

Nový expertní systém NEST

Petr Berka, Vladimír Laš, Vojtěch Svátek

FIS, VŠE nám. W. Churchilla 4,
130 67, Praha

berka@vse.cz, vladalas@hotmail.com, svatek@vse.cz

Abstrakt. Příspěvek popisuje koncepci a implementaci expertního systému NEST vyvíjeného na VŠE v Praze. Systém vychází z expertního systému SAK a navazuje tak na výzkum z poloviny 80. let. NEST tedy používá výroky (odvozené od binárních, nominálních a numerických atributů), pravidla (kompozicionální s neurčitostí i nekompozicionální bez neurčitosti), kontexty a integritní omezení. Odvozování je založeno na kombinovaném zpětném a přímém řetězení, pro práci s neurčitostí se nabízejí čtyři různé způsoby.

Klíčová slova: expertní systémy

1 Úvod

Všeobecně se soudí, že expertní systémy mají své nejlepší roky již za sebou. Na vrcholu popularity byly na přelomu 70. a 80. let, jakožto hmatatelný výsledek té etapy vývoje umělé inteligence, která se zaměřila na reprezentování a zpracovávání specializovaných znalostí získaných od experta. Klasickým příkladem systémů té doby, které pak ovlivnily řadu dalších tvůrců, jsou systémy MYCIN [8] nebo PROSPECTOR [2]. V zemích koruny České zájem o expertní systémy kulminoval v druhé polovině 80. let. Z tehdy vzniklých tuzemských systémů zmiňme například systém Equant [6], systémy řady FEL-Expert [7] a systém SAK [4].

Připomeňme, že expertní systém můžeme chápat jako inteligentní počítačový program, který užívá znalosti a inferenční procedury k řešení problémů, které jsou natolik obtížné, že pro své řešení vyžadují významnou lidskou expertízu [3]. Znalosti nezbytné k činnosti na této úrovni plus použitá inferenční procedura mohou být chápány jako model expertízy nejlepších praktiků v oboru. Za základní principy (klasických) expertních systémů bývá považováno oddělení znalostí a mechanismu pro jejich využívání, práce s neurčitostí, dialogový režim a vysvětlování.

Přestože výzkum v umělé inteligenci se mnohdy přesunul jinam (např. otázkou neurčitosti se dnes zabývá zejména oblast soft computing), některé principy expertních systémů založených na pravidlech našly své oživení v oblasti strojového učení a dobývání znalostí (indukce rozhodovacích pravidel z dat) i v oblasti zpracování informací z dokumentů a webu (tvorba pravidel pro automatickou extrakci informací z textů, případně „pravidlová vrstva“ sémantického webu). Tato skutečnost spolu s potřebami výuky a s potřebami grantových projektů nás vedla k tomu, že jsme “resuscitovali” specifikace expertního systému z druhé poloviny 90. let [9] a dovedli je do stadia implementace.

2 Expertní systém NEST

Expertní systém NEST (Nový Expertní SysTém) navazuje na expertní systém SAK (Systém Automatizovaných Konzultací) vyvinutý na VŠE v polovině 80. let [4]. Dědictví svého předchůdce pak NEST dále rozvíjí tak, jak si to vyžádala zkušenost s reálnými aplikacemi systému SAK. Změny nalezneme jak v prostředcích pro reprezentaci znalostí, tak v práci inferenčního mechanismu i ve způsobech provádění konzultace. Nová je samozřejmě i implementační podoba systému – od DOSu a jazyka Basic jsme přešli k Windows a vývojovému prostředí Delphi.

2.1 Reprezentace znalostí

Atributy a výroky. Atributy slouží k popsání vlastností konzultovaných případů i prostředí ve kterém konzultace probíhá. Toto explicitní rozlišení na dva typy atributů v bázi znalostí je dáno jejich “stálostí” pro více konzultací. Atributy případu popisují konkrétní konzultovaný případ (např. příjem žadatele o úvěr), atributy prostředí popisují širší kontext konzultací, který se nemusí měnit (např. úrokovou míru).

Atributy se používají k vytváření výroků; každý atribut má nějakou hodnotu, podle které se přiřazují váhy příslušným výroky. Systém NEST umožňuje pracovat s atributy binárními, nominálními a numerickými. Od binárních atributů je odvozen jediný výrok (a případně jeho negace). Od nominálních atributů jsou odvozeny výroky, které odpovídají přípustným hodnotám atributu (systém rozlišuje jednoduché nominální atributy – tyto atributy mohou při konzultaci získat pouze jednu hodnotu a množinové nominální atributy – tyto atributy mohou při konzultaci získat více hodnot). Od numerických atributů jsou odvozeny výroky, které odpovídají rozdělení oboru přípustných hodnot na fuzzy intervaly (používáme lichoběžníkový případně trojúhelníkový tvar charakteristické funkce).

