

Opis aplikacji dla szkieletowego systemu ekspertowego EXSYS do generowania procesu technologicznego w oparciu o wielopoziomowy model sieciowy.

Wstęp

Przedstawiana aplikacja dla szkieletowego systemu ekspertowego EXSYS umożliwia zapis i przetwarzanie wiedzy technologicznej niezbędnej do projektowania procesu technologicznego w oparciu o wielopoziomowy model sieciowy. Wiedza technologiczna jest podzielona na dwie grupy: wiedzę projektową zapisywaną w postaci reguł produkcyjnych i na wiedzę sterującą, opisującą strukturę sieci, zapisywaną w postaci ram. Aplikacja umożliwia przechodzenie przez model sieciowy czyli: zmianę poziomu na niższy, zmianę wierzchołka w obrębie jednego poziomu, wyjście na poziom wyższy, sprawdzenie wartości logicznej reguł związanych z danym wierzchołkiem oraz udostępnia mechanizmy objaśniające. Dla prawidłowego działania konieczne jest budowanie reguł i ram zgodnie z przytoczonymi poniżej zaleceniami. Celem niniejszej instrukcji jest jedynie opisanie sposobu korzystania z aplikacji, bez omawiania wielopoziomowego modelu sieciowego.

Czynności wstępne

1. Założyć, dla swojego projektu, katalog na dysku np. C:\Exsys\Walek.
2. Z poziomu PROGRAM MANAGER należy ustawić dla ikony ExsysEditor, programu Exsys, ścieżkę w PROPERTIES\WORKING DIRECTORY na kartotekę w której znajduje się projekt. W Windows95/98 PL nacisnąć prawy klawisz myszy na ikonie Exsys (właściwie to jest skrót na pulpicie, który należy tam wyciągnąć) i wybrać polecenie właściwości. Na zakładce skrót zmienić na właściwy katalog „Rozpocznij w”.
3. Aplikacja zapisana jest w kilku plikach, które należy skopiować do kartoteki w której będzie zapisywany projekt. Są to programy: **select.exe**, **liberate.exe**, **demo.cmd**, **initial.rpt**, **work.rpt**, **poziomn.ord**, **poziomw.ord**, **zmienw.ord** oraz podkatalog \NETDAT. Pliki przechowujące dane i generowane podczas działania systemu to: *select.cmd*, *result.cmd*, *work.dat*, *order.dat*, *netlevel.dat*.
4. W przypadku tworzenia nowego systemu ekspertowego, należy zmienić nazwę pliku **demo.cmd** na <NAZWA>.cmd, gdzie <NAZWA> jest nazwą utworzonego wcześniej systemu ekspertowego.
5. Teraz można uruchomić edytor wiedzy (ikona ExsysEditor) i wczytać lub założyć nowy system ekspertowy *.RUL. System ten powinien posiadać zadeklarowane pewne zmienne, które są niezbędne do prawidłowego działania aplikacji. Konieczne jest więc zdefiniowanie (menu *Rules\Variable List*) następujących zmiennych: *[NetLevel]* - sieć bieżąca, *[NetNode]* - bieżący wierzchołek, *[ToNet]* - sieć docelowa, *[ToNode]* - wierzchołek docelowy, *[Stage]* - etap, wszystkie typu **string**.

Po wykonaniu tych czynności można przystąpić do wprowadzania wiedzy technologicznej.

Uwagi

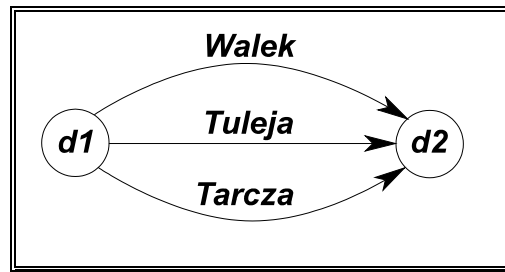
- **Nazwa reguły** (określana po kliknięciu na przycisk Name w okienku edycji reguł), **nazwa celu projektowego** (choice) i nazwa wpisywana w kolumnie **cel w ramie** (choice/rule w pliku *.net) opisującej strukturę sieci muszą być identyczne, również co do wielkości znaków,
- Plik zawierający opis ramy dla sieci poziomu I musi posiadać nazwę 1_level.net i znajdować się w podkatalogu \NETDAT naszego projektu,
- Dodanie do części THEN składowej typu: *[ToNet]* IS GIVEN THE VALUE "WALEK.FRM" wymaga kliknięcia na przycisk Var./Math w okienku edycji reguły, wybrania odpowiedniej zmiennej z wyświetlanej listy zmiennych Variables i ostatecznie

wpisania wartości, która ma być nadana. Wartość tę należy ująć w znaki "" dla zmiennych typu string.

