

Zintegrowany pakiet sztucznej inteligencji
Sphinx 4.0

Robert Florczyk Stanisław Zych

27 lipca 2005

Spis treści

| | | |
|----------|--|-----------|
| 1 | Czym jest Sphinx? | 3 |
| 1.1 | Wprowadzenie | 3 |
| 1.2 | Moduły systemu <i>Sphinx 4.0</i> | 3 |
| 2 | Moduł CAKE 4.0 | 5 |
| 2.1 | Przeznaczenie modułu CAKE 4.0 | 5 |
| 2.2 | Praca z modułem <i>CAKE</i> | 7 |
| 2.2.1 | Podsystem zarządzania bazami wyjaśnień systemu <i>PC-Shell</i> | 11 |
| 3 | Moduł DemoViewer | 13 |
| 3.1 | Przeznaczenie systemu DemoViewer | 13 |
| 4 | Moduł DeTreeX 4.0 | 15 |
| 4.1 | Przeznaczenie modułu DeTreeX 4.0 | 15 |
| 4.2 | Praca z modułem DeTreeX | 16 |
| 4.2.1 | Współpraca systemu DeTreeX z bazami danych | 18 |
| 5 | Moduł Dialog Edytor | 20 |
| 5.1 | Przeznaczenie modułu Dialog Edytor | 20 |
| 6 | Moduł HybRex 4.0 | 22 |
| 6.1 | Przeznaczenie modułu HybRex 4.0 | 22 |
| 6.2 | Praca z programem | 25 |
| 7 | Moduł Neuronix 4.0 | 27 |
| 7.1 | Przeznaczenie modułu Neuronix 4.0 | 27 |
| 8 | Moduł PC-Shell 4.0 | 31 |
| 8.1 | Wprowadzenie do systemów ekspertowych | 31 |
| 8.1.1 | Szkielety systemów ekspertowych | 31 |
| 8.1.2 | Budowa systemu ekspertowego | 32 |
| 8.1.3 | Ogólna charakterystyka modułu PC-Shell | 33 |

| | | |
|----------|--|-----------|
| 8.1.4 | Struktura systemu PC-Shell | 34 |
| 8.1.5 | Architektura | 35 |
| 8.1.6 | Zastosowanie | 36 |
| 8.1.7 | Wnioskowanie | 37 |
| 8.1.8 | Parametryzacja baz wiedzy | 37 |
| 8.1.9 | Interfejs do baz danych | 38 |
| 9 | Moduł Predyktor 4.0 | 40 |
| 9.1 | Przeznaczenie modułu Predyktor 4.0 | 40 |
| 9.2 | Użytkownicy systemu | 40 |
| 9.3 | Zastosowania systemu Predyktor | 41 |

Rozdział 1

Czym jest Sphinx?

1.1 Wprowadzenie

Sztuczna inteligencja (*ang. Artificial Intelligence – AI*) to technologia i kierunek badań na styku informatyki, neurologii i psychologii. Jego zadaniem jest konstruowanie maszyn i oprogramowania zdolnego rozwiązywać problemy nie poddające się algorytmizacji w sposób efektywny, w oparciu o modelowanie wiedzy (inaczej: zajmuje się konstruowaniem maszyn, które robią to, co obecnie ludzie robią lepiej). Problemy takie bywają nazywane AI-trudnymi i zalicza się do nich między innymi analiza (i synteza) języka naturalnego, rozumowanie logiczne, dowodzenie twierdzeń, gry logiczne (szachy, warcaby, go) i manipulacja wiedzą - systemy doradcze, diagnostyczne.

Przykładem polskiego zintegrowanego pakietu sztucznej inteligencji jest system *Sphinx*, którego struktura zostanie w niniejszym opracowaniu przybliżona.

1.2 Moduły systemu *Sphinx 4.0*

1. CAKE 4.0
2. DemoViewer
3. DeTreex 4.0
4. Dialog Edytor
5. HybRex 4.0
6. Neurinix 4.0

7. PC-Shell 4.0

8. Predyktor 4.0

Poniższe opracowanie zawiera omówienie poszczególnych modułów; ich przeznaczenie a także pracę z nimi.

Rozdział 2

Moduł CAKE 4.0

2.1 Przeznaczenie modułu CAKE 4.0

Opis systemu *Sphinx* zaczniemy od modułu *CAKE* (ang. *Computer-Aided Knowledge Engineering*) jest przeznaczony do wspomagania procesu realizacji dziedzinowych aplikacji szkieletowego systemu eksperckiego. *PC-Shell*. Moduł ten częścią pakietu narzędziowego *Sphinx* firmy *AITECH*, po instalacji jest on obecny jako jeden z elementów utworzonej grupy.

Dzięki systemowi *CAKE* możemy zrealizować następujące operacje:

- zarządzanie projektem aplikacji systemu *PC-Shell*;
- wspomaganie procesu tworzenia, rozbudowy i pielęgnacji baz wiedzy;
- weryfikacja poprawności wprowadzonej 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;

System *CAKE* umożliwia realizację aplikacji systemu ekspertowego *PC-Shell* nawet bez dokładnej znajomości języka opisu wiedzy. Dzięki wygodnym narzędziom wspomagającym system eliminuje konieczność żmudnego wprowadzania kodu. Zapis baz wiedzy w postaci binarnej zapewnia z jednej strony ochronę zgromadzonej wiedzy przed niepowołanym dostępem,

z drugiej zaś poprawia efektywność wykonania aplikacji w środowisku *PC-Shell*. System ochrony w postaci ograniczeń dostępu i haseł uniemożliwia dostęp do aplikacji zarówno na etapie jej tworzenia jak i wykonywania.

System *CAKE* realizuje również funkcje systemu dbMaker, zatem system *CAKE* jest odpowiedzialny za tworzenie, utrzymywanie oraz aktualizację baz wyjaśnień typu: "co to jest?" a także *metafor*.

Dzięki modułowi *CAKE* możliwa jest realizacja następujących funkcji:

- tworzenie bazy wiedzy w oparciu o specjalizowany edytor baz wiedzy, dzięki czemu inżynier wiedzy nie musi znać dokładnie języka *Sphinx*;
- realizowanie na bieżąco kontroli poprawności wprowadzanych informacji;
- automatyczną generację kodu źródłowego bazy wiedzy;
- automatyczną generację bazy w tzw. *postaci binarnej*, która nie będzie podlegała procesowi translacji.
- ochronę wiedzy zapisanej w bazie binarnej w oparciu o system haseł i uprawnień;

Podstawowe bloki funkcjonalne systemu:

- moduł edycji bazy wiedzy
 - edytor bloku atrybutów;
 - edytor bloku faktów;
 - edytor bloku reguł;
 - edytor bloku sterującego;
- moduł uprawnień;
- moduł zarządzania bazą wiedzy;
- translator języka *Sphinx*;
- moduł raportów;

Moduł edycji bazy wiedzy realizuje funkcje warstwy dialogowej systemu. W interaktywny sposób użytkownik systemu może realizować wszystkie operacje na bazie wiedzy. Główny edytor składa się ze specjalizowanych edytorów operujących na poszczególnych blokach bazy wiedzy. Każdy edytor posługuje się odpowiednimi oknami dialogowymi, sterującymi przebiegiem wykonania poszczególnych operacji. Na etapie edycji bazy wiedzy przeprowadzone są w sposób automatyczny podstawowe czynności weryfikacyjne, dotyczące głównie kontroli lokalnej spójności wprowadzanych informacji.

Moduł zarządzania bazą wiedzy moduł ten odpowiedzialny jest za realizację wszystkich operacji związanych z bezpośrednimi manipulacjami na bazie wiedzy. Wszelkie operacje zmieniające stan bazy są zlecane temu modułowi przez odpowiednie moduły edycyjne. To tutaj następuje kontrola poprawności wprowadzanych czy zmienianych elementów - atrybutów, faktów reguł. Moduł zarządzania bazą wiedzy realizuje również wszystkie operacje dyskowe. Istotnym elementem tego modułu jest zarządca źródeł - odpowiedzialny za nadzorowanie realizacji aplikacji tablicowych, zawierających bazę wiedzy rozdzieloną na źródła.

Translator języka opisu bazy wiedzy jest odpowiednikiem translatora wbudowanego w system ekspertowy *PC-Shell*. Zadaniem translatora jest kontrola poprawności bloku sterowania - jedyne go bloku bazy wiedzy, który jest tworzony w sposób klasyczny, tzn. za pośrednictwem edytora tekstowego. Drugim ważnym elementem translatora jest kontrola poprawności baz zapisanych w postaci tekstowej, importowanych do systemu *CAKE* w celu wygenerowania wersji binarnej lub weryfikacji.

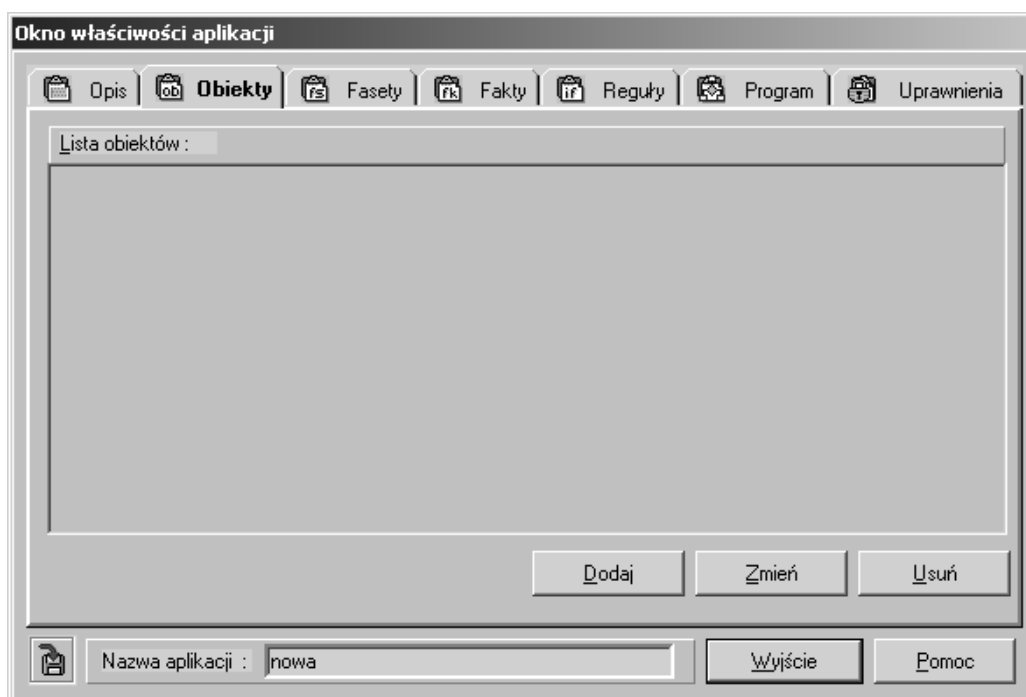
Moduł raportów - dla każdej bazy wiedzy opracowanej czy rozbudowanej w systemie *CAKE* można automatycznie utworzyć dokumentację zawierającą zarówno opisowe, ogólne informacje o bazie jak i informacje o poszczególnych blokach i ich właściwościach. Tak stworzona dokumentacja może być następnie wydrukowana w postaci odpowiedniego raportu.

