

Uživatelská příručka k systému FEL-Expert 4.0

23. 11. 1999

Marek Obitko
xobitko@fel.cvut.cz

1. Obsah

1. OBSAH	1
2. ÚVOD.....	2
3. ZÁKLADNÍ POJMY	2
3.1 INFERENČNÍ SÍŤ	2
3.2 KONZULTACE.....	2
3.3 VAZBY	3
3.4 TAXONOMIE	3
4. ZÍSKÁVÁNÍ DAT.....	4
4.1 OBECNÝ POPIS	4
4.2 IMPLEMENTOVANÉ METODY	4
4.2.1 Pravděpodobnost.....	4
4.2.2 Činitel jistoty.....	5
4.2.3 Hodnota	5
4.2.4 Výběr z možností.....	6
5. TEXTOVÝ TVAR BÁZE ZNALOSTÍ.....	7
5.1 PŘÍKLAD.....	7
5.2 SYNTAXE A SÉMANTIKA	9
5.3 PARAMETRY OBJEKTŮ PRO ZÍSKÁVÁNÍ DAT	10
5.3.1 Parametry pro vstup hodnoty	10
5.3.2 Parametry pro vstup volby	10
5.4 PARAMETRY SÍTĚ	11
6. PRÁCE SE SYSTÉMEM	11
6.1 OBECNÉ INFORMACE	11
6.2 EDITOR BÁZE ZNALOSTÍ	11
6.3 VYTVOŘENÍ INFERENČNÍ SÍTĚ.....	12
6.4 KONZULTACE.....	12
6.4.1 Vstup taxonomie	13
6.5 POPIS UZLŮ	14
6.6 DALŠÍ MOŽNOSTI.....	15
7. PRÁCE S DATABÁZEMI	15
7.1 STRUKTURA TEXTOVÝCH DATABÁZÍ	15
7.2 STRUKTURA RELAČNÍCH DATABÁZÍ.....	16
7.3 POUŽITÍ VSTUPNÍ DATABÁZE	16
7.4 POUŽITÍ VÝSTUPNÍ DATABÁZE	17
7.5 DÁVKOVÉ ZPRACOVÁNÍ	17
7.6 PROHLÍŽEČ DATABÁZÍ.....	18
8. KONZOLOVÁ VERZE SYSTÉMU.....	19
9. SHRUTÍ.....	19
10. LITERATURA.....	20

2. Úvod

Tento dokument má sloužit jako příručka k seznámení se s prací se systémem FEL-Expert. Popisuje se především uživatelské rozhraní. Předpokládá se povědomí o DOSovské verzi systému FEL-Expert [5, 7] nebo alespoň o diagnostických expertních systémech obecně. Nelze zde najít podrobný popis použitých algoritmů. Jejich podrobné vysvětlení lze nalézt v literatuře [2, 3, 4].

3. Základní pojmy

V této kapitole jsou stručně popsány pojmy, které budou dále používány. V následujícím textu se používá pojem pravděpodobnost, mělo by však zůstat na zřeteli, že se obecně nejedná o pravděpodobnost definovanou matematicky dle frekvence výskytu ve všech možných případech, ale o subjektivní pravděpodobnost. Tato pravděpodobnost bývá také nazývána pseudopravděpodobnost.

3.1 Inferenční síť

Základem pro reprezentaci báze znalostí expertního systému jsou pravidla ve tvaru

if předpoklad E **then** závěr H **with** pravděpodobnost $P(H|E)$
else závěr H **with** pravděpodobnost $P(H|not\ E)$

Pro výpočet (pseudo)pravděpodobnosti závěru se používá Bayesův vztah [2, 3].

Soubor těchto pravidel tvoří orientovaný graf, kde vrcholy jsou tvrzení (E , H) a hrany (ohodnocené pravděpodobnostmi) tvoří pravidla. Tento graf se nazývá inferenční síť.

Základní inferenční síť doplňují logické kombinace výroků ve tvarech *not* E , E_1 and E_2 , E_1 or E_2 (dílčích předpokladů může být více). Rozlišuje se tedy mezi dvěma typy uzlů – bayesovský uzel a logický uzel.

Bayesovský uzel reprezentuje tvrzení, jehož pravděpodobnost se dá vyhodnotit dle Bayesova vztahu. Má danou apriorní pravděpodobnost. Aposteriorní pravděpodobnost (po pozorování) se buď spočítá z pozorovaných předpokladů nebo se získá přímým pozorováním.

Logický uzel reprezentuje tvrzení, jehož pravděpodobnost se dá vyhodnotit dle logické kombinace předpokladů. Pravděpodobnost tvrzení se vyhodnocuje pomocí vztahů převzatých z fuzzy logiky.

Uzly (tvrzení) se také dají rozdělit dle své polohy v inferenční síti na vrcholové, listové a mezilehlé. Vrcholové uzly jsou uzly, z nichž nevede žádná orientovaná hrana. Vrcholové uzly reprezentují vrcholové hypotézy. Listové uzly jsou uzly, do nichž nevede žádná orientovaná hrana, reprezentují tedy tvrzení, která musí být ověřena pozorováním. Ostatní uzly jsou mezilehlé. Ty reprezentují mezilehlá (dílčí) tvrzení.

Každý uzel může být navíc dotazovatelný (listový uzel musí být dotazovatelný) nebo může reprezentovat cílovou hypotézu.

3.2 Konzultace

Konzultace je proces, při kterém se systém snaží zjistit platnost cílových hypotéz. Systém při konzultaci vybírá nejpravděpodobnější cílovou hypotézu a snaží se jí dokázat nebo vyvrátit. Hledá tvrzení, které by mohlo k tomuto nejvíce přispět (pomocí skórovací funkce, viz [3]), dokud nedorazí k dotazovatelnému uzlu. Na jím reprezentované tvrzení se dotáže a

poté zpětně od něj šíří získanou informaci v inferenční síti a upravuje tak aktuální model. Tento proces se opakuje, dokud není hypotéza kompletně vyšetřena. Konzultace končí v okamžiku vyšetření všech cílových hypotéz.

3.3 Vazby

Vyšetřování se dá ovlivnit tzv. vazbami (linky).

Prioritní (nepodmíněná) vazba určuje, že některý uzel se musí vyšetřit před vyšetřením jiného uzlu. Je tedy dána dvěma uzly a orientovanou vazbou mezi nimi.

Kontextová (podmíněná) vazba je dána dvěma uzly a rozsahem pravděpodobností. Pokud je pravděpodobnost vzájemného uzlu v daném rozsahu, systém položí dotaz na vázaný uzel (bez dalších omezení). Pokud pravděpodobnost vzájemného uzlu leží mimo daný rozsah, snaží se systém nejprve splnit kontext, tzn. vyšetřuje vázaný uzel. Pokud se podaří splnit kontext (pravděpodobnost vzájemného uzlu je v daném rozsahu), systém se ptá na vázaný uzel. Pokud se kontext nepodaří splnit, systém se již v průběhu konzultace na vázaný uzel neptá.