- Dodanie komendy typu REPORT następuje poprzez wybranie przycisku Command, następnie z wyświetlanego okienka Data Acquisition Commands przycisku Report Generator. W pojawiającym się okienku dialogowym należy wpisać jedno z poleceń: *PoziomW.ord*, *PoziomN.ord* lub *ZmienW.ord*.
- Dodatkowe uwagi zawarte w celu samo udokumentowania np. objaśnienie skrótowej nazwy celu projektowego można wpisywać po kliknięciu na przycisk Note w okienku edycji reguły.

Przykład

Wiedzę projektową możemy podzielić na trzy poziomy jej uszczegółowienia. Poniżej znajduje się przykład, który obrazuje sposób postępowania z aplikacją przy tworzeniu wiedzy dla poziomu I. Wiedza poziomu I pozwala zakwalifikować przedmiot do określonej grupy i określić strukturę procesu obróbki. W ramach projektu analizowane są tylko części obrotowe, z pośród których zostały wybrane następujące klasy: wały, tarcze i tuleje. Sieć **poziomu I** ma strukturę pokazaną na rysunku 1.



Rys. 1 Sieć dla Poziomu I.

Występuje tu następujący zbiór decyzji projektowych $D = \{d1, d2\}$:

- **d1** - decyzja wyboru USPT,
 - **d2** - decyzja zakończenia lub powrotu do ponownego wyboru USPT,
- oraz zbiór działań projektowych $T = \{Walek, Tuleja, Tarcza\}$:

- **Walek** - USPT dla części typu wałek,
- **Tuleja** - USPT dla części typu tuleja,
- **Tarcza** - USPT dla części typu tarcza,

W celu praktycznego przedstawienia sieci wykorzystano, dla pokazania relacji R pomiędzy decyzjami i działaniami, zapis tej struktury w postaci rama. Poniżej przedstawiona jest rama dla I poziomu *-1_level.net*: (plik ASCII)

StartNode	Choice/Rule	EndNode
d1	Walek	d2
d1	Tuleja	d2
d1	Tarcza	d2

gdzie:

- **StartNode** - decyzja poprzedzająca,
- **Choice/Rule** - działanie projektowe, jest to nazwa reguły z bazy wiedzy *.rul jak również nazwa celu projektowego (*ang. choice*),
- **EndNode** - decyzja następna, wierzchołek docelowy.

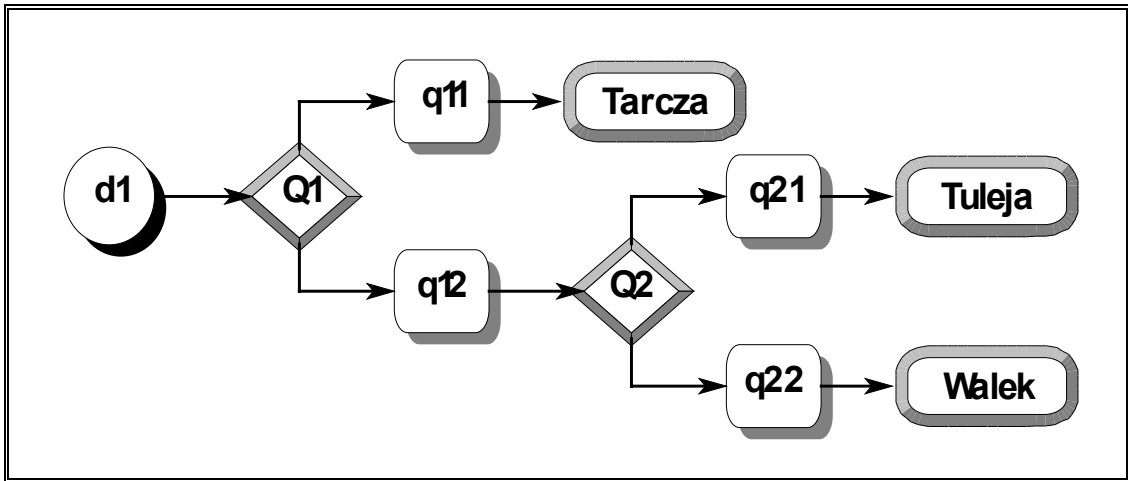
Następnym krokiem jest określenie zbioru atrybutów $Q=\{Q1,Q2,\dots\}$ i wartości

przyjmowanych przez te atrybuty. Atrybutami mogą być zarówno zmienne proste (*ang. variable*), wyrażenia arytmetyczne na zmiennych prostych, jak również zmienne wyliczeniowe (*ang. qualifier*). Tabela 1 przedstawia atrybuty dla **poziomu I**, gdzie:

- [LPOMAX] - zmienna prosta, typ: numeric; Całkowita długość PO,
- [DPOMAX] - zmienna prosta, typ: numeric; Max średnica PO.