Pravidla. IF-THEN pravidla představují základní prostředek pro odvozování v systému NEST. Předpoklad pravidla je tvořen disjunktivní normální formou (disjunkce konjunkcí) literálů (výroků nebo jejich negací), závěr pravidla je tvořen seznamem (konjunkcí) literálů. Systém NEST nabízí tři typy pravidel:

- *kompozicionální pravidla* jsou pravidla, kde každý závěr pravidla je doplněn váhou² pravidla, která zhruba řečeno vyjadřuje do jaké míry tento závěr platí, pokud předpoklad platí s jistotou. V průběhu konzultace se skládají dílčí příspěvky pravidel, která vedou ke stejnému závěru – proto označení kompozicionální,
- *apriorní pravidla* jsou kompozicionální pravidla bez předpokladu; tato pravidla umožňují přiřadit výroky implicitní váhu,
- *logická pravidla* jsou nekompozicionální pravidla bez vah. Je-li předpoklad takového pravidla s jistotou splněn, váha závěru je známa rovněž s jistotou a závěr se považuje za plně vyhodnocený – v ostatních případech se pravidlo neaktivuje. Pouze logická pravidla mohou dát absolutní jistotu +1.

²V normovaném tvaru je váha pravidla číslo z intervalu [-1, 1].

S každým pravidlem může být asociován seznam akcí – samostatně spustitelných programů. Tyto akce se provedou, pokud je dané pravidlo aktivováno, tedy je-li splněn předpoklad pravidla. Pravidla mohou být rovněž doplněna o priority, které určují pořadí vyhodnocování pravidel.

Integritní omezení. Integritní omezení umožňují formulovat dodatečné podmínky na vztahy které by měly platit mezi literály po skončení konzultace. Obvykle se integritní omezení používají pro kontrolu logické konzistence průběhu a výsledků konzultace. Integritní omezení mají podobu implikací $ANT \Rightarrow SUC$ (*stupeň*), kde *ANT* i *SUC* jsou DNF literálů a *stupeň* je číslo vyjadřující důležitost integritního omezení. Příkladem integritního omezení může být

diagnoza(chřipka) \Rightarrow not diagnoza(zdráv),

nebo

velikost(velká) \Rightarrow not velikost(malá).

V prvním případě vyjadřuje integritní omezení vztah, který musí platit mezi cíli konzultace, ve druhém případě vyjadřuje integritní omezení vztah, který musí platit mezi dotazy.

Kontexty. Kontexty jsou, podobně jako předpoklady pravidel, DNF literálů, které (v případě kladné váhy) určují že je aplikovatelné nějaké pravidlo či integritní omezení. Kontexty se tedy používají pro členění báze znalostí (pravidel, integritních omezení) do relativně samostatných částí. Tak například pravidla týkající se problémů v těhotenství má smysl vyhodnocovat pouze v kontextu toho, že pacient je žena.

Kontexty tedy slouží k urychlení konzultace (umožňují „oříznout“ v daných souvislostech „nepotřebné“ části báze, takže systém klade méně dotazů), ale namají přímý vliv na odvození vah cílů („nepotřebná“ pravidla se stejně nebudou aktivovat).

Báze znalostí má podobu XML dokumentu. Pro zápis pravidel v podobě XML existuje několik způsobů, které se pokoušejí prosadit jako standard. Je to zejména PMML [11] snažící se standardizovat způsob zápisu modelů získaných metodami strojového učení (současná verze 2.0 definuje zápis regresních modelů, shlukování, rozhodovacích stromů, neuronových sítí, bayesovského klasifikátoru a asocičních pravidel) a RuleML [12] zaměřený výhradně na pravidla. Bohužel ani jeden z formalismů nemůžeme přímo použít. Ani PMML ani RuleML neumožňuje zapsat pravidlo s neurčitostí, o kontextech, prioritách a akcích ani nemluvě. Navíc v bázích znalostí systému NEST hrají roli i další informace (různé typy atributů, volba inferenčního mechanismu apod.). Nevyhovuje ani syntaxe vlastního pravidla – předpoklad pravidla je v PMML i v RuleML tvořen jednou konjunkcí, závěr jedním atomem. Úpravy obou formalismů jsou samozřejmě možné (slibnější a hodnější pozornosti se zdá být RuleML), ale byly by natolik zásadní, že jsme zatím šli cestou definování vlastního DTD „od nuly“. Příklad zápisu pravidla v syntaxi použité v systému NEST ukazuje obrázek 1. Souhlasíme ale s tím, že standardizace zápisu bází znalostí umožní jejich snadnou přenositelnost a chceme tedy tento směr dále sledovat.

