AKADEMIA OBRONY NARODOWEJ

Projekt badawczy 0500A 01923:
MODELOWANIE ZAGROŻEŃ DLA BEZPIECZEŃSTWA INFORMACYJNEGO PAŃSTWA. TEORIA WALKI INFORMACYJNEJ

WSPOMAGANIE WYBORU STRATEGII PRZECIWDZIAŁANIA ZAGROŻENIOM INFORMACYJNYM

(Koncepcja systemu ekspertowego)

Tom III

Raport z badań

WARSZAWA

AKADEMIA OBRONY NARODOWEJ

Projekt badawczy 0500A 01923:
MODELOWANIE ZAGROŻEŃ DLA BEZPIECZEŃSTWA INFORMACYJNEGO PAŃSTWA. TEORIA WALKI INFORMACYJNEJ

WSPOMAGANIE WYBORU STRATEGII PRZECIWDZIAŁANIA ZAGROŻENIOM INFORMACYJNYM

(Koncepcja systemu ekspertowego)

Tom III

Raport z badań

Warszawa **2004**

Opracował zespół autorski:

Gen. broni prof. dr inż. Tadeusz **JEMIOŁO**	**T.I. roz. 1.2.**
Płk prof. dr hab. inż. Piotr **SIENKIEWICZ**	**T.I. roz. 1.1., 1.9.,1.10., T.II.roz.1.1.**
Płk dr hab. Ryszard **SZPYRA**	**T.I roz.1.3.**
	T.II.roz.2.2., 2.3.
Mjr dr hab. Henryk **SPUSTEK**	**T.I. roz. 1.8.**
Płk dr inż. Marek **KINASIEWICZ**	**T.I. roz. 1.6.**
	T.II. roz.2.4.
Płk dr inż. Marian **URBANEK**	**T.I. roz. 1.7.**
Ppłk dr inż. Janusz **WOCIAL**	**T.I. roz. 1.7.**
Ppłk dr inż. Piotr **GÓRNY**	**T.I. roz. 1.4.**
Ppłk dr inż. Wiesław **BŁAŻEJCZYK**	**T.III roz.1.,3.,4.**
Mjr mgr inż. Jan **KUCHARSKI**	**T.I. roz. 1.5. T.III.roz.4.,5.**
Kpt. mgr inż. Jerzy **GRZYB**	**T.III. roz.2.,4.**
Mgr Halina **ŚWIEBODA**	**T.I. roz.1.9. redakcja całości**

Spis treści

1. Ogólne założenia i wymagania stawiane ESWD

W sytuacji ryzyka obok kompetencji decydentów istotne stają się również dobrze opracowane i opanowane procedury decyzyjne. Jednym z podstawowych nurtów we wspomaganiu decyzji stały się systemy eksperckie.

Pod pojęciem decyzji kryje się proces selekcji (wyboru), prowadzący do szczególnego działania, które ostatecznie jest podejmowane[1]. Jest to zatem przemyślany wybór wariantu działania, wybór jednej z możliwych w danej sytuacji akcji. Sytuacja decyzyjna zawiera następujące składniki:

1. Listę rozważanych, możliwych do wyboru wariantów działania (akcji);
2. Listę szacowanych (możliwych) skutków wynikających z podjęcia akcji ujętych w liście 1;
3. Dane opisujące efekty możliwych kombinacji akcji i ich skutków;
4. Szacunkowe wartości prawdopodobieństwa zaistnienia możliwych skutków działania;
5. Kryteria decyzji umożliwiające ewaluację przyjętych działań (decyzji).

Wspomaganie decyzji to proces opracowania informacji niezbędnej do podjęcia decyzji z wykorzystaniem różnych metod i na różnych etapach procesu podejmowania decyzji. System ekspercki[2] to program komputerowy, który wykorzystuje wiedzę i procedury wnioskujące do rozwiązania zadań wymagających ludzkiej ekspertyzy. Wiedza w systemie eksperckim składa się z faktów i heurystyk. Fakty stanowią podstawę informacji, która jest szeroko dostępna i akceptowana przez ekspertów w danej dziedzinie. Heurystyki natomiast stanowią bardziej prywatną informację, która charakteryzuje proces oceny i rozwiązania zadania przez konkretnego specjalistę. Są to intuicyjne domysły, przypuszczenia i zasady wynikające z wieloletnich doświadczeń. Poziom ekspertyzy oferowany przez dany system ekspercki jest funkcją rozmiaru i jakości bazy wiedzy tego systemu.

Systemy eksperckie „symulują" procesy rozwiązywania problemów wykonywane przez człowieka – eksperta. Przy budowie systemu eksperckiego

---

[1] Najgebauer A., Informatyczne systemy wspomagania decyzji w sytuacjach konfliktowych. Modele, metody i środowiska symulacji interaktywnej, dodatek do biuletynu WAT, Warszawa 1999.
[2] Schneider M., Kandel A., Fuzzy Expert System Tools, Wiley Publikations 1996.

wiedza jednego lub więcej ekspertów musi być ogarnięta i zmagazynowana w taki sposób, żeby mogła być wykorzystana do podejmowania decyzji.

System ekspercki ma w swoim założeniu naśladowanie rozumowania eksperta w procesie rozwiązywania problemów z określonej dziedziny wiedzy. Ekspert (człowiek) ma pewne specyficzne cechy, które predestynują go do tego miana[3]:

- uczy się, wykorzystując zdobyte doświadczenia;
- modyfikuje zbiór swoich pojęć;
- kieruje się zdrowym rozsądkiem (nawet wbrew pozorom);
- ma intuicję;
- może rozumować na podstawie analogii.

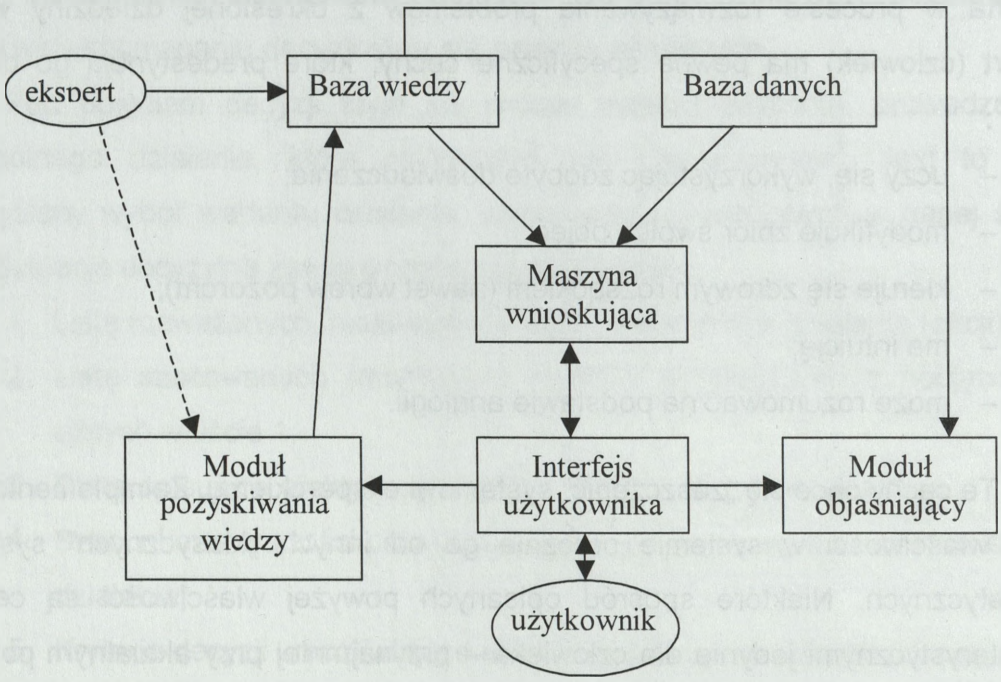
Te cechy chce się „zaszczepić" systemowi eksperckiemu. Zaimplementowanie takich właściwości w systemie odróżnia go od innych „klasycznych" systemów informatycznych. Niektóre spośród opisanych powyżej właściwości są cechami charakterystycznymi jedynie dla człowieka – przynajmniej przy aktualnym poziomie technologicznym.

Przyjmuje się, że system ekspercki powinien cechować się następującymi charakterystykami:

1. Być przyjazny dla użytkownika.
2. Mieć możliwość uczenia się na podstawie doświadczeń z przeszłości
   - nauka przez przykłady;
   - nauka przez analogię;
   - nauka przez doskonalenie umiejętności.
3. Posiadać możliwość objaśniania.
4. Umożliwiać wnioskowanie przybliżone.
5. Umożliwiać ograniczanie czasu opracowania ekspertyz.

[3] Na podstawie: Świątnicki Z., Wojskowe systemy eksperckie, Bellona, Warszawa 1995.

W składzie systemu eksperckiego wyróżnić można następujące moduły[4]: baza wiedzy, baza danych, maszyna wnioskująca, moduł objaśniający, interfejs użytkownika oraz moduł pozyskiwania wiedzy.



Rys. 1.1. Architektura systemu eksperckiego

Baza wiedzy – moduł zawierający wiedzę problemową z danej dziedziny.

Baza danych - moduł zawierający bieżące fakty i dane uprzednio zgromadzone.

Maszyna wnioskująca – program kontrolujący funkcjonowanie całego systemu. Zawiera mechanizm wnioskujący umożliwiający wyprowadzanie rozwiązania danego zadania.

Moduł objaśniający – moduł śledzący proces wnioskowania systemu eksperckiego i przekształcający zebrane informacje do postaci zrozumiałej dla użytkownika systemu w celu wyjaśnienia sposobu uzyskania rozwiązania.

Interfejs użytkownika – moduł komunikacji z użytkownikiem systemu eksperckiego.

Moduł pozyskiwania wiedzy – moduł przeznaczony do gromadzenia i wprowadzania nowej wiedzy do systemu.

[4] Kasabov N.K.,Foundations of neural networks, fuzzy systems and knowledge engineering, A Bradford Book The MIT Press, London 1996.

Ogólne założenia przyjmowane przy projektowaniu systemów eksperckich obejmują następujące stwierdzenia:

- systemy eksperckie naśladują proces myślowy ekspertów;
- siła systemów eksperckich wywodzi się ze zgromadzonej wiedzy w postaci bazy wiedzy lecz nie koniecznie z formy jej reprezentacji, czy sposobu wnioskowania;
- ekspertyzy uzyskiwane są na podstawie treningu, dostępu doświadczonych użytkowników;
- eksperci wspomagani systemem ekspertowym mogą podejmować dobre decyzje nawet w bardzo skomplikowanych sytuacjach;
- wiedza zgromadzona w systemach ekspertowych jest dostępna dla użytkowników nie będących ekspertami;
- systemy ekspertowe dopuszczają błędne ekspertyzy;
- niektóre systemy ekspertowe pozwalają na uzyskanie ekspertyz w trybie czasu rzeczywistego.

Wymagania odnośnie projektowania systemu eksperckiego zdeterminowane są odpowiedzią na zasadnicze pytania:

- Jaka jest dziedzina przedmiotowa projektowanego systemu lub inaczej, czemu ma służyć projektowany system?
- Kto będzie użytkownikiem systemu?
- Kto będzie dostarczał danych do systemu, w jakiej formie i jak często?
- Jakie są wymagania dotyczące interfejsu użytkownika?

Niezależnie od typu przyszłego użytkownika systemu eksperckiego taki system wspomagania decyzji powinien spełniać następujące wymagania:

- łatwy opis wspomaganego zagadnienia poprzez wprowadzanie uczestników (podmiotów) według pewnego schematu, ich możliwości i preferencji w określonych parametrach czasowych;
- identyfikację typu (rodzaju) wspomaganego zagadnienia w oparciu o wprowadzony jego opis;
- wprowadzanie własnych sugestii analityka zagadnienia, co do możliwych scenariuszy;
- definiowanie własnych kryteriów oceny zdarzeń (akcje i skutki);

- uzyskiwanie w trybie interaktywnym odpowiedzi na pytania dotyczące uczestników (podmiotów, przedmiotów) w różnych przekrojach informacyjnych;
- monitorowanie (zobrazowanie) rozwoju sytuacji wraz z wariantami jej rozwiązania;
- zapamiętywanie historii sytuacji i poszerzanie bazy informacyjnej o jej przebiegu wraz z graficzną reprezentacją;
- przygotowanie raportów z prowadzonej analizy;
- prowadzenie treningów osób podejmujących decyzje.

Przedstawiony powyżej zestaw wymagań odnośnie systemu wspomagania decyzji w zależności od rodzaju wspomaganego zagadnienia może być poszerzony, co nakłada dodatkowy wymóg elastyczności projektowanego systemu.

W odniesieniu do funkcjonujących - spełniających określone powyżej wymagania systemów ekspertowych wyróżnić można ewentualne wymagania nie związane bezpośrednio z zasadniczymi zadaniami systemów, które powinny dotyczyć (po uzasadnieniu potrzeby) takich funkcji jak:

- dostęp poprzez infrastrukturę telekomunikacyjną do usług rozproszonych terytorialnie serwerów;
- powszechne komunikowanie się użytkowników między sobą; wspólny język wymiany informacji i definiowania danych;
- tworzenie sieci i ich integracja w jednorodne środowisko obliczeniowe;
- centralne zarządzanie systemami funkcjonującymi w rozproszonym środowisku obliczeniowym i ich infrastrukturą telekomunikacyjną;
- zabezpieczenie systemów przed włamaniami i ujawnianiem zasobów informacyjnych i chronionych procedur działania;
- jakość usług obliczeniowych i telekomunikacyjnych
- priorytetowanie ruchu według rodzaju abonentów i typu wiadomości oraz możliwość definiowania sposobu reagowania na sytuacje wyjątkowe
- konieczność pracy w czasie rzeczywistym

2. Analiza i ocena możliwości wykorzystania pakietu sztucznej inteligencji SPHINX w tworzeniu ESWD[5]

System ekspercki jest programem komputerowym, wykorzystującym zapisaną wiedzę eksperta do rozwiązywania problemów, które normalnie wymagają ludzkiej inteligencji. Działanie systemu eksperckiego naśladuje eksperta w sposobie wyciągania wniosków i podejmowania decyzji - poprzez analizę faktów i odpowiedzi użytkownika na pytania kierowane do niego przez system. W wielu zastosowaniach, systemy eksperckie wspomagają eksperta w uzyskaniu profesjonalnej ekspertyzy. Podstawowymi cechami systemu eksperckiego są:

- zdolność rozwiązywania problemów specjalistycznych, w których dużą rolę odgrywa doświadczenie, tzw. wiedza ekspercka;
- unifikacja wnioskowania - te same przesłanki determinują tą samą konkluzję;
- stabilność ekspertyzy - 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 eksperckiego, w zakresie rozwiązywania danego problemu, tkwi w zakodowanej w nim wiedzy, a nie w formalizmie i schematach wnioskowania, których ten program używa. W zasadzie można powiedzieć, że wiedza dziedzinowa (z zakresu problemu) jest odseparowana od mechanizmów wnioskujących. Posiadanie pełnej wiedzy w bazie wiedzy, a nie sposób realizacji procesu wnioskowania systemu eksperckiego, decyduje o jego skuteczności. Oznacza to, że aby zbudować inteligentny program, należy go wyposażyć w dużą ilość, dobrej jakości, specyficznej wiedzy o danym przedmiocie.

Ze względu na sposoby realizacji systemy eksperckie możemy podzielić na dwie grupy[6]:

---

[5] Opracowano na podstawie dokumentacji użytkowej pakieu SPHINX i materiałów szkoleniowych firmy AITECH

[6] Mulawka J., Systemy ekspertowe, Wydawnictwa Naukowo-Techniczne, Warszawa 1996, str.27

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

Ze względu na metodę prowadzenia wnioskowania systemy ekspertowe dzieli się na:

- z logiką dwuwartościową (Boole'a);
- z logiką wielowartościową;
- z logiką rozmytą.

Ze względu na rodzaj przetwarzanej informacji systemy ekspertowe dzielą się na dwie grupy

- systemy z wiedzą pewną, czyli zdeterminowaną;
- systemy z wiedzą niepewną, w przetwarzaniu której wykorzystuje się przede wszystkim aparat probabilistyczny.

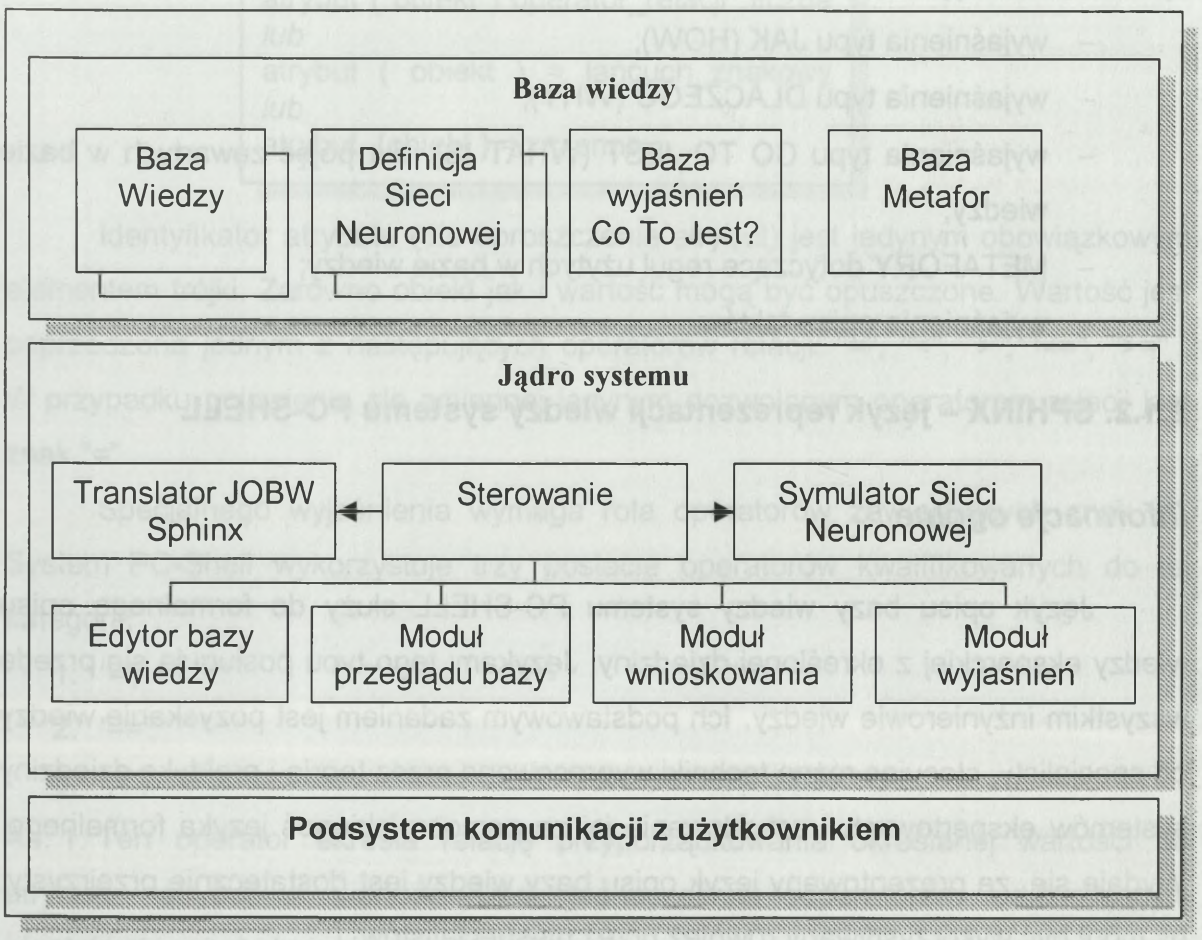
Pakiet sztucznej inteligencji SPHINX jest w pełni zintegrowany i zapewnia współpracę poszczególnych systemów wchodzących w jego skład. Są to:

- PC-Shell – szkieletowy system ekspercki,
- Neuronix – system do tworzenia sieci neuronowych,
- CASE – system wspomagania inżynierii wiedzy,
- DeTreeX – system indukcyjnego pozyskiwania wiedzy,
- Predyktor – system do budowy modeli prognostycznych.

2.1. Szkieletowy system ekspercki PC-SHELL

2.1.1. Charakterystyka systemu

PC-SHELL jest podstawowym elementem pakietu sztucznej inteligencji SPHIX. Jest to dziedzinowo niezależne narzędzie do budowy systemów ekspertowych. System PC-SHELL współpracuje z innymi elementami pakietu – systemem NEURONIX przeznaczonym do tworzenia sieci neuronowych, systemem CAKE przeznaczonym do wspomagania pracy inżyniera wiedzy oraz realizującym funkcje systemu, dbMaker zarządzającego bazami wyjaśnień. Architektura systemu PC-SHELL pokazana została poniżej:



Rys.2.1. Architektura systemu PC-SHELL.

PC-SHELL charakteryzują następujące cechy:

1) W zakresie struktury systemu:
   - elementy architektury tablicowej;
   - hybrydowość systemu.
2) W zakresie reprezentacji wiedzy:
   - deklaratywna reprezentacja wiedzy w formie reguł i faktów;
   - algorytmiczna (proceduralna) reprezentacja wiedzy w formie programu zawartego w bloku sterowania (control);
   - pełne rozdzielenie wiedzy eksperckiej i procedur sterowania;
   - wiedza o charakterze rozproszonym zawarta w sieci neuronowej;
   - faktograficzna w formie tekstów, grafiki, dźwieku, sekwencji wideo;
   - wiedza ekspercka może być zawarta w kilku źródłach wiedzy.
3) W zakresie wyjaśnień:
   - wyjaśnienia typu JAK (HOW);
   - wyjaśnienia typu DLACZEGO (WHY);
   - wyjaśnienia typu CO TO JEST (WHAT IS), dla pojęć zawartych w bazie wiedzy;
   - METAFORY dotyczące reguł użytych w bazie wiedzy;
   - wyjaśnienia opisu faktów.

2.1.2. SPHINX – język reprezentacji wiedzy systemu PC-SHELL

Informacje ogólne

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. 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 x 10^{-38} do 3,4 x 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 określanej wartości do atrybutu: atrybut = wartość. Jest jednocześnie separatorem oddzielającym identyfikator atrybutu od jego wartości.

```
średnia_odległość = 20.5
poziom_bezpieczeństwa= "dobry"
```

Operator '=' wykorzystywany jest głównie w bloku reguł i faktów.

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.

```
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.

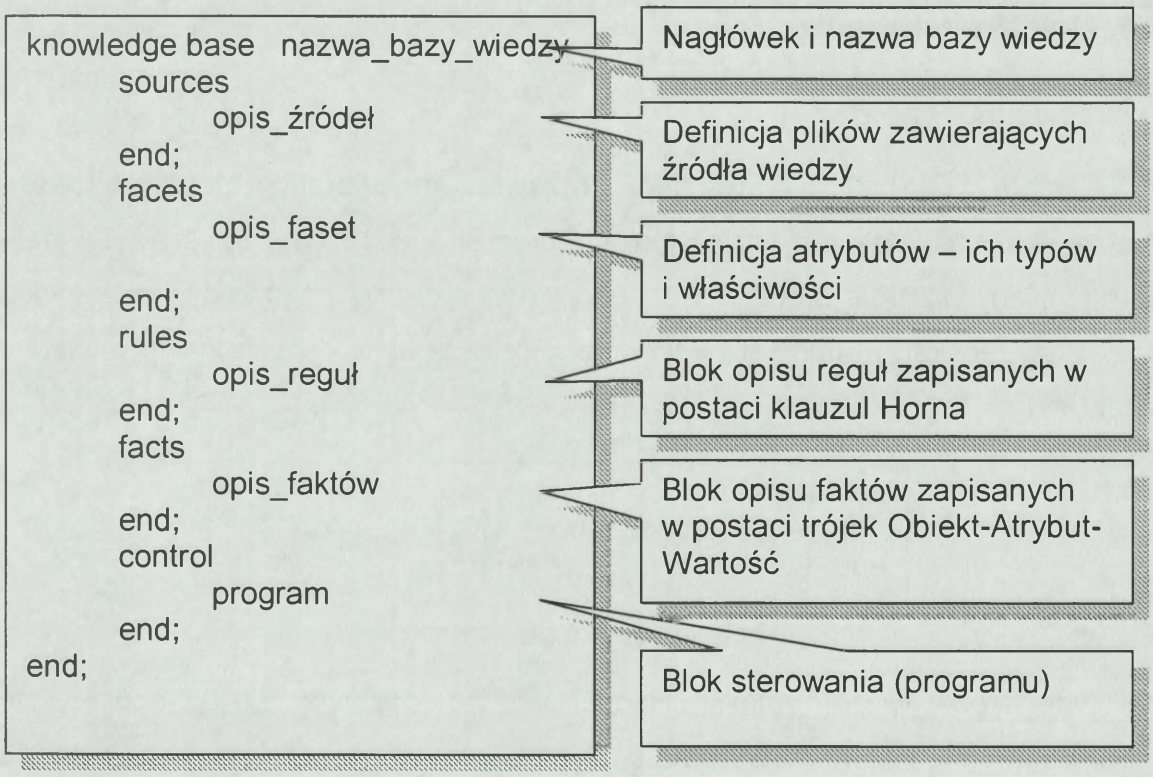
```
//deklaracja zmiennej typu znakowego
        char Moja_zm1;
//deklaracje zmiennych o wartościach
rzeczywistych
      float Moja_zm2, Moja_zm3;

      Moja_zm2 := 120.95;
      Moja_zm1 := "dowolny_łańcuch";
      Moja_zm2 := Moja_zm3;
```

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.

Struktura języka opisu bazy wiedzy

Opis bazy wiedzy w systemie PC-Shell jest podzielony na pięć bloków: opisu plików, faset, reguł, faktów oraz sterowania. Obowiązkowe jest wystąpienie przynajmniej jednego bloku - faktów lub reguł. Ogólną strukturę opisu bazy wiedzy przedstawiono poniżej. Nazwa_bazy_wiedzy jest dowolnym symbolem.



Jak już wspomniano, system PC-Shell ma możliwość korzystania z wielu źródeł wiedzy. Strukturę opisu źródła wiedzy przedstawiono poniżej:

```
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. Opis źródeł wiedzy

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.

```
sources
        opis_źródeł_wiedzy
end;
```

Opis_źródeł_wiedzy musi składać się co najmniej z jednego opisu, składającego się z nazwy źródła oraz specyfikacji właściwości źródła. Nazwa źródła jest dowolną nazwą ustaloną przez inżyniera wiedzy, natomiast specyfikacje składają się z wyrażeń zawierających słowa kluczowe systemu. Po nazwie źródła musi wystąpić znak ':'.

```
nazwa_źródła :
        właściwość1...właściwośćN ;
```

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

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.

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. Nie jest dozwolone użycie w źródle opisów plików, faset oraz 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:

```
// Ekspertowy sytem wspomagania decyzji - wyboru strategii
// przeciwdziałania zagrożeniom informacyjnym.
knowledge base system_wspomagania_decyzji
sources
 bezpieczeństwo_fizyczne:
   type kb
   file "bezpieczeństo_fizyczne.zw";
 .bezpieczeństwo_programowe:
   type kb
   file "bezpieczeństwo_programowe.zw";
```

Źródła typu neural_net

Podobnie jak w przypadku innych deklaracji źródeł wiedzy, deklaracje sieci neuronowych składają się ze zbioru oddzielnych deklaracji sieci neuronowych. Począwszy od wersji 2.3 zniesiono limit ilości dopuszczalnych źródeł.

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_zagrożeń :
                        type neural_net
                        file
"c:\\system\\sieć\\prognoza.npr" ;
        end;
```

Źródła typu metaphor i what_is

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

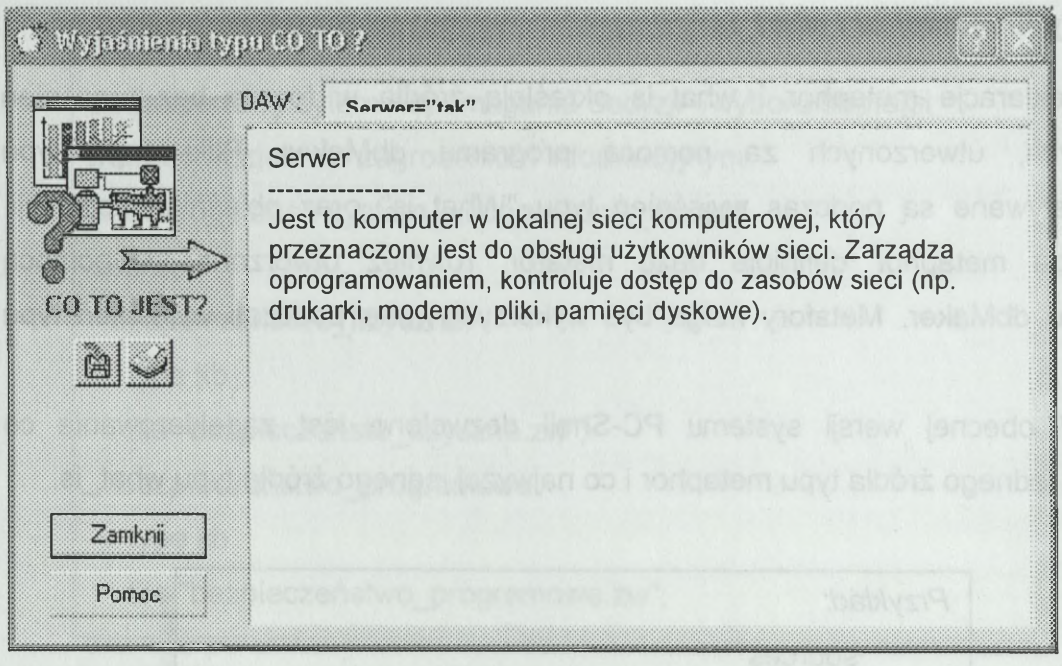
W obecnej wersji systemu PC-Shell dozwolone jest zadeklarowanie co najwyżej jednego źródła typu metaphor i co najwyżej jednego źródła typu what_is.

Przykład:

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

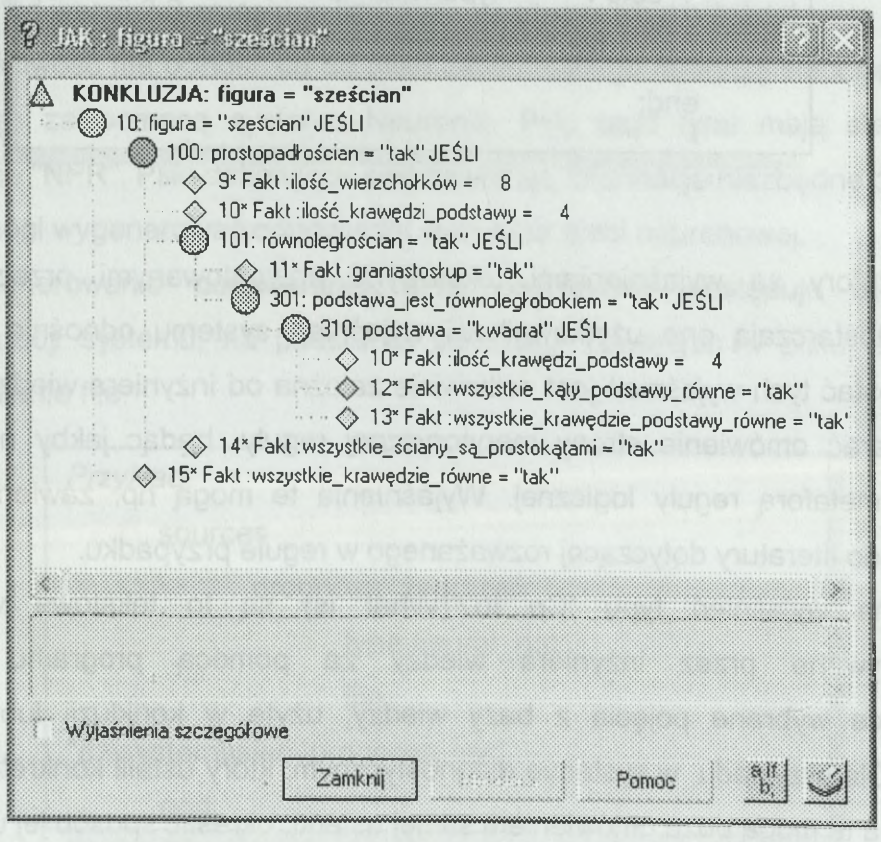
Metafory są wyjaśnieniami tekstowymi przygotowanymi przez inżyniera wiedzy. Dostarczają one użytkownikowi wyjaśnień systemu odnośnie konkretnej reguły. Postać tych wyjaśnień jest całkowicie zależna od inżyniera wiedzy. Powinny one zawierać omówienie strony merytorycznej reguły, będąc jakby nieformalną, tekstową metaforą reguły logicznej. Wyjaśnienia te mogą np. zawierać również odnośniki do literatury dotyczącej rozważanego w regule przypadku.

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.



Rys.2.2. Przykładowe okno wyjaśnień typu „Co to?"

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).



Rys.2.3. Przykładowe okno wyjaśnień typu „Jak?"

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. Opis faset

Ogólna struktura opisu 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
                yes
        [ ask {     } ]
                no

                yes
        [ single {     } ]
                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. 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 faset. System przyjmuje domyślnie: ask yes oraz single no.

W systemie PC-Shell 3.0 dostępne są następujące rodzaje faset: ask, unit, val (values), single, query oraz fasety param, picture, sound oraz video. Spośród wymienionych, musi wystąpić przynajmniej jedna; kolejność deklaracji jest dowolna.

Szczegółowy opis faset

Faseta ask

ask { yes / no } ;

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 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 fasecie, zamiast pytań automatycznie generowanych przez system.

Wystąpienie listy zapytań musi być powiązane z zadeklarowanie 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
        poziom_bezpieczeństwa:
                ask yes
                query "Podaj poziom bezpieczeństwa:" ;
        stan_zagrożenia :
                val oneof { "niski", "wysoki" }
                query { "Czy stan zagrożenia jest niski ?",
                        "Czy stan zagrożenia jest wysoki ?" };
end;
```

Faseta single

```
single { yes | no } ;
```

Faseta single umożliwia zadeklarowanie, że w bazie wiedzy może wystąpić tylko jeden fakt zawierający atrybut, do którego odnosi się faseta single. Ma to najczęściej miejsce w odniesieniu do atrybutów, których wartości wzajemnie się wykluczają. Dla przykładu, jeśli w bazie wiedzy występuje pewien fakt, np.: "ilość_stacji_roboczych=250" i atrybut "ilość_stacji_roboczych " opisano fasetą "single yes", to system przyjmie, że w bazie wiedzy nie ma innego faktu stwierdzającego np. że " ilość_stacji_roboczych=400". 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;
```

Faseta unit

```
unit jednostka_miary;
```

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
        wartość_sprzętu :
          unit "tysiące zł"
        ask yes
        query "Podaj wartość sprzętu:" ;
end;
```

Faseta val

Określa zbiór dopuszczalnych wartości danego atrybutu. Wartości mogą być liczbami rzeczywistymi (typ float) lub łańcuchami znakowymi. W obecnej wersji systemu do określenia dozwolonych lub niedozwolonych wartości służą następujące deklaracje związane z 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.

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

Faseta val 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 fasecie 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
        stan_bezpieczeństwa :
                query "Określ stan bezpieczeństwa systemu:" ;
                val oneof { "niski", "średni", "wysoki" };
end;
```

Faseta val 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.

Faseta val 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 którejś z wymienionych faset: oneof, someof, except. Dla oznaczenia wartości minimalnej oraz maksymalnej w danej implementacji można używać odpowiednio symboli min oraz max.

```
Przykład:
facets
  wielkość_sieci :
        val range < 10, 200 >;
  parametr_X:
        val range { < -50, -10 >, < 10, 50 > };
  parametr_Y:
        val range { < min, 0 ), ( 0, max > };
end;
```

Faseta val 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 którejś z wymienionych faset: oneof, someof, range. Podobnie jak w przypadku fasety range, dla oznaczenia wartości minimalnej oraz maksymalnej w danej implementacji, można używać odpowiednio symboli min oraz max.

```
Przykład:
facets
 wielkość_sieci :
        val except { < MIN, 36), ( 42, MAX > };
 parametr_X:
        val except { < MIN,-50 ),( -10, 10 ),( MAX,50 > };
 parametr_Y:
        val except { 0 };
end;
```

Faseta param

param *{ zmienna1 = wartość1,..., zmiennan = wartośćn }*

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

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

```
Przykład:
facets
 ..
 atrybut_1 :
        val range < 1, 3 >
        param { PMIN = 2.1, PMAX = 2.8 };
 atrybut_2 :
        val oneof { "biały", "zielony", "niebieski" >
        param { P1 = " biały ", P2 = " niebieski " };
end;
```

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 związanym 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.

```
Przykład :
facets
        komputer :
                val oneof { "terminal", "laptop", "serwer" }
                picture { "termina.bmp", "notebook.bmp",
"serwer.bmp" };
        czujnik:
                val oneof { "ruchu", "brak" }
                picture "czujnik_ruchu.bmp";
```

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).

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

Faseta video

Faseta video umożliwia związanie atrybutów z animacjami, np. w formie 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.

2.1.5. Opis faktów i reguł

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 oraz ich poprawnej budowy, bez zaznaczenia wymienionych ograniczeń, pokazano poniżej.

Składnia faktów :

```
        trójka_OAW;
  lub
        not trójka  OAW;
```

Poprawnie zbudowane fakty:

```
facts
        miesięczne_straty_zakładu (zakład_X) = 205.5;
        ocena_poziomu_zabezpieczeń (stacja_robocza1) =  "dobra";
        nie wystąpił_atak_na_system;
        rodzaj_ zabezpieczeń = "oprogramowanie antywirusowe";
        rodzaj_ zabezpieczeń = "firewall";
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).

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 dzielimy na proste i złożone. Reguły proste są odzwierciedleniem reguł Horne'a czyli schematu:

konkluzja jeśli warunek$_1$ i warunek$_2$ i ... i warunek$_N$

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 :] konkluzja$_1$ if warunek$_1$ & warunek$_2$ &...& warunek$_n$;

[numer_reguły :] konkluzja$_2$ if warunek$_1$ | warunek$_2$ &...& warunek$_n$;

Należy zastrzec, że operacja łącznika logicznego 'i' jest silniejsza od operacji alternatywy 'lub'.

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 poniżej.

```
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 poniżej :

```
X := 6.2
Y1 := Y2
Z3 := 4 + 7
X4 := Y5 - Z6
Y6 := 23 + sin ( X7 )
Z8 := sin ( X9 ) + cos ( Y10 )
```

2.1.6. Programowanie w systemie PC-SHELL

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 doskonalony i rozwijany również w przyszłych wersjach tego systemu, z uwzględnieniem specyficznych wymagań stawianych przez system ekspertowy. Zakładać należy, że rozwój języka będzie następował z zachowaniem zgodności "w górę".

```
control
        program
end;
```

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

```
[deklaracje zmiennych]
instrukcje
```

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, łańcuch_znakowy;
float X, Y,Z, średnia;
```

Deklarowane zmienne mogą być zmiennymi prostymi (przykład wcześniejszy) lub zmiennymi indeksowymi (tablicami).

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 zmiennych języka SPHINX

Począwszy od wersji 2.1 system PC-SHELL został rozszerzony o nowe typy zmiennych. Do dyspozycji użytkownika dodano nowe typy numeryczne oraz wprowadzono możliwość definiowania rekordów.

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 pokazano poniżej.

Zakres typu int w wersji 2.3, w związku z przejściem na architekturę 32-bitową uległ zmianie.

Zakresy typów systemu PC-SHELL przedstawiono poniżej.

Typ	Ilość bitów	Zakres
int	32	-2 147 483 648 .. 2 147 483 648
longint	32	-2 147 483 648 .. 2 147 483 648
float	32	$3.4 * 10^{-38}$ .. $3.4 * 10^{38}$
double	64	$1.7 * 10^{-308}$ .. $1.7 * 10^{308}$

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 dającym się automatycznie rzutować;
- rzutowanie 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 konwersji liczb na inne sposoby zapisu liczb (w szczególności polski z przecinkiem).

```
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 +1.4;  // 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 w niektórych innych językach (np. w Fortranie lub PL/1).

Przykład deklaracji tablic:

```
float TABLICA1[10],     Tablica jednowymiarowa
TABLICA2[5,10];         Tablica dwuwymiarowa
int TABLICA3[2,2];      Tablica dwuwymiarowa
```

Rekordy

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

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

opcjonalna_lista_zmiennych;
```

Przykładowa definicja rekordu wygląda następująco:

```
record Czas
    begin
        int godzina, minuta, sekunda;
    end Czas;

record Dane_personalne
    begin
        char Nazwisko, Imie;
        int  Wiek;
        float  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. Przykłady poprawnych wyłuskań pól przedstawiono poniżej:

```
Czas.godzina := 12;
Czas.minuta := 10;
Czas.sekunda := 0;
Czas.minuta := Czas.minuta + 20;
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 '#'.

Instrukcje programowania

Instrukcja przypisania

Instrukcja przypisania może przyjmować jedną z następujących postaci:

```
zmienna := wyrażenie_arytmetyczne;
zmienna := symbol
zmienna := łańcuch_znakowy
zmienna := zmienna
```

Zmienna może być zmienną dowolnego typu prostego (char, float, double, int lub long). Jeśli zmienna jest zadeklarowana jako char, to po prawej stronie operatora przypisania może pojawić się zmienna typu char, symbol lub łańcuch znakowy.

```
char C1, C2, C3[4,5];
C2 := wartość_typu_symbol;
C2 := "wartość typu 'łańcuch znakowy'";
C1 := C2;
C3[2,3] := C1;
```

Jeśli po lewej stronie operatora ":=" występuje zmienna zadeklarowana jako zmienna numeryczna, to po prawej stronie operatora może pojawić się wyłącznie wyrażenie arytmetyczne, którego szczególnym przypadkiem jest stała liczbowa lub zmienna typu numerycznego, zgodnie z zasadą typowania typów.
W wyrażeniu arytmetycznym mogą pojawić się:

- operatory arytmetyczne +, -, *, /;
- liczby;

- zmienne typu numerycznego;
- wywołania funkcji matematycznych;
- nawiasy okrągłe.

W odróżnieniu od wyrażeń arytmetycznych stosowanych jako warunki reguł, wyrażenia w bloku control mogą zawierać większą liczbę argumentów.

Wyrażenie arytmetyczne dla operatorów o tych samych priorytetach jest wykonywane od strony lewej do prawej. Jeśli argument występuje między operatorami o różnych priorytetach, to następuje jego związanie z operatorem o priorytecie wyższym. Operatory + i - mają ten sam priorytet. Jest on mniejszy od takich samych priorytetów operatorów * i /. Kolejność obliczania wyrażeń może być zmieniona przez zastosowanie nawiasów okrągłych na zasadach powszechnie obowiązujących dla tego typu wyrażeń.

```
float   X, Y, Z[2];
int     I;
long LL;
double  Zm_X;
X :=(2+3)*(4-1);
Y :=15-X*2;                  //(Y=-15)
Z[1]:=0;
Z[0]:=2*sin(Z[1])+Y;         //(Z=-15)
I := 1;
Zm_X := I + Y*1.45;
X := I;
X := Zm_X;                     // błędne przypisanie bardziej dokładniejszego typu
                               // do mniej dokładnego
I := LL;                       // błąd
LL := I;                       // poprawne wyrażenie
```

W przypadku używania zmiennych różnego typu obowiązuje zasada, że zmienna po lewej stronie musi być zmienną o największym zakresie spośród zmiennych występujących po prawej zmiennych (stałych). Zmienne o mniejszym zakresie są przypisywane automatycznie do postaci ogólniejszej i w tej postaci

następuje obliczanie wyrażeń. Kolejność od najbardziej ogólnego (największego zakresu) do najmniejszego jest następująca : double - float - long - int. Oznacza to, że np. zmiennej typu float może być przypisana zmienna typu long lub int, ale nie double.

Instrukcja złożona

Instrukcje złożone zbudowane są w formie bloku rozpoczynającego się słowem begin i zakończone słowem end. Wewnątrz bloku zawarty jest niepusty zbiór instrukcji prostych. Instrukcje nie będące instrukcjami złożonymi są nazywane instrukcjami prostymi. Instrukcje złożone wykorzystywane są w powiązaniu z niektórymi instrukcjami prostymi, np. for, while, if.

```
begin
        instrukcja_prosta_1;
                -
                -
        instrukcja_prosta_n;
end;
```

Funkcje w języku SPHINX

Jednym z najważniejszych elementów współczesnych języków programowania strukturalnego są funkcje i procedury. Również język SPHINX udostępnia możliwość deklarowania funkcji przez użytkownika (w rozumowaniu np. języka Pascal są to procedury). Funkcje muszą być zadeklarowane na początku bloku control przed jakąkolwiek inną instrukcją. Możliwe jest wcześniejsze zadeklarowanie nagłówków 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. Deklaracja funkcji jest zakończona znakiem średnika.

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

Definiowanie funkcji

Definicja funkcji rozpoczyna się od nagłówka funkcji, czyli podania jej nazwy i ewentualnie (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 wykonajSilnię( long X )

  begin
     long Wynik;
     char Łańcuch;
     silnia( X, Wynik );
     sprintf( Łańcuch, " %ld! = %ld", X,
Wynik );
   messageBox( 0,0, "Silnia", Łańcuch );
   end;
// wykonanie poniższego wywołania
// powoduje obliczenie i wyświetlenie
//  wyniku obliczenia wartości 10!
  wykonajSilnię( 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 wykonywanie 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 (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. 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.