3.4 Taxonomie

Taxonomie [5, 6, 7] umožňují dále strukturovat znalosti uložené v inferenční síti. Taxonomie má tvar stromového grafu, kde jednotlivé uzly jsou taxonomické třídy. Třídy jsou ohodnoceny seznamem uzlů, které do dané třídy patří (seznam může být prázdný). Podtřídy dané třídy jsou třídy směrem k listům stromu, nadtřídy jsou třídy směrem ke kořenu stromu. Taxonomie jsou významné zejména pokud máme inferenční síť s mnoha uzly a cílovými hypotézami.

Taxonomie určují hierarchii tvrzení (uzlů). Pokud máme nějaké doplňující informace o charakteru konzultovaného problému, lze pomocí taxonomií zúžit pozornost na podmnožinu cílových hypotéz. Také lze ovlivnit množství zodpovědaných dotazů, pokud je nějaká informace získaná z okolního světa jednoznačná (pravděpodobnost 0/1).

První typ taxonomie, dotazovatelný (hypothesis¹), udává hierarchii cílových hypotéz. Pokud je vybraná některá třída této taxonomie, vyšetřují se pouze cílové hypotézy spadající do této třídy a podtříd (určí se jako sjednocení hypotéz v třídě a podtřídách). Pokud takových taxonomií existuje více, vyšetřuje se průnik všech hypotéz spadajících do zvolených tříd taxonomií.

Další tři typy taxonomií jsou podobné. Všechny určují možnost šíření další informace po kategorickém výsledku vyšetření nějakého uzlu. Opět je určen strom taxonomií, přičemž každá třída může obsahovat libovolné uzly. U prvního typu, šíření kladné odpovědi (hidden yes), dochází k šíření kategorické odpovědi *ano* směrem k nadtřídám (tj. pokud něco určité platí pro nějakou třídu, pak to platí i pro jejich nadtřídy²). Naopak u druhého typu, šíření záporné odpovědi (hidden no), dochází k šíření kategorické odpovědi *ne* směrem k podtřídám (pokud něco neplatí pro třídu, pak to neplatí i pro její podtřídy³). Poslední typ je kombinací předchozích dvou, dochází zde k šíření kladné kategorické odpovědi směrem k nadtřídám a záporné kategorické odpovědi směrem k podtřídám.

Taxonomické stromy by měly být konzistentní s inferenční sítí, viz [6].

¹ Názvy taxonomií jsou inspirovány DOSovskou verzí FEL-Experta.

² Např. pokud *jistě* víme, že *je* objekt šelma kočkovitá, pak je i savec, zvíře atd.

³ Např. pokud *jistě* víme, že objekt *není* šelma kočkovitá, pak není ani kočka, rys, jaguár atd.

4. Získávání dat

Při konzultaci je třeba získávat data z okolního světa. Tato data mohou být v nejrůznější podobě, je však třeba vždy zajistit jejich převedení na pravděpodobnost, která se použije pro inferenci.

4.1 Obecný popis

Systém používá tzv. získávače dat (retriever) a překladače dat (translator)⁴. Každý dotazovatelný uzel má přiřazeny tyto objekty.

V okamžiku, kdy je třeba se dotázat na dané tvrzení, uvede se do chodu získávač dat, který získá z okolí (uživatel, databáze, měřicí karta, externí program...). Každý uzel může mít takovýchto získávačů uvedeno více, pokud selže první (např. při chybě připojení k databázi), volá se další. Pokud i poslední získávač selže, inference se ukončí chybou. Získávače se dají sloučit dle typu dat (číslo, interval, a cokoliv jiného), kterým odpoví. Jeden uzel musí mít všechny získávače kompatibilní vzhledem k typu dat.

Překladač je určen typem dat a tedy typem získávače. Obdrží získaná data a přeloží je na pravděpodobnost, která se použije pro inferenci. Překladač je pouze jeden pro každý uzel.

Podobně je třeba získat z prostředí informace o taxonomiích. Ve stávajícím systému je získávač pro taxonomie pouze jeden a nelze ho tedy volit.

4.2 Implementované metody

Získávače a překladače se dají vytvářet v neomezeném množství, závisí pouze na zamýšlené aplikaci. Zde stručně popíšeme metody získávání a překladu dat v implementovaném systému. Vždy je popsán vstup z prostředí Windows a vstup z databáze. Lze též uvést získávač, který vždy odpoví hodnotu „nevím“. Ten lze použít například jako poslední možnost, pokud nechceme ukončit inferenci chybou při chybě předchozích získávačů.

Názvy získávačů jsou vždy ve tvaru <vstup><typ>, kde <vstup> udává odkud se bude hodnota získávat (např. win pro uživatele ve Windows, db pro databázi) a <typ> udává typ hodnoty (typ uzlu, např. prob pro pravděpodobnost, cf pro činitel jistoty apod.).

Popis přístupu k databázím je uveden níže.

4.2.1 Pravděpodobnost

Od uživatele se získá přímo hodnota pravděpodobnosti tvrzení, překlad je zde triviální. Při použití databáze musí být vstupem číselná hodnota pravděpodobnosti. Pokud je hodnota v databázi *NULL* nebo *?*, uvažuje se vstup „nevím“, tj. kopíruje se apriorní pravděpodobnost. Při jiné hodnotě (např. nečíselné) nastává chyba.

Získávač pro pravděpodobnost získanou od uživatele ve Windows se jmenuje winprob, získávač pro databázi se jmenuje dbprob, získávač „nevím“ má název dnprob.

⁴ Na rozdíl od DOSovské verze FEL-Experta. Zde tedy odpadávají např. tzv. S a Q uzly a velmi se tak zjednodušuje přidávání dalších "typů" uzlů, neboť není nutný zásah do inferenční sítě.

Question - probability

Enter the probability for node

there is a battery in the car

Probability: 0.25

Don't know Accept

Obrázek 1 – vstup pravděpodobnosti

4.2.2 Činitel jistoty

Uživatel může odpovědět *ano/ne/nevím* a případně pomocí dalšího jemnějšího rozlišení mezi těmito hodnotami. Hodnota *ano* se přeloží jako pravděpodobnost 1, hodnota *ne* jako 0, hodnota *nevím* jako apriorní pravděpodobnost tvrzení. Zbylé hodnoty se určí lineární interpolací.

Při vstupu z databáze se číselné hodnoty uvažují jako činitele jistoty od -5 do 5 . Hodnota *NULL* nebo *?* znamená „nevím“, hodnoty *y* a *Y* znamenají *Yes* (činitel jistoty 5), hodnoty *n* a *N* znamenají *No* (činitel jistoty -5).

Názvy získávačů jsou postupně (s významem jako výše) *wincf*, *dbcf*, *dncf* (*cf* = *certainty factor*).