2.2 Inferenční mechanismus

Odvozování v síti pravidel. Použité typy pravidel nabízejí dva možné způsoby odvozování:

- *kompozicionální odvozování* v rámci kompozicionálních a apriorních pravidel. V průběhu konzultace se skládají dílčí příspěvky všech pravidel vedoucích ke stejnému závěru. Někaké pravidlo je pak aktivováno a dá nenulový příspěvek k váze závěru, pokud (1) kontext pravidla (je-li zadán) je odvozen s kladnou vahou a rovněž (2) předpoklad pravidla je odvozen s kladnou vahou – kontext (je-li zadán) přitom modifikuje váhu pravidla. Jde tedy o způsob odvozování známý ze systému SAK (a analogický systémům Equant, FEL-EXPERT, MYCIN, PROSPECTOR a pod.) ,
- *nekompozicionální odvozování* v rámci logických pravidel. Jak již bylo uvedeno v předcházejícím odstavci, logická pravidla odvozují závěr s jistotou – s využitím klasického dedukčního pravidla modus ponens. V průběhu konzultace pak jedno aktivované pravidlo plně vyhodnotí daný závěr. “Skládání” příspěvků pravidel tedy má charakter disjunkce, tak jak je tomu například v komerčních expertních systémech Nexpert Object nebo Kappa PC.

```
<compositional_rule>
  <id>c1</id>
  <condition>
    <conjunction>
      <literal>
        <id_attribute>velikost_nohy</id_attribute>
        <id_proposition>malá_noha</id_proposition>
        <negation>0</negation>
      </literal>
      <literal>
        <id_attribute>dobrá_rodina</id_attribute>
        <negation>0</negation>
      </literal>
    </conjunction>
  </condition>
  <conclusions>
    <conclusion>
      <id_attribute>princ</id_attribute>
      <negation>0</negation>
      <weight>3,000</weight>
    </conclusion>
  </conclusions>
</compositional_rule>
```

Obr. 1. Zápis pravidla „Pokud má adept malou nohu a je z dobré rodiny, pak je princ“

V obou případech se kombinuje zpětné a přímé řetězení. Zpětně (od závěrů pravidel k předpokladům) se hledají relevantní otázky, které systém klade uživateli v průběhu konzultace. Přímě (od předpokladů pravidel k závěrům) se propagují získané váhy. Aby se co nejlépe využila získaná informace, propagace vah se provádí všemi dostupnými směry – tedy nejen k cíli aktuálně vyhodnocovanému (vybranému) v rámci zpětného řetězení.

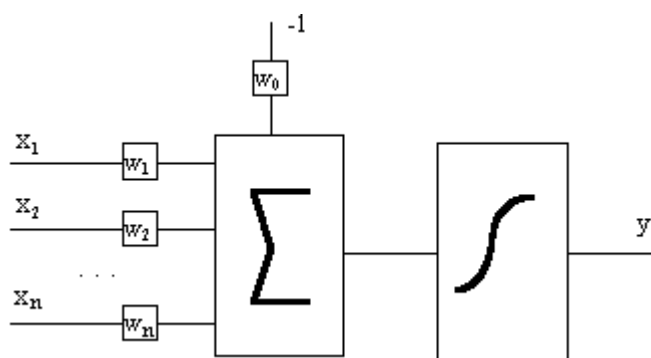
Neurčitost při práci s pravidly. Systém NEST umožňuje pracovat s váhami, které vyjadřují neurčitost v bázi znalostí i neurčité odpovědi během konzultace. Způsob práce s neurčitostí rozšiřuje možnosti, které nabízel systém SAK, vychází tedy z algebraické teorie skládání vah [4]. Připomeňme, že tato teorie definuje axiomy, které musí splňovat funkce² pro výpočet váhy negace výroku (funkce $NEG(w)$), funkce pro výpočet váhy konjunkce výroků (funkce $CONJ(w_1, w_2)$) funkce pro výpočet váhy disjunkce výroků (funkce $DISJ(w_1, w_2)$), funkce pro výpočet příspěvku pravidla a pro modifikaci váhy pravidla kontextem (funkce $CTR(a, w)$) a funkce pro skládání příspěvků více pravidel se stejným závěrem (funkce $GLOB(w_1, \dots, w_k)$). Rozšíření práce s neurčitostí v systému NEST oproti systému SAK spočívá:

- ve větší nabídce inferenčních mechanismů, rozuměj konkrétních podob kombinačních funkcí. NEST nabízí standardní, logický, neuronový a hybridní inferenční mechanismus. Názvy inferenčních mechanismů zasluhují bližší vysvětlení. Standardní inferenční mechanismus vychází ze způsobu práce s neurčitostí v systémech MYCIN (funkce CTR) a PROSPECTOR (funkce $GLOB$). Logický inferenční mechanismus je založen na vícehodnotové logice – funkce CTR je fuzzy modus ponens a funkce $GLOB$ je fuzzy disjunkce v Lukasiewiczově sémantice³. Neuronový inferenční mechanismus je inspirován modelem neuronu (obr. 2); vážené vstupy $w_i x_i$ neuronu můžeme chápat jako příspěvky pravidel (vyhodnocované funkcí CTR), aktivační funkci neuronu (nelineární transformaci váženého součtu vstupů $y = f(\sum_i w_i x_i) = \tanh(\sum_i w_i x_i)$) pak může realizovat vhodná funkce $GLOB$ [1]. Hybridní inferenční mechanismus je kombinací standardního a logického. Ve všech inferenčních mechanismech je stejná funkce NEG ($NEG(w) = -w$), $CONJ$ ($CONJ(w_1, w_2) = \min(w_1, w_2)$) a $DISJ$ ($DISJ(w_1, w_2) = \max(w_1, w_2)$). Podobu funkcí CTR a $GLOB$ pro různé inferenční mechanismy pak ukazuje tabulka 1;
- v možnosti vyjadřovat neurčitost výroku intervalem vah. Interval chápeme tak, že výrok může získat libovolnou váhu z daného intervalu. Interval tedy simuluje opakované konzultace pro všechny možné hodnoty z intervalu. Vzhledem k monotonii kombinačních funkcí stačí pracovat pouze s mezemi intervalů. Přesněji řečeno, pro negaci platí $NEG([w_1, w_2]) = [NEG(w_2), NEG(w_1)]$, pro všechny ostatní funkce platí $F([w_1, w_2], [v_1, v_2]) = [F(w_1, v_1), F(w_2, v_2)]$. Zatímco váha pravidla je vždy pouze jedno číslo, u dotazů lze v některých případech odpovědět „plným“ intervalem $[-1, 1]$. Tato odpověď je zatím jediný způsob, jak lze do konzultace vnést práci s intervaly.

² Všechny funkce jsou definovány na normovaném intervalu $[-1, 1]$, kde 1 znamená „jistě ne“, 0 znamená „nevím“ a 1 znamená „jistě ano“.

³ Jde tedy o fuzzy obdobu výše uvedeného nekompozicionálního odvozování.

Vyhodnocení jednoho pravidla probíhá tak, že nejprve se (je-li zadán kontext) vyhodnotí pomocí funkcí *NEG*, *CONJ* a *DISJ* váha kontextu pravidla. Pak se z váhy kontextu $w(CTX)$ a původní váhy pravidla w spočítá aktualizovaná váha pravidla $w' = CTR(w(CTX), w)$. Poté se (pomocí funkcí *NEG*, *CONJ* a *DISJ*) vyhodnotí váha předpokladu pravidla a nakonec se z váhy předpokladu a aktualizované váhy pravidla spočítá příspěvek pravidla (pomocí funkce *CTR*). Příspěvky více pravidel se pak kombinují pomocí funkce *GLOB*.



Obr. 2. Model neuronu.

Tabulka 1. Funkce *CTR* a *GLOB* pro různé inferenční mechanismy.

