

# **Expertní systémy**

Distanční studijní text

**Jan Górecki**

**Karviná 2017**



**SLEZSKÁ  
UNIVERZITA**  
OBCHODNĚ PODNIKATELSKÁ  
FAKULTA V KARVINĚ

**Obor:** Umělá inteligence

**Klíčová slova:** expertní systém, znalostní inženýrství, umělá inteligence, NEST

**Anotace:** Studijní text je určen studentům magisterského stupně na vysokých školách ekonomického zaměření. Student se seznámí s pojmem expertní systém v souvislosti s vědním oborem Umělá inteligence. Dále se student seznámí s možnostmi návrhu a tvorby expertního systému v rámci disciplíny nazývané Znalostní inženýrství. Aby si student poznatky mohl prakticky vyzkoušet, je zde také představen akademický nástroj NEST, který expertní systémy umožňuje vytvářet.

**Autor:** **Ing. Jan Górecki, Ph.D.**

## 1 Obsah

|  |    |
|--|----|
| ÚVODEM.....  | 6  |
| RYCHLÝ NÁHLED STUDIJNÍ OPORY.....                          | 7  |
| 1 EXPERTNÍ SYSTÉMY V KONTEXTU UMĚLÉ INTELIGENCE.....       | 8  |
| 1.1 Umělá inteligence.....                                 | 8  |
| 1.2 Pojem expertní systém .....                            | 11 |
| 1.3 Základní principy znalostního inženýrství .....        | 12 |
| 1.1.1 Princip expertních systémů .....                     | 12 |
| 1.1.2 Reprezentace znalostí.....                           | 12 |
| 1.1.3 Vysvětlovací schopnost .....                         | 13 |
| 1.1.4 Charakteristické rysy .....                          | 14 |
| 1.4 Typy aplikací expertních systémů.....                  | 15 |
| 1.1.5 Příklady diagnostických expertních systémů.....      | 16 |
| 1.5 Struktura expertního systému.....                      | 18 |
| 1.6 Znalosti.....  | 20 |
| 1.7 Reprezentace znalostí.....                             | 21 |
| 1.8 Nepravdivé a hybridní expertní systémy.....            | 24 |
| 1.8.1 Sémantické sítě .....                                | 24 |
| 1.8.2 Rámce .....  | 26 |
| 1.8.3 Objekty.....   | 28 |
| 1.8.4 Hybridní systémy .....                               | 30 |
| 1.8.5 Systémy typu tabule.....                             | 31 |
| 1.9 Infereční mechanismus.....                             | 33 |
| 1.10 Neurčitost.....                                       | 34 |
| 2 PROSTŘEDÍ NEST.....                                      | 39 |
| 2.1 HISTORIE NESTu.....                                    | 39 |
| 2.2 NEST.....  | 40 |
| 2.2.1 Popis instalačního balíku a vlastní instalace .....  | 40 |
| 2.2.2 Nastavení.....                                       | 42 |
| 2.2.3 NEST – základní možnosti a spuštění konzultace ..... | 44 |
| 2.2.4 Možnosti okna Dotaz a průběh konzultace .....        | 47 |
| 2.2.5 Ukládání odpovědí .....                              | 50 |

|       |   |     |
|-------|---|-----|
| 2.2.6 | Výsledky konzultace .....                                     | 51  |
| 3     | STRUKTURA BÁZE ZNALOSTÍ A INFERENČNÍ MECHANIZMUS V NEST ..... | 56  |
| 3.1   | NEST editor.....  | 56  |
| 3.1.1 | Zadávání nových prvků báze znalostí.....                      | 59  |
| 3.2   | Reprezentace znalostí.....                                    | 61  |
| 3.2.1 | Atributy a výroky .....                                       | 61  |
| 3.2.2 | Pravidla .....  | 66  |
| 3.2.3 | Kontexty.....   | 67  |
| 3.2.4 | Integritní omezení .....                                      | 67  |
| 3.3   | Inferenční mechanismus.....                                   | 68  |
| 4     | NEURČITOST V NEST .....                                       | 72  |
| 4.1   | Práce s neurčitostí.....                                      | 72  |
| 4.2   | Funkce pro práci s neurčitostí .....                          | 73  |
| 4.2.1 | Funkce NEG.....   | 74  |
| 4.2.2 | Funkce CONJ.....  | 75  |
| 4.2.3 | Funkce DISJ.....  | 76  |
| 4.2.4 | Funkce CRT .....  | 77  |
| 4.2.5 | Funkce GLOB.....  | 78  |
| 4.2.6 | Funkce NORM.....  | 80  |
| 4.2.7 | Funkce IMP.....   | 80  |
| 4.3   | Příklad výpočtu .....   | 81  |
| 4.4   | Srovnání přístupů práce s neurčitostí v NEST .....            | 83  |
| 5     | APLIKACE V NEST .....   | 87  |
| 5.1   | Výběr problému a grafická reprezentace .....                  | 87  |
| 5.2   | Příklad zpracování báze znalostí.....                         | 88  |
| 5.3   | Založení nové báze.....                                       | 93  |
| 5.4   | Konstrukce báze znalostí.....                                 | 95  |
| 5.5   | Ladění báze .....   | 100 |
| 5.6   | Tvorba dokumentace .....                                      | 102 |
|       | LITERATURA .....  | 105 |
|       | SHRNUTÍ STUDIJNÍ OPORY .....                                  | 107 |
|       | PŘÍLOHA Č. 1 VZOROVÁ SEMINÁRNÍ PRÁCE.....                     | 108 |
|       | ÚVOD.....   | 110 |

|       |                                       |     |
|-------|---------------------------------------|-----|
| 1     | NÁVRH BÁZE.....                       | 110 |
| 2     | STATISTIKY .....                      | 112 |
| 3     | PŘÍKLADY DIAGNÓZ .....                | 113 |
| 3.1   | Diagnóza Privát Karviná 1, 3, 4 ..... | 113 |
| 3.1.1 | Typický uživatel.....                 | 113 |
| 3.2   | Diagnóza Privát Karviná 6, 7, 8 ..... | 114 |
| 3.2.1 | Typický uživatel.....                 | 114 |
| 3.3   | Diagnóza Kolej Vyhlídka.....          | 115 |
| 3.3.1 | Typický uživatel.....                 | 115 |
| 3.4   | Diagnóza Ubytovna Kosmos.....         | 116 |
| 3.4.1 | Typický uživatel.....                 | 116 |
| 4     | ZÁVĚR .....                           | 117 |
|       | PŘEHLED DOSTUPNÝCH IKON.....          | 119 |

## ÚVODEM

Tento studijní text vznikl se záměrem stát se oporou při studiu předmětu Expertní systémy na Obchodně-podnikatelské fakultě v Karviné, Slezská univerzita v Opavě (OPF SU) a je určen studentům magisterského stupně. Text částečně vychází z učebního textu Znalostní inženýrství (Ivánek, Laš, Kempný (2007)), což reflektuje sloučení předmětů Znalostní inženýrství a původní verzi předmětu Expertní systémy vyučovaných na OPF SU.

Při studiu textu je výhodou, pokud je čtenář již seznámen s Informačními systémy, avšak jejich neznalost není pro úspěšné absolvování předmětu překážkou.

Text je také doplněn o šablonu semestrální práce, která je pro úspěšné absolvování předmětu vyžadována.

Každá kapitola textu zahrnuje několik distančních prvků, pomocí kterých se student může v rychlosti seznámit s obsahem dané kapitoly, zjistí, co je cílem dané kapitoly, a kterým klíčovým pojmem se kapitola věnuje. V závěru každé kapitoly najde čtenář shrnutí toho nejdůležitějšího, čemu se kapitola věnovala, sadu kontrolních otázek, pomocí nichž si čtenář může své nabyté znalosti ověřit a také prostor pro vlastní poznámky.

## RYCHLÝ NÁHLED STUDIJNÍ OPORY

Student se seznámí s pojmem *expertní systém* v souvislosti s vědním oborem Umělá inteligence, v rámci kterého se zhruba v 70. letech 20. století začalo o expertních systémech jako takových hovořit. Dále se student seznámí s možnostmi návrhu a tvorby expertního systému v rámci disciplíny nazývané Znalostní inženýrství, která se věnuje možnostem získávání, reprezentace a využití znalostí, které tvoří jednu z nejdůležitějších složek každého expertního systému. Aby si student mohl získané poznatky prakticky vyzkoušet, je zde také představen akademický nástroj NEST, který umožňuje vytvářet určitý typ expertních systémů. V rámci tohoto nástroje je zde také představená práce s neurčitostí, která je nedílnou součástí většiny expertních systémů.

# 1 EXPERTNÍ SYSTÉMY V KONTEXTU UMĚLÉ INTELIGENCE



## RYCHLÝ NÁHLED KAPITOLY

V této kapitole se čtenář seznámí se základními pojmy týkajícími se expertních systémů a Umělé inteligence, mezi které se řadí znalosti a jejich reprezentace, neurčitost, inferenční mechanismus.

---



## CÍLE KAPITOLY

Cílem kapitoly je uvést čtenáře do oblasti Umělé inteligence a Znalostního inženýrství, které se zabývá prací se znalostmi a vytvářením expertních systémů. Po prostudování kapitoly bude student schopen vysvětlit pojem expertní systém, popsat jeho obecnou strukturu a charakteristické rysy.

---



## KLÍČOVÁ SLOVA KAPITOLY

expertní systém, umělá inteligence, znalostní inženýrství, neurčitost, inferenční mechanismus

---

## 1.1 Umělá inteligence

Za počátek vědního oboru Umělá inteligence je považována práce A. M. Turinga z roku 1950, v níž jsou blíže vymezeny prostředky a cíle umělé inteligence (artificial intelligence, AI). Inteligenci stroje navrhuje Turing testovat imitační hrou: úkolem stroje bude v komunikaci s experimentátorem (pomocí dálnopisu) předstírat, že je člověk.



## DEFINICE

Odtud vychází známá Minského definice, v níž je **umělá inteligence** vymezena jako věda, jejímž úkolem je naučit stroje, aby dělaly věci, které vyžadují inteligenci, jsou-li prováděny člověkem.

---



První výsledky výzkumu v 50. letech vedly k velkému nadšení a optimismu. Newell, Shaw a Simon sestrojili v roce 1956 program **Logic Theorist**, který automaticky dokázal převážnou část vět v Principia Mathematica, což je třísvazkové dílo, ve které se A. N. Whitehead s jeho žákem B. Russellem pokusili upevnit a sjednotit logické základy matematiky. Jejich **General Problem Solver** (1957-59) byl po řadu let nejsložitějším a nejpodrobnějším modelem lidského myšlení, který kdy byl zkonstruován a ověřován. Simonovy práce z této doby, týkající se racionality, rozhodování a řešení problémů (adresované zejména ekonomům a psychologům), byly později (v roce 1978) oceněny Nobelovou cenou. Zdálo se, že větší a rychlejší počítače brzy umožní již vyvinutými obecnými postupy simulovat většinu mentálních aktivit člověka.

V šedesátých letech došlo ke zpomalení vývoje a ke zjištění, že obecné programy nestačí pro řešení skutečně reálných úloh. Postupně se ukázalo, že stroj k řešení úlohy potřebuje znát to, co o problému ví člověk, který jej řeší. V kostce řečeno, člověka expertem nedělá jen vysoké IQ, ale hlavně specializované znalosti, „know-how“. Zde je možno sledovat počátek linie, která vedla ke vzniku **znalostního inženýrství**.

V sedmdesátých letech se v oboru umělé inteligence výrazně odlišily tři podobory:

- zpracování přirozeného jazyka,
- robotika (včetně strojového vidění),
- znalostní inženýrství.

Toto rozdělení odpovídá třem typů lidských inteligentních aktivit: **komunikace**, **manipulace** (a rozpoznávání), **řešení problémů** (usuzování). Názvy podoborů a větší specializace výzkumu v umělé inteligenci odpovídají také nové situaci, do níž se umělá inteligence propracovala: stala se předmětem komerčních a státních projektů.

Motivace a zaměření výzkumníků v umělé inteligenci jsou často velmi odlišné. První, **inženýrský směr**, se orientuje na vytvoření inteligentních systémů či metod a technik, které jsou k tomu potřebné. Neomezuje se tedy pouze na simulaci lidských technik řešení úloh. Druhý, **psychologický směr**, se orientuje na vytvoření umělých modelů lidských intelektuálních činností. Obecné studium a formalizaci intelektuálních mechanismů nezávisle na jejich realizaci lze chápat zvlášť jako třetí, filosoficko-matematický směr.

Všechny podobory, směry a koncepce v umělé inteligenci se vzájemně prolínají v konstrukci integrálních systémů, vybavených více schopnostmi. Zřetelně se to projevuje v robotice a v projektech komplexních informačních systémů.

Výzkum v umělé inteligenci souvisí též s řadou dalších disciplín, jako je např. kybernetika, psychologie, bionika, teorie řízení, lingvistika, filozofie, atd. Významná je spjatost s rozvojem počítačů a vůbec technických prostředků zpracování informací na jedné straně a s matematickou informatikou (teoretickou vědou o počítačích) na druhé straně. Lze říci, že výzkum umělé inteligence se stává vřídícím faktorem v celé informatice a informačním průmyslu.

První velkou aplikací umělé inteligence na problémy jiné než hraní her a řešení hlavolamů byl program **DENDRAL** (1971). Systém DENDRAL byl určen pro identifikaci struktury molekul. Byl vyvíjen déle než deset let za účasti řady odborníků a v oblasti své specializace překonal své tvůrce a další specialisty.

Během vývoje systému DENDRAL se v polovině 60. let stal zjevným význam, který pro konstrukci prakticky použitelných inteligentních systémů, má zabudování speciálních znalostí z oboru jejich aplikace. V prvním souhrnu výsledků projektu DENDRAL bylo zdůrazněno, že člověk je schopen dobře řešit pouze úlohy z oblasti, v nichž je specialistou – v ostatních oblastech mu chybí příslušné znalosti. Z tohoto hlediska je **expert člověk se znalostmi specialisty v daném oboru a se speciálními metodami a heuristikami.**

Postupně došlo k posunu v paradigmatu výzkumu umělé inteligence: za hlavní začala být považována speciální, problémově orientovaná znalost a pozornost se soustředila na hledání metod a technik, jak tuto znalost reprezentovat tak, aby mohla být efektivně využívána.

Během 70. let dvacátého století byla na tomto principu vybudována řada úspěšných aplikací, tzv. expertních systémů a posléze se pro tento směr výzkumů a aplikací umělé inteligence začal používat název znalostní inženýrství.

Domníváme se, že znalostní inženýrství je možno definovat v kontextu informačního inženýrství – disciplíny, která se zabývá zpracováním informací do použitelné podoby. Znalostní inženýrství se pak vyděluje jako část informačního inženýrství, soustřeďující se na informace o tom, jak dospívat k novým informacím, tj. informace o usuzování v konkrétních situacích. Tyto informace jsou jednak věcné (znalosti), jednak logické (inferenční procedury). Část znalostního inženýrství, která se zabývá speciálně inferenčními mechanismy, lze nazvat logické inženýrství.

Praktickým výsledkem znalostního inženýrství jsou **expertní systémy – programy poskytující erudované závěry na základě modelu znalostí.**

## 1.2 Pojem expertní systém

Definice pojmu **expertní systém** se různí. Uvedme si tři populární definice tohoto pojmu:

### DEFINICE



Podle E. Feigenbauma je expertní systém inteligentní počítačový program, který užívá znalosti a inferenční procedury k řešení problémů, které jsou natolik obtížné, že pro své řešení vyžadují významnou lidskou expertízu.

Expertní systém je počítačový program simulující rozhodovací činnost lidského experta při řešení složitých úloh a využívající vhodně zakódovaných speciálních znalostí převzatých od experta s cílem dosáhnout ve zvolené problémové oblasti kvality rozhodování na úrovni experta.

Expertní systém je systém na řešení takových problémů, který by lidský expert v dané oblasti dokázal vyřešit po telefonu.

Expertní systémy se objevily v 70. letech 20. století, kdy se vývoj umělé inteligence zaměřoval převážně na reprezentování a zpracování znalostí získaných od experta. Jsou tedy jedním z oborů umělé inteligence, kde kromě nich můžeme nalézt robotiku, porozumění přirozenému jazyku a mluvené řeči, neuronové sítě a další. Dnes již mnoho lidí považuje expertní systémy za zastaralé a překonané. Toto tvrzení vychází především z vědeckých kruhů, neboť principy expertních systémů byly již natolik důkladně prozkoumány, že nový objev v tomto směru je jen velmi málo pravděpodobný a úsilí vědy se proto obrací jiným směrem. V podnikové praxi jsou expertní systémy využívány spíše jako aplikace drobnějšího charakteru uvnitř podniků. Nejsou určeny pro „nahrazení“ experta v dané oblasti, ale jako „pomocník“ experta či manažera. Můžeme vidět aplikace pro „vyhledávání vhodného pracovníka s příslušnými znalostmi“ či „kontrolu určitého procesu ve firmě“

Expertní systémy tedy často slouží pro nalezení rozhodnutí, ke kterému by uživatel dospěl sám za delší čas, nebo jako „seznam“ navazujících procesů, aby uživatel na něco nezapomněl. Upřednostňování expertních systémů před jinými, mnohdy i „inteligentnějšími“ systémy, může být způsobeno tím, že jsou často „průhlednější“ než většina jiných „inteligentních“ systémů. To znamená, že uživatel ví, proč se systém rozhodl právě takto (buďto uživatel sám naplnil systém příslušnými pravidly, nebo si od systému vyžádal příslušné vysvětlení) a navíc má možnost si systém upravit dle svých představ.

## 1.3 Základní principy znalostního inženýrství

Znalostní systém je programový systém, který se od běžných programů liší způsobem, jakým jsou v něm znalosti organizovány, začleněny a využívány a způsobem interakce s uživatelem. Na rozdíl od většiny konvenčních programů, ve kterých jsou znalosti implicitně roztroušeny v programu, je rozhodující většina znalostí důležitých pro řešení problémů daného typu v rámci znalostního systému soustředěna explicitně v samostatně modulární struktuře – **bázi znalostí**.

### 1.1.1 PRINCIP EXPERTNÍCH SYSTÉMŮ

Nejdůležitějším principem expertních systémů je tak **oddělení modelu znalostí od způsobu jejich používání**. Což znamená, že při doplňování a modifikacích báze znalostí není nutno modifikovat ostatní části znalostního systému. Báze znalostí obsahuje **faktické a heuristické znalosti** o dané oblasti. Inferenční mechanismus pracuje s bázi znalostí při řešení konkrétní úlohy.

**Faktické znalosti** jsou různé údaje, poznatky, výsledky, atd. Tyto znalosti obvykle expert předává veřejnosti pomocí odborných publikací a vystoupení. S jádrem faktických znalostí některého oboru jsou seznamováni studenti v klasické výuce.

**Heuristické znalosti** jsou různé pracovní postupy, pravidla úsudků a odhadů, způsoby odhadů, způsoby odhalování a vynalézání nového, atd. Heuristické znalosti jsou zpravidla zkušenostní povahy, jsou nejisté. Vzhledem k tomu zůstávají soukromé a jen zřídka je umí expert formulovat a předat veřejnosti.

Báze znalostí určuje do značné míry kvalitu celého expertního systému. Potvrdilo se, že většina podstatných znalostí je heuristické povahy a expert sám často není schopen tyto znalosti uspokojivě vysvětlit laikovi v daném oboru. Pro získání znalostí je proto nutná zdlouhavá práce znalostního inženýra s dotyčným expertem. Východiskem je obvykle pečlivá analýza mnoha řešených problémů.

Strastiplnou týmovou prací experta a znalostního inženýra lze extrahovat reálné znalosti, které nejsou součástí učebnic, ale jsou přitom rozhodující pro činnost experta. Vysvětlení těchto heuristických znalostí je sama o sobě významným výsledkem aplikace znalostního inženýrství.

### 1.1.2 REPREZENTACE ZNALOSTÍ

Pro **reprezentaci znalostí** byla v umělé inteligenci vytvořena řada prostředků. Kromě logických kalkulů se používají např. sémantické sítě, tzv. rámce, gramatiky atd. V expertních systémech bývá báze znalostí vyjádřena nejčastěji **systémem pravidel typu situace – akce**. Situace je vystižena kombinací podmínek na okamžitý stav údajů a poznatků o řešené úloze, akce pak tyto poznatky doplňuje a mění. Vyjádření podmínek akcí se musí opírat o

jistou datovou strukturu odpovídající základním pojmům a poznatkům v dané oblasti. Pravidla pak vedou od rozpoznání stavu této databáze k její modifikaci.

Jednotlivá **pravidla** jsou vyjádřením jakýchsi atomických (na zvolené úrovni modelování) znalostí. Granularita báze znalostí musí odpovídat dané problémové oblasti: příliš rozdrobená znalost je těžko programově zvládnutelná a špatně pochopitelná, v opačném případě velká agregace systému nedovoluje dostatečnou diskriminaci jevů.

Reprezentace báze znalostí jako systému jednoduchých heuristik má výhodu ve snadné interpretaci a modifikaci. Při ladění systému lze pružně bázi znalostí doplňovat, vypouštět některá pravidla nebo je měnit.

## K ZAPAMATOVÁNÍ



**Expertní systém je tvořen konkrétní bází znalostí a inferenčním mechanismem.**

---

Tento mechanismus a použitá reprezentace znalostí musí být založeny na některé teorii usuzování a řešení úloh dané problémové oblasti (daným expertem). Jedině tak může expertní systém simulovat příslušnou expertní činnost.

### 1.1.3 VYSVĚTLOVACÍ SCHOPNOST

Aby bylo možno přijmout závěry expertního systému, je nutné vědět, jak byly získány. Expertní systémy jsou proto vybaveny možnostmi informovat o průběhu inference a jednotlivé kroky podrobně objasnit. Vysvětlení předkládají ve formě vhodné pro uživatele a v rozsahu, který si uživatel určí. Vysvětlovací schopnost systému je nezbytná též při ladění a modifikacích báze znalostí.

### 1.1.4 CHARAKTERISTICKÉ RYSY



#### K ZAPAMATOVÁNÍ

Za charakteristické rysy expertních systémů se tedy považují:

**Oddělení znalostí a mechanismu pro jejich využívání.** Znalosti experta jsou uloženy v bázi znalostí odděleně od inferenčního mechanismu. To umožňuje vytvářet problémově nezávislé (prázdné) expertní systémy (expert system shells), kde jeden inferenční mechanismus může pracovat s různými bázemi znalostí, obdobně, jako lze stejným způsobem místo „klíče k určování bonity klienta“ procházet např. „klíč k určování hub“ nebo „klíč k určování prince“.

**Neurčitost v bázi znalostí.** V bázi znalostí jsou uloženy nejen exaktně dokázané znalosti, ale i nejrůznější heuristiky, které se např. expertovi osvědčily při rozhodování za dlouhou dobu jeho praxe. Zde se pak objevují pojmy jako „často“, „většinou“, které je potřeba kvantifikovat (např. v nějaké škále od „určitě ano“ přes „nevím“ až k „určitě ne“). Takovou znalostí s neurčitostí může například být „jestliže má pacient teplotu, obvykle je mu předepsán acylpyrin“; pacient totiž „často“ má teplotu v důsledku chřipky, ale „někdy“ může mít teplotu, protože je v pórúrazovém šoku.

**Neurčitost v datech.** Konkrétní data o daném případě bývají zatížena neurčitostí způsobenou nepřesně určenými hodnotami nebo subjektivním pohledem uživatele (odpovědi na míru jistoty v nějakém tvrzení). Může to být například odpověď „snad ano“ na dotaz, zda má pacient teplotu.

**Dialogový režim.** Expertní systémy jsou nejčastěji konstruovány jako tzv. konzultační systémy. Uživatel komunikuje se systémem způsobem „dotaz systému – odpověď uživatele“ obdobně, jako s lidským expertem. Tento systém práce byl do značné míry usnadněn nástupem osobních počítačů a s tím souvisejícím přechodem od dávkového zpracování na sálech výpočetních středisek k interaktivnímu zpracování přímo na pracovním stole uživatele.

**Vysvětlovací činnost.** Aby se zvýšila důvěra uživatelů v závěry a doporučení expertního systému, měl by systém poskytovat vysvětlení svého uvažování. Obvykle systém vysvětluje právě položený dotaz, znalosti relevantní k nějakému tvrzení, právě zkoumanou cílovou hypotézu, právě probíhající odvozování.

**Modularita a transparentnost báze znalostí.** Pro účinnost expertního systému je rozhodující kvalita báze znalostí. Modularita umožňuje snadnou aktualizaci báze znalostí, transparentnost umožňuje její snadnou čitelnost, srozumitelnost a kontrolu. Samotné vytváření báze znalostí probíhá iterativním způsobem při opakovaných konzultacích experta z dané problémové oblasti s odborníkem na tvorbuází, tzv. znalostním inženýrem, kdy je báze znalostí postupně „laděna“, až chování expertního systému (alespoň při konzultaci pro vzorové příklady) odpovídá představám experta.

Jiné druhy charakteristik, které bychom mohli spíše označit jako požadavky, uvádí:

**Vysoká spolehlivost** – systém musí být schopen odpovědět na stejné nebo vyšší úrovni než lidský expert.

**Přiměřená doba odezvy** – systém musí odpovídat v přijatelném čase. Vyřeší-li expert problém za hodinu, je systém podávající stejnou odpověď za rok nepoužitelný.

**Stabilita** – systém musí být stabilní a nehroutit se, jinak nebude používán.

**Srozumitelnost** – systém musí být schopný podávat vysvětlení o jednotlivých krocích odvozování, jinak bude nesrozumitelný a stane se z něj „černá skříňka“.

**Flexibilita** – systém musí jednoduše umožňovat přidávání, měnění a mazání znalostí.

## 1.4 Typy aplikací expertních systémů

Podle charakteru řešených úloh se aplikace dají rozdělit na:

- **diagnózu** – proces nalezení chyb či chybných funkcí systému (živého nebo neživého);
- **interpretaci** – analýza dat s cílem určení jejich významu;
- **monitorování** – průběžná (on line) interpretace signálů a dat a určení okamžiku, kdy je nutná intervence;
- **plánování** – nalezení posloupnosti akcí k dosažení cíle;
- **návrh** (design) – vytváření konfigurací objektů vyhovujících daným podmínkám;
- **predikce** – předpověď běhu budoucích událostí na základě modelu minulosti a současnosti.

Celkově lze expertní systémy (ES) rozdělit na dvě základní skupiny: diagnostické a generativní.

- **Diagnostické expertní systémy** zahrnují systémy, které slouží výběru jedné z předem daného pevného seznamu možností. Historicky prvními a stále základními příklady diagnostických expertních systémů jsou MYCIN pro diagnostiku jistých infekčních onemocnění a PROSPECTOR pomáhající při hledání nalezišť rud. Mezi současnými ekonomickými aplikacemi diagnostických expertních systémů je např. řada systémů odhadujících bonitu klienta, nebo vyhodnocujících investiční příležitosti. Zajímavý je systém FinCEN pomáhající při odhalování podezřelých transakcí.
- **Generativní expertní systémy** nevedou k výběru jedné z několika variant řešení, ale slouží k tvorbě návrhů přijatelných řešení. Možných řešení lze přitom potenciálně zkonstruovat nepřeberné množství. Systém jich ovšem generuje pouze omezený počet a vyhodnocuje je testováním. Kromě již zmíněného prvního expertního systému DENDRAL do této skupiny patří třeba i známý systém XCON pro konfiguraci počítačů podle uživatelských přání zákazníků.

### 1.1.5 PŘÍKLADY DIAGNOSTICKÝCH EXPERTNÍCH SYSTÉMŮ

Úkolem expertního systému MYCIN je diagnóza jistých druhů infekcí ve složitých pooperačních stavech a doporučení pro jejich medikamentózní léčbu. Uživatelem je lékař - chirurg, který nemusí být nutně expertem na infekční choroby (na rozdíl od experta, který byl zdrojem báze znalostí systému MYCIN). Během dialogu se systém MYCIN ptá ošetřujícího lékaře na stav, zdravotní historii pacienta a výsledky laboratorních testů. Na závěr systém MYCIN vydá diagnózu a doporučí vhodnou terapii.