Question - yes/no

How certain are you, that

there is a battery in the car

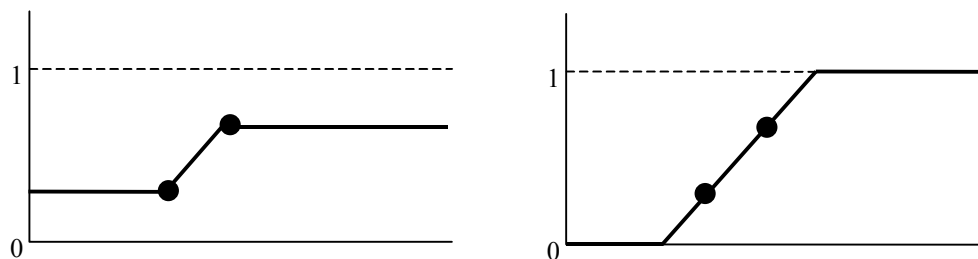
No Don't know Yes

Obrázek 2 – vstup činitele jistoty

4.2.3 Hodnota

Zde uživatel odpovídá konkrétní numerickou hodnotou pozorované veličiny. Expert zadá významné body a jejich odpovídající pravděpodobnosti. Hodnota od uživatele se na pravděpodobnost přepočítá interpolací (příp. extrapolací) mezi body zadanými expertem. K dispozici jsou dva druhy tohoto uzlu, které se liší pouze v počítání pravděpodobnosti mimo zadané meze. Jeden (*winvalue* případně *wins*) počítá hodnoty mimo meze jako hodnoty

mezi, druhý (winvaluex) provádí lineární extrapolaci dle krajních dvou bodů až do hodnoty 0 nebo 1 (viz obrázek).



Obrázek 3 – porovnání přepočtu pravděpodobnosti (wins a winsx)

Při použití vstupu z databáze se načítá číselná hodnota, ? opět znamená „nevím“. Názvy získávačů jsou postupně winvalue (wins), dbvalue (dbs) a dnvalue (dns) pro první typ a winvaluex (winsx), dbvaluex (dbsx) a dnvaluex (dnsx) pro druhý typ.

The image shows a window titled 'Question - exact value'. Inside, there's a label 'Enter exact value...' above a text area containing the question 'what is the temperature?'. Below the text area, there's a label 'Value:' followed by a text input field containing the number '67'. At the bottom, there are two buttons: 'Don't know' and 'Accept'.

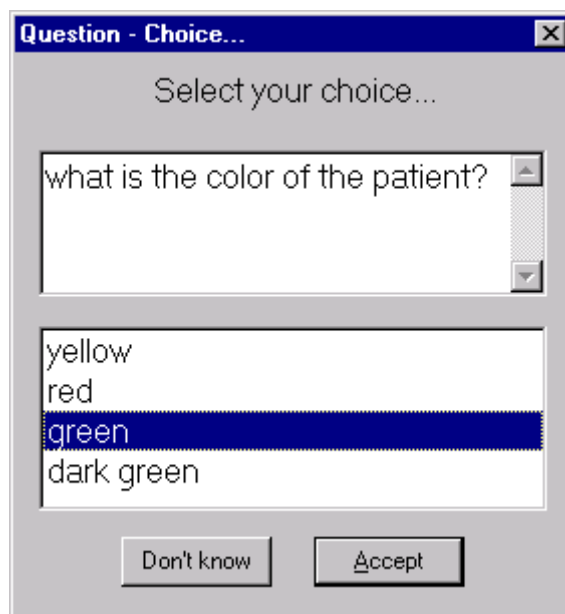
Obrázek 4 – vstup přesné hodnoty

4.2.4 Výběr z možností

Uživatel odpovídá výběrem z nabídnutých možností. Expert zadává možnosti a odpovídající subjektivní pravděpodobnosti, dle kterých se vyhodnotí odpovědi uživatele.

V databázi je tato odpověď uložena buď jako pořadové číslo volby (1, 2, 3, ...) nebo jako písmeno volby (a, b, c, ...).

Názvy získávačů jsou postupně winchoice, dbchoice, dnchoice.



Obrázek 5 – výběr z možností

5. Textový tvar báze znalostí

Báze znalostí včetně všech podrobností týkajících se konzultace je popsána ve speciálním textovém tvaru, který je vysvětlen v této kapitole.

5.1 Příklad

Pro rychlý náhled uvedeme příklad:

```
/* příklad definice báze znalostí (inf. sítě) - pouze ilustrace bez
konkrétního významu */

nodes // definice uzlů
{
  je_benzin // jméno uzlu
  {
    'v nadrzi je benzin'
    // komentář, použito pro kladení otázek - nepovinné
    (bayesian, // bayesovský uzel
     0.5)      // apriorní pravděpodobnost
    retriever ( wincf )
    // získání dat od wincf (zdroj windows, certainty factor)
    /* pokud se neuvede retriever, považuje se uzel za
nedotazovatelný */
  }

  auto_ok
  {
    (bayesian, 0.6, goal) // goal = cílová hypotéza
  }

  logicky_uzel
  {
```



```

    (logical, and, goal ) // logicky, typ and
}

teplota
{
    (bayesian, 0.2)
    retriever (winvalue) // hodnota
    translator
    (
        ( 1, 2, 3, 4 ) // vrcholy
        ( 0.1, 0.5, 0.6, 0.8 ) // hodnoty ve vrcholech
    )
}

// ... a další uzly ...
}

rules // pravidla
{
    { je_benzin, auto_ok, 0.6, 0.1 } // pravidla a jejich váhy
        // tj. auto_ok platí s pstí 0.6, když je benzín
        // a platí s pstí 0.1, když není benzín
    { predpoklad1, logicky_uzel } // pravidlo k logickému uzlu
    // ... a další pravidla
}

links // vazby
{
    { je_volant, je_benzin } // prioritní link (nepodminena vazba)
    { je_baterie, je_volant, 0.1, 0.2 } /* kontextová vazba */
    // ...
}

taxonomies // taxonomie
{
    taxonomy2 hypothesis // dotazovatelné
    {
        taxonomy2.1 (hypo21)
        {
            taxonomy2.1.1 (hypo211)
            taxonomy2.1.2 (hypo212a, hypo212b)
        }
        taxonomy2.2 (hypo22)
        taxonomy2.3 (hypo23)
    }
    taxonomy3 hidden_yes // šíření „ano“
    {
        taxonomy3.1 (hypo31)
    }
    taxonomy4 hidden_no // šíření „ne“
    {
        taxonomy4.1 (hypo41)
    }
}

```

5.2 Syntaxe a sémantika

Následuje syntaktický popis v Backus-Naurově formě.