Tabela 1 Zbiór atrybutów dla Poziomu I

I.p.	Typ	Nazwa	Treść	Wartości
Q1	expression	[LPOMAX]/[DPOMAX]	Stosunek długości PO do jego średnicy.	q11= (<=1), q12= (>1)
Q2	qualifier	ISHOLE	Czy istnieją przelotowe otwory osiowe?	q21=TAK, q22=NIE



Rys. 2 Drzewo decyzyjne dla Poziomu I.

Na podstawie powyższych danych możemy stworzyć drzewa decyzyjne, które obrazują sposób podejmowania decyzji. Dla przykładu możemy zobaczyć na rysunku 2 drzewo decyzyjne dla wierzchołka d1.

Teraz na podstawie drzew decyzyjnych możemy formułować reguły, czyli warunki. Reguła dla poziomu I, dla wyboru działania projektowego **Walek** z wierzchołka **d1** będzie miała postać (jeżeli jest drugi poziom w pliku o nazwie 2_walek.net):

```

RULE NUMBER: 1    (Walek)
IF:
    [LPOMAX]/[DPOMAX]>1
    and Czy istnieja przelotowe otwory osiowe? : NIE

THEN:
    Walek - Confidence=1
    and [TONET] IS GIVEN THE VALUE "2_walek.net"
    and [TONODE] IS GIVEN THE VALUE "d1"
    and REPORT(PoziomN.ord)

NOTE:
    USPT dla przedmiotów klasy walek.
  
```

Reguła dla poziomu I, dla wyboru działania projektowego **Tuleja** z wierzchołka **d1** będzie miała postać (jeżeli nie ma drugiego poziomu i następuje tylko zmiana wierzchołka):

```

RULE NUMBER: 2    (Tuleja)
IF:
    [LPOMAX]/[DPOMAX]>1
    and Czy istnieja przelotowe otwory osiowe? : TAK

THEN:
    Tuleja - Confidence=1
    and [STAGE] IS GIVEN THE VALUE "NoLevel"
  
```

and REPORT (ZmienW.ord)

NOTE:
USPT dla przedmiotów klasy walek.

Przedstawiony powyżej schemat konstruowania wiedzy technologicznej (sieci - ramy - atrybuty - drzewa - reguły) będzie powielany przy tworzeniu kolejnych poziomów.

W przypadku reguł, w których następuje wyjście na wyższy poziom (powrót) należy:

```
RULE NUMBER: X      (XXX)
IF:
    warunek
    and warunek ...

THEN:
    XXX - Confidence=1
and REPORT (PoziomW.ord)
```

Użycie hipertekstu

W sieciach 1, 2 i 3 poziomu w sieciach (ramach) dla oznaczenia działań projektowych (celi decyzyjnych/nazw reguł) używamy nazw skrótowych. W celu ich objaśnienia w systemie należy użyć *hypertext'u*. Należy tego dokonać w następujący sposób. Dla wszystkich celów decyzyjnych (Choice list) należy dodać po dwa znaki ^ przed i po.

Przykład:

Było	Ma być
OZWD	^^OZWD^^
BOZWD	^^BOZWD^^
WRWP	^^WRWP^^

Wówczas system wie, że ma do czynienia z hipertekstem. Objawia się to niebieskim tłem pod tekstem *choice'a* w okienku *Results*. Po dwóokrotnym kliknięciu wywoływany jest skojarzony ekran z właściwym opisem danego kodu.

Takie ekrany należy samemu zdefiniować. Ich definicja powinna znajdować się w pliku <Nazwa>.scr. Przy użyciu języka *ScreenLanguage* można definiować dowolne ekrany. Oto przykładowa zawartość pliku *.scr:

```
~~Walek                               /* nazwa hipertekstu
~SET_WIN 30,50,400,200                 /* rozmiar okna: x,y,len_x,len_y
~PATTERN SOLID                         /* wypełnienie ciągłe
~BACKGROUND LTGRAY                    /* tło jasnoszare
~CLS                                   /* zmazanie ekranu
USPT obróbki dla walka sztywnego     /* opis, tekst
~BUTTON "OK", "",175,160,50,30 -X -I  /* klawisz OK,wymiary,opcje
~END                                   /* koniec definicji ekranu

~~Finish
~SET_WIN 30,50,400,200
~PATTERN SOLID
~BACKGROUND LTGRAY
~CLS
Obróbka wykarczująca powierzchni zewnętrznych
~BUTTON "OK", "",175,160,50,30 -X -I
~END
```

Definicje wszystkich ekranów mogą znajdować się w jednym pliku. Początek ekranu hipertekstu zaczyna się od *~~Choice* a kończy *~END*. Pomiędzy tymi liniami możemy używać instrukcji, których opis możemy znaleźć w helpie pod hasłem *ScreenLanguage*.

