającą się na programowaniu systemów sztucznej inteligencji.

W systemie ekspertowym istnieje również możliwość wyjaśniania decyzji dowodzenia. Możliwość taką udostępnia szkieletowy system ekspertowy (tzn. system ekspertowy, który posiada mechanizmy wnioskowania, nie posiada natomiast bazy wiedzy). Decyzja jest wyjaśniana poprzez pokazanie drzewa reguł (czyli kolejno sprawdzanych reguł), które zadecydowało o podjęciu takiej a nie innej decyzji.

Możliwość określania alternatywnych rozwiązań jest związana z wnioskowaniem w warunkach niepewnej lub niekompletnej informacji. Taka informacja jest zwykle dostępna na polu walki. Dlatego też muszą istnieć mechanizmy zmiany decyzji systemu ekspertowego w przypadku, kiedy okaże się, że dostaliśmy nową informację o sytuacji na polu walki, lub ta, którą mieliśmy, była błędna. System ekspertowy powinien mieć możliwość wnioskowania w warunkach niepewności.. Niektóre zmienne decyzyjne mogą mieć domyślne wartości..

Bardzo istotny w systemie wspomaganie decyzji jest mechanizm śledzenia obiektów. Wiadomo, że czym wyższy szczebel dowodzenia, tym więcej nowych informacji przybywa na stanowisko dowódcy. Nie wszystkie one są istotne, nie wszystkie wnoszą bowiem coś nowego do sytuacji na polu walki. Z jednej strony prawdą jest, iż czym więcej mamy informacji, tym łatwiej podjąć decyzję. Z drugiej jednak strony wiadomo, że nadmiar zbędnej informacji rzutuje na poziom odczytu i zrozumienia informacji istotnej. Dlatego też zwykle istotne jest obrazowanie najważniejszych elementów ugrupowania i tylko tych elementów, które wnoszą coś nowego do sytuacji na polu walki. System ekspertowy (a konkretnie ta część systemu wspomaganie decyzji, która jest odpowiedzialna za zobrazowanie) musi eliminować zbędną informację.

Pozadane cechy ekspertowego systemu wspomagania dowodzenia

System ekspertowy stworzony na potrzeby wojska (zwłaszcza system działający na polu walki) spełniać musi szereg dodatkowych warunków, które gwarantują jego przydatność i uniwersalność. Wśród tych warunków poza już wymienionymi są:

- zapewnienie rozległości systemu :
 - stworzenie niezawodnych sieci rozległych;
 - konstrukcja kanałów przesyłania danych o dużej przepustowości;
 - połączenie sieci lokalnych z różnymi serwerami sieciowymi, z sieciami rozległymi i wyznaczenie serwerów dominujących;
 - integracja sieci rozległych funkcjonujących w wojsku z sieciami cywilnymi;
 - zapewnienie redundantnych połączeń strategicznych w sieci rozległej;
 - stworzenie oprogramowania pracującego na wielu platformach;
- zapewnienie dokładności i fachowości systemu:
 - zaangażowanie ekspertów z danej dziedziny oraz programistów z komórek danego rodzaju sił zbrojnych;
 - zapewnienie zrozumiałego i przejrzystego interfejsu systemowego;
 - wykorzystanie wielu dziedzin nauki i wielu sposobów pozyskiwania i przetwarzania wiedzy;
 - określenie zasad testowania systemu;
 - określenie sposobów weryfikacji systemu podczas jego pracy, • określenie sposobu szkoleń systemowych;
 - przydzielenie komórki (zespołu) projektantów systemu do pracy w nowym środowisku wraz z docelową obsługą;
 - rzetelna dokumentacja faz projektowych systemu;
 - prowadzenie ewidencji błędów systemu w czasie jego pracy;
- stworzenie systemu o dużej niezawodności :
 - zabezpieczenie systemu przed zniszczeniem fizycznym (pożarem, trzęsieniem ziemi);
 - zabezpieczenie systemu przed włamaniem programowym;
 - zminimalizowanie ilości błędów;
 - stworzenie systemu z redundancją najważniejszych modułów;

- stworzenie systemu spójnego, bez nadmiarowych i zbędnych elementów opóźniających działanie;
- opracowanie procedur postępowania w przypadku upadku systemu;
- zapewnienie ochrony sprzętowej (niektóre komputery takie jak CRAY, AS/400 posiadają taką ochronę);
- zapewnienie ochrony programowej systemu (hasła, kody, prawa dostępu);
- zapewnienie bezpieczeństwa przesyłanej po sieci informacji;
- umożliwienie działania systemu w czasie rzeczywistym :
 - opracowanie architektury serwerów sterujących systemem;
 - wykorzystanie architektur równoległych;
 - zapewnienie mechanizmów komunikacji sieci lokalnych;
 - wykorzystanie urządzeń peryferyjnych (sensorów itp.) wraz z modułami komunikującymi te urządzenia z sieciami;