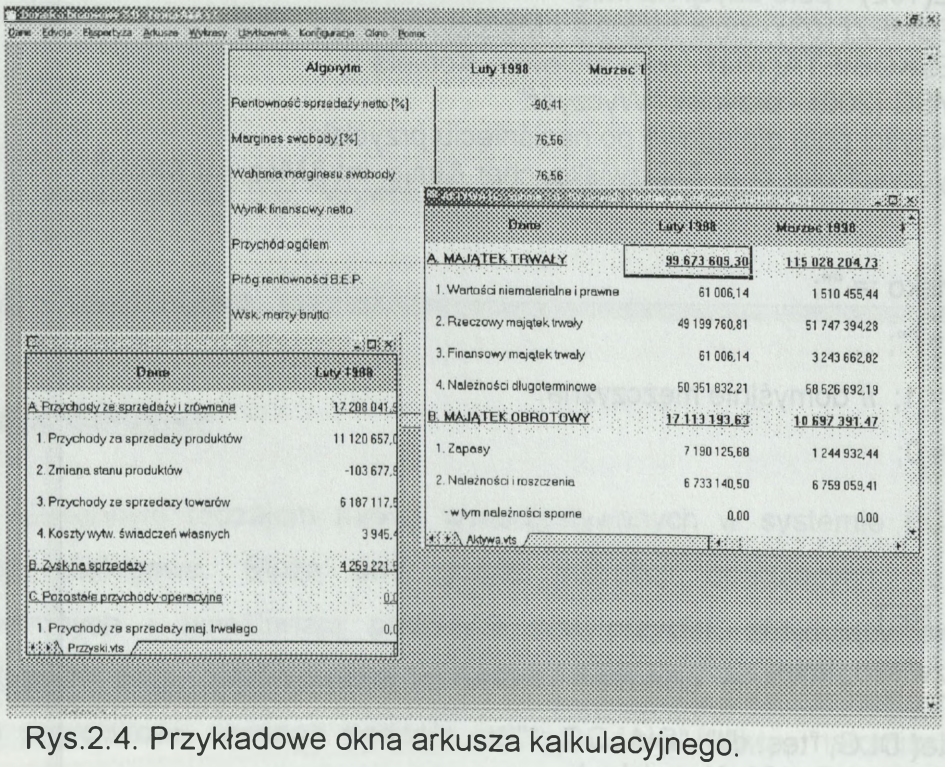
Przykład

```
// Założenie:
// W biliotece test.dll znajduje się definicja okna dialogowego
// o nazwie DIALOG_1, zawierającego pola :
//   ID_NAZWISKO(101) - pole edycji na Nazwisko
//   ID_IMIE(102) - pole edycji na Imię
//   ID_PLECMEZ(103) - pole radiowe - Mężczyzna
//   ID_PLECKOBIETA(104) - pole radiowe - Kobieta
//   ID_CZYSC (200) - przycisk "Wyczyść"
// Funkcja czyści pola dialogowe po naciśnięciu przycisku
function wyczyśćPola( char &Nazwisko, char &Imie, int &Mez, int &Kob )
  begin
      Nazwisko := "";
      Imie := "";
      Mez := 1;  // domyślnie mężczyzna
      Kob := 0;
  end;
....
   int DLG, RET;
   char Nazwisko, Imie;
   int K,M;
   dlgCreate( DLG, "test.dll", "DIALOG_1" );
   dlgBindEdit( DLG, 101, Nazwisko );
   dlgBindEdit( DLG, 102, Imie );
   dlgBindRadioButton( DLG, 103, M );
   dlgBindRadioButton( DLG, 104, K );
   dlgBindButton( DLG, 200, "wyczyśćPola" );
   Nazwisko := "Kowalski"; // domyślne wartości
   Imie := "Jan";
   M := 1;
   K := 0;
   dlgExecute( DLG, RET );
....
```

2.1.7. Współpraca PC-SHELL z innymi aplikacjami.

Arkusze kalkulacyjne

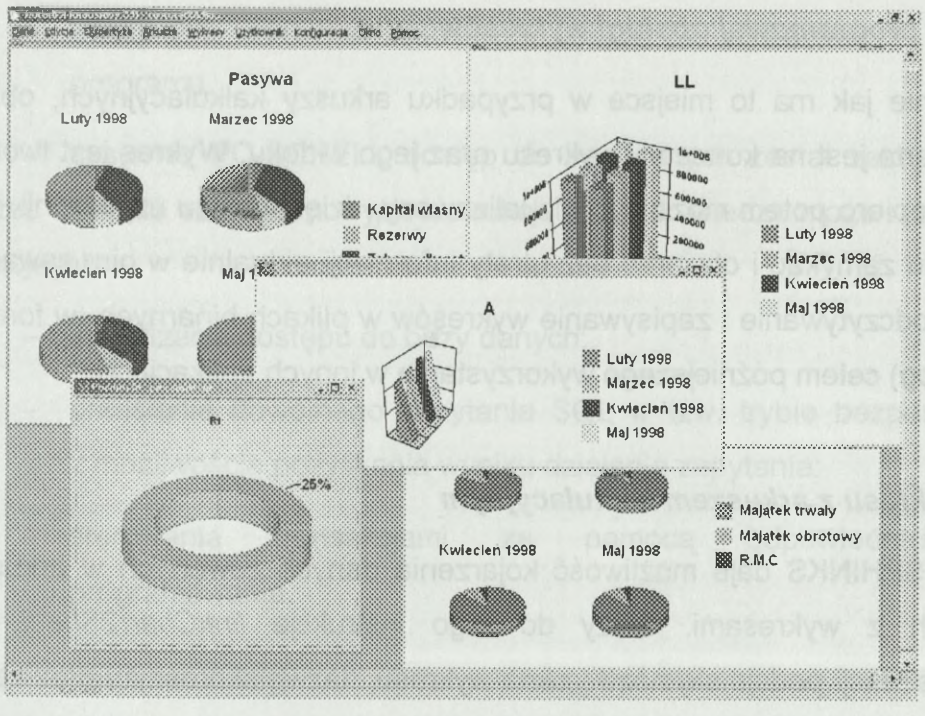
Język SPHINX wyposażony jest w instrukcje umożliwiające wykorzystanie w aplikacjach arkuszy kalkulacyjnych (np. MS EXCEL). Z poziomu aplikacji PC-SHELL można otworzyć okno zawierające arkusz kalkulacyjny, a następnie wykorzystać możliwości arkuszy do gromadzenia danych, ich przeliczania i zapisywania w postaci plików. Język SPHINX dysponuje mechanizmami programowej obsługi arkuszy kalkulacyjnych. Udostępnia instrukcje dostępu do pojedynczych komórek, zakresów, a także całych arkuszy.



Rys.2.4. Przykładowe okna arkusza kalkulacyjnego.

Wykresy

Pakiet SPHINKS posiada możliwość programowej obsługi wykresów. Pod pojęciem wykresu rozumieć należy okno zawierające graficzną formę reprezentacji danych. Każdy wykres identyfikowany jest przez swoją unikalną nazwę, używaną w instrukcjach operujących na wykresach. Nazwa ta stanowi zarazem etykietę okna.



Rys.2.5. Przykładowe wykresy w PC-SHELL.

Podstawowe elementy wykresu, które mogą się w nim pojawić to:

- pole wykresu;
- tytuł wykresu;
- stopka;
- legenda.

Poszczególne elementy wykresu (za wyjątkiem oczywiście samego wykresu) mogą być widoczne lub nie. Edycja wykresów może być wykonywana na przykład za pomocą bazy wiedzy *wykresy.bw* dostarczanej wraz z pakietem SPHINKS. Dla każdego z elementów można zmienić treść informacji, rodzaj, wielkość i kolor czcionki oraz szereg innych parametrów. Część z opcji może być wykorzystana w sposób programowy (z poziomu języka SPHINKS), dostęp do pozostałych możliwy jest za pośrednictwem menu podręcznego pojawiającego się po kliknięciu na wykresie prawym przyciskiem myszy. W trybie edycji wykresu pojawia się wtedy opcja *Designer,* która umożliwia wywołanie standardowego okna właściwości wykresu (ponieważ okno właściwości stanowi element biblioteki funkcji obsługujących wykres, nazwy poszczególnych opcji podane są w języku angielskim).

Mechanizmy programowej obsługi wykresów

Podobnie jak ma to miejsce w przypadku arkuszy kalkulacyjnych, obsługa wykresów oparta jest na koncepcji wykresu oraz jego widoku. Wykres jest tworzony w pamięci i dopiero potem może być wizualizowany, dzięki czemu użytkownik może widok wykresu zamykać i otwierać bez utraty informacji aktualnie w nim zawartych. Możliwe jest odczytywanie i zapisywanie wykresów w plikach binarnych (w formacie *bmp*, *gif* lub *jpg*) celem późniejszego wykorzystania w innych aplikacjach.

Łączenie wykresu z arkuszem kalkulacyjnym

Język SPHINKS daje możliwość kojarzenia danych zawartych w arkuszach kalkulacyjnych z wykresami. Służy do tego instrukcja *linkChart2Sheet*. Przy wywołaniu instrukcji podaje się nazwę okna wykresu, nazwę skoroszytu oraz żądany zakres danych, a także określa tryb połączenia wykresu z arkuszem:

- 0 – zerwanie powiązania z arkuszem;
- 1 – powiązanie z arkuszem oraz aktualizacja wartości danych;
- 2 – powiązanie z arkuszem z przeformatowaniem wykresu, a więc uaktualnieniem ilości wierszy i kolumn na wykresie zgodnie z wielkością wskazanego zakresu danych.

Bazy danych

System PC-SHELL posiada mechanizm dostępu do konwencjonalnych baz danych. Zastosowane w systemie rozwiązanie opiera się na dostępie poprzez mechanizm ODBC (ang. *Open Database Connectivity*). Mechanizm ten udostępnia dostęp do dowolnego systemu zarządzania bazą danych (ang. *DBMS*) pod warunkiem posiadania odpowiednich interfejsów (ang. *drivers*) dostarczanych z reguły przez producentów systemów zarządzania bazami danych.

Zalety mechanizmów ODBC to:

- niezależność od różnych baz danych (w taki sam sposób odwołujemy się do baz np. firmy Oracle i np. szeroko stosowanych baz typu dBase);
- pełna otwartość poprzez wykorzystanie do komunikacji z bazami standardowego języka zapytań SQL;

- możliwość tworzenia zapytań w trakcie kompilacji lub w trakcie pracy programu.

W systemie PC-SCHELL dostęp do baz danych został zaimplementowany poprzez dodanie szeregu nowych instrukcji. Inżynier wiedzy może realizować takie operacje jak:

- inicjalizacja dostępu do bazy danych;
- przesłanie dowolnego zapytania SQL w tzw. trybie bezpośrednim wraz z możliwością pozyskania wyniku działania zapytania;
- sterowania transakcjami za pomocą odpowiednich instrukcji programowania.

Proces komunikacji z bazą danych musi być obramowany etapem inicjalizacji dostępu oraz na końcu - etapem zakończenia dostępu. Do tego celu służą instrukcje *sqlInit* i *sqlDone*. Instrukcją do przesyłania zapytań SQL jest instrukcja *sqlQuery*, natomiast instrukcjami służącymi do pobrania danych po wykonaniu zapytania są instrukcje *sqlInitBinding*, *sqlBind* oraz *sqlFetch*. Ostatnią instrukcją związaną z dostępem do baz danych jest instrukcja sterowania transakcjami *sqlTransact*.

Sterowanie transakcjami

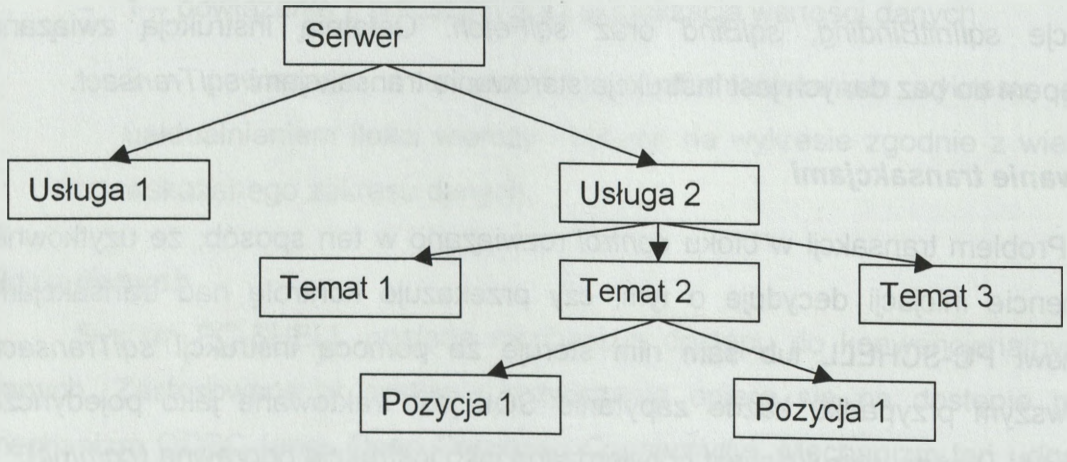
Problem transakcji w bloku *control* rozwiązano w ten sposób, że użytkownik w momencie inicjacji decyduje o tym, czy przekazuje kontrolę nad transakcjami systemowi PC-SCHELL lub sam nim steruje za pomocą instrukcji *sqlTransact*. W pierwszym przypadku każde zapytanie SQL jest traktowane jako pojedyncza transakcja – każda instrukcja jest potwierdzana jako instrukcja poprawna (*commit*).

Dynamiczna wymiana danych (DDE)

System PC-SCHELL może współpracować z innymi aplikacjami wykorzystując mechanizm dynamicznej wymiany danych DDE (ang. *Dynamic Data Exchange*). System ekspertowy PC-SCHELL może pracować zarówno jako klient DDC – wykorzystując zbiór instrukcji języka SPHINKS w bloku sterowania, jak i serwer DDE – udostępniając swoje usługi innym aplikacjom obsługującym DDE.

Mechanizm DDE oparty jest na architekturze „klient – serwer". Aplikacja, która udostępnia swoje usługi innym nazywana jest serwerem DDE, natomiast aplikacja

korzystająca z tych usług nazywana jest klientem DDE. Każdy serwer DDE ma określoną przez producenta strukturę umożliwiającą pracę z dowolnym klientem DDE. Struktura ta jest hierarchiczna i składa się z trzech poziomów. Pierwszy poziom to tzw. „usługa" (ang. *service*), kolejny poziom to zbiór „tematów" (ang. *topics*), w zakresie których dany serwer może nawiązać łączność. Każdy temat z kolei posiada zbiór „pozycji" (ang. *items*), reprezentujących pewne dane, które mogą być przedmiotem wymiany. Przykładowo dla arkuszy kalkulacyjnych tematami są otwarte arkusze, natomiast pozycjami jego komórki lub ich zakresy. Lista pozycji i tematów w trakcie pracy serwera może dynamicznie ulegać zmianie. Dokładne nazwy usług, tematów i pozycji udostępniane są przez producentów serwerów. Standardowo przyjmuje się, że każda usługa udostępnia temat o nazwie *System,* w ramach którego znajdują się między innymi takie pozycje jak *Topic* (udostępnia nazwy tematów udostępnianych przez usługę) oraz *TopicItemList* (zawiera listę pozycji dostępnych w danym temacie). Lista usług z reguły ogranicz się do jednej o nazwie takiej, jak nazwa aplikacji np. Excel, Progman lub PC-SCHELL.



Rys.2.6. Hierarchiczna struktura wymiany danych (DDE)

Komunikacja w ramach DDE opiera się na wykorzystaniu tzw. kanałów. Klient, chcąc nawiązać konwersację z serwerem, musi w pierwszym rzędzie zainicjować kanał, podając jako argumenty nazwę usługi oraz nazwę tematu. W przypadku nawiązania kontaktu pomiędzy aplikacjami ustalany jest uchwyt kanału, którym należy się posługiwać przy każdej kolejnej operacji dotyczącej nawiązywania „konwersacji". Możliwe jest otwieranie dowolnej ilości kanałów jednocześnie, przy czym zwiększenie ilości otwartych kanałów powoduje zmniejszenie wydajności

systemu Windows. Obsługa kanałów od strony serwera DDE jest automatyczna i nie wymaga oprogramowania jej przez użytkownika.

Mając nawiązaną komunikację, aplikacja może wykonywać trzy rodzaje operacji:

- pobieranie danych z serwera (komenda *request*);
- wysyłanie danych do serwera (komenda *poke*);
- wykonywanie poleceń na serwerze (komenda *execute*).

Pobieranie danych może polegać na przykład na pobraniu zawartości komórki w arkuszu kalkulacyjnym, pobranie przez klienta PC-SCHELL zawartości zmiennej bloku sterowania (*control*), pobraniu wartości z aplikacji odczytującej stan urządzeń zewnętrznych (np. jakiegoś miernika lub sterownika przemysłowego). Podobnie wysłanie danych od klienta do serwera może służyć parametryzacji pracy serwera, przygotowaniu wartości pracy serwera itp. Dane pobierane i wysyłane w ramach DDE mogą być zarówno danymi tekstowymi jak i numerycznymi. Wykonywanie komend na serwerze DDE umożliwia zainicjowanie pewnych procesów, makr na serwerze DDE, dzięki czemu można dokonywać obliczeń i innych operacji przewidzianych przez producenta serwera.

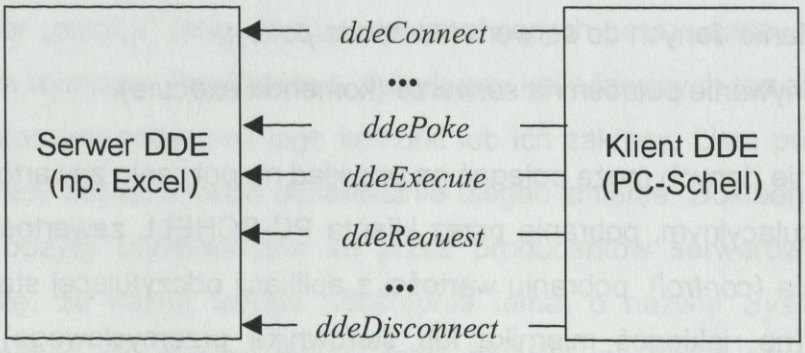
PC-SCHELL jako klient DDE

Praca systemu PC-SCHELL jako klienta DDE polega na możliwości programowego wykorzystania możliwości komunikowania się z innymi aplikacjami, pracującymi jako serwery DDE. Przykładowymi aplikacjami, z którymi może połączyć się PC-SCHELL są arkusze kalkulacyjne (np. Microsoft Excel), edytory tekstów i inne. Poszczególne etapy implementacji obsługi klienta DDE w języku SPHINKS opisane zostaną poniżej i zobrazowane przykładem komunikacji z arkuszem kalkulacyjnym Microsoft Excel. Aplikacja Excel pracuje zarówno jako klient jak i serwer DDE, dlatego można wykorzystać ją do współpracy z systemem PC-SCHELL. Może się ona stać dla nas źródłem danych jak również może stanowić aplikację, do której przesyłamy dane.

Język SPHINKS udostępnia pięć instrukcji, które służą do implementacji funkcji klienta DDE:

- instrukcja inicjująca nawiązanie komunikacji (*ddeConnect*);

- instrukcja kończąca komunikację (*ddeDisconnect*);
- instrukcja przesłania danych do serwera DDE (*ddePoke*);
- instrukcja pobrania danych z serwera (*ddeRequest*);
- instrukcja zainicjowania komendy (*ddeExecute*).



Rys. 2.7. Komunikacja pomiędzy serwerem i klientem

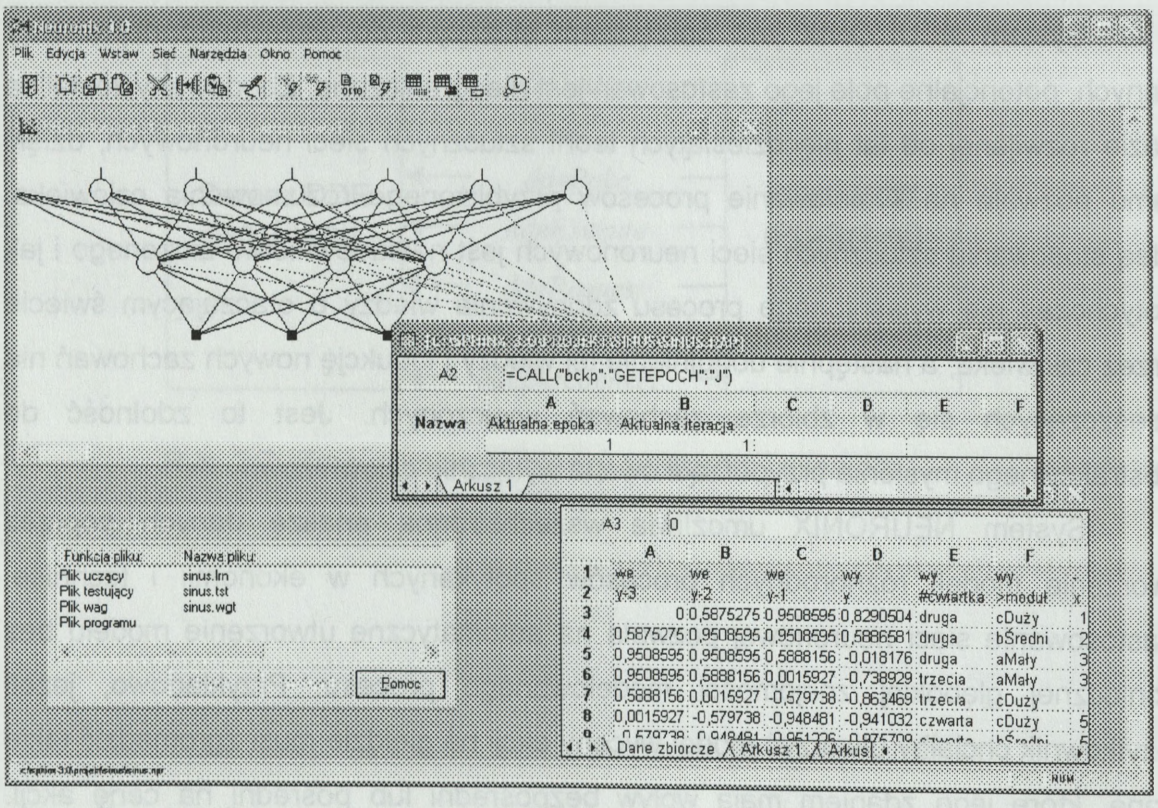
2.2. System do tworzenia sieci neuronowych NEURONIX

2.2.1. Charakterystyka systemu

System NEURONIX jest narzędziem służącym do wszechstronnej analizy danych, potencjalne pole jego zastosowań jest nieograniczone. Zbudowany został na bazie rozwijanej od lat pięćdziesiątych teorii sztucznych sieci neuronowych, dzięki temu pozwala na symulowanie procesów przybliżonego rozumowania człowieka. Ideą stosowania sztucznych sieci neuronowych jest naśladownictwo złożonego i jak dotychczas nie wyjaśnionego procesu zdobywania wiedzy o otaczającym świecie przez człowieka, a następnie uogólnienie tej wiedzy i indukcję nowych zachowań nie mieszczących się w zbiorze zachowań wyuczonych. Jest to zdolność do abstrakcyjnego myślenia przyporządkowana wyłącznie człowiekowi.

System NEURONIX umożliwia wszechstronną analizę danych poprzez tworzenie modeli różnorodnych procesów spotykanych w ekonomii i technice. Zastosowanie sieci neuronowej pozwala na automatyczne utworzenie modelu bez koniecznej głębokiej znajomości modelowanego procesu. Przykładowo, jeżeli Użytkownik chciałby utworzyć model wyceny akcji na giełdzie musi zgromadzić takie dane, które jego zdaniem mają wpływ bezpośredni lub pośredni na cenę akcji. Użytkownik może włączyć do analizy dane które mogą nie mieć żadnego wpływu na cenę akcji, w tym wypadku sieć neuronowa cechuje się zdolnością do rozróżniania danych istotnych od mniej istotnych i sama dokona wyboru. Cały mechanizm budowy modelu sprowadza się więc do zapisania danych w arkuszu kalkulacyjnym podobnie jak to czynimy na przykład w programie Excel w postaci tabeli, w której określimy wejścia i wyjścia jako cenę akcji. Do przykładowych wejść modelu można zaliczyć wycenę innych akcji, poziomy indeksów giełdy, zmiany stóp procentowych i inne czynniki wpływające na kształtowanie się ceny akcji. Znaczącym osiągnięciem twórców pakietu NEURONIX jest opracowanie modułu kodującego, który pozwala tworzyć tzw. modele jakościowe. W modelu jakościowym użytkownik może pracować bezpośrednio na danych w postaci symbolicznej, są to tzw. zmienne lingwistyczne. Oznacza to w praktyce, że jeżeli chciałby opisać aktualny stan giełdy może użyć zmiennej wejściowej której jedną z możliwych wartości jest "umiarkowana hossa", pojęcia tego nie można precyzyjnie wyrazić w postaci konkretnej liczby. W podobny sposób trudno byłoby sprecyzować nastrój inwestorów. Czy można go określić za

pomocą jakiegoś wskaźnika ? Oczywiście jest to możliwe, ale konstrukcja takiego wskaźnika byłaby bardzo złożona i powstałby problem jak to zrobić ? Dużo prościej jest taką daną wprowadzić na wejście sieci i pozwolić jej podjąć samodzielną decyzję o włączeniu jej do budowanego modelu.



Rys.2.8. Przykładowy interfejs Neuronixa.

Program NEURONIX pozwala na bardzo elastyczne posługiwanie się wyuczoną siecią neuronową. Wyuczona sieć neuronowa może być uruchamiana na trzy różne sposoby:

- z poziomu symulatora NEURONIX,
- z poziomu programu napisanego w języku reprezentacji wiedzy SPHINX i uruchamianego w jednym z programów pakietu PC-SHELL, co sprawia, że możliwe jest współdziałanie systemu ekspertowego i sieci neuronowej,
- z poziomu programu wykonywalnego, który może korzystać z bibliotek dynamicznych dll.

2.2.2. Współpraca z innymi aplikacjami pakietu SPHINX

Możliwe jest uruchomienie sieci z poziomu programu napisanego przy użyciu języka reprezentacji wiedzy "SPHINX". Uruchamianie sieci z poziomu bloku "control" bazy wiedzy umożliwia budowanie aplikacji hybrydowych, które wykorzystują system ekspertowy i sieć neuronową. W systemie ekspertowym PC-SHELL można wykorzystywać wszystkie instrukcje odwołujące się do wyuczonej sieci neuronowej. W celu uruchomienia sieci neuronowej muszą istnieć następujące pliki:

- plik projektu: rozszerzenie .npr,
- plik aktualnych wartości parametrów sieci neuronowej: rozszerzenie .cpv ,
- plik skali: rozszerzenie .nn_,
- plik słownika symboli (jeżeli plik uczący zawierał kolumny wejściowe lub wyjściowe z wartościami symbolicznymi): rozszerzenie .dc_ .

Pliki binarne typu .nn_ oraz .dc_ są plikami wygenerowanymi przez symulator NEURONIX. Aby stworzyć wspomniane zbiory, należy uruchomić uczenie sieci lub uruchomić generację plików binarnych.

W języku SPHINX został zdefiniowany rekord NeuralNet, który służy do opisu wejść i wyjść sieci.

```
Record NeutralNet
begin
        char Name;
        double DValue;
        char Symbol;
end;
```

W bloku "sources" można zadeklarować kilka sieci neuronowych, które zostały wcześniej wyuczone. Jako źródło wiedzy traktuje się tu plik projektu (rozszerzenie .npr). Inicjalizacja sieci odbywa się poprzez instrukcję *initNetwork*, której parametrem jest nazwa źródła wiedzy. Uruchomienie sieci uzyskuje się instrukcją *runNetwork*, której pierwszym parametrem jest źródło wiedzy, drugim tablica wejściowa a trzecim

tablica wyjściowa. Zarówno tablica wejściowa jak i wyjściowa są tablicami rekordów *NeuralNet*. W obu przypadkach konieczne jest przypisanie prawidłowych wartości polu "Name", które oznacza nazwę wejścia lub wyjścia sieci. W tablicy wejściowej należy również podstawić odpowiednie wartości w polu "DValue" lub "Symbol" w zależności od rodzaju danych jakie podawane są sieci. Jeżeli wejście operuje na wartościach numerycznych, to wartość liczbową przypisuje się polu "DValue". Jeżeli wejście sieci to jedna z wartości symbolicznych to podstawia się ją pod pole "Symbol". Kolejność wejść i wyjść w tablicy nie jest ważna. Po uruchomieniu sieci następuje przepisanie wartości wyjściowych do pól rekordów tablicy wyjściowej. Jeżeli wyjście jest numeryczne to wypełniane jest pole "DValue" a jeżeli symboliczne to "Symbol". Usunięcie sieci następuje po wywołaniu instrukcji *delNetwork*, której jedynym parametrem jest źródło wiedzy. W bloku "conrol" można zadeklarować dowolną ilość sieci neuronowych i używać ich jednocześnie bez konieczności usuwania sieci po jej użyciu. W ten sposób wyjście jednej sieci może posłużyć jako wejście innej, dając tym samym sposobność do kaskadowego łączenia kilku sieci neuronowych.

2.3. Komputerowy system wspomagania inżynierii wiedzy CAKE

2.3.1. Charakterystyka systemu

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. Możliwości systemu CAKE można opisać następująco:

- zarządzanie projektem aplikacji systemu PC-SHELL;
- wspomaganie procesu tworzenia, rozbudowy i pielęgnacji baz wiedzy;
- weryfikacja poprawności zgromadzonej wiedzy;
- generowanie baz wiedzy w klasycznej postaci tekstowej;
- generowanie baz wiedzy w postaci binarnej;
- ochronę projektu aplikacji systemem uprawnień i haseł;
- wspomaganie organizacji pracy grupowej.

Główne funkcje systemu CAKE:

- tworzenie bazy wiedzy w oparciu o specjalizowany edytor baz wiedzy, dzięki czemu inżynier wiedzy nie musi znać dokładnie języka SPHINX i pisać stosownego tekstu źródłowego;
- realizowanie na bieżąco kontroli poprawności wprowadzonych informacji;
- automatyczna generacja tekstu źródłowego bazy wiedzy;
- efektywne i ergonomiczne zarządzanie bazą wiedzy za pośrednictwem narzędzi w postaci specjalizowanych edytorów bloków baz wiedzy.

Tworzenie nowej aplikacji w systemie CAKE polega na wskazaniu typu aplikacji (aplikacja ekspercka lub baza wyjaśnień), a następnie ustalenie poszczególnych właściwości nowo utworzonej struktury.

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:

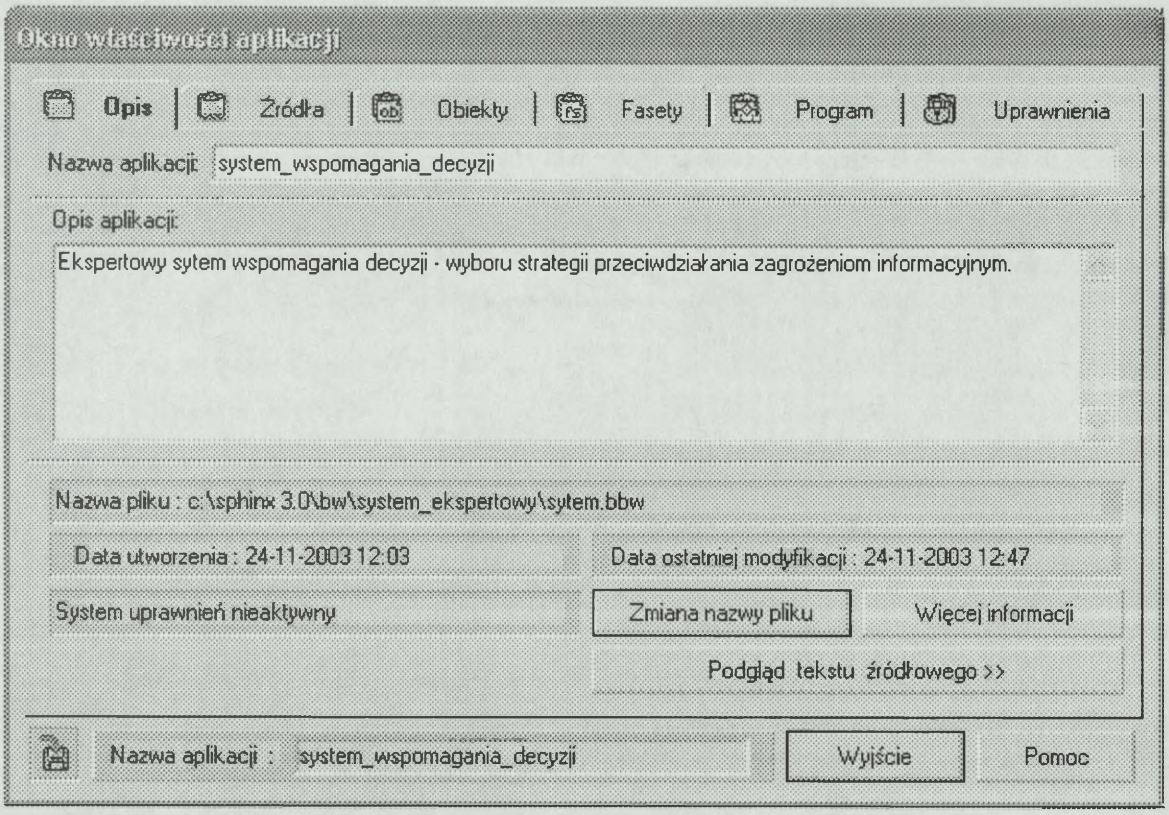
- opis - informacje ogólne na temat aplikacji;
- źródła - wykaz źródeł wiedzy przypisanych do bazy wiedzy;
- obiekty - lista obiektów zdefiniowanych w bazie wiedzy;
- fasety - zestawienie atrybutów i faset zdefiniowanych w bazie wiedzy;
- fakty - wykaz faktów zawartych w bazie wiedzy;
- reguły - wykaz reguł określonych w bazie wiedzy;
- program - tekst bloku sterującego aplikacji;
- uprawnienia - lista zarejestrowanych użytkowników aplikacji.

2.3.2. Okno właściwości aplikacji

Opis

W oknie zamieszczone są ogólne informacje 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.



Rys.2.9. Opcje zakładki „Opis" okna właściwości aplikacji.

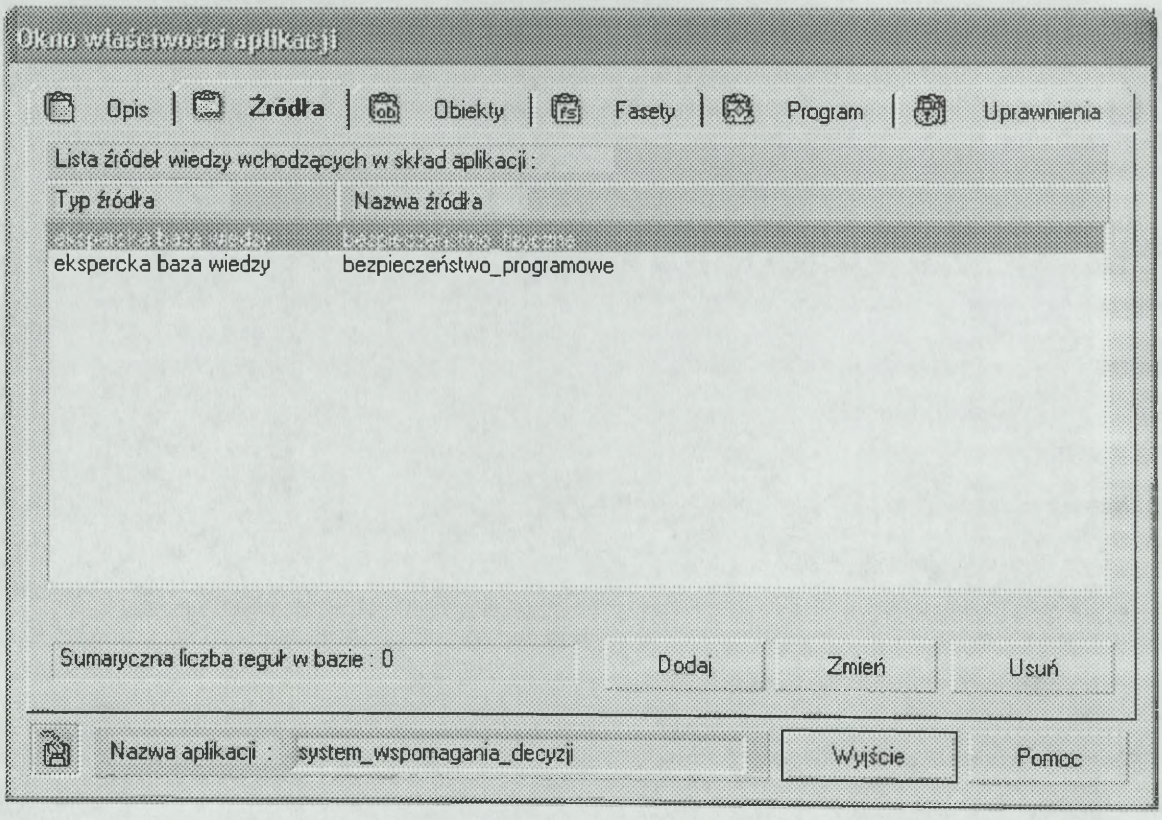
Aby zmienić nazwę pliku bazy wiedzy lub przenieść go w inne miejsce na dysku należy nacisnąć przycisk Zmiana nazwy pliku - spowoduje to otwarcie okna dialogowego umożliwiającego wprowadzenie nowej nazwy i wskazanie katalogu docelowego dla pliku.

Naciśnięcie przycisku „Więcej informacji" powoduje wyświetlenie raportu zawierającego szczegółowe informacje o aplikacji, który może następnie zostać zapisany w pliku tekstowym lub wydrukowany na przyłączonej do komputera drukarce.

Naciśnięcie przycisku „Podgląd tekstu źródłowego>>" umożliwia wygenerowanie tekstu źródłowego bieżącej aplikacji i jego zapis na dysku lub wydruk.

Źródła

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.



Rys.2.10. Opcje zakładki „Źródła" okna właściwości aplikacji.

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.

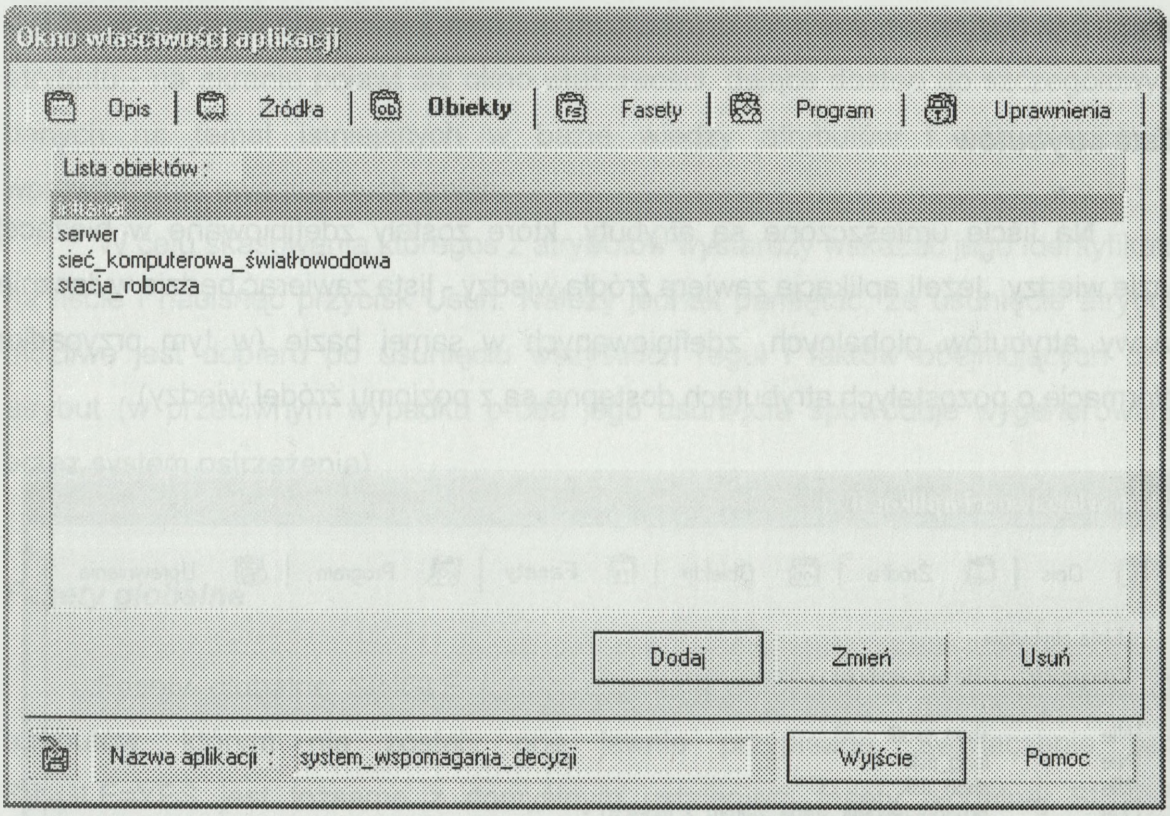
Naciśnięcie przycisku Zmień lub dwukrotne kliknięcie na nazwie wybranego źródła powoduje otwarcie okna właściwości źródła wiedzy, pozwalającego na modyfikację wszystkich elementów wchodzących w skład wskazanego źródła (np. dopisanie nowych reguł, faset, itd.). Należy pamiętać, że w przypadku źródeł typu 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.

Obiekty

W oknie 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).



Rys.2.11. Opcje zakładki „Obiekty" okna właściwości aplikacji.

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.

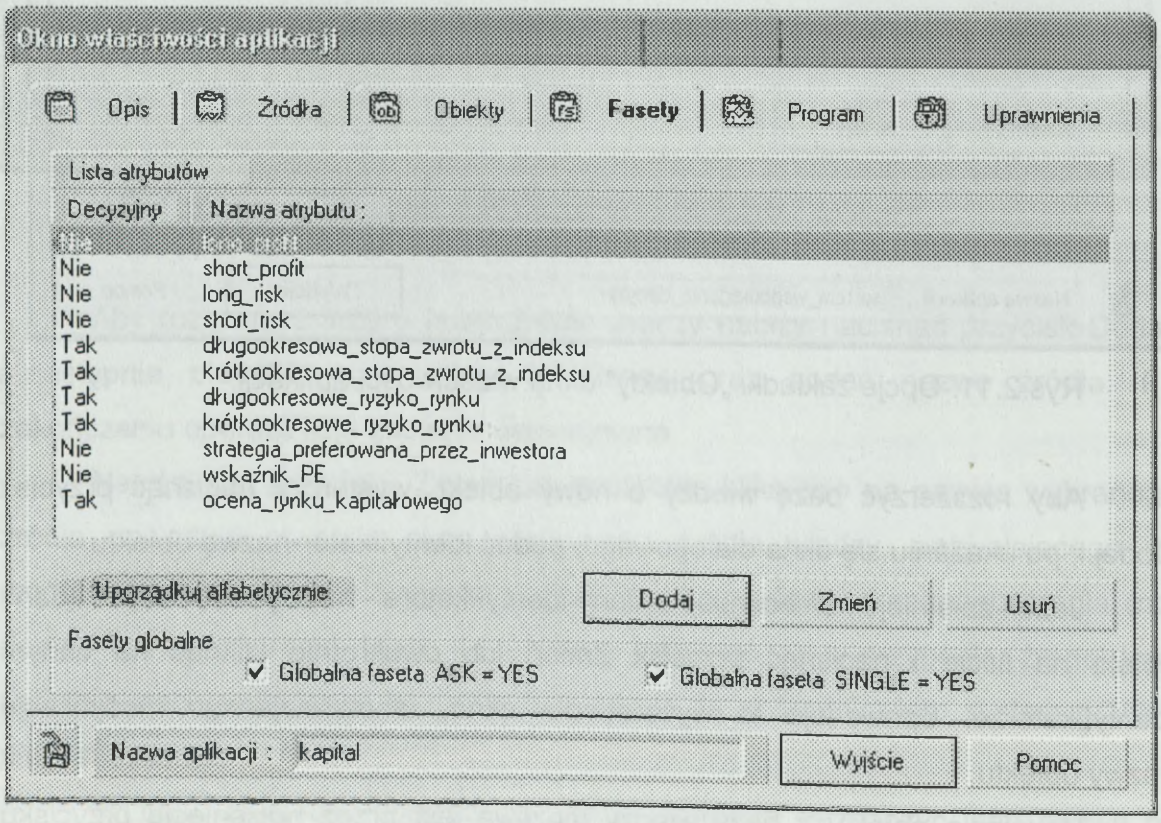
Usunięcie obiektu z bazy wiedzy możliwe jest przez naciśnięcie przycisku Usuń - po zatwierdzeniu operacji wskazany obiekt zostanie skasowany. Należy jednak pamiętać, że usunięcie obiektu możliwe jest dopiero po usunięciu wszystkich reguł i faktów obejmujących dany obiekt (w przeciwnym wypadku próba jego usunięcia spowoduje wygenerowanie przez system ostrzeżenia).

Fasety

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).



Rys.2.12. Opcje zakładki „Fasety" okna właściwości aplikacji.

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

Aby utworzyć w bazie wiedzy nowy atrybut, należy nacisnąć przycisk Dodaj. Po określeniu typu atrybutu, na ekranie wyświetlone zostanie okno właściwości

atrybutów, umożliwiające określenie dalszych szczegółów dotyczących nowo utworzonego atrybutu.