Báze znalostí systému MYCIN obsahuje 451 pravidel tvaru: **Jestliže (konjunkce podmínek), pak (závěr) s jistotou (číslo).**

Každá podmínka nebo závěr je výrok tvaru (atribut) (objekt) (hodnota).

Jistota je číslo z jednotkového intervalu, které udává stupeň potvrzení závěru, jsou-li potvrzeny (pravdivé) podmínky pravidla. Stupeň jistoty je pravidlům subjektivně připisován expertem.

Následuje ukázka pravidla systému MYCIN.

Rule 85

If

The site of the culture is blood

The gramstain of the organism is gramneg

The morphology of the organism is a compromised host

Then

The identity of the organism is pseudomona aeruginosa

With certainty 0.6

Inferenční mechanismus systému MYCIN je založen na zpětném řetězení a modelu zpracování nejistoty, který interpretuje stupně jistoty jako tzv. míry důvěry.

Systém MYCIN vysvětluje své chování jak během konzultace (uživatel se může zeptat systému, proč položil některou otázku), tak po jejím skončení (uživatel se může zeptat systému, jak dospěl k některému závěru). Výsledky expertního systému MYCIN se při hodnocení v 90% shodovaly nebo byly stejně dobré jako rozhodnutí odborných lékařů.

Ze systému MYCIN byl separován inferenční mechanismus, který se stal základem „prázdného“ systému EMYCIN, jehož naplněním jinou bází znalostí vzniká nový expertní systém. Tak byl vyvinut např. systém PUFF pro interpretaci výsledků plicního vyšetření



spirometrem a systém **SACON** doporučující vhodnou strategii strukturní analýzy pevnosti konstrukce metodou konečných prvků.

Úkolem expertního systému **PROSPECTOR** je odhad věrohodnosti toho, že se na daném místě vyskytuje rudné ložisko typu reprezentovaného zadaným modelem. Uživatel pracuje s expertním systémem v dialogu, během něhož předkládá svá pozorování a je žádán o další informace.

Typická báze znalostí (o geologických charakteristikách měděných ložisek) je tvořena inferenční sítí o 94 vrcholech a 105 pravidlech. Inferenční mechanismus systému **PROSPECTOR** je založen na zpětném řetězení s modifikovaným Bayesovským schématem zpracování nejistoty (pseudopravděpodobnostní model).

U systému **PROSPECTOR** bylo provedeno srovnání jeho závěrů s usuzování geologa, který byl zdrojem expertních znalostí. Průměrný rozdíl mezi odhady pořízenými systémem a expertem byl kolem 10%. Systém **PROSPECTOR** se proslavil tím, že brzy po svém nasazení pomohl objevit cenné ložisko molybdenových rud.

Struktura expertního systému **PROSPECTOR** se stala základem pro jeden z prvních programových prostředků pro konstrukci expertních systémů – Advice Language / X.

Úkolem expertního systému **FinCEN** (Financial Crimes Enforcement Network) je odhad podezřelosti finanční transakce z hlediska praní špinavých peněz. Provozuje jej stejnojmenný úřad při americkém ministerstvu financí, který kontroluje všechny transakce nad 10 tis. \$ hlášené finančními institucemi. Expertní znalosti o podezřelosti transakcí jsou reprezentovány ve formě cca 2000 pravidel a tzv. rámců zachycujících stereotypní situace, což je výhodné při používání formulářů, na nichž jsou zprávy o jednotlivých transakcích zasílány a vkládány do databáze.

Příkladem expertního systému pro hodnocení bonity klienta je systém **FAST**. Obsahuje základní analýzu finančních charakteristik pomocí jednoduchého tabulkového kalkulátoru, databází kritérií vyjadřujících úvěrovou politiku konkrétní banky (preferenze, rizikové faktory, ...) a vlastní znalostní systém pro výsledné hodnocení založený na bázi cca 900 pravidel.

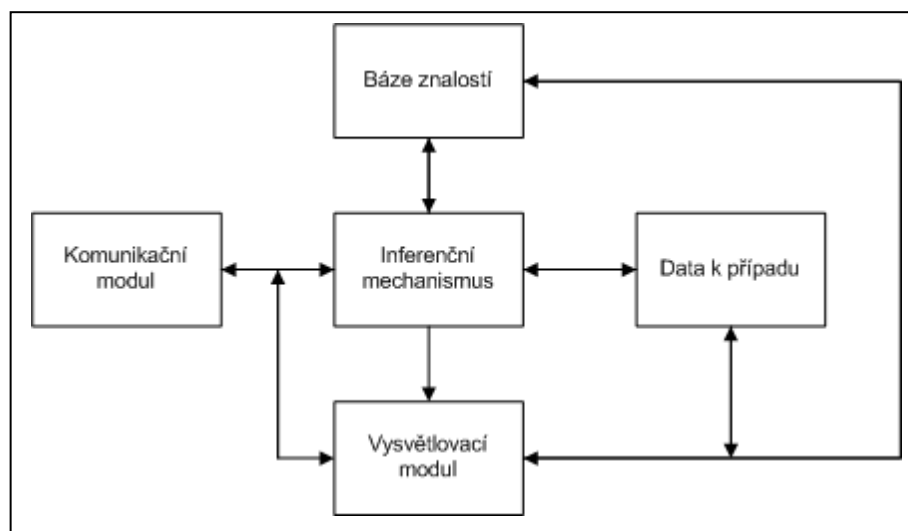
Systémem na analýzu trhu cenných papírů je **ESTA** (Expert Systems for Technical Analysis), který pomocí expertních pravidel vyhodnocuje klouzavé průměry cen za 10, 20, 40 a 75 dnů.

## 1.5 Struktura expertního systému

Expertní systém je tvořen dvěma základními částmi: bází znalostí a inferenčním mechanismem. Názornou analogii uvádí:

algoritmus + struktura dat = program

inference + znalosti = expertní systém.

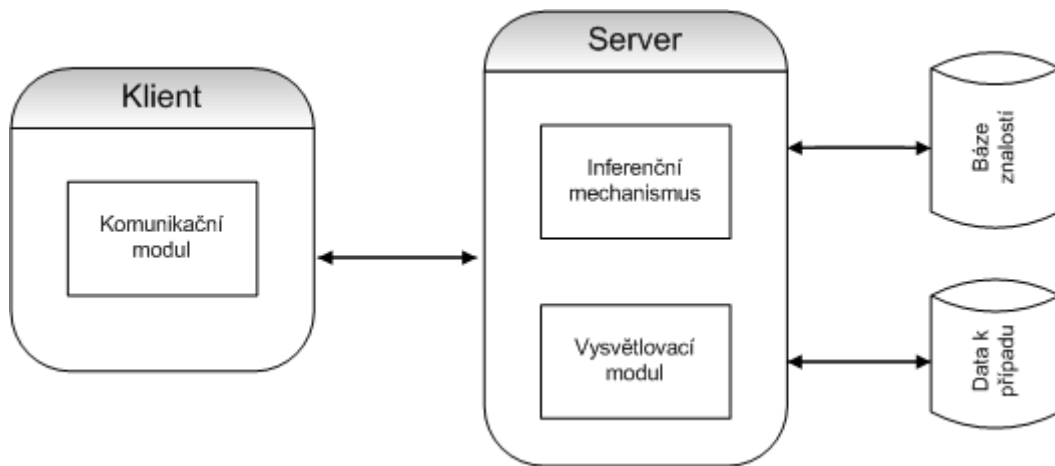


Obrázek 1: Obecné schéma expertního systému (Berka a kol. (1998)).

„V **bázi znalostí** jsou uloženy znalosti experta z dané oblasti, inferenční (odvozovací) mechanismus umožňuje tyto znalosti využívat při konzultaci pro konkrétní případ. Dalšími komponentami jsou **data ke konzultovanému případu**, **vysvětlovací modul umožňující (do jisté míry) zdůvodnit postup systému při odvozování a modul pro komunikaci s uživatelem**. Toto schéma (viz Obrázek 1) odpovídá „klasickému“ systému, který pracuje v dialogu s uživatelem.“ (Berka a kol. (1998))

Toto „klasické“ schéma můžeme jednoduše převést na schéma „klient server“ (se třístupňovou architekturou), ve kterém bývají zobrazovány „moderní“ programy (viz Obrázek 2).

Klíčovou složkou znalostního systému je **inferenční mechanismus** (odvozovací mechanismus). Je to programový modul, jehož úlohou je vybírat z báze znalostí ty znalosti, které jsou právě potřebné, interpretovat je a na jejich základě a v závislosti na okamžitém stavu řešení problému odvozovat nové, případně modifikovat už existující informace. Jeho dalším úkolem je řídit případný dialog s uživatelem – požadovat od něj podle potřeby a v závislosti na řídicí strategii data potřebná pro další postup řešení, pro jejichž odvození nemá systém potřebné znalosti v bázi znalostí.



Obrázek 2: Klient-server schéma expertního systému.

Pozn.: Inference (z latinského *infero* – vnášet) – usuzování, odvozování určitých výroků z jiných. V anglické literatuře se setkáme s termíny *inference* nebo *inference technique* – pro odvozování jakož to získání závěrů na základě znalostí které již známe, a *inference rule* nebo *rule of inference* čili pravidla usuzování (odvozování), které definují funkci pro odvozování závěrů (výroků, *conclusions*) na základě platnosti předpokladů (*premissis*). Pro psaní logických vztahů odvozování existuje jistý aparát, který ovšem překračuje rámec této studijní opory.

Pojem **znalosti** je vhodné chápat právě v rámci logiky usuzování a inferenčního mechanismu, který ji prakticky realizuje: za novou znalost považujeme tu informaci, která není daným inferenčním mechanismem odvoditelná z jemu již dostupných znalostí.

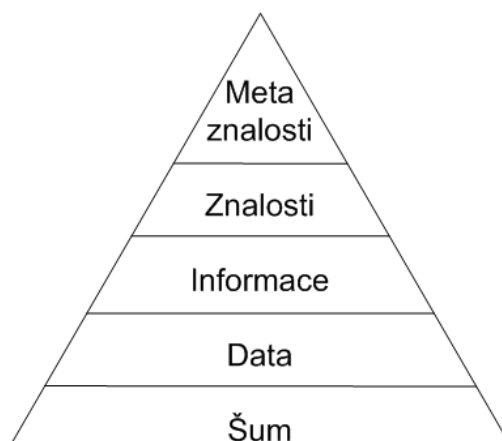
Důležitou komponentou znalostního systému je pracovní paměť, „tabule“. Jde o dočasnou pracovní databázi, ve které jsou v průběhu řešení udržována všechna data, která systém získal zvenku (od uživatele, jiných systémů nebo měřících přístrojů) nebo která odvodil z jiných dat na základě znalostí v bázi znalostí. U složitějších a pokročilejších systémů bývá její součástí i záznam posloupnosti všech důležitých aktivit systému, označovaný obvykle jako protokol.

Báze znalostí, inferenční mechanismus a pracovní paměť tvoří jádro znalostního systému. Při řešení úlohy jsou znalosti pomocí inferenčního mechanismu aplikovány na údaje v pracovní paměti.

Vyčlenění báze znalostí jako samostatné struktury a s tím spojená explicitní, symbolická a kompaktní reprezentace znalostí umožňuje zvýšit pohodlí uživatele při práci se systémy tohoto typu. Formalismy mohou být navrženy tak, aby byly srozumitelné a přijatelné i pro „neprogramátora“. Významnou roli přitom hraje vysvětlovací modul umožňující systému podat uživateli na požádání vysvětlení k danému případu, zdůvodnit dosažené výsledky, ke kterým systém dospěl, i postup, jakým k nim dospěl.

## 1.6 Znalosti

Jak již bylo výše řečeno, znalosti (resp. báze znalostí) tvoří jeden ze dvou základních prvků expertních systémů (druhým je inferenční mechanismus). Definovat pojem „znalost“ je velmi obtížné, navíc pro různé lidské činnosti se hodí jiná definice tohoto pojmu. „Proto se často vychází z předpokladu, že znalost je základním pojmem, který je každému jasný a srozumitelný, a že jej není třeba definovat.“ (Jiroušek (1995)) Představu o pojmu znalosti, jak mu rozumíme v kontextu expertních systémů, naznačuje Obrázek 3.



Obrázek 3: Hierarchie znalostí. Zdroj: (Giarratano a Riley (1993)).

Pojmy data, informace a znalost jsou mnohdy používána jako synonyma. Všimněme si jednoho přístupu k rozlišení a definování těchto pojmů.

Wiederhold (1986) definuje znalosti a data z hlediska budoucích informačních systémů jako komplementární (doplňující se) pojmy. Vychází z rozboru rozhodovacího procesu, v němž expert vybavený znalostmi zvažuje data relevantní pro daný problém a činí rozhodnutí. Znalosti expert získal vzděláním a zkušeností. Vybraná data představují informace.

Data obecně můžeme chápat jako zachycení vlastností okolní objektivní reality – popisují okolní svět (tzn. charakteristiky, míry, váhy, vlastnosti mohou být kvantitativní i kvalitativní). Pokud data vstoupí do rozhodovacího procesu a poslouží nám v tomto procesu (přispějí k snížení míry neuspořádanosti systému – subjektu, který se rozhoduje), pak tato data můžeme označit za informace.

Typické je, že **zdroj znalostí** (expert) a **zdroj informací** (sebraná data) jsou odlišné – zvláště v hospodářských aplikacích nemá expert čas sbírat a ověřovat data. Rozlišení dat a znalostí pro účely automatizace rozhodovacích procesů je proto následující:

- „Jestliže se můžeme spolehnout při sběru materiálu na automatický proces nebo úředníka, hovoříme o datech. Správnost dat vzhledem k reálnému světu může být objektivně verifikována srovnáním s jeho opakovaným pozorováním. Případně se stav světa změní a musíme se spolehnout na předchozí pozorování.“

- „Jestliže hledáme experta, který by poskytl materiál, potom hovoříme o znalostech. Znalosti obsahují abstrakce a generalizace objektivně verifikovány. Mnohé definice, nutné k organizaci systému, jsou také znalostmi; očekáváme od expertů definice, které jsou důležitými stavebními kameny pro další abstrakce, kategorizace a generalizace.“

Data odrážejí současný stav světa na úrovni instancí, takže obsahují mnoho detailů, jsou objemná a často se rychle mění. Znalosti se tak často nemění. Pojednávají o generalizacích, a proto se vztahují spíše k typům entit než k jejich instancím. Rozdíl mezi daty a znalostmi se jeví podstatný v úvahách o aktualizaci, chybách v datech a nejistotě znalostí. Aplikace na podporu řízení a plánování budou záviset zejména na znalostech adekvátních pro rozhodovací postup a výjimky v datech budou spíše ignorovat.

Nově získaná data mohou být v rozporu s existujícími znalostmi. Ve velkých databázích se vždy vyskytují chybné údaje, a proto je nutné rozpory pečlivě prověřovat. Avšak ani správná data, která jsou v rozporu se znalostmi, nemusí ihned vést k revizi znalostí. Jsou-li znalosti vybaveny faktory jistoty, je třeba spíše tyto faktory za pomoci automatické učící se procedury aktualizovat. Někdy konflikt mezi daty a znalostmi ukazuje na neúplnost báze znalostí a lze jej vyřešit doplněním dalších znalostí, které pokrývají nové případy.

Jeden z problémů vztahu dat a znalostí spočívá v tom, že data na vyšší úrovni zpracování (např. již agregovaná nebo jinak odvozená) se mohou jevit jako znalost vůči nižší úrovni zpracování.

## 1.7 Reprezentace znalostí

Reprezentace znalostí je spojena s hledáním modelů a datových struktur, které by umožnily reprezentovat významnou část znalostí a řešení daného problému v podobě báze znalostí, a potom takto reprezentované znalosti efektivně využívat při řešení problému počítačem.

Účelem vhodné reprezentace znalostí je kromě snadného vyhledávání a modifikace uložených znalostí hlavně možnost odvozovat z nich pomocí inferenčního mechanismu potřebné závěry. Zejména v tom se znalostní systémy liší od databázových, které se soustřeďují především na otázku vyhledávání. Další odlišnost spočívá v neurčitosti a neúplnosti, které jsou pro soustavy znalostí typické a musí na ně být při jejich reprezentaci brán zřetel.

Reprezentace znalostí je někdy považována za centrum zájmu výzkumu v umělé inteligenci, domníváme se však, že prvořadá je otázka způsobu odvozování, jemuž se musí reprezentace znalostí podříditi.

Na **reprezentaci znalostí** jsou **klaďeny často protichůdné požadavky**. Všimněme si například požadavku modularity báze znalostí, který je pro získávání znalostí zvláště významný. Při vytváření báze znalostí je důležité, aby použitá reprezentace dovolovala jed-

noduše upřesňovat znalosti, zejména je postupně rozšiřovat. Báze znalostí bude zřejmě modulární, bude-li složena z nezávislých, jednoduchých částí, které vyjadřují jakési atomické (na zvolené úrovni modelování dále nedělitelné) znalosti.

Na druhé straně však existují důvody proti takovému uspořádání znalostí. Jednotlivé znalosti totiž vyjadřují rozmanité závislosti, které mohou být od sebe svojí formou značně odlišné, avšak významově si mohou být blízké. Proto je vhodné použít neuniformní reprezentaci, postihující co nejlépe podstatu znalostí, a doplnit ji o možnost vyjádřit vazby mezi znalostmi. Požadavek volného sdružování příbuzných znalostí vychází z potřeby rychle vyvolávat jednotlivé znalosti. Jak je zřejmé, v konkrétních případech je nutno přistoupit k vhodnému kompromisu.

V expertních systémech jsou znalosti reprezentovány nejrůznějšími způsoby. Za „klasické“ lze považovat **logiku** (ať již výrokovou či predikátovou), **pravidla, rámce** (v programování známé spíše pod názvem objekty), **sémantické sítě** či **ontologie**. Zmíníme se zde stručně pouze o pravidlech, o jiných způsobech reprezentace znalostí se můžete více dozvědět např. v (Giarratano a Riley (1993)) či (Berka a kol. (1998)).

Soustava **produkčních pravidel tvaru „situace → akce“** je jedním z nejčastěji užívaných způsobů vyjádření báze znalostí. Pravidlo je vyvoláno tehdy, nastala-li podmínka v jeho situační části. Obecně není tvar a obsah pravidla nijak omezen, většinou jsou však v konkrétních realizacích dodrženy tyto zásady:

- vyhodnocení situační části spočívá ve srovnání podmínky se stavem pracovní paměti, která při tom nesmí být měněna;
- akční část představuje vykonání jednoduchých zásahů do pracovní paměti, a to na úrovni pojmů v ní používaných.

Pravidla tedy představují elementární akce systému. Vhodná volba těchto akcí usnadňuje práci inferenčního i vysvětlovacího mechanismu. Výhodou této reprezentace báze znalostí je její značná modularita a přirozená deklarativní sémantika pravidel.

Pravidla jsou zřejmě nejpoužívanější způsob reprezentace znalostí a jsou zároveň nejlépe pochopitelná pro lidské myšlení. Lidé často sami při řešení problémů používají výrazy IF THEN, jako například „JESTLIŽE prší, PAK si vezmu deštník“. Pravidla v expertních systémech mají právě podobu IF THEN pravidel. Dalším důvodem pro „oblíbenost“ těchto pravidel může být v tom, že tyto struktury se často používají v oborech, které mají blízko k expertním systémům. V programovacích jazycích se IF THEN struktura vyskytuje jako jeden z nejčastěji používaných příkazů, v logice není IF THEN struktura nic jiného než jiný výraz pro implikaci (a jako implikace jsou pravidla většinou vyhodnocována).

IF THEN struktura může být samozřejmě rozšířena do podoby IF THEN ELSE. Toto rozšíření slouží především pro zvýšení pohodlí uživatele při tvorbě báze. Je snadno vidět, že pravidlo

IF a THEN b ELSE c

lze zapsat jako dvě pravidla

IF a THEN b

IF NOT a THEN c.

V předpokladu (antecedentu, levé straně, ...) pravidla se mohou vyskytovat různé výrazy (opět záleží na systému), jako např. výrok, konjunkce výroků, disjunktivní normální forma, dvojice atribut – hodnota, trojice objekt – atribut – hodnota, atd. V závěru (sukcedentu, pravé straně, ...) pravidla mohou být v zásadě tytéž výrazy, v některých systémech však bývají omezeny oproti předpokladu (např. místo konjunkce výroků v předpokladu může být v závěru pouze jeden výrok).

Pravidla v bázi znalostí je možno znázornit v podobě tzv. AND/OR grafu. Zde uzly grafu představují výroky a orientované hrany představují pravidla. Konjunktivní vazba mezi výroky v předpokladu pravidla se znázorňuje obloučkem. Tedy sadě pravidel:

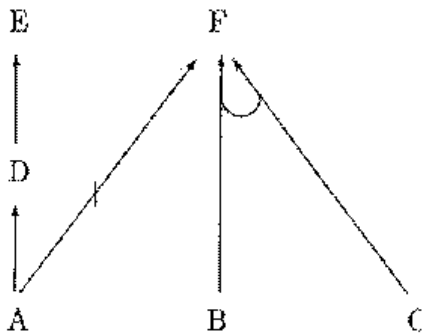
$A \rightarrow D$

$D \rightarrow E$

$\neg A \rightarrow F$

$B \ \& \ C \rightarrow F$

odpovídá AND/OR graf viz Obrázek 4.



Obrázek 4: AND/OR graf. Zdroj: (Berka a kol. (1998)).

## 1.8 Nepravidlové a hybridní expertní systémy<sup>1</sup>

Mezi nepravidlové reprezentace znalostí patří např. *rozhodovací stromy, sémantické sítě, Petriho sítě, rámce a objekty*.

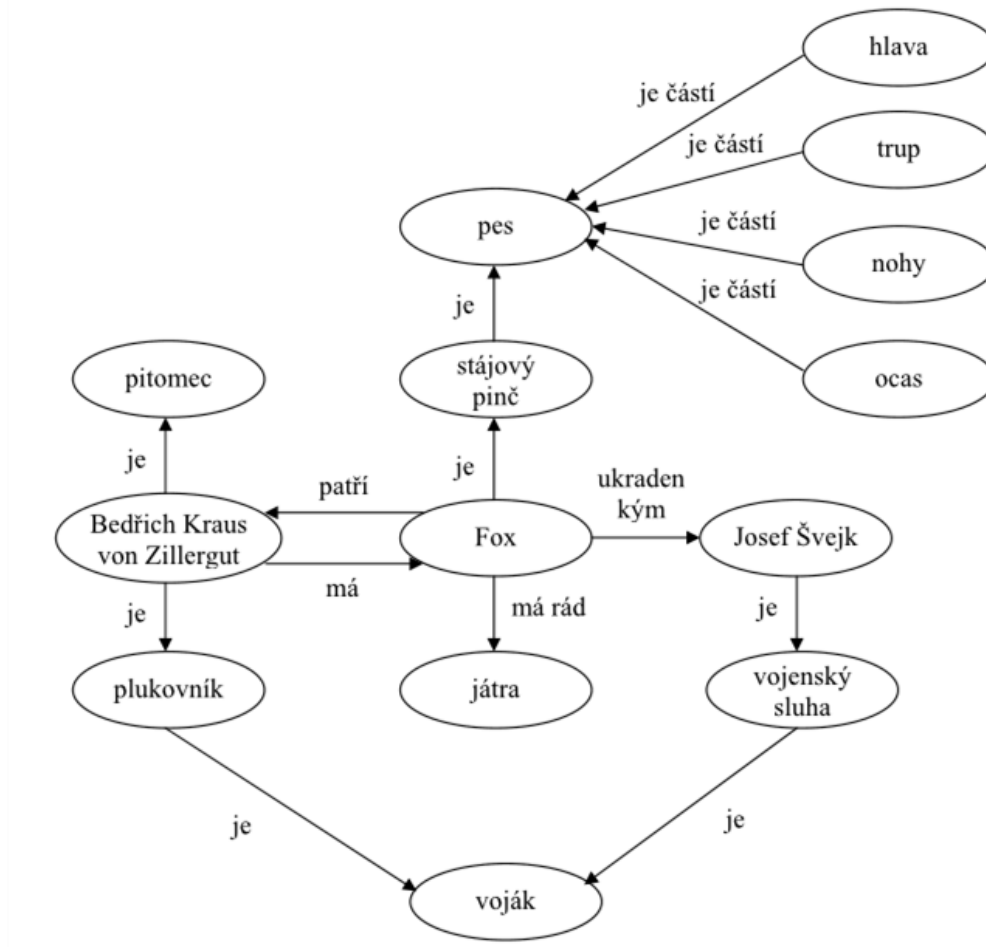
### 1.8.1 SÉMANTICKÉ SÍTĚ

Sémantická síť (*semantic net*) je ohodnocený orientovaný graf. Uzly reprezentují objekty a hrany představují vztahy mezi objekty. Místo pojmu sémantická síť se také používá pojem *asociativní síť*. Sémantická síť poskytuje vyšší úroveň porozumění akcím, příčinám a událostem, které se vyskytují v odpovídající doméně. To umožňuje úplnější usuzování znalostního systému o problémech z této domény. Sémantická síť umožňuje reprezentaci fyzikálních, kauzálních a taxonomických vztahů.

---

<sup>1</sup> Tato podkapitola byla převzata z (Dvořák, 2004).





Obrázek 5: Příklad sémantické sítě

Ukázka sémantické sítě je uvedena na Obrázek 5. Jsou zde zachyceny vztahy *je*, *je částí*, *má*, *patří*, *má rád*, *ukraden kým*. Zejména vztahy *je* (*is-a*, *ISA*), *je částí* (*part-of*), *má* (*has-a*) se v sémantických sítích vyskytují velmi často. Je nutné si dát pozor na interpretaci vztahu *is-a*, který může mít např. tyto významy: *je instancí*, *je prvkem*, *je podmnožinou*, *je podtřídou*, *je ekvivalentní s*. Např. *Fox* je instancí třídy *stájový pinč* a třída *stájový pinč* je podtřídou třídy *pes*. V některých systémech a knihách se *is-a* používá ve významu *je instancí*, zatímco vztah mezi třídami se označuje jako *a-kind-of* (*AKO*). Ale např. v systému *ART* *is-a* označuje vztah mezi třídami a pro označení vztahu *je instancí* se užívá *instance-of*.

Sémantická síť podporuje dědičnost a tranzitivitu. Např. třída *stájový pinč* dědí všechny vlastnosti třídy *pes* a tyto vlastnosti se tedy předpokládají také u všech jejích instancí. Příkladem tranzitivního uvažování je tato úvaha: *Bedřich Kraus von Zillergut* je *plukovník*, *plukovník* je *voják* a tedy *Bedřich Kraus von Zillergut* je *voják*.

#### Výhody sémantické sítě:

- explicitní a jasné vyjádření,
- redukce doby hledání (pro dotazy typu dědičnosti nebo rozpoznávání).

### Nevýhody:

- neexistence standardních definic jmen vazeb,
- neexistence interpretačních standardů,
- nebezpečí chybné inference,
- nebezpečí kombinatorické exploze (zvláště tehdy, je-li odpověď na otázku negativní).

### 1.8.2 RÁMCE

Rámce (*frames*) jsou struktury pro reprezentaci stereotypních situací a odpovídajících stereotypních činností (scénářů). Tento prostředek reprezentace vychází z poznatku, že lidé používají pro analyzování a řešení nových situací rámcové struktury znalostí získaných na základě předchozích zkušeností.