Vysvětlivky: *neterminální symbol*, **terminální symbol**, ::= - pravidlový přechod,
| - alternativa, [...] - možnost výskytu, [...]* - možnost výskytu s opakováním

```
báze_znalostí ::= [ vlastnosti_sítě ] definice_struktury
definice_struktury ::= uzly definice_struktury
                    | pravidla_definice_struktury
                    | vazby_definice_struktury
                    | taxonomie_definice_struktury
                    | popis_definice_struktury
                    | ε
vlastnosti_sítě ::= net { [ parametr_sítě ]* }
parametr_sítě ::= defretr ( identifikátor mc identifikátor )
                // definice získávače (synonymum) – první parametr je název nového
                | setparam ( identifikátor mc [ identifikátor ] )
                // nastavení hodnoty parametru – název, hodnota
                | askparam ( identifikátor )
                // na hodnotu parametru se zeptá systém při startu konzultace
uzly ::= nodes { [ uzel ]* }
uzel ::= identifikátor { vlastnosti_uzlu }
vlastnosti_uzlu ::= [ otázka ] typ_uzlu [ vstup_dat ] [ překlad_dat ]
otázka ::= ‘ bližší popis uzlu pro otázku ‘ // tázaný text v apostrofech
typ_uzlu ::= ( logical , log_typ [ , goal ] )
            | ( bayesian , pst [ , goal ] )
log_typ ::= and | or | not
vstup_dat ::= retriever ( dvstup [ dvstup ]* ) // pořadí udává prioritu
dvstup ::= identifikátor [ ( další_položky_určené_typem_získávání_dat ) ]
překlad_dat ::= translator ( další_položky_určené_typem_získávání_dat )
pravidla ::= rules { [ pravidlo ]* }
pravidlo ::= { identifikátor mc identifikátor } // k logickému uzlu
            | { identifikátor mc identifikátor mc pst mc pst } // k bayesovskému uzlu
vazby ::= links { [ vazba ]* }
vazba ::= { identifikátor mc identifikátor } // prioritní vazba
        // první uzel musí být vyšetřený před druhým uzlem
        | { identifikátor mc identifikátor mc pst mc pst } // kontextová vazba
        // pravděpodobnost prvního uzlu musí být před vyšetřováním druhého uzlu
        // v zadaných mezích (pokud ne, snaží se systém splnit kontext jeho
        // vyšetřením, pokud se splnit nepovede, druhý uzel se nevyšetřuje)
taxonomie ::= taxonomies { [ taxonomiet ]* }
taxonomiet ::= identifikátor typ_taxonomie [ uzly_taxonomie ] [ podtřídy ]
typ_taxonomie ::= hypothesis | hidden_yes | hidden_no | hidden_both
uzly_taxonomie ::= ( [ identifikátor ] [ mc identifikátor ]* )
podtřídy ::= { [ podtax ]* }
podtax ::= identifikátor [ uzly_taxonomie ] [ podtřídy ]
popis ::= description { [ popis_uzlu ]* }
popis_uzlu ::= identifikátor { text_popisu_bez_znaku_‘_’ } // identifikátor je název uzlu
```

```

mc ::= [ , ]                                // možná (ne nutná) čárka
identifikátor ::= posloupnost znaků bez mezer
                    | “posloupnost znaků s mezerami“
pst ::= reálné číslo z intervalu <0,1>      // pravděpodobnost
číslo ::= reálné číslo                      // pro účely parametrů překladačů

oddělovač ::= mezera
                    | nový_řádek
                    | // komentář... nový_řádek
                    | /* komentář... */
                    | oddělovač [ oddělovač ]*

```

Sémantika by měla být zřejmá z názvu symbolů a doplňujících komentářů. K distribuci programu jsou přiloženy funkční příklady, které dále osvětlují význam zápisu.

5.3 Parametry objektů pro získávání dat

V současnosti jsou implementovány objekty pro překlad dat *cf*, *prob*, *value* (*wins*), a *choice*. Z překladačů očekávají parametry překladače *value* a *choice*. Ze získávačů očekává parametry pouze získávač pro volbu.

5.3.1 Parametry pro vstup hodnoty

Syntaxe popisu překladače pro hodnotu (*value*):

```

parametry ::= vrcholy hodnoty
vrcholy ::= ( číslo číslo [ číslo ]* )
hodnoty ::= ( pst pst [ pst ]* )    // počet vrcholů a hodnot musí být stejný

```

Příklad:

```

leaf
{
    (bayesian, 0.2)
    retriever (wins)
    translator ( ( 1, 2, 3 ) ( 0.1, 0.5, 0.6 ) )
}

```

5.3.2 Parametry pro vstup volby

Syntaxe popisu překladače pro volbu (*choice*):

```

parametry ::= ( pst [ pst ]* )
                // pravděpodobnosti odpovídající volbám (dle pořadí uvedeného v získávači)

```

Syntaxe popisu získávače pro volbu (*choice*):

```

parametry ::= ( volba [ volba ]* )
volba ::= ( identifikátor )           // popis volby (tak jak se ukáže uživateli)

```

Příklad:

```

leaf
{
    (bayesian, 0.5)
    retriever (winchoice ("první volba", druhá třetí))
    translator ( 0.1, 0.5, 0.9 )
}

```

5.4 Parametry sítě

V parametrech sítě (oblast `net {...}`) jsou uvedeny parametry společné pro celou síť. Navíc lze definovat synonyma pro získávače dat.

Parametr lze definovat pomocí `setfparam` nebo `askparam`. Pomocí `setparam` se parametr napevno definuje, například `setparam(parametr, hodnota)`. Pokud se hodnota neuvede, dosadí se za ní prázdný řetězec. Pomocí `askparam` lze říci, že se má systém na hodnotu parametru zeptat před začátkem konzultace. Hodnota se zde samozřejmě neuvádí, například tedy `askparam(jmeno_parametru)`.

Synonymum pro získávač se definuje pomocí `defretr`, například `defretr(cf, wincf)` definuje nový získávač `cf`, který znamená to samé co `wincf`. Pak lze v celé bázi uvádět `cf` místo `wincf`. Výhoda tohoto přístupu je například při konverzi celé sítě na jiný způsob získávání dat. Pokud máme odladěnou bázi znalostí, kde jsou data získávána od uživatele, stačí v tomto případě přepsat definici synonyma (spolu s ostatními) na `defretr(cf, dbcf)` a data pro inferenci budou čtena z databáze.