2.2 Praca z modułem *CAKE*

Aby rozpocząć tworzenie aplikacji w module *CAKE* najpierw trzeba określić rodzaj aplikacji (aplikacja baz źródeł wiedzy lub aplikacja oparta na

źródłach wiedzy) lub typ bazy wyjaśnień (mamy tu do wyboru wyjaśnienia typu "co to jest ?" lub *metafory*). Opcje podzielone są na dwie oddzielne grupy (osobno dla aplikacji, osobno dla baz wyjaśnień). Dostęp do żądanych opcji możliwy jest po wskazaniu jednej z "zakładek", umieszczonych w górnej części okna.

Po zdefiniowaniu rodzaju aplikacji wyświetli się *Okno właściwości aplikacji*, które spełnia rolę okna głównego systemu *CAKE*. Poniżej przedstawiono wygląd okna głównego.



Rysunek 2.1: Okno właściwości aplikacji

Umożliwia swobodne poruszanie się pomiędzy poszczególnymi składnikami aplikacji.

Dostęp do żądanej informacji możliwy jest po wskazaniu jednej z zakładek, reprezentujących konkretne elementy struktury:

1. **Opis** - informacje ogólne na temat aplikacji takie jak:

- Nazwa aplikacji;
- Opis aplikacji;
- Data utworzenia;
- Data ostatniej modyfikacji;

- System uprawnień;
2. **Źródła** - wykaz źródeł wiedzy przypisanych do bazy wiedzy;

System pozwala na stworzenie czterech rodzajów źródeł wiedzy:

- **ekspercka baza wiedzy** - źródło zawierające fragment wiedzy eksperckiej;
- **baza metafor** - zbiór *metafor* przypisanych regułom;
- **baza wyjaśnień "co to jest?"**;
- **definicja sieci neuronowej**;

W oknie źródeł zestawione są 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. Możemy rozszerzyć listę źródeł wiedzy a także zmodyfikować wszystkie elementy wchodzące w skład wskazanego źródła, pamiętając, że w przypadku źródeł typu *definicja sieci neuronowej* bezpośrednia edycja za pośrednictwem systemu *CAKE* nie jest możliwa.

3. **Obiekty** - w oknie obiektów 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).
4. **Fasety** - w oknie faset wyszczególnione są nazwy *atrybutów* oraz aktualna postać *faset globalnych* bazy wiedzy.
5. **Fakty** - okno to 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. Wszystkie fakty zawarte w bazie wiedzy mają postać trójki OAW. Ogólną postać faktu można przedstawić następująco:

$$[not]atrybut[(obiekt)][= wartosc] \quad (2.1)$$

Identyfikator obiektu jest składnikiem opcjonalnym. Dla atrybutów logicznych nie określa się wartości.

6. **Reguły** - zestawione są tutaj wszystkie *reguły* określone w bieżącej bazie wiedzy.
7. **Program** - widoczna jest tutaj aktualna postać *bloku sterowania* bieżącej aplikacji, w celu jego edycji należy nacisnąć *Edycja bloku sterowania* lub kliknąć w obszarze okna podglądu tekstu.
8. **Uprawnienia** - zawiera ono listę *użytkowników* uprawnionych do otwierania bieżącej bazy wiedzy. Możemy oczywiście dodać nowego użytkownika, zmienić już istniejące dane czy usunąć konta wybranego użytkownika, aby tego dokonać musimy być użytkownikiem typu *Administrator*.

W dolnej części okna umieszczona jest nazwa bieżącej aplikacji, określona przez inżyniera wiedzy.

W atualnej wersji systemu zaimplementowano podstawowe metody weryfikacji baz wiedzy polegające na wykrywaniu wybranych anomalii.

Podsystem weryfikacji baz wiedzy umożliwia wykrycie:

- reguł powtórzonych,
- reguł pochłaniających,
- reguł sprzecznych,
- reguł niespójnych,
- reguł brakujących,
- nieużywanych atrybutów,
- nieużywanych wartości atrybutów,

Przy pomocy dostępnych narzędzi możemy usunąć anomalie oraz zmodyfikować odpowiedni atrybut.

Rysunek poniżej przedstawia podstawowe bloki funkcjonalne systemu *CAKE* oraz ilustruje występujące pomiędzy nimi zależności:



Rysunek 2.2: Bloki funkcjonalne systemu CAKE

2.2.1 Podsystem zarządzania bazami wyjaśnień systemu *PC-Shell*

W trakcie sesji konsultacyjnej systemu *PC-Shell* istnieje możliwość wykorzystania definiowanych przez twórcę aplikacji tekstów wyjaśnień "co to jest?" oraz *metafor*. W obecnej wersji systemu wyróżnia się dwa typy baz wyjaśnień odpowiadające dwóm klasom wyjaśnień dostępnych podczas sesji wnioskowania systemu ekspertowego *PC-Shell*. Bazy pierwszego typu zawierają teksty *metafor*, nazywane będą *bazami metafor*. Bazy drugiego typu zawierają teksty wyjaśnień "co to jest?". Wyjaśnienia typu "co to jest?" mają charakter bardziej szczegółowy. Odnoszą się one do warunku reguły jak i jej konkluzji. Zarówno przesłanka jak i konkluzja reguły ma postać trójki {Obiekt, Atrybut, Wartość} (w skrócie OAW) zatem wyjaśnienia typu "co to jest?" są wiązane właśnie z takimi trójkami OAW, które przybierają często postać dwójki {Atrybut, Wartość} w przypadku braku definicji

obiektu. W trakcie pracy z systemem ekspertowym *PC-Shell*, użytkownik może przeglądać teksty tego typu wyjaśnień przypisane do określonych trójek OAW. Dzięki takiej konwencji definiowania wyjaśnień można je różnicować np. ze względu na wartość atrybutu.

Wszystkie bazy danych są automatycznie indeksowane, stąd dla każdej z nich tworzone są dwa pliki:

- podstawowy;
- indeksowy;

W obecnej wersji pakietu *metafory* przechowywane są w plikach o następujących rozszerzeniach:

- "dmb" - plik podstawowy,
- "mix" - plik indeksowy,

Wyjaśnienia typu "co to jest?" są przechowywane w plikach o następujących rozszerzeniach:

- "dbw" - plik podstawowy,
- "wix" - plik indeksowy,

Nazwy plików podstawowego i indeksowego są jednakowe dla danego typu wyjaśnień. W szczególnym przypadku nazwy wszystkich plików mogą być jednakowe, może to być np. nazwa bazy wiedzy z którą związane są wyjaśnienia "co to jest?" i *metafory*. Dzięki temu można uzyskać jednolite nazewnictwo wszystkich plików systemu *PC-Shell*, zróżnicowanie typów poszczególnych plików znajdować będzie odbicie w różnych rozszerzeniach.

Rozdział 3

Moduł DemoViewer

3.1 Przeznaczenie systemu DemoViewer

Podstawowym przeznaczeniem systemu jest prezentacja poszczególnych jego elementów - w tym aplikacji demonstracyjnych - co jest zwłaszcza użyteczne w zastosowaniach dydaktycznych.

System DemoViewer służy również jako integrator aplikacji systemów pakietu *Sphinx*. Jego zadaniem może być również ułatwienie zarządzania i organizacją aplikacji.

DemoViewer umożliwia wprowadzenie następujących informacji o nowo wprowadzonych aplikacjach:

- Informacji dotyczącej zastosowanej technologii (system ekspertowy, sieć neuronowa itd.);
- Odpowiednich ścieżek umożliwiających bezpośrednie uruchamianie aplikacji z poziomu systemu DemoViewer;
- Opis aplikacji;

Wygląd głównego okna *DemoViewer* przedstawia się następująco:



Rysunek 3.1: Okno główne DemoViewer

Rozdział 4

Moduł DeTreex 4.0

4.1 Przeznaczenie modułu DeTreex 4.0

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 (reguły są najczęściej stosowaną metodą reprezentacji wiedzy w bazach wiedzy systemów ekspertowych).

DeTreex może być stosowany wszędzie tam, gdzie pojawia się problem:

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

System *DeTreex* został opracowany z zastosowaniem metody indukcji drzew decyzyjnych. Indukcji dokonuje się na podstawie zgromadzonych wcześniej danych historycznych (ilościowych i jakościowych wartości atrybutów opisujących dany problem). System nie posiada żadnego ograniczenia liczby atrybutów i ich wartości oraz liczby rekordów w bazie danych, z jakich zostanie pozyskana wiedza. Stosując mechanizm ODBC system umożliwia import przykładów z baz danych. W każdym banku, biurze maklerskim, przedsiębiorstwie, jednostce naukowo-dydaktycznej prowadzącej szereg badań, klinice medycznej itp. gromadzone są obszerne bazy danych, które stosowane są w większości przypadków do analiz statystycznych i marketingu. Posiadane bazy danych to nie tylko szereg wartości liczbowych i znaków alfanumerycznych. Bazy danych zawierają wiedzę, która nie jest jeszcze znana

explicite w danym momencie. Stosując *DeTreeX* można tą wiedzę wydobyć z baz danych i użyć znacznie efektywniej niż do tej pory, uzyskując z tego tytułu wymierne korzyści.