Systemy ekspertowe wspomaganie decyzji są uznawane za najtrudniejsze w realizacji, dlatego też przy ich tworzeniu należy wykorzystywać dostępne metodyki, w zakresie ich konstrukcji :

- funkcji systemu (modelowanie funkcji i diagramów funkcji);
- przepływów danych (modele łączące funkcje i obiekty);
- obiektów istotnych występujących w systemie (modelowanie obiektów czyli encji oraz ich atrybutów i połączeń między nimi);
- wywoływanych zdarzeń - najważniejszy moment przy projektowaniu systemu czasu rzeczywistego;
- modelowanie modułów i procedur.

W celu zapewnienia tego, że system jest dokładny i rzetelnie wykonuje swoje funkcje, należy angażować ekspertów z danych dziedzin nauki (do wizualizacji ekspertów zajmujących się grafiką wirtualną, do tworzenia baz wiedzy inżynierów wiedzy i projektantów baz danych itd.), a także ekspertów z danej dziedziny wiedzy wojskowej, którą automatyzuje się. Jest to najważniejszy czynnik podczas tworzenia nowego systemu ekspertowego na potrzeby armii. Ważnym elementem systemu jest interfejs graficzny.

Powinien on zapewniać:

- wizualizację sytuacji z pola walki (przez wizualizację rozumie się nie tylko wrysowywanie elementów, o których wiemy, ale także elementów, które przypuszczalnie występują) itp.;
- możliwość komunikacji głosem (zapewniana w każdym liczącym się systemie wojskowym - komunikacja szczególnie istotna w sytuacji, kiedy określone służby posługują się slangiem np.: systemy eksperckie floty USA rozpoznają slang);
- możliwość zadawania pytań i analizowania decyzji systemu (obecnie element obowiązkowy w każdym systemie wojskowym);
- możliwość cofania decyzji systemu, uzupełniania go wiedzą itp.;
- mechanizm wyjaśniania decyzji przez system ekspercki.