Rámce se pokoušejí reprezentovat obecné znalosti o třídách objektů, znalosti pravdivé pro většinu případů. Mohou existovat objekty, které porušují některé vlastnosti popsané v obecném rámci. Rámce jsou preferovaným schématem reprezentace v modelovém a případovém usuzování (*model-based reasoning, case-based reasoning*). Pro reprezentaci rámců se používají speciální jazyky, jako např. KRYPTON, FRL, KSL.

Rámec je tvořen jménem a množinou *atributů*. Atribut (*rubrika, slot*) může dále obsahovat položky (*links, facets*), jako např. aktuální hodnotu (*current*), implicitní hodnotu (*default*), rozsah možných hodnot (*range*).

Dalšími položkami slotu mohou být speciální procedury, jako např. *if-needed, if-changed, if-added, if-deleted*. Tyto procedury jsou automaticky aktivovány, jestliže nastanou příslušné situace. Např. v systému FLEX se používají následující typy událostmi řízených procedur:

- *launches* (aktivují se při vytváření instance rámce)
- *watchdogs* (aktivují se při přístupu k aktuální hodnotě slotu)
- *constraints* (aktivují se před změnou hodnoty slotu)
- *demons* (aktivují se po změně hodnoty slotu)

### Příklady rámců:

Rámec Majetek

Jméno: majetek

Specializace čeho: objekt

Typ: rozsah: (auto, loď, dům)

if-added: procedure PŘIDEJ\_MAJETEK

Vlastník: if-needed: procedure NAJDI\_VLASTNÍKA

Umístění: rozsah: (doma, práce, mobilní)  
Stav: rozsah: (chybí, špatný, dobrý)  
Záruka: rozsah: (ano, ne)

#### Rámec Auto

Jméno: auto  
Specializace čeho: majetek  
Typ: rozsah: (sedan, sportovní\_vůz)  
Výrobce: rozsah: (GM, Ford, Chrysler)  
Umístění: mobilní  
Kola: 4  
Převodovka: rozsah: (manuální, automatická)  
Motor: rozsah: (benzínový, naftový)

#### Rámec Janovo auto

Jméno: Janovo\_auto  
Specializace čeho: auto  
Typ: sportovní\_vůz  
Výrobce: GM  
Vlastník: Jan Chodec  
Převodovka: automatická  
Motor: benzínový  
Stav: dobrý

Záruka: ano

Rámce mohou být *generické* nebo *specifické*. Příkladem generických rámců jsou rámce *Majetek* a *Auto*, zatímco rámec *Janovo auto* je specifický. Rámec *Auto* reprezentuje celou třídu aut, kdežto v případě rámce *Janovo auto* se jedná o určité konkrétní auto.

Mezi rámci mohou existovat vztahy dědičnosti, které umožňují distribuovat informace bez nutnosti jejich zdvojení. Rámec může být specializací jiného obecnějšího rámce (vztah typu *specialization-of*) a současně může být zobecněním jiných rámců (vztah typu *generalization-of*). Vztah *specialization-of* může mít podobu *a-kind-of* (rámec je podrámcem jiného rámce) nebo *an-instance-of* (rámec je instancí jiného rámce). Ve výše uvedeném příkladu je rámec *Majetek* podrámcem rámce *Objekt*, rámec *Auto* je podrámcem rámce *Majetek* a rámec *Janovo\_auto* je instancí rámce *Auto*.

V systému FLEX existují např. tyto vztahy mezi rámci:

Rodič – potomek (*is-a, is-a-kind-of*):

Tento vztah může být typu 1:1, 1:n, n:1. Dědění některého atributu může být pro určitý rámec potlačeno.

Rámec – instance rámce (*is-an-instance-of*).

Tento vztah je typu 1:1. Přitom je navíc možné dědění nějakého specifického atributu od nějakého specifického rámce.

Vlastnictví rámce (atributem rámce může být jiný rámec).

#### **Výhody rámců:**

- snazší usuzování řízené očekáváním (na základě využití démonů),
- organizace znalostí (větší strukturovanost a organizace než v sémantických sítích),
- samořízení (schopnost rámců určit svou vlastní aplikovatelnost v dané situaci),
- uchovávání dynamických hodnot (ve slotech rámců); výhodné při simulaci, plánování, diagnostice, ... .

#### **Možné nevýhody:**

- potíže s odlišností objektů od prototypu,
- obtížné přizpůsobení novým situacím,
- obtížný popis detailních heuristických znalostí.

### **1.8.3 OBJEKTY**

Objekty podobně jako rámce sdružují deklarativní znalosti a procedurální znalosti. Objekt je programová struktura, obsahující jak *data*, tak *metody* (procedury), které s těmito daty pracují. Data objektu jsou přístupná pouze prostřednictvím metod objektu. Tato vlastnost se označuje jako *zapouzdření* (*encapsulation*). Objekt je *instance třídy*. Třída je skupina objektů, které mají stejné vlastnosti (datové složky) a stejné chování (metody).

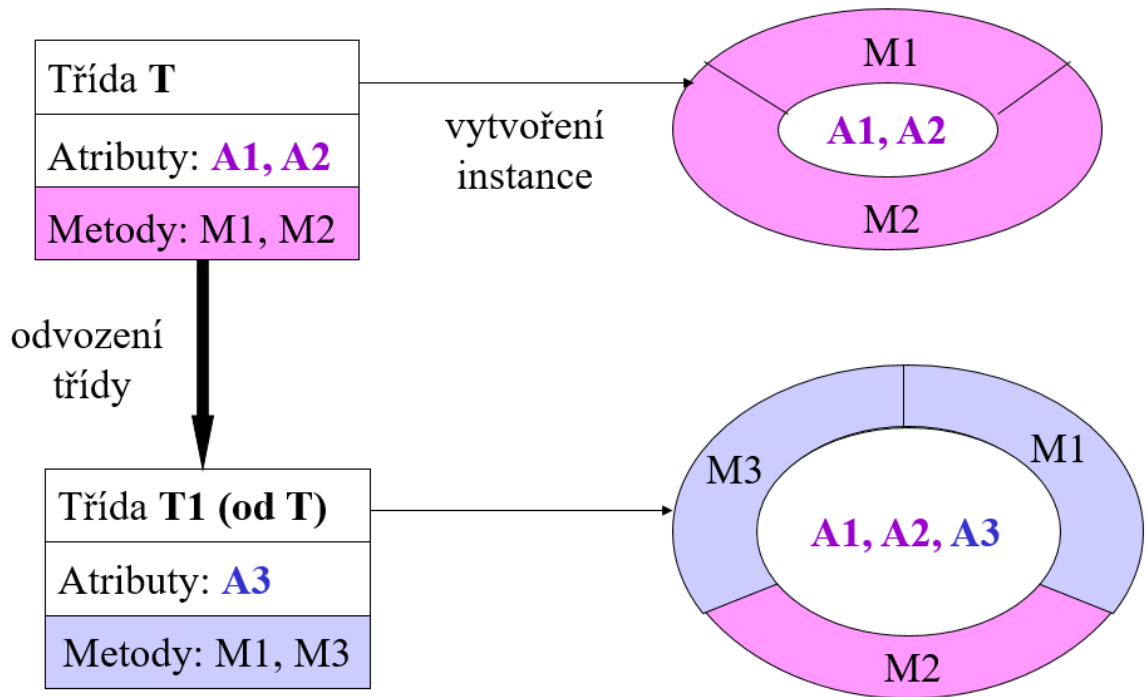
Mezi třídami mohou existovat tyto typy vztahů:

- Dědičnost:

Od jedné třídy (bázové, rodičovské) můžeme odvodit třídu jinou (odvozenou, dceřinou). Dceřinná třída dědí všechny složky své rodičovské třídy a k nim může přidat svoje vlastní. Zděděné metody je možno předefinovat. Objekt třídy *předek* může být v programu zastoupen objektem třídy *potomek* (může se jednat i o nepřímého potomka). Tento typ vztahu je ilustrován Obrázek 6.

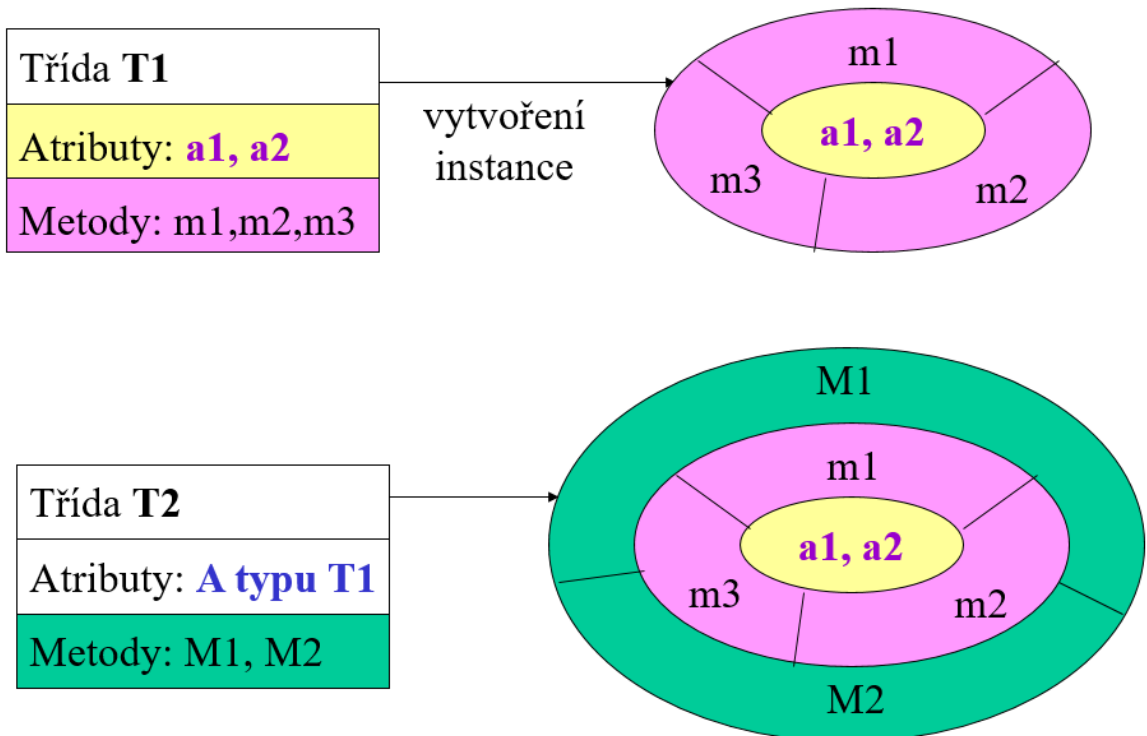
- Vlastnictví (skládání) :

Složkou třídy může být jiná třída. Skládání objektů ilustruje Obrázek 7.



Obrázek 6: Třídy, objekty a dědičnost

Na Obrázek 6 třída T1 dědí od třídy T atributy A1 a A2 a metodu M2. Dále definuje atribut A3, metodu M3 a předefinovává metodu M1.



Obrázek 7: Skládání tříd a objektů

Objekty spolu komunikují tak, že si navzájem posílají zprávy; obvykle to znamená, že jeden objekt (odesílatel zprávy) volá metodu jiného objektu (příjemce zprávy). Existují dva způsoby určení příjemce zprávy:

- časná vazba (*early binding*) – příjemce zprávy je určen v okamžiku kompilace.
- pozdní vazba (*late binding*) – příjemce zprávy je určen až za běhu programu; pomocí pozdní vazby se realizuje *polymorfismus* (mnohotvarost).

Jazyky pro objektové programování rozdělujeme na:

- jazyky čistě objektové (Smalltalk, Actor, CLOS, ...)
- jazyky podporující OOP, ale umožňující programovat i neobjektově (Borland Pascal, Object Pascal, C++, ...)

#### **Výhody objektů:**

- abstrakce
- zapouzdření (ukrývání informace)
- dědičnost
- polymorfismus
- znovupoužitelnost kódu

#### **Nevýhody (podobné jako u rámců):**

- jak se vypořádat s odchylkami od normy?
- jak zohlednit nové dosud neuvažované situace?

### **1.8.4 HYBRIDNÍ SYSTÉMY**

Zatímco u 1. generace expertních systémů byl v rámci jednoho systému používán pouze jeden způsob reprezentace znalostí, expertní systémy 2. generace obvykle používají hybridní (kombinované) reprezentace znalostí. Hybridní reprezentace, které jsou dostupné v nejčastěji používaných prostředcích pro vývoj ES, kombinují pravidlově, rámcově a objektově orientované techniky. Umožňují tzv. *modelový* přístup k tvorbě systému a usnadňují vývoj modelů.

Hybridní systémy, kombinující rámce a pravidla, používají např. rámce pro reprezentaci strukturálních znalostí a pravidla pro usuzování o těchto znalostech. Rámce také mohou být využity pro implementaci sémantických sítí.

Mezi hybridní systémy patří např.:

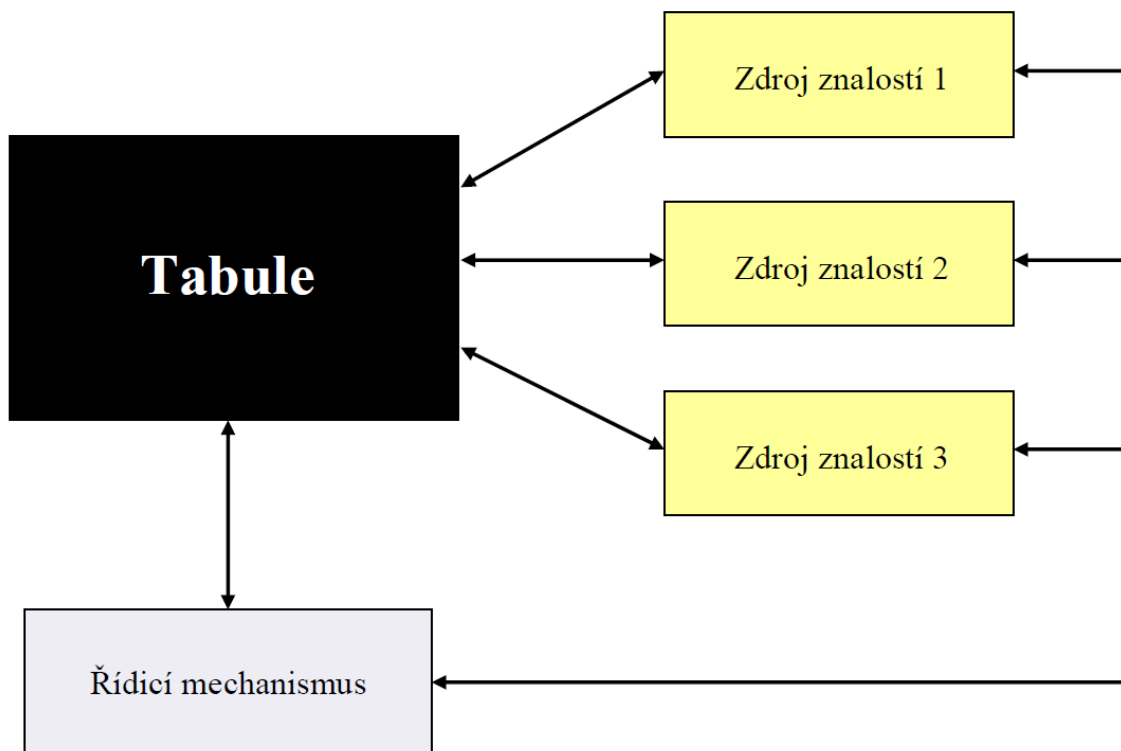
**ACQUIRE**      prázdný ES, pravidla, rámce a objekty

|                  |   |
|------------------|---|
| <b>CLIPS</b>     | programové prostředí, objekty a pravidla  |
| <b>FLEX</b>      | programové prostředí implementované v Prologu, pravidla a rámce, dopředné a zpětné řetězení   |
| <b>G2</b>        | objektově orientované prostředí, pravidla, modely a procedury, diagnostické a řídicí aplikace |
| <b>Rete++</b>    | programové prostředí, pravidla a objekty, dopředné a zpětné řetězení                          |
| <b>Rtworks</b>   | programové prostředí, objekty a pravidla  |
| <b>XpertRule</b> | programové prostředí, rozhodovací stromy a tabulky příkladů                                   |

### 1.8.5 SYSTÉMY TYPU TABULE

Architektura typu tabule (*blackboard architecture*) je příkladem implementace tzv. oportunistického usuzování (*opportunistic reasoning, opportunistic problem solving*). Při tomto způsobu znalosti nejsou striktně aplikovány v přímém nebo zpětném usuzování, ale jsou používány v nejpříhodnější době nejvhodnějším způsobem. Metoda usuzování je volena dynamicky v závislosti na tom, co systém naposledy zjistil. Tato forma usuzování je vhodná v aplikacích, kde znalosti o řešení problémů mohou být rozčleněny do nezávislých modulů, které pak kooperují při řešení problému.

Práci systému typu tabule je možno charakterizovat následujícím příkladem. Experti jsou ve třídě s tabulí. Komunikovat mohou pouze písemně přes tabuli a k dispozici mají pouze jeden kousek křídý. Řídicí mechanismus sleduje myšlenky expertů, vyhodnocuje možné příspěvky a rozhoduje, kdo dostane křídu.



Obrázek 8: Struktura systému typu tabule

Strukturu systému typu tabule ukazuje Obrázek 8. Hlavní komponenty jsou tyto:

- **Zdroje znalostí:** Obsahují dílčí znalosti potřebné pro řešení dílčích problémů (mohou to být individuální znalostní systémy); jsou přípustné různé reprezentace znalostí.
- **Tabule:** Společná databáze, přes niž zdroje znalostí komunikují.
- **Řídicí mechanismus:** Koordinuje zdroje znalostí; doporučuje akce, které mohou provést; určuje, které zdroje jsou nejvhodnější k tomu, aby přispěly k nalezení řešení; rozhoduje, co je aktuálně v popředí zájmu.

Činnost systému typu tabule probíhá v následujících krocích:

1. Zdroj znalostí provádí nějakou změnu na tabuli. Záznam o těchto změnách je zapsán do oblasti řídicích dat.
2. Každý zdroj znalostí zkoumá relevantní informace na tabuli, určuje, které akce by mohl provést a navrhuje tyto akce řídicímu mechanismu.
3. Řídicí mechanismus zkoumá informace z předchozích dvou kroků a určuje ohnisko zájmu.
4. Řídicí mechanismus vybere zdroj znalostí a objekt tabule. Systém se vrací na krok 1.



Kritéria ukončení jsou zajišťována při vytváření systému. Obvykle jsou zabudována do jednoho ze zdrojů znalostí.

Příkladem systému typu tabule je systém GBB (*Generic Blackboard Builder*), což je objektově orientované programové prostředí pro vývoj systémů typu tabule. GBB byl vytvořen v jazyku Common Lisp. GBB je používán nejen pro vývoj nových aplikací, ale také pro integrování existujících aplikací jako komponent sofistikovanějších aplikací. GBB představuje otevřené a rozšiřitelné prostředí, kde moduly mohou být psány v jakémkoli jazyku. K dalším rysům systému GBB patří:

- velmi účinné prostředky pro vyhledávání objektů,
- prostředky pro inteligentní řízení a integraci,
- grafické rozhraní pro tvorbu, ladění a používání komponent.

Systémy typu tabule jsou vhodné pro diverzifikované problémy (různé formy vstupních dat, nejasně definované cíle, použití mnohonásobných linií uvažování) a pro distribuovaná prostředí. K dalším výhodám patří hierarchická organizace, datová abstrakce, možnost odložení rozhodnutí, volné seskupování znalostí a jejich užití. Nevýhodami těchto systémů je to, že jsou drahé pro vytváření a používání a že je obtížné určit vhodné rozčlenění znalostí.

## 1.9 Infereční mechanismus

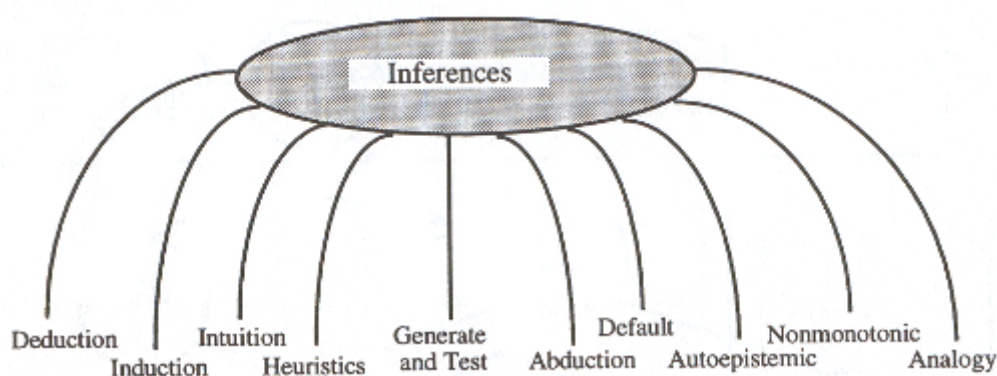
Zatímco u lidských expertů mluvíme spíše o usuzování, u expertních systémů pro získávání nových znalostí či informací používáme výraz inference nebo odvozování.

Metody inference, jak je popisuje (Giarratano a Riley (1993)), jsou:

- **Dedukce** – logické usuzování, při kterém závěr musí vyplývat z předpokladů.
- **Indukce** – odvozování ze speciálního případu na obecný.
- **Intuice** – bez teorie. Expertní systémy tento druh inference zatím neimplementují.
- **Heuristika** – pravidla „vycucaná z prstu“ založená na zkušenosti.
- **Generování a testování** – pokus a omyl, často používané pro efektivní plánování.
- **Abdukce** – usuzování z platných závěrů na předpoklady, které mohly tyto závěry způsobit.
- **Default** – pokud chybějí speciální znalosti, usuzuje se na základě obecných či běžných znalostí.
- **Vlastní znalosti**.
- **Nemonotónní usuzování** – předchozí znalosti mohou být chybné po získání nových poznatků.
- **Analogie** – odvození závěru na základě podobnosti s jinou situací.

Inferenční mechanismus organizuje používání báze znalostí při řešení zadané úlohy a komunikaci s uživatelem. Například prohledávání pravidlové báze znalostí může být řízeno mnoha strategiemi. Základní typy jsou:

- **Antecedentové řízení** (forward chaining, bottom-up). Vyhledávají se pravidla, jejichž antecedent (situační předpoklady) je v datech splněn. Tato pravidla se okamžitě provádějí a tím se data modifikují, doplňují o nové poznatky.
- **Sukcedentové řízení** (backward chaining, top-down). Vyhledávají se pravidla s vytčeným sukcedentem (akcí, konsekventem). K jejich antecedentům se opět hledají pravidla, která by je mohla potvrdit atd., až se dojde k prověřitelným pravidlům.
- **Agendové řízení**. Používají se váhové funkce, které neustále oceňují výhodnost vyvolání pravidel nebo jejich skupin.



Obrázek 9: Typy inference. Zdroj: (Giarratano a Riley (1993)).

Důsledkem použití některých pravidel může být dotaz na uživatele či sdělení uživateli. Po průchodu bázi znalostí formuluje inferenční mechanismus závěry.

## 1.10 Neurčitost

Neurčitost můžeme chápat jako nedostatek znalostí (nebo nepřesné znalosti) při řešení rozhodovacího problému. „Neurčitost je problém, neboť nám zabraňuje ve výběru nejlepšího řešení a může způsobit výběr špatného řešení.“ (Giarratano a Riley (1995) Neurčitost může způsobit nejrůznější typy chyb (viz Tabulka 1). Na druhou stranu, téměř veškeré znalosti a informace v našem světě jsou zatíženy určitou dávkou neurčitosti. Lidé jsou zvyklí s neurčitostí pracovat a rozhodovat se i při nepřesných informacích, je tedy na místě zabývat se otázkou, jak přenést neurčitost do expertních systémů.

Tabulka 1: Příklady typů chyb. Zdroj: (Giarratano a Riley (1993)).

| Příklad | Chyba | Důvod |
|---------|-------|-------|
|         |       |       |

|  |                                       |  |
|--|---------------------------------------|--|
| zavřít ventil  | nejednoznačnost (ambiguous)           | který ventil?                          |
| přepněte ventil 1  | neúplnost (incomplete)                | kam přepnout?                          |
| zavřete ventil 1   | nesprávnost (incorrect)               | správně je otevřít                     |
| ventil je zablokován   | špatně pozitivní (false positive)     | ventil není zablokován                 |
| ventil není zablokován   | špatně negativní (false negative)     | ventil je zablokován                   |
| přepněte ventil 1 do polohy 5                                      | nepřesnost (imprecise)                | přesně je 5,4                          |
| přepněte ventil 1 do polohy 5,4                                    | nesprávnost (inaccurate)              | správně je 9,2                         |
| přepněte ventil 1 do polohy 5,4, 4 nebo 0                          | nejistota (unreliable)                | chyba zařízení                         |
| ventil 1 je v poloze 5,4, 5,5 nebo 5,1                             | náhodná chyba (random error)          | statistická fluktuace                  |
| ventil 1 je v poloze 7,5   | systematická chyba (systematic error) | špatná kalibrace                       |
| ventil 1 není zablokovaný, protože nebyl zablokovaný nikdy předtím | nesprávná indukce (invalid induction) | ventil je zablokovaný                  |
| výstup je normální, proto je ventil 1 v dobrém stavu               | nesprávná dedukce (invalid deduction) | ventil je zablokován v otevřené pozici |

V expertních systémech může být neurčitost dvojího druhu. Je to jednak **neurčitost v expertních znalostech** a jednak **neurčitost v datech**.

Neurčitost v expertních znalostech se v těchto systémech modeluje tak, že jednotlivým pravidlům bývají přiřazeny prvky z nějaké (alespoň částečně uspořádané) struktury. obvykle se používá interval  $[-1,1]$ ,  $[0,1]$  nebo  $[0,\infty]$ . Pravidla potom mají tvar např.:

```
IF rozumný_záměr AND seriózní_klient THEN půjčit WITH WEIGHT
0.8
```

kde váha (weight) 0,8 vyjadřuje, do jaké míry je expert přesvědčen, že je splněn závěr (výrok „půjčit“), je-li si jist, že je splněn předpoklad (tj. současně splněny výroky „rozumný\_záměr“ a „seriózní\_klient“).

Naproti tomu neurčitost v datech přiřazuje určitou váhu počátečním uzlům, tj. např. rozumný\_záměr WITH WEIGHT 0.9 vyjadřuje poměrně velké přesvědčení uživatele o platnosti tohoto výroku (nikoli však jistotu).

K práci s neurčitostí jsou využívány různé přístupy, např. **pseudopravděpodobnostní přístup, míry důvěry a nedůvěry, Bayesovské sítě, Demster-Shaferův přístup, fuzzy logika** či **zobecnující algebraické pojetí**. Jejich popis můžete nalézt např. v (Berka a kol. (1998)) nebo (Giarratano a Riley (1993)). Neurčitost v systému NEST vychází právě ze zobecněného algebraického pojetí.



## KONTROLNÍ OTÁZKA

1. Jmenujte systém považovaný za první expertní systém.
2. Rozhodněte, zda je následující výrok pravdivý: „Expertní systém je systém rozhodování a odvozování založený na porovnávání přesně zjištěných a měřitelných požadavků uživatele.“
3. Vyberte, který z následujících úloh je vhodný pro zpracování expertním systémem: a) doporučení vhodného automobilu pro klienta, b) výběr nejvhodnějšího automobilu v rámci výběrového řízení.
4. Uveďte název expertního systému vyhodnocujícího v USA podezřelost peněžních transakcí.
5. Co je základním principem znalostního inženýrství?



## SHRNUTÍ KAPITOLY

Po prostudování této kapitoly zhruba víte, jak došlo ke zrodu disciplíny znalostního inženýrství. Umíte vysvětlit, co se skrývá pod pojmem expertní systém, a umíte rozlišit, kdo to je expert a znalostní inženýr. Dokážete pomocí vyložení základních principů a charakteristik expertního systému vysvětlit odlišnost od ostatních programů a především informačních systémů. Obeznamenali jste se s pojmem inference a její úlohou v expertních systémech.