System *DeTreeX* może być stosowany przez:

- departamenty kredytów konsumpcyjnych i gospodarczych banków;
- firmy udzielające różnego rodzaju kredytów (pożyczek) osobom fizycznym;
- firmy tworzące lub wykorzystujące hurtownie danych, w celu pogłębionej analizy danych i odkrywania wiedzy zawartej w bazach danych;
- w trakcie szkoleń nowych specjalistów ds. kredytów konsumpcyjnych rozwijając umiejętności i doświadczenie w danej dziedzinie;
- biura maklerskie, otwarte i zamknięte fundusze inwestycyjne i emerytalne oraz inne przedsiębiorstwa w zakresie podejmowania decyzji strategicznych i operacyjnych;
- w dydaktyce i pracach badawczych wyższych uczelni;

4.2 Praca z modułem DeTreeX

Główne okno systemu pozwala na dostęp do poleceń *DeTreeX*'a. Umożliwiają one:

- wizualizację zbiorów przykładów uczących;
- wyników testowania drzewa;
- wizualizację zapisanych baz\źródeł wiedzy;
- przedstawiane w dodatkowych oknach: drzewo decyzyjne i opcje budowy drzewa;
- uruchomienie dodatkowych systemów: systemu ekspertowego *PC-Shell* oraz systemu wspomagania inżynierii wiedzy *CAKE*

Funkcje systemu dostępne są poprzez wybór odpowiedniego polecenia z menu głównego. Dostęp do poleceń możliwy jest także za pośrednictwem przycisków (ikon) umieszczonych na pasku narzędziowym.

Najważniejszą pozycją menu głównego jest *Drzewo decyzyjne*, dzięki niemu mamy dostęp do następujących funkcji systemu:

- **Generuj drzewo** - generowanie drzewa decyzyjnego dla wskazanych wcześniej przykładów uczących;
- **Przerwij pracę** - przerwanie budowy drzewa decyzyjnego. Polecenie dotyczy także przerywania testowania drzewa decyzyjnego oraz przerywania zapisu bazy\źródła wiedzy;
- **Pokaż dziedzinę** - wizualizacja dziedziny problemu. Uruchomienie polecenia powoduje wyświetlenie dodatkowego okna z tabelaryczną prezentacją atrybutów i ich wartości;
- **Pokaż drzewo graficzne** - wizualizacja drzewa decyzyjnego w postaci graficznej. Uruchomienie tego polecenia powoduje wyświetlenie dodatkowego okna z prezentacją drzew;
- **Pokaż drzewo tekstowe** - wizualizacja drzewa decyzyjnego w postaci tekstowej. Uruchomienie tego polecenia powoduje wyświetlenie dodatkowego okna z prezentacją drzewa;
- **Testuj drzewo** - testowanie drzewa decyzyjnego zbiorem przykładów testowych zapisanych w dodatkowym pliku (z rozszerzeniem *.tst);
- **Zapisz reguły do bazy wiedzy** - zapisanie reguł z drzewa decyzyjnego do pliku bazy wiedzy (z rozszerzeniem * bw);
- **Zapisz reguły do źródła wiedzy** - zapisanie reguł z drzewa decyzyjnego do pliku źródła wiedzy (z rozszerzeniem *.zw);
- **Uruchom bazę wiedzy w *PC-Shell'u*** - uruchamia zapisaną poleceniem "Zapisz reguły do bazy wiedzy" bazę wiedzy w systemie ekspertowym *PC-Shell*;
- **Otwórz bazę\źródło wiedzy w *CAKE'u*** - Uruchamia zapisaną poleceniem "Zapisz reguły do bazy wiedzy" lub "Zapisz reguły do źródła wiedzy" bazę lub źródło wiedzy w systemie *CAKE*;
- **Opcje** - wybranie tego polecenia powoduje wyświetlenie dodatkowego okna dialogowego, w którym określa się parametry budowy drzewa decyzyjnego;

W celu testowania drzewa decyzyjnego (poprawności klasyfikacji przez to drzewo) konieczne jest przygotowanie tzw. pliku testowego. Format pliku uczącego i testowego jest taki sam, różnią się one rozszerzeniem nazw plików:

- **plik uczący**: - *.lrn;

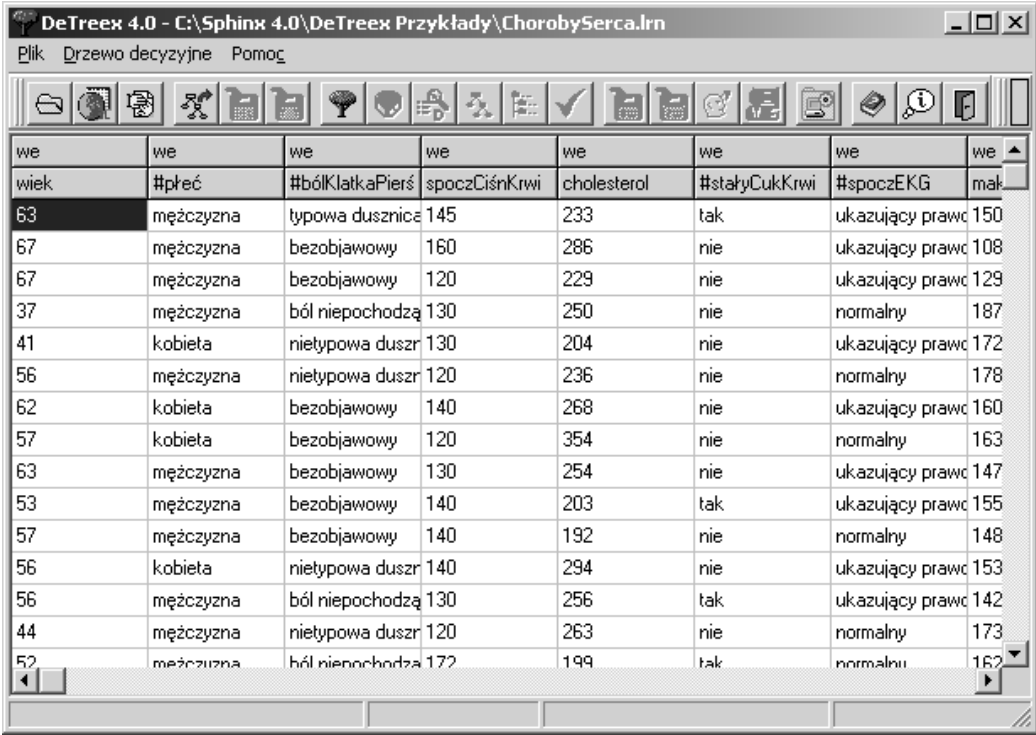
- plik testowy: - *.tst;

W systemie DeTreex możliwe jest operowanie na następujących wartościach:

- liczbowych rzeczywistych (ilościowych)
- symbolicznych wyliczeniowych (jakościowych)

Wartości liczbowe mogą należeć do przedziału od $3.4 * 10^{-38}$ do $3.4 * 10^{+38}$. Wartości symboliczne wyliczeniowe mogą zawierać ciąg znaków nie dłuższy niż 255 znaków. Wartości te powinny reprezentować pewne kategorie, pomiędzy którymi nie zachodzi żadna relacja większości, np. nazwy kolorów lub płeć. Dodatkowo istnieje możliwość zapisu przykładów z brakującymi wartościami atrybutów oznaczonych jako wejściowe (symbol *we*).

Okno główne aplikacji:



The screenshot shows the DeTreex 4.0 application window with a menu bar (Plik, Drzewo decyzyjne, Pomoc) and a toolbar. The main area displays a table with the following data:

| we | we | we | we | we | we | we | we |
|------|-----------|------------------|--------------|-------------|--------------|-----------------|-----|
| wiek | #płeć | #bólKlatkaPiersi | spoczCiśnKrw | cholesterol | #stafyCukKrw | #spoczEKG | mak |
| 63 | mężczyzna | typowa dusznica | 145 | 233 | tak | ukazujący prawc | 150 |
| 67 | mężczyzna | bezobjawowy | 160 | 286 | nie | ukazujący prawc | 108 |
| 67 | mężczyzna | bezobjawowy | 120 | 229 | nie | ukazujący prawc | 129 |
| 37 | mężczyzna | ból nie pochodzą | 130 | 250 | nie | normalny | 187 |
| 41 | kobieta | nietyпова duszr | 130 | 204 | nie | ukazujący prawc | 172 |
| 56 | mężczyzna | nietyпова duszr | 120 | 236 | nie | normalny | 178 |
| 62 | kobieta | bezobjawowy | 140 | 268 | nie | ukazujący prawc | 160 |
| 57 | kobieta | bezobjawowy | 120 | 354 | nie | normalny | 163 |
| 63 | mężczyzna | bezobjawowy | 130 | 254 | nie | ukazujący prawc | 147 |
| 53 | mężczyzna | bezobjawowy | 140 | 203 | tak | ukazujący prawc | 155 |
| 57 | mężczyzna | bezobjawowy | 140 | 192 | nie | normalny | 148 |
| 56 | kobieta | nietyпова duszr | 140 | 294 | nie | ukazujący prawc | 153 |
| 56 | mężczyzna | ból nie pochodzą | 130 | 256 | tak | ukazujący prawc | 142 |
| 44 | mężczyzna | nietyпова duszr | 120 | 263 | nie | normalny | 173 |
| 52 | mężczyzna | ból nie pochodzą | 172 | 199 | tak | normalny | 162 |