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

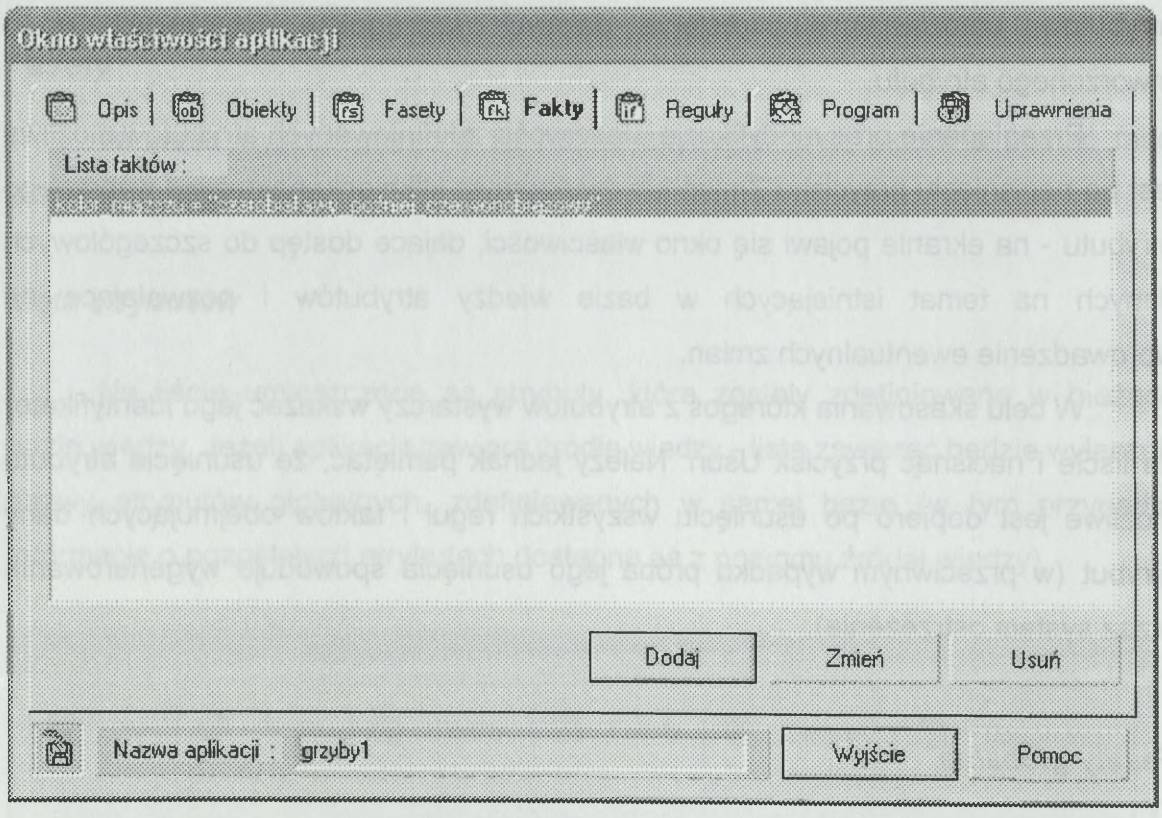
W celu skasowania któregoś z atrybutów wystarczy wskazać jego identyfikator na liście i nacisnąć przycisk Usuń. Należy jednak pamiętać, że usunięcie atrybutu możliwe jest dopiero po usunięciu wszystkich reguł i faktów obejmujących dany atrybut (w przeciwnym wypadku próba jego usunięcia spowoduje wygenerowanie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

Fasety globalne

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

Fakty

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.



Rys.2.13. Opcje zakładki „Fakty" okna właściwości aplikacji.

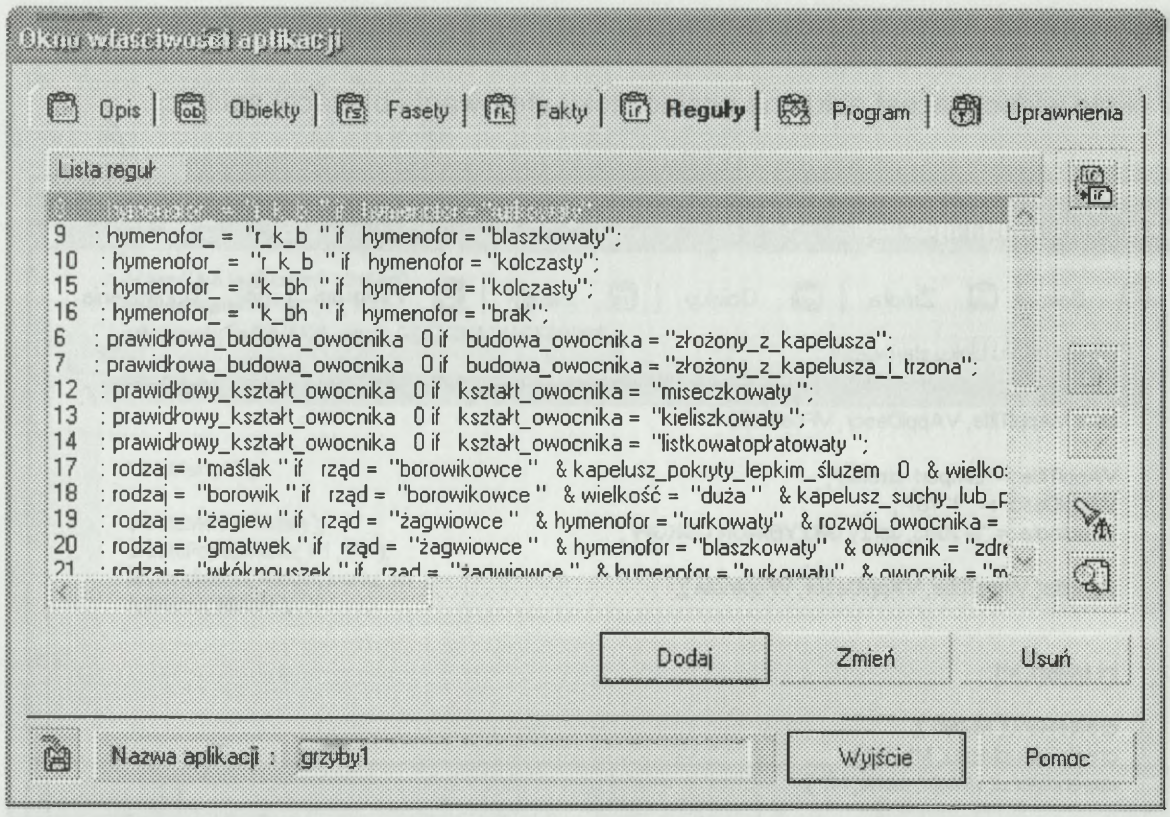
Wprowadzenie nowego faktu do bieżącej bazy wiedzy odbywa się po naciśnięciu przycisku Dodaj. Na ekranie pojawi się okno edycji faktu, umożliwiające określenie postaci nowego faktu.

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.

Reguły

W 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.



Rys.2.14. Opcje zakładki „Reguły" okna właściwości aplikacji.

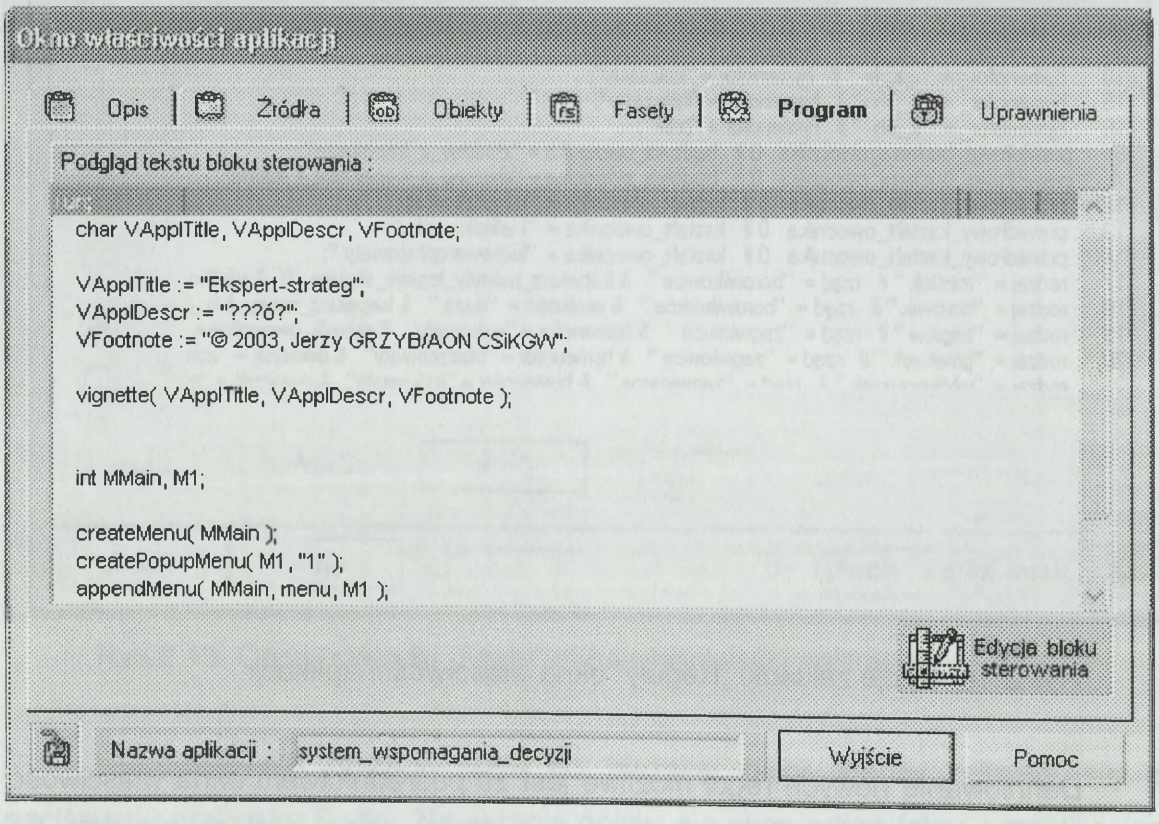
Definiowanie nowych reguł możliwe jest za pośrednictwem okna właściwości reguł. Aby dodać regułę do bazy wiedzy należy nacisnąć przycisk Dodaj, a następnie, po pojawieniu się okna edycji, określić postać nowej reguły.

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.

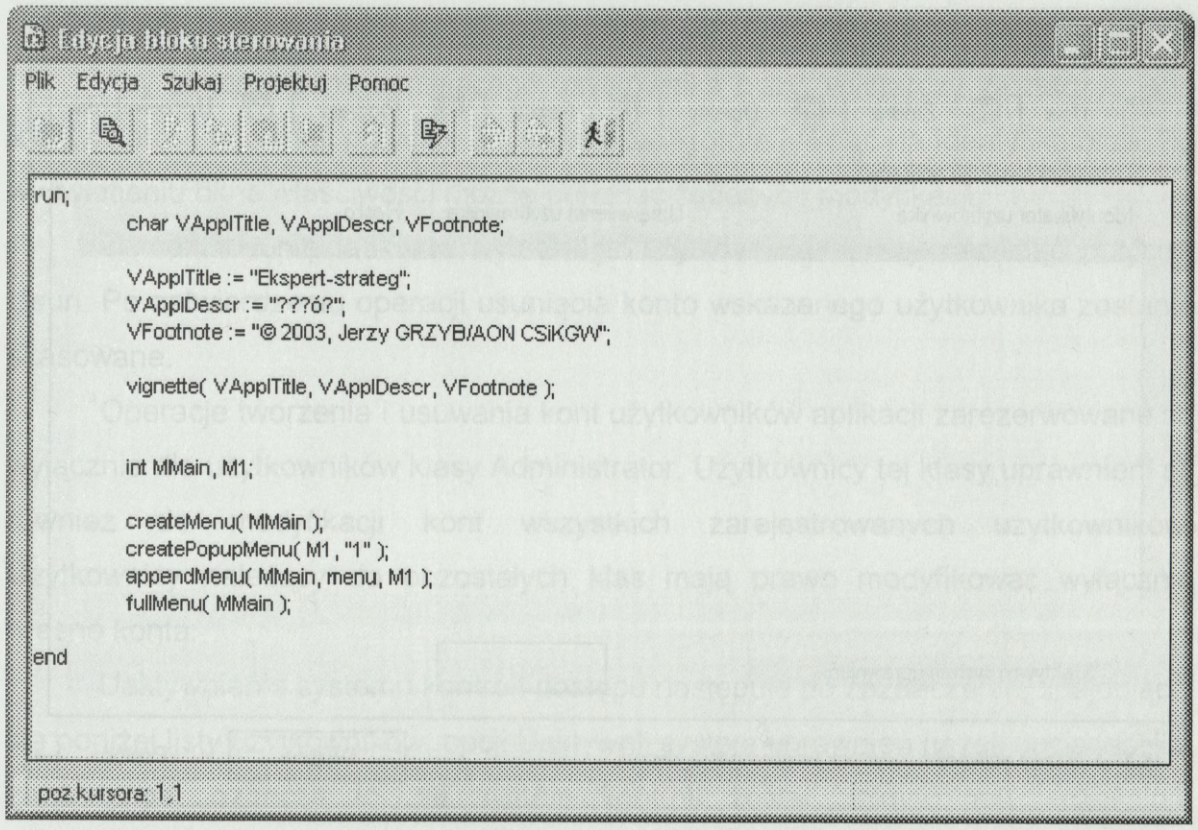
Program

W oknie widoczna jest aktualna postać bloku sterowania bieżącej aplikacji.



Rys.2.15. Opcje zakładki „Program" okna właściwości 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.

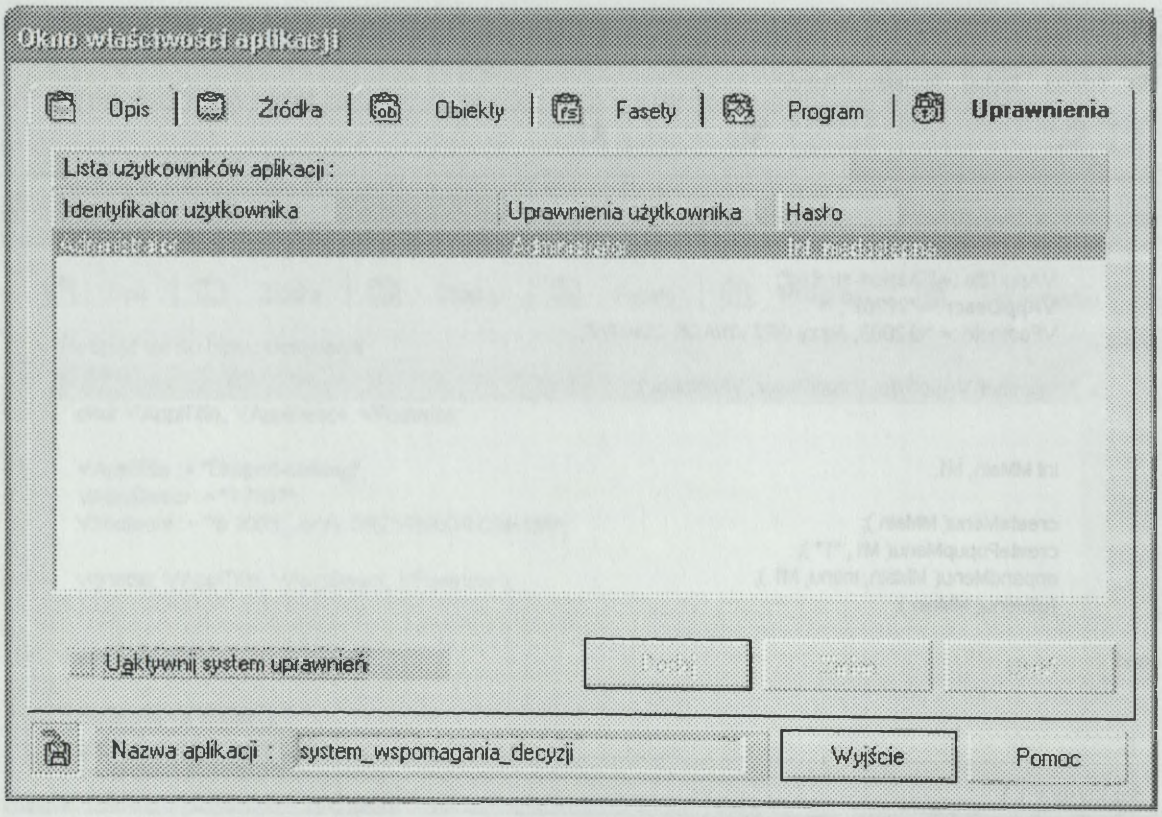


Rys.2.16. Okno edycji bloku sterowania.

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.

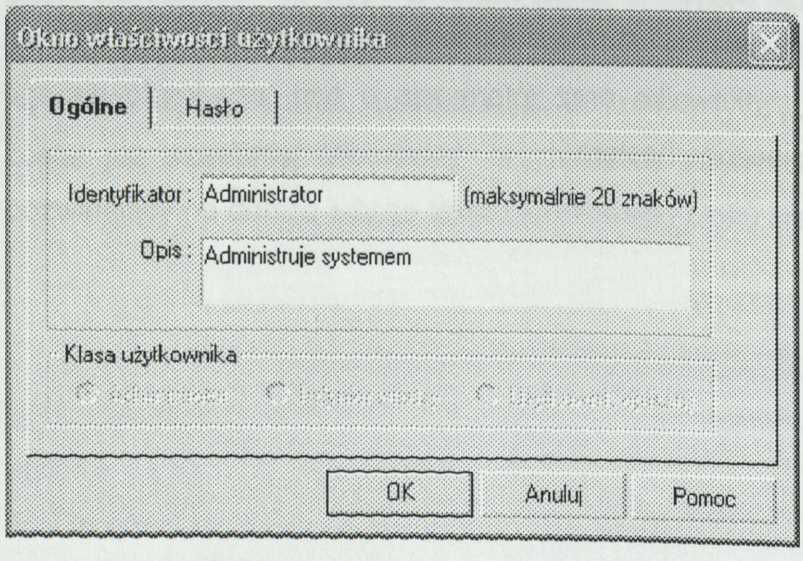
Uprawnienia

Okno systemu kontroli uprawnień zawiera listę użytkowników uprawnionych do otwierania bieżącej bazy wiedzy. Oprócz identyfikatora, na liście znajduje się również określenie klasy użytkownika oraz informacja o tym, czy konto danego użytkownika zostało zabezpieczone hasłem.



Rys.2.17. Opcje zakładki „Uprawnienia" okna właściwości aplikacji.

Dodanie nowego użytkownika bazy wiedzy jest możliwe po naciśnięciu przycisku Dodaj. Na ekranie pojawi się okno właściwości użytkownika, umożliwiające wprowadzenie wszystkich wymaganych informacji, takich jak identyfikator, klasa użytkownika, itd. Każdorazowo po utworzeniu nowego użytkownika ich lista jest automatycznie uaktualniana.



Rys.2.18.Okno właściwości użytkownika.

Aby zmienić dane istniejącego użytkownika bazy, wystarczy zaznaczyć jego identyfikator i nacisnąć przycisk Zmień lub dwukrotnie kliknąć na identyfikatorze. Po wyświetleniu okna właściwości można dokonać żądanych modyfikacji.

W celu usunięcia konta wybranego użytkownika należy nacisnąć przycisk Usuń. Po zatwierdzeniu operacji usunięcia konto wskazanego użytkownika zostanie skasowane.

Operacje tworzenia i usuwania kont użytkowników aplikacji zarezerwowane są wyłącznie dla użytkowników klasy Administrator. Użytkownicy tej klasy uprawnieni są również do modyfikacji kont wszystkich zarejestrowanych użytkowników. Użytkownicy należący do pozostałych klas mają prawo modyfikować wyłącznie własne konta.

Uaktywnienie systemu kontroli dostępu następuje po zaznaczeniu, znajdującej się poniżej listy użytkowników, opcji Uaktywnij system uprawnień (jeżeli opcja jest już zaznaczona, oznacza to, że system kontroli został uaktywniony wcześniej). Bezpośrednio po uaktywnieniu systemu kontroli dostępu zalecane jest zapisanie, zamknięcie i ponowne otwarcie danej bazy wiedzy - pytanie o identyfikator i hasło użytkownika oznaczać będzie, że system kontroli uprawnień funkcjonuje prawidłowo.

2.3.3. Okno właściwości atrybutów

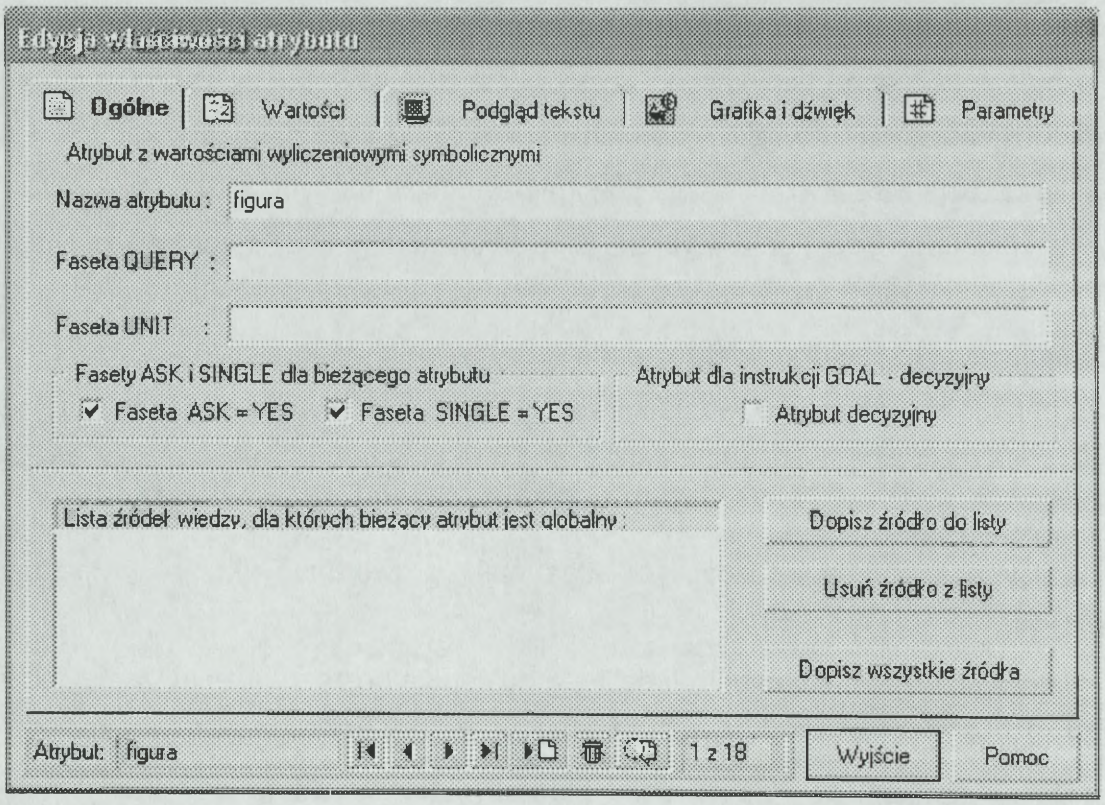
Okno właściwości umożliwia dostęp do szczegółowych informacji o atrybutach zdefiniowanych w bieżącej bazie lub źródle wiedzy.
Dane atrybutów zebrane są w kilku grupach:

-

Ogólne

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.



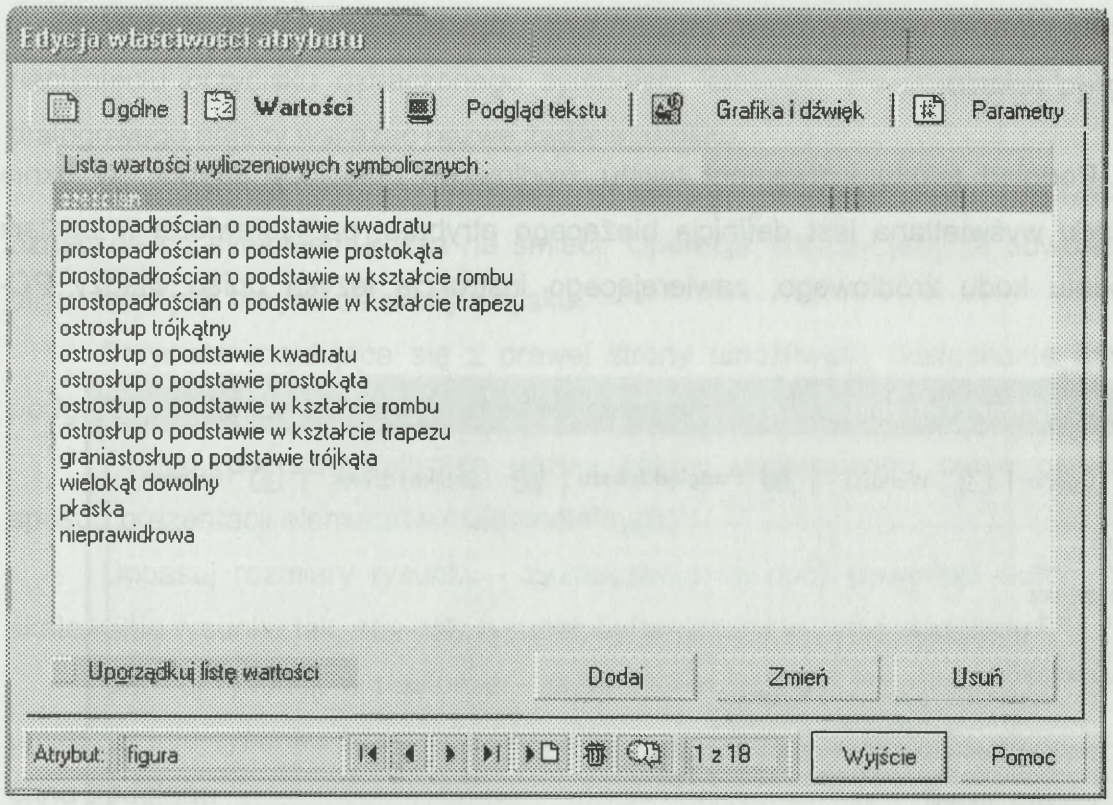
Rys.2.19. Opcje zakładki „Ogólne" okna edycji właściwości atrybutu.

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

Wartości - lista dopuszczalnych wartości wyliczeniowych atrybutu. W oknie wyszczególnione są dopuszczalne wartości wyliczeniowe (numeryczne lub symboliczne) bieżącego atrybutu.



Rys.2.20. Opcje zakładki „Wartości" okna edycji właściwości 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.

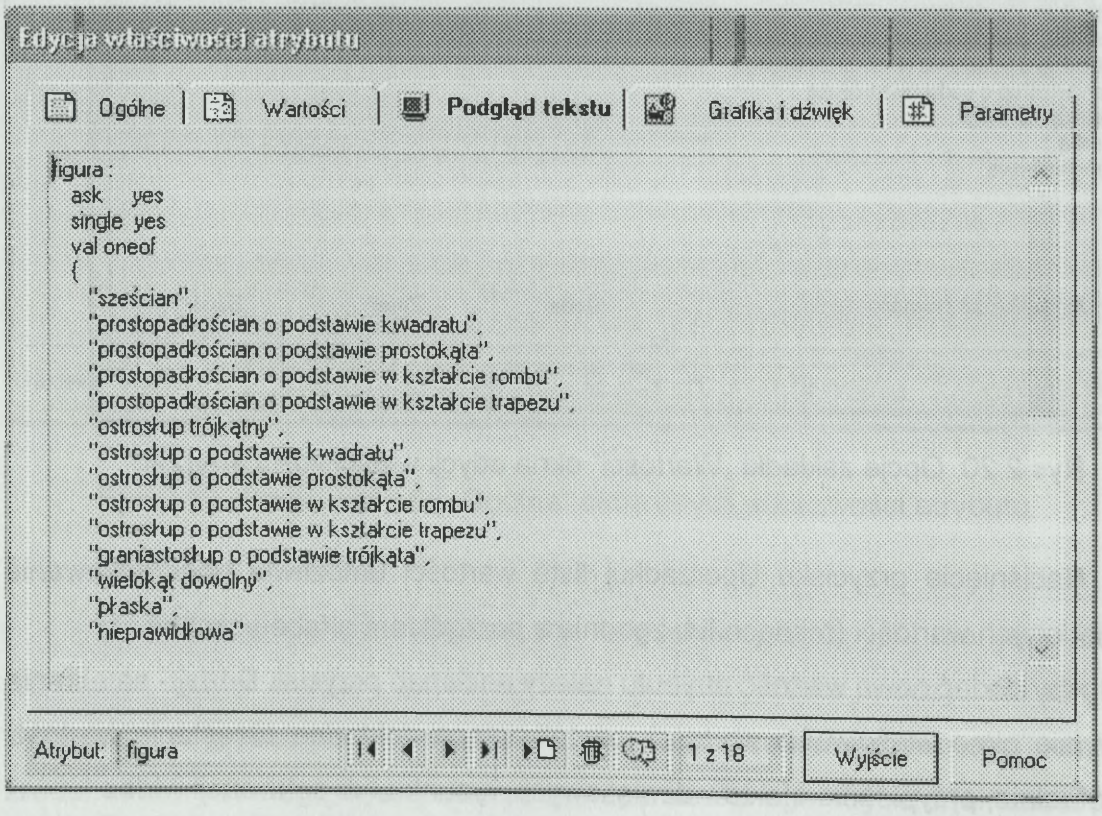
Jeżeli zachodzi potrzeba modyfikacji jednej z wartości, wystarczy wskazać wartość na liście i nacisnąć przycisk Zmień lub kliknąć dwukrotnie na pozycji danej wartości. Po wyświetleniu okna właściwości można przystąpić do wprowadzania wymaganych zmian.

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.

Podgląd tekstu

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.



Rys.2.21. Opcje zakładki „Podgląd tekstu" okna edycji właściwości atrybutu.

Bezpośrednia edycja tekstu w oknie nie jest możliwa.

Grafika i dźwięk

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.

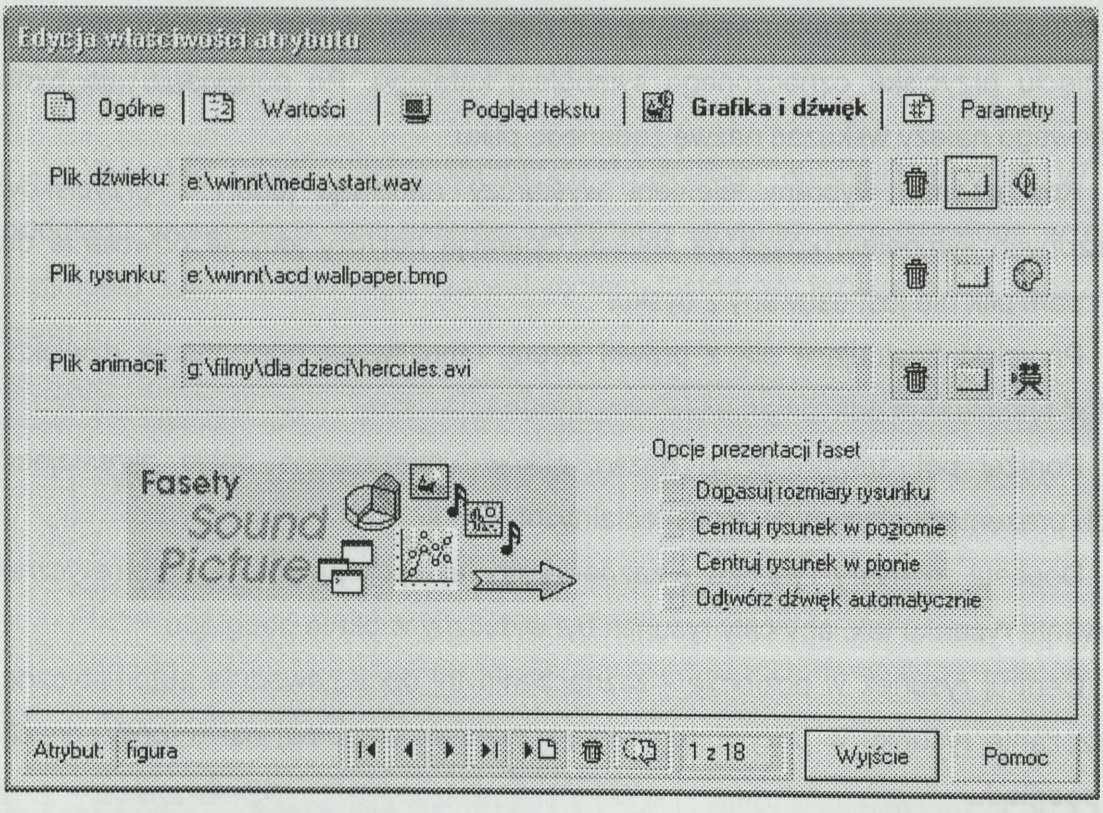
Poniżej pól zawierających nazwy plików umieszczono opcje określające sposób prezentacji elementów multimedialnych:

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.

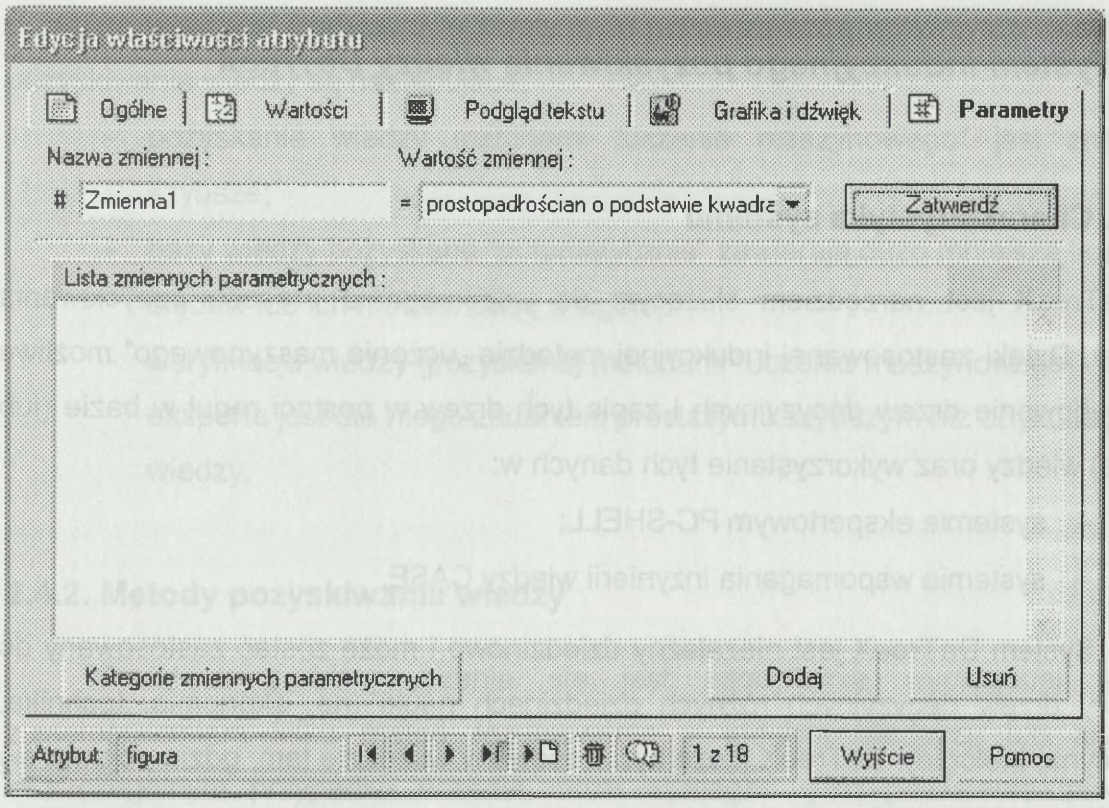


Rys.2.22. Opcje zakładki „Grafika i dźwięk" okna edycji właściwości atrybutu.

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

Parametry

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.



Rys.2.23. Opcje zakładki „Parametry" okna edycji właściwości atrybutu.

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.

2.4. System indukcyjnego pozyskiwania wiedzy DeTreeX

2.4.1. Charakterystyka systemu

DeTreeX jest narzędziem służącym do wspomagania procesu pozyskiwania wiedzy. Dzięki zastosowanej indukcyjnej metodzie „uczenia maszynowego" możliwe jest budowanie drzew decyzyjnych i zapis tych drzew w postaci reguł w bazie (lub źródle) wiedzy oraz wykorzystanie tych danych w:

- systemie ekspertowym PC-SHELL;
- systemie wspomagania inżynierii wiedzy CASE.

System DeTreeX jest niezależny dziedzinowo i może zostać zastosowany do budowy drzew decyzyjnych różnych dziedzinach nauki (np. ekonomia, technika, medycyna, biologia). DeTreeX może być stosowany wszędzie tam, gdzie pojawia się problem:

- podejmowania decyzji (klasyfikacji obiektów);
- szybkiego pozyskiwania reguł decyzyjnych ze zbioru przykładów uczących;
- szybkiej weryfikacji pozyskanych reguł.

Jednym z atrybutów inteligencji jest możliwość uczenia się. Do zadań systemów komputerowych wyposażonych w umiejętność uczenia się należy zaliczyć:

- formułowanie nowych pojęć;
- wykrywanie nieznanych dotychczas prawidłowości w danych;
- tworzenie drzew i reguł decyzyjnych;
- przyswajanie nowych pojęć i struktur drogą uogólniania przykładów i analogii;
- modyfikowanie, uogólnianie i precyzowanie danych;
- zdobywanie wiedzy drogą konwersacji z ludźmi;
- generowanie wiedzy i wyjaśnień zrozumiałych dla użytkownika.

Badania w dziedzinie "uczenia maszynowego" odnoszą się do tworzenia programów komputerowych zdolnych pozyskiwać nową wiedzę i/lub ulepszać wiedzę już pozyskaną na podstawie pewnych informacji wejściowych. Informacje wejściowe takie jak przykłady, fakty, opisy itp. są zwykle wprowadzane na wejście systemu

przez użytkownika. Podstawowe zalety jakie niesie ze sobą wykorzystanie systemu pozyskiwania wiedzy są następujące:

- pozyskanie wiedzy metodami "uczenia maszynowego" jest znacznie szybsze;
- bazy wiedzy pozyskane "automatycznie" zawierają dużo mniejszą liczbę błędów lub ich nie zawierają w ogóle;
- weryfikacja wiedzy (pozyskanej metodami "uczenia maszynowego") przez eksperta jest dla niego zadaniem prostszym i szybszym niż artykulacja tej wiedzy.

2.4.2. Metody pozyskiwania wiedzy

Podstawową ideą uczenia się jest zdobywanie wymaganej wiedzy z zastosowaniem kilku metod rozumowania – indukcji, dedukcji lub analogii. W szczególnym przypadku uczenie może wymagać tylko powielania informacji dostarczonych przez otoczenie lub transformację tej informacji i/lub wydzielenie z niej pewnej istotnej dla nas części. Proces uczenia (pozyskiwania wiedzy) sklasyfikowany jest zależnie od wielu kryteriów. Jednym z takich kryteriów może być ilość informacji przekazanej do systemu doradczego. Można wyróżnić tutaj następujące metody uczenia (pozyskiwania wiedzy):

- *bezpośrednie zapisanie wiedzy* (tzw. uczenie na pamięć) – nie wymaga od systemu podlegającego uczeniu wnioskowania i/lub transformacji wiedzy; realizowane np. przez bezpośrednie zaprogramowanie; metoda stosowana dla prostych baz wiedzy;
- *pozyskiwanie wiedzy na podstawie instrukcji* (tzw. uczenie przez przekazanie informacji) – wymaga konieczności współdziałania uczącego z uczącym się; realizowane poprzez zastosowanie odpowiednich źródeł wiedzy wskazanych przez uczącego i ich przekształcenia na język akceptowalny przez uczącego się;
- *pozyskanie na podstawie analogii* – polega na takim przekształceniu istniejącej informacji aby mogła być użyta do opisów faktów podobnych do tych, które już zostały zawarte w bazie wiedzy; realizowane przez np.: modyfikację programu komputerowego;

- *pozyskiwanie wiedzy na podstawie przykładów* – metoda bardzo często stosowana przy konstruowaniu baz wiedzy; polega na generowaniu ogólnego opisu klas na podstawie zbioru przykładów i kontrprzykładów reprezentujących te klasy; ogólny opis otrzymywany jest na podstawie zasady indukcji;
- *pozyskiwanie wiedzy na podstawie obserwacji* (tzw. uczenie bez nauczyciela) – metoda ta wymaga większego udziału uczącego się podczas procesu uczenia; uczący może dokonywać obserwacji biernych oraz czynnych.

Spośród wielu metod indukcji wiedzy na podstawie przykładów wyróżnić można następujące metody:

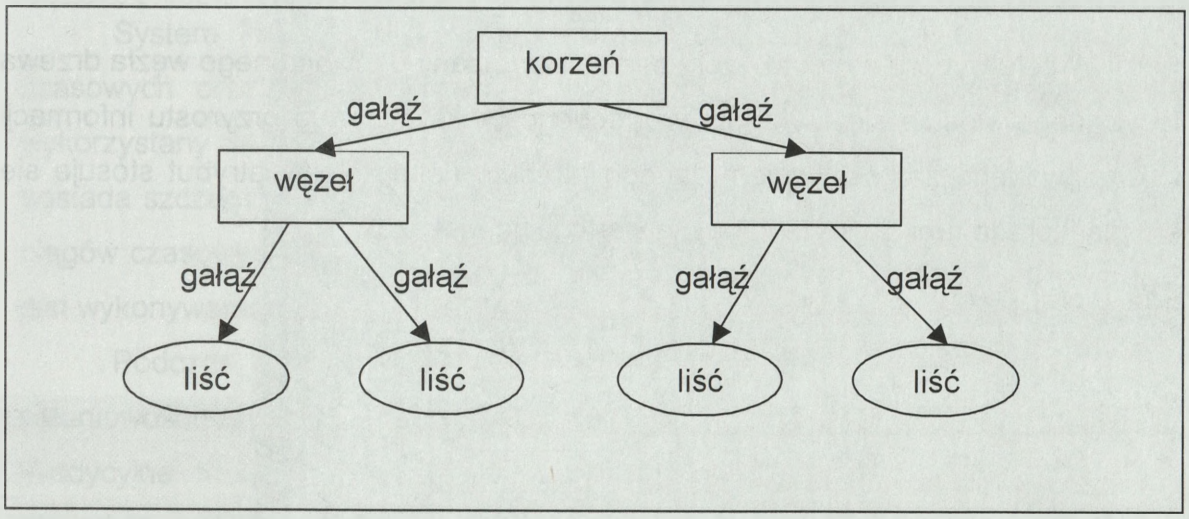
- indukcja reguł za pomocą generowania pokryć;
- indukcja drzew decyzyjnych;
- indukcja reguł za zastosowaniem zbiorów przybliżonych.

W celu zastosowania tych metod konieczne jest użycie tzw. atrybutowego modelu opisu przestrzeni zainteresowania, czyli określenie tzw. dziedziny. Aby możliwe było pozyskanie wiedzy na podstawie przykładów konieczne jest wyszczególnienie:

- obiektów (zjawisk, problemów, itp.) podlegających klasyfikacji;
- atrybutów opisujących dane obiekty (zjawiska, problemy, itp.);
- oraz wartości jakie przyjmą poszczególne atrybuty.

Wartości atrybutów dla wybranej grupy obiektów tworzą tzw. zbiór uczący, a przykłady opisujące obiekty nazywa się przykładami uczącymi.

Drzewem decyzyjnym nazywamy strukturę drzewiastą, której każdy węzeł odpowiada przeprowadzeniu pewnego testu na wartości jednego atrybutu, zaś każdy liść zawiera decyzję o klasyfikacji przykładu. Z poszczególnych węzłów wychodzi tyle gałęzi, ile jest możliwych wyników testu odpowiadających tym węzłom. Każda z tych gałęzi prowadzi do poddrzewa (węzła) służącego do klasyfikacji tych obiektów, dla których ten test ma określony wynik.



Rys. 2.24. Struktura drzewa decyzyjnego.

Głównym problemem w trakcie budowy drzewa decyzyjnego jest określenie kryterium, umożliwiającego wybór atrybutu stosowanego do rozbudowy tego drzewa. W tym przypadku stosuje się tzw. *entropię*. Informacja zawarta w zbiorze przykładów uczących (entropia) jest równa[7]:

$$I(E) = -\sum_{i=1}^{|E|} \frac{|E_i|}{|E|} \bullet \log_2\left(\frac{|E_i|}{|E|}\right)$$

gdzie: E – zbiór przykładów uczących,

$|E_i|$ – liczba przykładów, które opisują i-ty obiekt,

$|E|$ – liczba przykładów w zbiorze uczącym E.

Oczekiwana wartość informacji po podziale zbioru przykładów E na podzbiory $E^{(m)}$, m = 1,...,$|V_a|$, dla których atrybut a ma wartość v_m, określona jest jako[8]:

$$I(E,a) = \sum_{m=1,..,|V_a|, E^{(m)} \neq 0} \frac{|E^{(m)}|}{|E|} \bullet I(E^{(m)})$$

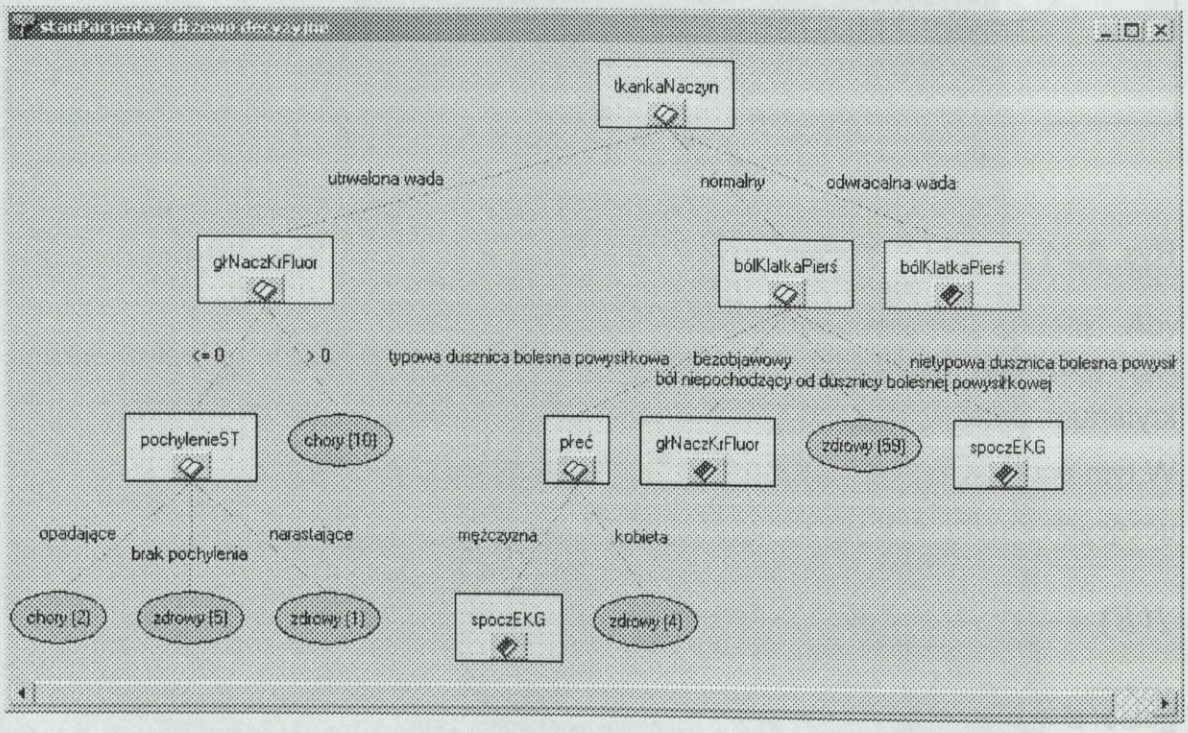
[7] Quinlan J.R., C4.5 Program for Machne Learning, Morgan Kaufmann, San Mateo, CA, 1993
[8] Tamże

gdzie: $|E|$ – liczba przykładów po podziale zbioru E względem wartości m danego atrybutu,

$|E|$ – liczba przykładów w zbiorze uczącym E.