5.2 Koncepcja systemu ekspertowego wspomagania decyzji dowódcy ZT

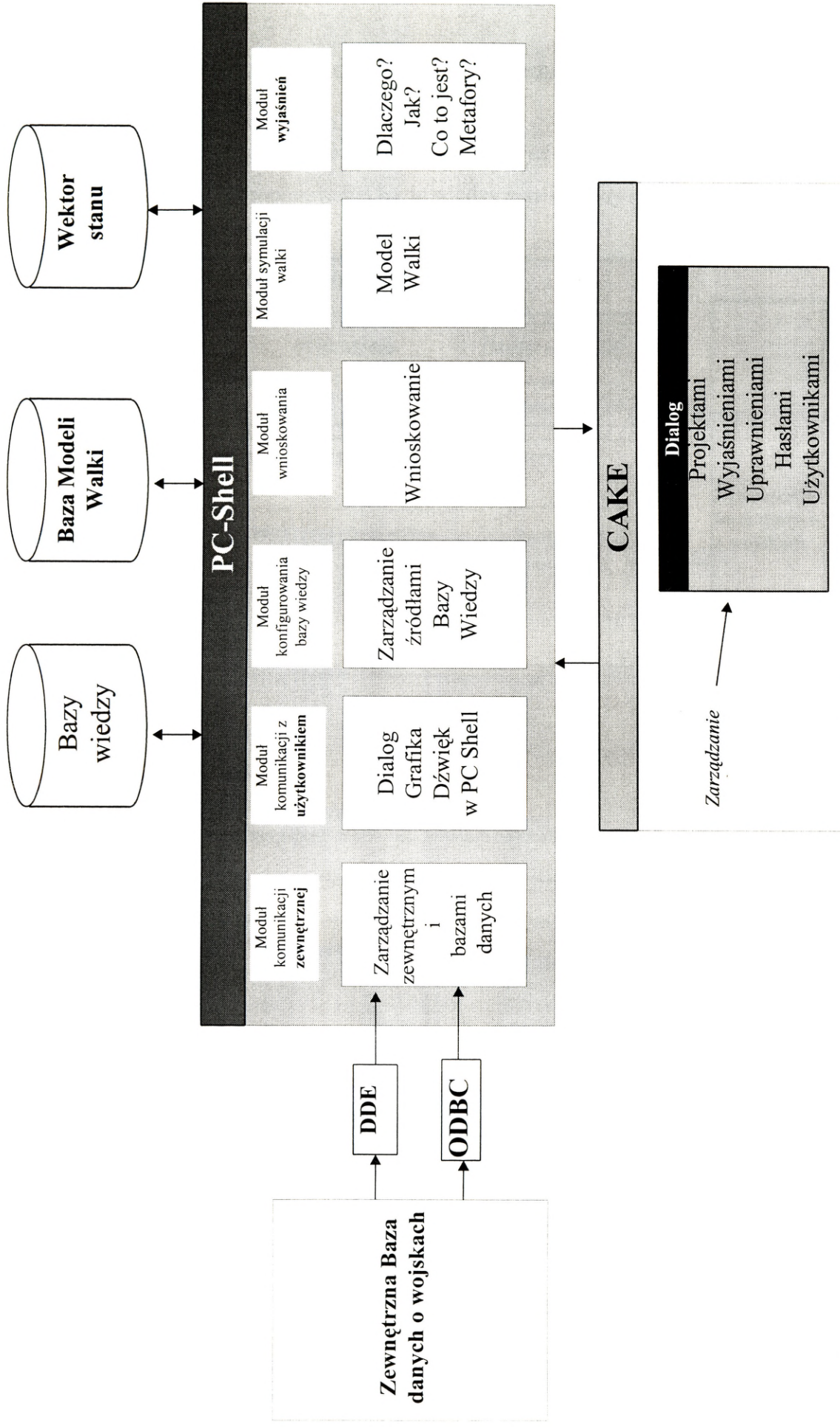
Projekt powinien być tak realizowany, aby umożliwił tworzenie, przebieg i analizę wyników „dobieranego” przez użytkownika modelu działania (walki zbrojnej). Podstawą tego podejścia jest baza wiedzy o modelowanych działaniach zbrojnych. Baza wiedzy ma zadanie pamiętać modele działań zbrojnych w postaci pewnego kodu. Użytkownik wprowadzając opis sytuacji konfliktowej - uruchamia mechanizm wnioskowania, który ma dać odpowiedź na pytanie „który z zapamiętanych wzorców walki odpowiada tej sytuacji?”. Ze względu na to, że sytuacja konfliktowa musi być wprowadzona w odpowiedniej formie, to należy przyjąć wstępny model walki. Użytkownik może ustalić ten model konfigurując go poprzez interfejs z wykorzystaniem zasobów bazy wiedzy i informacji z zewnątrz.

Struktura systemu ekspertowego

Projektowany system ekspertowy analiz decyzyjnych mógłby zawierać następujące elementy:

- użytkownik (człowiek) (U);
- zewnętrzna baza danych (ZBD);
- baza wiedzy (BW);
- maszyna wnioskująca (MW);
- interfejs użytkownika wraz z modulem objaśniającym (IU);
- moduł konfiguracji walki (MKK);
- moduł przyjętego do analizy modelu walki (RMK);
- moduł wektora stanu analizy walki (WS);
- moduł komunikacji z innymi systemami (MKS);

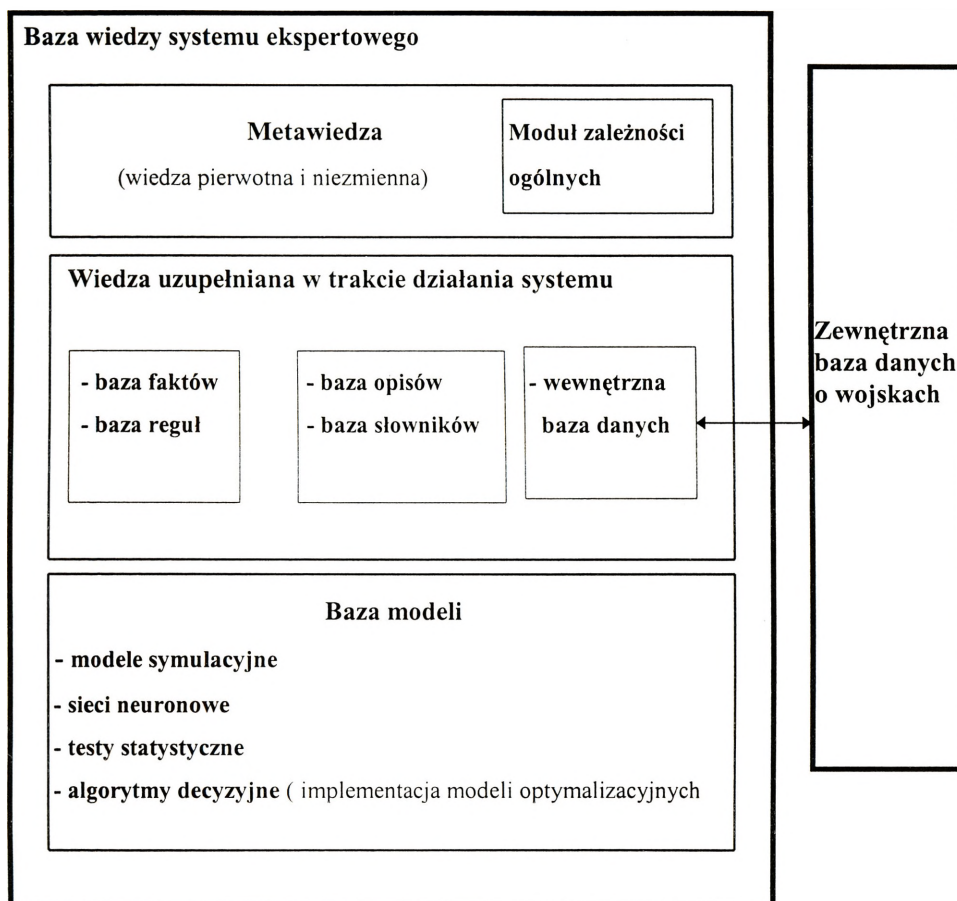
Interakcje między poszczególnymi elementami systemu zostały przedstawione w postaci schematu na rys.5.1.