Umíte rozlišit mezi pojmy data, informace a znalosti. Víte, jak lze reprezentovat znalosti. A dokážete vysvětlit, jak souvisí neurčitost se znalostmi.

Reprezentace znalostí pomocí IF THEN pravidel a jejich grafické znázornění jsou dovednosti, které dále využijeme při vlastní tvorbě báze znalostí – tedy pro tvorbu Vaší seminární práce.



**ODPOVĚDI**



1. DENDRAL
2. není
3. a)
4. FinCEN
5. Oddělení modelu znalostní od způsobu jejich používání



## **POZNÁMKY**

## 2 PROSTŘEDÍ NEST

### RYCHLÝ NÁHLED KAPITOLY



V této kapitole se čtenář seznámí s prostředím NEST, které slouží pro vytváření expertních systémů.

---

### CÍLE KAPITOLY



Cílem kapitoly je naučit čtenáře stáhnout, nainstalovat a spustit prostřední NEST, spustit si v tomto prostředí konzultaci a prohlédnout si její výsledky.

---

### KLÍČOVÁ SLOVA KAPITOLY



NEST, prázdný expertní systém, konzultace

---

## 2.1 HISTORIE NESTu

Jelikož systém NEST volně navazuje na systém SAK, je vhodné alespoň stručně charakterizovat i tento systém.

Systém Automatizovaných Konzultací (SAK) je prázdný expertní systém diagnostického typu.

Práce na systému SAK započala v 80. letech ve skupině kolem J. Ivánka. První implementace (v roce 1983) byla na počítači Wang v jazyku Basic (systém SAK OPTIMALI). V letech 1987 – 1991 byly postupně implementovány dvě verze na počítačích IBM PC pod DOSem, jedna v jazyku TurboProlog (J. Ivánek, J. Ferjenčík, J. Švenda), druhá v jazyku TurboBasic (J. Ivánek, J. Ferjenčík, P. Berka). Do roku 1995 byly prováděny ještě některé drobnější úpravy ve verzi TurboBasic.

Systém používá pro reprezentaci znalostí výroky a pravidla. Kromě klasických výroků je v systému zavedena koncepce sdružených výroků (výroky, které spolu fakticky souvisí, ale nemusí se vylučovat), a numerické veličiny, jejichž hodnoty lze získávat i z vnějších procedur.



## K ZAPAMATOVÁNÍ

Pravidla mají tvar

$$K: A \Rightarrow S (w)$$

kde kontext  $K$  je výrok, antecedent  $A$  je tvořen elementární konjunkcí (tj. konjunkcí výroků nebo jejich negací), sukcedent  $S$  je výrok, který se nevyskytuje v  $A$  a  $w$  je váha pravidla (v normalizovaném tvaru leží v intervalu  $[-1;1]$ ).

---

Pravidla mohou mít nejvýše čtyřčlenný antecedent. Systém umožňuje vyjadřovat neurčitosti v pravidlech i v odpovědích uživatele, v systému je implementován standardní a logický inferenční mechanismus. Konzultace probíhá v režimu zpětného řetězení; kromě odvození závěrů nabízí systém i standardní vysvětlování How, Why, What if.

Systém NEST volně navazuje na systém SAK, dále jej rozvíjí a převádí do podoby aktuálních počítačových aplikací. Práce na návrhu nového systému započaly v polovině 90. let (P. Berka, V. Svátek):

- 1995 – 97: specifikace systému NEST;
- 1996: konzultační modul v rámci systému pro hodnocení bonity klienta banky implementovaný komerční firmou.

V květnu 2003 začíná práce na implementaci systému NEST v podobě, kterou popisují následující kapitoly.

## 2.2 NEST

Program NEST je volně přístupný na osobních internetových stránkách spoluautora expertního systému V. Laše (vladalas.info). Se svolením autorů je program NEST, respektive instalační balík dostupný v rámci předmětu Expertní systémy (vždy v příslušném aktuálním akademickém roce).

### 2.2.1 POPIS INSTALAČNÍHO BALÍKU A VLASTNÍ INSTALACE

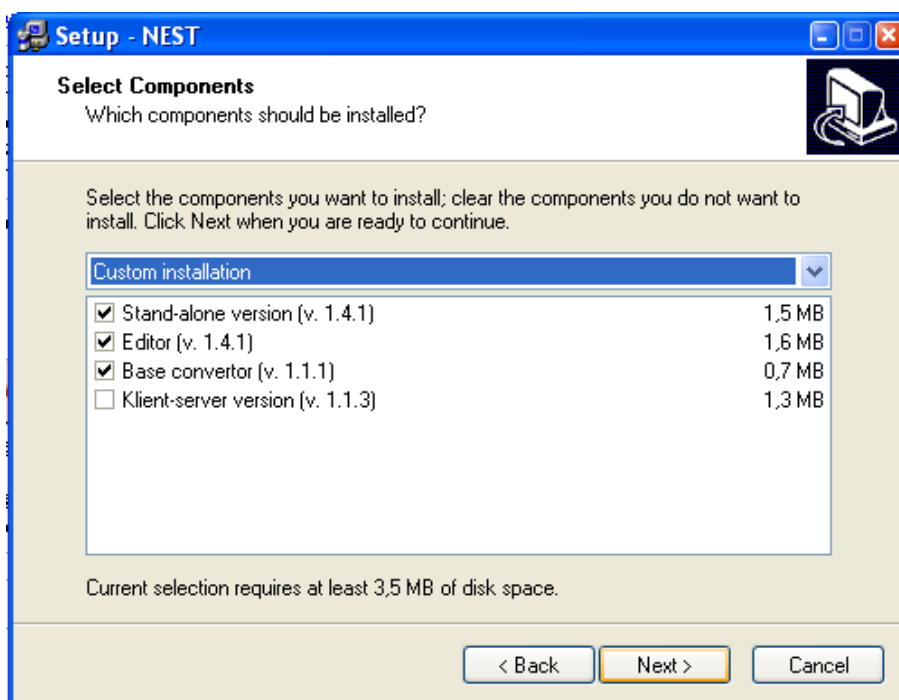
Jak již bylo zmíněno, instalační balík NEST (verze pro lokální počítač, nikoli síťová verze) je dostupná studentům OPF (příslušný vyučující v aktuálním semestru studentům předvede, kde tento balík bude dostupný) nebo také z adresy <http://vladalas.info/files/nest.zip>.



Instalační balík je zip archiv – tedy po úspěšném stažení byste měli najít na vámi zvoleném místě určeném pro stahování souboru soubor *nest.zip*. Balík obsahuje skupinu souborů bez instalátoru, a tedy po rozbalení můžete program respektive programy spouštět a používat. Studenti předmětu Expertní systémy, které je vypisováno v rámci navazujícího studia, prošli kurzem Informatika pro ekonomy I, a proto předpokládáme znalost ovládnání operačního systému, práci s archívy atp.

Po úspěšném rozbalení nalezneme mezi soubory dva programy:

- NEST.exe - software pro provádění konzultací nad zadanou (zavedenou, nahranou) bází znalostí.
- NEST\_Editor.exe - software pro tvorbu vlastní báze znalostí.



**Obrázek 10: Okno instalátoru s volbou jednotlivých složek programu.**

Na internetu nebo ve vlastním instalátoru se můžete setkat s následujícími označeními jednotlivých komponent NESTu:

- Stand-alone version – program pro provádění konzultací.
- Editor – editor pro tvorbu a úpravu bází znalostí.
- Base convertor – program pro konverzi bází znalostí.
- Klient-server version – NEST pro vzdálený přístup – síťová verze.

Instalaci můžeme považovat za dokončenou až po stáhnutí existujících a fungujících bází znalostí, na kterých si můžeme později ukázat spouštění konzultací a úpravu bází

v NEST editoru. Báze znalostí jsou ke stažení ze stejných zdrojů jako vlastní NEST a to konkrétně:

- [Princ.xml](#) - výuková báze imaginárního příkladu určování, zda-li konzultující osoba je či není „princ“ (obsahuje 1 cílový výrok, 7 pravidel),
- [Nemoce.xml](#)- malá ilustrační báze s diagnostikou nachlazení, TBC nebo chřipky (obsahuje 3 cílové výroky, 14 pravidel),
- [Uver.xml](#) - báze vytvořená na základě dolování z imaginárních dat (data miningu) s diagnostikou klienta, zda-li by mu mohl být poskytnut úvěr či nikoli (obsahuje 1 cílový výrok, 8 pravidel),
- [OPTCS.xml](#) - volba vhodné matematické metody rozhodování (obsahuje 53 cílových výroků, 402 pravidel),
- [TEAM.xml](#) - báze znalostí sloužící ke zdokonalení práce řešitelského týmu (obsahuje 13 cílových výroků, 84 pravidel).

| Název       | Velikost | Typ                              | Změněno          |
|-------------|----------|----------------------------------|------------------|
| answers.dtd | 1 kB     | Soubor DTD                       | 8.9.2003 16:44   |
| base.dtd    | 5 kB     | Soubor DTD                       | 28.11.2004 16:11 |
| HELPA.G     | 87 kB    | Soubor nápovědy                  | 13.12.2003 22:05 |
| helpcz      | 54 kB    | Kompilovaný soubor nápovědy HTML | 20.10.2003 23:42 |
| HELPCZ      | 89 kB    | Soubor nápovědy                  | 23.10.2003 11:04 |
| language    | 63 kB    | Dokument ve formátu XML          | 15.5.2005 15:56  |
| Nemoce      | 11 kB    | Dokument ve formátu XML          | 28.11.2007 16:04 |
| Nest        | 1 246 kB | Aplikace                         | 26.8.2006 17:09  |
| Nest_Editor | 1 464 kB | Aplikace                         | 8.10.2006 14:33  |
| OPTCS       | 197 kB   | Dokument ve formátu XML          | 28.11.2007 16:04 |
| Princ       | 8 kB     | Dokument ve formátu XML          | 28.11.2007 16:04 |
| TEAM        | 44 kB    | Dokument ve formátu XML          | 28.11.2007 16:04 |
| uver        | 7 kB     | Dokument ve formátu XML          | 28.11.2007 16:04 |

Obrázek 11: Adresář po dokončené instalaci včetně ukázkových bází.

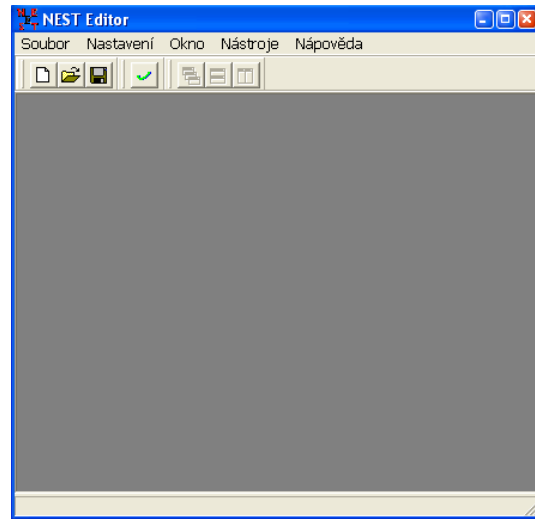
## 2.2.2 NASTAVENÍ

Je velmi důležité se věnovat také krátce nastavení programů NEST a NEST editor. Po spuštění (úvodní obrazovky programů dokumentují Obrázek 12 a Obrázek 13) v menu obou programů vybereme položku Nastavení/Nastavení. V následujícím okně pak můžeme

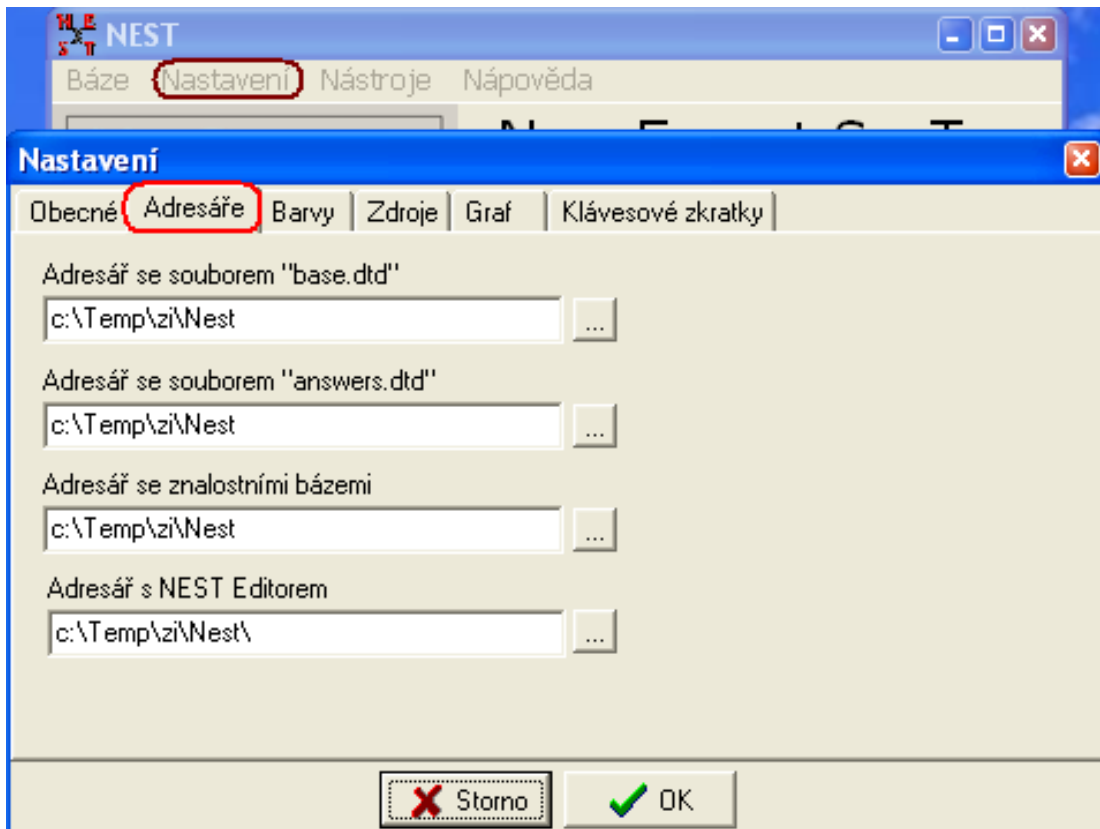
uživatelsky nastavit nejrůznější parametry programu (customization), jako jsou jazyk, počet zobrazovaných desetinných míst, barvy, nastavení okna pro znázornění grafu, klávesové zkratky a nastavení adresářů (viz Obrázek 14).



Obrázek 12: Úvodní obrazovka programu NEST.



Obrázek 13: Úvodní obrazovka programu NEST editor.



Obrázek 14: Okno Nastavení v NESTu.

Na tomto místě bychom rádi poznamenali, že NEST je program určený především pro akademické účely, proto také nebyl kladen důraz na vzhled, ale na funkčnost programu

zaměřenou na tvorbu znalostní báze, porovnávání výsledků konzultací při volbě různých druhů práce s neurčitostí atp. **NEST není komerčním produktem a jeho využití mimo výukové účely je nutno konzultovat s autory.**

### 2.2.3 NEST – ZÁKLADNÍ MOŽNOSTI A SPUŠTĚNÍ KONZULTACE

Program NEST je prázdný expertní systém, který obsahuje inferenční mechanismus a vysvětlovací modul (viz Základní principy znalostního inženýrství). Program NEST je tedy především program, který uživateli poskytuje grafické rozhraní (GUI) pro: nahrávání existujícíchází znalostí, jejich spuštění, nastavení přístupu zpracování neurčitosti, vedení konzultace (způsobu získávání dat od uživatele), vyhodnocení a doporučení cílového výroku s možností vysvětlení závěrů.

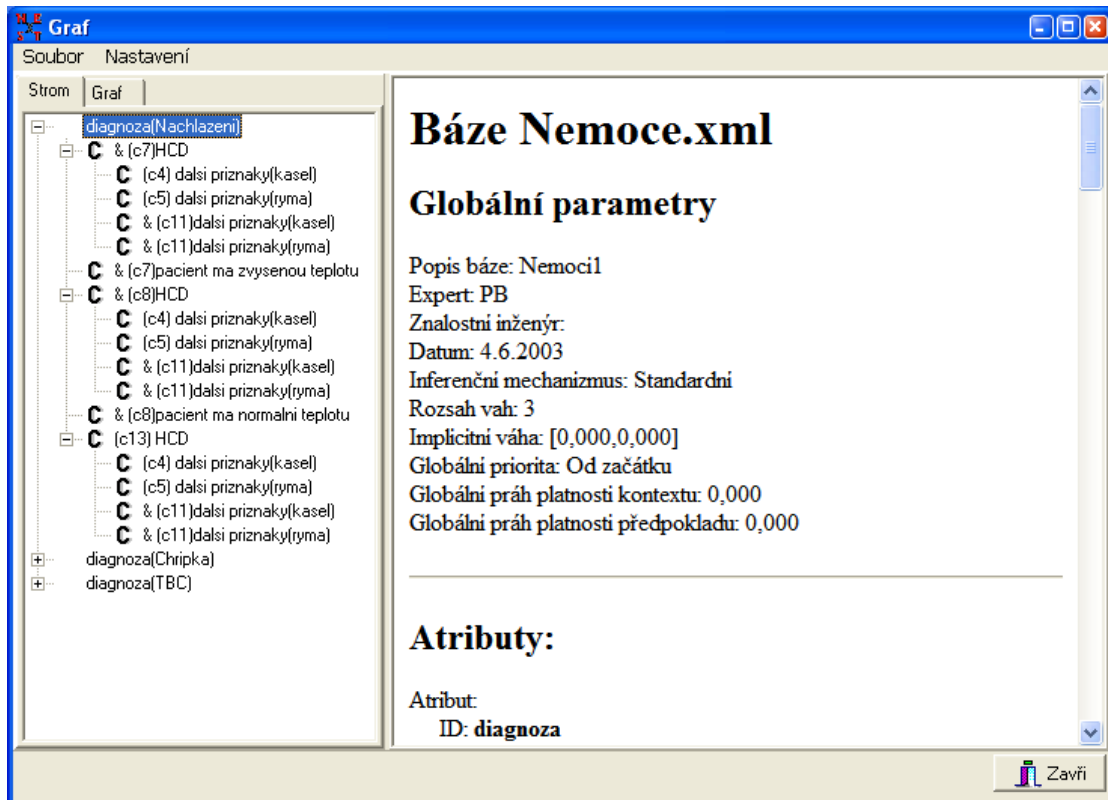
| Globální vlastnosti                 |             | Atributy        |    | Výroky             |    |
|-------------------------------------|-------------|-----------------|----|--------------------|----|
| JmenoLabel                          | Nemoce.xml  | Celkem          | 7  | Celkem             | 11 |
| Popis báze                          | Nemocí      | Binární         | 4  | Binární            | 4  |
| Expert                              | PB          | Jednoduchý      | 0  | Jednoduchý         | 0  |
| Znalostní inženýr                   |             | Množinový       | 2  | Množinový          | 5  |
| Datum                               | 4.6.2003    | Numerický       | 1  | Numerický          | 2  |
| Rozsah vah                          | 3           | dotaz           | 4  | dotaz              | 6  |
| Globální práh platnosti kontextu    | 0,000       | mezilehlý       | 2  | mezilehlý          | 2  |
| Globální práh platnosti předpokladu | 0,000       | cíl             | 1  | cíl                | 3  |
| Inferenční mechanismus              | Standardní  | osamocený       | 0  | osamocený          | 0  |
| Implicitní váha                     | Irelevantní | Zdroje          | 0  | Akce               | 0  |
| Globální priorita                   | Od začátku  | Akce            | 0  |                    |    |
|                                     |             | Pravidla        |    | Další              |    |
|                                     |             | Celkem          | 14 | Kontexty           | 0  |
|                                     |             | Apriorní        | 0  | Integrovní omezení | 0  |
|                                     |             | Logické         | 0  |                    |    |
|                                     |             | Kompozicionální | 14 |                    |    |
|                                     |             | Akce            | 0  |                    |    |

Obrázek 15: Statistika báze znalostí „Nemoce.xml“.

Zavedení báze znalostí nebo její nahrání realizujeme pomocí úvodní obrazovky NEST (viz Obrázek 12). Pomocí tlačítka „Nahraj bázi“ nebo z menu „Báze/Nahraj bázi“ najdeme na disku bázi, se kterou chceme dále pracovat (např. Nemoce.xml). Po otevření báze se na úvodní obrazovce objeví, pokud v samotné bázi tyto údaje vyplněny jsou, několik nových údajů:

- Vpravo od tlačítka „Nahraj bázi“ v rámečku „Aktuální báze“ se zobrazí název souboru aktuální zavedené báze znalostí (v našem případě „Nemoce.xml“).
- V rámečku „Popis báze“ nalezneme popis báze znalostí (v našem případě vidíme text „Nemocí“).
- V políčku „Expert“ nalezneme jméno experta, člověka odborníka na danou problematiku, který nám poskytl znalosti z dané oblasti. Znalosti také můžeme čerpat analýzou odborné literatury (v našem případě vidíme iniciály „PB“).

- V políčku „**Znalostní inženýr**“ nalezneme jména specialistů, kteří formalizovali a zaznamenali znalosti získané od expertů či z literatury, dat a jiných zdrojů do báze znalostí (v našem případě je políčko prázdné).
- V políčku „**Datum**“ nalezneme termín vytvoření báze znalostí (našem případě „4.6.2003“).



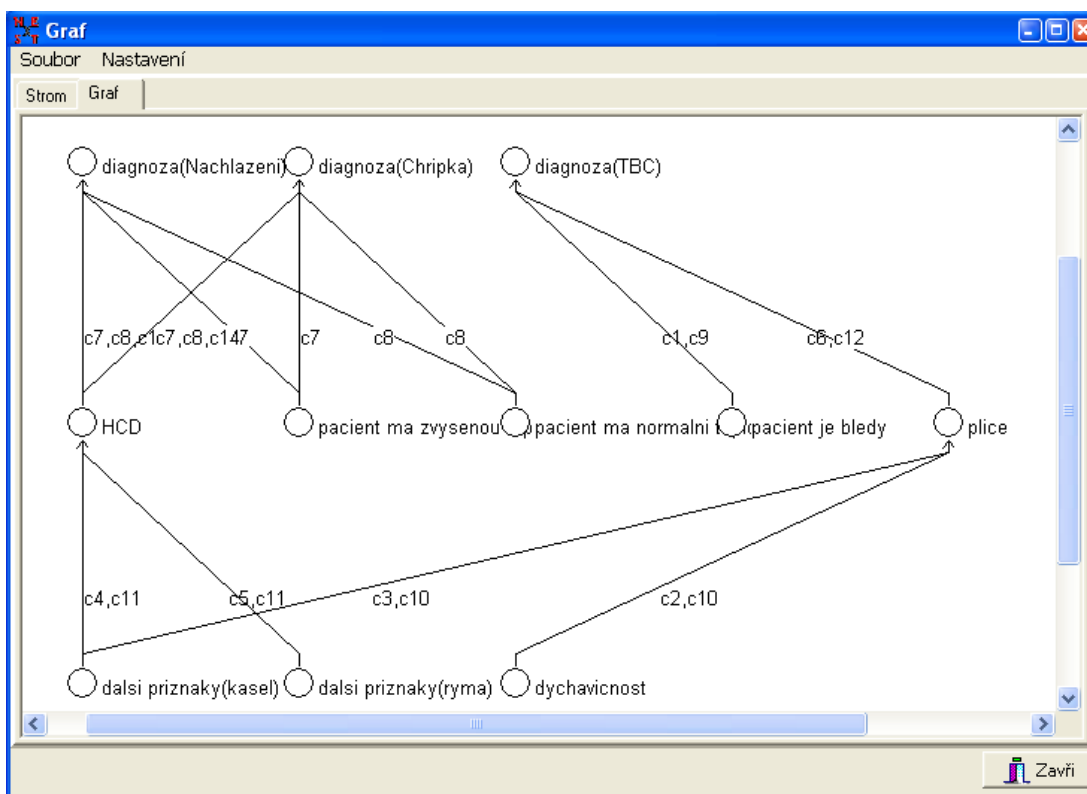
Obrázek 16: Zobrazení stromu pravidel pomocí okna Graf.

Na úvodní obrazovce (viz Obrázek 12) programu NEST dále vidíme:

- tlačítko „**Zavřít**“, odpovídá volbě z menu „**Báze/Konec**“ nebo stisknutí klávesové zkratky **ALT+F4**. Tímto způsobem ukončíme program.
- Tlačítko „**Konzultuj**“, odpovídá volbě z menu „**Báze/Konzultuj**“, pokračujeme ke spuštění konzultace.
- Volbou z menu „**Báze/Statistika**“ vyvoláme okno s podrobnějším popisem aktuální báze znalostí. Údaje pro bázi „Nemoce.xml“ dokumentuje Obrázek 15.
- Volba z menu „**Nastavení/Nastavení**“ je popsána již výše.
- Volbou „**Nástroje/NEST editor**“ spustíme program NEST editor – samozřejmě pokud máme správně nastavenou cestu ke složce, ve které se editor nachází (viz Obrázek 14).
- Volbou „**Nástroje/Graf báze znalostí**“ vyvoláme okno, kde:

- V pravém sloupečku nalezneme podrobný popis báze včetně pravidel a vah. Šířku sloupce můžeme pohodlně upravovat tažením myši za levý okraj (viz Obrázek 16).
- V levé části okna na záložce „**Strom**“ si můžeme nechat zobrazit – „rozklikat“ strom pravidel (viz Obrázek 16).
- V levé části okna na záložce „**Graf**“ je nám k dispozici graf aktuální báze znalostí (viz Obrázek 17).
- Z menu volbou „**Sobor/Ulož graf jako obrázek**“ můžeme exportovat graf do formátu **BMP**.
- Funkční klávesou **F1** nebo z menu „**Nápověda/Nápověda**“ vyvoláme nápovědu k program.

Informace o programu pak zjistíme volbou „**Nápověda/O programu**“.