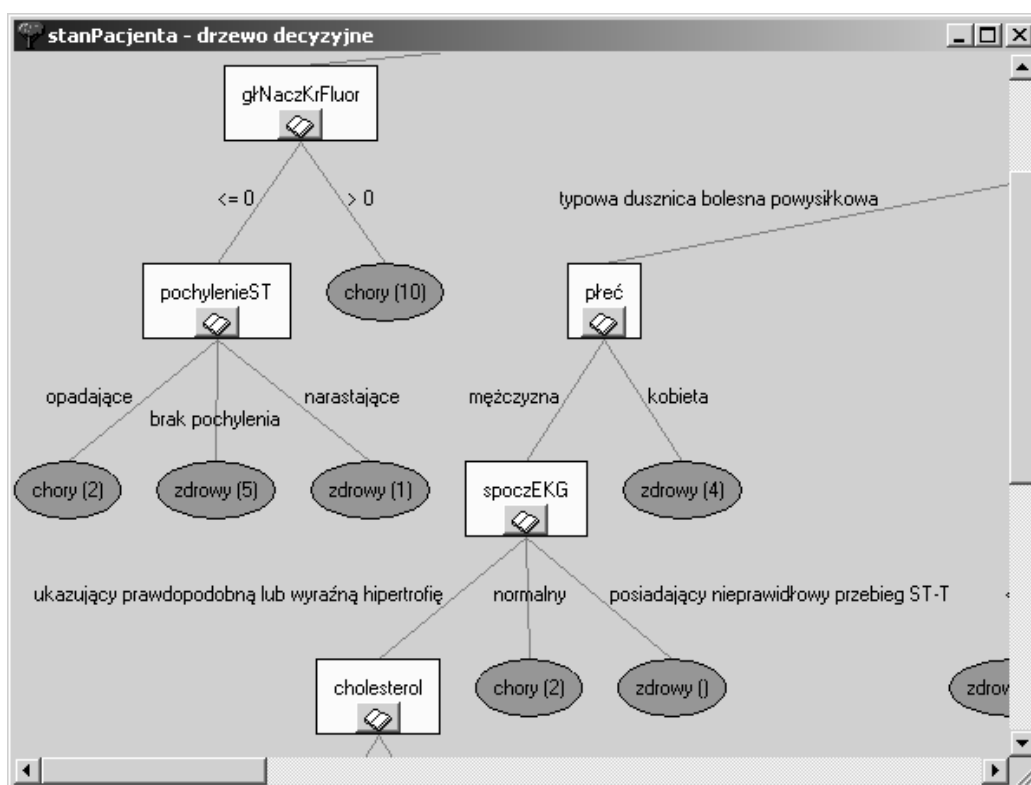
Rysunek 4.1: Okno główne DeTreex

4.2.1 Współpraca systemu DeTreex z bazami danych

W większości przypadków dane zapisywane i przechowywane są w bazach danych. System *DeTreex* umożliwia przygotowanie odpowiedniego pliku

uczącego importując zapisane w takiej bazie dane. Wymagane jest aby w systemie *Windows* zainstalowany był odpowiedni sterownik *ODBC* umożliwiający dostęp do danych w bazie.

Przykład drzewa decyzyjnego wygenerowanego przez DeTree:



Rysunek 4.2: Drzewo decyzyjne systemu *DeTree*

Rozdział 5

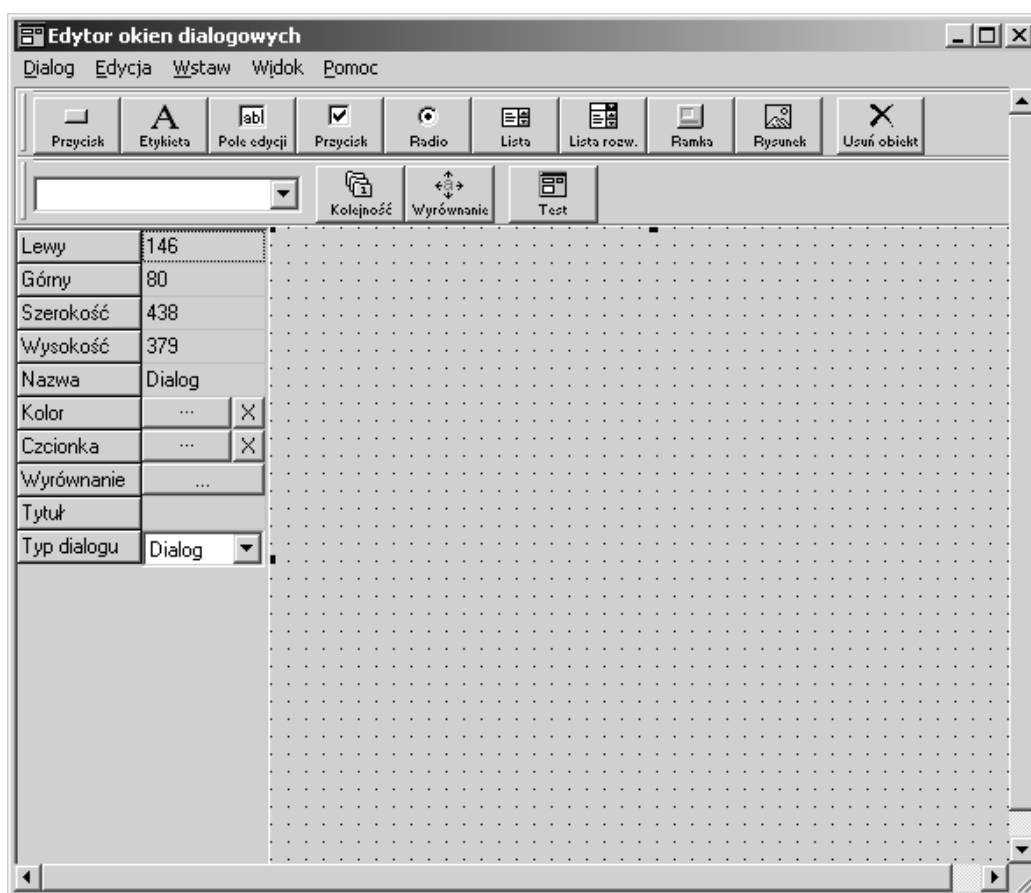
Moduł Dialog Edytor

5.1 Przeznaczenie modułu Dialog Edytor

Dialog Edytor jest pomocniczą aplikacją pakietu *Sphinx*. Jego zadaniem jest ułatwienie tworzenia i modyfikacji okien dialogowych wykorzystywanych przez użytkownika w bloku programowym języka *Sphinx*. Dzięki tzw. *instrukcjom dialogowym* języka *Sphinx*'a użytkownik tworzący program może zbudować i wykorzystać typowe okna dialogowe spotykane w środowisku *Windows*. Stwarza to nieograniczone możliwości tworzenia interfejsu komunikacji z użytkownikami aplikacji. Dotychczasowe wersje pakietu *Sphinx* wykorzystywały okna dialogowe zdefiniowane w zasobach dynamicznych bibliotek tzw. bibliotek DLL. Wymagało to posiadania przez użytkownika dodatkowego narzędzia do budowy i modyfikacji tychże dialogów. Wraz z opisywaną wersją pakietu *Sphinx* instrukcje zostały poszerzone o wykorzystywanie nowych, binarnych definicji. Wraz z pakietem dostarczany jest osobny program do edycji tychże okien. Definicje okien dialogowych przechowywane są w postaci binarnej w postaci plików najczęściej z rozszerzeniem *dlg*.

Elementy edytora: Górna część zawiera dwa panele: *panel elementów* i *panel narzędzi*. Poniżej po lewej stronie znajduje się *panel właściwości* zawierający zmienną listę właściwości wybranego elementu oraz po prawej główne *pole edycji* okna dialogowego. W polu edycji obszar z siatką to powierzchnia okna dialogowego. Na tej powierzchni umieszczane są poszczególne obiekty okna dialogowego. Obsługa tego programu jest intuicyjna i nie powinna sprawiać większych problemów.

Poniżej znajduje się obrazek przedstawiający wygląd głównego okna edytora:

Rysunek 5.1: Okno główne *Dialog Edytor*

Rozdział 6

Moduł HybRex 4.0

6.1 Przeznaczenie modułu HybRex 4.0

System HybRex jest systemem, którego zadaniem jest integracja hybrydowych technik sztucznej inteligencji oraz konwencjonalnych metod statystyki i informatyki stosowanych w rozwiązywaniu złożonych problemów decyzyjnych. Systemy wchodzące dotąd w skład pakietu *Sphinx* takie jak *PC-Shell* czy *Neuronix* umożliwiają budowę aplikacji eksperckich czy też neuronowych, zapewniając m.in. pewien poziom walidacji i udogodnień ergonomicznych. Jednak w przypadku rozwiązywania bardziej złożonych problemów wykorzystujących różne techniki informatyczne w formie hybrydowej, potrzebne jest narzędzie o odmiennej filozofii działania. Systemy dedykowane do pracy z jedną określoną technologią (np. SE lub NN) siłą rzeczy są ukierunkowane na obsługę, ergonomię tej dziedziny. Natomiast system *HybRex* jest narzędziem, którego zadaniem jest gromadzenie danych oraz metod pomocnych przy ocenie i podejmowaniu decyzji na podstawie zebranych danych. System *HybRex* nie jest systemem, który ma zastąpić jakikolwiek system zarządzania bazami danych, lecz jedynie gromadzi dane w minimalnej ilości potrzebnej do podjęcia decyzji. Główny nacisk kładziony jest na łatwość integracji i otwartość różnych technologii operujących na tych samych danych w ramach jednego systemu. Umożliwia to swobodny przepływ informacji (danych) pomiędzy metodami jest to nowa jakość, która tworzy z systemu *HybRex* doskonale środowisko do tworzenia zaawansowanych technologicznie aplikacji. System *HybRex* jest narzędziem dziedzinowo niezależnym.

Zastosowanie aplikacji opartych o ten system może być bardzo szerokie i obejmuje m.in. następujące klasy problemów:

- analiza i interpretacja danych;
- klasyfikacja;

- monitoring i systemy wczesnego ostrzegania;
- symulacje;
- prognozy;

Aplikacje systemu *HybRex* mogą być wykorzystywane m.in. w:

- przedsiębiorstwach np. do analizy kondycji finansowej, oceny budżetowania projektów będących w trakcie realizacji;
- bankach np. do oceny wniosków kredytowych zarówno firm jak i klientów indywidualnych, do oceny kondycji wewnątrzbankowej;
- holdingach i korporacjach np. do ujednocionej oceny podległych jednostek;
- biurach maklerskich np. do prognozowania i oceny kondycji finansowej spółek, oraz do prognozowania kursów giełdowych;
- medycynie np. do gromadzenia danych o pacjencie oraz oceny tych danych przez systemy ekspertowe i sieci neuronowe;
- firmach wykorzystujących hurtownie danych, stanowiąc ich dodatkowe oprzyrządowanie, np. do głębokiej analizy różnego rodzaju danych;

Podstawowe pojęcia oraz ich definicje używane w systemie *HybRex*:

Aplikacja - zdefiniowany projekt w środowisku systemu *HybRex* - na który składają się: dane zdefiniowane w *banku danych* zbiór *aplikacji metod* (zdefiniowany w *banku metod*) operujących na tych danych, oraz scenariusz rozwiązywania problemu, zdefiniowany w *banku scenariuszy*.

Bank danych Oznacza odpowiednio zorganizowany zbiór danych używanych przez daną aplikację i dostępnych dla każdej z metod.

Bank metod - system organizujący przechowywanie, utrzymanie oraz dostęp do zbioru zdefiniowanych przez twórcę aplikacji metod, przeznaczonych do rozwiązywania określonych problemów. Bank metod zawiera organizator ułatwiający wyszukiwanie metod i ich konkretyzację. Podstawowym narzędziem są tu tzw. kreatory metod, definiujące podstawowe właściwości wybranej metody. W rezultacie zdefiniowania metody powstaje *aplikacja metody*.