Rys. 5.1. Struktura funkcjonalna systemu ekspertowego wspomaganie decyzji dowódcy ZT w środowisku zintegrowanego pakietu sztucznej inteligencji Sphinx.

Struktura bazy wiedzy systemu ekspertowego i jej elementy

Baza wiedzy składa się z wielu komponentów a jej skład zależny jest od metody reprezentacji wiedzy. W proponowanej bazie wiedzy wyróżniamy następujące elementy:



Rys. 5.2. Wewnętrzna struktura bazy wiedzy

- *moduł metawiedzy* - czyli wiedzy o wiedzy w systemie ekspertowym, która nie powinna podlegać zmianie w trakcie funkcjonowania systemu. Jakkolwiek jej uzupełnienie wiąże się z koniecznością ponownego testowania poprawności działania całego systemu ekspertowego. Można więc powiedzieć, że zmiana metawiedzy powoduje konstrukcję nowego systemu ekspertowego. W projektowanym systemie w skład metawiedzy wchodzi reguły i trójki, które określają sposób dochodzenia do rozwiązań wybranych problemów decyzyjnych.

W skład metawiedzy wchodzi także parametry konfigurujące mechanizm wnioskowania, takie jak:

- ◆ wybór sposobu przeglądania drzew reguł;
 - typ wnioskowania (wstępujące, zstępujące, bezpośrednie);
 - modyfikacje typu wnioskowania;
 - ◆ rodzaj parametrów decyzyjnych akceptowanych przez system;
 - ◆ sposób wyjaśniania podjętych decyzji;
 - ◆ sposób i zakres generacji raportów.
- *baza opisów* - zawierająca ogólne informacje z zakresu teorii konfliktów, definicje podstawowych pojęć, opisy historycznych konfliktów itp.;
 - *baza danych* - opisująca w postaci relacji dane podstawowe o potencjalnych uczestnikach konfliktów (typ uczestnika, dane adresowe, parametry ilościowe);
 - *baza reguł* - zawierająca opis reguł typu JEŻELI „niepełna wiedza o możliwych strategiach graczy” TO „przyjmij model konfliktu: hipergra”; wprowadzony zestaw reguł stanowi podstawę dla działania maszyny wnioskującej, ponadto jest źródłem dodatkowych faktów;
 - *baza faktów* - wszystkie fakty zbierane w systemie i uzyskiwane w drodze analizy są przechowywane w bazie faktów;
 - *baza modeli* - zawierająca wszystkie możliwe wzorce modeli konfliktów (gry konfliktowe z zerową sumą wpłat, z niezerową sumą wpłat, gry różniczkowe, stochastyczne, z wyborem momentu działania, koalicyjne, kooperacyjne itd.), testy statystyczne dla potrzeb badania zależności statystycznych, algorytmy decyzyjne, modele walki; wśród modeli zasługujących na szczególną uwagę znajduje się model symulacyjny konfliktu, który ze względu na możliwość oddziaływania na bazę wiedzy powinien być traktowany specjalnie, gdyż może dostarczać wiedzę o przypuszczalnych zachowaniach stron konfliktu;
 - *baza słowników* - zawierająca określenia i sformułowania z zakresu teorii konfliktów

Reprezentacja wiedzy i metody jej gromadzenia

Najczęściej spotykana i wykorzystywana metoda reprezentacji wiedzy, to regułowa reprezentacja wiedzy, której pewną odmianą jest reprezentacja w postaci drzewa. Każda gałąź rozpoczynająca się w korzeniu drzewa stanowi osobną regułę.