Użycie ReportGenerator'a

W celu dokumentowania pracy systemu ekspertowego należy wykorzystać *Report Generator*. Pozwala on między innymi na zapisywanie do plików ASCII danych wyprowadzanych z systemu. Najczęściej stosuje się go w regułach w części THEN. Dla przykładu:

```
RULE NUMBER: X      (XXX)
IF:
    warunek
  and warunek ...

THEN:
    XXX - Confidence=1
  and REPORT(Zapisz.rpt)
```

linia REPORT(Zapisz.rpt) oznacza wywołanie i uruchomienie pliku o nazwie *zapisz.rpt*. W tym pliku znajdują się polecenia w języku ReportGenerator Language, które możemy znaleźć w help'ie pod tym hasłem. Kolejne wiersze pliku są interpretowane i wykonywane. I tak przykładowo może wyglądać plik raportu:

```
File Wyniki.dat/a
"Dobrana wielkość naddatku dla powierzchni"
"o długości L=[[L]] i średnicy D=[[D]]"
"wynosi: [[NadFNS]] mm"
"-----"
/*Koniec raportu
```

Komenda *File <nazwa pliku>* powoduje utworzenie pliku o podanej nazwie. Opcja */a* powoduje, że kolejne dane będą dopisywane do pliku (append). Bez tej opcji zawartość pliku zostaje najpierw skasowana (zobacz do pliku initial.rpt gdzie pliki ze zmiennymi systemu są przy każdym restracie systemu kasowane). W cudzysłowiaach „,” zapisujemy tekst, który ma znaleźć się w pliku. Jeżeli chcemy aby pewne wartości zmiennych Exsysa zostały zapisane, wówczas dla zmiennych numerycznych należy je podać w podwójnych nawiasach kwadratowych [[ExsysVar]]. Po znakach */** może wystąpić komentarz nie zapisywany do pliku docelowego. Aby zapisać do pliku tekst wybranego Choice'a należy napisać linię: *C /T*, np. "Wybrane działanie: " *C /T*

Wynik zapisany do pliku *Wyniki.dat* może być następujący:

```
Dobrana wielkość naddatku dla powierzchni
o długości L=38 i średnicy D=23
wynosi: .19 mm
-----
Dobrana wielkość naddatku dla powierzchni
o długości L=77 i średnicy D=47
wynosi: .24 mm
-----
Dobrana wielkość naddatku dla powierzchni
o długości L=50 i średnicy D=33
wynosi: .2 mm
-----
```

Metody doboru parametrów w systemie Exsys

Na niższych poziomach wiedzy zachodzi potrzeba doboru konkretnych parametrów. Dane opisujące te parametry powinny być zapisane w taki sposób aby umożliwić ich dobór. W systemie Exsys mamy do dyspozycji bardzo wiele możliwości (patrz DataAcquisition). Przypadek taki wystąpi na przykład podczas generowania półfabrykatu i stanów pośrednich, gdzie dla poszczególnych powierzchni należy dobrać odpowiednie naddatki obróbkowe. Poniżej opisane zostały najczęściej stosowane sposoby.

Zapis danych w regułach

Założmy, że mamy regułę wybierającą na poziomie 3 transformator FNS01 obróbki wykańczającej dla powierzchni zewnętrznych cylindrycznych:

```

Rule: FNS01
IF
    Spełnione warunki dla szlifowania: kłowego
THEN
    FNS01 - Confidence=1
and    REPORT(ZmienW.ord)

```

Możemy utworzyć qualifier postaci:

```

Q: Stan powierzchni przed obróbką:
    1 - wykańczająca
    2 - kształtująca
    3 - zgrubną
    4 - półfabrykatu

```

i dodać go do części IF powyższej reguły i otrzymamy:

```

Rule: FNS01
IF
    Stan powierzchni przed obróbką: wykańczająca
    Spełnione warunki dla szlifowania: kłowego
THEN
    FNS01 - Confidence=1
and    REPORT(ZmienW.ord)

```

Wówczas system dodatkowo nie zna wartości tego qualifiera. Ale możemy wykorzystać dostępny w Exsysie mechanizm zagnieżdżania reguł (max 128 poziomów). Jeśli system nie zna jakiejś wartości to:

1. Sprawdza, czy dana zmienna nie jest akojarzona z jakimś zewnętrznym źródłem danych,
2. Sprawdza wszystkie dostępne reguły, czy w części THEN nie ma przypisania wartości. Jeśli znajdzie takie reguły to wówczas je uruchamia i zadaje pytania w części IF tych reguł aby stwierdzić jakie będą wartości w części THEN,
3. Jeśli powyższe metody zawiodą to wyświetla okienko z prośbą o podanie tej wartości przez użytkownika.