W celu wyboru atrybutu, który będzie przypisany do tworzonego węzła drzewa decyzyjnego stosuje się kryterium względnego maksymalnego przyrostu informacji spowodowanego zastosowaniem danego atrybutu (jako kolejny atrybut stosuje się ten, dla którego funkcja kryterialna ma wartość największą)[9]:

$$\Delta I(E,a) = I(E) - I(E,a)$$



Rys. 2.25. Przykład drzewa decyzyjnego.

[9] Tamże

2.5. System do budowy modeli prognostycznych Predykator

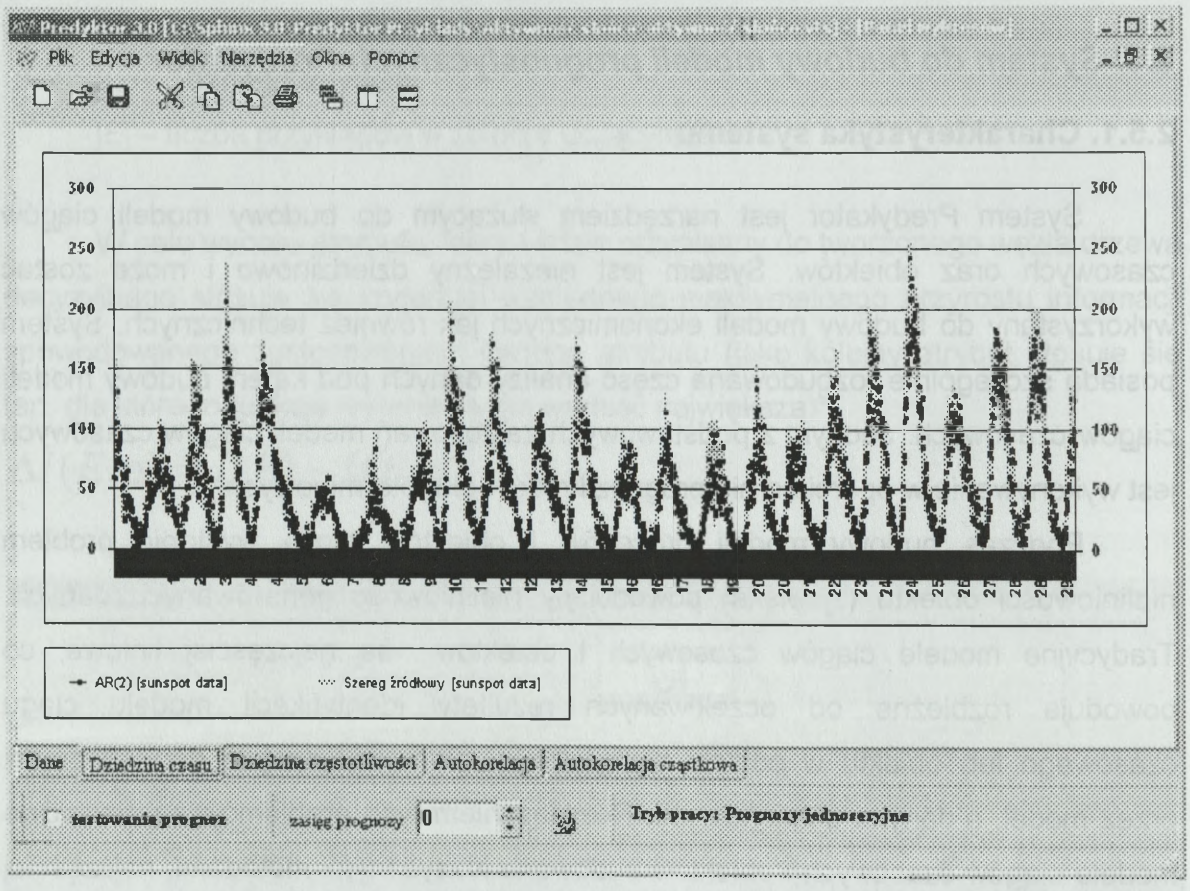
2.5.1. Charakterystyka systemu

System Predykator jest narzędziem służącym do budowy modeli ciągów czasowych oraz obiektów. System jest niezależny dziedzinowo i może zostać wykorzystany do budowy modeli ekonomicznych jak również technicznych. System posiada szczególnie rozbudowaną część analizy danych pod kątem budowy modeli ciągów czasowych. Jednym z podstawowych zastosowań modeli ciągów czasowych jest wykonywanie w oparciu o nie prognoz krótko i długoterminowych.

Podczas budowy modeli procesów i obiektów może wystąpić problem nieliniowości obiektu (zjawiska) powodujący nieliniowość generowanych danych. Tradycyjne modele ciągów czasowych i obiektów są najczęściej liniowe, co powoduje rozbieżne od oczekiwanych rezultaty identyfikacji modelu ciągu czasowego lub obiektu w postaci niezgodności pomiędzy danymi generowanymi przez model, a danymi empirycznymi. Rozwiązaniem tych problemów są nieliniowe modele ciągów czasowych i obiektów budowane klasycznymi metodami, najczęściej złożone obliczeniowo i z trudną do ustalenia strukturą. Wykorzystując system NEURONIX możliwe jest zbudowanie modelu nieliniowego przy pomocy sieci neuronowej. Może to być zarówno model ciągu czasowego jak również model obiektu o nadzwyczaj dobrych właściwościach w zakresie modelowania nieliniowości.

System Predykator umożliwia wykonywanie prognoz dla wielu różnych postaci modeli. Przyjęcie danego trybu pracy powoduje konsekwentne jego stosowanie podczas całego cyklu życia projektu. System może pracować w trzech trybach:

- *dla pojedynczych ciągów czasowych* – w tym trybie można budować modele autoregresyjne oparte o jednowymiarowy ciąg danych wejściowych,
- *dla modeli obiektów* (regresja wielowymiarowa, wieloraka) – w tym trybie można budować model o wielu wejściach oraz jednym wyjściu (tzw. model MISO – Multi Input Single Output),
- *dla wyłącznej identyfikacji wielu ciągów czasowych* – w tym trybie możliwe jest tworzenie modeli analogicznie jak w trybie dla pojedynczych ciągów czasowych, z tą różnicą, że jednocześnie analizowanych jest kilkadziesiąt ciągów czasowych



Rys. 2.26. Przykładowy interfejs Predykatora.

W systemie Predykator zostały wykorzystane wybrane metody z teorii identyfikacji systemów oraz teorii ekonometrii. Dla opisu struktury modelu wykorzystano:

- postać wielomianową - parametry przedstawiane są jako *współczynniki wielomianu*,
- sieci neuronowe – parametry przedstawiane są w postaci *zbioru wag*.

System Predykator posiada narzędzia niezbędne do wykonywania wykresów rozkładów empirycznych w postaci histogramów, jak również umożliwia obliczenie podstawowych miar statystycznych I na ich podstawie wyrysowanie modelu rozkładu normalnego.

Dane wejściowe mogą być przechowywane w arkuszach kalkulacyjnych, a dane wyjściowe mogą być wygenerowane w postaci raportów (w oparciu o wykonane modele ciągów czasowych oraz modele pełne) pozwalających na wizualizację zarówno samych danych, jak również miar statystycznych charakteryzujących te dane.

2.6. Wnioski

W wyniku konfrontacji wymagań ujętych w założeniach budowy i działania ESWD z możliwościami oferowanymi przez pakiet sztucznej inteligencji SPHINX należy stwierdzić, że powyższe oprogramowanie spełnia wymagania stawiane przed systemem eksperckim wspomagania decyzji. Posiada zarówno narzędzia wspomagające pozyskiwanie wiedzy, jak również jej gromadzenie oraz przetwarzanie i prezentowanie.

Do zalet tego systemu należy zaliczyć:

- przygotowanie raportów z prowadzonej analizy;
- prowadzenie treningów osób podejmujących decyzje.
- łatwy opis wspomaganego zagadnienia poprzez wprowadzanie uczestników (podmiotów) według pewnego schematu, ich możliwości i preferencji w określonych parametrach czasowych;
- identyfikację typu (rodzaju) wspomaganego zagadnienia w oparciu o wprowadzony jego opis;
- monitorowanie (zobrazowanie) rozwoju sytuacji wraz z wariantami jej rozwiązania;
- wprowadzanie własnych sugestii analityka zagadnienia, co do możliwych scenariuszy;
- definiowanie własnych kryteriów oceny zdarzeń (akcje i skutki);
- uzyskiwanie w trybie interaktywnym odpowiedzi na pytania dotyczące uczestników (podmiotów, przedmiotów) w różnych przekrojach informacyjnych;
- zapamiętywanie historii sytuacji i poszerzanie bazy informacyjnej o jej przebiegu wraz z graficzną reprezentacją;

Szkieletowy system ekspercki PC-SHELL posiada nowoczesny, przyjazny dla użytkownika interfejs w języku polskim. Dzięki temu praca z programem jest prosta dla każdej z grup użytkowników:

- ekspertów (osoby o wysokich kwalifikacjach w dziedzinie, ale charakteryzujących się niewielką znajomością systemu);

- inżynierów wiedzy (twórcy systemu, doskonale znający system, ale posiadający wiedzę mniejszą niż eksperci);
- zwyczajnych użytkowników (np. ćwiczący - osoby o nieokreślonej znajomości obu składowych).

Pakiet sztucznej inteligencji SPHINX oferuje narzędzia wspomagające użytkownika na każdym etapie tworzenia systemu wspomagania decyzji:

- na etapie gromadzenia wiedzy (Neuronix, CAKE, DeTreeX, Predyktor, PC-Shell)
- na etapie jej przetwarzania (Neuronix, PC-Shell)
- na etapie jej prezentowania (Pc-Shell)

Współpraca aplikacji pakietu z bazami danych, edytorami tekstu, arkuszami kalkulacyjnymi, możliwość tworzenia wykresów, schematów, drzew decyzyjnych, grafów, tabel znacznie poszerza możliwości gromadzenia, przetwarzania i prezentacji wyników działania systemu. Bazy wyjaśnień, okna dialogowe, obrazy, dźwięk i filmy wideo rozszerzają i wzbogacają komunikację pomiędzy systemem a jego użytkownikiem.

3. Specyfikacja istotnych wymagań dotyczących projektowania ESWD w zakresie bezpieczeństwa informacyjnego

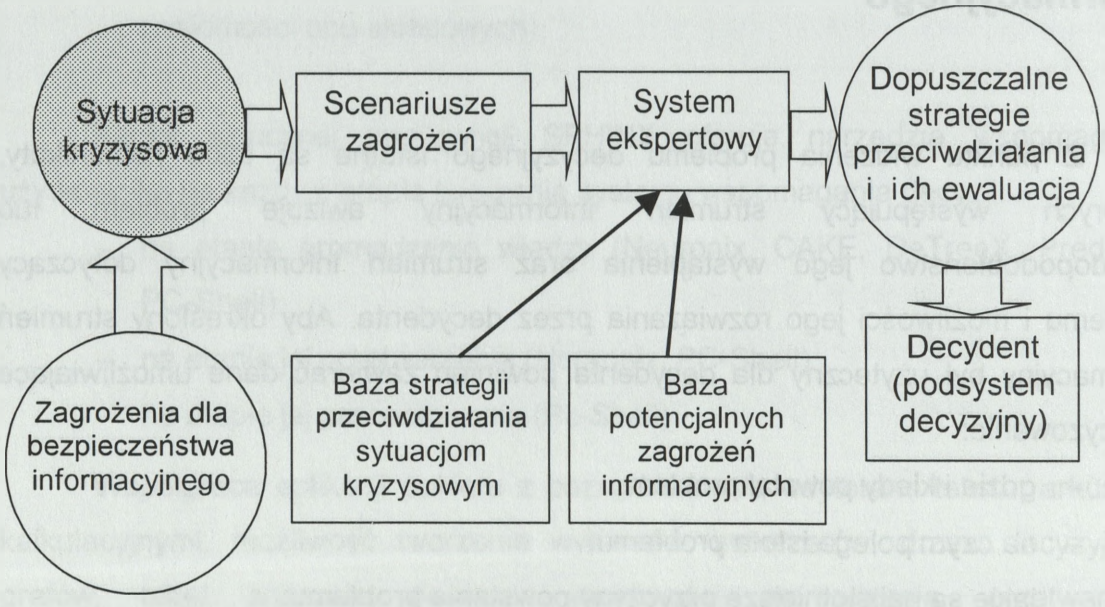
Z punktu widzenia problemu decyzyjnego istotne są takie komunikaty, w których występujący strumień informacyjny awizuje problem lub prawdopodobieństwo jego wystąpienia oraz strumień informacyjny dotyczący problemu i możliwości jego rozwiązania przez decydenta. Aby określony strumień informacyjny był użyteczny dla decydenta powinien zawierać dane umożliwiające sprecyzowanie:

- gdzie i kiedy powstał problem;
- na czym polega istota problemu;
- jakie są najistotniejsze przyczyny powstania problemu;
- jakie skutki wywołuje (może wywołać) problem;
- stopień powtarzalności problemu;
- co jest możliwe do zrobienia w sytuacji problemowej;
- jakie mogą być skutki podjętej decyzji.

Dobrze zaprojektowany i funkcjonujący system elektronicznego wspomagania decyzji w zakresie bezpieczeństwa informacyjnego powinien udzielać odpowiedzi na wyżej przedstawione pytania. Uznając fakt, że bezpieczeństwo informacyjne w dzisiejszym skomputeryzowanym świecie jest wielce skomplikowanym zagadnieniem gdzie nie wszystko da się opisać równaniami, system taki powinien być wspomagany wiedzą ekspertów – powinien być systemem ekspertowym.

Wizja takiego systemu ekspertowego wymagałaby zaangażowania zespołu ekspertów z różnych dziedzin (o ich ilości i specjalnościach decydować powinien szczebel organizacyjny na którym zastosowano system) posiadających możliwość komunikacji za pośrednictwem sieci Internet. W przypadku wystąpienia zagrożenia (problemu) system powinien oceniać sytuację w oparciu o wiedzę ekspertów, określać możliwe strategie przeciwdziałania i w oparciu o określone kryteria dokonywać ich ewaluacji, a następnie przedstawiać strategię optymalną z uwzględnieniem skutków jej zastosowania. W przypadku braku zagrożenia możliwe

powinno być samodoskonalenie systemu i opracowywanie strategii prewencyjnych w oparciu o analizę stanu bezpieczeństwa i hipotetycznych zagrożeń.



Rys. 3.1. System ekspertowy w procesie decyzyjnym w zakresie bezpieczeństwa informacyjnego

Jak widać z przedstawionego powyżej schematu systemu newralgiczną sprawą dla funkcjonowania systemu ekspertowego są bazy potencjalnych zagrożeń i strategii przeciwdziałania (rozumianych jako sekwencje działań realizowanych w przypadku wystąpienia konkretnego incydentu, bądź zagrożenia dla bezpieczeństwa informacyjnego). Bazy te muszą być opracowane przez ekspertów i uwzględniać możliwie wyczerpujące zbiory potencjalnych zagrożeń i możliwych reakcji na nie z uwzględnieniem indywidualnego charakteru zabezpieczanego podmiotu.

3.1. Wyodrębnienie obiektów i zagrożeń w systemie

Społeczeństwo informacyjne jest w swej istocie tworem ludzkim, działającym na wysokim poziomie interakcyjnej złożoności. Jednym z niepożądanych tego efektów jest większa współzależność elementów składowych oraz kruchość i niepewność życia społecznego. Prowadzi to, między innymi do spadku zaufania w stosunku do powstających skomplikowanych struktur zarządzania i współudziału, które często okazują się słabo przystosowane do oczekiwanych elastycznych reakcji. Przewidywany w tym kontekście, wzrastający poziom anomii[10] w społeczeństwie, osłabienie struktury społeczeństwa obywatelskiego przy lawinowo narastających możliwościach technologicznych i umacnianiu się nierówności społecznych, powodują, że potencjalne zagrożenia dla jedności społecznej nie są li tylko iluzoryczne. Fakt nadejścia ery komputerów osobistych przyczynił się do stopniowego przewartościowania hierarchii zagrożeń. Technologie informatyczne uzależniać zaczęła od siebie prawie każdy rodzaj kontaktu, od wielkich inwestycji gigantycznych korporacji zaczynając, a na zwykłych listach przesyłanych pocztą elektroniczną kończąc.

Z roku na rok rośnie również uzależnienie gospodarki światowej od teleinformatyki. Rodzi to wiele pozytywnych efektów współzależności, ale powoduje też wzrost wpływu pojedynczych incydentów na funkcjonowanie całej gospodarki i społeczeństwa. W niektórych dziedzinach takich, jak np. informatyka, łączność, bankowość, finanse itp., powiązania te mają wymiar ponadgraniczny, co powoduje, że skutki ewentualnych zaburzeń mogą rozciągać się poza terytorium jednego kraju.

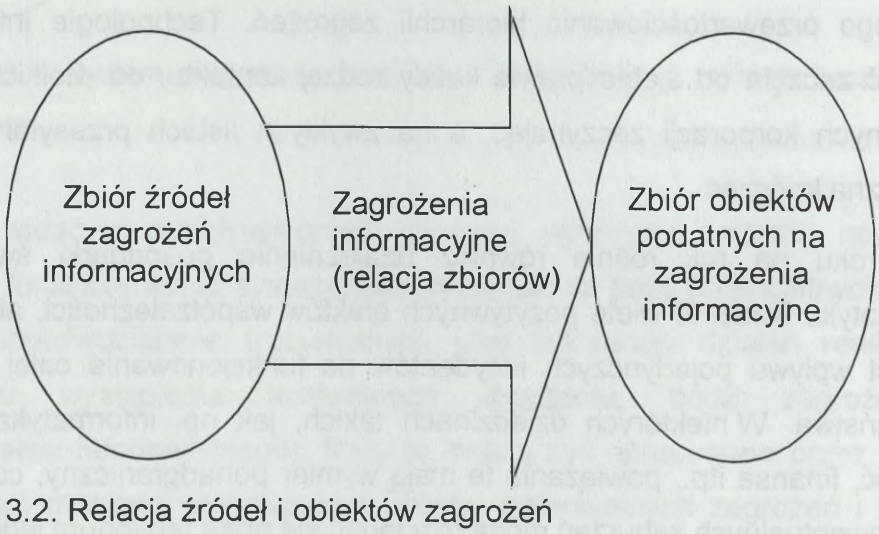
---

[10] **Anomia** (z greckiego "a" - nie, bez, "nomos" - prawo)
1) W teorii socjologicznej stan osłabienia lub wzajemnej sprzeczności norm społecznych - prowadzić może do naruszenia równowagi społecznej, głównie przez niewydolność w kierowaniu zachowaniami jednostek, a w skrajnych przypadkach do rozpadu danego systemu norm jako całości i zaniku więzi społecznej. Wg współczesnych teorii - brak wzajemnego dopasowania akceptowanych w danej społecznosci celów działania i faktycznie dostępnych, a przy tym dopuszczalnych środków ich realizowania przez jednostki.
2) W psychologii jedna z form alienacji: stan apatii, zniechęcenia, poczucia utraty celu działania i życia, wiąże się z ryzykiem samobójstwa. Wiąże się ze zjawiskiem anhedonii, niezdolności do odczuwania radości i satysfakcji. Pojawia się w nerwicach, zaburzeniach osobowości, w przebiegu chorób afektywnych, depresji endogennej czy schizofrenii. Występuje jednak, zwykle jako reakcja sytuacyjna u osób zdrowych. Ostatnio obserwuje się także anhedonię indukowaną. Osoby wchodzące w struktury, grupy kultowe czy religijne często podporządkowują się wymogom skromności i powściągliwości emocjonalnej, dokonując swoistej "autoamputacji" uczucia radości i satysfakcji czerpanej z codziennego życia, wyzwalając jednocześnie u siebie stany ekstazy religijnej i satysfakcji w kontaktach z grupą.

Wraz ze wzrostem znaczenia systemów i sieci teleinformatycznych dla prawidłowego funkcjonowania państwa, jego administracji i podmiotów gospodarczych, rośnie również potencjalne ryzyko przeprowadzenia ataku terrorystycznego, którego celem byłoby sparaliżowanie obiegu informacji.

W ostatnich latach lawinowo rośnie liczba incydentów zagrażających bezpieczeństwu, w systemach i sieciach teleinformatycznych, stopień ich złożoności, agresywności i skala negatywnych skutków.

Rozważając zależności pomiędzy potencjalnym przedmiotem zagrożenia informacyjnego, zagrożeniem - jako swoistym desygnatem narzędzia walki (kooperacji negatywnej) i podmiotem tego zagrożenia (niekoniecznie rzeczywistym), stwierdzić należy, że zasadniczym determinantem zaistnienia sytuacji zagrożenia, jego stopnia i rodzaju jest istnienie przedmiotu (obiektu) potencjalnego szkodliwego oddziaływania - podatnego na zagrożenia informacyjne.



Rys. 3.2. Relacja źródeł i obiektów zagrożeń

Zagrożenie informacyjne potraktować można jako relację zbioru obiektów podatnych na zagrożenia informacyjne i zbioru źródeł tych zagrożeń. Relacjom tym można przypisać pewne cechy np. prawdopodobieństwo i założyć „interwencję" systemu w przypadkach gdy prawdopodobieństwo to (proste lub skumulowane) przekroczy założoną wartość progową.

$$p_{zo}(\tau) \geq p$$

Gdzie: p_{zo} – prawdopodobieństwo zagrożenia obiektu;

τ – określenie przedziału (momentu) czasu;

p – prawdopodobieństwo progowe.

Trudno jest jednoznacznie określić co jest lub co może być obiektem zagrożenia informacyjnego. Pozornie jest to bardzo proste, jest to jakiś system informacyjny lub bardziej ogólnie system, którego prawidłowe funkcjonowanie zależy od sprawności jego podsystemu informacyjnego. Biorąc jednak pod uwagę stopień zintegrowania żywotnych organów zabezpieczających funkcjonowanie społeczeństwa trudno jest znaleźć dziedzinę (obiekt – nie tylko rzeczywisty), który nie podlegałby takim zagrożeniom. Obiektem może być pojedyncze rzeczywiste urządzenie lub też skomplikowana niejednorodna struktura obejmująca swoim zasięgiem nawet wiele państw.

W skali kraju, w tym kontekście w kategoriach bezpieczeństwa państwa używa się pojęcia infrastruktura krytyczna. Pojęcie „infrastruktura krytyczna państwa" ma ścisły związek z funkcjonującym ogólnym pojęciem infrastruktury. W jej zakres wchodzi zespół podstawowych urządzeń i instytucji usługowych niezbędnych do należytego funkcjonowania produkcyjnych działów gospodarki[11]. Samo pojęcie rozpatrywane również może być w dwóch aspektach:

- ekonomicznym; obejmować wówczas będzie urządzenia z dziedziny transportu, komunikacji, energetyki itp.;
- socjologicznym; infrastruktura społeczna, obejmować wówczas będzie instytucje z dziedziny prawa, bezpieczeństwa, kształcenia i oświaty czy służby zdrowia (itp.).

W ujęciu specjalistów od spraw bezpieczeństwa pojęcie *infrastruktura krytyczna państwa oznacza obiekty (budynki i budowle) i urządzenia, służby odpowiedzialne za obsługę tych obiektów i urządzeń, komputerowe systemy informatyczne istotne dla bezpieczeństwa i ekonomicznego dobrobytu państwa oraz jego efektywnego funkcjonowania.* Swoim zakresem obejmuje:

a) systemy energetyczne, telekomunikacyjne, pocztowe, teleinformatyczne, finansowe i bankowe, zarządzania zasobami wodnymi, dostaw żywności i wody, opieki zdrowotnej, transportowe;

b) usługi w zakresie bezpieczeństwa powszechnego i porządku publicznego;

c) zapewnienie prawidłowego funkcjonowania najważniejszych struktur administracji państwowej i publicznej w sytuacjach nadzwyczajnych zagrożeń (w tym centra zarządzania kryzysowego i stanowiska kierowania);

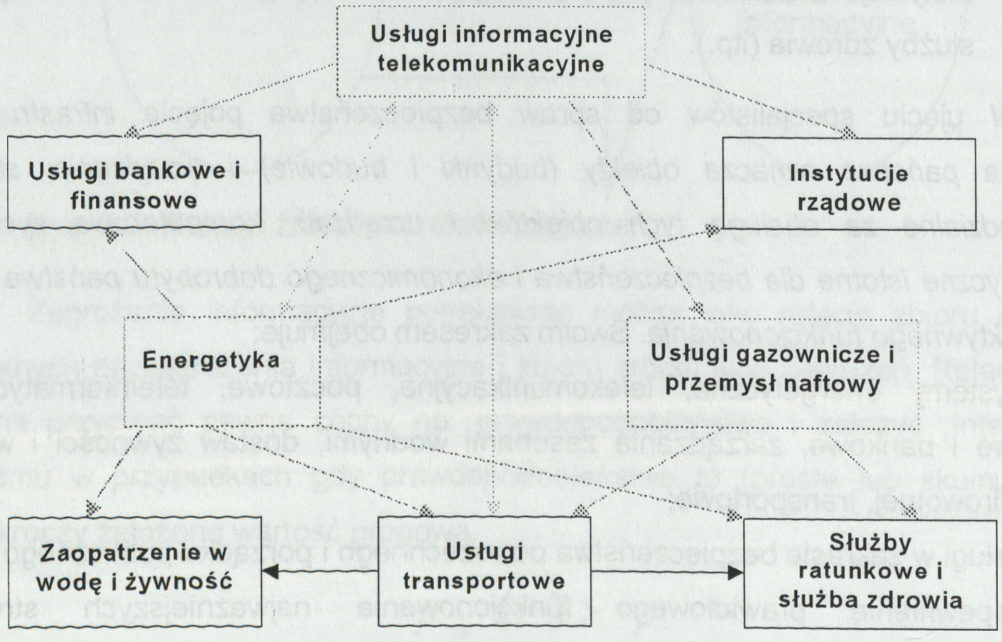
[11] Słownik języka polskiego, Warszawa, PWN 1988, t. 1, s. 185.

d) ochronę przemysłu o strategicznym dla państwa znaczeniu, w tym obronnego.

Ogólnie rzecz ujmując, krytyczna infrastruktura państwa obejmuje obszary zarówno publiczne, jak i prywatne:

- telekomunikacji;
- systemy energetyczne;
- systemy bankowe i finansowe;
- transport;
- systemy zaopatrywania w wodę;
- system opieki zdrowotnej;
- ratownictwo.

Ochrona powyższych systemów przez służby państwowe jest zadaniem czysto praktycznym, nie leżącym w sferze wirtualnej. Wraz z rozwojem Internetu oraz powiązanych z nim wirtualnych infostrad zakres przedmiotowy polityki państwa w stosunku do omawianego problemu musiał ulec znaczącej modyfikacji a samo pojęcie uległo redefinicji. Zaistniała potrzeba, a nawet konieczność wypracowania metod postępowania, umożliwiających wykorzystanie zarówno publicznych, jak i prywatnych zasobów w celu ochrony bezpieczeństwa państwa[12].



Rys. 3.3. Graf współzależności sektorów infrastruktury krytycznej.

[12] Wystąpienie szefa Biura Bezpieczeństwa Narodowego z 9 września 2002 r. na seminarium w Pałacu Prezydenckim nt. „Infrastruktura krytyczna w Polsce".

Postęp technologiczny oraz nowoczesne technologie teleinformatyczne wpływają bezpośrednio na zmianę uwarunkowań polityki bezpieczeństwa państwa. Obecnie informacja stała się zasobem strategicznym gospodarki, którym dysponuje państwo i społeczeństwo. Nie ulega wątpliwości, że celem przyszłych ataków mogą być nie tylko obiekty wojskowe, ale również centra informacyjne i kluczowe sieci teleinformatyczne oraz system finansowy państwa.

Zagrożenie informacyjne w skali państwa, dla jego infrastruktury krytycznej jest oczywiście problemem bardzo istotnym, jednak znakomita większość incydentów naruszających bezpieczeństwo informacyjne nie dotyczy bezpośrednio struktur państwowych i w skali kraju jest praktycznie niezauważalna. Nie zmienia to faktu że każde naruszenie bezpieczeństwa informacyjnego niesie za sobą określone negatywne skutki nawet jeżeli występuje w skali lokalnej. Różnica jest tylko w skali problemu i szeroko rozumianych kosztach wystąpienia incydentu. Nie sposób opisać wszystkich możliwych zagrożeń systemu informacyjnego np. lokalnej instytucji, można jednak skupić się na kilku najważniejszych tzw. „słabych punktach".

Jednym z najważniejszych są zbiory przechowywane na twardych dyskach, CD ROM – ach, dyskietkach, archiwach. Podstawowe zagrożenia dla takich zbiorów to:

- kradzież;
- kopiowanie;
- nieupoważniony dostęp.

W przeszłości i częściowo jeszcze obecnie zagrożenia tego typu powiązane były z koniecznością fizycznego dostępu sprawcy. Wiadomym jest ze statystyk dotyczących naruszeń bezpieczeństwa, że najsłabszym ogniwem każdego systemu ochrony informacji we wszystkich instytucjach, bez względu na specjalność, jest człowiek. Zakładając, iż nawet każdy urząd samorządu terytorialnego w niedalekiej przyszłości stanie się elementem programu rządowego „ePOLSKA", będzie konieczne włączenie ich do sieci publicznej ze wszystkimi wynikającymi z tego konsekwencjami. Wypływają z tego kolejne niebezpieczeństwa systemu informacyjnego urzędu. Zaistnieje wówczas możliwość kradzieży, kopiowania oraz nieuprawnionego dostępu do danych bez konieczności wchodzenia do budynku urzędu.

Kolejnym zagrożeniem dla lokalnych systemów informacyjnych są wszelkiego rodzaju katastrofy i kataklizmy rzadko odczuwalne w skali całego, czy znacznej części kraju. Utrata danych, często nieodwracalna, może przysporzyć niesamowitych trudności a ich odtworzenie, często niemożliwe, pochłonąć duże sumy środków finansowych.

Wydaje się że w związku z postępującą decentralizacją węzłowych punktów systemów informacyjnych wspomagających elementy składowe infrastruktury krytycznej państwa, rodzajami przewidywanych zagrożeń w sferze informacyjnej oraz doświadczeniami w tym zakresie, odpowiedzialność za bezpieczeństwo informacyjne również musi być zdecentralizowana przy zachowaniu wysokiej jakość i niezawodności systemów zabezpieczeń. Według niektórych sugestii[13] struktura odpowiedzialności za bezpieczeństwo informacyjne obejmuje 5 zróżnicowanych obszarowo poziomów (patrz tabela).

Obszar oddziaływania	**Podstawowy opis**
Prywatny	Obejmuje prywatną (osobistą) odpowiedzialność związaną z domem i własnością rodziny
(Municipal) Gminny, lokalny	Obejmuje odpowiedzialność lokalnej władzy, małych organizacji i/lub przedsiębiorstw o lokalnym zasięgu
Regionalny (np. województwo)	Obejmuje odpowiedzialność regionalnej władzy, średnich organizacji i/lub przedsiębiorstw o dużym ale krajowym zasięgu
Narodowy (państwowy)	Obejmuje odpowiedzialność władzy państwowej, dużych organizacji i/lub przedsiębiorstw o zasięgu obejmującym cały kraj
Międzynarodowy	Obejmuje współpracę międzynarodową, stosunki trans-graniczne, globalne przedsiębiorstwa o światowym zasięgu i wpływie (konsekwencjach)

Z punktu widzenia możliwości i zasadności zastosowania systemów ekspertowych rozważania ograniczyć można do 3 poziomów z pominięciem prywatnego i międzynarodowego. W takim przypadku rozważania skupione są na obiektach, które dla uproszczenia określić można jako obiekty typu „mikro", „mezo" i „makro". Typ mikro to obiekt obejmujący przykładowo ośrodki administracji samorządowej, małe organizacje lub przedsiębiorstwa o lokalnym zasięgu, a w

[13] Gogela R., Novak L., Sefcik A., Critical Infrastructure Modelling, Security and Protection of Information 2003.

przypadku struktur wojskowych – stanowiska dowodzenia oddziału/ZT. Cechą charakterystyczną tego typu obiektów jest występowanie jednego węzła telekomunikacyjnego (centrali), lokalnej sieci informatycznej i skupienie przestrzenne wrażliwych na zagrożenia informacyjne elementów składowych. Łatwo jest zidentyfikować i zabezpieczyć newralgiczne punkty systemu informacyjnego przed zagrożeniami zarówno naturalnymi jak i zdeterminowanymi. W przypadku tego typu obiektu efektem negatywnego zdarzenia (ataku informacyjnego) może być całkowity paraliż systemu informacyjnego.

Typ mezo to obiekt obejmujący przykładowo ośrodki administracji samorządowej wyższego szczebla (powiat, województwo), średnie organizacje lub przedsiębiorstwa o dużym zasięgu. Cechą charakterystyczną tego typu obiektów jest występowanie większej ilości węzłów telekomunikacyjnych (centrali), zbioru lokalnych sieci informatycznych i rozproszenie przestrzenne (na znacznym ale niekoniecznie bardzo dużym obszarze) wrażliwych na zagrożenia informacyjne elementów składowych. Występuje różnorodność środków przekazywania informacji, wielopodmiotowość zarządzania mediami transmisji i częściowe ich zrównoleglenie. Zidentyfikowanie najbardziej wrażliwych na zagrożenia elementów jest utrudnione ze względu na niejednorodność organizacyjną. Podobnie trudna sytuacja jest z zabezpieczeniem przed zagrożeniami. Poziom tych zabezpieczeń bywa silnie zróżnicowany w zależności od charakteru podmiotu będącego użytkownikiem systemu. W przypadku tego typu obiektu efektem negatywnego zdarzenia (ataku informacyjnego) może być całkowite wyłączenie z działania pojedynczych podsystemów informacyjnych nie mające jednak decydującego wpływu na funkcjonowanie pozostałych podsystemów. Występujące utrudnienia w pracy sieci informacyjnych (poza przypadkami ekstremalnymi – zmasowanych ataków) nie powinny mieć znaczącego wpływu na funkcjonowanie zasadniczych dziedzin życia społecznego. Granice (zakres) tego typu obiektu określany być może ze względu na podział administracyjny lub np. strukturę organizacyjną przedsiębiorstw (organizacji).

Typ makro to obiekt obejmujący swoim zasięgiem cały obszar kraju, duże organizacje lub przedsiębiorstwa o zasięgu ogólnokrajowym. Cechą charakterystyczną tego typu obiektów jest występowanie bardzo dużej (trudno policzalnej) ilości węzłów telekomunikacyjnych (centrali), rozbudowanej zarówno w sensie ilościowym jak i jakościowym infrastruktury telekomunikacyjnej, dużego zbioru lokalnych i regionalnych sieci informatycznych i rozproszenie przestrzenne na

bardzo dużym obszarze wrażliwych na zagrożenia informacyjne elementów składowych. Występuje duża różnorodność środków przekazywania informacji, skomplikowana struktura hierarchicznych i równorzędnych struktur informacyjnych zdywersyfikowana struktura zarządzania mediami transmisji i czasem wielokrotne ich zrównoleglenie. Zidentyfikowanie rzeczywiście najbardziej wrażliwych na zagrożenia elementów jest wysoce utrudnione (a nawet wręcz niemożliwe w pełni i ustalane głównie *post factum*) ze względu na zakres przedmiotowy i niejednorodność organizacyjną. Podobnie trudna sytuacja jest z zabezpieczeniem przed zagrożeniami. Poziom tych zabezpieczeń bywa silnie zróżnicowany w zależności od rozmieszczenia przestrzennego, znaczenia w systemach społecznych, charakteru podmiotu będącego użytkownikiem systemu i przede wszystkim jego możliwości finansowych. W przypadku tego typu obiektu efektem negatywnego zdarzenia (ataku informacyjnego) w odniesieniu do pojedynczych podsystemów informacyjnych nawet jeżeli skutkuje całkowitym wyłączeniem z działania, nie ma w większości przypadków znaczącego wpływu na funkcjonowanie pozostałych podsystemów. Występujące utrudnienia w pracy sieci informacyjnych (poza przypadkami ekstremalnymi – zmasowanych ataków) nie powinny mieć znaczącego wpływu na funkcjonowanie zasadniczych dziedzin życia społecznego. Przestrzenne granice tego typu obiektu określane są obszarem terytorium państwa, natomiast zakres przedmiotowy w zakresie bezpieczeństwa informacyjnego może być utożsamiany z infrastrukturą krytyczną państwa.

Aby udzielić odpowiedzi na zasadnicze pytanie: dlaczego mamy chronić informację?, określić należy zasadnicze aspekty zagrożeń, jakie mogą wystąpić w stosunku do określonego obiektu (rodzaju obiektów), gdzie występują informacje, których całkowite lub częściowe pozyskanie lub zmienienie ich treści, może mieć destrukcyjny wpływ na funkcjonowanie nie tylko tych obiektów ale również obiektów nadrzędnych.

Ponieważ informacja jest przetwarzana głownie w systemach informatycznych, ich bezpieczeństwo nabiera coraz większego znaczenia. Skala zjawiska związanego z incydentami bezpieczeństwa w systemach informatycznych rośnie z dnia na dzień. Podyktowane jest to ciągłym rozwojem tych systemów, wprowadzaniem nowych wersji oprogramowania, dołączaniem nowego sprzętu, awariami technicznymi (uszkodzenia podzespołów, brak zasilania, brak łączności), zdarzeniami losowymi (pożar, powódź, wybuch), błędami ze strony użytkowników

i administratorów, jak również celową działalnością osób nieuprawnionych do dostępu do informacji w systemach (nieautoryzowany dostęp, modyfikacja czy niszczenie danych). Zagrożenia mogą być zamierzone lub przypadkowe, ludzkie bądź środowiskowe. Do najczęstszych należy przenikanie informacji, włamanie (fizyczne i elektroniczne), akt terrorystyczny, podsłuch i podgląd, kradzież, modyfikacja, ale także pomyłki i usterki. Do zagrożeń środowiskowych należą powodzie, huragany, trzęsienia ziemi, katastrofy budowlane, pożary.
Najbardziej ogólnym podziałem zagrożeń informacyjnych jest podział na:

- zagrożenia wewnętrzne – powstające wewnątrz organizacji (obiektu);
- zagrożenia zewnętrzne – powstające poza organizacją (obiektem).

Ten ogólny podział jest przydatny przy konstruowaniu obiektowej „polityki" bezpieczeństwa, gdzie rozróżnienie miejsca powstawania zagrożenia bezpośrednio wpływa na wybierane sposoby zabezpieczeń. Bardzo istotnym zagadnieniem (w kontekście zastosowania systemów eksperckich) jest również możliwość określenia położenia źródła zagrożenia, a w zasadzie zakłócenia (ataku), na podstawie jego skutków, co w oczywisty sposób wpływa na możliwość i skuteczność działań zabezpieczających zarówno w zakresie usuwania skutków jak i szeroko rozumianej prewencji. W nawiązaniu do dokonanego uprzednio podziału obiektów przypuszczać należy, że z punku widzenia obiektu typu mikro udział zagrożeń zewnętrznych jest znacznie większy niż w przypadku obiektu typu makro.

Zagrożenia klasyfikuje się również według przyczyn ich powstawania.

Rys. 3.4. Podział przyczyn (źródeł) zagrożeń bezpieczeństwa[14]

Warto zauważyć, że powyższy podział nie jest podziałem uniwersalnym dla wszystkich obiektów. Klasyfikacji takiej należałoby dokonywać nie tylko z uwzględnieniem rodzaju obiektu, którego zagrożenia dotyczą, ale nawet z dokładnością do konkretnego obiektu. Zdarzenie, które dla określonej klasy obiektów jest spowodowane świadomym, zdeterminowanym działaniem dla obiektów innej klasy może być zagrożeniem typowo losowym o większym lub mniejszym prawdopodobieństwie. Faktem jest, iż każdy z elementów systemów informacyjnych, również tych zaliczanych do grupy infrastruktury krytycznej państwa narażony jest na ryzyko zniszczenia, awarii zarówno z powodu przyczyn spodziewanych (awarie sprzętu elektronicznego), jak też występowania zjawisk naturalnych, jak: powodzie, trzęsienia ziemi, nieprawidłowe działanie ludzi czy też np. wzajemne oddziaływania na siebie sprzętu.

Zdarzeniami niespodziewanymi są oczywiście te, których nie można przewidzieć np. błędy ludzkie, nieprawidłowa praca systemu, kradzież, ataki fizyczne i inne.

Zdarzenia zamierzone są wynikiem celowych działań ludzi i mają na celu spowodowanie szkód lub strat. Sposoby ataku na obiekty infrastruktury informacyjnej można podzielić na pięć kategorii:

[14] Na podstawie: Sienkiewicz P., Górny P., Analiza systemowa sytuacji kryzysowych, Zeszyty Naukowe AON nr 4, Warszawa 2001.

- ataki fizyczne;
- ataki elektromagnetyczne na strukturę telekomunikacyjną;
- ataki skierowane przeciwko systemom informacyjnym;
- zagrożenia społeczne;
- uzależnienie od innych sektorów społeczeństwa.

W sytuacji nieskomplikowanej, zagrożenia powyższe występują pojedynczo. Oczywiście w czasie kryzysu lub wojny sytuacja będzie znacznie bardziej skomplikowana.

Nową jakością samą w sobie jest coraz popularniejsze zjawisko cyberterroryzmu. Cyberterroryzm jest najczęściej efektem eskalacji działań noszących znamiona przestępstw komputerowych. Ze względu na poziom niezbędnej wiedzy i potencjalne negatywne skutki można wyróżnić następujące zagrożenia związane z nadużyciami komputerowymi:

- piractwo komputerowe, w różnych dziedzinach i postaci (muzyka, film), usuwanie zabezpieczeń w programach itp.;
- proste próby ataku z wykorzystaniem ogólnie dostępnych w sieci Internetu narzędzi do przełamywania zabezpieczeń (tzw. narzędzi hakerskich);
- wykorzystywanie technik i narzędzi hakerskich przeciwko stronom internetowym przeciwników, np. politycznych;
- cyberterroryzm jako motywowane operacje hakerskie prowadzone w celu spowodowania wielkich strat ekonomicznych, np. penetracja systemu kontroli ruchu powietrznego, systemów sterowania, wytwarzania energii, zasilania w wodę, kontroli systemów giełdowych itp.

Granice oddziaływania są bardzo płynne z uwagi na wykorzystywanie tych samych środków, tj.: poczty elektronicznej i wirusów lub „innego złośliwego oprogramowania", jako elementu pojedynczego lub zmasowanego ataku.

Zagrożenie fizyczne, zazwyczaj zamierzone, będzie miało na celu zniszczenie elementów infrastruktury informacyjnej, generalne zredukowanie jej możliwości technicznych lub, w przypadku infrastruktury telekomunikacyjnej, odcięcie infrastruktury sieciowej (np. zniszczenie anten radiowych czy odcięcie obsługiwanych abonentów). Niewątpliwie w terenie nie jest trudno dokonać odcięcia kabli i radiolinii łączących instalacje, a obsługujących ruch o dużej koncentracji.

Środki ataku elektromagnetycznego zakłócają propagację sygnałów radiowych, prace systemów elektronicznych i komponentów elektronicznych. Podczas stosowania zakłócenia elektromagnetycznego silne sygnały radiowe są kierowane do lub z nadajników radiowych, nakładają się na nadawcze częstotliwości radiowe i mogą zniszczyć przesyłaną informację. Zazwyczaj zakłócanie nie powoduje fizycznego zniszczenia zespołów elektronicznych. Technika ta najczęściej stosowana jest przez wojsko. Stosuje się ja również w czasie pokoju, w celu zakłócenia publicznie dostępnych programów radiowych lub telewizyjnych.

Rozwój technologiczny spowodował, iż firmy wyposażone są w systemy komputerowe, służące do zarządzania siecią oraz do realizacji właściwych funkcji usługowych. Celem zagrożenia mogą być oba typy systemów. Zagrożenia te obejmować mogą atak bezpośredni na system skierowany na konkretne komponenty sprzętowe lub systemy komunikacyjne oraz na oprogramowanie, np. pakiety programów lub systemy informacyjne.

Bezpośredni atak na systemy komunikacyjne może doprowadzić do wyłączenia węzłów bądź przesyłanej informacji. Może także spowodować usunięcie danych lub programów, a jeżeli będzie powiązany z atakiem fizycznym, to efekty takich działań będą niewątpliwie rozległe i długotrwałe. Potencjalne negatywne skutki takich działań będą znacznie większe w środowisku wykorzystującym sprzężone systemy zarządzania siecią z dostępem do użytkownika. Takie rozwiązania stanowią największe potencjalne możliwości przeprowadzenia ataku systemowego na oprogramowanie i komponenty zarządzania siecią.

Oczywiście bezpośredni atak na system może być przeprowadzony zarówno z zewnątrz, jak i od wewnątrz. Przy ataku z zewnątrz, intruz może posłużyć się Internetem do złamania zabezpieczeń systemu telekomunikacyjnego i podjąć próbę uzyskania dostępu do systemów wewnętrznych. Trzeba dodać, że informacje i narzędzia służące do analizowania słabych punktów systemu i zabezpieczeń można łatwo znaleźć w Internecie.

Bezpośredni atak na system, przeprowadzony od wewnątrz, może być najczęściej umyślnym dziełem pracownika firmy pragnącego wyrządzić szkody własnemu pracodawcy. Ochrona przed takimi pracownikami zazwyczaj w ogóle nie istnieje, z uwagi na występujące zaufanie do pracownika i dlatego wykrycie takiej nielojalnej osoby i zapobieżenie jej działaniom jest trudne i skomplikowane.