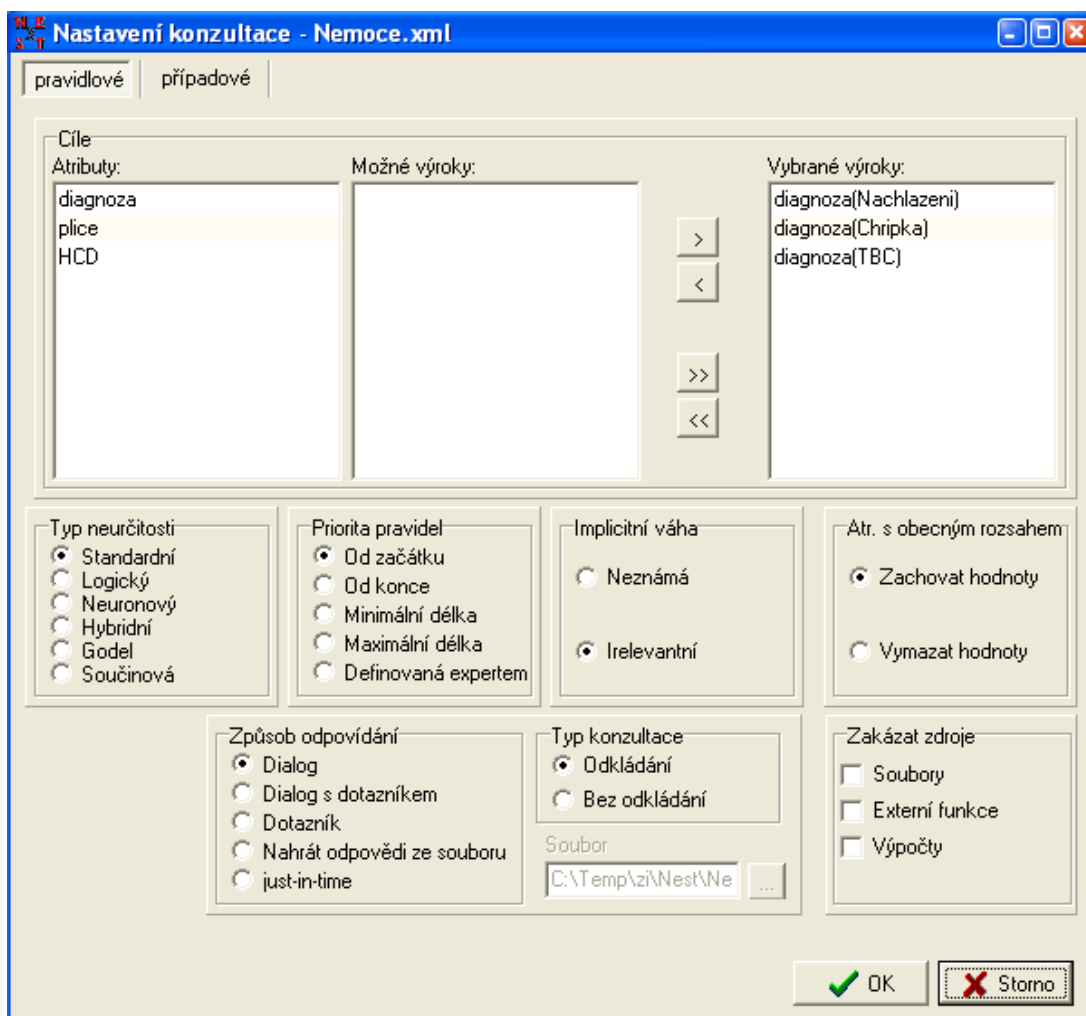
**Obrázek 17: Zobrazení grafu pomocí okna Graf.**

Nyní tedy po zmáčknutí tlačítka konzultuj se přesuneme do okna „**Nastavení konzultace**“ (viz Obrázek 18). Možnosti nastavení pro naše účely není potřeba dále komentovat stiskneme proto tlačítko „**OK**“ a tím konečně spustíme konzultaci (stále pracujeme z bázi „Nemoce.xml“).

## 2.2.4 MOŽNOSTI OKNA DOTAZ A PRŮBĚH KONZULTACE

Postupujete-li stále dle textu, pak nyní budete zadávat odpovědi na dotazy v rámci konzultace. Každému dotazu odpovídá jedno okno. Dle typu dotazu se okno „Dotaz“ trochu liší. Společné pro všechny dotazy zůstává (viz např. Obrázek 19):

- Jméno atributu – Obrázek 19 „pacient je bledý“
- Komentář atributu – pod jménem atributu, slouží uživateli vykonávajícímu konzultaci k upřesnění (pokud to není zřejmé se samotného jména atributu) nač se systém ptá, co a jakým způsobem má uživatel hodnotit. V našem případě (Obrázek 19) je políčko prázdné.
- V levém sloupci nahoře jsou tlačítka:
  - „**Proč**“ – zobrazí okno s příslušnou **výsečí grafu** nebo **stromu** (podobně jako okno Obrázek 16 nebo Obrázek 17) dokumentující proč systém potřebuje zjistit hodnotu váhy příslušného výroku a na jaký další atribut to má vliv.
  - „**Průběžné výsledky**“ – otevře okno podobné oknu „Výsledky pravidlového odvozování“, jehož možnosti jsou popsány v kapitole 2.2.6.
  - „**Uložit odpovědi**“ – tlačítko pro ukládání odpovědí i v průběhu konzultace více v kapitole 2.2.5.
- V levém sloupci dole jsou tlačítka:
  - „**Jistě ano**“ – nemusíme vypisovat váhu odpovědi automaticky, po stisknutí se zaznamená maximální, tj. v našem případě váha 3.
  - „**Irrelevantní**“ – nahrazuje zapsání váhy 0.
  - „**Jistě ne**“ - nemusíme vypisovat váhu odpovědi automaticky po stisknutí se zaznamená minimální tj. v našem případě váha -3.
  - „**Neznámý**“ – slouží pro odpověď ve smyslu nevím, kterou nelze pomocí váhy vyjádřit.
  - „**Odložit odpověď**“ – nechá otázku nezodpovězenou na pozdější dobu.
  - „**Potvrdit odpověď**“ – potvrzení odpovědí pokud zadáváme hodnoty z klávesnice a přesun na další dotaz.



Obrázek 18: Okno nastavení konzultace.

Vaše odpovědi budou subjektivně hodnotit pravdivost výroků příslušných atributů. Můžeme se setkat se čtyřmi typy dotazů:

- **Binární** (viz Obrázek 19) – hodnotíme pouze pravdivost výroku v rámci rozsahu vah.
- **Jednoduchý** (viz Obrázek 20) – vybíráme **jeden** z výroků nabízených v padací liště (v našem případě „vysoký příjem“ nebo „nízký příjem“); tento výrok opět hodnotíme vahou, vlevo od výběru.
- **Množinový** (viz Obrázek 21) – z množiny přípustných výroků může vybrat **jeden či více** výroků. Výrok vždy vybereme zakliknutím (podbarví se modře) ve sloupečku „**Přípustné hodnoty**“. Přetáhneme jej do sloupce „**Vybrané hodnoty**“ pomocí tlačítka „>“. Nyní můžeme daný výrok ze sloupce „Vybrané hodnoty“ opět myší zakliknout a vlevo jeho pravdivost oceníme vahou.
- **Numerický** (viz Obrázek 22) – na konkrétní dotaz odpovíme zadáním hodnoty z rozsahu z dolní části okna („Zadejte hodnotu v rozmezí“).



Dotaz - Nemoce.xml

Jaká je váha binárního atributu

pacient je bledy

Ano Váha

Zadejte váhu pro "Ano" v rozmezí -3 3

Proč

Průběžné výsledky

Uložit odpovědi

Jistě ano

Irrelevantní

Jistě ne

Neznámý

Odložit odpověď

Potvrdit odpověď

Obrázek 19: Dotaz na binární atribut.

Dotaz - uver.xml

Jaká je hodnota a váha jednoduchého atributu

příjem klienta

vysoký příjem Váha

Vyberte hodnotu a nastavte pro ni váhu v rozmezí 0 1

Proč

Průběžné výsledky

Uložit odpovědi

Jistě ano

Irrelevantní

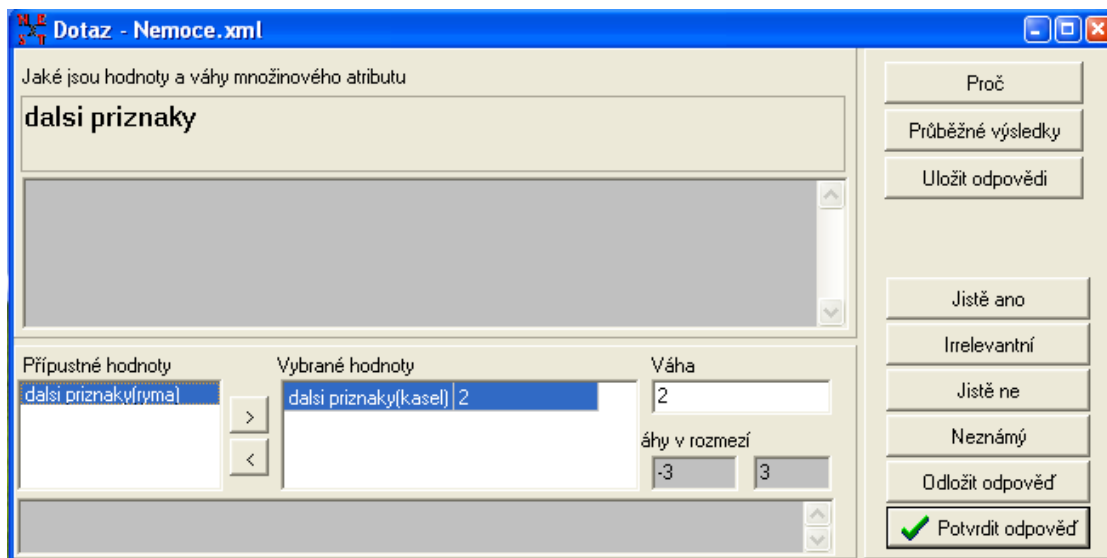
Jistě ne

Neznámý

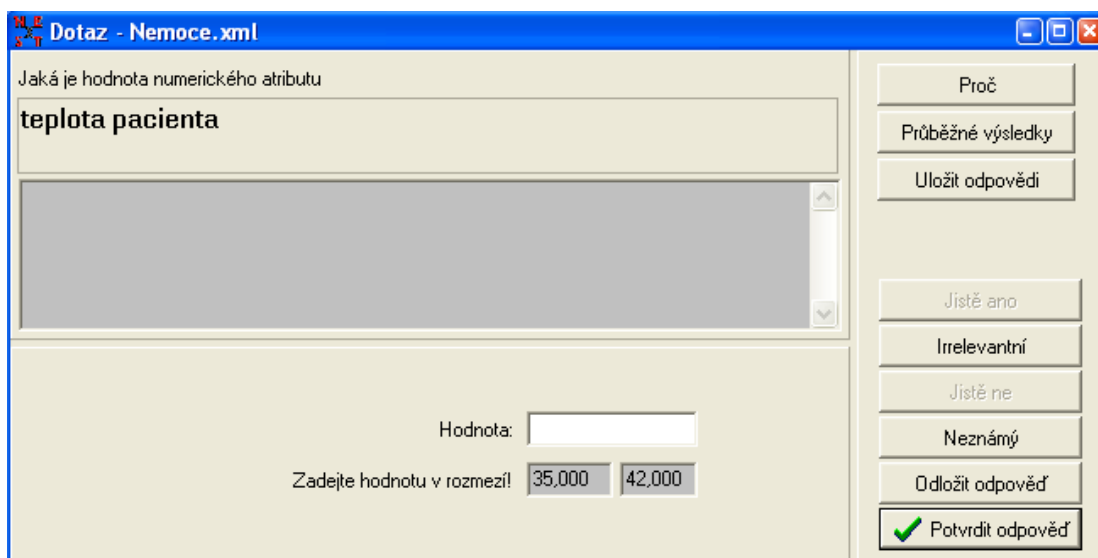
Odložit odpověď

Potvrdit odpověď

Obrázek 20: Dotaz na jednoduchý atribut.



Obrázek 21: Dotaz na množinový atribut.



Obrázek 22: Dotaz na numerický atribut.

### 2.2.5 UKLÁDÁNÍ ODPOVĚDÍ

Uložit odpovědi můžeme pomocí tlačítka „**Uložit odpovědi**“ v průběhu samotné konzultace (např. Obrázek 22) a nebo po jejím vyhodnocení v okně „**Výsledky pravidlového odvozování**“ (viz Obrázek 24).

Po stisknutí tlačítka „**Uložit odpovědi**“ systém otevře okno „**Uložení konzultace**“, ve kterém můžeme zadat popis konzultace (údaje zde vložené slouží uživateli pouze pro pozdější lepší orientaci - čeho se uložené odpovědi týkají). Po potvrzení tlačítkem „OK“ vybereme klasicky cílový adresář pro uložení, případně můžeme zvolit (vypsáním) příponu.

Soubor s odpověďmi má vnitřní strukturu „xml“. Uložení odpovědí se hodí pro přerušování konzultace bez ztráty již zadaných odpovědí. Nebo k testování a ladění báze, kdy si uložíme typické odpovědi, které by měly vést k jednomu z cílů.

**Upozornění pro studenty používající nižší verzi NESTu než 2.1.0.** V nižších verzích se stávalo, že soubory s výsledky i soubory editované v NEST editoru nebylo možno nahrát do programu NEST. Problémům se dalo vyhnout po zeditování souboru (např. v poznámkovém bloku) s výsledky či bázi znalostí, ve kterém bylo nutno odstranit, ne pouze vymazat obsah řádku, druhý řádek shora (viz Obrázek 23: jedná se o začerněný řádek s cestou k definičnímu souboru „base.dtd“).

```

Listner - [c:\Temp\zi\Nest\Nemoce.xml]
Soubor Editace Možnosti Nápověda 6 %
<?xml version="1.0" encoding="windows-1250"?>
<?DOCTYPE base SYSTEM "base.dtd">
<base>
  <global>
    <description>Nemoci1</description>
    <expert>PB</expert>
    <date>4.6.2003</date>
    <inference_mechanism>standard</inference_mechanism>
    <weight_range>3</weight_range>
    <default_weight>irrelevant</default_weight>
    <global_priority>first</global_priority>
    <context_global_threshold>0,000</context_global_threshold>
    <condition_global_threshold>0,000</condition_global_threshold>
  </global>
  <attributes>
    <attribute>
      <id>diagnoza</id>
      <type>multiple</type>
      <propositions>
        <proposition>
          <id>Nachlazení</id>

```

Obrázek 23: Úprava xml souboru.

## 2.2.6 VÝSLEDKY KONZULTACE

Po zodpovězení všech dotazů inferenční mechanismus vyhodnotí váhy a výsledky NEST zobrazí v okně „Výsledky pravidlového odvozování“ (viz Obrázek 24). Systémově je nastaveno zobrazení vyhodnocení cílových výroků, ale uživatel si může přepínači v levé části okna zvolit zobrazení vyhodnocení všech výroků v kombinaci s omezením jen kladné, jen záporné nebo všechny.

V tabulce, kde čteme výsledky konzultace vidíme sloupce: se jménem výroků, minimální a maximální hodnotou váhy, stavem a druhem výroku.

Stav výroku může nabývat hodnot:

- **Nedotčený** znamená, že výrok vůbec nebyl při odvozování použit.
- **Částečný** znamená, že systém při odvozování na tento výrok narazil, ale při odvozování výsledků nakonec nebyl zcela potřeba, a tak jeho váha nemusí být dopočítána.
- **Finální** znamená, že tento výrok byl pro odvozování potřeba a že jeho váha má finální podobu.

**Provizionální** znamená, že uživatel při některém dotazu odložil odpověď (viz složitější způsoby konzultace) a že po zodpovězení odložených odpovědí by se mohla váha tohoto výroku ještě změnit.

| Jméno                       | Min váha < | Max váha | Stav    | Druh      |
|-----------------------------|------------|----------|---------|-----------|
| HCD                         | 2,323      | 2,323    | finální | mezilehlý |
| plíce                       | 2,227      | 2,227    | finální | mezilehlý |
| pacient je bledý            | 2,000      | 2,000    | finální | dotaz     |
| dýchavivost                 | 2,000      | 2,000    | finální | dotaz     |
| dalsi príznaky(kasael)      | 2,000      | 2,000    | finální | dotaz     |
| diagnostika(TBC)            | 1,938      | 1,938    | finální | cíl       |
| diagnostika(Nachlazení)     | 1,286      | 1,286    | finální | cíl       |
| pacient ma normalni teplotu | 1,286      | 1,286    | finální | dotaz     |
| dalsi príznaky(tyma)        | 1,000      | 1,000    | finální | dotaz     |
| pacient ma zvýšenou teplotu | 0,000      | 0,000    | finální | dotaz     |
| diagnostika(Chripka)        | -0,857     | -0,857   | finální | cíl       |

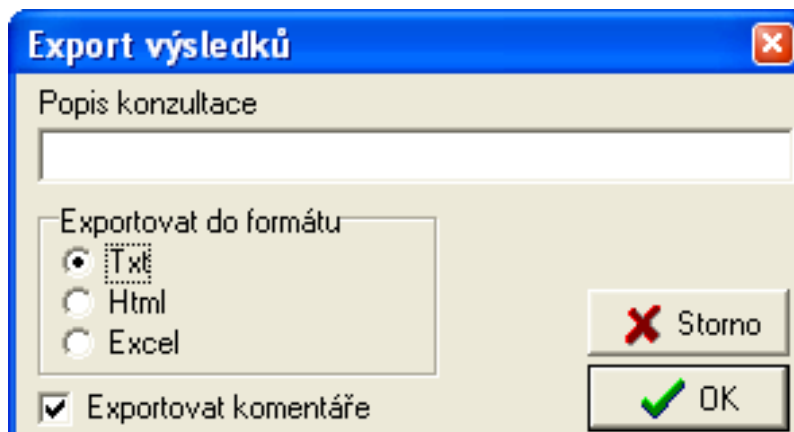
Obrázek 24: Výsledky pravidlového odvozování.

Druhy výroků jsou:

- **Cíl** – výroky, jejichž váhu se systém snaží určit.
- **Mezilehlý** – výrok, který je použit jak v předpokladu, tak v závěru nějakého odvozovacího pravidla.
- **Dotaz** – výrok, který se nevyskytuje v závěru žádného pravidla, takže pokud chce systém zjistit jeho váhu, musí se na něj zeptat uživatele.

Povšimněme si prosím, že zelenou barvou vidíme kladně vyhodnocené výroky. Výroky hodnocené nula (irelevantní výroky) jsou napsány černým fontem. Výroky hodnocené záporně pak vidíme vyobrazeny červeným fontem.

Výsledky pro pozdější použití - ať již k analýze nebo například pro dokumentaci - lze samozřejmě exportovat do souboru. Export provedeme pomocí tlačítka „**Export výsledků**“. Po jeho stlačení systém zobrazí okno téhož jména (viz Obrázek 25), v němž můžeme zadat další podrobnější popis výsledků do textového pole „**Popis konzultace**“, přepínači v oddílu „**Exportovat do formátu**“ určíme formát souboru s výsledky a zatrhávací volbou „**Exportovat komentáře**“ povolíme nebo zakážeme export komentářů.



Obrázek 25: Export výsledků.

### KONTROLNÍ OTÁZKA



1. S jakou formou znalostí pracuje systém NEST?
2. S jakými typy atributů pracuje systém NEST?
3. S jakými druhy výroků pracuje systém NEST z hlediska inference?

### SHRNUTÍ KAPITOLY



V této kapitole jsme se seznámili s prostředím NEST, ve kterém lze vytvářet a spouštět vlastní expertní systémy. Ukázali jsme si, jak prostředí nainstalovat a spustit, a také, jakým způsobem nahrát do NESTu bázi znalostí a provést nad takovou bází konzultaci. Taktéž jsme si ukázali, jak výsledky konzultace uložit či exportovat.



## ODPOVĚDI

1. S pravidly tvaru  $K: A \Rightarrow S(w)$  ( $K$  = kontext,  $A$  = antecedent,  $S$  = sukcedent,  $w$  = váha)
  2. Binární, jednoduchý, množinový a numerický.
  3. Cíl, mezilehlý a dotaz.
-

## **POZNÁMKY**

## 3 Struktura báze znalostí a inferenční mechanismus v NEST



### RYCHLÝ NÁHLED KAPITOLY

V této kapitole se čtenář seznámí s editorem prostředí NEST, který slouží pro vytváření báze znalostí, což je klíčová součást každého expertního systému.

---



### CÍLE KAPITOLY

Cílem kapitoly je seznámit čtenáře se všemi prvky editoru prostředí NEST, které budou potřeba pro vytvoření báze znalostí expertního systému. Zásadní částí této kapitoly je především seznámení se s atributy a jejich typy, výroky odvozené od těchto atributů a pravidly, které formálně vyjadřují znalosti experta v oboru zaznamenané v dané bázi.

---



### KLÍČOVÁ SLOVA KAPITOLY

NEST editor, atributy, výroky, pravidla, kontexty, integritní omezení, inferenční mechanismus

---

Než přejdeme k objasňování struktury báze znalostí, věnujme nejprve několik odstavců nástroji pro její tvorbu, tedy NEST editoru.

### 3.1 NEST editor

NEST editor je součástí instalačního balíčku NEST. Po spuštění programu nám naběhne obrazovka programu (viz Obrázek 13) a my můžeme založit novou bázi, nebo nahrát již existující, dokončenou či rozpracovanou, bázi za účelem její editace.

Základní okno editoru obsahuje následující menu:

- **Soubor** – základní nabídka s nástroji pro manipulaci s bázi na úrovni souboru.
  - **Nový** – založení nové báze klávesová zkratka CTRL+N.
  - Nový sklad případů NEST.
  - **Otevři** – otevření již existující báze znalostí CTRL+O.

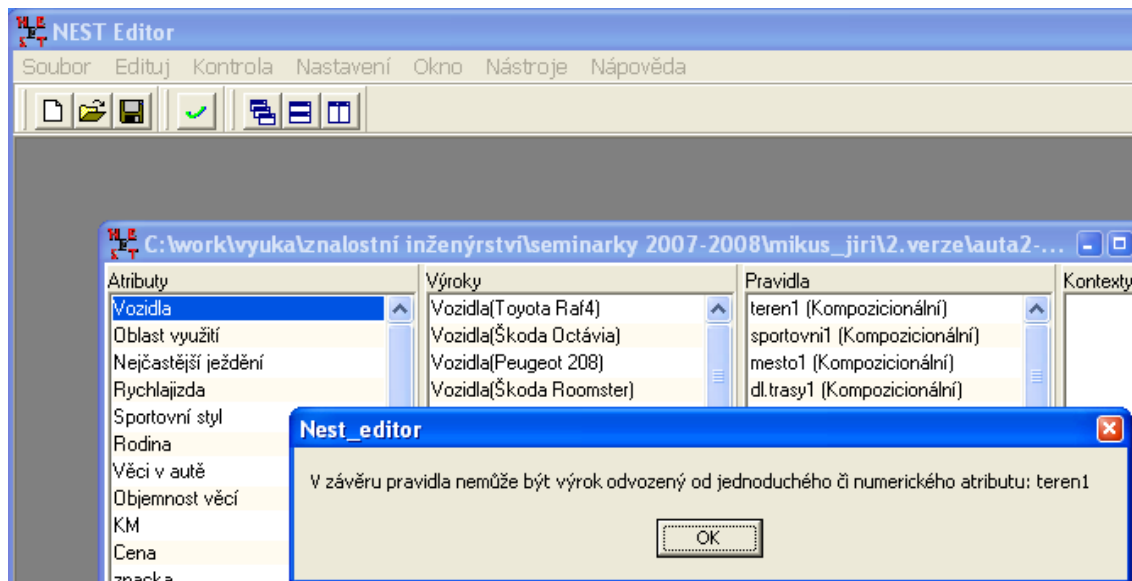


- **Zavři** – uzavření aktuálně otevřené báze.
- **Ulož** – uložení změn CTRL+S
- **Ulož jako ...** – uložení pod jiným jménem nebo volba formátu – výchozí formát ukládání je xml, v této nabídce nkb (NEST Knowledge Base).
- **Export báze** – export báze znalostí do formátů txt nebo htm. Tedy do formátů vhodných pro tvorbu dokumentace atp.
- **Konec** – ukončení programu.
- **Nastavení.**
  - **Nastavení ...** – stejné jako v NEST (viz kapitola 2.2.2).
- **Okno** – množina voleb ovlivňující rozložení a vizáž okna při práci s více bázemi najednou. Položky jsou taktéž dostupné pomocí ikon pod systémem menu.
  - Kaskáda.
  - Horizontální dlaždice.
  - Vertikální dlaždice.
  - Minimalizuj vše.
  - Uspořádej vše.
- **Nástroje.**
  - **Zobraz graf** – zobrazení grafu aktuální báze znalostí.
  - **Statistika** – výpis informací o aktuální bázi znalostí.
  - **NEST** – spuštění programu NEST – předpokládá správnou instalaci a nebo správné nastavení (viz 2.2.2).
  - **Base convertor** – spuštění konvertoru bází. Base convertor byl součástí instalačního balíku NEST (verze 1.4). Volba předpokládá opět správnou instalaci balíku nebo správné nastavení (viz 2.2.2).
- **Nápověda**
  - Nápověda
  - O aplikaci

Po založení nové báze či načtení již existující báze znalostí přibudou v menu následující položky:

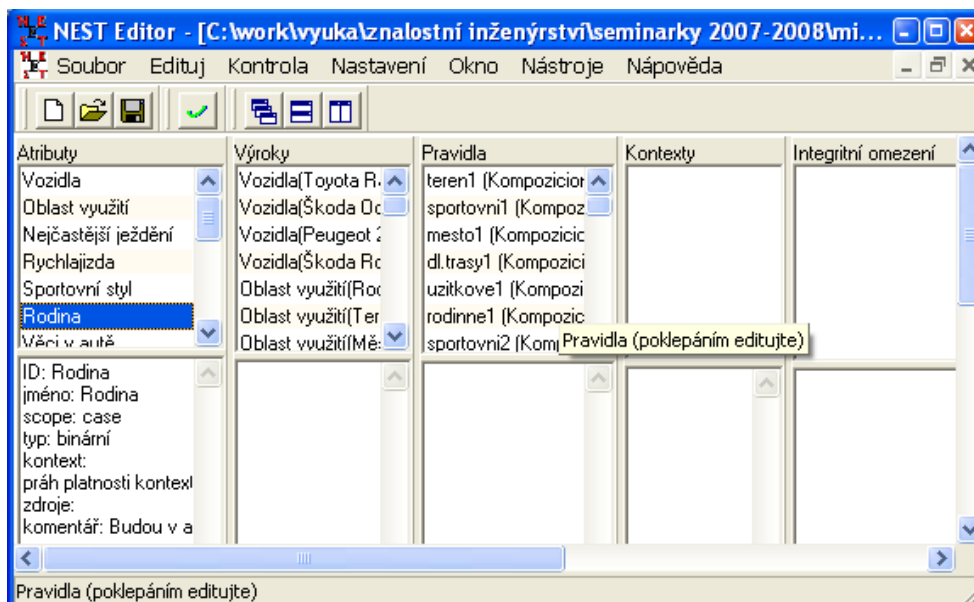
- **Edituj** – soubor nabídek pro editaci jednotlivých složek báze znalostí. K jednotlivým položkám, vzhledem k výkladu uváděnému v další části kapitoly, není potřeba uvádět podrobnosti.
  - Atributy a výroky
  - Pravidla
  - Kontexty
  - Integritní omezení
  - Globální vlastnosti
- **Kontrola**

**Kontrola** – volba je dosažitelná pomocí ikony v hlavním okně. Po zvolení systém provede kontrolu báze a případně ohlásí chyby (viz Obrázek 26).



Obrázek 26: Ukázka chybového hlášení po spuštění kontroly báze znalostí.

Po načtení existující báze znalostí či založení nové báze znalostí se v okně NEST editoru otevře okno aktuální báze, jež je rozděleno do sloupců, v nichž nalezneme postupně atributy, výroky, pravidla, kontexty a integritní omezení. Každý ze sloupců je rozdělen (defaultně) na poloviny, v horní části je vždy seznam např. atributů a v dolní části sloupce pak nalezneme, po zakliknutí konkrétní položky z horní části sloupce, další upřesňující parametry (viz Obrázek 27).



Obrázek 27: Okno NEST editoru po načtení existující báze znalostí.