Inferenční mechanismus	Funkce $CTR(a, w)$ pro váhu předpokladu $a > 0$	Funkce $GLOB(w_1, \dots, w_k)$
standardní	$a w$	$(w_1 + w_2) / (1 + w_1 w_2)$
logický	$\text{sign}(w) \max(a + w - 1)$	$\min(\sum_{w > 0} w, 1) - \min(\sum_{w < 0} w , 1)$
neuronový	$a w$	$\min(\max(\sum_i w_i, -1), 1)$
hybridní	standardní	logický

Vyhodnocení integritních omezení. Integritní omezení se vyhodnocují až po skončení konzultace. Nejprve se zjišťuje, zdali je integritní omezení porušeno, nebo-li do jaké míry neplatí implikace $ANT \Rightarrow SUC$. K tomu slouží funkce $IMPL(w(ANT), w(SUC)) = \max(0, \min(1, w(ANT) - w(SUC)))$. Poté se z míry porušení integritního omezení $IMPL(w(ANT), w(SUC))$ a stupně jeho důležitosti (případně stupně důležitosti aktualizovaného kontextem integritního omezení, tato aktualizace se podobně jako u pravidel opět provádí pomocí funkce *CTR*) spočítá pomocí funkce *CTR* výsledný příspěvek integritního omezení k inkonzistenci celé konzultace. Stupeň důležitosti integritního omezení tedy hraje podobnou roli jako váha u pravidla – nejde jen o to, jak silně bylo integritní omezení porušeno ale i o to, jak je toto integritní omezení významné.

2.3 Konzultace se systémem

System nabízí čtyři základní režimy práce (obr. 3)⁴:

- dialogový režim je klasický režim práce s expertním systémem. System se zeptá na váhu výroku-dotazu a na základě odpovědi uživatele vybere další dotaz. Kladení dotazů se přitom řídí zpětným řetězením;
- režim dialog s dotazníkem umožňuje zrychlit klasickou konzultaci. Místo sekvenčního kladení otázek systém nabídne seznam všech dotazů. Pokud uživatel neodpoví na dotaz, který bude relevantní v dalším průběhu konzultace, systém se v příslušnou chvíli zeptá v klasickém dialogovém režimu – nezodpovězené dotazy se tedy vybírají pomocí zpětného řetězení;
- v režimu dotazník se uživateli rovněž zobrazí seznam všech dotazů. Na rozdíl od předcházejícího režimu se ale systém k nezodpovězeným dotazům v dalším průběhu konzultace nevrací a přiřadí jim implicitní váhu (viz dále). Tento režim práce tedy simuluje klasické přímé řetězení, kdy se odvozuje pouze z faktů známých před začátkem konzultace;
- při načítání odpovědi ze souboru se uživateli zobrazí předvyplněný dotazník, uživatel samozřejmě může libovolnou odpověď změnit. Odvozování v tomto režimu probíhá přímým řetězením – dotazy na nezměněné (tedy i v souboru neznámé) odpovědi se již nepokládají.

Z dalších možností volby režimu konzultace zmiňme:

- možnost odkládat odpovědi – tento globální režim umožní odložit odpověď na některý dotaz. Odložené dotazy se v případě potřeby pokládají po skončení konzultace;
- možnost měnit priority pravidel podle přednastavených voleb;
- možnost „nulovat“ atributy prostředí.

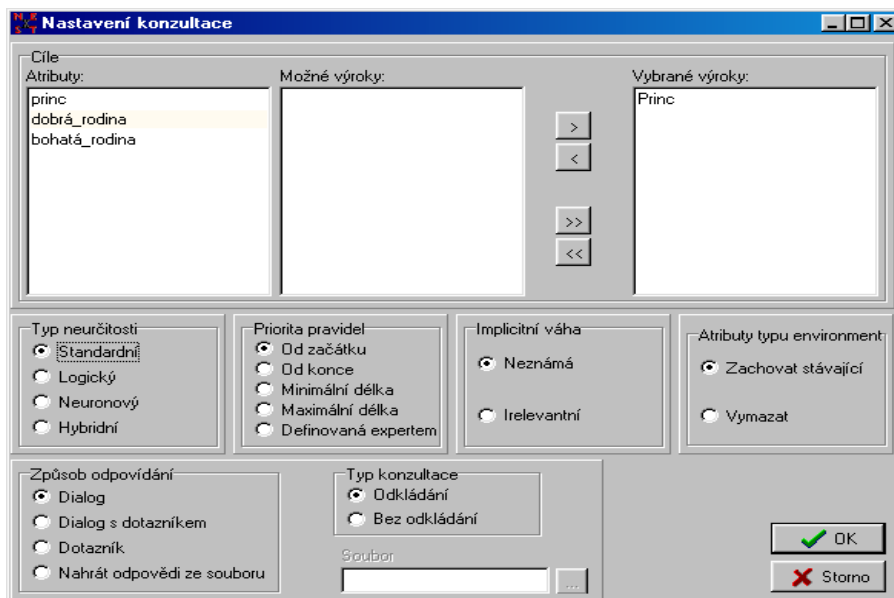
V průběhu konzultace uživatel odpovídá na otázky, které mu klade expertní systém NEST (obr. 4). Podle typu atributu zadává:

- váhu výroku (pro binární atribut),
- hodnotu atributu (pro numerický atribut),
- hodnotu atributu a jeho váhu (pro jednoduchý nominální atribut),
- seznam hodnot atributů a jejich vah (pro množinový nominální atribut).

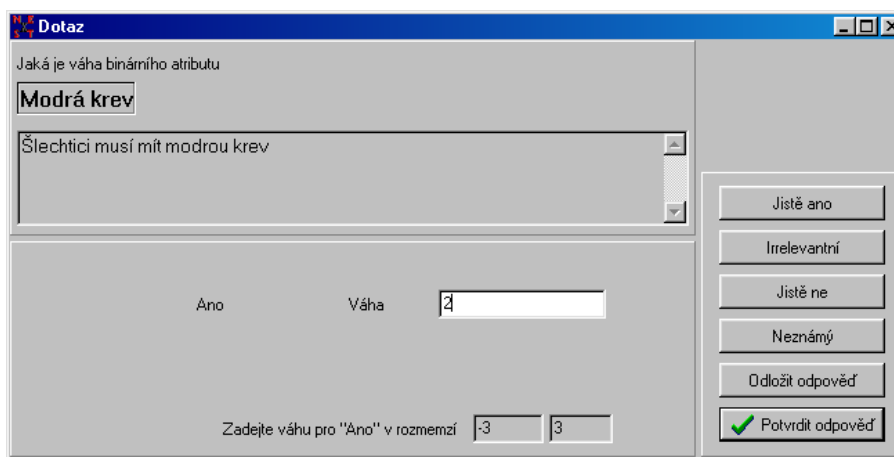
Při zadávání neurčitosti může uživatel odpovědět konkrétní číselnou vahou, případně jednou z předdefinovaných hodnot „jistě ano“ (váha 1), „irelevantní“ váha 0), „jistě ne“ (váha -1), „neznámý“ (normovaná váha [-1,1]), případně může odpověď odložit (výroku se přiřadí váha [-1,1] a po skončení konzultace je možno se k němu vrátit). Dotazy nezodpovězené v průběhu konzultace získávají automaticky implicitní váhu. Volba této hodnoty (obr. 3) souvisí s dvojitým možným chápáním odpovědi „nevím“. V prvním případě je interpretace „může platit cokoliv“ – tedy plný váhový interval [-1, 1], ve druhém případě je interpretace „není podstatné“ – tedy váha 0.

⁴ Obrázky ukazují konzultaci jejímž cílem je rozpoznávat prince podle velikosti nohy, barvy krve a majetku. Jde o jednoduchý ilustrační příklad převzatý z výukového videa.

Odpovědi na dotazy získává systém z různých zdrojů. Implicitním zdrojem je přímé zadání z klávesnice. Dalšími zdroji mohou být spuštění externí procedury, výpočet z hodnot jiných atributů (tyto dvě možnosti se nabízejí pouze pro numerické atributy) nebo automatické načtení z předem připravených souborů. Informace o typu a pořadí zdrojů pro jednotlivé atributy je součástí báze znalostí.



Obr. 3. Volba režimu konzultace.



Obr. 4. Odpovídání na dotaz systému (binární atribut).

Výsledkem konzultace je seznam cílů (implicitně výroků, které jsou pouze v závěrech pravidel, uživatel má ale možnost zvolit pro odvozování libovolný výrok, který není dotazem – viz obr. 3) uspořádaný podle vah. Uživatel má možnost změnit odpovědi na některé otázky (a znovu nechat odvodit váhy cílů), uložit odpovědi z konzultace do souboru, který je možno použít jako vstup do nové konzultace (tato možnost je obzvláště vhodná pro ladění a testování báze znalostí), nebo exportovat výsledky konzultace (do textového souboru, do HTML nebo do Excelu). Podobu obrazovky s výsledky konzultace ukazuje obr. 5.