Bank scenariuszy - zbiór równorzędnych scenariuszy przeznaczonych dla różnych użytkowników, lub różnych podejść do rozwiązania problemu.

Dana - podstawowy element przechowujący elementarne dane typu liczba lub tekst. Dane mogą być *typu pojedynczego* - istnieje tylko jedna instancja takiej danej w ramach jednego wariantu, lub *dane czasowe*, wtedy istnieje osobna instancja danej dla każdego z okresów. Dane dzielą się również na *dane proste* oraz na *formuły*.

Formuła - wyrażenie zawierające sekwencje wartości, odwołań do danych, funkcji lub wyrażeń arytmetycznych. Obliczenie wyrażenia daje w rezultacie *wynik formuły* (danej typu formuła).

Metoda - narzędzie, algorytm, procedura lub technologia wyodrębniona do rozwiązywania oraz wizualizacji określonych klas problemów. Obecnie w systemie zdefiniowane są następujące metody:

- arkusz;
- arkusz czasowy;
- dialog;
- dostęp do bazy danych;
- interfejs zewnętrzny;
- kostka;
- paczka;
- prognoza;
- raport;
- siec neuronowa;
- symulacja;
- system ekspertowy;
- weryfikacja;
- wykres;

Scenariusz - sekwencja aplikacji metod udostępnionych użytkownikowi, określająca kolejność ich wykonywania. Realizacja scenariusza prowadzi do rozwiązania problemu. Aktywny scenariusz jest zawsze obecny w oknie nawigatora aplikacji. Z okna tego użytkownik uruchamia interesujące go metody (aplikacje metod). System umożliwia zdefiniowanie do 255 scenariuszy analizy danych. Scenariusze te operują na tych samych

danych, lecz umożliwiają m.in. kontrolę dostępu do poszczególnych elementów. Aby użytkownik mógł zmienić aktywny scenariusz musi mieć dostęp do banku scenariuszy co najmniej w trybie "do odczytu".

Stała - dana, której wartość jest taka sama we wszystkich zestawach. Jest ona przechowywana w zestawie stałych i zmian jej wartości może dokonywać tylko i wyłącznie użytkownik z prawami administratora.

Wariant danych - zbiór danych zdefiniowany w określonym zestawie, obejmujący pewien zakres analizy np. wariant optymistyczny. W ramach jednego zestawu danych może być kilka wariantów, przy czym każdy z tych wariantów dotyczy tego samego okresu analizy.

Zakres analizy - przedział czasu, w jednostkach zdefiniowanych dla danego zestawu danych, dla którego przechowywane są dane czasowe.

Zestaw danych - gromadzi zbiór wariantów (jeden lub więcej) obejmujących identyczny zakres analizy. Aby pracować z danymi musi być zdefiniowany co najmniej jeden zestaw. Każdy zestaw musi mieć określoną strukturę (typów) okresów danych czasowych i zdefiniowany pewien skończony zbiór okresów.

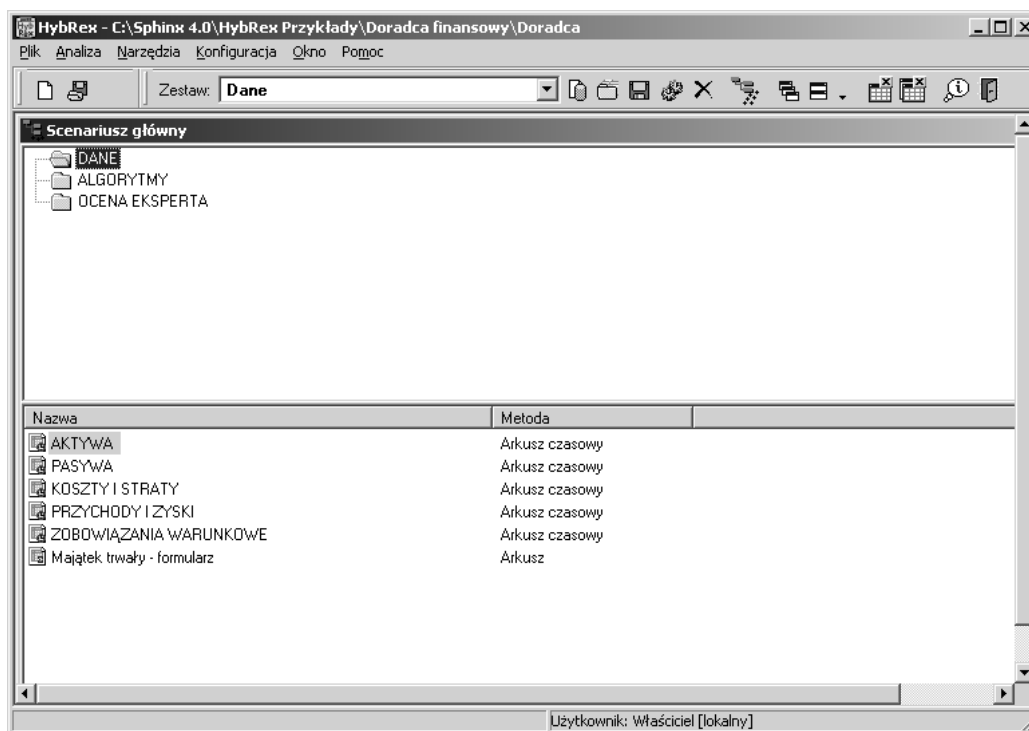
Zestaw stałych - zawiera definicje wartości danych stałych. Jest to zestaw, który jest zawsze automatycznie wczytywany. Dane z tego zestawu są wspólne dla wszystkich zestawów stąd służy on do przechowywania i udostępniania pewnych stałych parametrów jak np wskaźniki inflacji, oprocentowanie itp. Zestaw stałych jest modyfikowalny tylko przez użytkownika o prawach administratora.

6.2 Praca z programem

Podstawowym oknem z którym najczęściej będzie miał do czynienia użytkownik aplikacji, to okno zawierające scenariusz analizy - okno nawigatora. Okno to podzielone jest na dwie części: część katalogową, która zawiera podzielone tematyczne grupy metod oraz część w której są ikony identyfikujące poszczególne metody dostępne do uruchomienia przez użytkownika. Użytkownik w czasie pracy z nawigatorem może wywołać poprzez naciśnięcie prawego przycisku podręczne menu. Dostępne są w tym momencie opcje zmiany podglądu metod (Widok), opcje rozwinięcia lub zwinięcia. Drzewa katalogu oraz możliwość zmiany układu z pionowego na poziomy i odwrotnie.

Zadaniem nawigatora jest udostępnienie aplikacji metod do wykonania w zdefiniowanym przez twórcę projektu scenariuszu. Metody są uszeregowane w katalogach i po zmianie bieżącego katalogu odświeżana jest zawartość części zawierającej dostępne metody. Każda z metod identyfikowana jest odpowiednią ikoną i nazwą. Wywołanie (uruchomienie) metody następuje po dwukrotnym naciśnięciu na ikonie lub jej nazwie. W zależności od typu metody reakcja jest różna - część metod wyświetla okna (np. arkusz, arkusz czasowy, wykres) część powoduje wykonanie akcji (dostęp do baz danych, system ekspertowy, sieć neuronowa, raport).

Aby sprawnie poruszać się po wszystkich opcjach, zakładkach i aby poznać wszystkie możliwości systemu *HybRex* trzeba najpierw przyswoić podstawową wiedzę zawartą w dokumentacji tego modułu.



Rysunek 6.1: Okno nawigatora systemu *HybRex*

Rozdział 7

Moduł Neuronix 4.0

7.1 Przeznaczenie modułu Neuronix 4.0

System *Neuronix* jest narzędziem służącym wszechstronnej analizie 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 naśladowanie złożonego i jak dotychczas niewyjaśnionego 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. Sztuczna sieć neuronowa jest tylko nieudolną próbą naśladowania tych procesów, jednak obecny poziom możliwości technicznych jakimi dysponuje człowiek w dziedzinie symulowania takich procesów uległ na tyle zmianie, że stało się możliwe stworzenie systemu *Neuronix*, który jest nowoczesnym systemem z zakresu sztucznej inteligencji.

Użytkownicy systemu *Neuronix*:

1. Menedżerowie zarządzający przedsiębiorstwami w zakresie:
 - taktycznego i strategicznego planowania finansowego;
 - zarządzania finansami przedsiębiorstw;
 - podejmowania decyzji strategicznych i operacyjnych;
 - prognoz strategicznych i operacyjnych;
 - badań operacyjnych;
2. Banki oraz instytucje finansowe o zbliżonym profilu działalności w zakresie:

- oceny kondycji finansowej kredytobiorcy;
 - oceny ryzyka dla banku;
3. Biura maklerskie w zakresie:
- tworzenia modeli wybranych procesów socjologicznych na rynkach finansowych;
 - tworzenia modeli wyceny instrumentów rynku kapitałowego;
 - typowej prognozy krótko i długoterminowej dla instrumentów rynku kapitałowego;
4. Otwarte i zamknięte fundusze inwestycyjne i emerytalne w zakresie:
- alokacji jednostek;
 - ocenie ryzyka alokacji;
 - wstępnego doboru instrumentów dla analizy portfelowej;
 - prognoz wskaźników finansowych;
 - tworzenia modeli;
5. Działy logistyki i zaopatrzenia przedsiębiorstw w zakresie:
- analizy stanu materiałów;
 - zarządzania produkcją w systemie *Just In Time*;
 - planowaniu zapotrzebowania na materiały w procesie produkcyjnym;
6. Działy produkcji i eksploatacji w zakresie:
- podejmowaniu decyzji o wyłączeniu obiektu z ruchu;
 - podejmowaniu bieżących decyzji remontowych;
7. Kliniki i szpitale w zakresie:
- efektywnej analizy danych o przebiegu procesu leczenia pacjenta;
 - tworzenia skutecznych modeli procesów fizjologicznych i patologicznych;
8. W przypadku gdy konieczne jest utworzenie modelu jakościowego, operującego pojęciami lingwistycznymi np. mały, średni, duży.
9. Towarzystwa ubezpieczeniowe itp.