### 3.1.1 ZADÁVÁNÍ NOVÝCH PRVKŮ BÁZE ZNALOSTÍ

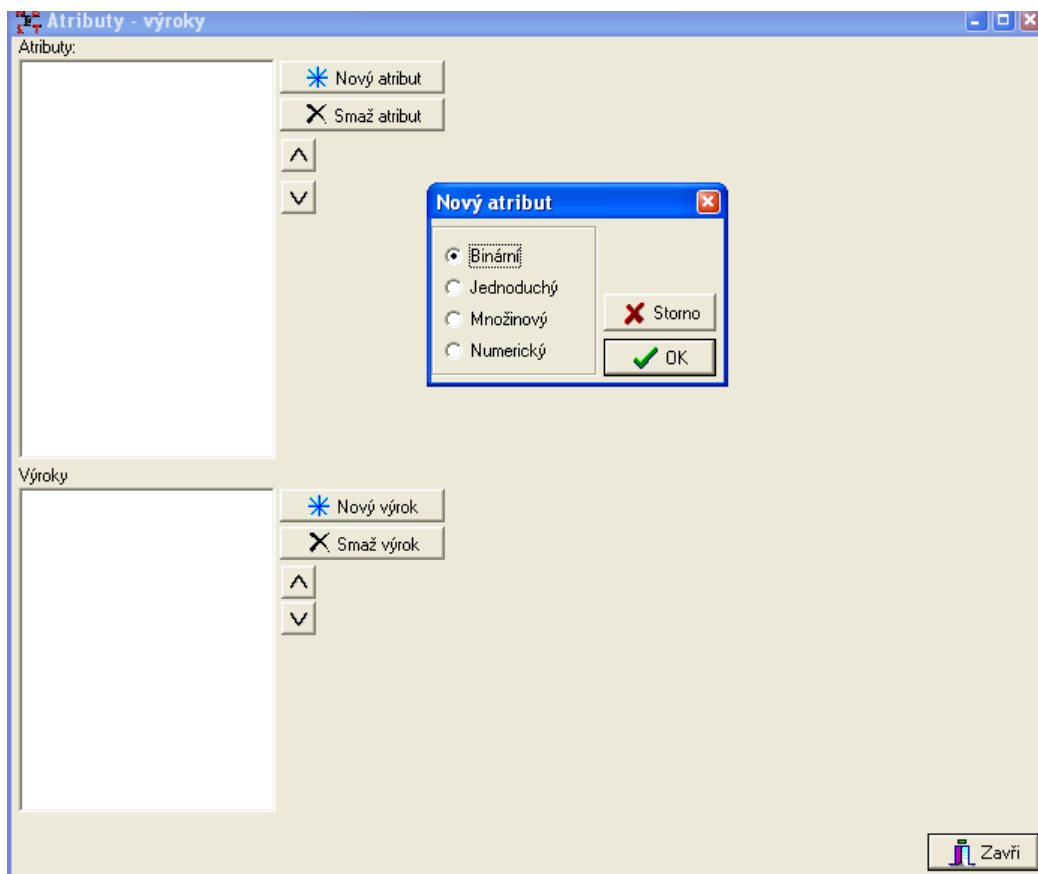
V následujícím textu ještě ukážeme, jakým způsobem se vytvoří nové prvky tzn. atributy, výroky a pravidla v NEST editoru. Jak již bylo zmíněno, po založení nové báze či načtení existující báze vybereme příslušnou nabídku z menu (např. Edituj/Atributy a výroky) nebo dvojklikem myši v horní polovině příslušného sloupce vyvoláme editační okno pro editaci či zadávání nového:

- **Atributu nebo výroku** (viz Obrázek 28) Nově vytvořené atributy nebo již existující respektive jejich názvy bychom viděli v levém horním poli a jejich výrokové stavy pak v poli pod atributy (viz např. Obrázek 31).
- **Pravidla** (viz Obrázek 29). Nově vytvořené pravidlo nebo seznam již existujících pravidel bychom viděli v levém sloupci okna Pravidla.

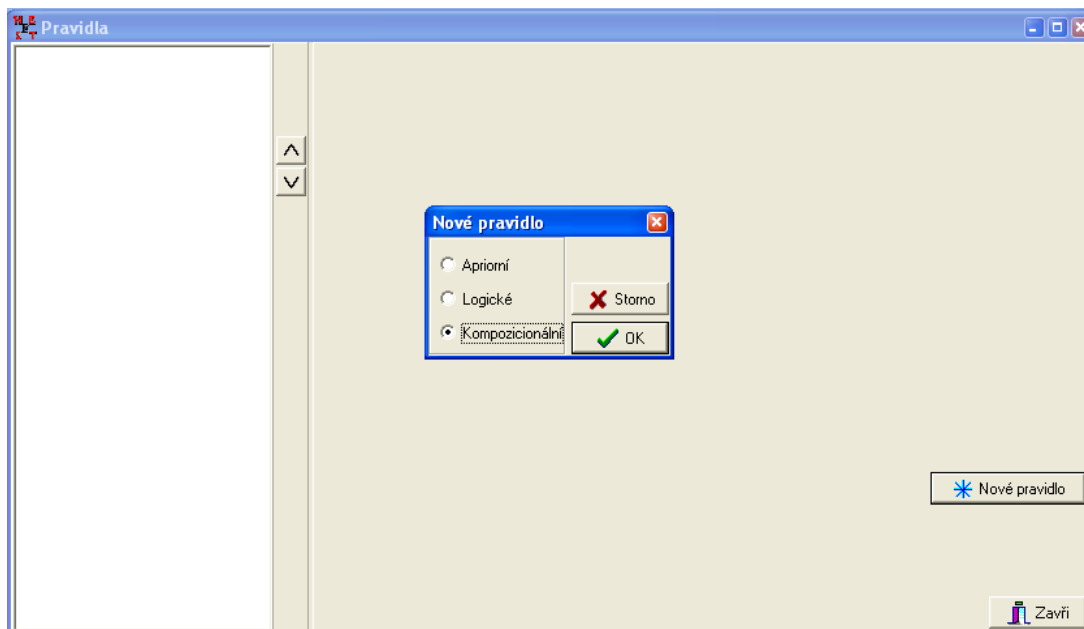
Je důležité si uvědomit (viz dále v rámci kapitoly) rozdíl mezi pojmy atribut a výrok. Každý z atributů může nabývat jednoho či více výrokových stavů. Pravdivost či nepravdivost těchto výroků hodnotí u dotazů osoba provádějící konzultaci zadáním subjektivně zvolené váhy v rozmezí, které systém dovoluje; u ostatních (mezilehlých či cílových) výroků určuje váhu inferenční mechanismus na základě odvozování pomocí báze znalostí.

Teprve po vytvoření atributů, alespoň dvou (jeden musí být binární nebo množinový, aby mohl být zakomponován do závěrové části pravidla), můžeme začít s tvorbou pravidel.

Pozn.: Tvůrce báze může (jak jste si jistě všimli) dále zadávat mimo atributů, výroků a pravidel také kontexty a integritní omezení. Kontexty a integritní omezení jsou však nástroje či prvky báze znalostí, které se s úspěchem dají využít při zpracovávání středně velkých či větších projektů, proto se zde jimi, respektive jejich tvorbou a editací dále nebudeme detailněji zabývat.



Obrázek 28: Editační okno Atributy – výroky pro vytváření nových atributů a jejich výroků s oknem Nový atribut v popředí (po stisknutí tlačítka Nový atribut).



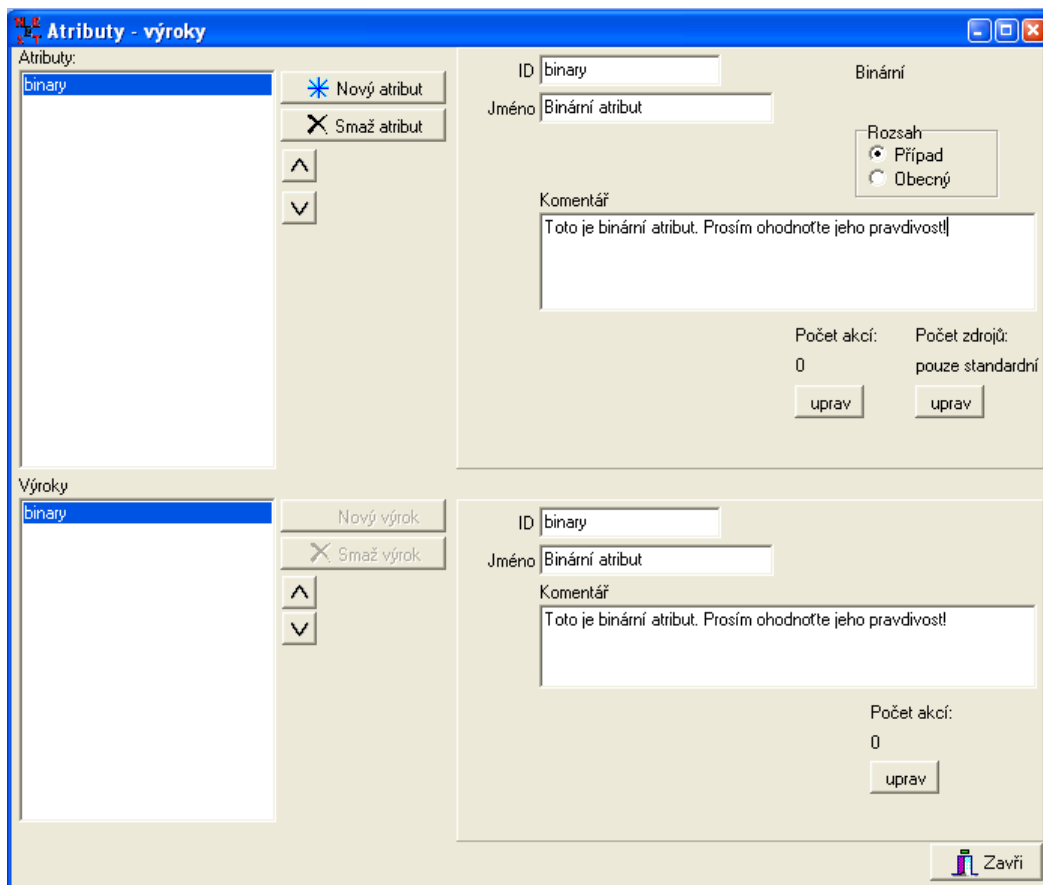
Obrázek 29: Editační okno Pravidla pro vytváření nových pravidel s oknem Nové pravidlo v popředí (po stisknutí tlačítka Nové pravidlo).

## 3.2 Reprezentace znalostí

### 3.2.1 ATRIBUTY A VÝROKY

Atributy slouží k popsání vlastností objektů reálného světa. Nabývají různých hodnot, podle kterých se pak určují váhy výroků, které jsou od atributů odvozeny. Existují čtyři typy atributů:

- binární;
- nominální jednoduchý;
- nominální množinový;
- numerický.

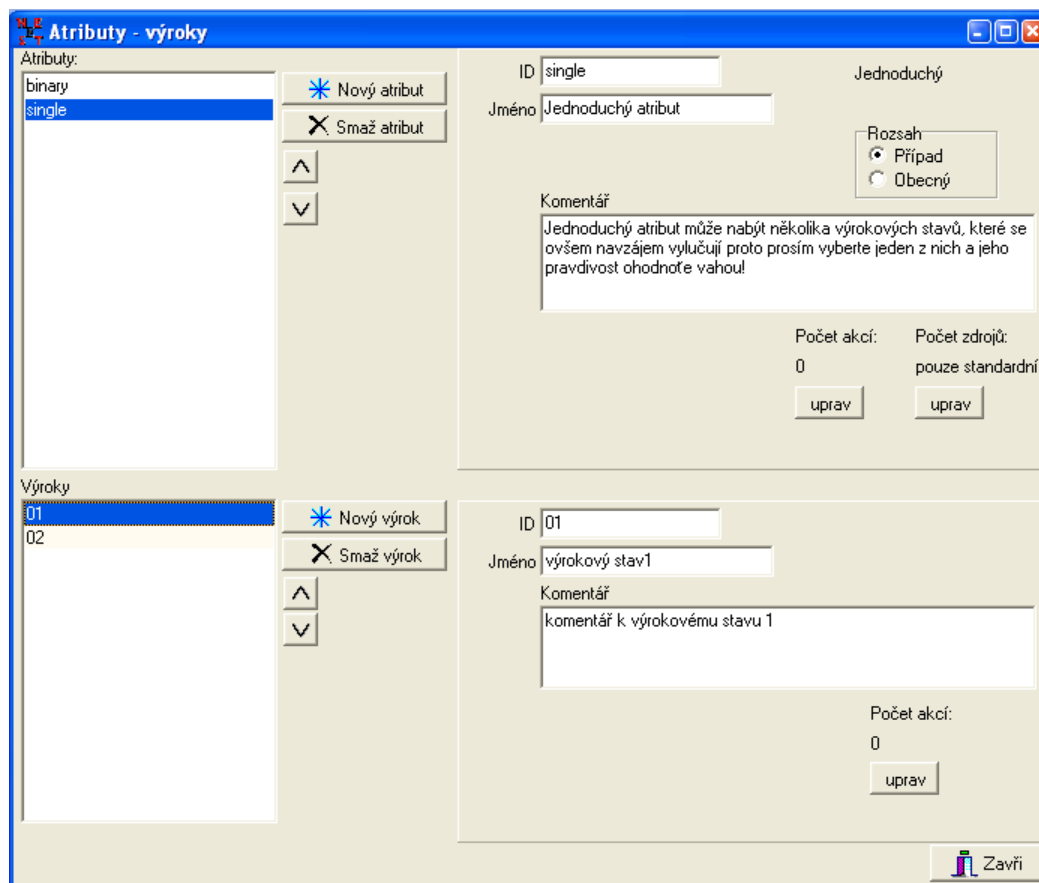


Obrázek 30: Zadávání nového binárního atributu.

**Binární atribut** je základním typem atributu. Je od něj odvozen **právě jeden výrok** se stejným jménem. Tento výrok, stejně jako atribut, od kterého je odvozen, pak nabývá určité váhy reprezentující jeho „pravdivost“. Váha je udávána pomocí intervalu, čímž je umožněno vyjadřovat i částečné znalosti o pravdivosti atributu, resp. výroku.

Například pracujeme-li na váhovém rozsahu  $[-1;1]$  (pokud nebude řečeno jinak, bude ve všech následujících příkladech používán tento váhový rozsah), následující váhy znamenají toto:

- $[-1; -1]$  – výrok jistě neplatí (FALSE);
- $[1; 1]$  – výrok jistě platí (TRUE);
- $[-1; 1]$  – o výroku nevíme nic (UNKNOWN);
- $[0; 0]$  – výrok je irelevantní (IRRELEVANT).

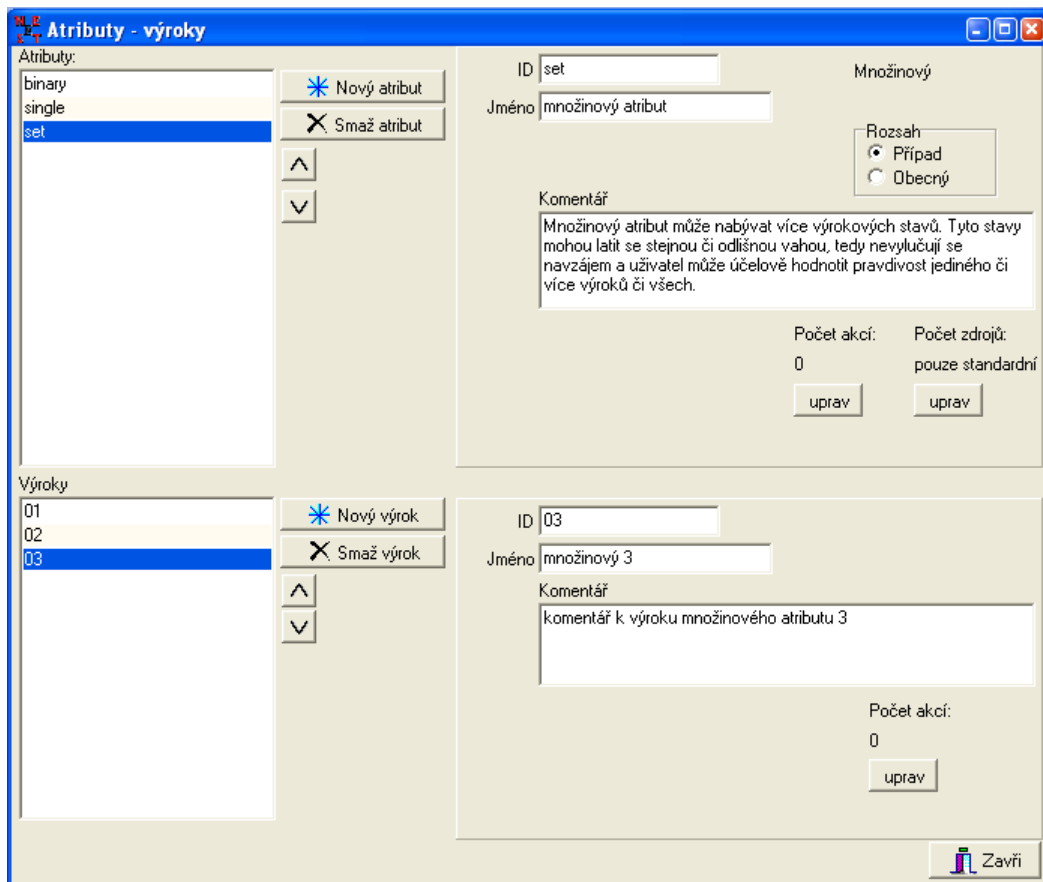


Obrázek 31: Zadávání jednoduchého atributu.

**Nominální jednoduchý atribut** (nebo jen Jednoduchý atribut) je atribut, který může nabývat právě jedné hodnoty z daného seznamu hodnot. Typickým příkladem je „typ účtu“ v bance. Od tohoto atributu je pak odvozeno právě tolik výroků, kolik je možných hodnot, kterých může atribut nabývat (výrok je již stejný, jako u binárního atributu). Pokud je atributu přiřazena nějaká hodnota (s vahou), tuto váhu dostane jen příslušný výrok a všem ostatním výroků je přiřazena váha „FALSE“.

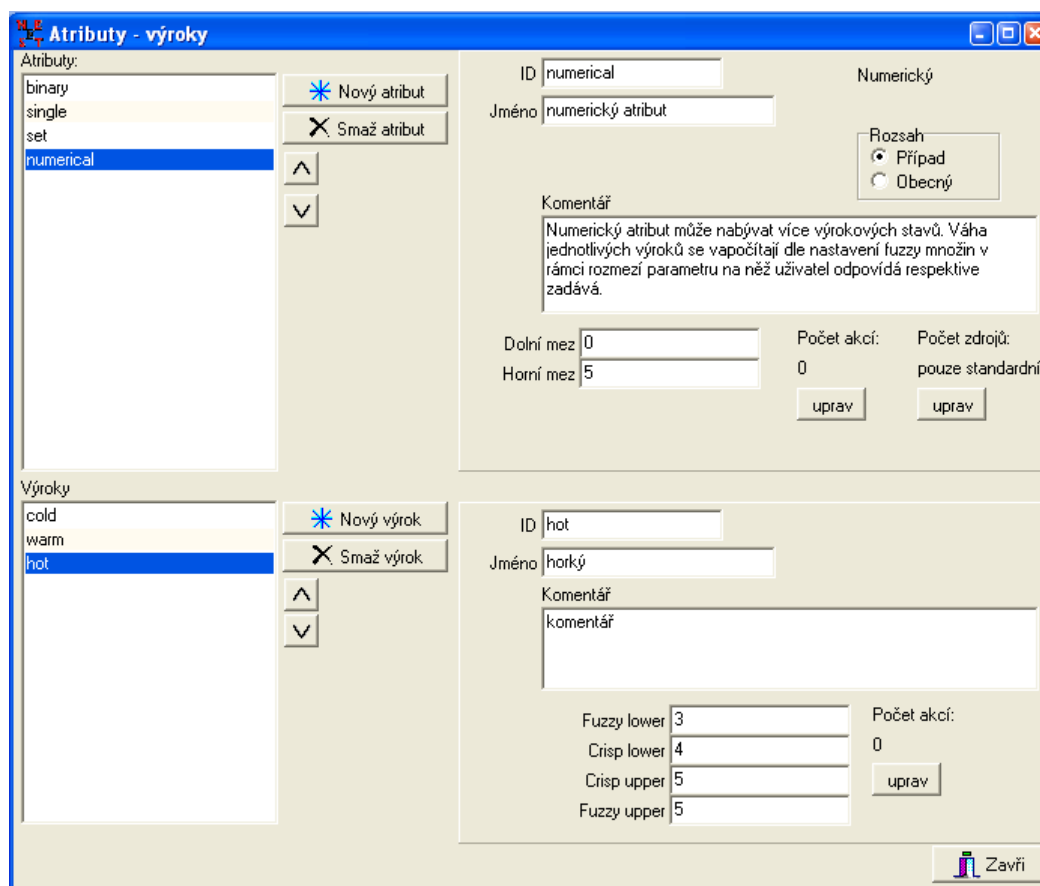
Je-li například zadáno, že uživatel si založil běžný účet s vahou  $[0,6;0,6]$ , pak všem ostatním typům účtu je přiřazena váha „FALSE“.

Všimněte si, že (na rozdíl od binárního atributu) atribut jednoduchý může nabývat jednoho z více výrokových stavů. Nové výroky zadáváme pomocí tlačítka „Nový výrok“.



Obrázek 32: Zadávání množinového atributu.

**Nominální množinový atribut** (nebo jen Množinový atribut) je podobný „Jednoduchému atributu“, jen může nabývat libovolného počtu hodnot z daného seznamu. Příkladem může být atribut „světové jazyky, kterými člověk hovoří“. Od tohoto atributu je opět odvozeno tolik výroků, kolik je možných hodnot, které může atribut nabývat. Při přiřazení určitých hodnot atributu se přenesou váha na příslušné výroky a hodnotám (resp. výroků), kterým nebyla přiřazena žádná váha, se pak přiřadí předem nastavená váha (UNKNOWN nebo IRRELEVANT).

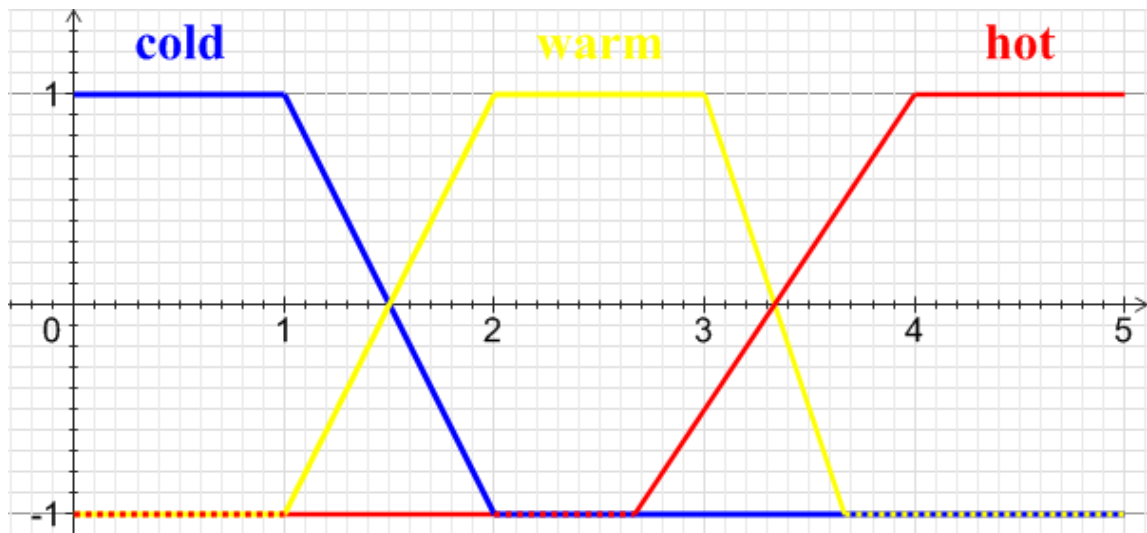


Obrázek 33: Zadávání numerického atributu.

**Numerický atribut** je atribut, který nabývá číselné hodnoty z předem zadaného rozmezí. Jako příklad lze uvést „velikost platu“. Od tohoto atributu může být odvozeno libovolné množství výroků, přičemž každý výrok má zadanou váhovou funkci, která převede číselnou hodnotu na váhu.

Váhová funkce je zadávána pomocí tzv. fuzzy intervalu, který má lichoběžníkový, případně trojúhelníkový tvar charakteristické funkce (viz Obrázek 34). Zadávání numerického atributu je dokumentováno pomocí obrázku Obrázek 33; zadání výroku „hot“ odpovídá obrázku Obrázek 34.





Obrázek 34: Fuzzy intervaly.

Podle postavení v bázi znalostí lze výroky dělit na **dotazy** (výrok není v závěru žádného pravidla), **mezilehlé výroky** (výrok je v závěru nějakého pravidla i v předpokladu nějakého pravidla) a **cíle** (výrok není v předpokladu žádného pravidla). Pokud není výrok použit ani v předpokladu, ani v závěru pravidla, jedná se o tzv. **osamocený výrok**, který NEST zcela ignoruje.

Atributy můžeme dle rozsahu dělit na „případové“ a „obecné“. Případové atributy popisují konkrétní konzultovaný případ, zatímco obecné atributy jsou stálé pro více konzultací (např. úroková míra).

**Hodnotu určitého atributu** může systém získat z různých zdrojů. Standardně se systém snaží získat hodnoty „**inferencí**“ (odvozováním) pomocí pravidel, jedná-li se o cíle či mezilehlé atributy, nebo dotazem „**od uživatele**“, jedná-li se o dotaz. Dalším zdrojem může být „**soubor**“, jehož první řádek se systém pokusí interpretovat jako příslušnou hodnotu. Čtvrtým zdrojem je „**externí funkce**“, která vytvoří soubor s příslušnou hodnotou a z tohoto souboru pak hodnotu načte. Poslední dva uvedené zdroje mohou být použity pouze pro binární a numerické atributy. Posledním zdrojem je „**výpočet**“ (lze použít jen pro numerické atributy), kdy hodnota atributu je spočítána z hodnot jiných numerických atributů a zadaných čísel s použitím matematických operací sčítání, odčítání, násobení a dělení. Zdroje lze nastavit tak, aby v případě neúspěchu jednoho byl použit jiný.

Každý atribut či výrok může mít přidělené určité akce. Akcí se rozumí externí program, který je spuštěn, pokud výrok nabude alespoň takové váhy, která je u akce zadána.

### 3.2.2 PRAVIDLA

Základní podoba pravidel v NESTu je následující:

kontext: IF předpoklad THEN závěr<sub>1</sub>[váha<sub>1</sub>], ..., závěr<sub>N</sub>[váha<sub>N</sub>]

**Kontext** rozhoduje o tom, zda se dané pravidlo bude aplikovat či nikoliv. Blíže je o kontextech pojednáno v následující sekci.

**Předpoklad** pravidla je tvořen disjunktivní normální formou, neboli disjunkcí libovolného množství „výrazů“, přičemž „výrazem“ může být buď výrok, nebo konjunkce výroků. Je-li předpoklad splněn, je závěru pravidla připočítán příslušný příspěvek tohoto pravidla. Aby byl předpoklad splněn, musí jeho celková váha (o jejím vypočítání viz sekce „práce s neurčitostí“) přesáhnout „práh platnosti předpokladu“, který je implicitně nastaven na nulu.

**Závěr** pravidla je tvořen výrokem a váhou, která slouží pro výpočet příspěvku pravidla k danému výroku a představuje neurčitost v bázi znalostí. Pokud má pravidlo více závěrů, výsledný efekt je stejný, jako by existovalo více pravidel se stejnými předpoklady a příslušnými závěry.

Na rozdíl od předpokladu **můžeme v závěru pravidla použít pouze výroky binárních a množinových atributů**. Výroky jednoduchých atributů nemůžeme použít, neboť jejich platnost se v reálu vylučuje (platí vždy pouze jeden výrok z množiny).

Každé pravidlo má zadanou svou prioritu (číslo z intervalu  $[0;1]$ ), podle které může být postupováno při výběru pravidel pro aplikaci při odvozování. (O odvozování blíže viz sekce „Inferenční mechanismus“.)

Pravidla se dělí do třech základních typů:

- apriorní pravidla;
- logická pravidla;
- kompozicionální pravidla.

**Apriorní pravidla** jsou taková pravidla, která platí vždy, tj. nemají předpoklad (ani kontext), resp. jako předpoklad je brána hodnota „TRUE“. Tato pravidla umožňují přiřadit výrokům implicitní váhu.

**Logická pravidla** jsou pravidla, jejichž závěr **nenabývá** hodnot z celého intervalu  $[-1;1]$  (resp. [-rozsah vah; rozsah vah]), ale nabývá jen „logických hodnot“ 1 (TRUE) nebo -1 (FALSE). Je-li předpoklad splněn (tj. pokud dolní hranice váhového intervalu přesáhne práh zadaný pro toto pravidlo), je váha závěru „TRUE“, resp. „FALSE“ (pokud je negace závěru), není-li předpoklad splněn, váha závěru je „IRRELEVANT“. Pouze tato pravidla mohou dát absolutní jistotu  $\pm 1$ .