Do znanych powszechnie metod ataku na oprogramowanie należą np. wirusy, odmowy usługi lub rozproszone odmowy usługi. We wszystkich metodach mogą być wykorzystane zasoby oprogramowania takie, jak: listy adresowe lub nagłe zniknięcie pliku lub katalogu, uszkodzenie plików, nieprawidłowe działanie plików, utworzenie nowych programów, zablokowanie możliwości zapisania plików w miejscu wybranym przez użytkownika, brak dostępu do systemu lub danych. Naruszona zostaje integralność danych i systemu.

Metody ataku charakteryzują się zazwyczaj różnymi poziomami złożoności. Mogą przy tym uderzać także w masowo produkowane markowe oraz specjalistyczne programy. Ataki na oprogramowanie i bezpośrednie ataki na system są wyjątkowo potężnym i wyrafinowanym środkiem ataku na newralgiczne informacje i systemy komunikacyjne.

Sieci telekomunikacyjne w normalnej sytuacji pracują w ruchu rozłożonym na wszystkie godziny doby. Mają przepustowość mniejszą od tej, jaka byłaby potrzebna, gdyby wszyscy chcieli w tym samym momencie z nich korzystać. Nowoczesne sieci zazwyczaj są odporne na zakłócenia. Dzięki temu rzadko dochodzi w nich do przeciążenia. Zwiększenie jednak w niej ruchów w sytuacjach awaryjnych lub kryzysowych może spowodować ich przeciążenie. Może ono również wystąpić w wyniku ataku na centralne systemy operacyjne, celowo generujące ruch przekraczający możliwości obsługowe sieci. Wynik takiego zablokowania sieci jest równy skutkom ataku fizycznego, blokującego połączenia, które powinny być realizowane. Usterki techniczne lub uszkodzenia łączy transmisyjnych w sieci ograniczają jej przepustowość. Jeżeli awarie lub uszkodzenia przekraczają pewien poziom, to ruch w sieci zostanie zablokowany i zatrzymany.

W omawianych zagrożeniach wskazano oprócz zagrożeń technicznych zagrożenia, jakie niesie dla informacji i systemów telekomunikacyjnych aspekt sfrustrowanego pracownika. Przekłada się on jednak znacznie szerzej i obejmuje celowe ataki społeczne, znane pod nazwą inżynierii społecznej. Przy braku odpowiednich zabezpieczeń błąd ludzki może spowodować usunięcie plików lub programów. Wraz ze zmianami społecznymi i ekonomicznymi zmienia się radykalnie sposób prowadzenia działalności gospodarczej. W skrajnych sytuacjach, gdy pracodawca nie może porozumieć się z pracobiorcą w sprawie zmian, może dojść do zachwiana systemu naprawy i funkcjonowania instalacji awaryjnych obsługujących priorytetowych klientów. Do inżynierii społecznej zalicza się więc działania

pracownicze, mające na celu pozyskanie tajnych informacji, które powinny być zlikwidowane w sposób poufny, przeglądanie korespondencji lub biur i zdobywanie informacji dających dostęp do systemów (np. haseł). Ten ludzki aspekt społecznego zagrożenia musi być koniecznie brany pod uwagę przy wdrażaniu środków zmniejszających wrażliwość systemu na atak. Należy mieć również na uwadze, że sieci teleinformatyczne stały się przedmiotem zabaw dla kilkunastoletnich chłopców. Liczą oni na anonimowość, bezkarność oraz lekceważenie tego zjawiska przez niektórych operatorów, a także na bezpieczeństwo ścian własnego domu - skąd najczęściej wędrują do cudzych systemów, nie zawsze z pełną świadomością wyrządzanych szkód.

Wspomniano już, iż obiekty infrastruktury krytycznej są ściśle z sobą powiązane i od siebie uzależnione. Nie tylko sprawność działania gospodarki zależy od telekomunikacji. Produkcja i informacja uzależnione są od zasilania energią elektryczną. Ale również energetyka, w równym stopniu, zależy od telekomunikacji, Załamanie się więc jednego z tych systemów może mieć dla państwa poważne konsekwencje, o ile nie zostaną zastosowane niezbędne środki awaryjne i zapobiegawcze. Handel i przemysł z kolei istotne są dla telekomunikacji choćby z uwagi na produkcję części zamiennych. Brak wystarczającego zaopatrzenia w części zamienne w sytuacji kryzysowej poważnie utrudni przywrócenie sprawności sieci. W przypadku jednak zastosowania zaawansowanych systemów logistycznych, odpowiedzialnych za realizację zamówień i magazynowanie części zamiennych, straty w wyniku występujących uzależnień zdestabilizować mogą całą gospodarkę.

Istotne zagrożenia wynikają również dla systemu energetycznego. Uszkodzenie fizyczne lub zablokowanie czy zniszczenie systemów nadzoru głównych linii przesyłowych, stacji czy podstacji zasilania może doprowadzić nie tylko do zniszczenia istotnych elementów systemu zasilania energetycznego. Szczególnie trudne do ochrony są obiekty liniowe typu rurociągi, gazociągi oraz obiekty przemysłu gazowego i naftowego. Uszkodzenie czy zniszczenie ich dodatkowo spowodować może klęskę ekologiczną na niespotykaną skalę. Braki dostaw materiałów pędnych spowodują przy tym zaburzenie funkcjonowania wielu gałęzi gospodarki. Rezerwy i systemy awaryjnego zasilania są zazwyczaj dostosowane do wystąpienia chwilowych braków, a nie na poważne zagrożenia o dużym obszarze i długim czasie działania.

Ataki na systemy bankowe i finansowe spowodują zakłócenia w funkcjonowaniu instytucji finansowych, zdestabilizują zarówno obieg pieniądza, jak i systemy płatności i rozliczeń. Doprowadzić również mogą nie tylko do opóźnień w transferach środków pieniężnych, ale i do braku dostępności środków płatniczych dla ludności (sieć bankomatów).

Nie bez znaczenia pozostają ataki dywersyjne czy sabotażowe w odniesieniu do środków transportu i infrastruktury transportowej. W ostatnich latach byliśmy świadkami wykorzystywania statków powietrznych jako broni terrorystycznej. Uniemożliwiona lub utrudniona może zostać praca portów lotniczych. Zniszczeniu mogą ulec statki powietrzne i jednostki pływające (np. promy). Zagrożone są obiekty, transporty kolejowe (jak i sam transport), szczególnie w dużych aglomeracjach miejskich (metro, dworce).

Jedną z metod oddziaływania na infrastrukturę krytyczną transportu lądowego może być zniszczenie lub wyłączenie z użycia istotnych węzłów komunikacyjnych, takich, jak: mosty wiszące na dużych przeszkodach wodnych, węzły drogowe, tunele, węzły kolejowe. Zagrożenie – jak wskazują ostatnie doniesienia – spowodowane może być udziałem samych środków transportowych przewożących materiały niebezpieczne.

Jednym z ostatnich zagrożeń, ale wyjątkowo ważnych dla funkcjonowania społeczeństw, jest zakłócenie dostarczania wody pitnej dla ludności, zwierząt hodowlanych oraz przemysłu spożywczego i farmaceutycznego. Skażenie lub zanieczyszczenie ujęć wody pitnej (chemiczne, biologiczne lub radiologiczne), zwłaszcza w dużych aglomeracjach miejskich, może mieć skutek katastrofalny.

Szczegółowa specyfikacja zagrożeń dla bezpieczeństwa systemów informacyjnych niewątpliwie jest problemem otwartym i wręcz niemożliwa do realizacji w czasie zbliżonym do rzeczywistego nawet w skali lokalnej, stąd też konieczność ciągłej aktualizacji baz danych o zagrożeniach i możliwych scenariuszach ich ewolucji. O skali problemu stojącego w tym zakresie przed ekspertami może świadczyć może ogólny wykaz dotyczący tylko rodzajów przestępstw związanych z fałszerstwem, oszustwem, sabotażem czy hackingiem komputerowym, jaki wykonał Komitet Ekspertów Rady Europy (wg Raportu Ekspertów Rady Europy - zał. do Rekomendacji Nr R(89)9 K.M.R.E.):

Lista minimalna:

Oszustwo komputerowe:

- Manipulacja danymi (wprowadzanie nieprawdziwych danych w celu uzyskania nienależnych korzyści majątkowych);
- Manipulacja programem (przygotowanie lub modyfikacja programu, by program wykonywał określone czynności niezależne od woli operatora);
- Manipulacja wynikiem (manipulacja urządzeniami wejścia-wyjścia np. bankomatami).

Fałszerstwo komputerowe:

- Komputerowe fałszerstwo dokumentu klasycznego (komputer, oprogramowanie i peryferia jako narzędzia);
- Fałszerstwa dokumentów elektronicznych (dokonywanie zmian w utworzonych i przyjętych dokumentach elektronicznych).

Nieuprawnione wejście(włamanie) do systemu (hacking):

- Włamanie się do systemu komputerowego poprzez przełamanie zabezpieczeń w postaci kodów i haseł broniących dostępu do tego systemu.

Niszczenie danych i programów komputerowych:

- Fizyczne (zniszczenie komputera lub nośnika z pomocą siły);
- Destrukcja programowa (zadziałanie konia trojańskiego, bomby logicznej, wirusa lub robaka komputerowego).

Sabotaż komputerowy;

- Uszkodzenie lub zniszczenie systemu komputerowego i informacji, gdzie celem działania sprawcy jest sparaliżowanie funkcjonowania systemu komputerowego.

Nieuprawnione przechwycenie informacji podsłuch komputerowy:

- Przejmowanie danych w czasie teletransmisji;
- Detekcja pola elektrycznego, magnetycznego, elektromagnetycznego i przewodów sprzętu komputerowego;
- Detekcja fal akustycznych generowanych przez drukarki.

Nielegalne kopiowanie, rozpowszechnianie lub publikowanie programów komputerowych chronionych prawem autorskim:

- Instalacja na twardym dysku dostarczanym przez sprzedawcę sprzętu nielegalnych kopii oprogramowania;
- Wykonywanie dodatkowych kopii;

- Nielegalne powielanie i sprzedaż oprogramowania chronionego prawem autorskim;
- Piractwo przez BBS;
- Naruszanie treści licencji (zasad korzystania);
- Wykorzystanie bez zgody właściciela kodu programu do tworzenia nowego programu.

Bezprawne kopiowanie topografii półprzewodników:

- Przez topografię półprzewodnika rozumie się rozwiązanie polegające na przestrzennym, wyrażonym w dowolny sposób rozplanowaniu elementów, z których co najmniej jeden jest elementem aktywnym, oraz wszystkich lub części połączeń układu scalonego.

Lista fakultatywna:

- Modyfikacja danych lub programów komputerowych (nieuprawniona ingerencja w treść danych lub cracking);
- Szpiegostwo komputerowe;
- Używanie komputera bez zezwolenia;
- Używanie prawnie chronionego programu komputerowego bez upoważnienia;

[...]

3.2. Strategie przeciwdziałania zagrożeniom informacyjnym

Poszczególne sektory społeczeństwa są wzajemnie mocno powiązane i uzależnione, w wyniku czego zakłócenia w funkcjonowaniu jednego sektora objawiać się będą konkretnymi konsekwencjami w innych sektorach. Z tego też względu państwo zobowiązane jest poprzez reprezentujące je instytucje zagwarantować bezpieczeństwo funkcjonowania, szczególnie infrastruktury krytycznej. Główne zadanie systemu ochrony infrastruktury krytycznej państwa realizowane jest przy użyciu środków mających uchronić przed awarią te elementy infrastruktury, które mają decydujące znaczenie dla jego prawidłowego funkcjonowania. W ubiegłych latach systemy ochrony infrastruktury krytycznej koncentrowały się głównie na likwidacji ryzyka i zagrożeń związanych z wojną. Obecnie systemy te w większym stopniu skupiają się na tych czynnikach ryzyka, które pojawiają się podczas kryzysu w okresie pokoju, a także na czynnikach zagrożeń przewidywanych w scenariuszach zagrożeń informacyjnych.

Kwestie bezpieczeństwa informacyjnego dotyczą nie tylko infrastruktury krytycznej państwa ale przede wszystkim większości obszarów działania każdej nowoczesnej instytucji: sposobu prowadzenia codziennych operacji, zarządzania finansami, polityki personalnej czy zapewnienia zgodności z wymaganiami prawnymi. Instytucje (firmy, organizacje) pełnią swoją misję (państwową, rynkową lub społeczną), czyli realizują swoje zadania statutowe, takie jak np. obsługa państwa, ludności, produkcja dóbr czy świadczenia usług między innymi poprzez przetwarzanie coraz większej ilości cennych informacji. W tym celu są wręcz zmuszone do stosowania środków teleinformatycznych, bez większej przesady użyć można stwierdzenia, że takie jest bezpieczeństwo misji instytucji (realizacji zadań statutowych, funkcjonowania w społeczeństwie), jakie jest bezpieczeństwo jej systemów teleinformatycznych.

Zapewnienie bezpieczeństwa złożonego systemu nie jest sprawą łatwą, wymaga wzorowej organizacji, dyscypliny, wiedzy i sprawnego zarządzania. Zarządzanie bezpieczeństwem informacji jest dziedziną młodą z pogranicza techniki, organizacji i prawa, zajmującą się definiowaniem, osiąganiem i utrzymaniem bezpieczeństwa rozumianego jako zapewnienie dla systemów poufności, integralności, dostępności, autentyczności oraz niezawodności. Wykorzystuje szereg

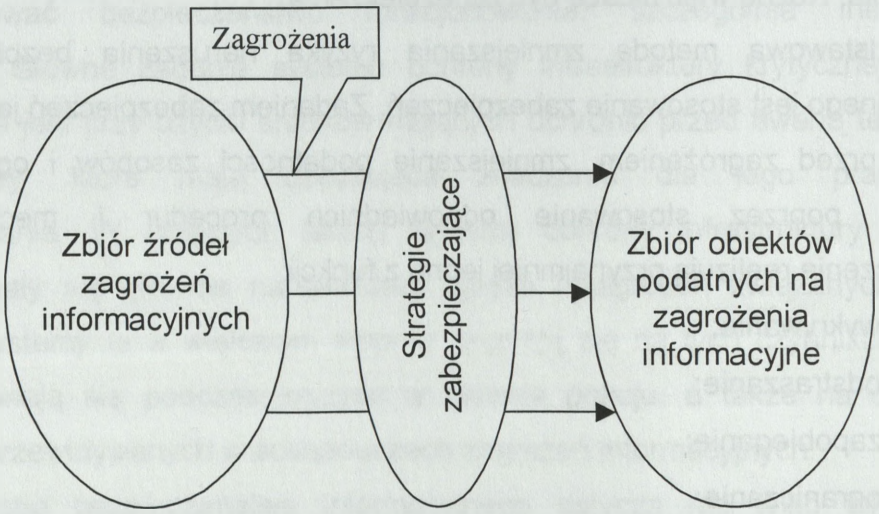
modeli, jak np. związków pomiędzy elementami bezpieczeństwa. Zakładając, że cechą wspólną instytucji jest realizacja zadań statutowych (misji) przy zaangażowaniu środków teleinformatycznych np. obsługa obywateli, ochrona odcinka granicy państwowej itp., to misje różnić się będą charakterem i rangą społeczną, zaś same instytucje mogą być uznawane za bardziej lub mniej uzależnione od środków teleinformatycznych. Różne są w nich zasoby, podatności i zagrożenia. Różne więc muszą być zabezpieczenia.

Podstawową metodą zmniejszania ryzyka naruszenia bezpieczeństwa informacyjnego jest stosowanie zabezpieczeń. Zadaniem zabezpieczeń jest ochrona zasobów przed zagrożeniem, zmniejszanie podatności zasobów i ograniczanie następstw poprzez stosowanie odpowiednich procedur i mechanizmów. Zabezpieczenia realizują przynajmniej jedną z funkcji:

- wykrywanie;
- odstraszanie;
- zapobieganie;
- ograniczanie;
- poprawianie;
- odtwarzanie;
- monitorowanie;
- uświadamianie.

Dobór zabezpieczeń powinien być efektem określenia celów i strategii przyjętych w polityce bezpieczeństwa. Racjonalna strategia zarządzania ryzykiem opiera się o wyznaczenie poziomu zabezpieczenia, który jest funkcją wartości kryteriów poufności, wiarygodności, dostępności, rozliczalności, autentyczności i niezawodności. Poziom ten powinien przyjmować największą wartość ze zbioru wartości kryteriów. Mimo zastosowania odpowiednich zabezpieczeń ryzyko nie jest nigdy redukowane do zera. Pozostaje tzw. ryzyko szczątkowe. Jeżeli jego poziom nie jest akceptowalny, to należy wprowadzić dodatkowe elementy zabezpieczenia (wiąże się to ze wzrostem kosztów). W przeciwnym wypadku należy dokładnie wyspecyfikować relacje między zasobami i zagrożeniami oraz rozważyć przyjęcie alternatywnych zabezpieczeń.

Środki używane w systemie ochrony informacyjnej infrastruktury mogą być skierowane na zmniejszenie możliwości wystąpienia awarii (nieprawidłowego działania) lub też mogą służyć ograniczeniu konsekwencji wynikających z awarii infrastruktur o newralgicznym znaczeniu. Innymi słowy możemy określić strategie przeciwdziałania zagrożeniom (prewencyjne), zapobiegające skutkom incydentów (wynikającym ze scenariuszy) oraz strategie likwidacji skutków.



Rys. 3.5. Działanie prewencyjne strategii zabezpieczających

Najprostszą, a zarazem najskuteczniejszą, podstawową formą ochrony, jest wypracowanie wśród zainteresowanych właściwym funkcjonowaniem infrastruktury krytycznej stron, świadomości ryzyka i zagrożeń, co wpłynie na jakość wykonywanych czynności. Znajomość słabych punktów w infrastrukturze określonej branży, połączona z odpowiednią wiedzą i systematyczną edukacją, pozwoli lepiej daną infrastrukturę chronić.

Bezpieczeństwo informacji ma charakter złożony i dotyczy:

- samej informacji, a zwłaszcza jej specyficznej postaci, często nieuchwytnej dla wielu osób (można zostać okradzionym nie będąc tego świadomym);
- systemów, w których jest ona wytwarzana, przetwarzana, przechowywana i przekazywana;
- środowiska, w którym te systemy działają, każdy szczegół dotyczący pomieszczenia, okablowania czy zasilania może okazać się decydujący w skutkach;

- personelu, który korzysta z tych systemów i który często bywa niedouczony, nieobliczalny w swoich działaniach i trudny do skontrolowania;
- całego otoczenia prawnego, które w sposób ciągły jest kształtowane i zmieniane, podążając za rozwojem technologii.

Kluczowym modelem przyjmowanym do rozważań w zakresie bezpieczeństwa jest zwykle model trójpoziomowy wyrażający hierarchie celów, strategii oraz polityki w instytucji. Dla każdego szczebla organizacji lub obszaru działania instytucji określa się trzy podstawowe pojęcia:

- cel; identyfikuje, co ma zostać osiągnięte;
- strategia; określa, jak osiągnąć wytyczony cel;
- polityka; podaje, co musi być konkretnie wykonane.

Taki model instytucji ma charakter interdyscyplinarny, hierarchiczny i wielopoziomowy. Cel globalny instytucji związany jest z misją, do której pełnienia instytucję powołano. Strategia globalna określa, w jaki sposób instytucja ma osiągać cel. Polityka globalna definiuje szczegółowy harmonogram przedsięwzięć techniczno-organizacyjnych. Wszystkie razem są uwarunkowane szeregiem czynników.

Przedstawiona poniżej schematyczna struktura całokształtu działalności na rzecz zapewnienia bezpieczeństwa postępowania w oczywisty sposób koresponduje z przedstawioną uprzednio trójpoziomową hierarchią obiektów zagrożeń informacyjnych.

Na pierwszym poziomie mówimy o polityce bezpieczeństwa w instytucji (państwie), którą określa się jako podstawowe zasady bezpieczeństwa i wytyczne dla całej instytucji. Polityka ta usadowiona jest w realiach prawnych instytucji, powinna odzwierciedlać politykę w szerszym zakresie, obejmując zawarte umowy, obowiązujące akty prawne, w tym dotyczące praw człowieka, np. do ochrony danych osobowych. Na tym poziomie chronione są takie aktywa, jak: ciągłość misji, ciągłość procesów biznesowych, zdolność produkcji i świadczenia usług, reputacja czy działanie zgodnie z prawem. Określa się tu dość ogólne potrzeby i wymagania dotyczące związku misji z bezpieczeństwem w instytucji (co chronimy, jak i dlaczego).

Bezpieczeństwo w instytucji jest takie, jakie jest bezpieczeństwo jej systemów teleinformatycznych, bezpieczeństwo fizyczne oraz osobowe, czyli takie, jakie jest bezpieczeństwo na drugim poziomie.

Operuje się tutaj pojęciem polityki bezpieczeństwa instytucji w zakresie systemów informatycznych, którą normy definiują jako zbiór prac, reguł i wskazówek praktycznych, które określają jak aktywa informatyczne, w tym informacje wrażliwe, są zarządzane, chronione i dystrybuowane w instytucji i jej systemach informatycznych. Chronione są tu takie aktywa, jak: zasoby informacji, oprogramowanie, sprzęt, kadry, dokumenty, w tym dotyczące eksploatacji oraz samych zabezpieczeń. Opracowywane są dość szczegółowe wytyczne związane ze specyfiką instytucji i jej systemów.

Podobnie jak uprzednio bezpieczeństwo w instytucji systemów teleinformatycznych jest więc takie, jakie jest bezpieczeństwo każdego z eksploatowanych w niej systemów, a ściślej najsłabszego spośród nich.

Na trzecim poziomie operujemy więc pojęciem polityki bezpieczeństwa systemu teleinformatycznego (konkretnego), którą określa się jako zbiór praw, reguł i praktycznych doświadczeń regulujących sposób zarządzania, ochrony i dystrybucji informacji wrażliwej wewnątrz określonego systemu. W tym przypadku przedmiotem zainteresowania jest ochrona informacji wrażliwych wewnątrz konkretnego systemu. Opracowuje się więc konkretne plany zabezpieczenia danego systemu. Cele, strategie i polityki bezpieczeństwa systemu informacyjnego wyrażają to, czego się od każdego z nich oczekuje. Odnoszą się więc do spełnienia atrybutów bezpieczeństwa, tj. poufności, integralności, dostępności, rozliczalności, autentyczności oraz niezawodności[15].

Potrzebne jest do tego zarówno wypracowanie stosownych procedur powiadamiania, jak i środków ochrony przed możliwymi incydentami. Oczywiście bezpieczeństwo pracy podnosi ewentualny, sprawny system wewnętrznych procedur powiadamiania o zagrożeniach i incydentach oraz ścigania sprawców. Kluczowym obecnie problemem staje się ocena występujących zagrożeń i metody ich analizy. Rozwiązanie tych problemów wpłynie na uodpornienie systemów informacyjnych instytucji na ataki i doprowadzenie do tego, by różne mechanizmy zabezpieczające powstawały w różnych organizacjach i na różnych poziomach technicznych.

---

[15] Biała A., Bezpieczeństwo sieci komputerowych, Bielsko-Biała, WSI i Zarządzania 2001.

Prawidłowo zaprojektowane sieci teleinformatyczne powinny wykorzystywać urządzenia aktywne, umożliwiające, dzięki zmianie ich konfiguracji, logiczne odseparowanie poszczególnych segmentów sieci. Stosowanie takiej architektury systemów i sieci, w przypadku ataku komputerowego, umożliwi szybkie odizolowanie (odłączenie od zewnętrznych sieci) najbardziej podatnych i wrażliwych segmentów.

Inną ze stosowanych prawidłowych praktyk ochrony infrastruktury krytycznej jest scentralizowane monitorowanie (wykrywanie włamań do najbardziej krytycznych systemów i sieci np. zautomatyzowany audyt, częste analizy zachowań użytkowników, szersze kryteria rejestracji nietypowych zdarzeń). Każdy z takich systemów powinien wykorzystywać narzędzia wykrywania włamań i przekazywania danych do centrum monitoringu bezpieczeństwa teleinformatycznego.

Ścisłej kontroli podlegać winna weryfikacja personelu mającego dostęp do krytycznych systemów i sieci teleinformatycznych, szczególnie osób, które mogłyby dokonywać zmian w architekturze czy konfiguracji systemów operacyjnych i urządzeń. Stosowane są procedury postępowań sprawdzających zgodnie z przepisami o ochronie informacji niejawnych.

Praktycznie dostęp do kodów źródłowych krytycznych systemów lub sieci przydzielany jest jedynie osobom, które poddane zostały postępowaniu sprawdzającemu i uzyskały odpowiednie poświadczenia bezpieczeństwa osobowego.

W stosunku do ochrony informacji wrażliwych, w kontekście elektronicznego przetwarzania, przechowywania i przesyłania informacji wrażliwych, bezpieczeństwo fizyczne związane jest z ochroną obiektów osób, materiałów i dokumentów przed dostępem fizycznym, podglądem, podsłuchem lub inną formą obserwacji, wraz z procedurami dopuszczającymi i sprawdzającymi.

System powyższy powszechnie jest kojarzony ze strefową organizacją ochrony, realizowaną przy pomocy wyspecjalizowanych służb ochrony, istniejącym systemem przepustek, szaf pancernych itp. Każda ze stref ma ustalone zadania do spełnienia, zasady dostępu oraz wypracowane metody oraz środki kontroli. W systemie ochrony fizycznej infrastruktury krytycznej funkcjonują także zabezpieczenia przeciwwłamaniowe oraz zasady postępowania, zalecające przechowywać szczególnie ważne z punktu widzenia bezpieczeństwa zasoby (sprzęt, dokumentacja, zapasowe kopie).

Wypracowany w powyższy sposób system ochrony struktury informacyjnej zorganizowany jest w sposób planowy, a każdorazowe jego naruszenie związane jest z koniecznością odtworzenia albo jego fizycznej zdolności funkcjonowania. Obowiązkiem samego użytkownika jest dążenie do jak najszybszego usunięcia występującego zakłócenia, udzielenie pomocy zaatakowanym innym elementom podlegającej jego jurysdykcji infrastruktury informacyjnej, zminimalizowanie wpływu z powodu występujących zakłóceń w usługach lub zasobach krytycznych systemu na skutek wystąpienia incydentu oraz złożenie instytucjom nadrzędnym stosownego meldunku. W zależności od wypracowanego dla określonego systemu (jakim jest system reagowania), szczegółowego postępowania z jednej strony pozwoli na sprawniejsze usunięcie występującego zakłócenia czy zagrożenia, z drugiej natomiast spowoduje obniżenie ryzyka związanego z możliwością ponownego wystąpienia incydentu. Odtwarzanie zdolności funkcjonowania infrastruktury krytycznej, np. sieci abonenckiej, z której korzysta administracja państwowa nastąpić może poprzez przejście na inne techniczne środki łączności lub częstotliwości albo całkowitą zmianę systemu przekazywania informacji.

Odbudowa infrastruktury telekomunikacyjnej częściowo możliwa będzie w ramach posiadanych rezerw, obejmujących m.in.:

- ruchome obiekty telekomunikacyjne;
- stacjonarne urządzenia radiowe;
- stacjonarne urządzenia teletransmisyjne;
- zespoły prądotwórcze.

Odtwarzanie naruszonych np. systemów łączności w obiektach o lokalnym zasięgu następować może także poprzez wykorzystanie mobilnych urządzeń telekomunikacyjnych oraz zespołów prądotwórczych z innych obiektów, a także urządzeń sprowadzanych w ramach sporządzonych wcześniej umów. Dyspozytorzy sieci telekomunikacyjnych nadzorować powinni kolejność usuwania awarii, organizować łączność zastępczą, prowadzić współdziałanie z innymi służbami dyspozytorskimi. Utrzymanie infrastruktury krytycznej w stałej sprawności wymaga ciągłego uaktualniania posiadanych planów operacyjnych pod kątem zagrożeń, a także praktycznego treningu obsługi w realnym działaniu.

Procedury realizowane w celu zapewnienia bezpieczeństwa dzielone są zwykle na procedury eksploatacyjne i procedury awaryjne. Celem procedur jest

zapewnienie sprawnego wykonywania rutynowych działań, których celem jest zapewnienia poprawnej eksploatacji w każdej sytuacji, także w sytuacji błędu. Są one dedykowane do każdego systemu indywidualnie. Są również procedury dedykowane dla konkretnej organizacji. Procedury poprawnej eksploatacji są związane z każdym wyodrębnionym zasobem oraz ryzykiem (zagrożeniem). Ich celem jest formalne określenie działań, osób wykonujących działanie, osób decydujących o wprowadzeniu danej procedury, opisu samego działania. Procedura winna obejmować szczegółowy opis podejmowanych czynności. Idealne procedury winny przewidywać każde działanie, każdą potencjalną decyzję i podać wszelkie przesłanki, ułatwiające podjęcie decyzji. Powinna eliminować samodzielne podejmowanie decyzji.

Procedury bezpieczeństwa stanowią- sformalizowany opis zasad zapewnienia bezpieczeństwa fizycznego , bezpieczeństwa dostępu, polityki szkoleniowej w zakresie bezpieczeństwa i stosowanych specjalistycznych programowo-sprzętych rozwiązań bezpieczeństwa oraz procedury obejmujące sytuacje nadzwyczajne takie jak pożar, powódź i inne sytuacje powodujące konieczność natychmiastowej ewakuacji. W grupie tej winny być również określone procedury reakcji na zagrożenie bezpieczeństwa w innej nieprzewidzianej dotychczas sytuacji. Procedury muszą wskazywać osoby wykonujące, osoby odpowiedzialne za wykonanie i zasady odbioru pracy.

Podstawowym problemem związanym z możliwością efektywnego wykorzystania procedur jest jej realna przydatność w sytuacjach, które opisuje. Dlatego podstawowym zadaniem zapewniającym stałą przydatność procedur jest testowanie ich w celu wykrycia najdrobniejszych różnic pomiędzy sytuacją, którą mają opisywać a sytuacją rzeczywistą, w której są stosowane. To testowanie i stosowne modyfikowanie procedur jest zadaniem stałym.

Kolejnym krokiem po zdefiniowaniu elementów (procedur) systemu bezpieczeństwa jest określenie wskaźników i wartości kryteriów zabezpieczenia i wynikającego z nich wyznaczania poziomu zabezpieczenia. Dopiero na bazie analizy ryzyka, możemy dokonać wyboru odpowiedniej strategii bezpieczeństwa informacyjnego.

Przyjmuje się, że dla organizacji o wymaganym niskim lub średnim poziomie zabezpieczenia właściwą strategią byłoby oparcie się na ogólnej analizie ryzyka. Jej

konsekwencją jest zastosowanie podstawowego rodzaju zabezpieczeń. Analiza tego typu ma jednak charakter bardzo ograniczony, może też być prowadzona siłami własnymi organizacji (firmy).pozwala to bazować na adaptacjach międzynarodowych lub krajowych normach i zaleceniach. W ramach tej strategii wykorzystuje się rozwiązania przyjęte przez organizacje o zbliżonych celach działania, podobnej strukturze, wielkości i podobnym systemie informacyjnym

Strategia nieformalnej analizy ryzyka jest przeznaczona dla organizacji, których poziom zabezpieczenia określono jako średni i zalecana jako minimum możliwych rozwiązań dla organizacji o wysokim poziomie bezpieczeństwa. Analiza ryzyka w tym przypadku polega na pragmatycznym, eksperckim zbadaniu zarówno zagrożeń, jak i możliwych do zastosowania zabezpieczeń.

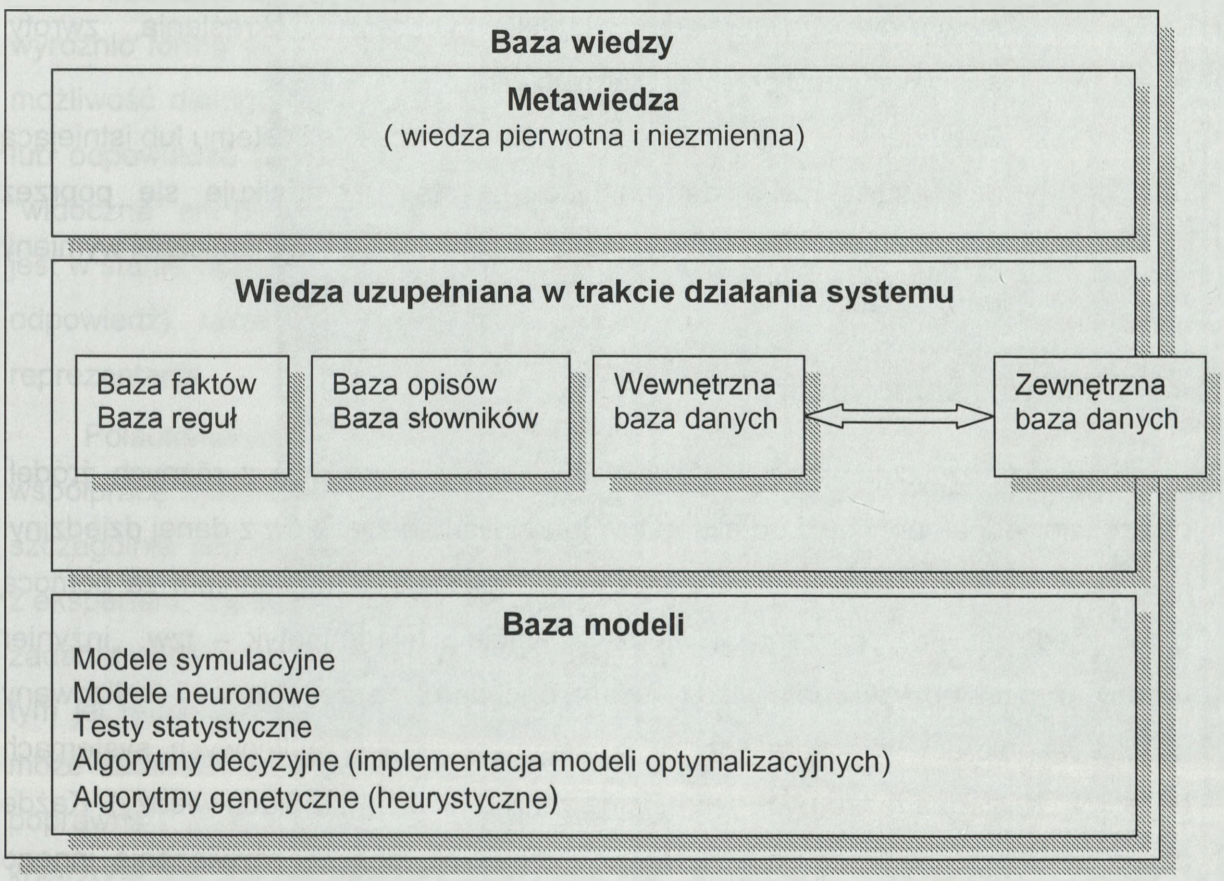
Strategia strukturalnej analizy ryzyka zalecana jest dla organizacji o wysokim poziomie zabezpieczenia i powinna być obowiązkowa dla firmy o zabezpieczeniu maksymalnym. Jest to metoda droga i czasochłonna, wymagająca zatrudnienia ekspertów oraz ciągłego uczestnictwa w pracach przedstawiciela kierownictwa badanej firmy (organizacji).stosuje się tutaj diagramy analityczne i zestawy analityczno kontrolne zwykle z automatyczną obróbką informacji.

Strategia kombinowanej analizy ryzyka jest najwłaściwszym rozwiązaniem dla większości organizacji o pożądanym poziomie zabezpieczenia wyższym niż niski. Polega głownie na zastosowaniu ogólnej analizy do określenia tych miejsc w systemach informacyjnych, które powinny zostać poddane dalszej , nieformalnej (eksperckiej) analizie ryzyka. To z kolei wyznaczy miejsca, systemy i procedury krytyczne dla misji, celów i zadań firmy (organizacji). Elementy krytyczne poddaje się szczegółowej analizie. Elementy ważne opracowuje się metodą nieformalną.

4. Koncepcja budowy struktury ESWD

4.1. Konstruowanie baz wiedzy dla potrzeb ESWD

Zasadniczą i integralną częścią systemu eksperckiego wspomagania decyzji przeciwdziałania zagrożeniom informacyjnym (obok maszyny wnioskującej) jest system bazy wiedzy. System ten przechowuje wszystkie informacje, które posiada system ekspercki i które w trakcie pracy może wykorzystywać. Z tego powodu właściwy wybór struktury bazy wiedzy gwarantuje możliwość poprawnej pracy systemu. Ogólną koncepcję struktury bazy wiedzy systemu eksperckiego można przedstawić w sposób następujący:



Rys. 4.1. Struktura bazy wiedzy systemu eksperckiego.

- Moduł metawiedzy – jest wiedzą zależności ogólnych, zawiera informacje na temat sposobów posługiwania się całą zgromadzoną wiedzą. Na metawiedzę składają się wszystkie reguły, procedury i parametry

umożliwiające kontrolę stanu wiedzy, komunikowanie się elementów bazy wiedzy ze sobą i z otoczeniem, w tym z maszyną wnioskującą.

- Baza modeli – pewna uniwersalna (niezależna od sposobu reprezentacji wiedzy i mechanizmu wnioskowania) część bazy wiedzy składająca się z formalnych, dokładnie zdefiniowanych modeli obliczeniowych symulacyjnych dotyczących zjawisk i procesów z rozpatrywanej dziedziny.
- Wiedza uzupełniana w trakcie działania systemu – czyli połączenie:
  - bazy faktów czyli stwierdzeń, które uważane są za prawdziwe (z możliwością generowania nowych faktów w trakcie procesu wnioskowania) wraz z bazą reguł, na której opiera się mechanizm maszyny wnioskującej.
  - bazy opisów (czyli metafory i wyjaśnień typu „what_is") wraz z bazą słowników, zawierającą wszelkie niezbędne określenia, zwroty, specjalistyczne pojęcia i sformułowania.
  - Bazy danych , która może być integralną częścią systemu lub istniejącą zewnętrzną bazą danych, którą system komunikuje się poprzez mechanizm ODBC lub wykorzystując mechanizm dynamicznej wymiany danych DDE.

4.1.1. Pozyskiwanie wiedzy

Wiedza znajdująca się w bazie wiedzy może pochodzić z różnych źródeł, najczęściej jednak pochodzi od ekspertów lub innych specjalistów z danej dziedziny. Pozyskiwaniem wiedzy eksperckiej oraz jej formalizacją, tj. zapisaniem za pomocą określonego języka reprezentacji wiedzy, zajmuje się informatyk – tzw. „inżynier wiedzy". Proces pozyskiwania wiedzy jest na ogół bardzo pracochłonny i realizowany w toku współpracy inżyniera wiedzy i eksperta. W tzw. tablicowych systemach eksperckich wiedza może być rozproszona w kilku plikach (źródłach wiedzy). Każde ze źródeł może wówczas przechowywać wiedzę służącą do rozwiązania innego podproblemu.

Szerokie spektrum metod pozyskiwania wiedzy wymaga pewnej klasyfikacji. Biorąc pod uwagę stopień zaangażowania oprogramowania w pozyskiwanie i kreowanie wiedzy, metody jej pozyskiwania można podzielić na:

- manualne;

- półautomatyczne;
- automatyczne.

W prostych wypadkach pozyskiwanie wiedzy sprowadza się do umieszczenia elementów wiedzy (reprezentowanej w postaci reguł, trójek, ram, sieci semantycznych czy odpowiedniej hybrydy) w bazie wiedzy. Taki sposób pozyskiwania wymaga współpracy między ekspertem dziedzinowym, będącym źródłem wiedzy, a inżynierem wiedzy. Oczywiście główna rola w manualnym pozyskiwaniu wiedzy przypada inżynierowi wiedzy. To on musi uzyskać od eksperta informacje dotyczące przedmiotowej dziedziny wiedzy i umieścić je w odpowiedniej formie w bazie wiedzy (mając cały czas na uwadze przyszłe funkcjonowanie systemu).

Posuwając się w kierunku automatycznych metod pozyskiwania wiedzy, można wyróżnić formę pośrednia - metody półautomatyczne. Ekspert ma w tym wypadku możliwość dialogu z systemem. Może przy tym rozwiązywać problemy (przykłady) lub odpowiadać na pytania systemu. Cała wewnętrzna struktura systemu nie jest "widoczna" ani dla eksperta, ani dla inżyniera wiedzy. Moduł pozyskiwania wiedzy jest w stanie wprowadzona przez eksperta wiedzę (np. w postaci związków pytanie - odpowiedź) umieścić w bazie zgodnie z przyjęta przez inżyniera wiedzy reprezentacja.

Półautomatyczne metody pozyskiwania wiedzy maja w swoim założeniu współpracę systemu z ekspertem (a nawet bezpośrednim użytkownikiem - co jest szczególnie cenne, gdyż umożliwia akwizycję wiedzy nie tylko w wyniku kontaktu z ekspertem, ale także w trakcie normalnej pracy użytkowej systemu). System ma za zadanie weryfikować i porządkować wiedzę pozyskiwana od użytkownika. Bada przy tym jej redundancję i ewentualna sprzeczność. W wypadkach wątpliwych system może zadawać dodatkowe pytania. W ten sposób pozyskana wiedza jest na ogół poprawna (choć nie musi być kompletna). Półautomatyczne pozyskiwanie wiedzy kojarzone jest zazwyczaj z tzw. maszynowym uczeniem. Polega ono na pozyskaniu wiedzy w trakcie dialogu systemu ekspertowego z ekspertem lub użytkownikiem. Ponieważ pozyskiwanie wiedzy odbywa się w tym przypadku na drodze wielokrotnego analizowania szczególnych wypadków (przykładów), metody półautomatycznego pozyskiwania wiedzy określone są czasami trenowaniem systemu.

Najbardziej zaawansowanymi metodami pozyskiwania wiedzy są metody automatyczne. Nie jest tu niezbędny udział eksperta czy inżyniera wiedzy. Ekstrahowanie nowej wiedzy odbywa się z wykorzystaniem wiedzy już wcześniej wprowadzonej. Nie ma więc potrzeby kontaktu z otoczeniem systemu. Procedury pozyskiwania wiedzy mogą, działać automatycznie, analizując wiedzę zgromadzona w bazie wiedzy. Działanie tych procedur ukierunkowane jest głównie na wykrywanie i usuwanie redundancji i sprzeczności w bazie wiedzy. Wiedza pozyskana w sposób automatyczny jest na bieżąco wykorzystywana w pracy systemu (do rozwiązywania problemów).

Doświadczenia uzyskane przy projektowaniu i budowie systemów ekspertowych wskazują, że najbardziej efektywną formą, pozyskiwania wiedzy jest analiza przykładów i kontrprzykładów. Forma ta jest stosowana w półautomatycznych metodach pozyskiwania wiedzy, ale zasadnicze znaczenie ma w odniesieniu do metod automatycznych. Przykłady są podstawą do indukcyjnego wyciągania wniosków ogólniejszych (tworzenia reguł). Zbiór przykładów jest wystarczającą podstawą, do tworzenia nowej wiedzy (bez żadnych dodatkowych źródeł zewnętrznych). Jednak ta nowa wiedza nie musi być w pełni poprawna, stąd zachodzi konieczność jej weryfikacji, w szczególności przez konfrontację ze sprawdzoną wiedzą zgromadzoną już wcześniej w systemie.

Jak wspomniano, pozyskiwanie wiedzy urasta do roli jednego z ważniejszych problemów w dalszym rozwoju systemów ekspertowych. Szczególnie metody automatycznego pozyskiwania wiedzy (wciąż jeszcze najtrudniejsze, najsłabiej rozwinięte) muszą być intensywnie doskonalone. Duże nadzieje w tym względzie są pokładane w sieciach neuronowych.

4.2. Baza obiektów

Na ogólną bazę obiektów składają się bazy obiektów z trzech poziomów:

- poziomu mini, na który składają się wszystkie sieci lokalne istotne dla rozpatrującego problem wraz z niezbędną infrastrukturą;
- poziomu mezo, który składa się z systemów ośrodków miejskich (regionalnych);
- poziomu makro, na który składają się wszystkie istotne systemy z poziomów mini i mezo.

W bazie obiektów skali mini umieszczone powinny być te zasoby (informacyjne i infrastrukturalne), które są lub mogą być użytkowane na danym obszarze (w danym obiekcie). Dla sektora mediów komunalnych będzie to np. miejscowa sieć wodociągowa i kanalizacyjna, sieci energetyczne, instalacje gazowe i inne. Stan tych obiektów może mieć wpływ na pracę obiektów z sektorów telekomunikacji i teleinformatyki (i odwrotnie).

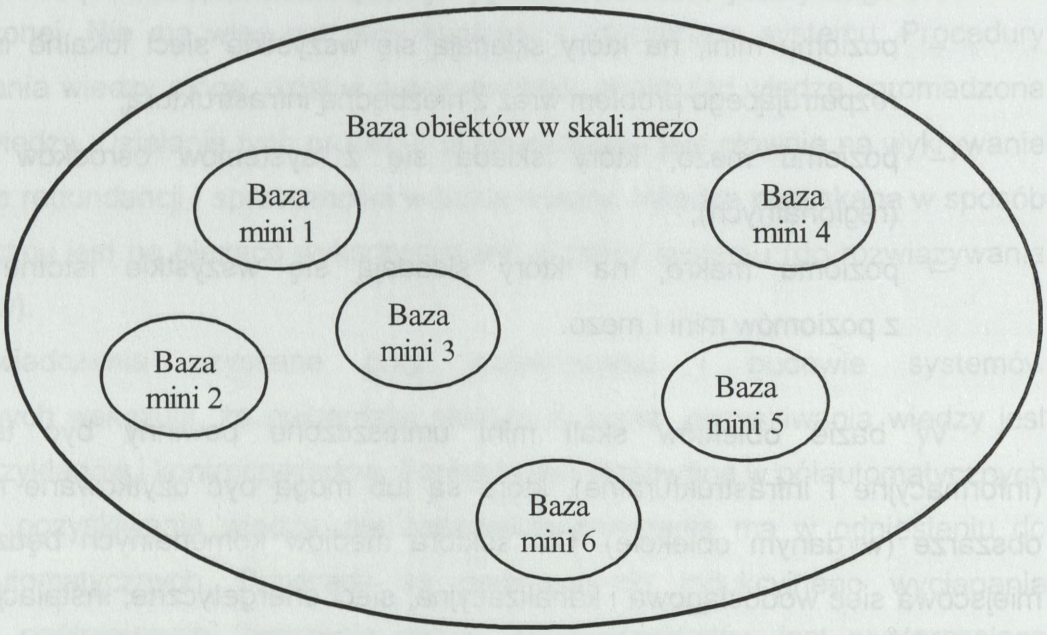
Dla sektora telekomunikacji istotnymi obiektami są:

- urządzenia końcowe klienta: czyli telefony, faksy, modemy;
- linie przesyłowe: napowietrzne i kablowe;
- lokalne centralki cyfrowe PAX;
- multipleksery ATM/SDH;
- w wyjątkowych przypadkach centrale telefoniczne.