Dla potrzeb systemu ekspertowego wspomaganie dowódcy ZT zaleca się stosować zarówno regułową reprezentacją wiedzy jak też ramową reprezentacją, gdzie modele walki

mogą być opisane w postaci wzorcowych ram, w których szczeliny reprezentują cechy, atrybuty modelu (np. pojedyncze wartości, reguły, trójki, inne ramy, wektory preferencji, liczność zbioru sytuacji itd.), a model przyjęty do realizacji jako wystąpienie ramy (realizacja). Reguły typu **JEŻELI ... TO ...** mogą być wykorzystywane w procesie identyfikacji w fazach przejściowych do ustalania faktów.

Bazy wiedzy systemu ekspertowego tworzyć można przy zastosowaniu mechanizmów pakietu sztucznej inteligencji Sphinx przedstawionego w rozdziale drugim pracy.

Metoda akwizycji(pozyskiwania) wiedzy.

W literaturze przedmiotu ze względu na czynniki progresywne zaleca się aby wiedza w bazie wiedzy była zmieniana w trakcie działania systemu. Wiedza taka, to przede wszystkim zbiór danych o konflikcie, jego uczestnikach ich strategiach i możliwych preferencjach, może być modyfikowana na dwa sposoby:

- ◆ zmian dokonuje się w bazach danych (wiedzy) systemu ekspertowego automatycznie;
- ◆ zmiany wprowadzane są przez użytkowników systemu ekspertowego.

Pierwsza metoda jest bardzo wiąże się z koniecznością utworzenia mechanizmu uczenia systemu ekspertowego przez analogie, który źle zaprojektowany prowadzi do poważnych błędów w trakcie działania systemu. Drugi mechanizm jest łatwy w konstrukcji, ale powoduje konieczność czasochłonnych korekt baz danych. W niniejszym projekcie(koncepcji) systemu ekspertowego zaleca się stosować drugi typ zmiany wiedzy w trakcie działania systemu.

Maszyna wnioskująca

Do jej zadań należy :

- identyfikacja modelu walki odpowiadającemu rozpatrywanej sytuacji taktycznej;
- identyfikacja zbioru sytuacji występujących w walce i redukcja sytuacji niedopuszczalnych (nieracjonalnych) z punktu widzenia preferencji stron konfliktu;
- oszacowanie korelacji między czynnikami opisującymi stan walki i redukcja zbioru czynników;
- ustalenie stopnia niepewności w rozważanej sytuacji konfliktowej, będącego jednocześnie miarą ryzyka proponowanych rozwiązań;
- dobór kryteriów stabilności w walce(opisujących różne typy stanów równowagi).