**Kompozicionální pravidla** pracují na celém intervalu  $[-1;1]$  tak, jak je vysvětleno v sekci „práce s neurčitostí“.

Pravidlu může být přiřazena „akce“ stejně jako je tomu u výroku. Tato akce je spuštěna, jakmile je pravidlo „použito“, tj. předpoklad pravidla je splněn a jsou učiněny příslušné závěry.

### 3.2.3 KONTEXTY

Kontexty jsou, podobně jako předpoklady pravidel, disjunktivní normální formy 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.

### 3.2.4 INTEGRITNÍ OMEZENÍ

Integritní omezení mají stejnou podobu jako kompozicionální pravidla. Základní rozdíl je v tom, že integritní omezení se nepodílejí na odvozování závěrů, ale jsou vyhodnocovány samostatně po skončení odvozování. Slouží pro formulování dodatečných podmínek, které by měly platit po skončení konzultace mezi literály. Pokud je integritní omezení splněno, systém to nedá nijak najevo. Pokud však splněno není, je uživatel upozorněn, že byla porušena omezení (a samozřejmě která).

### 3.3 Inferenční mechanismus

Inference v systému NEST je kombinací zpětného a přímého řetězení. Na začátku konzultace systém vezme všechny cíle (cílem je buď výrok, který není použit v předpokladu žádného pravidla, nebo výrok, který uživatel označí jako cílový), postupně je označuje jako „aktuální“ a snaží se je vyhodnotit.

„Aktuální“ výrok je nejprve vyhodnocován metodou zpětného odvozování následujícím způsobem:

- Je zkontrolováno, není-li aktuální výrok již vyhodnocen (nachází se ve stavu „final“). Pokud ano, je vyhodnocování ukončeno.
- Následuje zjištění, z jakých zdrojů se má váha výroku zjišťovat. Pokud je zdroj jiný než „inference“, je vyhodnocování přerušeno a je spuštěn příslušný zdroj. Pokud zdroj není uveden, použije se „inference“.
- Jsou zjištěna všechna pravidla, ve kterých je výrok v závěru. Pokud neexistuje žádné takové pravidlo, je vyhodnocování přerušeno a je položen dotaz uživateli (efekt je stejný, jako by byl u výroku zadán zdroj „uživatel“). Jsou-li známy příspěvky všech pravidel potřebných pro odvození váhy výroku (všechna kompozicionální pravidla nebo část logických pravidel, u kterých se příspěvky skládají pomocí disjunkce), je vypočtena výsledná váha výroku (pomocí funkce GLOB – viz Sekce 4 „neurčitost“), stav výroku je označen jako finální a vyhodnocování je ukončeno (nikoliv přerušeno). Pokud nejsou známy příspěvky potřebných pravidel, pravidla jsou seřazena dle zadané priority (od začátku, od konce, minimální délka, maximální délka, dle zadání experta), systém vezme první nevyhodnocené pravidlo a pokusí se zjistit jeho příspěvek následovně:
  - Pokud není známa váha kontextu, tato váha se zjistí (probíhá stejně jako zjišťování váhy předpokladu). Je-li váha kontextu (její maximální hranice) menší nebo rovna prahu platnosti kontextu, je příspěvek pravidla „IRRELEVANT“ (tj. nulový).
  - Je zjištěna váha předpokladu tak, že nejprve je zjištěna váha všech výrazů (u výroků se vezme jejich váha, u konjunkcí je váha jednotlivých výroků vyhodnocena pomocí funkce CONJ) a jednotlivé výrazy jsou vyhodnoceny pomocí funkce DISJ. Pokud váha některého výroku není známa (výrok není ve stavu „final“), je tento výrok označen za „aktuální“ a je pro něj spuštěno vyhodnocování.

Pomocí funkce CTR je spojena váha předpokladu a váha kontextu (má-li pravidlo kontext) a výsledná váha je opět pomocí funkce CTR spojena s vahou jednotlivých závěrů. Tím je pravidlo vyhodnoceno a jeho stav je označen jako „final“ (zároveň jsou spuštěny jeho „akce“).

Je-li odvozování přerušeno, je zjištěn atribut, od kterého je odvozen výrok, při jehož vyhodnocování došlo k přerušení. Hodnota tohoto atributu je zjištěna z příslušného zdroje

(dotazem od uživatele, ze souboru či externí funkce nebo pomocí výpočtu). Poté, co je hodnota zjištěna, je spočítána příslušná váha všech výroků, které jsou od daného atributu odvozeny, a jejich stav je označen jako „final“. (Stejně jsou vzaty i výroky odvozené od numerických atributů, které mají jako zdroj daný atribut.) Pro tyto výroky systém postupně spouští přímé odvozování, které probíhá takto:

- Pro výrok jsou zjištěna všechna pravidla, ve kterých je použit v předpokladech.
- Pravidlo je vyhodnoceno stejným způsobem, jako tomu bylo při zpětném odvozování s tím rozdílem, že pokud při vyhodnocování nějakého výroku z předpokladu pravidla dojde k přerušení, toto přerušení je ignorováno a místo toho je ukončeno vyhodnocování tohoto pravidla (to znamená, že se neposílá dotaz uživateli ani není použit jiný zdroj).

Je-li pravidlo úspěšně vyhodnoceno, procházejí se postupně výroky ze závěrů pravidla. Tento výrok je nejprve vyhodnocen metodou zpětného odvozování s tím, že je opět ignorováno přerušení a místo něj je vyhodnocování výroku ukončeno. Pokud se povede výrok vyhodnotit, je pro něj spuštěna metoda přímého odvozování.

### KONTROLNÍ OTÁZKA



1. Jakým *druhem* atributu byste popsali v bázi znalostí zabývající se rozhodováním o výpočetní technice situaci, že počítač může být na základě jistých dříve zjištěných výroků desktop, notebook nebo PDA? Zdůvodněte.
2. Vysvětlete, jak rozumíte následujícímu pravidlu:

```
IF HCD AND teplota(zvysena) THEN diagnoza(Chripka) [3,000],
      diagnoza(Nachlazení) [-2,000]
```

### SHRNUTÍ KAPITOLY



V právě prostudované kapitole jsme se seznámili s programem NEST editor. Již víme, ze kterých komponent se skládá vlastní báze znalostí, a po prostudování přiložených obrázků a splnění úkolů na konci kapitoly umíme zadat či editovat jednotlivé atributy a jejich výroky. Nahlédli jsme také do principu práce inferenčního mechanismu NESTu.



## ODPOVĚDI

1. Jelikož je v zadání otázky napsáno, že platnost stavů je určována na základě dříve zjištěných výroků, předpokládáme, že se bude jednat o atribut, mezilehlý případně cílový, tzn. výroky tohoto atributu musí být obsaženy v závěrech nějakých pravidel, proto ačkoli v praxi počítač může být pouze jedním z vyjmenovaných druhů (tedy jen desktop, nebo jen notebook nebo jen PDA), použijeme atribut *množinový*. Nicméně, pokud by byl tento atribut atributem základním, použili bychom samozřejmě pro jeho konstrukci atributu jednoduchého.
  2. Jestliže platí výrok HCD a zároveň výrok *zvysena* atributu *teplota* pak platí: výrok *Chripka* atributu *diagnoza* s vahou 3 a výrok *Nachlazení* atributu *diagnoza* s vahou -2. Protože toto pravidlo je vybráno z báze znalostí *nemoce.xml*, můžeme volně formulovat, že daným pravidlem je reprezentována zhruba následující znalost: Pokud má pacient zánět horních cest dýchacích (HCD) **a zároveň** má zvýšenou teplotou pak **zcela jistě** trpí chřipkou a **téměř jistě** netrpí nachlazením.
-

## **POZNÁMKY**

## 4 Neurčitost v NEST



### RYCHLÝ NÁHLED KAPITOLY

V první části této kapitoly budou představeny funkce pro zpracování neurčitosti standardním přístupem systému NEST. Ve druhé části si pak na praktickém příkladu ukážeme konkrétní výpočet neurčitosti. A konečně si ukážeme, jak postupovat při vlastní konzultaci, chceme-li, aby NEST zpracovával neurčitost jiným než standardním způsobem.



### CÍLE KAPITOLY

Hlavním cílem této kapitoly je seznámit se s přístupem ke zpracování neurčitosti, který je zabudován v systému NEST. K tomuto účelu je potřeba se seznámit čtyřmi typy inferenčních mechanismů, které systém NEST nabízí. Čtenář nahlédne přímo dovnitř fungování inference v NESTu a naučí se, jakým způsobem je v NESTu inference prováděna.



### KLÍČOVÁ SLOVA KAPITOLY

inference, neurčitost, konjunkce, disjunkce, negace, implikace, modus ponens, fuzzy logika

## 4.1 Práce s neurčitostí

Systém NEST pracuje s váhami, které vyjadřují neurčitost v bázi znalostí i neurčité odpovědi během konzultace. Zatímco váhy pravidel jsou zadávány pouze **jednou hodnotou**, během konzultace se používá pro vyjádření váhy **interval**. Váhy pravidel  $w$  (a případně i jednohodnotové odpovědi uživatele či jiných zdrojů) jsou převedeny na jednobodový interval  $[w; w]$ , např. místo váhy 0,6 se v systému pracuje s intervalem  $[0,6; 0,6]$ . Pro jednoduchost se omezíme na práci s jednohodnotovými odpověďmi.

Způsob práce s neurčitostí vychází z algebraické teorie skládání vah, jak uvádí Hájek (1985) (lze též nalézt v (Berka a kol. (1998))). NEST používá 5 základních funkcí (NEG, CONJ, DISJ, CTR, GLOB) a 2 pomocné (NORM, IMP). Tyto funkce jsou definované na intervalu  $[-1;1]$  (funkce NORM je definovaná na množině reálných čísel a slouží pro normování intervalů na tento rozsah).



System nabízí **4 inferenční mechanismy** (tj. 4 podoby těchto funkcí) – **standardní, logický, neuronový a hybridní**. Standardní inferenční mechanismus vychází ze způsobu práce s neurčitostí v systémech MYCIN a PROSPECTOR, logický inferenční mechanismus je založen na vícestupňové logice – funkce CTR je fuzzy modus ponens a funkce GLOB je fuzzy disjunkce v Łukasiewiczově<sup>2</sup> logice, neuronový inferenční mechanismus je inspirován modelem neuronu a hybridní inferenční mechanismus je kombinací standardního a logického. Pět funkcí (NEG, CONJ, DISJ, NORM, IMP) je definováno pro všechny typy inferenčních mechanismů stejně, ve zbylých dvou funkcích (CTR a GLOB) jsou odlišnosti dané zvoleným inferenčním mechanismem.

## 4.2 Funkce pro práci s neurčitostí

V této kapitole si popíšeme přesný tvar funkcí, které umožňují s neurčitostí pracovat. Funkce si ilustrováme na příkladu z bankovního prostředí. Např., pokud má potenciální bankovní klient zájem o půjčku, banka se vždy zajímá, jaké je riziko, že klient půjčku nebude schopný splatit. Toto riziko lze částečně určit z měsíčního příjmu klienta. Banka se tedy klienta může zeptat, zda má klient vysoký příjem (předpokládejme, že klient přesnou výšku svého měsíčního příjmu chce utajit). Banku by tedy zajímala platnost výroku

klient má vysoký příjem

Pokud klient vydělává statisíce korun měsíčně, zřejmě odpoví jasně – ano, mám vysoký příjem. V tom případě pracovník pobočky banky ohodnotí daný výrok jako platný. V případě, že je klient nezaměstnaný a tudíž žádný měsíční příjem nemá, zřejmě odpoví také jasně – ne, nemám vysoký příjem. V tomto případě pracovník pobočky banky ohodnotí daný výrok jako neplatný. Co se ale stane v případě, kdy klient banky vydělává např. 120% průměrné měsíční mzdy v ČR? V takovém případě určení platnosti tohoto výroku již tak zřejmá není, jelikož není možné jednoznačně říct ani že klient má vysoký příjem, ani že nemá vysoký příjem. Platnost by navíc mohla záviset např. na regionu, kde klient pracuje (30 000,- měsíčně v Praze má zcela jinou reálnou hodnotu než 30 000,- měsíčně např. v Karviné). Klient by tedy mohl odpovědět něco ve smyslu „spíše ne“ (pro Prahu) či „spíše ano“ (pro Karvinou). Pracovník pobočky banky by pak mohl daný výrok analogicky ohodnotit jako např. **spíše neplatný** (pro Prahu) či **spíše platný** (pro Karvinou).

Jednou z možností, jak umožnit v expertních systémech práci s takto neurčitými, avšak velmi běžnými, odpověďmi (spíše ano, celkem ano, téměř jistě ano, apod.), je povolit platnost výroků (odpovědí) i mimo klasické hodnoty pravda (platný) a nepravda (neplatný) a rozšířit hodnoty platnosti výroku na celou množinu hodnot, tak jak je např. běžné v tzv. fuzzy logice. Zde může platnost daného výroku nabývat libovolné hodnoty z intervalu [-1;

<sup>2</sup> Jan Łukasiewicz - (21.12.1878 – 13.2.1956) polský matematik a filozof; tvůrce více hodnotové logiky někdy také nazývané jako Łukasiewiczova logika.

1]<sup>3</sup>, kdy **minimum** (hodnota -1) pak **odpovídá platnosti neplatného výroku** (nepravda) a **maximum** (hodnota 1) **odpovídá platnosti platného výroku** (pravda). Hodnoty mezi těmito extrémami pak můžeme použít právě k interpretaci výroků s neurčitou platností, jako např. výše zmíněné spíše ano, celkem ano, téměř jistě ano, apod.

Takto vstupuje do hry **neurčitost v odpovědích (v datech)** i v systému NEST. Aby bylo možno s takovými neurčitými odpověďmi logicky pracovat, tzn. spočítat např. platnost **negace, konjunkce, disjunkce** či **implikace** daného výroku (daných výroků), využívá NEST následující funkce. V následujícím textu budeme předpokládat **rozsah platnosti výroků interval [-1; 1]**. **Platnosti výroku** budeme také říkat **váha výroku**.

### 4.2.1 FUNKCE NEG

Funkce NEG slouží pro výpočet váhy negace výroku. Je definována následovně:

$$\text{NEG}(w) = -w$$

Předpokládejme nezaměstnaného klienta žádajícího o půjčku. Tzn., že výrok `klient má vysoký příjem` by byl ohodnocen vahou -1. Do systému tedy vstupuje tento výrok

`klient má vysoký příjem (-1)`

Jak by se dala vyjádřit váha negace původního výroku, kterou lze formulovat např. jako `klient nemá vysoký příjem`? Váha původního výroku je -1, tedy  $w = -1$ . Váha negace výroku se spočte pomocí funkce NEG jako

$$\text{NEG}(w) = \text{NEG}(-1) = -(-1) = 1$$

Vypočtená váha výroku `klient nemá vysoký příjem` je tedy 1, což dobře koresponduje se skutečností, že se jedná o nezaměstnaného klienta.

---

<sup>3</sup> velmi často se také používá interval [0; 1]

## 4.2.2 FUNKCE CONJ

Funkce CONJ (a zároveň, z angl. conjunction) slouží pro výpočet váhy **konjunkce** (a zároveň) dvou výroků. Je definována následovně:

$$\text{CONJ}(v, w) = \min(v, w)$$

Předpokládejme klienta, pro kterého platí dva následující výroky s danými vahami

- 1) klient má vysoký příjem (-1)
- 2) klient má vysoké konto (1)

Dále předpokládejme **velmi přísnou** banku, která klientovi půjčí jedině v tom případě, že klient má **jak** vysoký příjem, **tak** vysoké konto. Tento požadavek lze přeformulovat tak, že banka je ochotná půjčit pouze klientovi, který má vysoký příjem **a zároveň** má vysoké konto (má hodně peněz na účtu). Jak bychom tedy vypočetli váhu následujícího výroku?

klient má vysoký příjem **a zároveň** klient má vysoké konto

Stačí dosadit do funkce CONJ našeho klienta, tedy  $v = -1$  a  $w = 1$ , tzn.

$$\text{CONJ}(-1; 1) = \min(-1; 1) = -1$$

Váha konjunkce dvou výše zmíněných výroků je tedy -1, tzn. do systému vstupuje (složený) výrok

klient má vysoký příjem **a zároveň**  
klient má vysoké konto (-1)

Předpokládejme dalšího klienta s

- 1) klient má vysoký příjem (0,4)
- 2) klient má vysoké konto (0,3)

Vypočtená váha stejného výroku složeného pomocí **a zároveň** je 0.3. Takovýto klient by měl určitě vyšší šanci na získání půjčky než první klient, přestože první klient má vyšší konto než tento.

Je důležité si uvědomit, že právě funkce  $\min$  není použita k výpočtu váhy konjunkce dvou výroků náhodou. Vysoká váha konjunkce dvou výroků náhodou je možná jen v případě, kdy každý z těchto dvou výroků má vysokou váhu. Pokud je jedna z vah nízká, váha konjunkce je také nízká. Toto dobře koresponduje se zažitou představou, že dva výroky platí dohromady pouze v případě, kdy platí jak jeden, tak druhý výrok.

### 4.2.3 FUNKCE DISJ

Funkce DISJ slouží pro výpočet váhy **disjunkce** (logická spojka nebo) dvou výroků. Je definována následovně:

$$\text{DISJ}(v, w) = \max(v, w)$$

Předpokládejme klienta, pro kterého platí dva následující výroky s danými vahami

- 1) klient má vysoký příjem (-1)
- 2) klient má vysoké konto (1)

Dále předpokládejme **méně přísnou** banku, které pro smlouvení půjčky stačí, aby klient měl **alespoň** vysoký příjem bez nutnosti mít vysoké konto nebo **alespoň** vysoké konto bez nutnosti mít vysoký příjem (pokud by měl jak vysoký příjem tak vysoké konto, tak to samozřejmě bude stačit také). Tento požadavek lze přeformulovat tak, že banka je ochotná půjčit klientovi, který má **bud'** vysoký příjem, **nebo** má vysoké konto (nebo samozřejmě obojí). Jak bychom tedy vypočetli váhu následujícího výroku?

klient má vysoký příjem **nebo** klient má vysoké konto

Stačí dosadit do funkce DISJ našeho klienta, tedy  $v = -1$  a  $w = 1$ , tzn.

$$\text{DISJ}(-1; 1) = \max(-1; 1) = 1$$

Váha konjunkce výše zmíněných dvou výroků je tedy 1, tzn. do systému vstupuje (složený) výrok

klient má vysoký příjem **nebo** klient má vysoké konto (1)

Předpokládejme dalšího klienta s

- 1) klient má vysoký příjem (0,4)
- 2) klient má vysoké konto (0,3)

Vypočtená váha výroku složeného pomocí **nebo** je 0.4. Takovýto klient by měl určitě nižší šanci na získání půjčky než první klient, přestože první klient má nižší příjem než tento.

Je důležité si uvědomit, že právě funkce  $\max$  není použita k výpočtu váhy disjunkce dvou výroků náhodou. Nízká váha disjunkce dvou výroků náhodou je možná jen v případě,

kdy každý z těchto dvou výroků má nízkou váhu. Pokud je jedna z vah vysoká, váha disjunkce je také vysoká. To znamená, že disjunkce dvou výroků platí v případech, kdy alespoň jeden z těchto výroků je platný.

Tento a předchozí příklad ilustrují, jak lze pomocí logických spojek (a zároveň, nebo) modelovat např. „přísnost“ banky a jak odlišná volba logické spojky výrazně ovlivňuje výsledek výpočtu, popř. celý rozhodovací proces o smlouvení půjčky - dvakrát dva stejní zákazníci, ale vždy zcela opačný výsledek.

#### 4.2.4 FUNKCE CRT

Funkce CRT (z angl.. contribution) slouží pro výpočet příspěvku pravidla, resp. pro složení vah kontextu a předpokladu pravidla. Je definována následovně pro:

- standardní, neuronový a hybridní inferenční mechanismus:

$$\begin{aligned} \text{CTR}(a, w) &= 0 && ; a \leq 0 \\ \text{CTR}(a, w) &= a * w && ; a > 0 \end{aligned}$$

- logický inferenční mechanismus:

$$\begin{aligned} \text{CTR}(a, w) &= 0 && ; a < 0 \\ \text{CTR}(a, w) &= \text{sign}(w) * \max(0, a + |w| - 1) && ; a \geq 0 \end{aligned}$$

Předpokládejme znovu nezaměstnaného klienta žádajícího o půjčku. Tzn., že výrok klient má vysoký příjem by byl ohodnocen vahou -1. Do systému tedy vstupuje tento výrok

klient **má** vysoký příjem (-1)

Předpokládejme velmi jednoduchou bázi znalostí, která obsahuje pouze dvě pravidla

- 1) klient **má** vysoký příjem => půjčíme (1)
- 2) klient **nemá** vysoký příjem => půjčíme (-1)

kde výrok půjčíme odpovídá situaci, kdy banka klientovi půjčku poskytne. Váhu tohoto výroku chceme odvodit z odpovědí pomocí pravidel. Zatím však o výroku půjčíme **nevíme nic**. Systém NEST v takovém případě přiřadí výroku **neutrální váhu 0**, tzn. není ani (trochu) platný ani (trochu) neplatný.

Jak upravíme váhu výroku půjčíme pro našeho klienta na základě jeho odpovědí a naší bázi znalostí? Nejdříve berme v úvahu pravidlo 1). Výrok klient **má** vysoký příjem má pro našeho klienta váhu -1, tzn. do funkce CRT dosadíme  $a = -1$ . Váha pravidla 1) je 1, tzn.  $w = 1$ . Tudíž

$$\text{CRT}(a, w) = \text{CRT}(-1, 1) = 0,$$

jelikož  $a < 0$ . Příspěvek pravidla 1) k neznámé váze výroku půjčíme je 0, tzn. můžeme ho ignorovat (zatím se spokojíme s tímto faktem bez dalšího vysvětlování a vše si vyjasníme u funkce GLOB), tudíž váha výroku půjčíme je stále 0. Na základě této váhy se nemůžeme ani přiklonit ani odklonit od smlouvy půjčky.

Nyní vezměme v úvahu pravidlo 2). Výrok klient **nemá** vysoký příjem má pro našeho klienta váhu 1, což jsme vypočetli pomocí funkce NEG (viz výše), tzn. do funkce CRT dosadíme  $a = 1$ . Váha pravidla 2) je -1, tzn.  $w = -1$ . Tudíž

$$\text{CRT}(a, w) = \text{CRT}(1, -1) = 1,$$

jelikož  $a > 0$ . Příspěvek pravidla 2) k váze výroku půjčíme je 1, tudíž novou váhu výroku půjčíme stanovíme na 1. To můžeme provést díky tomu, že váha půjčíme byla doposud rovná 0 (pokud by nebyla rovná 0, použili bychom funkci GLOB, viz níže). Na základě této nové váhy se již můžeme silně odklonit od smlouvy půjčky, což reálně odpovídá situaci, kdy nezaměstnaný člověk má je velmi malou šanci na získání půjčky.

#### 4.2.5 FUNKCE GLOB

Funkce GLOB slouží pro skládání příspěvků více pravidel. Je definována následovně:

- standardní inferenční mechanismus

$$\begin{aligned} \text{GLOB}(v, w) &= 0; \quad (v = 1, w = -1) \text{ nebo } (v = -1, w = 1) \\ \text{GLOB}(v, w) &= (v + w) / (1 + v * w) \end{aligned}$$

- logický a hybridní inferenční mechanismus

$$\text{GLOB}(w_1, \dots, w_n) = \min(\sum_{w>0} w, 1) + \max(\sum_{w<0} w, -1)$$

- neuronový typ neurčitosti

$$\text{GLOB}(w_1, \dots, w_n) = \min(\max(s * w, -1), 1),$$

kde  $s$  je vektor vah v neuronu (neplést s vahami  $w_1, \dots, w_n$ ),  $|s| = 1$ , tzn. Eukleidovská norma vektoru  $s$  je rovna 1,  $w = (w_1, \dots, w_n)$  a  $*$  značí skalární součin dvou vektorů.

Opět předpokládejme jednoduchou bázi znalostí, která obsahuje čtyři pravidla

- 1) klient **má** vysoký příjem => půjčíme (0,9)
- 2) klient **nemá** vysoký příjem => půjčíme (-0,9)
- 3) klient **má** vysoké konto => půjčíme (0,5)
- 4) klient **nemá** vysoké konto => půjčíme (-0,3)

Všimněme si dvou pravidel 3) a 4). Váhy již nejsou nastaveny taky (absolutně) vysoce jako u pravidel 1) a 2). Takto lze modelovat situaci, kdy na **některých výrocích záleží méně a na některých více**. V tomto případě banka klade hlavní důraz na výšku příjmu klienta, zatímco výška konta klienta hraje při rozhodování o půjčce méně výraznou roli. Ilustrujme si to na následujících dvou klientech.

#### První klient:

- 1) klient má vysoký příjem (1)
- 2) klient má vysoké konto (-1)

Nenulové příspěvky, jelikož  $a > 0$  (viz funkce CRT), k váze výroku půjčíme plynou z pravidel 1) CRT(1; 0,9) = 0,9 a 4) CRT(NEG(-1), -0,3) = -0,3. Kolik je tedy váha výroku půjčíme? Tento výpočet lze provést pomocí funkce GLOB. Za předpokladu standardního inferenčního mechanismu by výpočet byl proveden následovně. Z pravidla 1) je  $v = 0,9$  a z pravidla 2) je  $w = -0,3$ , tedy

$$\text{GLOB}(v; w) = \text{GLOB}(0,9; -0,3) = (0,9 + (-0,3)) / (1 + 0,9 * (-0,3)) = 0,82$$

#### Druhý klient:

- 1) klient má vysoký příjem (-1)
- 2) klient má vysoké konto (1)

Nenulové příspěvky k váze výroku půjčíme plynou z pravidel 2) CRT(NEG(-1); -0,9) = -0,9 a 3) CRT(1, 0,5) = 0,5. Kolik je tedy váha výroku půjčíme? Tento výpočet lze provést opět pomocí funkce GLOB. Za předpokladu standardního inferenčního mechanismu by výpočet byl proveden následovně. Z pravidla 2) je  $v = -0,9$  a z pravidla 3) je  $w = 0,5$ , tedy

$$\text{GLOB}(v; w) = \text{GLOB}(-0,9; 0,5) = (-0,9 + 0,5) / (1 + (-0,9) * 0,5) = -0,73$$

U prvního klienta bychom se (jako banka) výrazně přiklonili k půjčce (s vahou 0,82), kdežto u druhého klienta bychom se výrazně (s vahou -0,72) od půjčky odklonili. Na

tomto příkladu lze dobře vidět, jak lze modelovat odlišnou důležitost jednotlivých vlastností klienta při rozhodování. Zatímco váhy výroků našich dvou klientů jsou pouze navzájem „prohozeny“, 1 za -1 a naopak, rozhodnutí o půjčce, pokud by záviselo pouze na váze výroku půjčíme, se zcela zásadně liší.

Všimněme si, že funkce GLOB je vždy symetrická v argumentech, tzn. je jedno, v jakém pořadí do funkce vstupují příspěvky jednotlivých pravidel. Dále si všimněme, že je-li jeden z argumentů funkce GLOB roven 0 a ten druhý např. roven  $v$  z intervalu  $[-1; 1]$ , pak

$$\text{GLOB}(v; 0) = \text{GLOB}(0; v) = v$$

pro všechny inferenční mechanismy. To vysvětluje jev, na který jsme narazili u funkce CRT. Je-li platnost nějakého výroku neznáma ( $= 0$ ) a pro tento výrok jsme vypočetli pomocí funkce CRT pouze jeden nenulový příspěvek  $v$  k tomuto výroku, můžeme novou váhu výroku stanovit rovnou na  $v$ .