Dla sektora teleinformatyki takimi obiektami są min.:

- urządzenia końcowe:
  - stacje robocze;
  - terminale sieci;
- linie transmisji danych;
- urządzania grupujące: huby, przełączniki i multipleksery;
- urządzenia dostępowe i zabezpieczające: routery, firewalle;
- serwery.

Na bazę obiektów w skali mezo składają się przede wszystkim bazy obiektów skali mini należące do jej obszaru zainteresowań (Rys. 4.2.).

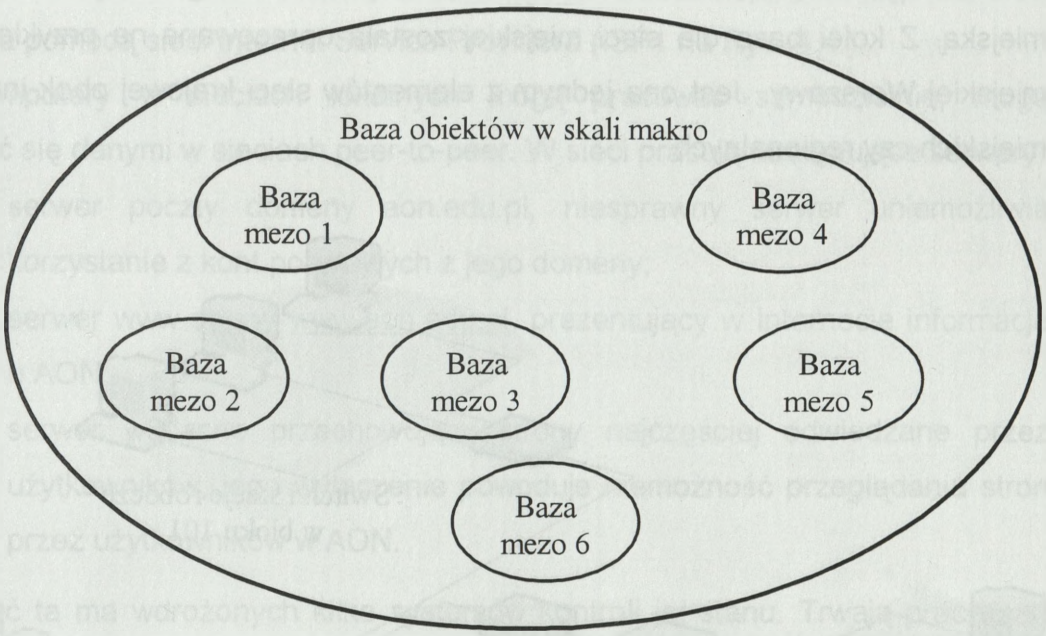


Rys. 4.2. Baza obiektów w skali mezo.

Baza w skali mezo może i powinna różnić się skalą szczegółowości od bazy w skali mini, przykładowo, obiekty takie jak pojedyncza stacja robocza bardzo istotne w skali mini mogą być bez większego znaczenia w skali mezo. Dodatkowymi obiektami, które mogą wystąpić w bazie w skali mezo, a nie występującymi w skali mini są sieci magistralne i duże linie przesyłowe.

W krajach, gdzie przeprowadzono deregulację systemów w poszczególnych sektorach może występować wiele równolegle prowadzonych sieci. Duży ośrodek miejski, np. Warszawa może posiadać wiele miejskich sieci z sektora teleinformatyki i telekomunikacji. Przykładowo w Warszawie istnieje kilka dużych sieci miejskich, czasem niezależnych od siebie. Osobne sieci mają min. TPSA, TELBANK, CWŁ WP, operatorzy kablowi. Studzienki kanalizacji należą nie tylko do TPSA (większość), ale również na przykład niektóre są własnością TELBANKU.

Systemy z sektora mediów komunalnych, energetyki należą zwykle do pojedynczych operatorów.



Rys.4.3 Baza obiektów w skali makro składa się w wielu obiektów skali mezo

Na bazę obiektów skali makro składać się powinna odpowiednia ilość obiektów skali mezo (Rys. 4.3.). Podobnie jak w poprzednim przypadku, gdy pewna ilość baz w skali mini składała się na bazę w skali mezo.

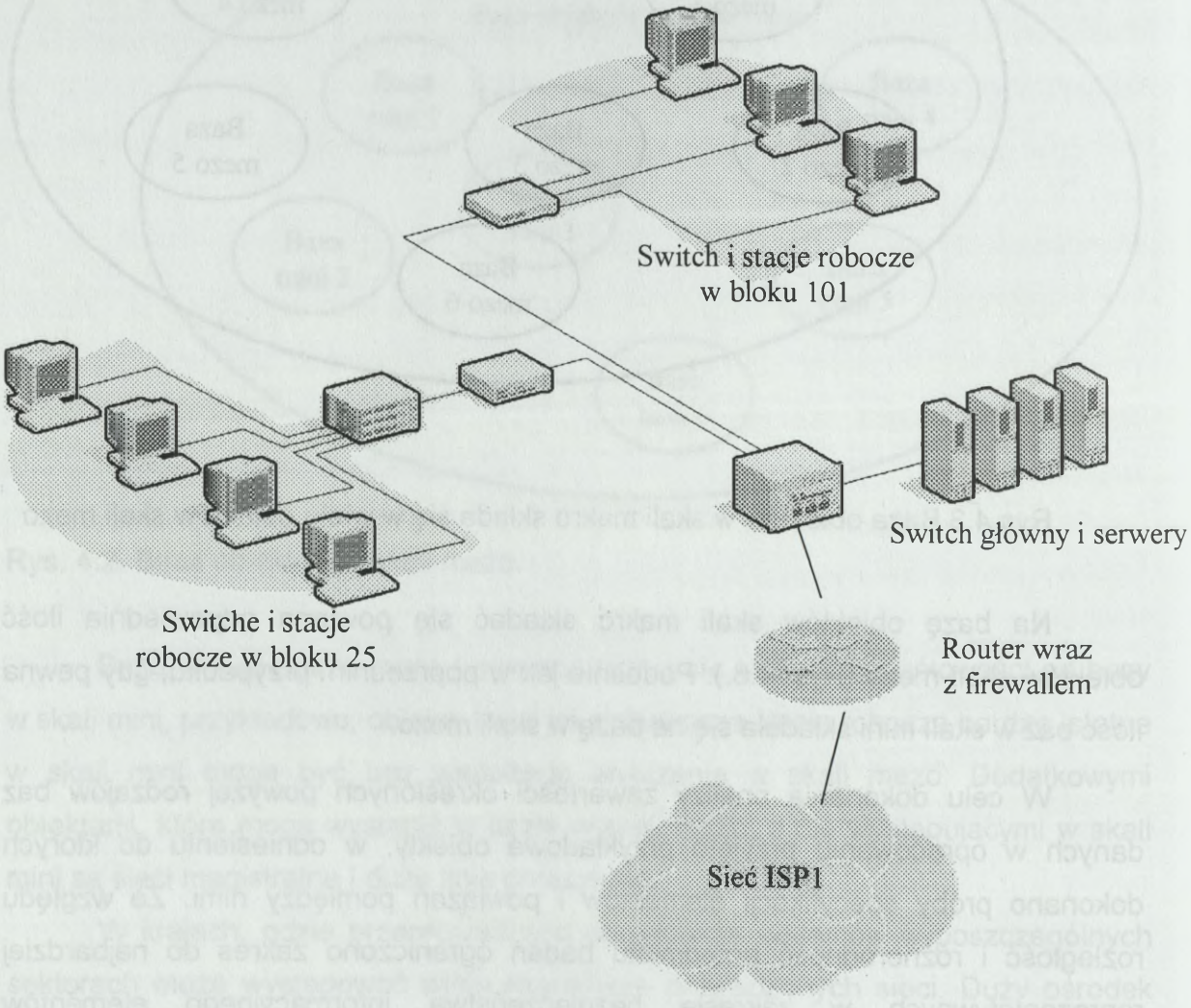
W celu dokonania analizy zawartości określonych powyżej rodzajów baz danych w opracowaniu przyjęto przykładowe obiekty, w odniesieniu do których dokonano próby specyfikacji elementów i powiązań pomiędzy nimi. Ze względu rozległość i różnorodność przedmiotu badań ograniczono zakres do najbardziej reprezentatywnych w zakresie bezpieczeństwa informacyjnego elementów poszczególnych wyodrębnionych rodzajów (grup) obiektów tj. części składowych ich systemów informatycznych.

Przykładowa baza obiektów została określona na trzech poziomach:

- baza ośrodka lokalnego (AON) – skala mikro;
- baza ośrodka regionalnego (miejskiego – Warszawa) – skala mezo;
- baza ośrodka krajowego – skala makro.

Przykładowa baza ośrodka lokalnego została przygotowana na przykładzie sieci kampusowej Akademii Obrony Narodowej. Sieć ta jest jednym z wielu

elementów (obok innych sieci kampusowych i lokalnych) składających się na sieć miejską. Z kolei baza dla sieci miejskiej została opracowana na przykładzie sieci miejskiej Warszawy. Jest ona jednym z elementów sieci krajowej obok innych sieci miejskich czy regionalnych.



Rys. 4.4. Schemat fragmentu sieci AON (opracowanie własne)

Przedstawiona powyżej sieć kampusowa AON składa się z kilku sieci lokalnych rozmieszczonych w poszczególnych budynkach (szkoleniowych i administracyjnych) należących do AON. Na Rys. 4.4. dla celów niniejszej pracy wyróżniono przykładowo dwie takie sieci (w bloku 25 i bloku 101). Każda z sieci lokalnych posiada swoje switche łączące między sobą stacje robocze w budynku. Sieci lokalne bloków łączone są za pomocą światłowodowego okablowania międzybudynkowego do switcha głównego sieci. Do switcha głównego dołączone są

również: serwery usług internetowych oraz router umożliwiający dostęp do sieci Internet za pomocą sieci Internet Service Providera (ISP1 na Rys. 4.4.).

Komputery w sieciach lokalnych mogą pracować samodzielnie, mogą wymieniać się danymi w sieciach peer-to-peer. W sieci pracują następujące serwery:

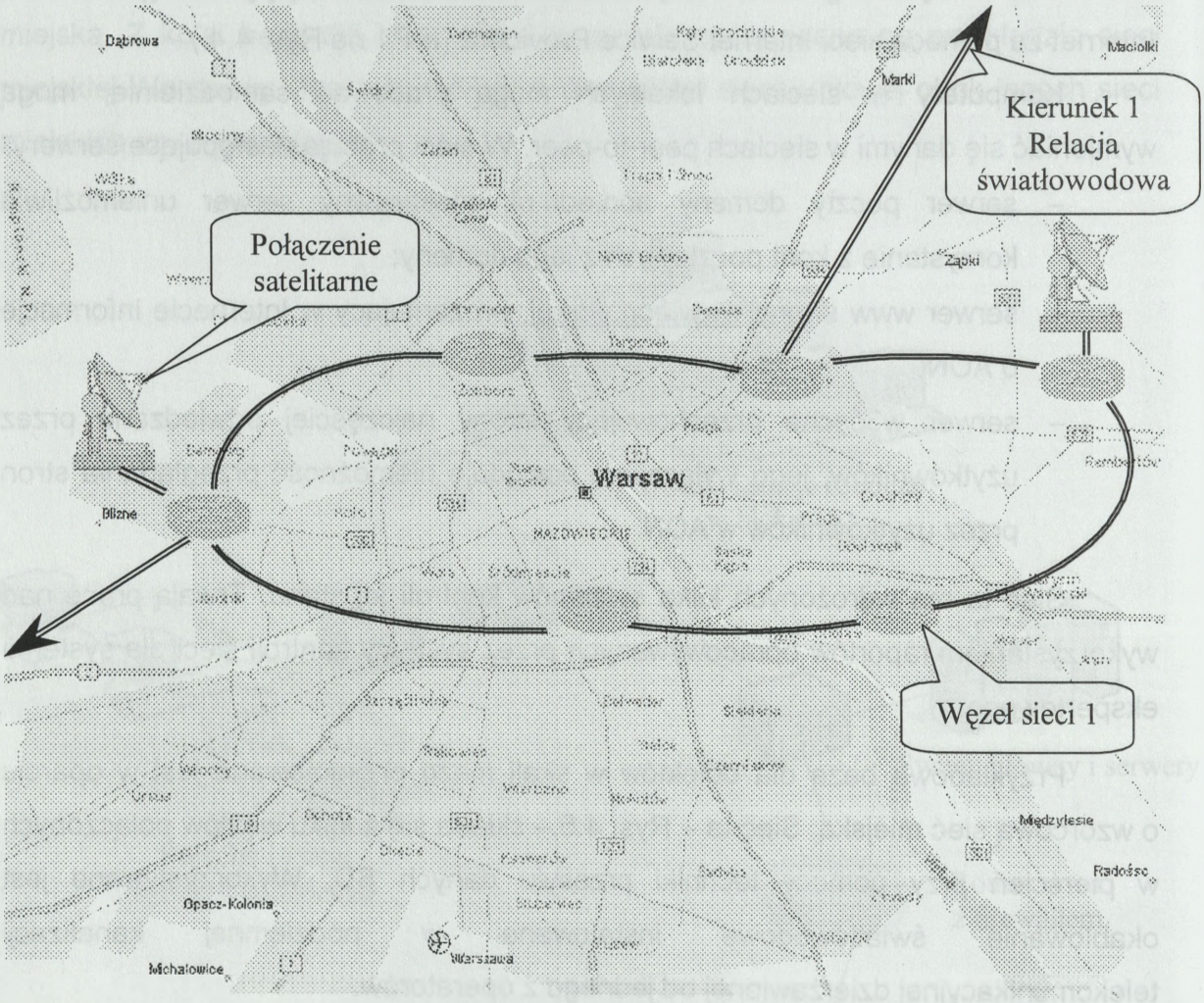
- serwer poczty domeny aon.edu.pl, niesprawny serwer uniemożliwia korzystanie z kont pocztowych z jego domeny;
- serwer www strony www.aon.edu.pl, prezentujący w Internecie informacje o AON;
- serwer w3cache przechowujący strony najczęściej odwiedzane przez użytkowników, jego wyłączenie powoduje niemożność przeglądania stron przez użytkowników w AON.

Sieć ta ma wdrożonych kilka systemów kontroli jej stanu. Trwają prace nad wykorzystaniem raportów opracowywanych przez systemy kontroli sieci dla systemu ekspertowego.

Przykładowa baza dla obiektów w skali mezo przygotowana jest w oparciu o wzorcową sieć miejską. Sieć ta – Rys. 4.5.– składa się z kilku węzłów połączonych w pierścień przy pomocy techniki przesyłu danych SD. Wykorzystywane jest okablowanie światłowodowe instalowane w podziemnej kanalizacji telekomunikacyjnej dzierżawionej od jednego z operatorów.

Do węzłów sieci dołączone są sieci lokalne (skala mikro), a w nich interesujące z punku widzenia tematyki opracowania zasoby: serwery WWW i poczty. Połączenie tej sieci miejskiej z innymi sieciami zapewniają następujące obiekty:

- połączenia światłowodowe łączące tę sieć z sieciami innych ośrodków miejskich;
- połączenia satelitarne zapewniające połączenia długodystansowe;
- połączenia miejscowe np. WIX (Warszawski Punkt Wymiany Internetu) zapewniające wymianę danych z innymi sieciami miejskimi Warszawy.



Rys. 4.5. Przykładowa baza obiektów sieci miejskiej

Zakres przedmiotowy elementów bazy obiektu w skali makro opracowano na przykładzie sektorów telekomunikacji i teleinformatyki o krajowym zasięgu.

W skali makro na terenie RP wyróżnić możemy trzech dużych operatorów posiadających własne sieci światłowodowe.

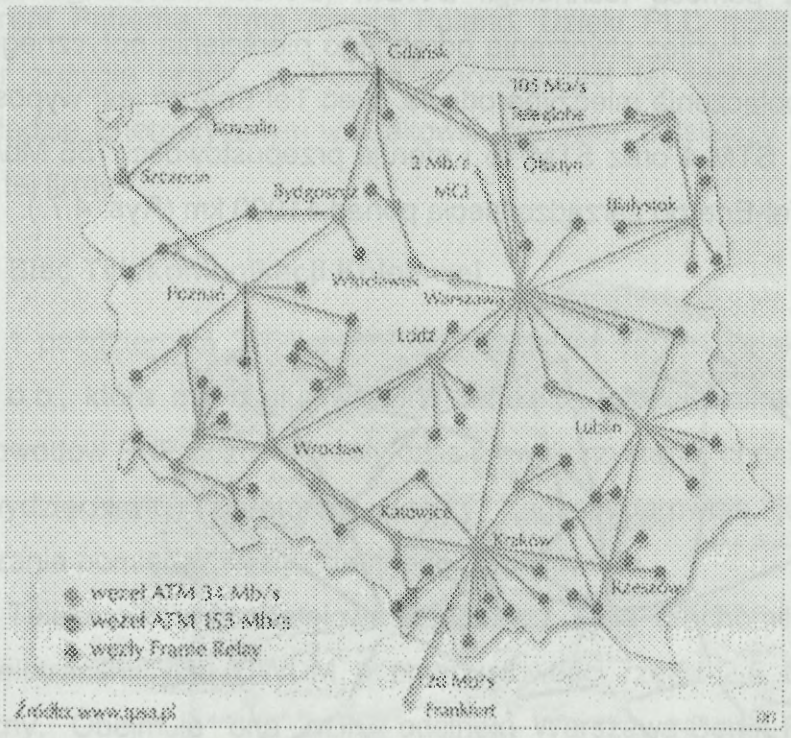
- Telekomunikacja – TPSA;
- Tel-Energo;
- TELEKOMUNIKACJA KOLEJOWA.

Sieć TPS.A

TP SA stale rozbudowuje swoje sieci światłowodowe. Aktualnie posiada łącza pozwalające osiągnąć maksymalną szybkość transmisji 400 Gb/s. Wykorzystano w niej technologię DWDM firmy Lucent Technologies, zwiększającą przepustowość

istniejącej infrastruktury dzięki przesyłaniu jednym przewodem od 40 do 80 fal świetlnych. Zastosowany system WaveStar OLS 400G pozwoli w przyszłości podwoić przepustowość łączy do 800 Gb/s. TPSA łącza swoje wykorzystuje przede wszystkim dla transmisji głosu, w mniejszym stopniu również do transmisji danych.

Dla transmisji internetowych partnerem zagranicznym TPSA do roku 2001 była firma Teleglobe.



Rys. 4.6. Stan sieci Polpak przed rozbudową do STM-16 (źródło: PC Kurier)

Od sierpnia 2001 roku umowa między TP SA a Telią zapewnia polskiemu operatorowi połączenia o sumarycznej przepustowości ponad 600 Mb/s. Pierwsze dwa łącza (typu STM-1) po 155 Mb/s zaczęły już działać w połowie 2001, co oznacza dwukrotne zwiększenie przepustowości połączeń wychodzących z Polski w porównaniu ze stanem z początku roku. Umowa będzie obowiązywać przez rok z możliwością jej przedłużenia i rozbudowy łączy ze światem do 2,5 Gb/s.

TP S.A. dzierżawi miejsce w swojej kanalizacji kablowej innym firmom telekomunikacyjnym.

Plan sieci Polpak przedstawia Rys. 4.6.

Kable Tel-Energo

Na ogólnopolską sieć światłowodową Tel-Energo składają się: sieć szkieletowa, sieci regionalne (wykonane w technologiach OPGW - Optical Ground Wire i ADSS - All-Dielectric Self Supporting) oraz sieci dostępowe, budowane na indywidualne zamówienia klientów. Tel-Energo planuje zwiększyć przepustowość sieci szkieletowej za pomocą technologii DWDM (Dense Wavelength Division Multiplexing). Rozważa również stworzenie odrębnego pierścienia optycznego SDH obsługującego tylko połączenia międzynarodowe. Sieć Tel-Energo jest wyposażona w urządzenia STM-1, STM-4 oraz STM-16 i oferuje przepustowości 155 Mb/s, 622 Mb/s oraz 2,5 Gb/s. Tel-Energo zarządza siecią ponad 11000 km (Rys. 4.7.).

Rys. 4.7. Sieć Tel-Energo wraz z łączami międzynarodowymi (źródło: materiały reklamowe firmy)

Do najważniejszych klientów sieć Tel-Energo zalicza: NOM, Telefonia Lokalna, POL-34, PTC, Polkomtel, TELBANK, Netia, Pro Futuro, Energis, PKO BP, Kredyt Bank PBI, BIG Bank GDAŃSKI, BRE Bank, NASK, AGORA, H. Bauer, i inni

Umowę z Tel-Energo podpisała polsko-amerykańska spółka Pattern Communications, która od lipca 2003r. prowadzi w zakładach energetycznych próby

technologii PLC, umożliwiającej dostęp do Internetu przez gniazdka elektryczne. Firma zawarła już wcześniej takie umowy z 5 zakładami energetycznymi, m.in. z największym w Polsce Górnośląskim Zakładem Elektroenergetycznym SA Pattern Communications wykorzystuje technologię izraelskiej firmy Main.net i chce zaoferować usługi energetycznego Internetu w 2002 r. (najpierw dla ok. 5 tys. odbiorców). W trzech zakładach energetycznych testy PLC prowadzi również konkurent Patterna, firma Ascom Poland. Dzięki PLC możliwy jest także zdalny pomiar zużycia prądu, w Polsce pierwszy test tej technologii prowadzony jest w Zakładzie Energetycznym w Krakowie. Na dzień dzisiejszy nieznana jest szersza oferta tej firmy.

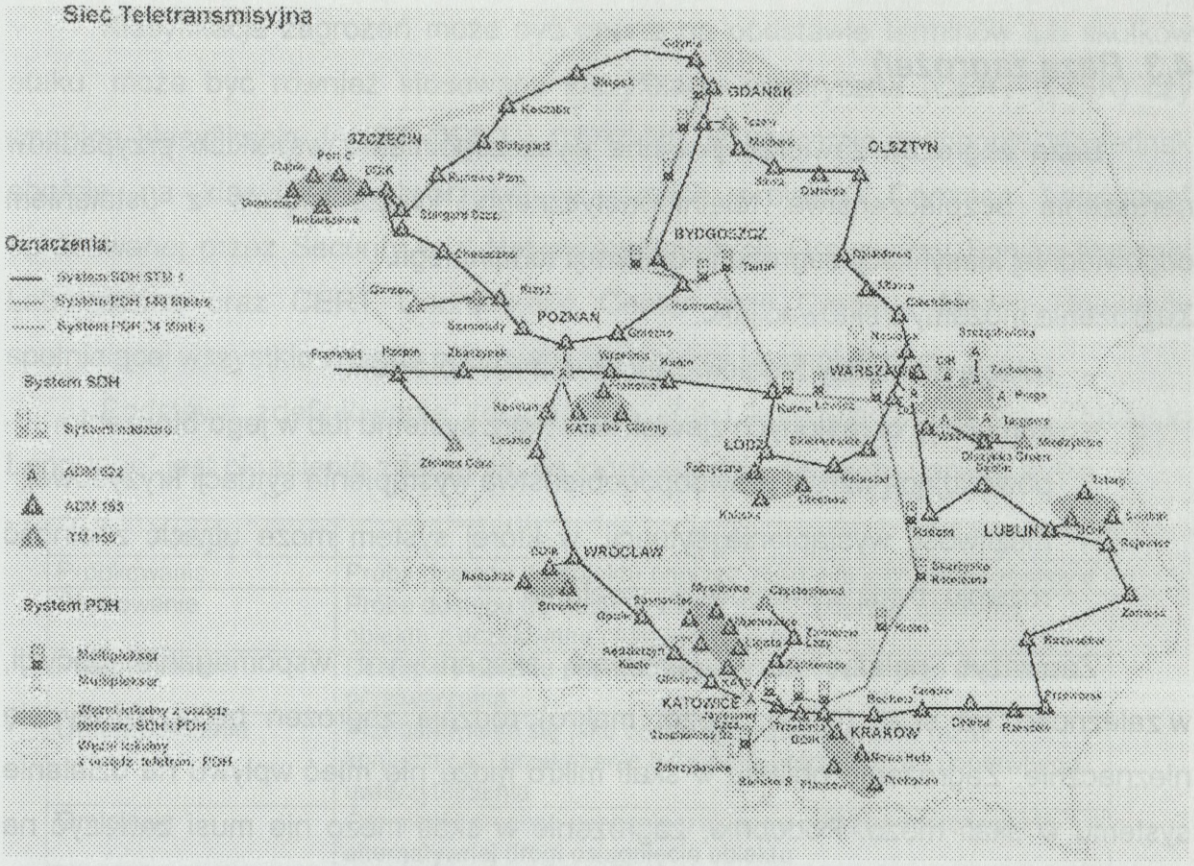
Sieć Telekomunikacji Kolejowej

1 października 2001 z PKP wydzieliła się spółka Telekomunikacja Kolejowa Sp. z o.o., która ma obsługiwać infrastrukturę telekomunikacyjną. Większa część światłowodów PKP pozostaje cały czas niewykorzystana, chociaż część z nich udało się wydzierżawić operatorom, którzy zagospodarowują je, włączając własne urządzenia pomiędzy węzły "ciemnego" światłowodu.

Telekomunikacja Kolejowa wspólnie z Telią uruchomiła pierwszą w Polsce sieć teletransmisyjną DWDM o przepustowości rzędu 2,5 Gb/s na każdy kanał optyczny. Obejmuje ona m.in. odcinek Warszawa-Berlin-Frankfurt n. Menem. Obecnie od TK można wydzierżawić łącza o przepustowości od wielokrotności 2 Mb/s, przez E3 (34 Mb/s), SDH (155 Mb/s STM-1, 622 Mb/s STM-4, 2,5 Gb/s STM-16) aż do 2,5 Gb/s (kanał optyczny) na połączeniach zarówno krajowych, jak i zagranicznych. W przyszłości przyłączenie do urządzeń dostępowych będzie możliwe nie tylko w Warszawie, ale również w Łodzi, Poznaniu i innych miastach. Spółka planuje też udostępnić klientom kanały optyczne o maksymalnej przepustowości 32 x 10 Gb/s.

Rys. 4.8. Sieć światłowodowa Telekomunikacji Kolejowej, dawniej KOPLAK (źródło: materiały reklamowe firmy)

Sieć Telekomunikacji Kolejowej (dawniej KOLPAK) - rysunki 4.8 i 4.9 - posiada dobre możliwości techniczne. Dzięki zastosowaniu specjalnego wagonu pozwalającego układać światłowody wzdłuż torów kolejowych. Na odcinkach pozbawionych rozjazdów i innych utrudnień terenowych w ten sposób można było układać nawet po kilkadziesiąt kilometrów kabla światłowodowego dziennie. Potencjał w postaci rozległej sieci torów kolejowych przechodzącej przez centra praktycznie wszystkich większych miast w Polsce (dworce to doskonała lokalizacja dla węzłów sieci) nie jest jednak w pełni wykorzystany. Problemy finansowe PKP przełożyły się na brak środków inwestycyjnych. Choć budowanie sieci światłowodowej wzdłuż torów jest tańsze od wielu innych metod, to w znacznej mierze zostało sparaliżowane przez niemoc deficytowej firmy państwowej.



Rys. 4.9. Systemy teletransmisyjne wykorzystujące sieć światłowodową Telekomunikacji Kolejowej

4.3. Baza zagrożeń

Baza zagrożeń zawierać powinna wyszczególnienie wszelkich przypadków naruszenia bezpieczeństwa teleinformatycznego. Wiąże się to z ustaleniem odpowiedniej klasyfikacji zagrożeń i unifikacji terminologii.

Zagrożenia możemy traktować jako:

- relację pomiędzy źródłem zagrożeń a ich obiektem;
- kumulację określonych zjawisk wewnątrz systemu lub w jego otoczeniu powodującą wzrost prawdopodobieństwa wystąpienia sytuacji kryzysowej;
- możliwość wystąpienia sytuacji, w której system może utracić zdolność rozwoju, bądź prawidłowego działania.

Zauważyć należy, że w systemie ekspertowym wspomagania decyzji, w zależności od skali (mikro, mezo, makro) rodzaje zagrożeń będą różniły się nieznacznie. Zagrożenie obiektu w skali mikro może nie mieć wpływu na działanie systemu w skali mezo. Podobnie, zagrożenie w skali mezo nie musi zaważyć na sytuację w skali makro. Niektóre zagrożenia występujące w skali makro nie wystąpią w skali mezo i mini.

Zagrożenia ujęte w bazie zagrożeń można, w zależności od ich własności podzielić na trzy grupy.

Własności relacyjne	Własności fizykalne	Czas trwania
Samozagrożenie	Materialne	Krótkotrwałe
Jednostronne	Energetyczne	Długotrwałe
Wzajemne	Informacyjne	Ewoluujące
Bezpośrednie	Ekologiczne	Cykliczne
Pośrednie	Społeczne	Potencjalne
	Ekonomiczne	Realne
	Militarne	

W eksperckim systemie wspomagania decyzji należy opracować standard nazewnictwa i klasyfikacji zagrożeń oparty na odpowiednich zasadach poprawnej taksonomii. Pozwoli to w sposób jednoznaczny zaszeregować dane zdarzenie (zagrożenie) i uniknąć nieścisłości i niejednoznaczności przy współpracy systemu z inżynierami wiedzy lub ekspertami.

Klasyfikacja zagrożeń może być oparta na podstawie terminów lub skutków ataku, może być również stosowana klasyfikacja empiryczna (CERT NASK) czy wspólna klasyfikacja JANET CERT i CERT-NL. W systemie proponuje się jednak oparcie na klasyfikacji stworzonej w ramach projektu „Common Language" opracowanej przez Security and Networing Research Group przy Sandia National Laboratories oraz CERT Coordination Center przy Carnelie Mellon University spełniającej wszystkie zasady poprawnej konstrukcji i kompletności.

Podstawą zdefiniowania konkretnego ataku komputerowego wg. „Common Language" jest określenie zdarzenia, na które składają się akcja i obiekt ataku.

Akcja	Opis
Próbkowanie	Próba dostępu do obiektu poprzez zbadanie jego charakterystyki
Skanowanie	Próba dostępu do wielu obiektów na raz poprzez ustalenie obiektu z oczekiwaną charakterystyką
Przepełnienie	Dostęp do obiektu poprzez nagłe przepełnienie jego możliwości przetwarzania
Uwierzytelnienie	Przedstawienie się jako osoba uprawniona oraz w razie konieczności przekazanie informacji potrzebnej do poprawnego uwierzytelnienia
Ominięcie	Ominięcie procesu zabezpieczającego poprzez zastosowanie alternatywnej drogi osiągnięcia obiektu
Podszywanie	Przedstawianie się, w trakcie połączenia sieciowego, jako użytkownik posiadający prawo dostępu do zasobów
Czytanie	Dostęp z prawami czytania do informacji przez osobę nieuprawnioną
Kopiowanie	Dostęp z możliwością kopiowania do informacji przez osobę nieuprawnioną
Kradzież	Przejęcie zasobów przez osobę nieuprawnioną bez pozostawienia kopii w uprawnionej lokalizacji
Modyfikacja	Zmian zawartości lub charakterystyki obiektu ataku
Usunięcie	Usunięcie (zniszczenie) obiektu ataku

Dodatkowymi elementami do klasyfikacji ataku (zagrożenia) są: narzędzie ataku, słabość układu który został zaatakowany oraz skutek (rezultat) ataku.

Narzędzia	a. atak fizyczny b. dostęp do informacji c. komendy systemowe d. skrypt lub program e. obiekt autonomiczny f. zestaw oprogramowania służący do ataku g. narzędzie ataku rozproszonego h. przechwycenie danych
Słabość systemowa	a. konstrukcja b. implementacja c. konfiguracja
Rezultat	a. nieautoryzowane rozszerzenie dostępu b. ujawnienie informacji

4.4. Baza scenariuszy

Baza scenariuszy zagrożeń obejmuje zarówno rodzaj ataku (zagrożenia) jak i jego bezpośrednie i dalsze skutki. Scenariusze podzielone są na trzy grupy i dotyczą ataków na obiekty w odpowiedniej skali: mini, mezo, makro. Baza scenariuszy powinna być przechowywana w zewnętrznej bazie danych, z której, w trakcie pracy z systemem PC-Shell, użytkownik ma możliwość wyboru jednego bądź kilku scenariuszy. Pozwoli to na „pielęgnację" bazy scenariuszy, jej rozwój i wprowadzanie zmian.

Użytkownik systemu powinien mieć możliwość wyboru scenariusza zagrożeń, w zależności od:

- rodzajów ataków;
- rodzajów obiektów podlegających zagrożeniu;
- skali rozpatrywanego problemu;
- skutków zagrożeń.

Pozwoli to na wykorzystanie systemu nie tylko jako wspomagającego podjęcie decyzji w sytuacji kryzysowej, lecz również na przeszkolenie użytkowników na wypadek jej wystąpienia. Udzielanie konsultacji przez system pozwala na przeszkolenie personelu i wykorzystanie tegoż systemu w celach prewencyjnych i edukacyjnych. W trakcie projektowania i budowy systemu należy zwrócić uwagę na możliwość tworzenia nowych scenariuszy, zmiany już istniejących, łączenie i wzajemne uzupełnianie.

Parametryzacja bazy wiedzy systemu PC-Shell pozwala na wielokrotne i wielorakie wykorzystanie istniejących scenariuszy poprzez zmianę odpowiednich współczynników czy wielkości charakteryzujących:

- obiekty podlegające zagrożeniom;
- rodzaje zagrożeń;
- skutki zagrożeń.

Wiąże się to z rygorystycznym przestrzeganiem nazewnictwa przyjętego przy tworzeniu baz zagrożeń i obiektów w systemie.

4.5. Baza strategii przeciwdziałania

Baza strategii przeciwdziałania obejmuje zarówno wszelkie działania prewencyjne mające nie dopuścić do zagrożenia obiektów w systemie, jak również działania, które należy podjąć, gdy do takiego zagrożenia (incydentu – ataku informacyjnego) już doszło. Strategie te muszą być dostosowane do rodzaju i wielkości obiektu oraz uwzględniać maksymalną liczbę możliwych zagrożeń i scenariuszy ich powstawania.

Strategie przeciwdziałania (w rozumieniu zbioru dopuszczalnych wariantów rozwiązania problemu związanego z potencjalnym bądź zaistniałym zagrożeniem) możemy podzielić na :

- aktywne:
  - antycypacyjne;
  - prewencyjne;
- reaktywne:
  - repulsyjne;
  - likwidacyjne.

Dane do bazy strategii przeciwdziałania otrzymywane są poprzez wykorzystanie wszelkich dostępnych metod pozyskiwania wiedzy od ekspertów. W bazie może istnieć kilka (kilkanaście, kilkadziesiąt czy też kilkaset) strategii stanowiących odpowiedź na zaistnienie określonego zagrożenia (zagrożeń). Ponieważ zawierać mają wiedzę na temat zapobiegania i usuwania skutków, stąd też bazy strategii podobnie jak i opisane uprzednio bazy zagrożeń i scenariuszy muszą być na bieżąco modyfikowane i uaktualniane w ślad za ewolucją potencjalnych czynników zagrożeń i możliwościami w zakresie przeciwdziałania tym zagrożeniom.

Dostosowane baz strategii do rodzaju i wielkości obiektu nie oznacza ich pełnej autonomii. Powtarzalność ogólnych struktur obiektów przy uwzględnieniu ich hierarchiczności pozwala na pewną unifikację zawartości tych baz danych z pewnymi modyfikacjami wynikającymi z wielkości i różnorodności obiektów. Stąd też jedną z podstawowych funkcji systemów ekspertowych powinna być możliwość dystrybucji (rozpowszechniania) danych o nowych (powstających) zagrożeniach oraz środkach i sposobach przeciwdziałania im.

4.6. Moduł wnioskujący

W odróżnieniu od konwencjonalnych systemów, systemy eksperckie nie zawierają jawnego opisu sposobu rozwiązania danego problemu (algorytmu). To system ekspercki, 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.

Moduł wnioskujący zawiera procedury umożliwiające operowanie na wiedzy systemu w celu wyciągania wniosków (poszukiwania rozwiązań). Opracowano do tej pory wiele metod wnioskowania. Użycie konkretnej metody zależy w dużej mierze od sposobu reprezentacji wiedzy w bazie wiedzy. Na przykład dla regułowej reprezentacji wiedzy opracowano metody prostego i odwrotnego łańcucha wnioskowania (wnioskowania wstępującego i zstępującego).

Można wyróżnić następujące sposoby wnioskowania, stosowane w systemach eksperckich:

- wnioskowanie na podstawie reguł;
- drzewo wnioskowania;
- wnioskowanie na podstawie opracowanych modeli;
- wnioskowanie z wykorzystaniem ramowej reprezentacji wiedzy;
- wnioskowanie na podstawie analizy przypadków.

Baza wiedzy stanowi tylko podstawę do wnioskowania w systemie eksperckim. Wnioskowaniem zajmuje się moduł wnioskowania. Zadaniem modułu wnioskowania jest znalezienie przesłanek potwierdzających postawiona 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.

Maszyna wnioskująca bezpośrednio operuje na bazie wiedzy i może w trakcie działania zmieniać jej zawartość przez wprowadzenie do niej nowych faktów.

Wszystkie stosowane metody wnioskowania maja na celu znalezienie jakieś drogi (najczęściej najkrótszej), prowadzącej do któregoś z możliwych rozwiązań (lub

do rozwiązania najlepszego pod jakimś względem). Najczęstszymi ograniczeniami efektywności procedur wnioskowania są zasoby komputera, takie jak objętość pamięci operacyjnej oraz szybkość pracy procesora.

Maszyna wnioskująca ma za zadanie analizować przesłanki i wyciągać z nich wnioski (konkluzje) lub znaleźć przesłanki potwierdzające postawioną hipotezę. Możemy mieć oczywiście do czynienia z jeszcze innymi strategiami wnioskowania, niemniej jednak dwie wymienione są najczęściej używane, a języki zorientowane na przetwarzanie wiedzy są na ogół wyposażone w stosowne konstrukcje programowe.

Poszukiwanie rozwiązań nie jest problemem samym w sobie. System ekspertowy nie jest po to, by poszukiwał rozwiązań. On ma je podawać (proponować użytkownikowi; najlepiej jeżeli nie będzie to jedno rozwiązanie, lecz kilka, uszeregowanych np. według wiarygodności). Stąd, jeżeli będziemy dysponować pełniejszą, wiedza (szersza bazą wiedzy), to na ogół będziemy mogli ograniczyć poszukiwania, co uczyni nasz system bardziej efektywnym (a przynajmniej szybszym).

Ogólnie rzecz biorąc, poszukiwanie rozwiązania polega na przejściu od stanu początkowego (założeń, danych wyjściowych), przez stany pośrednie (etapy rozwiązania) do stanu końcowego (celu, wyniku). Poszukiwanie odbywa się w tak zwanej przestrzeni poszukiwań. Jeżeli przestrzenią poszukiwań jest zbiór strategii przeciwdziałania zdeterminowany przez rodzaj obiektu i zagrożenia (efekt ataku) to efektem poszukiwania jest określenie podzbioru strategii przeciwdziałania adekwatnych do określonej sytuacji kryzysowej systemu informacyjnego i możliwych scenariuszu jej rozwoju. Etapem pośrednim w poszukiwaniu jest ustalenie przyczyn zaistniałej sytuacji problemowej na podstawie dostępnych przesłanek czyli widocznych efektów (skutków) ataku bądź opisu stanu systemu zabezpieczenia w działalności prewencyjnej poprzez skonfrontowanie tych danych z zasobami baz wiedzy.

4.7. Moduł ewaluacji strategii przeciwdziałania

Wynikiem działania opisanego uprzednio modułu wnioskującego jest zbiór strategii dopuszczalnych (propozycji strategii działania) w określonej sytuacji problemowej. Ewaluacja jest konieczna w celu umożliwienia wyboru jednej z nich – strategii optymalnej. Ponieważ sytuacje kryzysowe z zakresie bezpieczeństwa informacyjnego trudno jest opisać jednoznacznymi równaniami, w procesie ewaluacji niezbędna jest wiedza ekspertów z różnych dziedzin, którzy oceniają zaistniałą sytuację za pomocą określonych kryteriów.

Jeżeli przyjmiemy:

$S=\{ s_1, s_2, \ldots s_m\}$ – zbiór dopuszczalnych strategii przeciwdziałania;

$K=\{ k_1, k_2, \ldots k_n\}$ – zbiór kryteriów;

$W=\{ w_1, w_2, \ldots w_n\}$ – zbiór współczynników wagowych ustalanych przez ekspertów z założeniem, że $\Sigma w_i=1$;

to można wyodrębnić macierz współczynników, których wielkość określona zostaje przez ekspertów na bazie określonych strategii i kryteriów ocen z wagami.

	Kryteria ocen z wagami			
Strategie	w_1 k_1	w_2 k_2	... ...	w_n k_n
s_1				
s_2	Macierz współczynników			
s_m				

Dobór zabezpieczeń powinien być efektem określenia celów i strategii przyjętych w polityce bezpieczeństwa. Racjonalna strategia zarządzania ryzykiem opiera się o wyznaczenie poziomu zabezpieczenia, który jest funkcją wartości kryteriów poufności, wiarygodności, dostępności, rozliczalności, autentyczności i niezawodności. Poziom ten powinien przyjmować największą wartość ze zbioru wartości kryteriów. Mimo zastosowania odpowiednich zabezpieczeń ryzyko nie jest nigdy redukowane do zera. Pozostaje tzw. ryzyko szczątkowe. Jeżeli jego poziom nie jest akceptowalny, to należy wprowadzić dodatkowe elementy zabezpieczenia (wiąże się to ze wzrostem kosztów). W przeciwnym wypadku należy dokładnie wyspecyfikować relacje między zasobami i zagrożeniami oraz rozważyć przyjęcie

alternatywnych zabezpieczeń. W procesie dobierania zabezpieczeń wziąć trzeba pod uwagę różnorodne ograniczenia. Ograniczenia mogą być natury:

- finansowej - koszt zabezpieczeń nie może przewyższać kosztu chronionych zasobów. Warto jednak rozważyć pominięcie tego ograniczenia w przypadkach, gdy prawdopodobieństwo wystąpienia zagrożenia jest bardzo duże;
- czasowej - termin wdrożenia lub działania zabezpieczenia musi uwzględniać wymagania systemu;
- technicznej - nie w każdym przypadku możliwe jest użycie optymalnych rozwiązań technicznych. Rozwiązania sprzętowe mogą być zastąpione rozwiązaniami programowymi;
- organizacyjnej - nie zawsze jest możliwe wdrożenie zabezpieczenia ze względu na przyjęte rozwiązania organizacyjne;
- socjologicznej - rozwiązania optymalne mogą być nie akceptowane przez personel;
- środowiskowej - otoczenie może nie pozwolić zastosować wybranych zabezpieczeń;
- osobowej - brak personelu będącego w stanie spełnić wymagania rozwiązania;
- prawnej - przepisy mogą określać użycie specyficznych rozwiązań.

Zestaw zastosowanych kryteriów (zarówno w zakresie ilościowym jak i jakościowym) powinien być dobierany w zależności od obiektu i rodzaju zagrożenia informacyjnego. Otrzymana macierz współczynników (np. według systemu punktowego) powinna pozwalać na określenie przez system strategii optymalnej, wybranej nie tylko na podstawie determinant określonych przez rodzaj obiektu i zagrożenia ale również wiedzy ekspertów z określonych dziedzin w uwzględnieniu istniejących uwarunkowań poza systemowych. Udział ekspertów to przede wszystkim kalkulacje koszt – zysk dla efektów ewentualnie podejmowanego działania nie tylko w aspekcie czysto finansowym.

4.8. Moduł decyzyjny

Decyzja podejmowana jest w określonej sytuacji decyzyjnej gdy postrzegana rzeczywistość odbiega od tego co być powinno. Efektem decyzji powinny być dane (działania) regulacyjne przywracające proces (system) do stanu oczekiwanego. Opracowanie strategii, jako decyzji dotyczącej wyboru sposobu realizacji założonego celu, wiąże się z jednoczesną oceną stopnia odpowiedzialności własnych kompetencji i zasobów do uwarunkowań tworzonych przez otoczenie.

Decyzja podjęta przez użytkownika jest wspomagana przez system poprzez wybór ze zbioru dopuszczalnych strategii kilku proponowanych rozwiązań, lecz decyzję o wyborze właściwej strategii uzależnia się od wiedzy ekspertów. Oznacza to, iż w oparciu o wagi ustalone przez ekspertów (np. poprzez zaprojektowanie odpowiedniej tablicy decyzyjnej w arkuszu kalkulacyjnym) zostanie podjęta optymalna w danych warunkach decyzja.

Moduł decyzyjny proponuje decyzję optymalną spośród dopuszczalnych na podstawie wyników otrzymanych z modułu ewaluacji, ich zliczenia i porównania. Decyzje te mogą mieć zarówno charakter prewencyjny jak również dotyczyć usuwania skutków ataku, naruszenia bezpieczeństwa systemu informacyjnego. Istnieje możliwość określenia drogą symulacji i prognozowania skutków podjętych decyzji.

Decyzje podejmowane z wykorzystaniem systemów ekspertowych oparte są na wiedzy ekspertów bez konieczności ich bezpośredniego stałego zatrudniania.