6. Práce se systémem

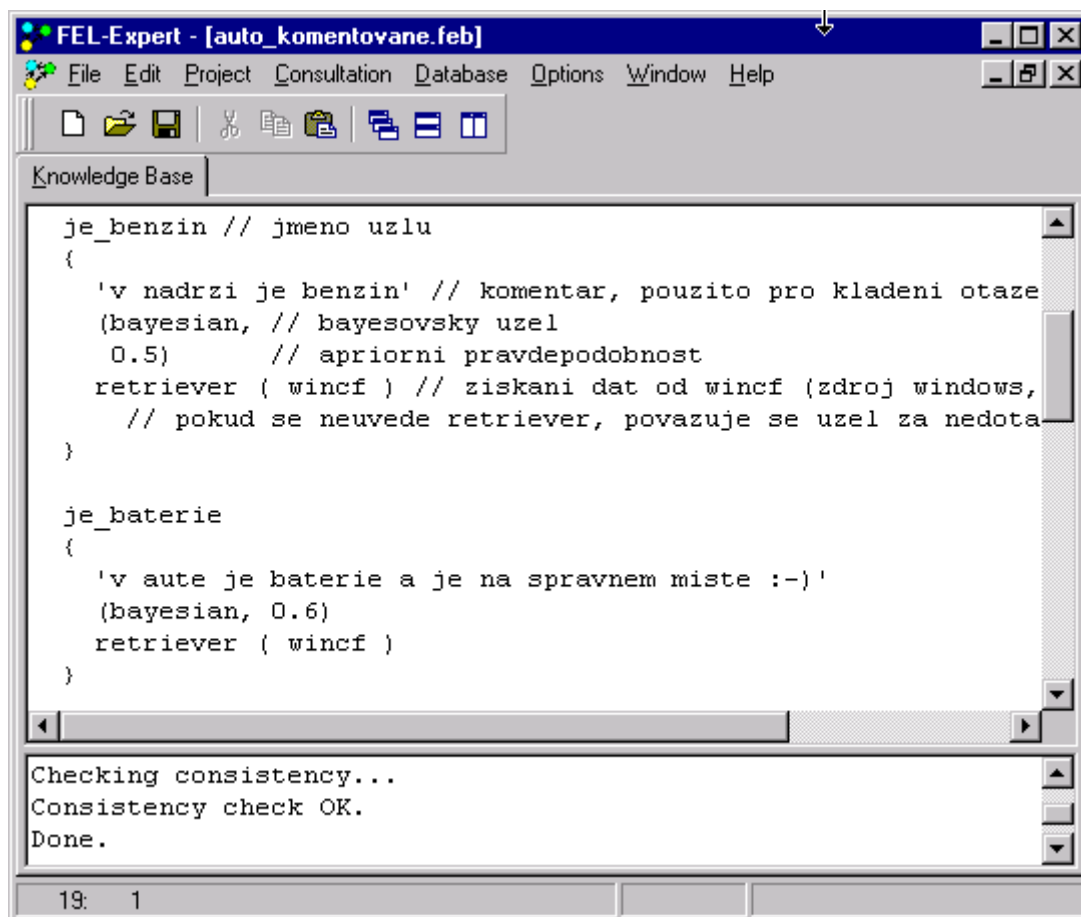
V této kapitole je popsána práce se systémem z hlediska uživatele, tj. vytváření báze znalostí, vytváření inferenční sítě a konzultace.

6.1 Obecné informace

Program je typu MDI, tj. lze mít najednou otevřeno více dokumentů, tj. lze mít otevřeno víceází znalostí nebo více konzultačních oken (v jednom okamžiku může být spuštěna pouze jedna konzultace). Bázi znalostí lze vytvořit pomocí položky menu `File->Open`, nový dokument lze vytvořit pomocí `File->New`. Rozložení oken lze spravovat pomocí menu `Window`.

6.2 Editor báze znalostí

Po otevření báze znalostí se vytvoří okno s editorem. Pro editaci lze použít standardní prostředky MS Windows, včetně práce se schránkou. Používané příkazy jsou dostupné z menu `Edit`. Tvar báze znalostí je popsán výše.



Obrázek 6 – editor báze znalostí

6.3 Vytvoření inferenční sítě

Inferenční síť se vytvoří pomocí Project->Build net. Po zvolení se otevře informační okno, ve kterém lze nalézt informace o úspěšnosti překladu textového tvaru báze znalostí do vnitřní reprezentace inferenční sítě. Toto okno se dá zavřít pomocí kontextového menu (pravým tlačítkem myši).

Pokud nebyl překlad úspěšný, vypíše se příčina chyby a kurzor v editačním okně je přemístěn za místo této chyby. Po úspěšném překladu lze spustit konzultaci.

6.4 Konzultace

Konzultace se spustí pomocí Consultation->Consult. Pokud je již vytvořená (přeložená) inferenční síť a nedošlo ke změně báze znalostí, použije se tato síť pro konzultaci. Jinak se nejprve automaticky spustí vytvoření inferenční sítě. Konzultaci je možné mít spuštěnou pouze jednu v rámci celé aplikace.

Pokud je vytvořena síť, otevře se konzultační okno a začne konzultace. Pokud jsou definovány dotazovatelné taxonomie, zeptá se systém nejprve na ně. Poté jsou vyšetřovány jednotlivá dotazovatelná tvrzení. Konzultaci lze přerušit pomocí Consultation->Abort.

Během konzultace se zobrazují informace o jejím průběhu. Je zobrazen celkový status, vyšetřovaný uzel a vyšetřovaná cílová hypotéza. Dále jsou zobrazeny uspořádané cílové

hypotézy (jen ty, které budou nebo jsou vyšetřovány v rámci dané taxonomie) , všechny uzly sítě a průběh, jak se došlo k dotazovanému uzlu (Call Stack). Zobrazování všech těchto informací může být pro velké inferenční sítě časově náročné, proto je možné je vypnout. Zapnutí nebo vypnutí zobrazování těchto informací pro celý systém se dá zvolit z menu Options (viz dále).

Status: **Investigating...**
Investigated goal: **auto_ok** Investigated node: **je_benzin**

☒ Ordered goals ☒ Call stack ☒ All nodes without goals

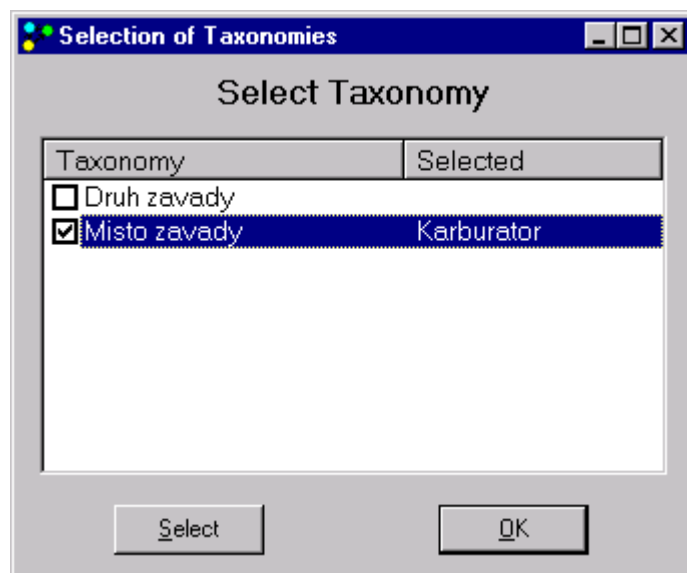
Goal	Aprior	Actual	Node name	Node	Aprior	Actual
do_servisu	0.1000	0.9986	auto_ok	je_benzin	0.5	0.5
natankuj	0.5	0.5	je_benzin	je_baterie	0.6000	0
auto_ok	0.6000	0.0081		je_volant	0.8999	0

Obrázek 7 – informace zobrazované během konzultace

Po skončení konzultace se ohlásí nejpravděpodobnější cílová hypotéza a zůstane zobrazen stav sítě včetně uspořádaných cílových hypotéz. Dále lze opět spouštět konzultaci nad stejnou inferenční sítí pomocí Consultation->Consult.