The screenshot shows a window titled "Výsledky" with a table of inference results and a control panel on the right. The table has columns for name, minimum weight, maximum weight, status, and type. The control panel includes a "Zobrazovat" section with radio buttons for "Cíle" and "Všechny výroky", and buttons for "Úprava odpovědí", "Export výsledků", "Uložit odpovědi", and "Zavřít".

Ujméno	Min váha <	Max váha	Stav	Druh
Malá noha	3.000	3.000	finální	dotaz
Princ	2.000	2.000	finální	cíl
dobrá rodina	2.000	2.000	finální	mezilehlý
Modrá krev	2.000	2.000	finální	dotaz
Bohatá rodina	2.000	2.997	finální	mezilehlý
majetek(penize)	2.000	2.000	finální	dotaz
Střední noha	-3.000	-3.000	finální	dotaz
Velká noha	-3.000	-3.000	finální	dotaz
majetek(pozemky)	-3.000	3.000	finální	dotaz

Obr. 5. Výsledek konzultace.

2.4 Implementace systému

Implementace systému je poplatná současnému způsobu programování – vychází tedy ze zásad objektového programování. Systém je implementován ve dvou podobách: jako stand-alone verze a jako verze klient-server. Obrázky ilustrující v příspěvku podobu systému jsou převzaty ze stand-alone verze. Obě verze jsou implementované ve vývojovém prostředí Delphi (verze 7.0) firmy Borland a jsou určeny pro operační systémy Windows (95 a výše).

Ačkoliv je stand-alone verze jediným programem (jeden „exe“ soubor), vnitřní architektura je vytvořena jako klient-server aplikace. To znamená, že je oddělen „základ“ systému od komunikace s uživatelem. Pro veškerou komunikaci jsou v „základu“ systému volány metody jediného objektu (Komunikace_s_klientem), který komunikuje s druhým objektem (Komunikace_se_serverem), jenž spouští příslušné formuláře pro předávání informací uživateli. Tato architektura poskytuje možnost vytvořit systém, ve kterém by „základ“ systému byl na jiném počítači než

komunikace s uživatelem. Stačí do příslušných objektů komunikace (resp. do jejich potomků) doplnit zaslání zpráv po síti.⁵

Klient-server verze je řešena jako webový server, ke kterému se uživatel připojuje pomocí klasických internetových prohlížečů (např. Internet Explorer). Jedná se tedy o trochu jinou variantu, než jaká byla naznačena při popisu vnitřní architektury stand-alone verze, kde by byl na straně klienta program „na míru“. Zde se jedná o klasický webový server, tedy program, který čeká na příslušném portu a při příchodu požadavku provede jeho vyhodnocení a vrátí uživateli odpověď. Víceuživatelský přístup je řešen tak, že pro každého přihlášeného uživatele (identifikovaného IP adresou) vytvoří server vlastní kopii báze znalostí. Po síti se tak přenáší pouze otázky a odpovědi na aktuální dotazy a nikoliv informace o celé bázi, čímž se přenos značně urychlí. Stránky, které server posílá uživateli, jsou uloženy v externích souborech a mají formát jakéhosi „pseudo HTML“. Jedná se o standardní HTML, ve kterém jsou však „pseudo tagy“ (např. `<#tabulka_vysledku>`), za které server při odesílání stránky dosadí hodnoty z aktuální báze znalostí. Správce systému může tyto soubory upravit dle své představy (se zachováním pseudo tagů), takže výsledné stránky každého systému mohou vypadat zcela libovolně. Navíc je možné vytvořit několik sad těchto stránek a každé bázi znalostí přiřadit jinou sadu.

Kromě konzultačního programu, kterému je věnován tento příspěvek existuje i editor bází, program NESTEditor. Implementace editoru je řešena podobně, jako konzultační program ve stand-alone podobě. Zde však již není vnitřní architektura klient-server, ale jedná se o MDI aplikaci (aplikaci, ve které může být otevřeno několik dokumentů – bázi znalostí – najednou). Vzhledem k tomu, že báze znalostí má podobu XML dokumentu, bylo by možné použít běžný textový (ascii) editor, ale editor šitý na míru DTD báze značně urychlí vývoj aplikací. Navíc kromě kontroly syntaxe báze (resp. automatické tvorby syntaxe) poskytuje editor i kontrolu základní logiky báze – kontrola stejných identifikátorů prvků, nemožnost přiřadit pravidlu neexistující výrok, hledání cyklů v pravidlech.