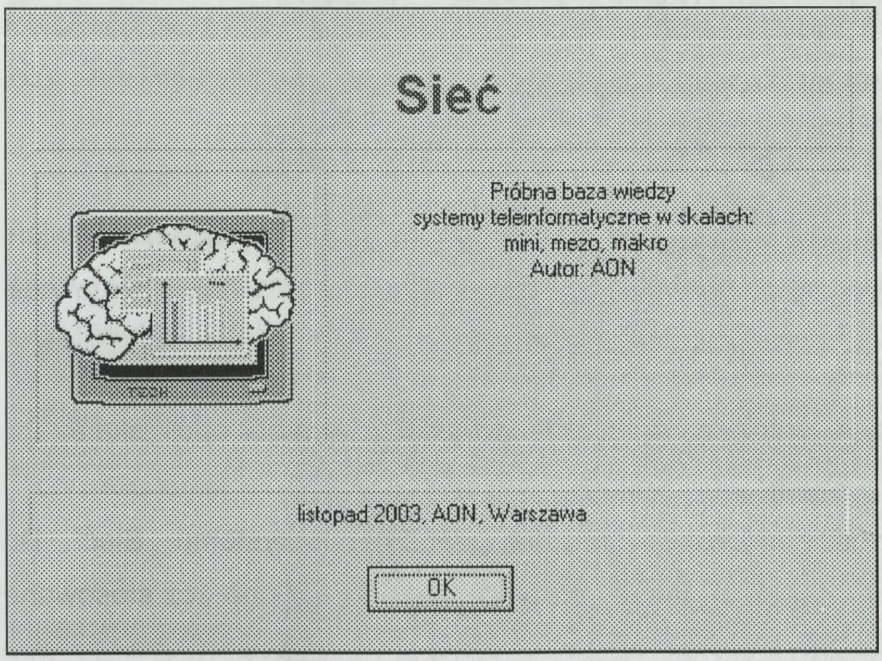
5. Przykłady zastosowania pakietu SPHINX

Przykładowa aplikacja dla skali mini została przygotowana na przykładzie sieci kampusowej Akademii Obrony Narodowej.

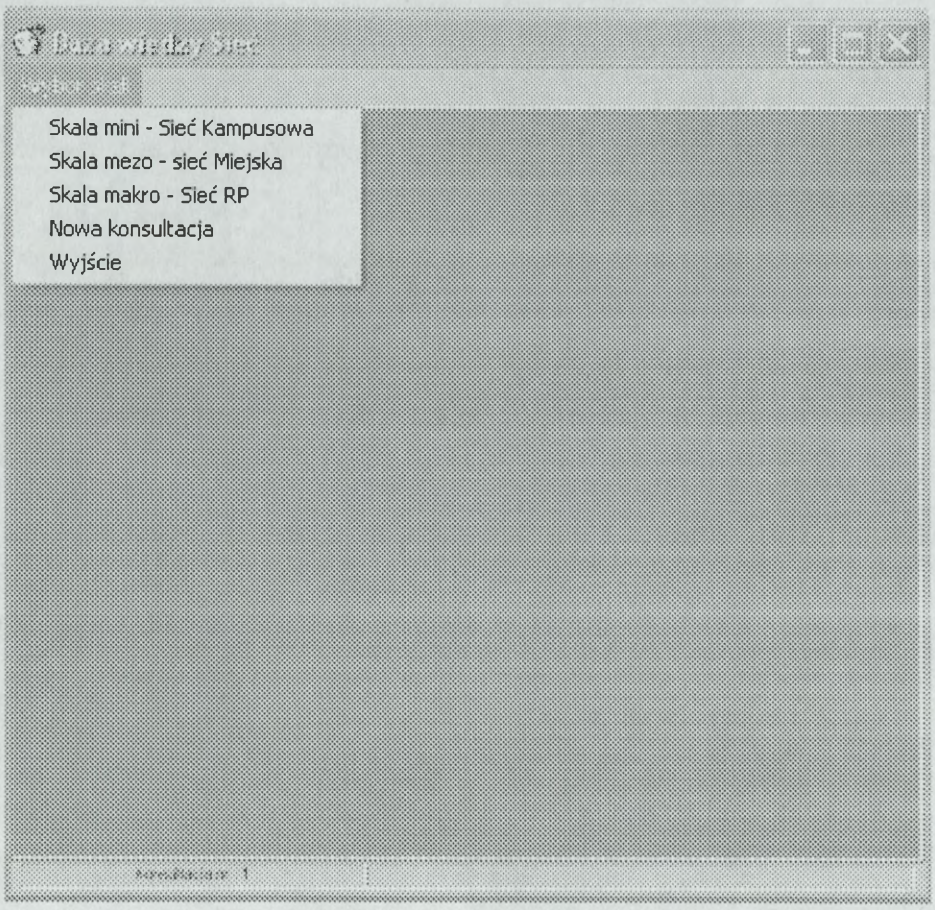
Sieć ta ma wdrożonych kilka systemów kontroli jej stanu. Trwają prace nad wykorzystaniem raportów opracowywanych przez systemy kontroli sieci dla systemu ekspertowego.

Zaprezentowano poniżej wyniki pracy systemu raportującego stan przykładowej sieci kampusowej.

Program został napisany dla pracy w środowisku PC-Shell. Po uruchomieniu zgłasza się stroną tytułową (patrz rysunek 5.1.).



Rys. 5.1. Tytułowe okienko programu

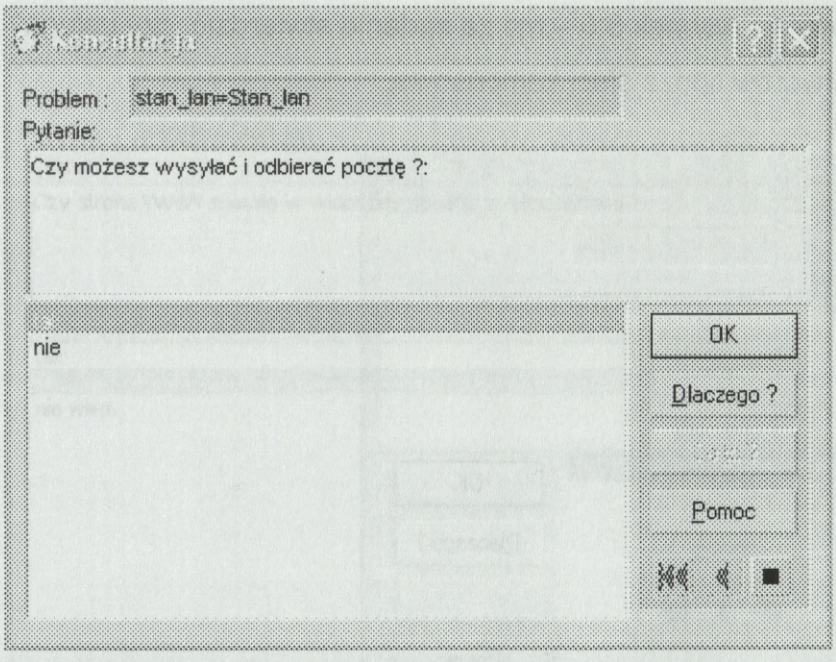


Rys. 5.2. Widok menu

Pracę z programem umożliwia proste menu (rys. 5.2.). Wybór opcji „skala mini" umożliwia przeprowadzenie konsultacji.

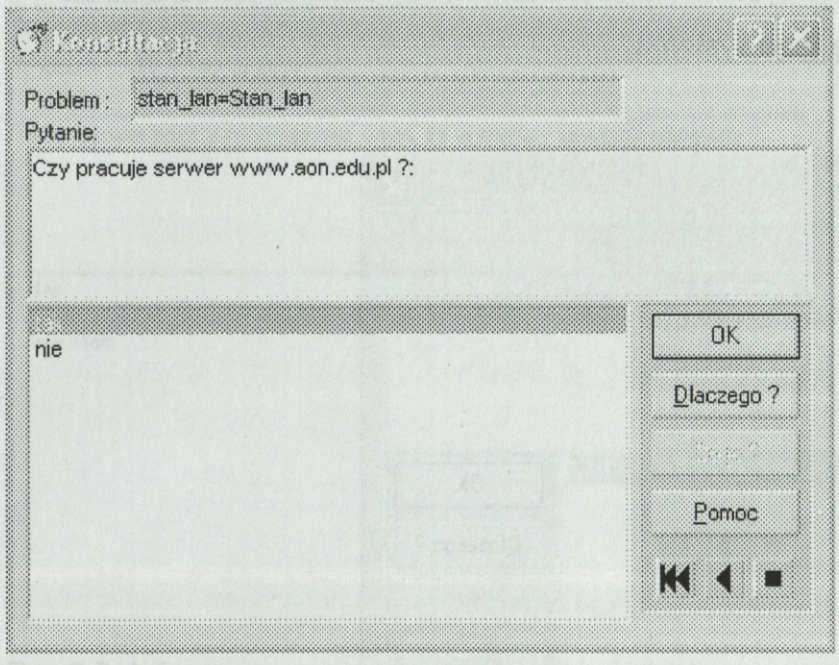
W czasie konsultacji ekspert odpowiada na pytania systemu „Sieć". Do odpowiedzi wykorzystuje dane z systemu zarządzania siecią lub poprzez stosowane „ad hoc" metody.

Odpowiedź na pytanie ukazane na rysunku 5.3. może wymagać np. próbnego uruchomienia aplikacji pocztowej. Innym sposobem, nad którym pracuje zespół tworzący system „Sieć", jest zastąpienie pytania eksperta specjalnym programowym testem.



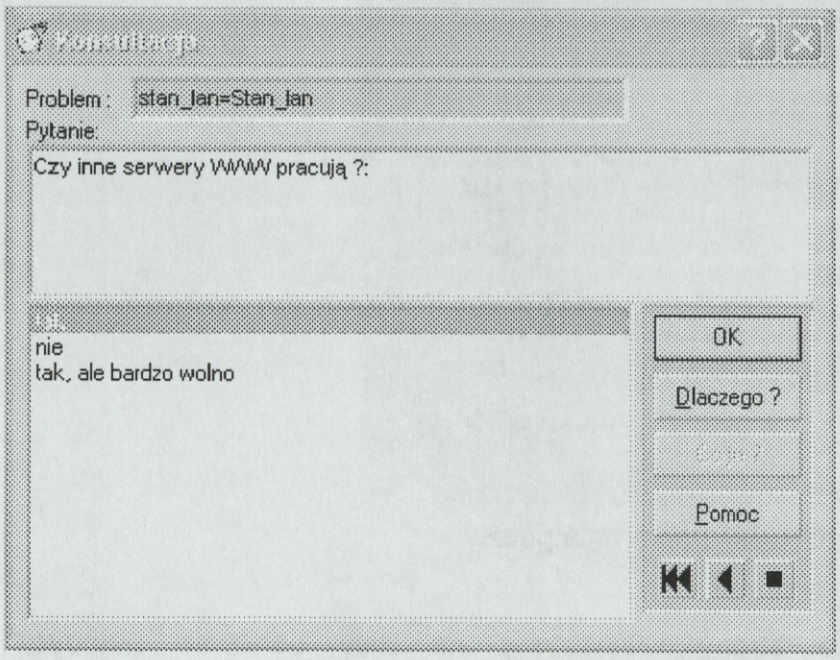
Rys. 5.3. Konsultacja o możliwość odbierania poczty

Pytanie dotyczące strony WWW ma na celu ustalenie „kondycji" najważniejszego serwisu informacyjnego – rysunek 5.4. Serwer ten jest jednym z serwerów sieci kampusowej.



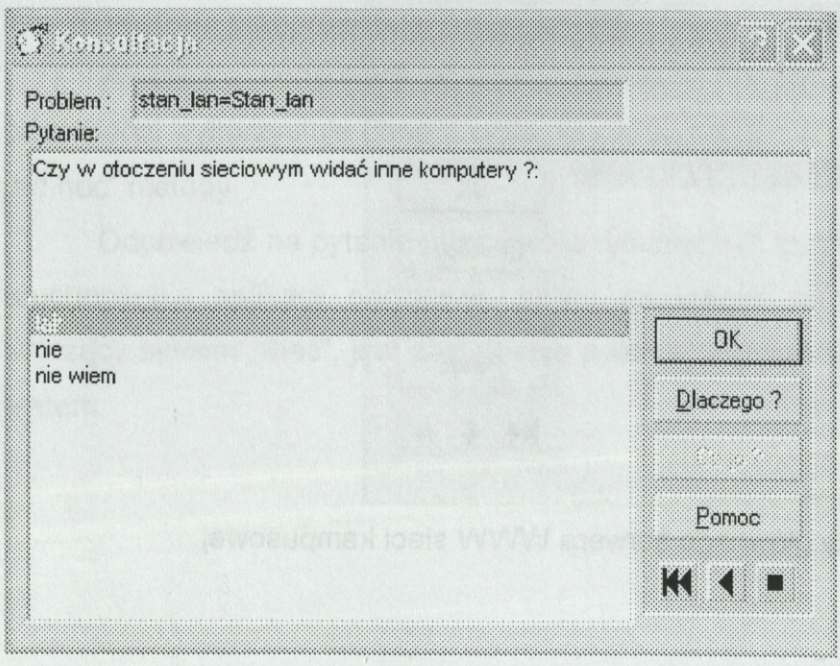
Rys. 5.4. Określenie pracy głównego serwera WWW sieci kampusowej.

Pytanie o inne serwery – rysunek 5.5 – ma za zadanie stwierdzić, czy usługa WWW jest dostępna z serwerów spoza sieci kampusowej.

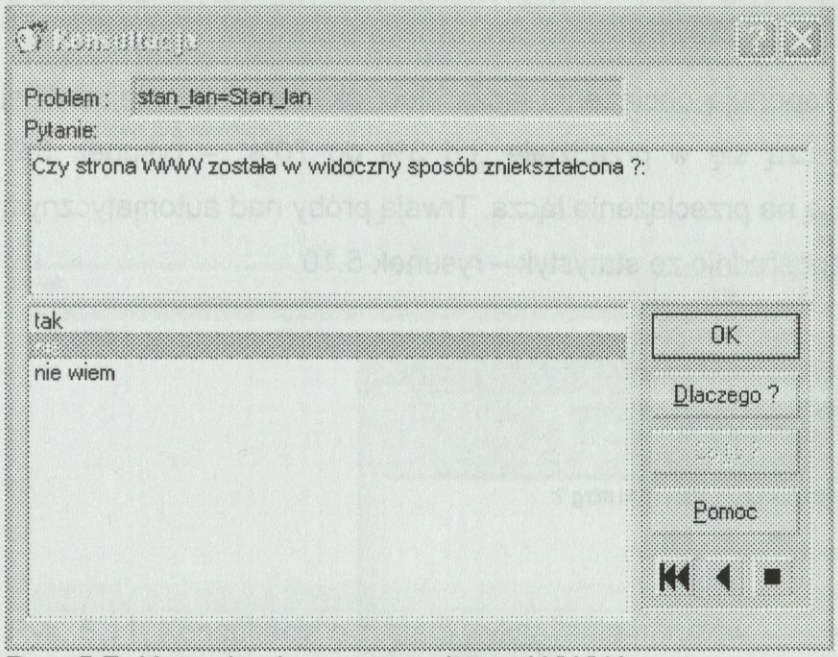


Rys. 5.5. Dostęp do innych serwerów WWW

W sieci AON pracuje system Microsoft Networking. Kolejna konsultacja za zadanie określić kondycję tego systemu.

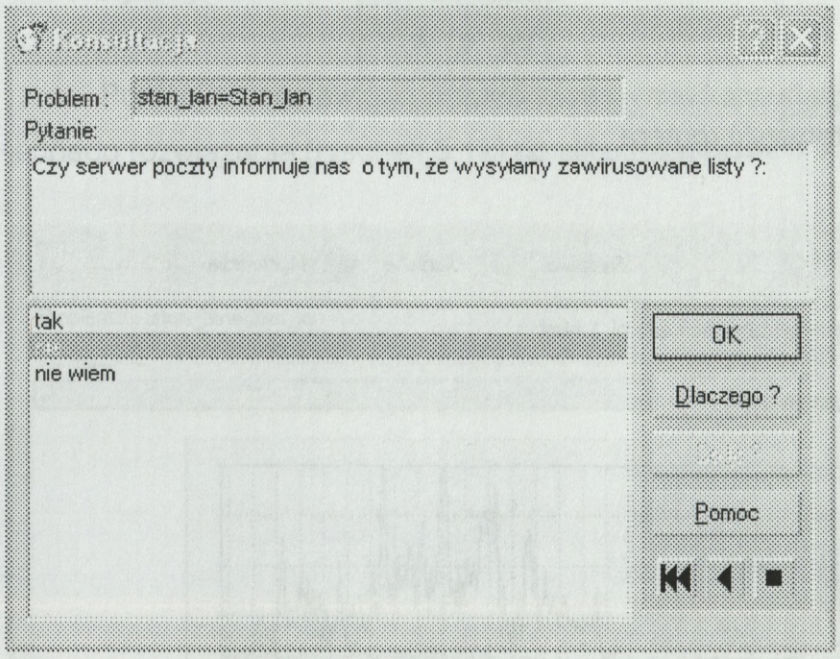


Rys. 5.6. Sprawdzenie funkcjonowania sieci MS Networking



Rys. 5.7. Konsultacja o pracę witryny WWW

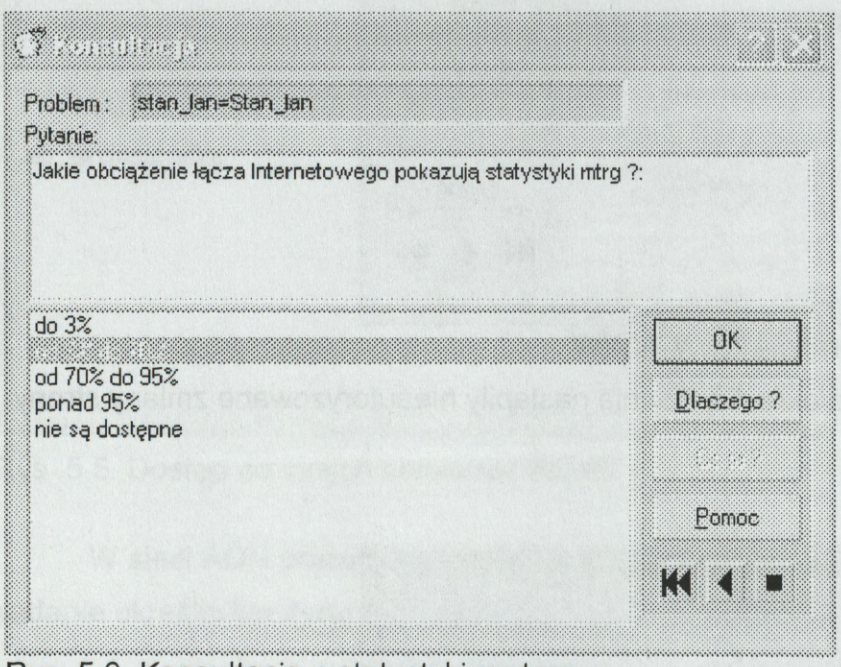
Konsultacja powyższa ma określić, czy nie nastąpiły nieautoryzowane zmiany strony WWW.



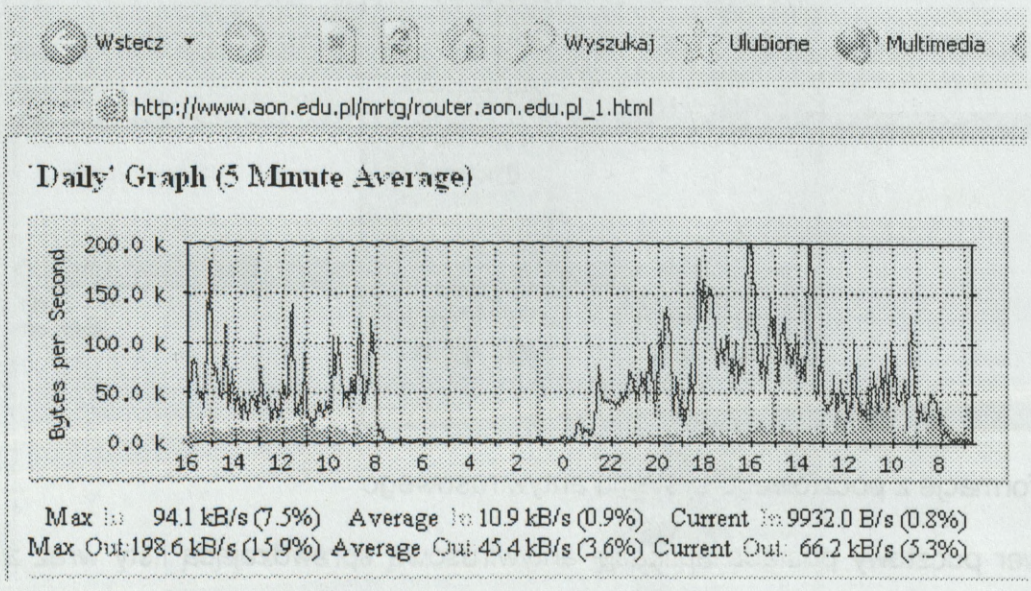
Rys.5.8. Informacje z pocztowego systemu antywirusowego

Serwer pocztowy posiada aplikację antywirusową sprawdzającą listy wraz z załącznikami, o ile przechodzą przez serwer AON. Duża ilość takich meldunków może świadczyć o pewnym zawirusowaniu komputerów w sieci.

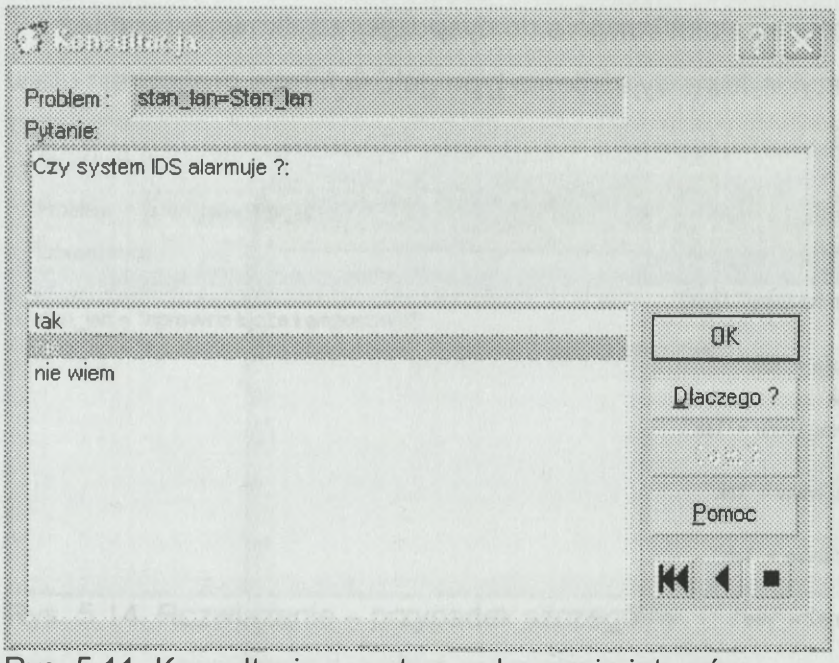
Statystyki routera raportują obciążenie łącza sieci akademickiej. Najbardziej pożądane wartości mieszczą się w przedziale „od 3% do 70%" – rysunek 5.9. Większe wartości wskazują na przeciążenie łącza. Trwają próby nad automatycznym odczytem tej wartości bezpośrednio ze statystyk – rysunek 5.10.



Rys. 5.9. Konsultacja o statystyki routera



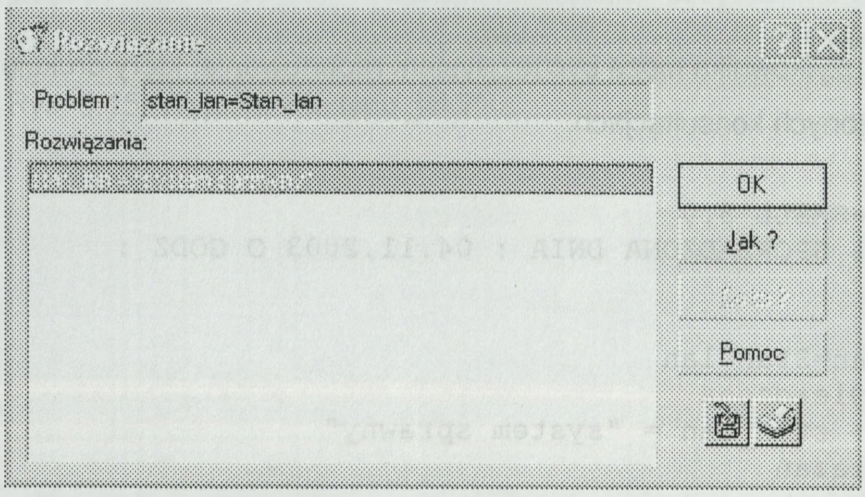
Rys. 5.10. Przykładowe statystyki routera



Rys. 5.11. Konsultacja o system wykrywania intruzów

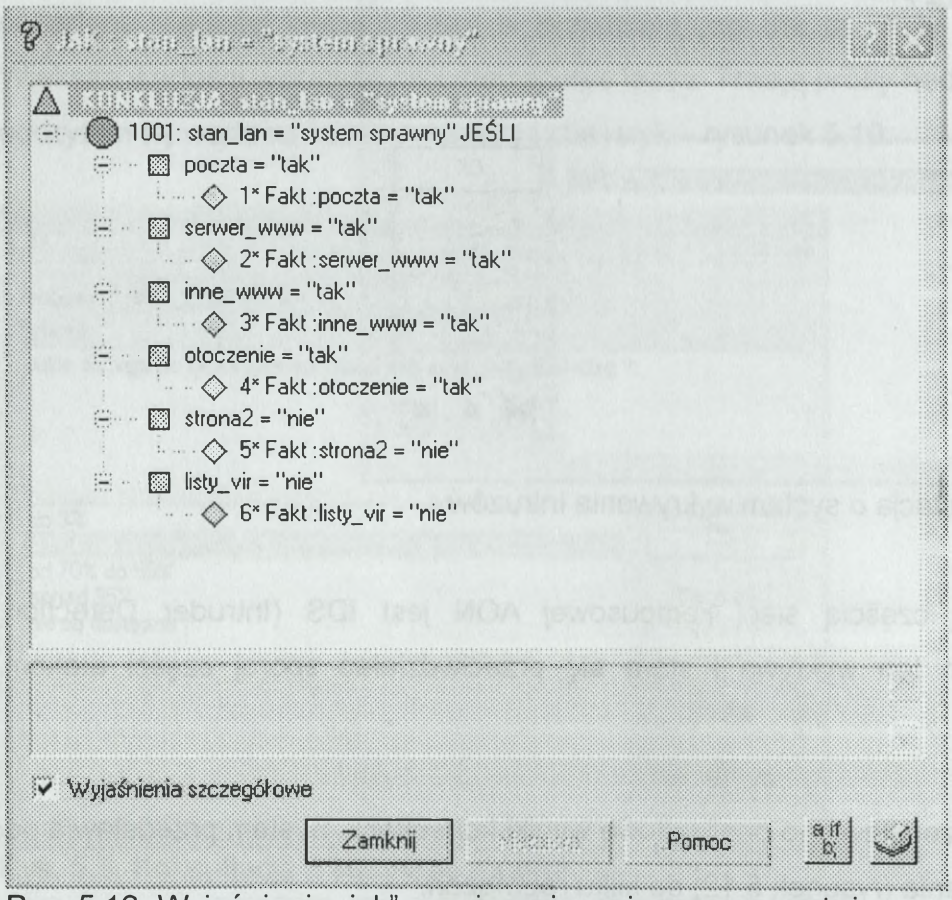
Integralną częścią sieci kampusowej AON jest IDS (Intruder Detection System). System ten wykrywa i stara się przeciwdziałać sporej części atakom informatycznym.

Poniżej przedstawiono przykładowe wyniki konsultacji. System przedstawia od jednego rozwiązania (rysunek 5.12) do kilku rozwiązań.



Rys. 5.12. Rozwiązanie: system sprawny.

Po zakończeniu procesu wnioskowania do dyspozycji użytkownika jest też cały system wyjaśnień. Przykładowe wyjaśnienia typu „jak" w formie graficznej przedstawia rysunek 5.13.

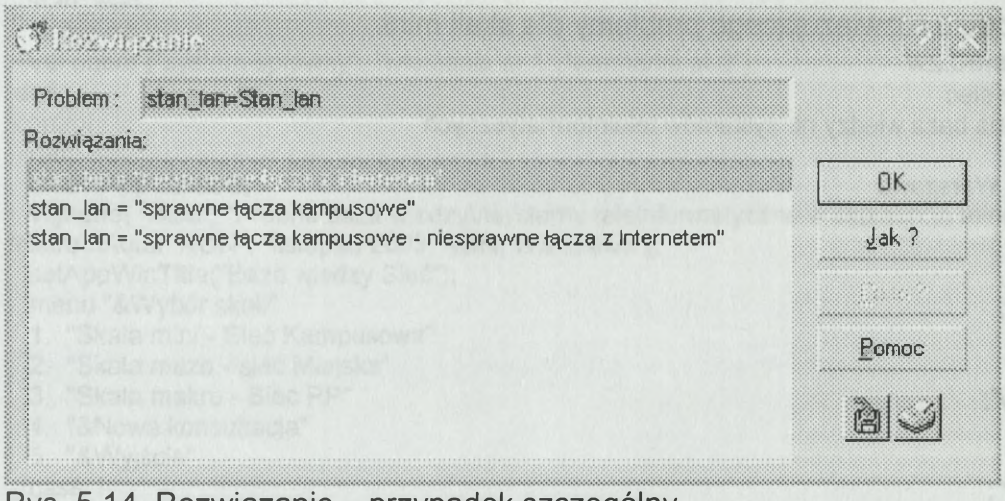


Rys. 5.13. Wyjaśnienie „jak" rozwiązania opisanego powyżej

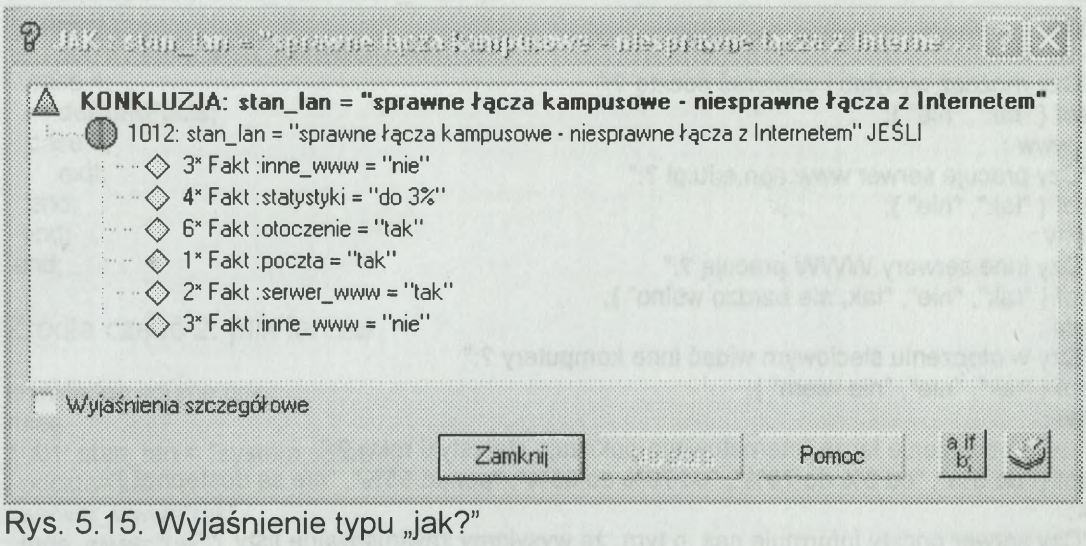
Użytecznym dla użytkownika może być też system raportowania, który drukuje raporty po przeprowadzonych konsultacjach.

```
RAPORT SYSTEMU PC-SHELL 3.0
KONSULTACJA NR : 1 SPORZĄDZONA DNIA : 04.11.2003 O GODZ :
09:43
ROZWIĄZANIA
Hipoteza : stan_lan=Stan_lan
TAK: hipoteza potwierdzona
Rozwiązanie nr 1 : stan_lan = "system sprawny"
Koniec listy rozwiązań
```

Przypadek szczególny: wiele rozwiązań. Program dopuszcza przedstawienie wielu rozwiązań, jeżeli tylko opisują one dany przypadek – rysunek 5.14.



Rys. 5.14. Rozwiązanie – przypadek szczególny



Rys. 5.15. Wyjaśnienie typu „jak?"

Źródło dla modułu rozwiązującego problemy dla skali mini.
Część 1 plik siec.zw

```
knowledge base siec
// Demonstracyjna baza wiedzy nt. systemów teleinformatycznych
// Autor : JK
// © 2003 AON, Warszawa
 sources
  lan:
    type kb
    file "lan.zw";
  man:
    type kb
    file "man.zw";
  wan:
    type kb
    file "wan.zw";
 end;
 facets
  ask yes;
  single yes;
  poczta :
  query "Czy możesz wysyłać i odbierać pocztę ?:"
  val oneof { "tak", "nie" };
  serwer_www :
  query "Czy pracuje serwer www.aon.edu.pl ?:"
  val oneof { "tak", "nie" };
  inne_www :
  query "Czy inne serwery WWW pracują ?:"
  val oneof { "tak", "nie", "tak, ale bardzo wolno" };
  otoczenie :
  query "Czy w otoczeniu sieciowym widać inne komputery ?:"
  val oneof { "tak", "nie", "nie wiem" };
  statystyki :
  query "Jakie obciążenie łącza Internetowego pokazują statystyki mtrg ?:"
  val oneof { "do 3%", "od 3% do 70%", "od 70% do 95%", "ponad 95%", "nie są dostępne" };
  listy_vir:
  query "Czy serwer poczty informuje nas  o tym, że wysyłamy zawirusowane listy ?:"
  val oneof { "tak", "nie", "nie wiem" };
  strona2:
  query "Czy strona WWW została w widoczny sposób zniekształcona ?:"
  val oneof { "tak", "nie", "nie wiem" };
  snmp :
  query "Czy system zarządzania siecią działa poprawnie ?:"
  val oneof { "tak", "nie", "nie wiem" };
  cisco_ids :
  query "Czy system IDS alarmuje ?:"
  val oneof { "tak", "nie", "nie wiem" };
 //LAN
  stan_lan :
  val someof { "system sprawny", "system przeciążony - sprawny", "system przeciążony - atak
informatyczny", "system całkowicie niesprawny", "poważne zagrożenie przejęcia nieuprawnionej
kontroli nad systemem", "niesprawne łącza z Internetem", "sprawne łącza kampusowe", "niesprawne
łącza kampusowe", "problem ze switchem budynkowym", "niesprawny serwer proxy", "atak
informatyczny", "sprawne łącza kampusowe - niesprawne łącza z Internetem"};
// Warszawa
  stan_man :
```

```
  single no
  val oneof { "system sprawny" };
// RP
  stan_wan :
  single no
  val oneof {"system sprawny", "włamanie nr 1","włamanie nr 2"};
 end;
 control
  run;
  createAppWindow;
  vignette( "Sieć", "Próbna baza wiedzy\nsystemy teleinformatyczne w skalach:\n mini, mezo,
makro\nAutor: AON", "listopad 2003, AON, Warszawa");
  setAppWinTitle("Baza wiedzy Sieć");
  menu "&Wybór skali"
  1. "Skala mini - Sieć Kampusowa"
  2. "Skala mezo - sieć Miejska"
  3. "Skala makro - Sieć RP"
  4. "&Nowa konsultacja"
  5. "&Wyjście"
  case 1;
     solve( lan, "stan_lan=Stan_lan" );
  case 2;
     solve( man, "stan_man=Stan_man" );
  case 3;
     solve( wan, "stan_wan=Stan_wan" );
  case 4;
     delNewFacts;
  case 5;
     exit;
  end;
 end;
end;
```

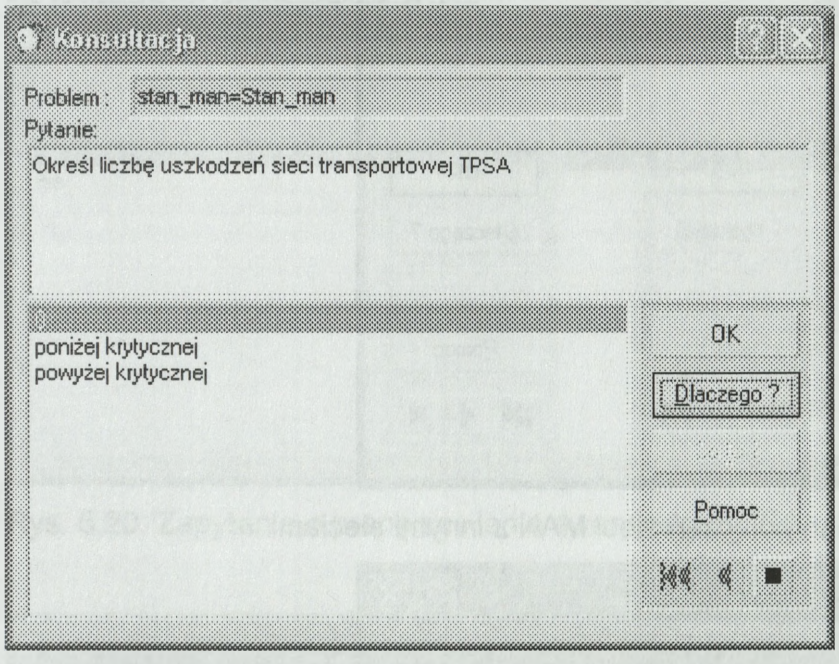
Źródła część 2: plik lan.zw

```
knowledge source nos
rules
1001: stan_lan = "system sprawny" if
 poczta="tak",
 serwer_www="tak",
 inne_www="tak",
 otoczenie="tak",
 strona2="nie",
 listy_vir="nie";
1002: stan_lan="system przeciążony - sprawny" if
 statystyki="ponad 95%",
 otoczenie="nie",
 listy_vir="nie";
1003:stan_lan="system przeciążony - atak informatyczny" if
 statystyki="ponad 95%",
 cisco_ids="tak",
 listy_vir="tak";
1004:stan_lan="system całkowicie niesprawny" if
 poczta="nie",
 serwer_www="nie",
 inne_www="nie",
 otoczenie="nie";
1005:stan_lan="poważne zagrożenie przejęcia nieuprawnionej kontroli nad systemem" if
 strona2="tak";
1006:stan_lan="niesprawne łącza z Internetem" if
```

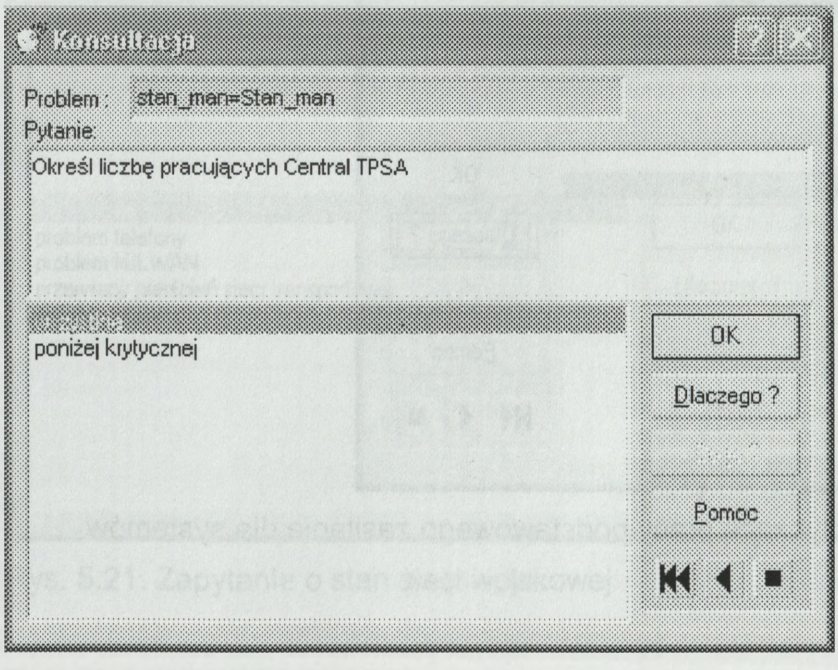
```
 inne_www="nie",
 statystyki="do 3%";
1007:stan_lan="sprawne łącza kampusowe" if
 otoczenie="tak",
 poczta="tak",
 serwer_www="tak",
 inne_www="nie";
1008:stan_lan="niesprawne łącza kampusowe" if
 otoczenie="tak",
 poczta="nie",
 serwer_www="nie",
 statystyki="nie są dostępne",
 inne_www="nie";
1009:stan_lan="problem ze switchem budynkowym" if
 otoczenie="nie",
 poczta="nie",
 serwer_www="nie",
 statystyki="nie są dostępne",
 cisco_ids="nie",
 inne_www="nie";
1010:stan_lan="niesprawny serwer proxy" if
 otoczenie="tak",
 poczta="tak",
 serwer_www="nie",
 inne_www="nie";
1011:stan_lan="atak informatyczny" if
 cisco_ids="tak",
 statystyki="nie są dostępne",
 listy_vir="tak";
1012:stan_lan="sprawne łącza kampusowe - niesprawne łącza z Internetem" if
 inne_www="nie",
 statystyki="do 3%",
 otoczenie="tak",
 poczta="tak",
 serwer_www="tak",
 inne_www="nie";
end;
end;
```

Przykładowa aplikacja dla skali mezo.

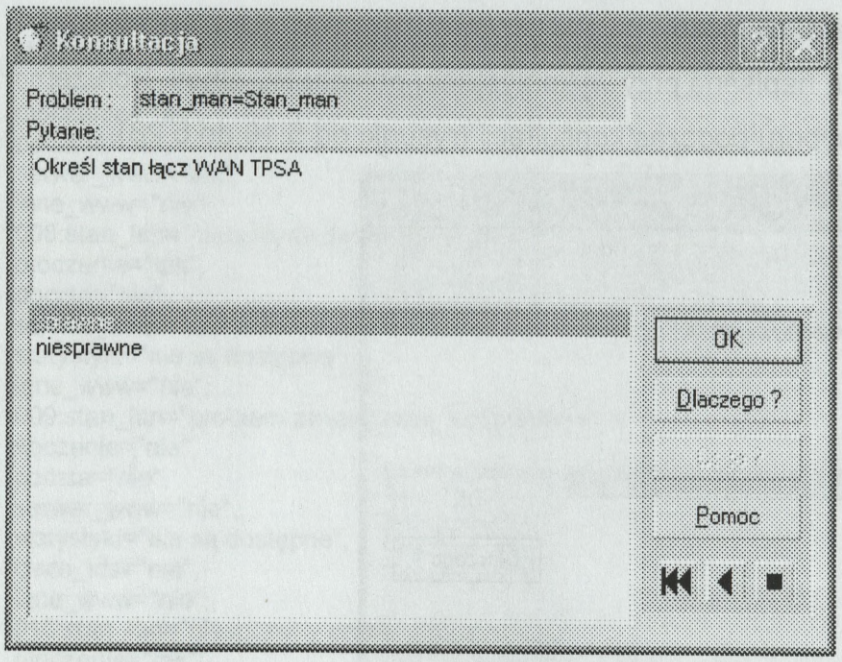
Aplikacja ta została napisana przy wykorzystaniu pakietu PC-Shell. Poniżej przedstawiono okienka wyboru podczas konsultacji (rysunki od 5.16 do 5.28).