6.4.1 Vstup taxonomie

Pokud jsou definovány dotazovatelné taxonomie, objeví se před vlastní konzultací výzva k volbě taxonomií.



Obrázek 8 – výběr taxonomie

Po vybrání taxonomie (tlačítkem Select nebo zaškrtnutím taxonomie nebo dvojkliknutím myši) lze vybrat konkrétní taxonomickou třídu, která se pak ukáže vpravo od dané taxonomie. Výběr lze zrušit odškrtnutím taxonomie.



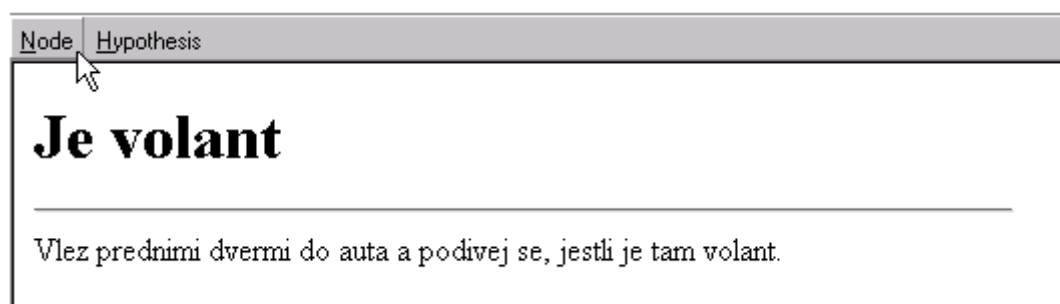
Obrázek 9 – výběr taxonomické třídy

Třidu lze vybrat vybráním myši a tlačítkem OK nebo dvojkliknutím myši. Tlačítko Select None způsobí zrušení výběru této taxonomie.

6.5 Popis uzlů

Stávající systém nabízí podporu zobrazování popisu uzlů v HTML. Popis se zapisuje do stejného souboru jako báze znalostí pod částí *description*. V popisu lze použít formátování textu dle HTML, tj. např. tvoření odstavců, tučné písmo, odkazy na jiné uzly, vkládání obrázků apod. Po přeložení popis zůstává k dispozici pro použití při konzultaci.

Takto vytvořený popis se dá vyvolat při konzultaci v dolní části konzultačního okna. Proporci obou částí oken lze měnit. Tlačítkem Node se vyvolá popis dotazovaného uzlu, tlačítkem Hypothesis se vyvolá popis vyšetřované cílové hypotézy.



Obrázek 10 – ukázka popisu

6.6 Další možnosti

Možnosti chování systému je možné zvolit v menu Options.

Položka Options->Environment vyvolá dialog pro určení vlastností celého prostředí systému. Pod záložkou Startup lze zvolit, zda se při spuštění programu automaticky otevře prázdný editor (Create new file on startup) a zda se budou nově vytvořená okna maximalizovat (Maximize new windows). Pod záložkou Configuration lze zvolit, zda se nastavení systému po ukončení programu uloží do systémových registrů (Save options on exit), případně lze v systému zaregistrovat soubory .feb (báze znalostí) – např. po změně umístění programu (tlačítko Register .feb files).

Pod položkou Options->Consultation lze zvolit možnosti chování při konzultaci – jaké informace se budou vypisovat (viz výše).

Položka Options->View Toolbars umožní ukázat nebo schovat ovládací panel.

7. Práce s databázemi

Současná implementace systému umožňuje pracovat s různými relačními databázemi. Přístup k databázím je zajištěn pomocí BDE (Borland Database Engine). Lze tak přistupovat k lokálním souborovým databázím (Paradox, dBASE, FoxPro, Access) nebo k databázovým serverům (Interbase, Oracle, Sybase, Informix, Microsoft SQL Server, DB2). Je též umožněno přistupovat k databázím přes ODBC, což zajišťuje přístup prakticky k jakékoliv databázi. Pro přístup k databázovým serverům je třeba nastavit alias BDE, pro lokální souborově orientované databáze to není třeba (viz níže).

Zvlášť je též implementován přístup k jednoduchým textovým databázím.

Databáze se dají z pohledu systému FEL-Expert rozdělit na vstupní a výstupní (může se však jednat stále o jednu databázi). Vstupní databáze poskytuje hodnoty pro konzultaci, výstupní databáze slouží pro uchování výsledků konzultace.

7.1 Struktura textových databází

Textové databáze slouží především pro jednoduché čtení a ukládání dat. Existují ve čtyřech variantách, které se liší strukturou a oddělovacími znaky. Možné struktury jsou jednoduchý soubor a relační databáze, oddělovací znaky mohou být tabelátory nebo čárky.

Jednoduchý soubor má tvar

```
<název uzlu><oddělovač><hodnota uzlu>  
<název uzlu><oddělovač><hodnota uzlu>  
<název uzlu><oddělovač><hodnota uzlu>
```

...

Textový soubor s „tabulkou relační databáze“ má strukturu

```
<název sloupce><oddělovač>[<název sloupce><oddělovač>]*  
<hodnota><oddělovač>[<hodnota><oddělovač>]*  
<hodnota><oddělovač>[<hodnota><oddělovač>]*
```

...

tj. struktura vyjadřuje strukturu relační databáze. Název sloupce definuje jméno uzlu, řádek se vybírá dle stejných pravidel jako u ostatních relačních databází (viz níže).

7.2 Struktura relačních databází

Tabulka relační databáze má obecně následující strukturu:

název1	název2	název3	...	názevN
hodn11	hodn12	hodn13		hodn1N
hodn21	hodn22	hodn23		hodn2N
...

Hlavička (jména sloupců) definuje jméno uzlu, ke kterému se daný sloupec vztahuje. Řádek, který se použije v dané konzultaci, se vybere dle hodnoty daného sloupce. Například pokud má tabulka strukturu

header	uzelA	uzelB	uzelC	uzelD
prvni	0,2	0,5	0,0	0,6
druhy	0,3	0,7	0,1	0,8
treti	0,4	0,9	0,1	0,9
...

Lze použít pro výběr sloupec s názvem "header". Pak lze například specifikovat, že se pro konzultaci použijí hodnoty z řádku, kde je ve sloupci "header" hodnota "druhy".

Jméno sloupce, který určuje výběr řádku je implicitně "header". Jméno lze změnit pomocí parametru `dbheadername` (tj. např. pomocí nastavení v definici sítě `setparam(dbheadername, rec_header)`). Hodnota v řádku s požadovanými hodnotami se určí pomocí parametru `dbrecord` (tj. např. `setparam(dbrecord, druhy)`).