Současná verze nabízí výběr ze dvou jazyků – angličtiny a češtiny. Veškeré texty v systému nejsou implementovány „napevno“, ale jsou načítány z externích souborů (XML dokumentů). To umožňuje snadné opravy stávajících textů i snadnou tvorbu jazykových mutací systému NEST. Jak editor báze, tak konzultační program jsou vybaveny kontextovou nápovědou popisující ovládání i základní principy systému.

3 Závěr

Systém NEST je v současné době ve stádiu ověřování a testování. Popisovaná verze odpovídá druhé polovině října 2003. Co v tuto chvíli zbývá implementovat je, kromě odstranění chyb, zejména doplnění vysvětlovacích možností. Předpokládáme standardní vysvětlování “how”, “why” a “what-if”. Další, větší rozšíření systému, které plánujeme, bude spočívat v doplnění případového usuzování a jeho kombinace

⁵ V současné stand-alone verzi systému jsou některé zprávy pro uživatele (např. chybové zprávy) volány přímo ze “základu” systému z důvodu urychlení. V případě oddělování jednotlivých částí však není problém tyto zprávy posílat přes příslušné objekty komunikace.

s usuzováním založeným na pravidlech. Aktuální dokumentace a stand-alone verze systému je k dispozici ke stažení na <http://lisp.vse.cz/NEST>.

Jednou z předpokládaných aplikací NESTu je odvozování nových informací při vícecestné analýze WWW v rámci distribuovaného znalostního systému *Rainbow* [10] vyvíjeného na stejném pracovišti. Pravidla pro převod elementárních charakteristik webových dat na abstraktní vlastnosti a pro slučování informací z různých zdrojů referujících o stejných webových objektech mají heuristický charakter, a zachycení neurčitosti je u nich nezbytné. Jinou zamýšlenou aplikací (tentokrát v rámci výzkumného centra EuroMISE-Kardio) je systém pro určování rizika aterosklerózy.

Reference

1. Berka P., Sláma M. Using Neural Nets to Learn Weights of Rules for Compositional Expert Systems. In: *Methodology and Tools in Knowledge-Based Systems*. Proc. 11th Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA-98-AIE, Springer, 1998.
2. Duda R.O., Gasching J.E. Model Design in the Prospector Consultant System for Mineral Exploration. In: Michie, D. *Expert Systems in the Micro-Electronic Age*. Edinburgh University Press, 1979.
3. Feigenbaum E.A. Themes and Case Studies of Knowledge Engineering. In: Michie, D. *Expert Systems in the Micro-Electronic Age*. Edinburgh University Press, 1979.
4. Ferjenčík J., Ivánek J., Švenda J. O prázdném expertním systému SAK. In: *Expertné systémy ČSVTS*, Bratislava, 1987.
5. Hájek P. Combining Functions for Certainty Factors in Consulting Systems. *Int. J. Man-Machine Studies* 22, 1985.
6. Hájek P., Hájková M., Havránek T. Expertní systém EQUANT. In: *Expertné systémy ČSVTS*, Bratislava, 1987.
7. Mařík V., Zdráhal Z. Expertní systémy řady FEL-EXPERT. In: *Expertné systémy ČSVTS*, Bratislava, 1987.
8. Shortliffe E.H. *Computer-Based Medical Consultations: MYCIN*. Elsevier, New York, 1976.
9. Svátek V., Berka P. Specifications of a Question/Answer Knowledge-Based Agent. In: *Artificial Intelligence Techniques AIT'95*, Brno, 1995.
10. Svátek V., Kosek J., Labský M., Bráza J., Kavalec M., Vacura M., Vávra V., Snášel V. Rainbow - Multiway Semantic Analysis of Websites. In: *2nd International DEXA Workshop on Web Semantics (WebS03)*, Prague 2003, IEEE Computer Society Press 2003.
11. The Data Mining Group. <http://www.dmg.org>
12. The Rule Markup Initiative. <http://www.dfki.uni-kl.de/ruleml/>

Annotation:

New Expert SysTem NEST.

This paper describes our ongoing work on a rule based expert system shell NEST, a successor of the expert system SAK developed at the University of Economics in mid 80. NEST works with propositions (derived from binary, nominal or numeric attributes), rules (compositional with uncertainties expressed as weights, and non-compositional without uncertainties), contexts and integrity constraints. The inference mechanism is based on a combination of backward and forward chaining of rules. Four different ways are available for uncertainty processing. The system is implemented in two versions: stand-alone version for PC's running Windows, and client-server version as web server and web browser.

Projekt je částečně podporován grantem GAČR 201/03/1318