Interfejs użytkownika i moduł objaśniający

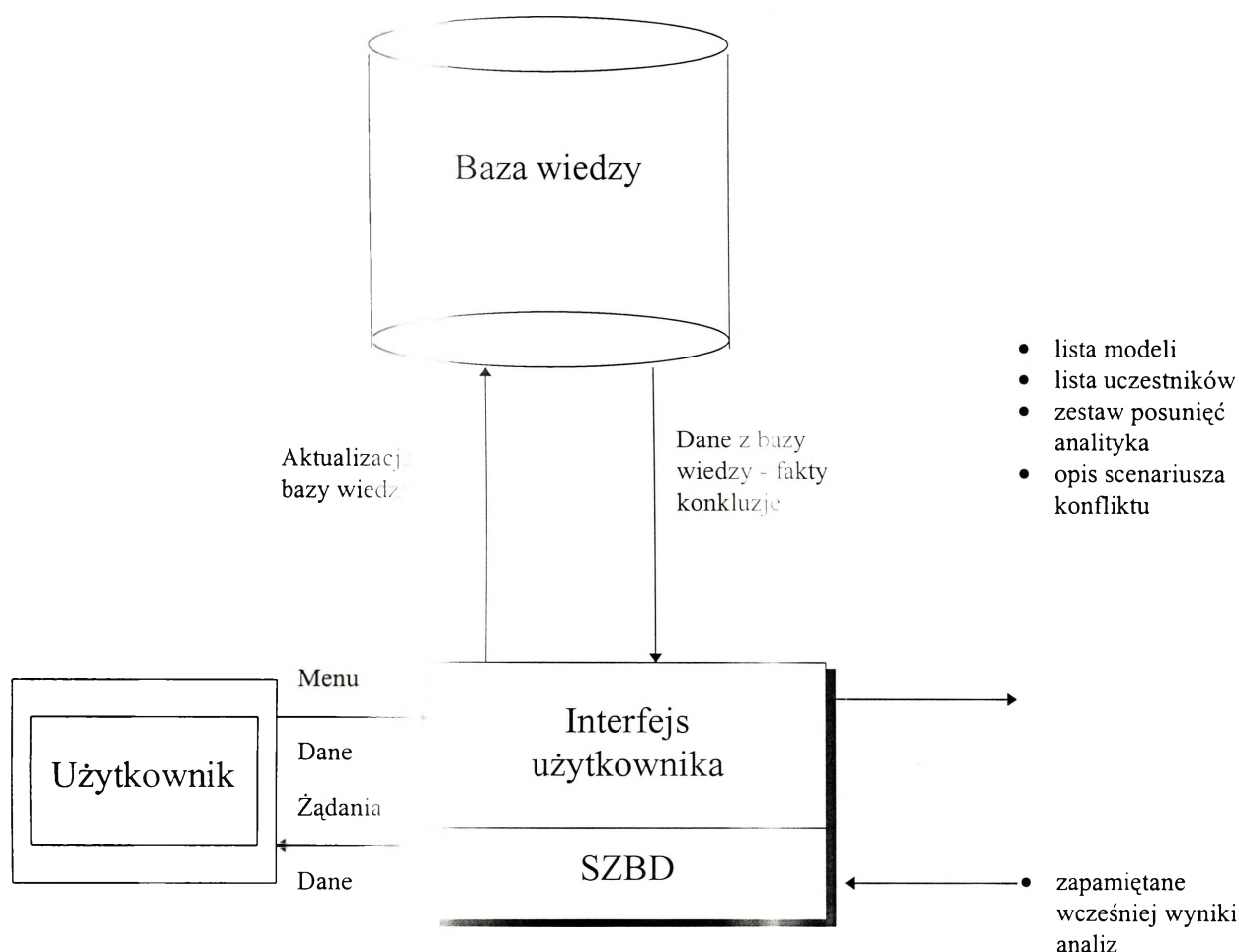
Ważną częścią systemu ekspertowego są moduły komunikacji z użytkownikiem. Nie jest to tylko jeden moduł przedstawiający sytuację w formie tekstowej czy graficznej. Jest to pakiet modułów zapewniających w większości systemów nowoczesnych także komunikację głosową. Tego typu komunikacja jest niezbędna ze względu na oszczędności czasowe jakie wprowadza. Stosuje się ją powszechnie we wszystkich systemach ekspertowych działających na potrzeby systemów szybkiego reagowania. Oprócz tego najbardziej zaawansowanego sposobu komunikacji systemu z użytkownikiem istnieją jeszcze następujące inne interfejsy udostępniania informacji użytkownikowi:

- tekstowy (w postaci drukowanych raportów; nie występuje nigdy sam, ze względu na swoją małą przejrzystość);
- graficzny (małoformatowy i wielkoformatowy);
- głosowo-graficzny.

Zadaniem interfejsu komunikacji z użytkownikiem jest przede wszystkim konieczność jasnego i przejrzystego udostępniania znanej informacji i podjętych decyzji a ponadto i przede wszystkim:

- wprowadzanie i konfigurowanie konfliktu w postaci opisu za pomocą specjalizowanego edytora z rozbudowanym menu, oknami dialogowymi, paletami obiektów graficznych reprezentujących poszczególne elementy sytuacji konfliktowej;
- uruchamianie procesu analizy z wykorzystaniem narzędzi zapisanych w bazie modeli (w ramach bazy wiedzy), wywoływanych za pomocą systemu menu;
- sterowanie przebiegiem analizy z możliwością pracy interaktywnej poprzez system zapytań;
- uruchamianie testów statystycznych do analizy uzyskiwanych w procesie analizy danych;
- prezentacja wyników analizy w różnych formach - wykresy, tabele, zestawienia, raporty.

Możliwy diagram przepływu danych jest przedstawiony na rys.5.3



Rys.5.3. Przepływ danych w interfejsie użytkownika

Moduł konfiguracji walki

Moduł ten zapamiętuje wszelkie informacje wstępne na temat walki, wprowadzane przez użytkownika z wykorzystaniem interfejsu oraz w oparciu o fakty zarejestrowane w bazie wiedzy, w szczególności w relacyjnej bazie danych oraz bazie faktów a ponadto korzysta z bazy opisów, w której mogą znajdować się dane historyczne o uczestnikach walki bądź opisy walki o podobnym charakterze. W ramach konfigurowania walki użytkownik może definiować własne kryteria oceny sytuacji taktycznej (w tym też stron walki) bądź może skorzystać z listy kryteriów zapisanych w bazie modeli. Przewiduje się utworzenie mechanizmu uzupełniania brakujących w opisie walki elementów poprzez system, bez których dalsza analiza byłaby niemożliwa.