Możemy więc napisać pewną grupę reguł bez nazwy, nie mających Choice'a w części THEN, następującej postaci:

```

Rule:
IF
    [Ra]<=2.5
and    [IT]<=8
and    [L]>63
and    [L]<=100
and    [D]>25
and    [D]<=40

THEN
    Stan powierzchni przed obróbką: wykańczająca
    [NadWyk] IS GIVEN THE VALUE 0.2
and    REPORT(Zapisz.rpt)

```

Tutaj dwa pierwsze warunki sprawdzają czy nasza powierzchnia była uzyskana obróbką wykańczającą, następne zaś ustalają odpowiednie zakresy długości L i średnicy D, dla których wartość naddatku powinna wynosić 0.2 [mm] (zmienna NadWyk). Ostatnia linijka powoduje wykonanie raportu Zapisz.rpt i zapisanie do pliku odpowiednich danych.

Zaleta: prostota wykonania.

Wada: należy napisać tyle reguł ile jest różnych zakresów do doboru naddatków dla poszczególnych rodzaju obróbki.

Odczyt danych poprzez Frame Interface – ramy

Zamiast pisać wiele reguł można te dane zapisać np. w postaci ramy. Przykładowa konstrukcja ramy znajduje się poniżej (plik *.frm ASCII):

Lmin	Lmax	Dmin	Dmax	Qnom
10	16	2.5	4	0.17
10	16	4	6.3	0.17
10	16	6.3	10	0.18
16	25	4	6.3	0.17
16	25	6.3	10	0.18
16	25	10	16	0.19
25	40	6.3	10	0.19
25	40	10	16	0.19
25	40	16	25	0.19
40	65	6.3	10	0.24
40	65	10	16	0.2
40	65	16	25	0.2
40	65	25	40	0.2
65	100	10	16	0.24

Opis wykorzystania Frame Interface możemy znaleźć w helpie. My będziemy korzystać z funkcji, która odczytuje konkretne wartości z ramy. Funkcja ma następującą postać:

```
FRAME („filename”, test_expression, slot_reference)
```

Gdzie:

Filename – nazwa pliku z ramą, musi być w „”

Test_expression – wyrażenie logiczne, użyte do odszukania pożądanej linii w ramie,

Slot_reference – żądana kolumna ramy, z której należy odczytać wartość. Jeżeli odczytywana wartość jest typu *string* to musi być: *\$name\$*, jeśli zaś jest typu *numeric* to *#name#*.

Przykładowo jeżeli z ramy o nazwie *ObrWyk.frm* chcemy odczytać wartość naddatku dla odpowiedniego zakresu D i L z kolumny *Qnom*, to musimy napisać tak:

```
FRAME ("ObrWyk.frm", (#LMIN#<[[L]]) AND (#LMAX#>[[L]]) AND (#DMIN#<[[D]]) AND (#DMAX#>[[D]]), #QNOM#)
```

Gdzie:

Filename – „ObrWyk.frm”

Test_expression – wyrażenie logiczne: (#LMIN#<[[L]]) AND (#LMAX#>[[L]]) AND (#DMIN#<[[D]]) AND (#DMAX#>[[D]]), W nawiasach [[x]] znajdują się zmienne zdefiniowane w Exsysie, natomiast kolumny *###* są odczytywane jako zmienne numeryczne,

Slot_reference – żądana kolumna ramy: *#Qnom#*, jako zmienna numeryczna.

Użycie odczytu z ramy w regule jest następujące:

```
Rule: FNS01
IF
    Spełnione warunki dla szlifowania: kłowego
THEN
    FNS01 - Confidence=1
and REPORT(ZmienW.ord)
and [NadWyk] IS GIVEN THE VALUE FRAME("ObrWyk.frm", (#LMIN#<[[L]])
AND (#LMAX#>[[L]]) AND (#DMIN#<[[D]]) AND (#DMAX#>[[D]]), #QNOM#)
and REPORT(Zapisz.rpt)
```