7.3 Použití vstupní databáze

Vstupní databáze slouží pro vstup hodnot do konzultace. Může být pouze jedna⁵. Databáze se určí opět pomocí parametrů. Konkrétní databáze se nastaví dle následující tabulky:

Typ databáze	Určení
Jednoduchá textová databáze, oddělení čárkami	<code>setparam(idbtxtcomma, dbdn.txt)</code> // <code>dbdn.txt</code> je soubor
Jednoduchá textová databáze, oddělení tabelátory	<code>setparam(idbtxttab, dbdn.txt)</code>
Relační textová databáze, oddělení čárkami	<code>setparam(idbtxtrelcomma, dbdn.txt)</code> <code>setparam(idbrecord, t2) // řádek</code>
Relační textová databáze, oddělení tabelátory	<code>setparam(idbtxtreltab, dbdn.txt)</code> <code>setparam(idbrecord, t2)</code>
Databáze BDE	<code>setparam(idbbde, IBFelex)</code> // <code>IBFelex</code> je alias BDE <code>setparam(idbbdetable, autol)</code> // <code>autol</code> je tabulka <code>setparam(idbrecord, treti) // řádek</code>

⁵ Systém postupuje tak, že se snaží inicializovat zadané databáze (tj. zkouší načíst tabulku) a použije první databázi, u které se mu to podaří.

S BDE lze použít i databázi bez nastaveného aliasu, v tom případě se místo aliasu uveďte adresář a místo tabulky jméno souboru s tabulkou. Pokud je pro přístup k databázi třeba autorizace, zeptá se systém na jméno a na heslo.

Před konzultací se do paměti načte daný řádek, přičemž dle názvů sloupců je určeno, pro jaký uzel se daná hodnota použije při konzultaci. Nevadí, pokud jsou v tabulce nepoužité sloupce. Další chování (například ptaní se uživatele při neúspěchu čtení hodnoty z databáze nebo uvažování neúspěchu jako “nevím”) lze definovat pomocí pořadí získávačů u každého uzlu.

7.4 Použití výstupní databáze

Výstupní databáze slouží pro výstup hodnot z konzultace. Výstupních databází může být obecně více (systém pak zapisuje do všech definovaných). Konkrétní databáze se nastaví dle následující tabulky:

Typ databáze	Určení
Jednoduchá textová databáze, oddělení čárkami	setparam(odbtxtcomma, dbdno.txt) // dbdn.txt je soubor
Jednoduchá textová databáze, oddělení tabelátory	setparam(odbtxttab, dbdno.txt)
Relační textová databáze, oddělení čárkami	setparam(odbtxtrelcomma, dbdno.txt) setparam(odbreCORD, t2) // řádek
Relační textová databáze, oddělení tabelátory	setparam(odbtxtreltab, dbdn.txt) setparam(odbreCORD, t2)
Databáze BDE	setparam(odbbde, IBFelex) // IBFelex je alias BDE setparam(odbbdetable, autoo) // auto1 je tabulka setparam(odbreCORD, treti) // řádek

Pro databáze BDE platí stejná pravidla jako u vstupních databází.

V případě jednoduché textové databáze se vypíše všechny cílové hypotézy. U ostatních databází lze určit, jaké uzly nás zajímají – podle názvů sloupců se určí jméno uzlu a jeho aposteriorní hodnota se zapíše do databáze.

7.5 Dávkové zpracování

Dávkové zpracování umožňuje provést konzultaci části tabulky nebo celé tabulky. Při tomto zpracování se zpracovávají postupně všechny řádky tabulky a pro každý řádek proběhne konzultace, jejíž výsledky se zapíše do výstupní databáze.

U databáze obsluhované pomocí BDE lze nastavit část tabulky, která se bude konzultovat pomocí volitelné hodnoty parametru batchidbbde. Tato hodnota má tvar například “header=’prvni’”. Obecně lze dodat jakoukoliv část SQL, která se uvede za klauzuli WHERE příkazu SELECT, tj. dotaz se vytvoří jako

```
SELECT * FROM tablename WHERE zadaný_filtr
```

Pokud se filtr neuvede, provede se dávkové zpracování celé tabulky.

Dávkové zpracování se určí nastavením příslušného parametru dle následující tabulky:

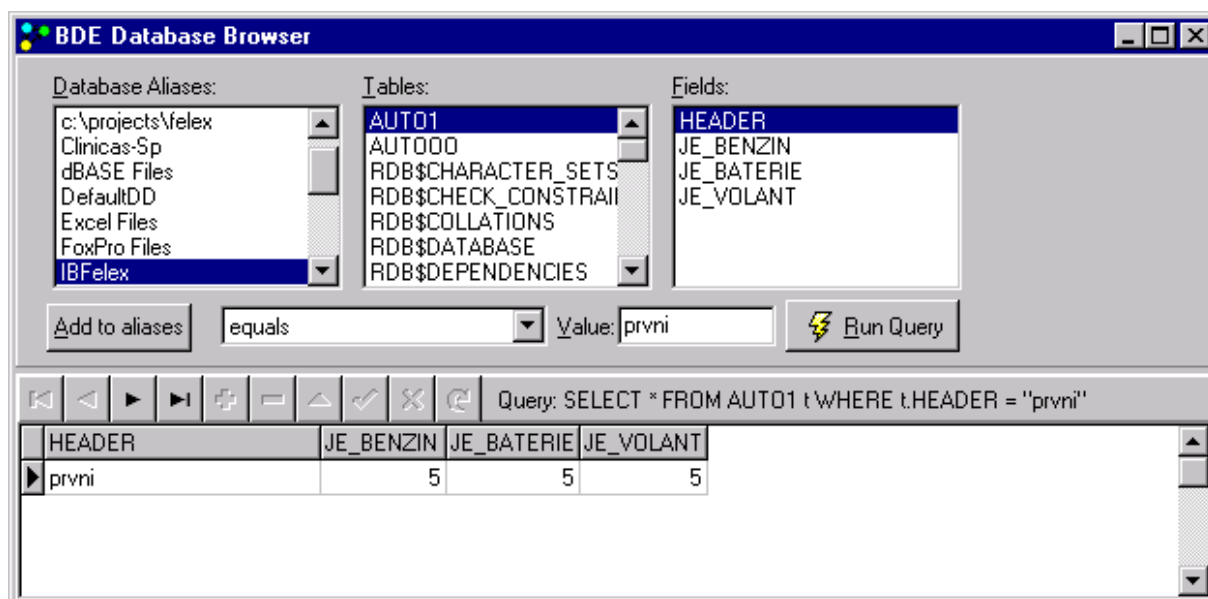
Typ databáze	Určení
Jednoduchá textová databáze, oddělení čárkami	setparam(batchidbtxtcomma)
Jednoduchá textová databáze, oddělení tabelátory	setparam(batchidbtxttab)
Relační textová databáze, oddělení čárkami	setparam(batchidbtxtrelcomma)
Relační textová databáze, oddělení tabelátory	setparam(batchidbtxtreltab)
Databáze BDE	setparam(batchidbbde) nebo setparam(batchidbbde, filtr)

V tomto případě nemá smysl nastavovat parametry dbrecord protože jsou nastavovány systémem automaticky (případné nastavení je ignorováno).