Moduł objaśniający ma za zadanie informować na żądania użytkownika o realizacji wniosku z podaniem szeregu konkluzji.

Moduł realizowanego modelu walki.

W wyniku uruchomienia mechanizmu wniosku przeprowadzana jest w oparciu o moduł konfiguracji walki identyfikacja modelu walki. W rezultacie w module realizowanego modelu walki pojawia się realizacja modelu walki w oparciu o wzorzec modelu zapisany w bazie modeli (bazy wiedzy). Na tym tzw. wykonawczym modelu użytkownik dokonuje analizę decyzyjną. W projektowanym systemie ekspertowym opracowany zostanie mechanizm, który pozwala na rekonfigurację modelu walki w wyniku kolejnych etapów analizy poprzez np. zmianę preferencji uczestników walki, rozszerzenie bądź redukcję dopuszczalnych sytuacji (czyli także posunięć poszczególnych stron walki), rozszerzenie lub zmniejszenie listy uczestników walki np. wprowadzenie organizacji międzynarodowej jako mediatora bądź jako niezależną stronę konfliktu zbrojnego z własnym wektorem preferencji sytuacji w konflikcie. Wskutek rekonfiguracji może pojawić się nowy typ modelu walki, który nie znajduje się w bazie modeli i wtedy po dokonanej analizie będzie zapisany jako nowy, poszerzając tym sposobem wiedzę systemu ekspertowego.

Moduł wektora stanu.

Moduł ten odgrywa ważną rolę w systemie, gdyż „trzymana” jest w nim aktualna informacja o stanie stron walki na ustaloną chwilę oraz stan procesu analizy walki. W przypadku przerwania analizy możliwy jest restart systemu po pewnym czasie od momentu przerwania. Pozwala to na zatrzymanie procesu i wykonanie przez użytkownika dodatkowych operacji potrzebnych mu w dalszym przebiegu analizy lecz nie związanych bezpośrednio z daną sytuacją. Moduł ten pozwala po zakończeniu analizy na odtworzenie historii tego procesu, co ma aspekt dydaktyczny i stanowi podstawę mechanizmów programowych, wykorzystywanych do treningu osób występujących w roli decydentów jednej ze stron walki.

Moduł komunikacyjny.

Jest to zestaw narzędzi programowych, umożliwiających połączenie z zewnętrznymi bazami danych o wojskach oraz symulatorem działań bojowych. Moduł ten jest odpowiedzialny za format informacji pobieranych z zewnątrz i po przekształceniu czytelnych dla systemu ekspertowego. Powinien realizować też funkcję odwrotną. Informacje pobierane z zewnątrz - z serwera bazy danych oraz z symulatora sytuacji konfliktowych a także z sieci stanowi źródło pozyskiwania wiedzy dla systemu ekspertowego, stąd musi być przedstawione w formie czytelnej dla bazy wiedzy. Stanowi także złącze z zewnętrzną bazą danych oraz symulatorem aktywnie „realizującym” przebieg walki i dostarczającym nowych danych a często też nowej wiedzy w postaci zachowań różnych stron walki.

6. WNIOSKI

W pracy przedstawiono propozycję wykorzystania podejścia ekspertowego do wspomagania dowodzenia ZT. Dużą wagę przyłożono do zagadnienia analizy zadań decyzyjnych i możliwości wspomagania procesu ich rozwiązywania. W efekcie proponuje się różne warianty usytuowania systemu ekspertowego wielopoziomowego i rozproszonego (na zadania decyzyjne) w systemie wspomagania decyzji dowódcy. Oczywiście dyskusyjne mogą być przedstawione koncepcje wewnętrznej struktury systemu ekspertowego, ponieważ istotnie zależy to od przyjętych do jego tworzenia narzędzi programowych. Zastosowanie tzw. systemów powłokowych lub szkieletowych (w tym system *Spnix*) narzuca pewien standard systemu ekspertowego oraz metod reprezentacji wiedzy a także mechanizmów wnioskowania. Ocenia się, iż modernizacja (dostosowanie do warunków SZ RP) funkcjonujących w państwach sojuszniczych ze względu na specyfikę naszego narodowego systemu obrony napotykać będzie istotne przeszkody. Podejmując decyzję o potrzebie konstruowania systemu ekspertowego dla potrzeb wspomagania dowodzenia na szczeblu związku taktycznego (i nie tylko) należy wziąć to pod uwagę. Stosowanie pewnych rozwiązań gotowych (np. systemy szkieletowe, gotowe aplikacje, itp.) obok niewątpliwych zalet posiada zwykle tę wadę, że wprowadza szereg ograniczeń co do szczegółowych rozwiązań.