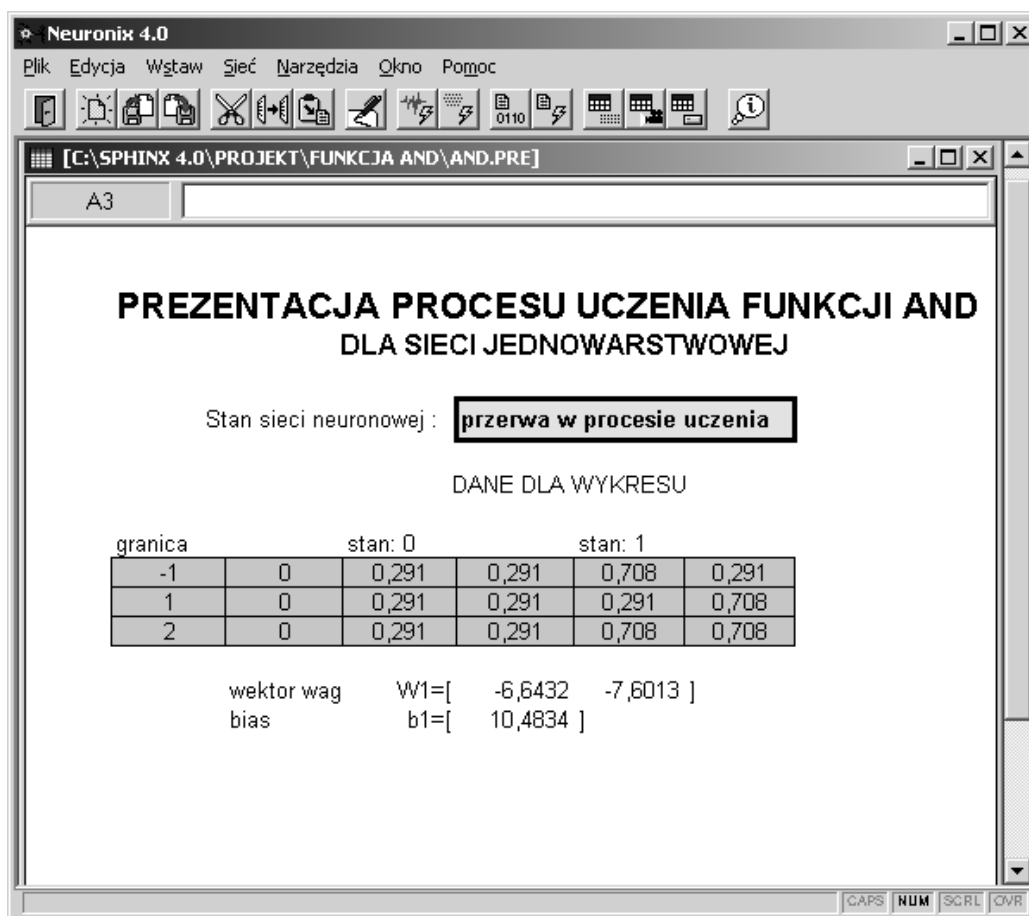
Główne funkcje systemu *Neuronix* to:

- realizacja procesu projektowania sieci, począwszy od gromadzenia próbek, poprzez generację plików uczących i testowych; uczenie i testowanie sieci, a kończąc na jej uruchamianiu dla wskazanych wartości wejściowych;
- wspomaganie niektórych etapów projektowania za pomocą wbudowanych kreatorów, m.in. Kreatora Nowego Projektu, Kreatora Plików Uczących i Testowych oraz Kreatora Formowania Szeregów Predykcyjnych;
- możliwość pracy na danych numerycznych i tekstowych konwertowanych do postaci numerycznej automatycznie, za pomocą specjalnego modułu transformującego;
- budowa aplikacji hybrydowych wykorzystujących sieć neuronową oraz szkieletowy system ekspertowy *PC-Shell*;
- dynamiczna wizualizacja struktury działającej sieci neuronowej wraz z wyświetlaniem wartości wejść i wyjść;
- monitorowanie, wizualizacja oraz automatyczne zapisywanie parametrów uczenia;
- tworzenie dowolnych raportów działania sieci poprzez wykorzystanie funkcji neuronowych pobierających dane bezpośrednio z pracującej sieci;

Najważniejszą cechą modelowania z użyciem systemu *Neuronix* jest możliwość uwzględnienia na wejściu i wyjściu modelu zmiennych lingwistycznych w postaci tekstowej. Umożliwia to budowanie modeli operujących pojęciami języka używanego przez człowieka. Modele takie są najbardziej zbliżone do rzeczywistych warunków pracy obiektów technicznych, funkcjonowania rynków finansowych oraz praw rządzących ich zachowaniem. W celu zbudowania modelu dowolnego procesu użytkownik nie musi znać w sposób jawny praw rządzących zachowaniem procesu, np. w postaci wiedzy eksperckiej lub równań matematycznych. Wystarczy jeżeli zgromadzi wymaganą liczbę danych na temat procesu. Mogą to być wszystkie informacje jakie uzna za ważne. Sieć neuronowa zachowa się w tym przypadku jak "czarna skrzynka", której zadaniem będzie przetworzenie tej informacji, znalezienie związków między wejściem i wyjściem "czarnej skrzynki" oraz uogólnienie wiedzy, którą zdobędzie sieć w taki sposób, aby wykorzystać ją do opisu podobnego procesu.

Program *Neuronix* informuje użytkownika o zaistniałych nieprawidłowych sytuacjach przy użyciu okien komunikatów. Format komunikatu o błędzie ma następującą postać: *numer_bledu: opis_bledu [dodatkowy_opis]* gdzie: *numer_bledu* jest wewnętrznym otoczeniem sytuacji nieprawidłowej, *opis_bledu* jest to słowny opis nieprawidłowości a *dodatkowy_opis* zawiera informacje, które pozwalają zlokalizować miejsce wystąpienia błędu.

Główne okno programu przedstawia się następująco:



Rysunek 7.1: Okno główne *Neuronix*

Rozdział 8

Moduł PC–Shell 4.0

8.1 Wprowadzenie do systemów ekspertowych

System ekspertowy (funkcjonuje też nazwa system ekspercki) jest to program, lub zestaw programów komputerowych wspomagający korzystanie z wiedzy i ułatwiający podejmowanie decyzji. Systemy ekspertowe mogą wspomagać bądź zastępować ludzkich ekspertów w danej dziedzinie, mogą dostarczać rad, zaleceń i diagnoz dotyczących problemów tej dziedziny.

Przykładowe obszary zastosowań systemów ekspertowych:

- diagnozowanie chorób;
- poszukiwanie złóż minerałów;
- identyfikacja struktur molekularnych;
- udzielanie porad prawniczych;
- diagnoza problemu (np. nieprawidłowego działania urządzenia);

8.1.1 Szkielety systemów ekspertowych

Klasycznym językiem używanym przy tworzeniu systemów eksperckich jest *Prolog*. Obecnie zamiast tworzyć je od podstaw, używa się gotowych szkieletów systemów ekspertowych (ang. *expert system shell*). Szkielet taki to właściwie gotowy system ekspertowy pozbawiony wiedzy.

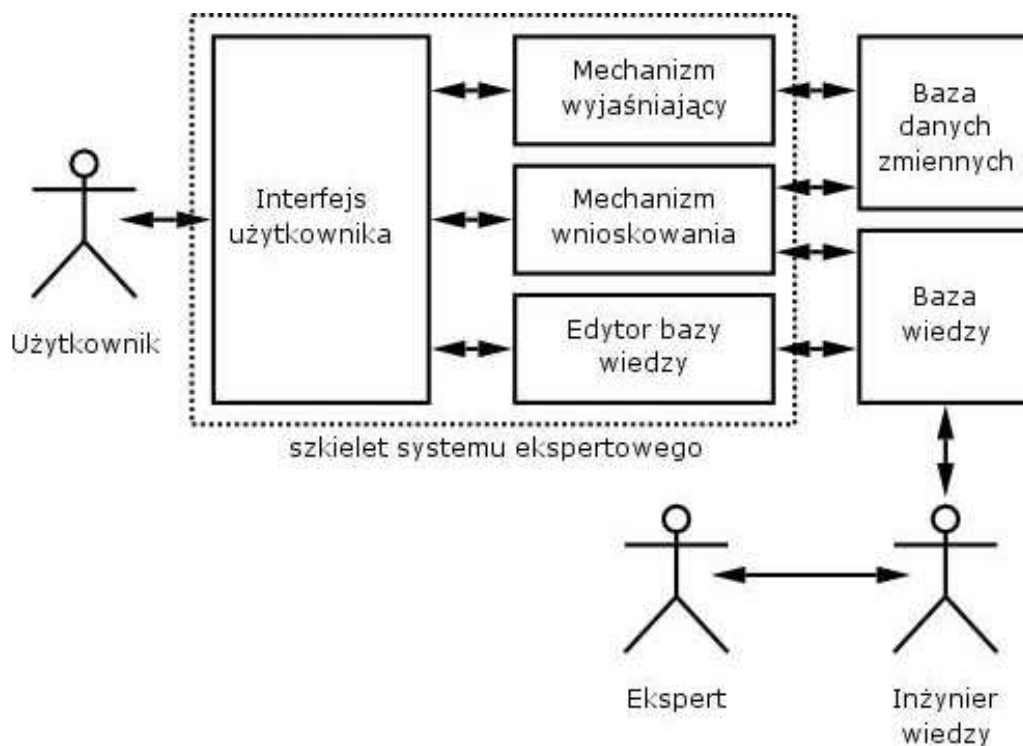
Najpopularniejsze, dostępne bezpłatnie szkielety sytemów ekspertowych to:

- CLIPS
- JESS
- MANDARAX

W Polsce w ramach opisywanego pakietu *Sphinx* rozwijany jest *PC-Shell*

8.1.2 Budowa systemu ekspertowego

Większość systemów ekspertowych jest zbudowana według następującego schematu:



Rysunek 8.1: Schemat budowy systemu ekspertowego

Składniki systemu ekspertowego to

1. Szkielet systemu składający się z:
 - Interfejsu użytkownika. Użytkownik korzysta z systemu komunikując się z nim za pomocą interfejsu użytkownika. Sprowadza się to najczęściej do zadawania pytań, udzielania informacji systemowi, oraz odbierania od systemu odpowiedzi i wyjaśnień.
 - Edytora bazy wiedzy. Dzięki wbudowanemu edytorowi możliwa jest modyfikacja wiedzy zawartej w systemie, co pozwala na rozbudowę systemu.
 - Mechanizmu wnioskowania. Jest to najważniejszy składnik systemu ekspertowego, jego zadaniem jest wyciąganie wniosków z przesłanek i pytań wprowadzanych przez użytkownika i generowanie odpowiedzi.
 - Mechanizmu wyjaśniającego. Mechanizm ten umożliwia wyjaśnienie na życzenie użytkownika dlaczego system udzielił takiej, a nie innej odpowiedzi, albo dlaczego system zadał użytkownikowi określone pytanie.
2. Baza wiedzy. Jest to drugi pod względem ważności składnik systemu. W bazie wiedzy zawarta jest wyekstrahowana od ludzkich ekspertów wiedza dotycząca określonej dziedziny. Wiedza ta zwykle zapisana jest za pomocą wybranego sposobu reprezentacji wiedzy, na przykład za pomocą reguł lub ram.
3. Baza danych zmiennych. Jest to pomocnicza baza danych w której przechowywane są wnioski uzyskane przez system podczas jego działania. Baza ta umożliwia odtworzenie sposobu wnioskowania systemu i przedstawienie go użytkownikowi za pomocą mechanizmu wyjaśniającego.