Opět, jako u vstupní databáze, lze další chování (například ptaní se uživatele při neúspěchu čtení hodnoty z databáze nebo uvažování neúspěchu jako “nevím”) definovat pomocí pořadí získávačů u každého uzlu (pokud se nepovede zpracování některého řádku, přeruší se celá dávková konzultace, proto lze nastavení získávačů doporučit).

7.6 Prohlížeč databází

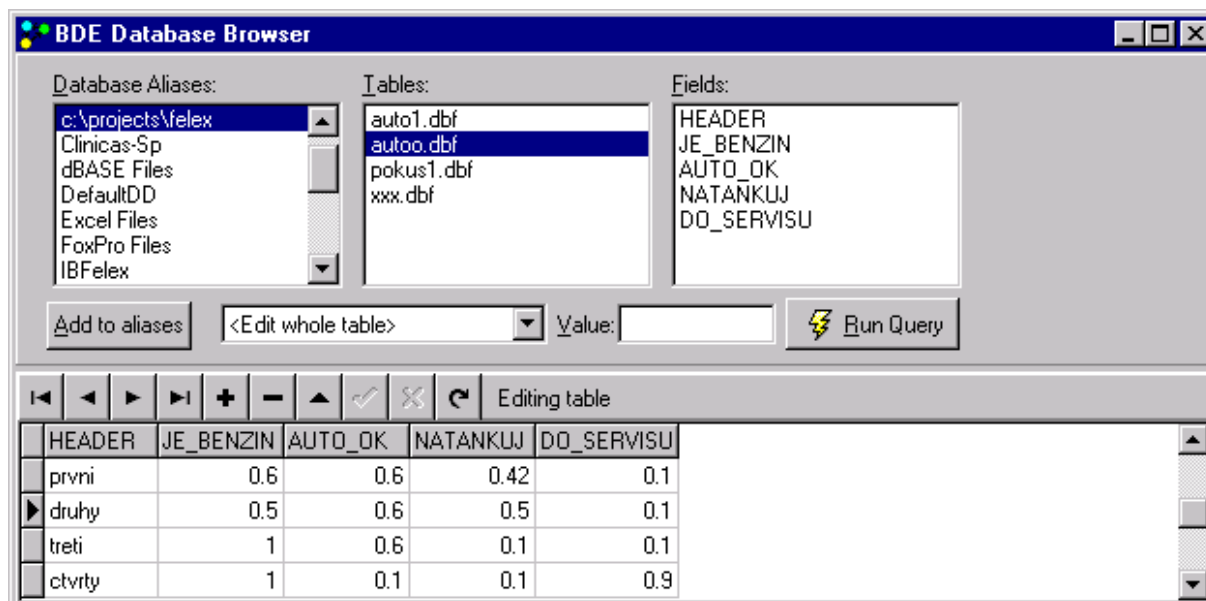
Součástí systému je jednoduchý prohlížeč (a editor) databází. Lze ho použít pro jednodušší práci s databázovými tabulkami, ke kterým lze přistupovat přes BDE. Vyvolat ho lze pomocí menu Database->Browse BDE Tables.



Obrázek 11 – výsledek dotazu do databáze

Prohlížeč načte definované databázové aliasy. Po vybrání aliasu databáze se načtou tabulky v dané databázi. Ty lze prohlížet podle zadaných kritérií – v položce Fields se vybere sloupec a zadá se pro něj podmínka. Dotaz se spustí tlačítkem Run Query a ve spodní části okna se objeví výsledek. Výsledek dotazu lze pouze prohlížet, nelze ho editovat.

Pokud chceme tabulku editovat, musí se vybrat <Edit whole table>.



Obrázek 12 – editace tabulky databáze

Lze také prohlížet a editovat souborové databáze (např. dBase, Paradox apod) i pokud pro ně není v BDE definován alias. Postup je podobný jako při zadávání vstupních a výstupních databází bez aliasu. Tlačítkem **Add to aliases** přidáme adresář, ve kterém se daná databáze (tabulky) nachází a po jejím vybrání v seznamu aliasů se v seznamu **Tables** vypíší tabulky v nalezeném adresáři.

8. Konzolová verze systému

Výše uvedený popis byl zaměřen na systém FEL-Expert 4.0 ve verzi pro Windows. Existuje i přenositelná konzolová verze systému, kterou lze provozovat na prakticky jakémkoliv systému (Sun, Linux, DOS, Windows atd.). Tato verze komunikuje s uživatelem pouze pomocí konzole, báze znalostí se zadává z příkazové řádky.

Hlavní odlišnosti od verze pro Windows jsou tyto:

- přenositelná konzolová verze umí spolupracovat pouze s textovými databázemi (nepřenositelná konzolová verze umí spolupracovat i s BDE)
- získávače mají jméno ve tvaru `con<typ>`, tj. například místo `wincf` se používá `concf` (to lze snadno změnit pomocí `defretr`)
- zatím není implementován výběr taxonomie (ale bázi s ní lze použít, pouze se neprovádí výběr)

9. Shrnutí

Tato příručka stručně shrnuje chování systému FEL-Expert. Další informace ohledně algoritmů lze nalézt v literatuře a v nadcházejících popisech systému. Systém bude dále rozšiřován a proto jsou vítány náměty a připomínky.

10. Literatura

- [1] P. Berka, P. Jirků, J. Vejnarová.: *Expertní systémy*, VŠE Praha, 1998
- [2] R. O. Duda, P. E. Hart, N. J. Nilsson: *Subjective Bayesian Methods For Rule-Based Inference Systems*, 1976, in G. Shaffer, J. Pearl: *Readings in Uncertain Reasoning*, Morgan Kaufman Publishers, 1990
- [3] Vl. Mařík: *Využití metod umělé inteligence pro řešení diagnostických úloh*, disertační práce, FEL ČVUT, 1988
- [4] Vl. Mařík, O. Štěpánková, J. Lažanský a kol.: *Umělá inteligence 2*, Academia, 1997
- [5] Vl. Mařík, T. Vlček, Z. Kouba, J. Lažanský, L. Lhotská, O. Štěpánková: *Expert System FEL-EXPERT version 3.5, Description and Users's Manual*, TR-PRG-IEDS-6/92, FAW Linz-Hagenberg, Praha-Wien, 1992
- [6] Vl. Mařík, T. Vlček, L. Lhotská, R. Wagner, W. Retschitzegger: *Exploation of External Programs and Taxonomies in Rule-based Expert System*, TR-PRG-IEDS-12/93, FAW Linz-Hagenberg, Praha-Wien, 1993
- [7] ?: *SPEL-EXPERT 4.0*