Po przeprowadzeniu szczegółowej analizy dostępnej w SZ RP zawartości bazy danych o wojskach, niestety nie jest możliwe rozwiązanie wielu problemów decyzyjnych ze względu na brak, bądź fragmentaryczność danych. Należy rozważyć modyfikację bazy informacyjnej dla potrzeb wspomagania dowodzenia, a ponadto zbudować mechanizmy uzupełniania wiedzy w bazie wiedzy w oparciu o bazę danych i zarysowane metody pozyskiwania wiedzy z baz danych, aby proponowane ekspertyzy były w pełni przydatne dowódcy oraz poddawały się weryfikacji.

Naturalnie w wielu krajach NATO doświadczenia w wykorzystaniu systemów ekspertowych do wspomagania dowodzenia są już niemałe. W latach osiemdziesiątych nastąpił istotny postęp w wykorzystaniu prezentowanego podejścia do wspomagania dowodzenia lub kierowania ogniem. Można tu podać jako przykład system BATTLE tworzony do wspomagania planowania ognia artylerii, pocisków ziemia-ziemia, manewrów i ognia śmigłowców i samolotów pola walki. Inny znany system to TACPLAN służący do planowania manewrów korpusu armijnego i dywizji zmechanizowanych. Zbudowano

znacznie więcej systemów ekspertowych wspomaganie dowodzenia i nie trzeba nikogo przekonywać o konieczności budowania tego typu systemów, ale należy mieć na względzie trudności z jakimi zetknęli się projektanci wymienionych i nie wymienionych systemów. Otóż podkreśla się trudności techniczne i architektoniczne w następujących elementach:

- interfejs użytkownika (konieczność dostosowania interfejsu do bardzo trudnych warunków pracy dowódców);
- bezpieczeństwo systemów;
- zarządzanie bazami danych;
- interoperacyjność (współpraca różnych aplikacji i środowisk);
- praca w sieciach (obciążenia i bezpieczeństwo);
- zakłócenia radioelektroniczne;
- duży poziom błędów;
- problemy pielęgnacji systemów;
- niezawodność;
- koszt.

Reasumując, należy wziąć pod uwagę wzmiankowane trudności, jednakże ze względu na stojące przed naszą armią zadania osiągnięcia interoperacyjności z armiami sojuszniczymi oraz wynikającą z międzynarodowych ustaleń konieczność utrzymania narodowego systemu obrony należy przedsięwziąć czynności zmierzające do budowy ekspertowego systemu wspomaganie do szczebla związku taktycznego.

7. LITERATURA

1. Chudy M., Nowicki T., Najgebauer A., Mielczarek K. - Projektowanie koncepcyjne systemu informatycznego wspomaganie analiz decyzyjnych w sytuacjach konfliktowych dla potrzeb kierowania obronnością państwa. Praca TEORIA II. Teoria konfliktów zbrojnych. AON WAT Warszawa 1996.
2. Piotr Sienkiewicz, Tadeusz Nowicki, Andrzej Najgebauer – „Metodyka projektowania eksperckiego systemu informatycznego wspomaganie analiz decyzyjnych w sytuacjach konfliktowych” Praca TEORIA II. Teoria konfliktów zbrojnych. AON, WAT Warszawa 1997.
3. Jałowicki M., Kręcikij J., Łukaszewicz Z., Urbanowicz J. - Model formalny procesu wspomaganie planowania i podejmowania decyzji w systemie dowodzenia wojskami. WAT, Warszawa 1997.
4. Zbigniew Świątnicki, Wojskowe systemy eksperckie, Wydawnictwo Bellona, Warszawa 1995.
5. Jan Mulawka, Systemy ekspertowe, Wydawnictwo Naukowo -Techniczne, Warszawa 1996.
6. Kulikowski, Wspomaganie decyzji, Systemy eksperckie, Materiały z konferencji, Polska Akademia Nauk, Warszawa 1995.
7. Liderman, Miernictwo i diagnostyka systemów komputerowych. Elementy systemów eksperckich, Wojskowa Akademia Techniczna, Warszawa 1992.
8. Świątnicki, R. Wantoch-Rekowski, Systemy ekspertowe, Wojskowa Akademia Techniczna, Warszawa 1995.
9. L. Raczyński, Istota i funkcjonowanie systemów eksperckich, Praca studyjna, Akademia Obrony Narodowej, Warszawa 1997.