Ekstrakcją wiedzy od ekspertów zajmują się na ogół **inżynierowie wiedzy**. Jest to zwykle długi i żmudny proces, ponieważ wiedza stosowana przez ludzkich ekspertów jest zwykle wiedzą praktyczną i intuicyjną.

8.1.3 Ogólna charakterystyka modułu PC-Shell

System *PC-Shell* jest systemem silnie hybrydowym o następujących cechach:

1. W zakresie struktury systemu:

- elementy architektury tablicowej;
 - pełna integracja w ramach systemu *PC-Shell*, systemu eksperckiego oraz symulatora sieci neuronowej.
2. W zakresie reprezentacji wiedzy:
- deklaratywna reprezentacja wiedzy w formie reguł i faktów;
 - algorytmiczna reprezentacja wiedzy w formie programu zawartego w bloku sterowania (*control*);
 - pełne rozdzielenie wiedzy eksperckiej i procedur sterowania;
 - wiedza o charakterze rozproszonym zawarta w sieci neuronowej;
 - wiedza ekspercka może być zawarta w kilku źródłach wiedzy.
3. W zakresie wyjaśnień:
- wyjaśnienia typu "jak?",
 - wyjaśnienia typu "dlaczego?",
 - wyjaśnienia typu "co to jest?", dla pojęć zawartych w bazie wiedzy,
 - *metafor*y dotyczące reguł użytych w bazie wiedzy.

8.1.4 Struktura systemu PC-Shell

System *PC-Shell* składa się z następujących elementów:

- **Modułu sterującego** - koordynacja wszelkich procesów realizowanych przez system *PC-Shell*. Jednym z jego zadań jest komunikacja z użytkownikiem poprzez interfejs, który pracuje w trybie tekstowym i opracowany został w taki sposób by sugerował użytkownikowi określone warianty działania w danym kontekście, uwalniając go tym samym od konieczności ich pamiętania.
- **Translatora języka opisu bazy wiedzy** - czyta plik dyskowy zawierający opis bazy wiedzy, tłumaczy i wprowadza odpowiedni kod do bazy systemu *PC-Shell*, mieszczącej się w całości w pamięci operacyjnej komputera.
- **Modułu wnioskującego** - rozwiązuje problemy z wykorzystaniem wiedzy zawartej w bazie wiedzy. Do tego celu wykorzystywane są

odpowiednie procedury wnioskowania (rozumowania). Moduł wnioskujący systemu *PC-Shell* wykorzystuje wnioskowanie wstecz (ang. *backward chaining*).

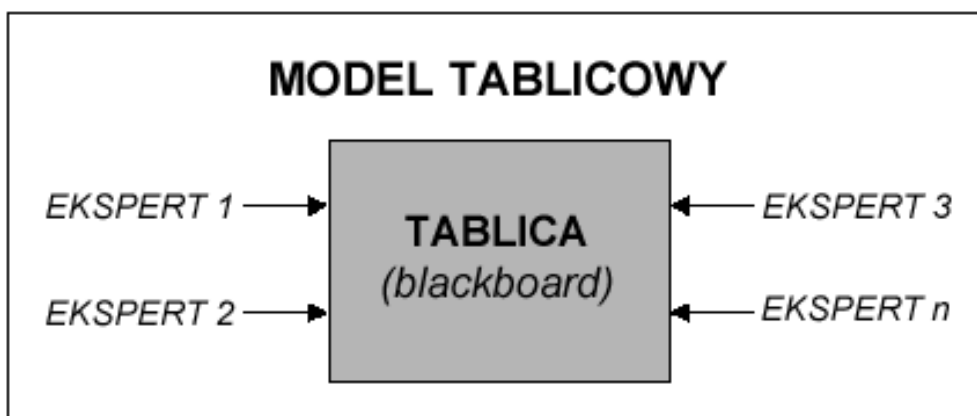
System zapewnia dwa tryby konsultacji:

- konwersacyjny;
 - programowy, sterowany programem zawartym w bloku *control* bazy wiedzy.
- **Modułu wyjaśnień** - system dostarcza retrospektywnych wyjaśnień typu *jak*, rozumowania, które umożliwiło wyprowadzenie danego zbioru konkluzji lub potwierdzenie postawionej hipotezy. W czasie konsultacji system ekspertowy często pyta o obecność określonych symptomów (faktów). W takich sytuacjach użytkownik może mieć wątpliwości, czy zadanie mu pytanie ma związek (lub jakiego rodzaju jest związek) z rozwiązywanym problemem. W tym celu system *PC-Shell* dostarcza wyjaśnień typu "dlaczego?" (ang. "*why*" *explanations*). W ramach tego typu wyjaśnień system pokazuje użytkownikowi, jaka hipoteza jest rozważana oraz w jaki sposób odpowiedź na pytanie systemu dostarczy informacji niezbędnej do potwierdzenia bądź odrzucenia tej hipotezy. *PC-Shell* dostarcza także wyjaśnień w formie tzw. *metafor*, będących objaśnieniami do reguł prezentowanych podczas wyjaśnień typu *how*. Ponadto dostępne są również wyjaśnienia typu "co to jest?", będące tekstowymi objaśnieniami pojęć zawartych w bazie wiedzy. System *PC-Shell* umożliwia kontrolowanie zakresu oraz głębokości wyjaśnień.
 - **Symulatora sieci neuronowych.**
 - **Zewnętrzny edytor bazy wiedzy.**
 - **Interfejsu użytkownika.**
 - **Interfejsu do plików dyskowych.**

8.1.5 Architektura

System *PC-Shell* jest systemem o architekturze hybrydowej, tj. łączącej w sobie różne metody rozwiązywania problemów i reprezentacji wiedzy. Interesującą właściwością systemu *PC-Shell* jest między innymi wbudowany, w pełni zintegrowany, symulator sieci neuronowej. Inną istotną cechą systemu *PC-Shell* jest jego struktura tablicowa, co umożliwia podzielenie dużej

bazy wiedzy na mniejsze moduły - zorientowane tematycznie, tzw. źródła wiedzy. Istota systemów tablicowych opiera się na analogii ze sposobem, w jaki może współpracować kilku ekspertów rozwiązujących pewien problem. Każdy z nich włącza się do rozwiązania wspólnego problemu w zakresie jego kompetencji a tworzone przez nich rozwiązanie pośrednie zapisywane są na tablicy. Tablicowe systemy ekspertowe (ang. *blackboard systems*) oparte są na takiej właśnie metaforze.



Rysunek 8.2: Istota modelu tablicowego

8.1.6 Zastosowanie

PC-Shell jest dziedzinowo-niezależnym narzędziem służącym do budowy systemów ekspertowych. Może być zastosowany w dowolnej dziedzinie: począwszy od bankowości i finansów a na zastosowaniach technicznych kończąc.

Typowe obszary zastosowań systemu *PC-Shell* to:

- systemy doradcze i wpomagania decyzji,
- dydaktyka (wyższe uczelnie i szkoły średnie).
- analizy finansowe (ekonomiczne),
- analizy wniosków kredytowych w bankach,
- doradztwo podatkowe,

- dzięki otwartej architekturze może być łatwo zintegrowany z Systemami Informowania Kierownictwa, służąc np. do automatycznej analizy wskaźników ekonomicznych,
- technika, np. do analizy danych pomiarowych.

System nadaje się do budowy zarówno małych, średnich jak i dużych aplikacji.

8.1.7 Wnioskowanie

Obecna wersja systemu wykorzystuje do rozwiązywania problemów wnioskowanie wstecz (ang. *backward chaining*). Zastosowana metoda wnioskowania wykorzystuje m.in. mechanizm nawrotów, podobny jak w spotykanych systemach prologowych. Mechanizm uzgadniania jest bardzo elastyczny i nie wymaga deklarowania typów wartości dla uzgadnianych zmiennych. Wartościami zmiennych mogą być liczby, symbole lub łańcuchy znakowe.

8.1.8 Parametryzacja baz wiedzy

Istotną cechą systemu *PC-Shell* jest możliwość parametryzacji baz wiedzy. Dzięki przyjętemu rozwiązaniu jest możliwa dynamiczna (automatyczna) zmiana wartości wybranych parametrów w bazie wiedzy, bez konieczności zmian tekstu źródłowego bazy. Dobrym przykładem zastosowania parametryzacji mogą być bazy wiedzy, w których sprawdzane są wartości pewnych wskaźników w odniesieniu do określonych wartości progowych. Jednocześnie niektóre wartości progowe mogą być zmienne, zależnie od kontekstu. Dla przykładu, inaczej można oceniać pewne wskaźniki finansowe firm należących do różnych branż. Podobne problemy pojawiają się w innych dziedzinach, np. w technice. *PC-Shell* ułatwia to zadanie, pozwalając również tworzyć kategorie parametrów.

Korzyści z zastosowania tej metody:

- może być z łatwością stosowana zarówno przez inżyniera wiedzy jak i użytkownika końcowego,
- obniża koszty wdrożenia i utrzymywania aplikacji opartej o system *PC-Shell*,
- pozwala zarówno na interakcyjną jak również programową (dynamiczną) zmianę wartości parametrów,

- umożliwia zmianę wartości wybranych parametrów oraz grup ujętych w ramach tzw. kategorii, zarówno programowo jak i interakcyjnie,
- zastosowane pojęcie kategorii zmiennych parametrycznych pozwala na lepsze dostosowanie aplikacji do specyfiki niektórych klas problemów,
- narzędzia do parametryzacji pozwalają zwiększyć czytelność oraz zmniejszyć bazy wiedzy niektórych aplikacji SE.