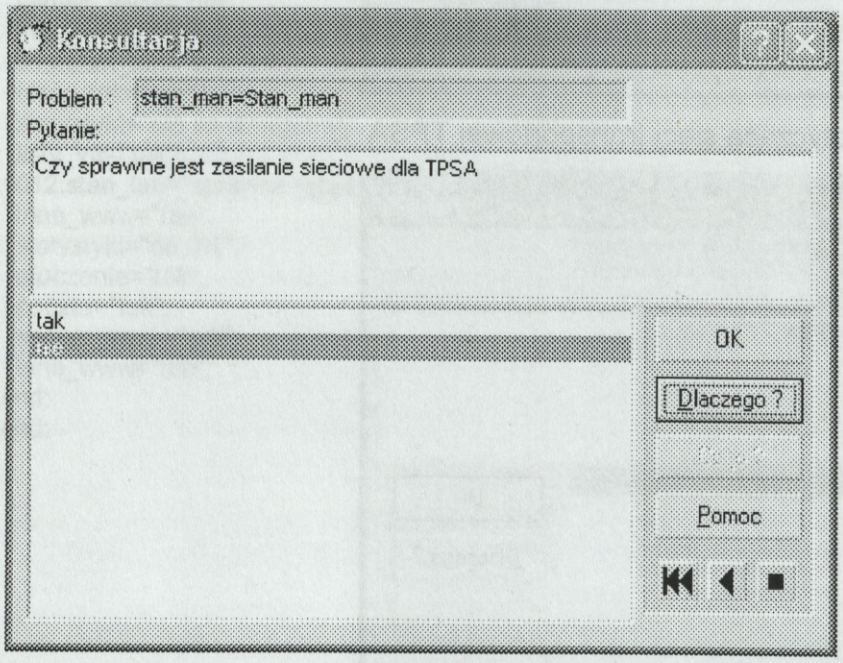
Rys. 5.16. Konsultacja dotycząca sieci transportowej TPSA



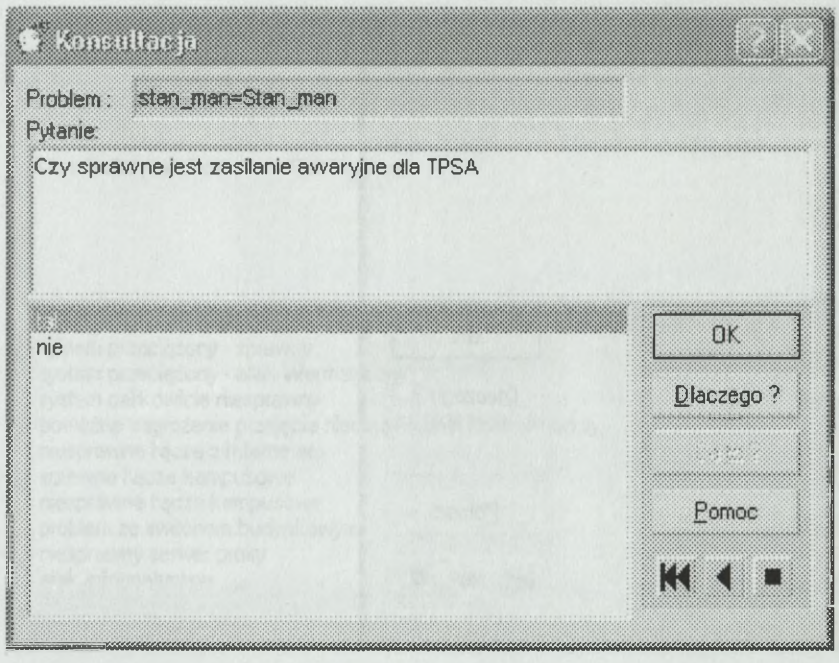
Rys. 5.17. Zapytanie o kondycję systemów telekomunikacyjnych TPSA



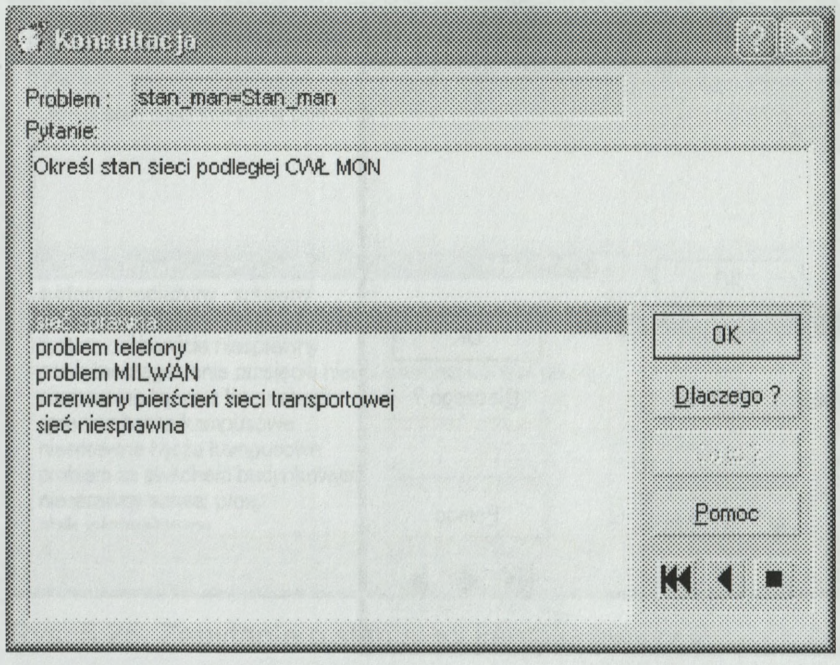
Rys. 5.18. Zapytanie o stan połączeń sieci MAN z innymi sieciami



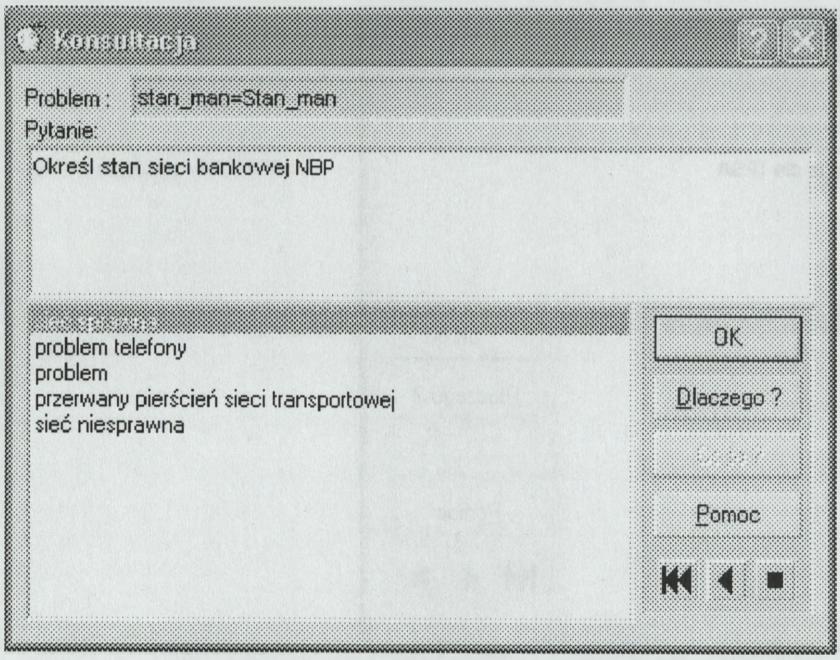
Rys. 5.19. Konsultacja dotycząca pracy podstawowego zasilania dla systemów TPSA



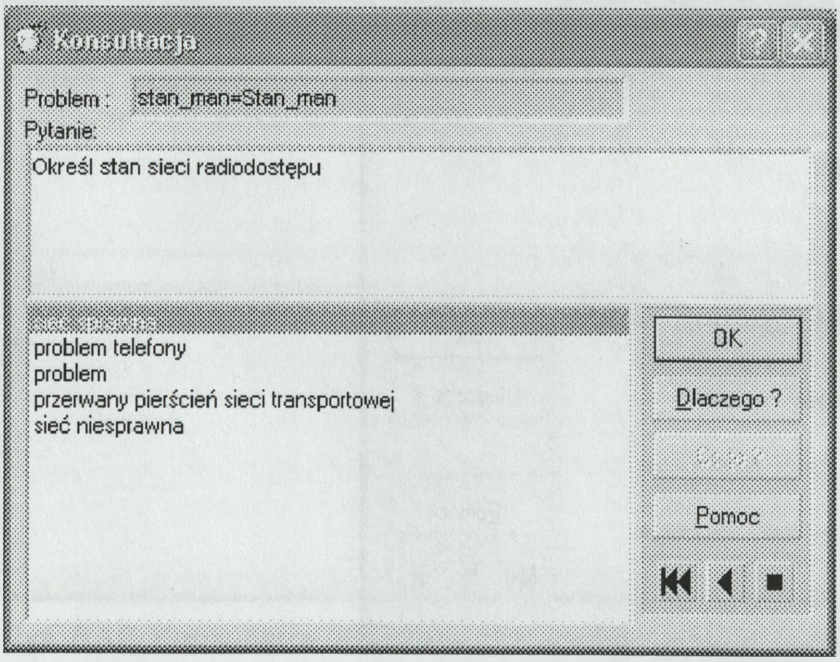
Rys. 5.20. Zapytanie o pracę zasilania awaryjnego TPSA



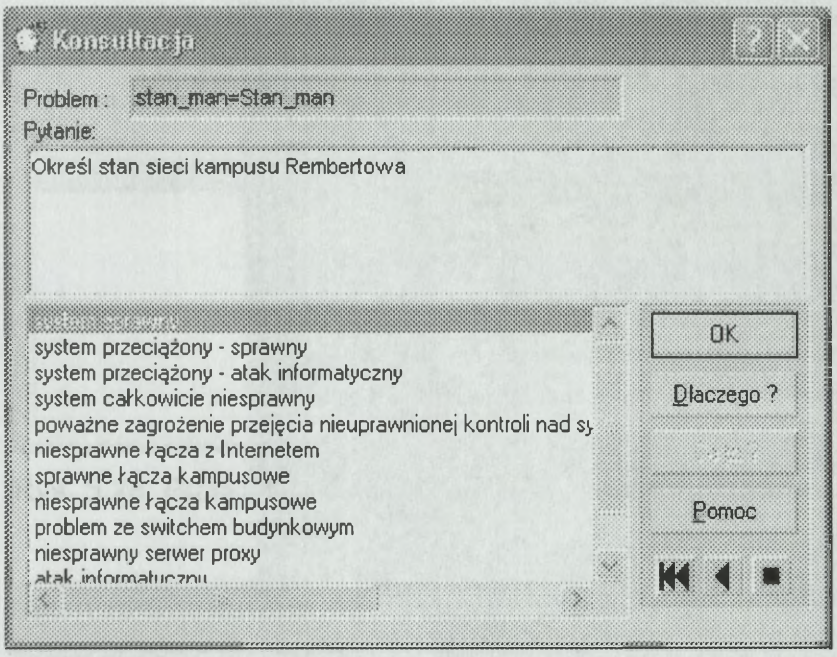
Rys. 5.21. Zapytanie o stan sieci wojskowej



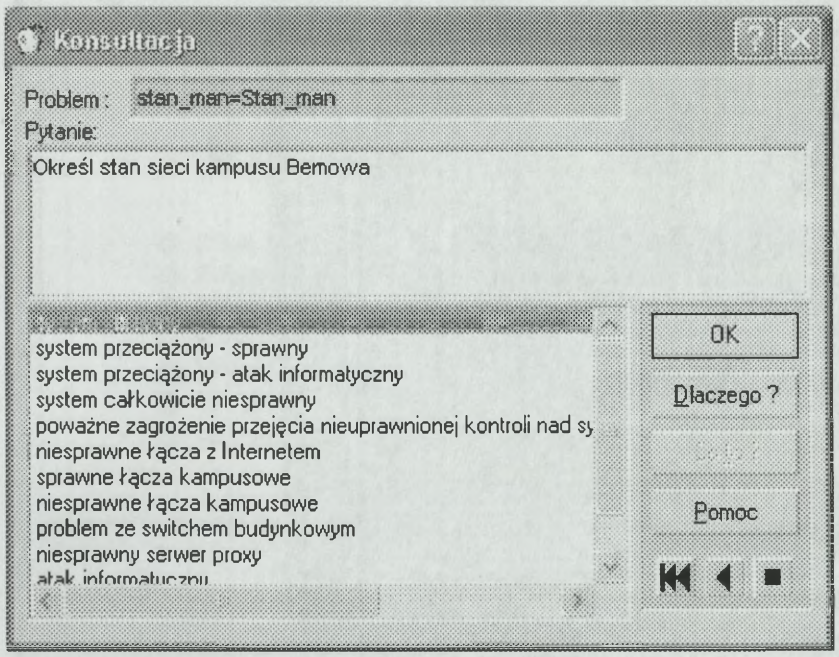
Rys. 5.22. Zapytanie o stan sieci bankowej (TELBANK)



Rys. 5.23. Konsultacja określająca stan sieci radiodostępu

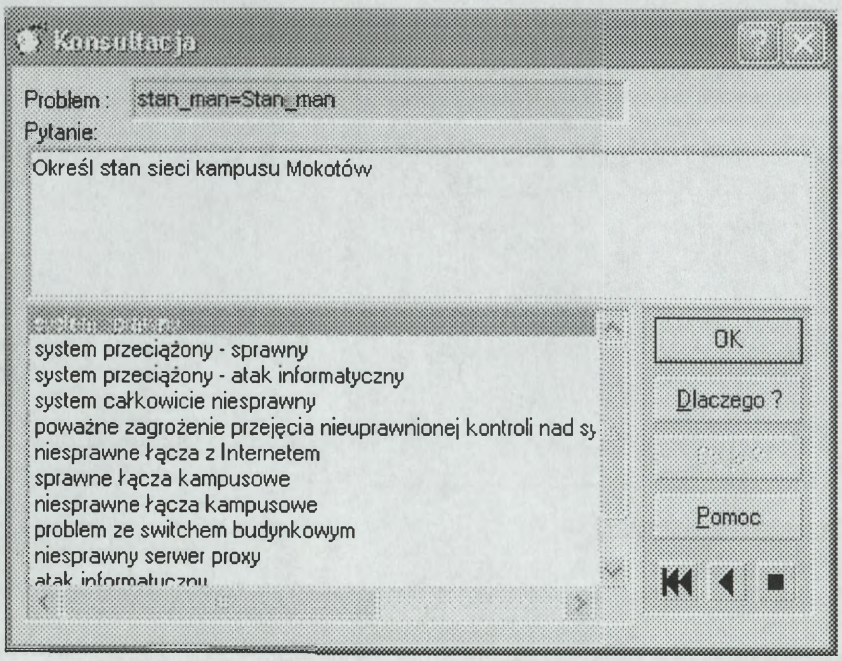


Rys. 5.24. Zapytanie o stan sieci kampusowej nr 1

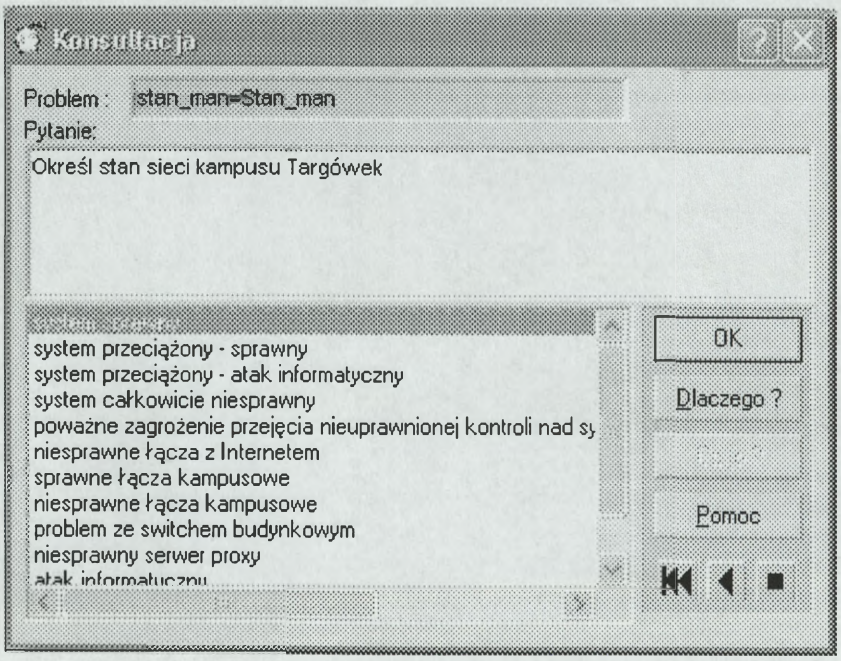


Rys. 5.25. Zapytanie o stan sieci kampusowej nr 2

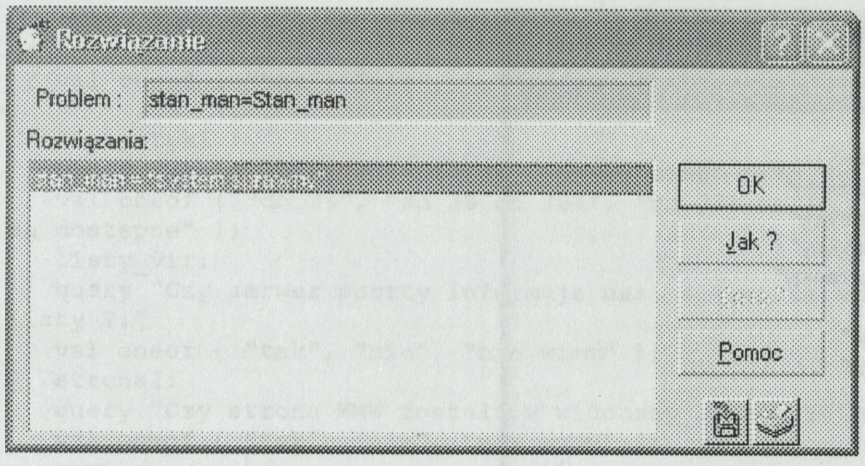
Informacje o stan sieci kampusowych (rysunki od 5.24 do 5.27) mogą pochodzić zarówno z bezpośrednich konsultacji jak i jako rozwiązania kilku konsultacji dla skali mikro.



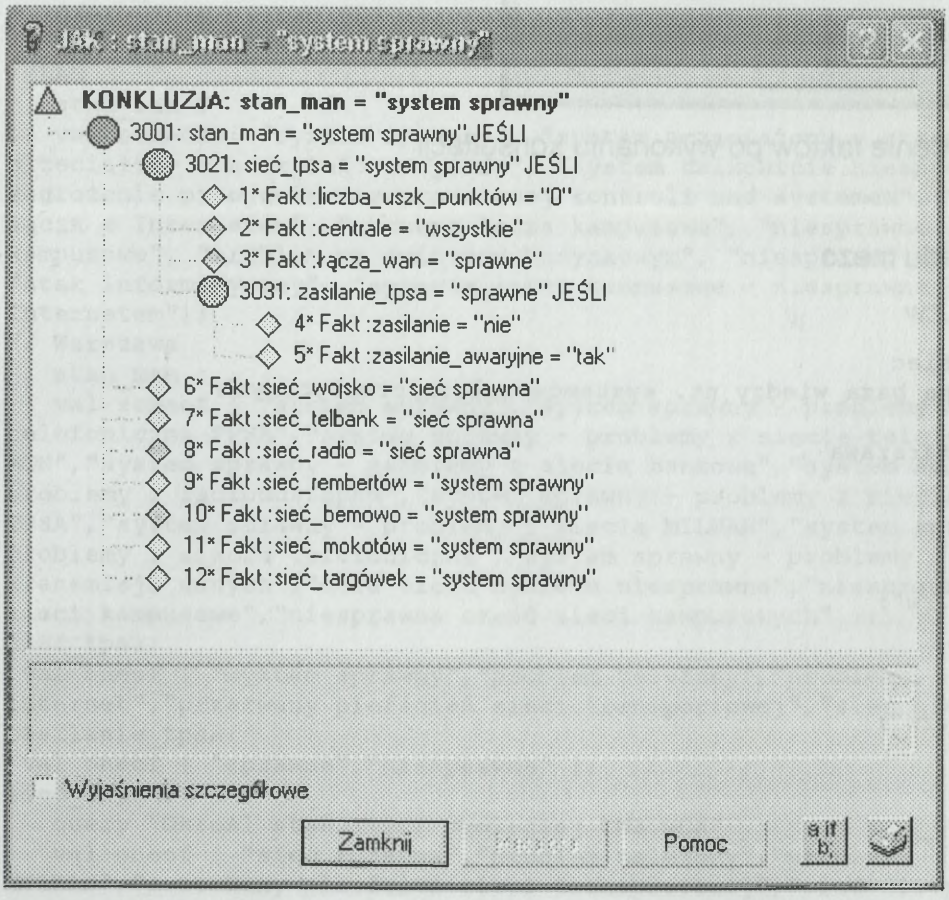
Rys. 5.26. Zapytanie o stan sieci kampusowej nr 3



Rys.5.27. Zapytanie o stan sieci kampusowej nr 4

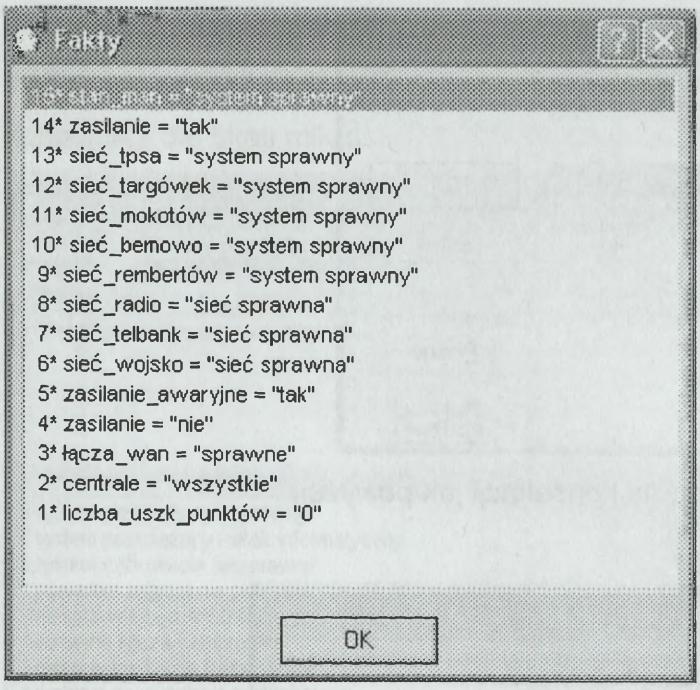


Rys. 5.28. Przykładowe rozwiązanie (dla konsultacji jak powyżej)



Rys.5.29. Wyjaśnienie „jak" rozwiązania opisanego powyżej

Po zakończeniu procesu wnioskowania do dyspozycji użytkownika jest też cały system wyjaśnień. Przykładowe wyjaśnienia typu „jak" w formie graficznej przedstawia rys.5.29.



Rys. 5.30. Zestawienie faktów po wykonaniu konsultacji

Źródła dla przykładu mezo

Źródło1. plik siec.zw

```
knowledge base siec
// Demonstracyjna baza wiedzy nt. systemów teleinformatycznych
// Autor : JK
// © 2003 AON, Warszawa
 sources

   lan:
      type kb
      file "lan.zw";

   man:
      type kb
      file "man.zw";

   wan:
      type kb
      file "wan.zw";
 end;
 facets
   ask yes;
   poczta :
   query "Czy możesz wysyłać i odbierać pocztę ?:"
   val oneof { "tak", "nie" };
   serwer_www :
   query "Czy pracuje serwer www.aon.edu.pl ?:"
   val oneof { "tak", "nie" };
   inne_www :
   query "Czy inne serwery WWW pracują ?:"
   val oneof { "tak", "nie", "tak, ale bardzo wolno" };

   otoczenie :
```

```
   query "Czy w otoczeniu sieciowym widać inne komputery ?:"

   val oneof { "tak", "nie", "nie wiem" };

   statystyki :
   query "Jakie obciążenie łącza Internetowego pokazują statystyki mtrg ?:"
   val oneof { "do 3%", "od 3% do 70%", "od 70% do 95%", "ponad 95%", "nie
są dostępne" };
   listy_vir:
   query "Czy serwer poczty informuje nas  o tym, że wysyłamy zawirusowane
listy ?:"
   val oneof { "tak", "nie", "nie wiem" };
   strona2:
   query "Czy strona WWW została w widoczny sposób zniekształcona ?:"
   val oneof { "tak", "nie", "nie wiem" };
   snmp :
   query "Czy system zarządzania siecią działa poprawnie ?:"
   val oneof { "tak", "nie", "nie wiem" };
   cisco_ids :
   query "Czy system IDS alarmuje ?:"
   val oneof { "tak", "nie", "nie wiem" };
//LAN
   stan_lan :
   val someof { "system sprawny", "system przeciążony - sprawny", "system
przeciążony - atak informatyczny", "system całkowicie niesprawny", "poważne
zagrożenie przejęcia nieuprawnionej kontroli nad systemem", "niesprawne
łącza z Internetem", "sprawne łącza kampusowe", "niesprawne łącza
kampusowe", "problem ze switchem budynkowym", "niesprawny serwer proxy",
"atak informatyczny", "sprawne łącza kampusowe - niesprawne łącza z
Internetem"};
// Warszawa
   stan_man :
   val someof { "system sprawny","system sprawny - problemy z siecią
telefoniczną TPSA","system sprawny - problemy z siecią telefoniczną CWŁ
MON","system sprawny - problemy z siecią bankową","system sprawny -
problemy z radiodostępem","system sprawny - problemy z siecią internetową
TPSA","system sprawny - problemy z siecią MILWAN","system sprawny -
problemy z siecią telefoniczną","system sprawny - problemy z sieciami
transmisji danych","duża część systemu niesprawna","niesprawne wszystkie
sieci kampusowe","niesprawna część sieci kampusowych" };
sieć_tpsa:
 val oneof { "system sprawny","problem telefony","problem
internet","przerwany pierścień sieci transportowej","sieć niesprawna" };
zasilanie_tpsa:
 val oneof { "sprawne","niesprawne" };
sieć_wojsko:
   query "Określ stan sieci podległej CWŁ MON"
   val oneof { "sieć sprawna","problem telefony","problem
MILWAN","przerwany pierścień sieci transportowej","sieć niesprawna" };
sieć_telbank:
   query "Określ stan sieci bankowej NBP"
   val oneof { "sieć sprawna","problem telefony","problem","przerwany
pierścień sieci transportowej","sieć niesprawna" };
sieć_radio:
   query "Określ stan sieci radiodostępu"
   val oneof { "sieć sprawna","problem telefony","problem","przerwany
pierścień sieci transportowej","sieć niesprawna" };
   sieć_rembertów:   query "Określ stan sieci kampusu Rembertowa"
   val someof { "system sprawny", "system przeciążony - sprawny", "system
przeciążony - atak informatyczny", "system całkowicie niesprawny", "poważne
zagrożenie przejęcia nieuprawnionej kontroli nad systemem", "niesprawne
```

```
łącza z Internetem", "sprawne łącza kampusowe", "niesprawne łącza
kampusowe", "problem ze switchem budynkowym", "niesprawny serwer proxy",
"atak informatyczny", "sprawne łącza kampusowe - niesprawne łącza z
Internetem"};
   sieć_bemowo:
   query "Określ stan sieci kampusu Bemowa"
   val someof { "system sprawny", "system przeciążony - sprawny", "system
przeciążony - atak informatyczny", "system całkowicie niesprawny", "poważne
zagrożenie przejęcia nieuprawnionej kontroli nad systemem", "niesprawne
łącza z Internetem", "sprawne łącza kampusowe", "niesprawne łącza
kampusowe", "problem ze switchem budynkowym", "niesprawny serwer proxy",
"atak informatyczny", "sprawne łącza kampusowe - niesprawne łącza z
Internetem"};
   sieć_mokotów:
   query "Określ stan sieci kampusu Mokotów"
   val someof { "system sprawny", "system przeciążony - sprawny", "system
przeciążony - atak informatyczny", "system całkowicie niesprawny", "poważne
zagrożenie przejęcia nieuprawnionej kontroli nad systemem", "niesprawne
łącza z Internetem", "sprawne łącza kampusowe", "niesprawne łącza
kampusowe", "problem ze switchem budynkowym", "niesprawny serwer proxy",
"atak informatyczny", "sprawne łącza kampusowe - niesprawne łącza z
Internetem"};
   sieć_targówek:
   query "Określ stan sieci kampusu Targówek"
   val someof { "system sprawny", "system przeciążony - sprawny", "system
przeciążony - atak informatyczny", "system całkowicie niesprawny", "poważne
zagrożenie przejęcia nieuprawnionej kontroli nad systemem", "niesprawne
łącza z Internetem", "sprawne łącza kampusowe", "niesprawne łącza
kampusowe", "problem ze switchem budynkowym", "niesprawny serwer proxy",
"atak informatyczny", "sprawne łącza kampusowe - niesprawne łącza z
Internetem"};
   liczba_uszk_punktów :
   query "Określ liczbę uszkodzeń sieci transportowej TPSA"
   val oneof { "0","poniżej krytycznej","powyżej krytycznej" };
   centrale:
   query "Określ liczbę pracujących Central TPSA"
   val oneof { "wszystkie","poniżej krytycznej" };
   łącza_wan:
   query "Określ stan łącz WAN TPSA"
   val oneof { "sprawne","niesprawne" };
   zasilanie:
   query "Czy sprawne jest zasilanie sieciowe dla TPSA"
   val oneof { "tak","nie" };
   zasilanie_awaryjne:
   query "Czy sprawne jest zasilanie awaryjne dla TPSA"
   val oneof { "tak","nie" };
// RP
   stan_wan :
   single no
   val oneof {"system sprawny", "włamanie nr 1","włamanie nr 2"};
 end;
 control
   run;
   createAppWindow;
   vignette( "Sieć", "Próbna baza wiedzy\nsystemy teleinformatyczne w
skalach:\n mini, mezo, makro\nAutor: AON", "listopad 2003, AON, Warszawa");
   setAppWinTitle("Baza wiedzy Sieć");

   menu "&Wybór skali"
   1.  "Skala mini - Sieć Kampusowa"
```

```
   2.  "Skala mezo - sieć Miejska"

   3.  "Skala makro - Sieć RP"
   4.  "&Nowa konsultacja"
   5.  "&Wyjście"

   case 1;
       solve( lan, "stan_lan=Stan_lan" );
   case 2;
       solve( man, "stan_man=Stan_man" );
   case 3;
       solve( wan, "stan_wan=Stan_wan" );
   case 4;
       delNewFacts;
   case 5;
       exit;
   end;
  end;

 end;
```

Źródło 2. Plik man.zw

```
knowledge source man
rules
3001:stan_man = "system sprawny" if
   sieć_tpsa="system sprawny",
   sieć_wojsko="sieć sprawna",
   sieć_telbank="sieć sprawna",
   sieć_radio="sieć sprawna",
   sieć_rembertów="system sprawny",
   sieć_bemowo="system sprawny",
   sieć_mokotów="system sprawny",
   sieć_targówek="system sprawny";

3002:stan_man = "system sprawny - problemy z siecią telefoniczną TPSA" if
   sieć_tpsa="problem telefony",
   sieć_wojsko="sieć sprawna",
   sieć_telbank="sieć sprawna",
   sieć_radio="sieć sprawna";

3003:stan_man = "system sprawny - problemy z siecią telefoniczną CWŁ MON"
if
   sieć_tpsa="system sprawny",
   sieć_wojsko="problem telefony",
   sieć_telbank="sieć sprawna",
   sieć_radio="sieć sprawna";
3004:stan_man = "system sprawny - problemy z siecią bankową" if
   sieć_tpsa="system sprawny",
   sieć_wojsko="sieć sprawna",
   sieć_telbank="problem",
   sieć_radio="sieć sprawna";

3005:stan_man = "system sprawny - problemy z radiodostępem" if
   sieć_tpsa="system sprawny",
   sieć_wojsko="sieć sprawna",
   sieć_telbank="sieć sprawna",
   sieć_radio="problem";

3006:stan_man = "system sprawny - problemy z siecią internetową TPSA" if
   sieć_tpsa="problem internet",
```

```
   sieć_wojsko="sieć sprawna",
   sieć_telbank="sieć sprawna",
   sieć_radio="sieć sprawna";

3007:stan_man = "system sprawny - problemy z siecią MILWAN" if
   sieć_tpsa="system sprawny",
   sieć_wojsko="problem MILWAN",
   sieć_telbank="sieć sprawna",
   sieć_radio="sieć sprawna";

3008:stan_man = "system sprawny - problemy z siecią telefoniczną" if
   sieć_tpsa="problem telefony",
   sieć_wojsko="problem telefony",
   sieć_telbank="sieć sprawna",
   sieć_radio="sieć sprawna";

3009:stan_man = "system sprawny - problemy z sieciami transmisji danych" if
   sieć_tpsa="problem internet",
   sieć_wojsko="problem MILWAN",
   sieć_telbank="sieć sprawna",
   sieć_radio="sieć sprawna";

3010:stan_man = "duża część systemu niesprawna" if
   sieć_tpsa="problem internet",
   sieć_wojsko="problem MILWAN",
   sieć_telbank="sieć niesprawna",
   sieć_radio="sieć sprawna";


3011:stan_man = "duża część systemu niesprawna" if
   sieć_tpsa="problem telefony",
   sieć_wojsko="problem telefony",
   sieć_telbank="sieć sprawna",
   sieć_radio="sieć niesprawna";

3012:stan_man = "niesprawne wszystkie sieci kampusowe" if
   sieć_rembertów="system całkowicie niesprawny",
   sieć_bemowo="system całkowicie niesprawny",
   sieć_mokotów="system całkowicie niesprawny",
   sieć_targówek="system całkowicie niesprawny";
3013:stan_man = "niesprawna część sieci kampusowych" if
   sieć_rembertów="system całkowicie niesprawny",
   sieć_bemowo="system całkowicie niesprawny",
   sieć_mokotów="system całkowicie niesprawny",
   sieć_targówek="system sprawny";
3014:stan_man = "niesprawna część sieci kampusowych" if
   sieć_rembertów="system całkowicie niesprawny",
   sieć_bemowo="system całkowicie niesprawny",
   sieć_mokotów="system sprawny",
   sieć_targówek="system całkowicie niesprawny";
3015:stan_man = "niesprawna część sieci kampusowych" if
   sieć_rembertów="system całkowicie niesprawny",
   sieć_bemowo="system sprawny",
   sieć_mokotów="system całkowicie niesprawny",
   sieć_targówek="system całkowicie niesprawny";
3016:stan_man = "niesprawna część sieci kampusowych" if
   sieć_rembertów="system sprawny",
   sieć_bemowo="system całkowicie niesprawny",
   sieć_mokotów="system całkowicie niesprawny",
   sieć_targówek="system całkowicie niesprawny";
3017:stan_man = "niesprawna część sieci kampusowych" if
```

```
   sieć_rembertów="system sprawny",
   sieć_bemowo="system sprawny",
   sieć_mokotów="system całkowicie niesprawny",
   sieć_targówek="system całkowicie niesprawny";
3018:stan_man = "niesprawna część sieci kampusowych" if
   sieć_rembertów="system sprawny",
   sieć_bemowo="system całkowicie niesprawny",
   sieć_mokotów="system sprawny",
   sieć_targówek="system całkowicie niesprawny";
3019:stan_man = "niesprawna część sieci kampusowych" if
   sieć_rembertów="system sprawny",
   sieć_bemowo="system całkowicie niesprawny",
   sieć_mokotów="system całkowicie niesprawny",
   sieć_targówek="system sprawny";
3020:stan_man = "niesprawna część sieci kampusowych" if
   sieć_rembertów="system całkowicie niesprawny",
   sieć_bemowo="system całkowicie niesprawny",
   sieć_mokotów="system sprawny",
   sieć_targówek="system sprawny";
3021:sieć_tpsa = "system sprawny" if
   liczba_uszk_punktów="0",
   centrale="wszystkie",
   łącza_wan="sprawne",
   zasilanie_tpsa="sprawne";
3022:sieć_tpsa = "sieć niesprawna" if
   liczba_uszk_punktów="0",
   centrale="poniżej krytycznej",
   łącza_wan="sprawne",
   zasilanie_tpsa="sprawne";
3023:sieć_tpsa = "sieć niesprawna" if
   liczba_uszk_punktów="0",
   centrale="poniżej krytycznej",
   łącza_wan="sprawne",
   zasilanie_tpsa="niesprawne";
3024:sieć_tpsa = "sieć niesprawna" if
   liczba_uszk_punktów="powyżej krytycznej",
   centrale="poniżej krytycznej",
   łącza_wan="sprawne",
   zasilanie_tpsa="sprawne";
3030:zasilanie_tpsa="sprawne" if
   zasilanie="tak",
   zasilanie_awaryjne="tak";
3031:zasilanie_tpsa="sprawne" if
   zasilanie="nie",
   zasilanie_awaryjne="tak";

3032:zasilanie_tpsa="sprawne" if
   zasilanie="tak",
   zasilanie_awaryjne="nie";
3033:zasilanie_tpsa="niesprawne" if
   zasilanie="nie",
   zasilanie_awaryjne="nie";
end;
end;
```

Wnioski końcowe

Właściwie realizowana polityka bezpieczeństwa w strukturach organizacyjnych obejmuje również budowę odpowiedniego systemu ochronnego wykorzystywanych systemów teleinformatycznych. W celu bezpiecznego ich funkcjonowania niezbędne jest opracowanie i wdrożenie właściwego systemu procedur bezpieczeństwa. Jest to sformalizowany opis zasad zapewnienia bezpieczeństwa fizycznego, bezpieczeństwa dostępu, polityki szkolenia. Procedury obejmują również sytuacje nadzwyczajne takie jak: pożar, powódź itp.

Wielu operatorów ważnych systemów (w tym również należących do infrastruktury krytycznej) nie posiada wystarczających informacji o jej słabościach i zagrożeniach, i dlatego nie jest w stanie szybko reagować na nagłe zdarzenia. Ważna jest więc znajomość występowania wszystkich współzależności np. kto zasila w informacje, jakimi środkami i do kogo te informacje są przekazywane. Konieczne jest więc wprowadzenie środków, za pomocą których firmy będą mogły wspólnie sprawować nadzór techniczny, przekazywać informacje i radzić sobie ze zdarzeniami naruszającymi bezpieczeństwo. Ważne informacje, które trzeba by wymieniać w ramach takiej współpracy, mogą obejmować odnotowane już zdarzenia, nowe zagrożenia i poważne awarie systemu. Ważne jest, by w krytycznych sytuacjach strony mogły szybko ostrzegać innych uczestników i przekazywać im informacje.

Coraz większa złożoność systemów oraz wyzwania stawiane przez nowe scenariusze zagrożeń doprowadziły do powstania luki kompetencyjnej wśród pracowników, którą zamierza się usunąć poprzez podniesienie poziomu edukacji w całym kraju. Zakłada się, iż bezpieczeństwo kosztuje i musi kosztować, adekwatnie do wartości dóbr, które ma się chronić. Nie można przy tym ukrywać, że działania związane z zapewnieniem bezpieczeństwa stanowią dla pracowników dodatkowe obciążenie i powszechnie uważane są za uciążliwe w pracy. Nie powinny jednak być uciążliwe na tyle, by paraliżować działalność instytucji. Konieczne jest tu więc zachowanie pewnej równowagi. Stąd przewidywana i rosnąca rola systemów wspomagających.

Nie ulega wątpliwości, że zautomatyzowana analiza ryzyka przynosi szereg wartości rekompensujących wysokie koszty jej przeprowadzenia. Do szczególnych zalet zastosowania tych narzędzi należą:

- ustalone sposoby wprowadzania danych, a następnie łatwość dostępu i posługiwania się danymi zestawionymi w bazie niezbędnej do przeprowadzenia analizy;
- możliwość manipulowania danymi w celu zobrazowania wpływu i efektów różnych kombinacji zastosowanych środków zabezpieczających i symulacji strat;
- możliwość szybkiego wprowadzania zmian do rozpoznawanego środowiska (aktywa i zasoby) oraz rozpoznania ryzyka w organizacji.

Dzisiejsze trudności technologii systemów eksperckich wspomagania decyzji wynikają w głównej mierze z następujących względów:

- ograniczenia stosowanych dziś metod reprezentacji wiedzy;
- słabo rozwiniętej metodologii pozyskiwania wiedzy dla systemu eksperckiego;
- wąskiej kompetencji projektowanych systemów;
- praktycznego braku wnioskowania zdroworozsądkowego;
- niedostatków komunikacji użytkownik-system;
- braku ekspertów gotowych podzielić się swoją wiedza;
- braku procedur ujednolicających, unifikujących i klasyfikujących poszczególne słownictwo używane przez różnych ekspertów tych samych specjalizacji.

Zastosowanie zaproponowanego systemu ekspertowego możliwego do zastosowania na różnych szczeblach z możliwością adaptacji i rozwoju (w przyszłości być może w oparciu o układy sztucznej inteligencji) powinno w wydatnym stopniu podnieść poziom bezpieczeństwa informacyjnego. Godnym podkreślenia jest też fakt, że zaproponowany system umożliwia wspomaganie w usuwaniu skutków oraz określanie przedsięwzięć prewencyjnych. Umożliwia jednak przede wszystkim szeroki dostęp do wiedzy eksperckiej oraz zastosowanie nowoczesnych metod zapewnienia bezpieczeństwa bez zbędnych nakładów finansowych, nawet w przypadku braku możliwości skorzystania z bezpośredniej pomocy profesjonalistów.

Literatura

1. Adamski A.: Prawo karne komputerowe, Wydawnictwo C.H.BECH, Warszawa 2000.
2. Adamski A. red.: Przestępczość komputerowa, TNOiK. Toruń 1994, str.267-278.
3. Andrukiewicz E.: Zarządzanie zabezpieczeniem systemu informacyjnego, Przegląd telekomunikacyjny, nr 10/98.
4. Atkins D. i in.: Bezpieczeństwo Internetu, LT&P 1997.
5. Barczak A. Sydoruk T.: Bezpieczeństwo systemów informatycznych. Wydawnictwo Akademii Podlaskiej. Siedlce 2002.
6. Białas A.: Bezpieczeństwo sieci komputerowych, Bielsko-Biała, Wyd. WSI i Zarządzania 2001.
7. Bógdał-Brzezińska A., Gawrycki M. F.: Cyberterroryzm i problemy bezpieczeństwa we współczesnym świecie 2003.
8. CCPC AHG: Ochrona infrastruktury telekomunikacyjnej o newralgicznym znaczeniu , kwiecień 2002.
9. Chudy M., Nowicki T., Najgebauer A., Mielczarek K.: Projektowanie koncepcyjne systemu informatycznego wspomagania analiz decyzyjnych w sytuacjach konfliktowych dla potrzeb kierowania obronnością państwa. Praca TEORIA II. Teoria konfliktów zbrojnych, AON WAT Warszawa 1996.
10. Chwiałkowska Ewa, Sztuczna inteligencja w systemach eksperckich, Warszawa1991.
11. Computer-Related Crime, Recommendation No. R(89)9 on computer-related crime and final report of the European Committee on Crime Problems, Counsil of Europe, Strasbourg 1990.
12. Gamdżyk P., Kosieliński S.: Kwestia wyobraźni, Nr 32 z 2002.
13. Garfinkel S. I Spafford G.: Bezpieczeństwo w Unixie i Internecie, Wydawnictwo RM, Warszawa 1997.
14. Gniłka M.J.: Model i kryteria oceny bezpieczeństwa systemu informatycznego. Materiały VI Międzynarodowej Wojskowej Konferencji Telekomunikacji i Informatyki. WIŁ, Jabłonna 1997.

15. Goban-Klas T.: Sienkiewicz P.: Społeczeństwo informacyjne: szanse, problemy, zagrożenia. Wydawnictwo Fundacji Postępu Komunikacji. Kraków 1999.
16. Gogela R., Novak L., Sefcik A.: Critical Infrastructure Modelling, Security and Protection of Information 2003.
17. Hacking J.: Ochrona krytycznej infrastruktury państwa, NASK, kwiecień 2002.
18. Informacja na wspólne posiedzenie BBN przy Prezydencie RP oraz Kolegium ds. Służb Specjalnych przy Radzie Ministrów w sprawie identyfikacji zagrożeń i stanu ochrony infrastruktury krytycznej, wydawnictwo wewnętrzne 2002.
19. Internet. Agresja i Ochrona, Wydawnictwo Robomatic 1998.
20. Kardas J. Loranty K.: Wybrane problemy bezpieczeństwa państwa w opiniach pracowników administracji publicznej. AON. Warszawa 2001.
21. Kasabov N.K.,Foundations of neural networks, fuzzy systems and knowledge engineering, A Bradford Book The MIT Press, London 1996.
22. Klander L.: Hacker Proof, czyli jak się bronić przed intruzami, Wydawnictwo MIKOM, Warszawa 1998.
23. Krauze M.: Kierowanie obronnością państwa w aspekcie wykorzystania technicznych środków najnowszej generacji, materiały z seminarium naukowego odbytego w Centrum Konferencyjnym MON [Ministerstwa Obrony Narodowej] w dniu 5 kwietnia 2001 roku / kier. nauk. Michał Krauze AON, Warszawa 2001.
24. Krytyczna infrastruktura teleinformatyczna w Polsce – stan aktualny i zagrożenia cybernetyczne, Departament Bezpieczeństwa TI ABW Warszawa, luty 2002.
25. Liderman K.: Bezpieczeństwo informatyczne. Wydawnictwo Naukowe PWN Warszawa 2001.
26. Mitnik K.: Sztuka podstępu, Wydawnictwo HELION 2003.
27. Molski M.: Podstawy bezpieczeństwa systemów informatycznych. Wydawnictwo MSG Media, Bydgoszcz 1998.
28. Mulawka J.: Systemy ekspertowe, Wydawnictwo Naukowo Techniczne, Warszawa 1996.
29. Najgebauer A.: Informatyczne systemy wspomagania decyzji w sytuacjach

konfliktowych. Modele, metody i środowiska symulacji interaktywnej, dodatek do biuletynu WAT, Warszawa 1999.

30. Piedziuk E.: Systemy alarmowe sygnalizacji zagrożeń, AON, Warszawa 1998.
31. Piedziuk E.: Telekomunikacja na potrzeby obronności państwa, AON, Warszawa 1997.
32. Pipkin D. L.: Bezpieczeństwo informacji. Ochrona globalnego przedsiębiorstwa. Wydawnictwo Naukowo – Techniczne, Warszawa 2002.
33. Pohorecki G.: Zarządzanie bezpieczeństwem systemów teleinformatycznych według Polskiej Normy PN-I-13335, Security IT Magazine, nr 1/1999.
34. Rozporządzenie Ministra SWiA z dnia 3 czerwca 1998 r. w sprawie określenia podstawowych warunków technicznych i organizacyjnych, jakim powinny odpowiadać urządzenia i systemy informatyczne służące do przetwarzania danych osobowych, Dz. U. Nr 80, poz. 521.
35. Rozporządzenie Prezesa RM z dnia 25 lutego 1999 r. w sprawie podstawowych wymagań bezpieczeństwa systemów i sieci teleinformatycznych, Dz. U. Nr 18, poz.162.
36. Schneider M., Kandel A.: Fuzzy Expert System Tools, Wiley Publikations 1996.
37. Sienkiewicz P., Górny P.: Analiza systemowa sytuacji kryzysowych, Zeszyty Naukowe AON nr 4, Warszawa 2001.
38. Sienkiewicz P., Nowicki T., Najgebauer A.: Metodyka projektowania eksperckiego systemu informatycznego wspomagania analiz decyzyjnych w sytuacjach konfliktowych, Praca TEORIA II. Teoria konfliktów zbrojnych. AON, WAT Warszawa 1997.
39. Stokłosa J, Bilski T, Pankowski T.: Bezpieczeństwo danych w systemach informatycznych, Wydawnictwo Naukowe PWN, Warszawa – Poznań 2001.
40. Szyjewski Z.: Komputerowe wspomaganie realizacji systemów informatycznych, Szczecin 1994
41. Świątnicki Z.: Wojskowe systemy eksperckie, Bellona, Warszawa 1995;
42. Techniczne Aspekty Przestępczości Teleinformatycznej, materiały konferencyjne - V Edycja, Szczytno 2002.
43. Telekomunikacyjne Centrum Informacyjne, Internet – www. gov.pl.

44. Ustawa o ochronie danych osobowych z dnia 29 sierpnia 1997 roku (Dz. U. Nr 133, poz. 883 z późn. zm.).
45. Ustawa Prawo telekomunikacyjne z dn. 21 lipca 2000 roku (Dz. U. Nr 73, poz. 852).
46. Ustawa z dnia 22 stycznia 1999 r. o ochronie informacji niejawnych, Dz. U. Nr 11, poz. 95.
47. Ustawa z dnia 22 stycznia 1999r. o ochronie informacji niejawnych. Dziennik Ustaw nr 11, pozycja 95.
48. Ustawa z dnia 29 sierpnia 1997 r. o ochronie danych osobowych, Dz. U. 133, poz. 883.
49. Vijay Ahuja: Bezpieczeństwo w sieciach, Wydawnictwo MIKOM, Warszawa, wrzesień 1997.
50. Wiadomości TVP, Internet – www.tel.pl .
51. Wójcik J. W.: Przestępstwa komputerowe. Część I – Fenomen cywilizacji, CIM, Warszawa 1999.
52. Zaskórski P.: Koncepcja systemu reagowania kryzysowego MON, Wydawnictwo AON, Warszawa 2002.