8.1.9 Interfejs do baz danych

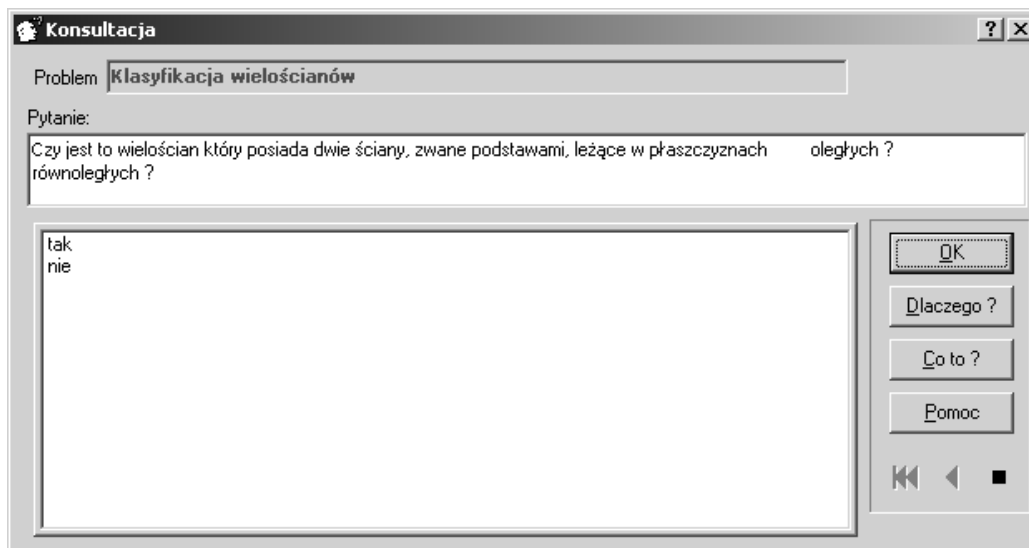
PC-Shell jest wyposażony w interfejs do typowych popularnych baz danych (np.: *dBase*, *Oracle*, itd.). Inżynier wiedzy ma do dyspozycji zestaw instrukcji do komunikacji z bazami danych z użyciem poleceń języka SQL.

Inżynier wiedzy może obecnie realizować takie operacje jak:

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

Proces komunikacji musi być obramowany etapem inicjacji dostępu oraz na końcu etapem zakończenia dostępu. Do tego celu służą instrukcje *sqlInit* oraz *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*.

Poniżej przedstawiono okno "konsultacja" systemu *PC-Shell* oraz okno wyjaśnień typu "co to?" systemu *PC-Shell*.



Rysunek 8.3: Okno "konsultacja"



Rysunek 8.4: Okno wyjaśnień typu "co to?"

Rozdział 9

Moduł Predyktor 4.0

9.1 Przeznaczenie modułu Predyktor 4.0

System *Predyktor* jest narzędziem służącym do budowy modeli ciągów czasowych oraz obiektów. System jest niezależny dziedzinowo i może zostać wykorzystany zarówno 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 ciągów 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 zbliżone obliczeniowo i z trudną do ustalenia strukturą. Wykorzystując osiągnięcia sztucznej inteligencji możliwe jest zbudowanie modelu nieliniowego przy pomocy sztucznej sieci neuronowej. Model w postaci sztucznej sieci neuronowej może być zarówno modelem ciągu czasowego jak również modelem obiektu o nadzwyczaj dobrych własnościach w zakresie modelowania nieliniowości.

9.2 Użytkownicy systemu

System Predyktor może być wykorzystywany przez:

1. Menedżerów zarządzających przedsiębiorstwami w zakresie:

- taktycznego i strategicznego planowania finansowego,
 - zarządzania finansami przedsiębiorstw,
 - podejmowania decyzji strategicznych i operacyjnych,
 - prognoz strategicznych i operacyjnych,
 - badań operacyjnych.
2. Biura maklerskie w zakresie:
- tworzenia modeli wybranych procesów socjologicznych na rynkach finansowych,
 - tworzenia modeli wyceny instrumentów rynku kapitałowego,
 - typowej prognozy krótko i długoterminowej dla instrumentów rynku kapitałowego.
3. Otwarte i zamknięte fundusze inwestycyjne i emerytalne w zakresie:
- prognoz wskaźników finansowych i kursów,
 - tworzenia modeli.
4. Działy logistyki i zaopatrzenia przedsiębiorstw w zakresie:
- planowaniu zapotrzebowania na materiały w procesie produkcyjnym.
5. Wyższe uczelnie o profilach ekonomicznych, technicznych i społecznych na etapie wspomagania dydaktyki zarówno z przedmiotów dziedzinowych, jak również dotyczących zagadnień sztucznej inteligencji.

9.3 Zastosowania systemu Predyktor

Potrzeba prognozowania występuje w każdej dziedzinie nauki i powinna być częścią niektórych decyzji badawczych (taka potrzeba występuje w każdym przedsiębiorstwie i powinna być częścią decyzji menedżerskich na poziomie zarządu oraz decyzji technicznych podejmowanych przez kadrę inżynierską w sprawach np. eksploatacji obiektów).

Dla przykładu dział marketingu potrzebuje prognoz dotyczących popytu i podaży, według segmentów rynku i kategorii wyrobów oraz wrażliwości popytu na zmiany niektórych czynników np. cen. Podobnych informacji potrzebuje dział przygotowania produkcji do sporządzenia planów wielkości i struktury produkcji. Rachunkowość jest zainteresowana przewidywaniem

kosztów, natomiast działy finansowe przygotowaniem prognoz obejmujących przychody i wydatki płynność finansową. Jednostki utrzymania ruchu są zainteresowane prognozami dotyczącymi żywotności eksploatowanych obiektów oraz bieżącego zapotrzebowania na poszczególne media niezbędne w ich pracy.

System *Predyktor* zawiera następujące algorytmy prognostyczne:

- Trendy liniowe i nieliniowe (wielomianowe, wykładnicze),
- Metody adaptacyjne (trend pełzający),
- Wygładzanie wykładnicze,
- Autoregresję przy wykorzystaniu algorytmu najmniejszych kwadratów oraz algorytmu propagacji wstecznej.

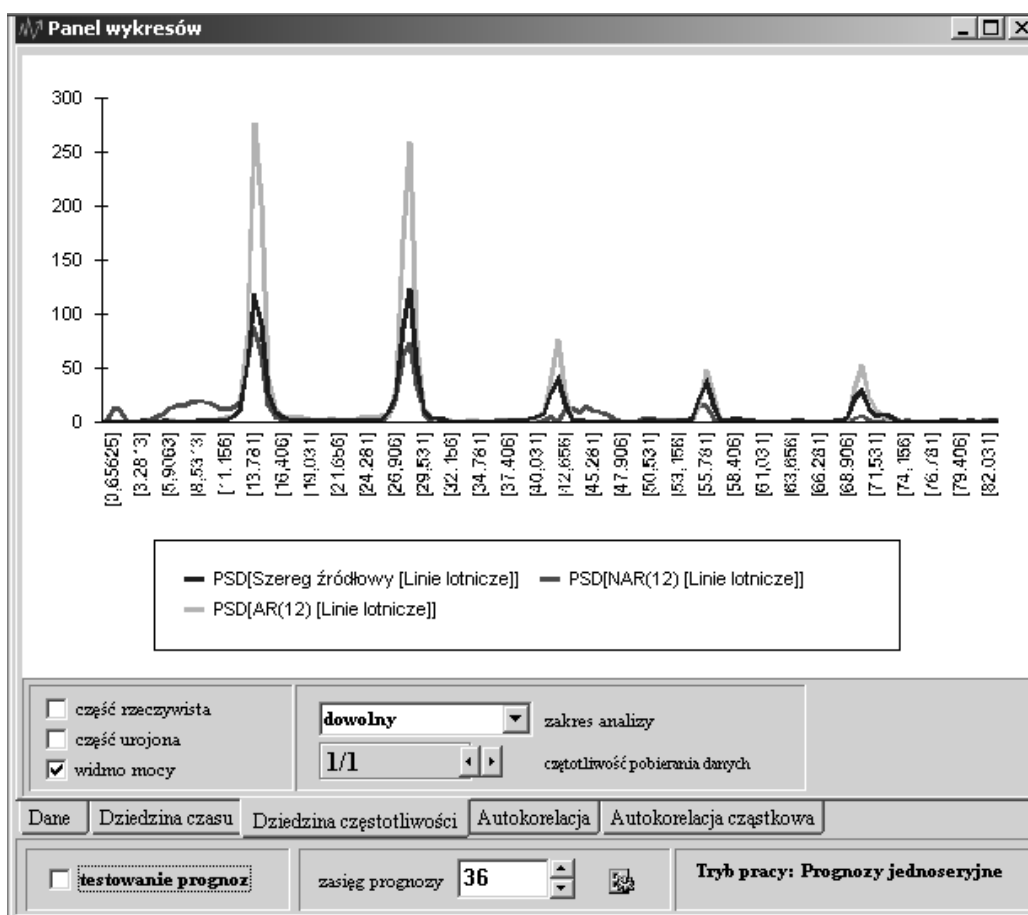
W zakresie kontroli poprawności wykonania prognozy oraz przygotowania danych system *Predyktor* korzysta z następujących metod:

- Średniej ruchomej,
- Różnicowania,
- Korekcji wartości szczytowych,
- Histogramów wykonanych na podstawie danych wejściowych i wyjściowych,
- Miar statystycznych,
- Przekształcenia Fourier'a (FFT),
- Podpowiedzi eksperta w zakresie oceny werbalnej jakości modelu prognostycznego.

System *Predyktor* posiada również możliwość raportowania w zakresie wykonanych prognoz. Szczególnie użyteczna jest opcja raportowania wieloseryjnego, wykorzystywana podczas monitoringu wielu zjawisk równocześnie. Raport uwzględnia również:

- miary statystyczne jakości prognozy,
- ocenę jakościową w postaci komentarza,
- szeregi referencyjne umożliwiające porównywanie danych rzeczywistych i prognozy.

Poniżej okno *Panelu wykresów* systemu *Predyktor*:



Rysunek 9.1: Okno *Panelu wykresu*