#### 4.2.6 FUNKCE NORM

Jediná funkce, která je definována i mimo interval  $[-1; 1]$ , slouží pro ořezávání vah právě na tento interval, pokud je někde přesažen. Funkce je používána po výpočtu příspěvku kompozicionálního pravidla pomocí funkce CTR a po složení všech příspěvků pravidel u výpočtu váhy výroku pomocí funkce GLOB. Funkce je definována následovně:

$$\text{NORM}(w) := \min(\max(w, \text{rozsah}), \text{rozsah})$$

kde  $\text{rozsah} = 0,999$  (resp. počet devítek za desetinnou čárkou odpovídá počtu desetinných míst, se kterými právě systém pracuje).

#### 4.2.7 FUNKCE IMP

Funkce IMP je používána pro výpočet výsledné váhy integritního omezení. Rozhoduje tedy o tom, zda integritní omezení bylo porušeno či nikoliv a do jaké míry. Je definována následovně:

$$\begin{aligned} \text{IMP}(w, v) &= 0; w \leq 0 \\ \text{IMP}(w, v) &= \max(0, \min(w - v)) ; w > 0 \end{aligned}$$

kde  $w$  je váha předpokladu integritního omezení (případně upravená pomocí kontextu funkcí CTR) a  $v$  je váha jeho závěru.



### 4.3 Příklad výpočtu

Nyní se zaměříme na praktické procvičení výpočtu a případné možnosti srovnání přístupů k neurčitosti na základě konzultací v NEST.

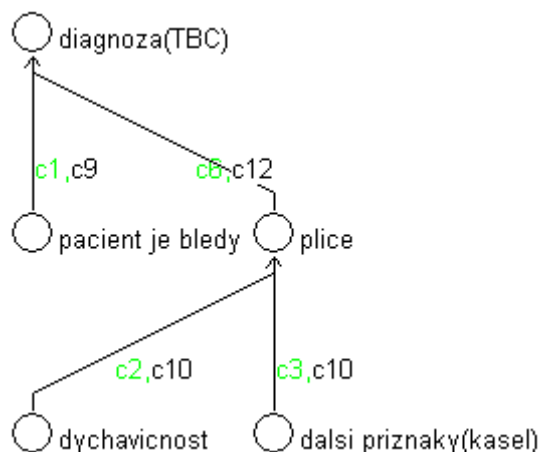
Spustíme konzultaci nemoce.xml se standardním inferenčním mechanismem a v rámci konzultace zadáme odpovědi viz Tabulka 2. Necháme vyhodnotit konzultaci a necháme si vysvětlit, jak systém postupoval v případě určení vah pro cílový výrok TBC. Použijeme informace z již zmiňované tabulky (Tabulka 2) a z vysvětlovacího modulu a pokusíme se ručně přepočítat váhy určené systémem dle teorie vysvětlené v této kapitole.

**Tabulka 2: Tabulka odpovědí uživatele v rámci konzultace nemoce při rozsahu vah od -3 (jistě ne) až po 3 (jistě ano). Atribut teplota je zadáván numericky.**

| Atribut          | Výrok            | Odpověď |
|------------------|------------------|---------|
| Pacient je bledý | Pacient je bledý | 1       |
| Dýchavičnost     | Dýchavičnost     | 1       |
| Teplota          | Teplota          | 37      |
| Další příznaky   | Kašel            | 3       |
| Další příznaky   | Rýma             | 1       |

**Tabulka 3: Tabulka výsledků konzultace nemoce.**

| Výroky                | Min váha < | Max váha | Stav    | Druh | Komentář |
|-----------------------|------------|----------|---------|------|----------|
| diagnoza (TBC)        | 1,779      | 1,779    | finální | cíl  |          |
| diagnoza (Nachlazení) | 1,285      | 1,285    | finální | cíl  |          |
| diagnoza (Chřipka)    | -0,857     | -0,857   | finální | cíl  |          |



**Obrázek 35: Graf vysvětlovacího modulu pro cílový výrok TBC. Zelené identifikátory pravidel zapůsobily kladným způsobem v rozhodování pro variantu či lépe cílový výrok TBC.**

Znalosti, které mají vliv při rozhodování či výpočtu konečné váhy cílového výroku TBC:

```

c1 IF bledost THEN diagnoza(TBC) [1,0]: 0,333
c2 IF dychavicnost THEN plice[2,0]: 0,667
c3 IF priznaky(kasel) THEN plice[2,0]: 2,000
c6 IF plice THEN diagnoza(TBC) [2,0]: 1,548
c9 IF NOT(bledost) THEN diagnoza(TBC) [-1,0]: 0,000
c10 IF NOT(dychavicnost) AND NOT(priznaky(kasel)) THEN
plice[-3,0]: 0,000
c12 IF NOT(plice) THEN diagnoza(TBC) [-3,0]: 0,000
    
```

Všimněme si nejdříve pravidel **c2**, **c3**, která vedou k mezilehlému výroku **plice** a mají (více či méně) splněný předpoklad. **Obě pravidla mají váhu 2,0**. Předpoklad pravidla **c3** je splněn s maximální jistotou, takže jeho příspěvek je roven jeho váze, tj. 2,0. Oproti tomu předpoklad pravidla **c2** je splněn pouze s váhou 1 (odpověď trochu ano), tj. z jedné třetiny, takže **příspěvek tohoto pravidla je 0,667** (ve standardním přístupu k práci s neurčitostí je dán součinnou funkcí **CTR=2\*1/3=0,667**). **Výsledná váha výroku plice** je pak dána agregací těchto dvou příspěvků **pomocí standardní funkce GLOB** (pozor – před použitím funkce musíme váhy převést na jednotkový interval a poté výsledek opět přepočítat na interval [-3;3]):

$$\begin{aligned}
 & \text{GLOB}(2,0/3; 0,667/3) = \\
 & = (0,667 + 0,222) / (1 + 0,667 * 0,222) = \\
 & = 0,889 / 1,148 = 0,774
 \end{aligned}$$

Váha výroku **plice** tedy bude **2,323**. Pravidlo **c6** z něj pomocí standardní funkce **CTR** odvodí příspěvek  $2,323/3 * 2,0 = 1,549$  k cílovému výroku **diagnoza(TBC)**. Příspěvek pravidla **c9** k témuž výroku je 0,333, neboť váha předpokladu i váha samotného pravidla je 1. Spojení obou příspěvků určuje opět funkce **GLOB**:

$$\begin{aligned}
 & \text{GLOB}(1,549/3; 0,333/3) = \\
 & = (0,516 + 0,111) / (1 + 0,507 * 0,111) = \\
 & = 0,627 / 1,057 = 0,593,
 \end{aligned}$$

takže výsledná váha výroku **diagnoza(TBC)** je 1,779.

Pokud bychom pro zpracování neurčitosti v této konzultaci použili logický inferenční mechanismus, situace by se značně změnila: Pravidlo **c2** by mělo pomocí přísné logické funkce **CTR** nulový příspěvek, neboť součet vah předpokladu (1/3) a pravidla (2/3) není větší než 1. Výrok **plice** je tak odvozen pouze pravidlem **c3** s váhou 2. Příspěvek pravidla **c6** je pak roven  $2/3 + 2/3 - 1 = 1/3$ , tj. 1 v naší škále. Pravidlo **c9** bude mít z logického hlediska také nulový příspěvek (součet neurčitostí předpokladu a pravidla je pouze 2/3, což je málo na to, aby mělo nějaký efekt). Výsledná váha výroku **diagnoza(TBC)** je tak při použití logického inferenčního mechanismu 1,0.

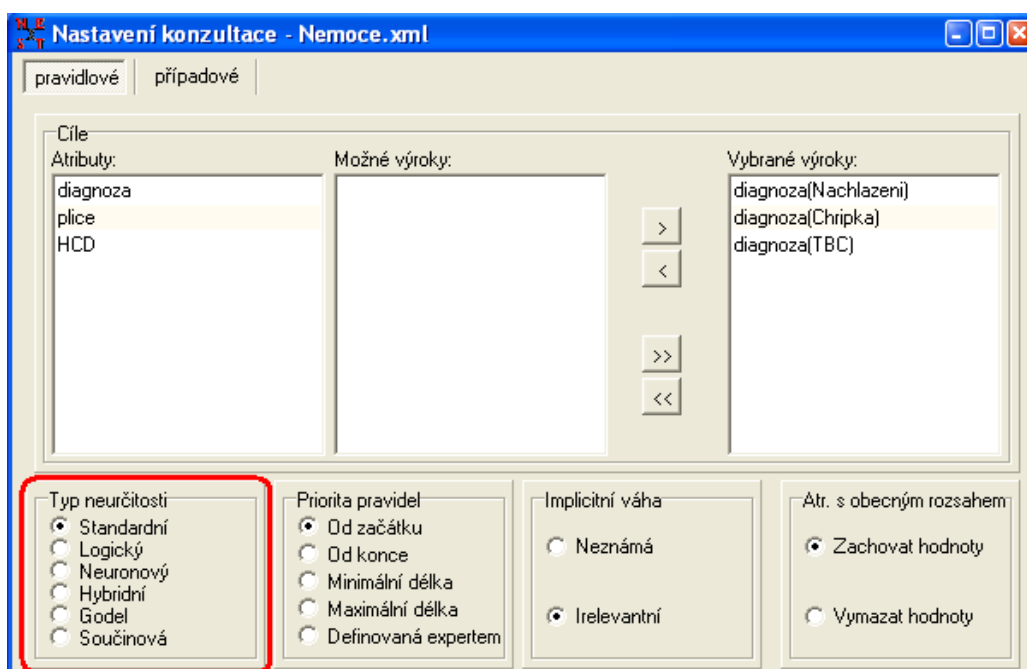
Poznamenejme, že tento příklad je spíše atypický, nejčastěji nedochází k tak výrazným rozdílům a pořadí cílových výroků podle získaných vah bývá obdobné ve všech implementovaných modelech práce s neurčitostí.

#### 4.4 Srovnání přístupů práce s neurčitostí v NEST

Jak již bylo zmíněno, jedním z účelů systému SAK a potažmo NEST bylo mimo jiné umožnit srovnání přístupů zpracování neurčitosti. Systém NEST umožňuje tvůrci báze znalostí zadat a porovnat výsledky konzultací při následujících přístupech práce s neurčitostí:

- Standardní přístup – založený na pravděpodobnosti.
- Logický přístup – založený na fuzzy logickém přístupu.
- Neuronový přístup – založený na přístupu z oblasti neuronových sítí.
- Hybridní přístup – je přístup kombinující standardní a fuzzy logický přístup.

Při průchodu bázi znalostí NEST přepočítává váhy pomocí funkcí popisovaných v kapitole 4.2 přístupem, který je zadán při vytváření báze znalostí. Uživatel však může přístup k zpracování neurčitosti změnit viz Obrázek 36 před zahájením vlastní konzultace.



Obrázek 36: Volba přístupu při zpracování neurčitosti v NEST

Za účelem srovnání výsledků doporučujeme vyhodnotit konzultaci standardním způsobem a výsledky konzultace jakož i vlastní odpovědi uložit. Před dalším spuštěním konzultace jednoduše nahrajete uložené odpovědi a změníte typ zpracování neurčitosti (Obrázek 36). Uložené výsledky pak lze jednoduše porovnat např. v tabulkovém kalkulátoru MS-Excel.



## KONTROLNÍ OTÁZKA

1. Jakých 5 základních funkcí používá systém NEST pro práci s neurčitostí a k jakém účelu každá z nich slouží?
  2. Jaké inferenční mechanismy nabízí systém NEST?
- 



## SHRNUTÍ KAPITOLY

V této kapitole jsme se seznámili s přístupem ke zpracování neurčitosti, který je zabudován v systému NEST, včetně čtyř typů inferenčních mechanismů, které systém NEST nabízí. Podívali jsme se dovnitř fungování inference v NESTu a naučili jsme se, jakým způsobem je v NEST inference prováděna.

---



## ODPOVĚDI

1. Jsou to:
  - NEG - negace
  - CONJ - konjunkce
  - DISJ - disjunkce
  - CTR - příspěvek pravidla k váze závěru
  - GLOB – složení příspěvků více pravidel ke stejnému závěru.
2. **Standardní, logický, neuronový a hybridní.** Standardní inferenční mechanismus vychází ze způsobu práce s neurčitostí v systémech MYCIN a PROSPECTOR, logický inferenční mechanismus je založen na vícehodnotové logice – funkce CTR je fuzzy modus ponens a funkce GLOB je fuzzy disjunkce v Łukasiewiczově<sup>4</sup> logice, neuronový inferenční mechanismus je inspirován modelem neuronu a hybridní inferenční mechanismus je kombinací standardního a logického.

---

<sup>4</sup> Jan Łukasiewicz - (21.12.1878 – 13.2.1956) polský matematik a filozof; tvůrce více hodnotové logiky někdy také nazývané jako Łukasiewiczova logika.



## **POZNÁMKY**

## 5 Aplikace v NEST

### RYCHLÝ NÁHLED KAPITOLY



Kapitola Aplikace v NEST bude ryze praktická. Pokusíme si nastínit jak vybrat vhodné téma (např. pro seminární práci), kterak postupovat při samotném uchopení problému a zorientování se, nalezení atributů, započítání vlastní práce v NEST editoru a tvorby pravidel, dále možnostem při ladění báze znalostí a možností výstupu použitelných pro dokumentační účely.

### CÍLE KAPITOLY



Cílem kapitoly je detailně seznámit čtenáře s praktickými aspekty při tvorbě báze znalostí pomocí NEST editoru. Kapitola je tedy zaměřena na získání ryze praktických znalostí a dovedností pomocí názorných ukázek, ve které je vytvořena báze znalostí pro expertní systém doporučující vhodný šicí stroj.

### KLÍČOVÁ SLOVA KAPITOLY



výběr problému, založení báze znalostí, konstrukce báze, ladění báze, tvorba dokumentace

### 5.1 Výběr problému a grafická reprezentace

Jak jsme již dříve zmínili, vhodná témata nebo oblasti pro zapojení expertního systému jsou takové, ve které v praxi existuje osoba, kterou bychom mohli nazvat expertem, a zároveň si umíme představit, že konzultace nebo porada s tímto expertem by mohla být prováděna po telefonu, tedy bez jeho fyzické přítomnosti, a to v délce nepřesahující přibližně čtvrt hodiny.

Obecně, i s přihlédnutím k již zpracovávaným pracím z dřívějších let, lze říci, že přijatelné a dobře zpracovatelné jsou aplikace na téma doporučení vhodného výrobku (namátkou doporučení: notebooku, auta, dovolené, mobilního telefonu, vhodného sportu pro dítě, domácího mazlíčka, finančního produktu atd.), nebo diagnostické úlohy zakládající se na zákla-

dech psychologických, lékařských či jiných dotazníků (např. diagnostika možnosti vrozených vad, diagnostika typu temperamentu, diagnostika vhodné role v týmu atd.) a nebo diagnostiky různých závad a poruch (nahrazení help linky, diagnostika poruch v počítačové síti, chyby počítačů, aut, vhodný postup při provádění první pomoci atp.).

V rámci volby oboru či hodnocení vhodnosti problému ke zpracování metodikou a prostředky znalostního inženýrství samozřejmě nesmíme zapomenout na znalosti v daném (vybraném) oboru. Musíme mít zdroj potenciačních znalostí. Znalosti můžeme formulovat na základě studia a analýzy odborných článků (v tištěné literatuře či internetu), nebo na základě rozhovorů s expertem v dané oblasti případně na základě syntézy a porovnávání obou zmiňovaných zdrojů.

Vybrali-li bychom si oblast ke zpracování např. doporučení vhodného fotoaparátu, můžeme s klidným svědomím prohlásit, že toto je vhodná oblast, neboť se jedná o **diagnostický problém** – doporučení vhodného výrobku na základě zjištění potřeb konzultující osoby. V praxi v roli experta vystupují obchodní asistenti prodeje s fotografickými potřebami či elektro prodejen a lze říci, že v rámci tohoto poradenství jsou schopni zákazníkovi pomoci s výběrem v rámci několika málo minut. Také si jistě dokážeme představit, že by byla takováto konzultace proveditelná bez nutnosti setkání experta (prodávajícího) a konzultujícího (kupujícího) či fyzické přítomnosti doporučovaného zboží (je možné, jak je to ostatně obvyklé v rámci e-shopů, suplovat obrázky či odkazy na konkrétní zboží). Konkrétní znalosti lze tedy čerpat od konkrétního prodejce či na základě prostudování článků zabývajících se problematikou výběru vhodného aparátu.

Zkuste se nyní zamyslet nad vám blízkou oblastí, ve které znáte jistá doporučení či dokonce pravidla výběru či hodnocení, nebo, ve které znáte osobu (experta), jež těmito znalostmi vládne a byla by ochotna s vámi spolupracovat na potenciační seminární práci.

## 5.2 Příklad zpracování báze znalostí

V rámci kapitoly si nastíníme vlastní postup při zpracování báze znalostí na úloze doporučení vhodného šicího stroje. Stručnost popisu bude podřízena cíli předvést postup výběru tématu, tvorby grafické reprezentace, identifikace atributů, tvorbu pravidel, odladění báze a možnostmi exportu informací pro dokumentaci.

Vybrali jsme si tedy vhodnou a nám přitažlivou oblast doporučení vhodného šicího stroje. Je zřejmé, že můžeme úlohu pojmout ve dvou rovinách:

1. obecné doporučení stroje pro občasné šití (pro začátečníky), pro domácí kutily (poloprofesionální) a nebo pro spíše profesionální účely, a to nezávisle na konkrétních modelech či značkách,
2. doporučení konkrétního typu a značky z omezené množiny zahrnutých modelů, na základě zjištění potřeb konzultujícího.



Na tomto místě je potřeba zdůraznit slova **doporučení na základě zjištění potřeb**. Což se může a většinou liší od zjišťování preferencí, tedy zjišťování, kterých vlastností si uživatel více váží, či které by preferoval (což je základní úlohou systému AHP<sup>5</sup>).

Ve vybrané oblasti si stanovíme cíl: výběr konkrétního modelu – čili program na konci konzultace po zjištění potřeb a jejich ohodnocení váhami na základě odpovědí od uživatele a pro projití či přepočtu inferenčním mechanismem doporučí (odliší číselnou vahou) jeden ze tří šicích strojů značky Brother: LS2125, BC2100 a nebo NV400. Stanovili jsme si tedy konkrétní cíle, které v již dříve rozebíraném názvosloví budeme reprezentovat jako cílový atribut množinový, který bude nabývat tří výrokových stavů (tj. konkrétních modelů). Tyto stavy můžeme zaznamenat do grafu popisujícího bázi (viz Obrázek 37).

Výběr cílů potažmo i zhodnocení vhodnosti námi vybrané oblasti pro např. seminární práci je také limitován možností, zda jsme na základě analýzy cílových výroků (chceme-li dodržet více-stupňovost báze) schopni identifikovat dostatek společných i odlišných vlastností těchto výroků, které by mohly být reprezentovány výroky a atributy v mezilehlé rovině. Potažmo jsme-li schopni formulovat laické dotazy na potřeby klienta, ze kterých jsme dále schopni vyvozovat a usuzovat pomocí obecně platných pravidel z oboru platnost zmiňovaných výroků mezilehlé roviny. Uvedme si 3 specifikace vybraných šicích strojů:

| LS2125  | BC2100  | NV400  |
|---|---|--|
| <ul style="list-style-type: none"> <li>• Cena: 2.701,00 Kč.</li> <li>• Jednoehlový základní šicí stroj s kyvným chapačem a volným ramenem a výbornou cenou.</li> <li>• 7 druhů stehů.</li> <li>• 21 šicích technik.</li> <li>• Pevně dané programy.</li> <li>• Šití knoflíkové dírky na 4 kroky.</li> </ul> | <ul style="list-style-type: none"> <li>• Cena: 6.289,00 Kč.</li> <li>• Elektronický šicí stroj s rotačním chapačem a volným ramenem.</li> <li>• LCD displej pro jednoduché nastavení parametrů šití.</li> <li>• Plná automatika pro šití knoflíkových dírek, 5 druhů.</li> <li>• 40 voleb stehů.</li> </ul> | <ul style="list-style-type: none"> <li>• Cena: 17.017,00 Kč.</li> <li>• Špičkový elektronicky řízený šicí stroj s přímou volbou funkcí tlačítky, LCD display, pro náročné zákazníky, robustní konstrukce, rozšířená šicí plocha v ceně, kroková a patchworková patka v příslušenství. Šicí stroj pro náročné zákazníky.</li> </ul> |

<sup>5</sup> [Analytic Hierarchy Process](#) – analytický hierarchický proces, se kterým se studenti OPF SU mohou seznámit blíže v rámci kurzu Rozhodovací analýza pro manažery či konkrétně ve skriptu prof. RNDr. Jaroslava Ramíka CSc. *Vícekritériální rozhodování - analytický hierarchický proces (AHP)*. ISBN 80-7248-047-2.

- Volné rameno.
- Zapošití a šití zpět páčkou.
- Regulace napětí horní nitě.
- Možnost šití dvoujehlou.
- Příslušenství: patka pro šití knoflíkové dírky, zipová patka, patka pro našívání knoflíků, sada jehel - 3 ks, dvoujehla, látací destička, přídavný kolík cívky, 3 ks cívek, šroubovák.
- Volné nastavení délky stehu a šířky cik – caku.
- Automatický navlékač horní nitě.
- Blesková příprava spodní nitě.
- Šestinásobně dělený podavač.
- Tlačítko pro zafixování polohy patky.
- Rychloupínací systém pro výměnu patek.
- Ovládání podavače zasunutím pod stehovou desku pro látání a vyšívání.
- Osvětlení šicí plochy LED diodami.
- Dobře přístupné ovládací prvky.
- Volné rameno pro šití uzavřených kruhů - nohavice, rukávy.
- Příslušenství: patka pro šití knoflíkové dírky, overloková patka, patka pro vyšívání monogramů, patka pro přišívání zipů, patka pro slepý steh, patka pro přišívání knoflíků, sada jehel + dvoujehla, 4 ks
- V paměti 294 druhů stehů a 10 druhů knoflíkových dírek šitých automaticky.
- 3 sady znaků.
- Přímá volba základních stehů.
- 10 pamětí pro uložení nejpoužívanějších šicích nastavení.
- Vlastní návrh vyšívacích stehů.
- Programovatelný automatický odstřih.
- Čelní uložení cívky nití.
- Prostor pro šití i rozměrných dílů, velký šicí stůl součástí příslušenství.
- Dobře přístupné ovládací prvky.
- Schránky s příslušenstvím.
- Šestinásobně dělený podavač.
- Regulace přítlaku patky.
- Tlačítko pro zafixování polohy patky.
- Automatický navlékač nitě.

cívky, páráček, šroubovák, přídavný kolík cívky, štěteček k čištění.

- Blesková příprava spodní nitě.
- Osvětlení LED diodami.
- Volné rameno pro šití rukávů.
- Volba polohy jehly při zastavení.
- Světelná signalizace pro START a STOP.
- Kolenní páka.
- Příslušenství: prošívací patka pod pružinou, krácející patka, patka pro knoflíkovou díрку, patka obnitkovací, patka pro monogramy, patka pro všívání zipu, patka pro přišívání knoflíků, patka s vodítkem, teflonová patka, patka s otevřenou špičkou, dvoujehla, páráček, šroubováky, kartáček.....

Z výše uvedeného vyplývá, že stroje intuitivně můžeme rozlišit na stroj základní (LS2125 - nabízí základní funkce a je za nejnižší cenu), stroj střední třídy (BC2100 – nabízí široké možnosti stehů a šicích technik spolu s možností vlastního nastavení, více příslušenství) a stroj nejvyšší třídy (NV400 – nejlépe vybavený co do příslušenství i funkcí).

Rozšíříme-li, pro jaký druh zákazníka doporučíme základní zmíněné druhy strojů, pak bychom to mohli na základě konzultace s krejčovou či vlastních zkušeností vyhodnotit následovně:

- Základní typ šicího stroje vhodný pro občasné šití (uživatel nemá nároky na rychlou přípravu stroje k šití), pro šití typu drobných správek (látání, sešívání roztrhaných švů atp.), sešívání dílů z obvyklých druhů látek (nebude využívat speciálních

patek) základními stehy (nebude používat ozdobných stehů a nebude mít potřebu upravovat délku a šířku stehů). Obecně tedy vhodný pro uživatele, kteří občas něco potřebují spravit nebo vyrobit a nemají ambice na profesionalitu provedení.

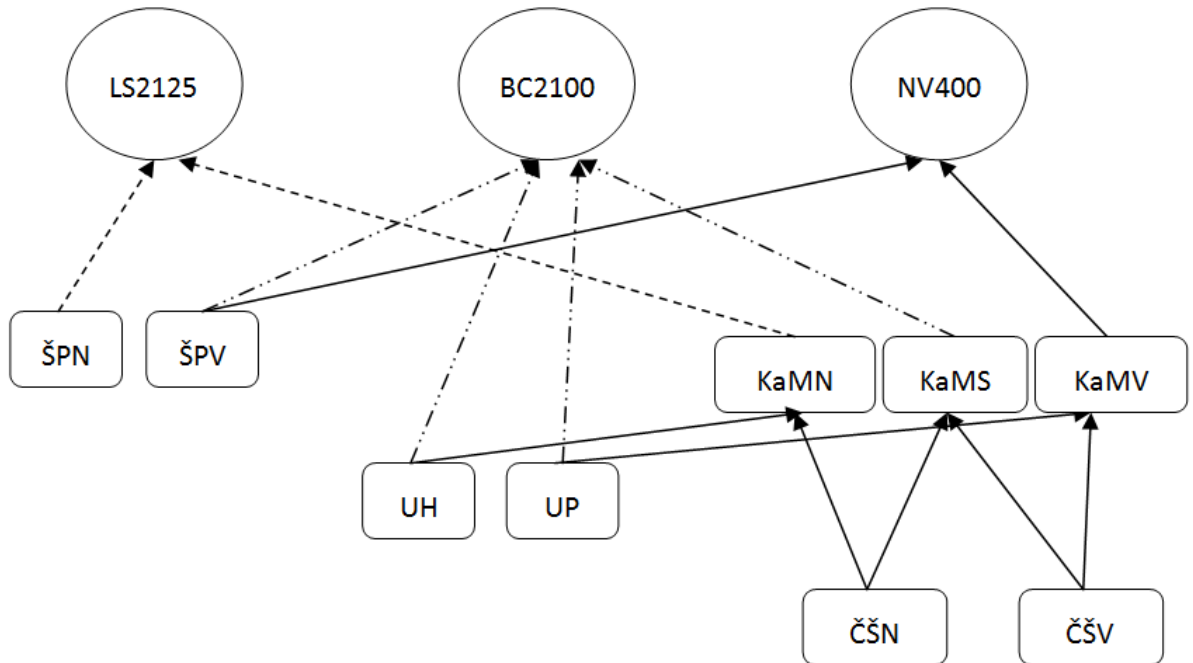
- Šicí stroj střední třídy by měl být kompromisem mezi základní a nejvyšší třídou. Tedy bude doporučován pro domácí ne příliš časté šití konfekce tedy výrobků, u kterých je potřeba mít jistou možnost variability v šířce a délce stehu, dále větší výběr stehů, předpokládáné jsou taktéž požadavky na automatické šití knoflíkové dírky a také více komfortu při přípravě šicího stroje. Protože předpokládáme šití konfekce je předpoklad, že uživatel může mít nároky na šití i z tenkých, z mačkových či kluzkých materiálů, nebo šití patchworku, nebo přišívání stuh, prýmků a perel či všívání zipů „skrytým“ způsobem (stroj nabízí možnost dokoupení rozšiřujícího příslušného příslušenství). Mohl by být vhodný i jako druhý stroj pro zařízení dílny pro oprávků (tedy pro obdobné typy šití jako nižší třída s tím rozdílem, že by se jednalo o časté šití).
- Šicí stroj nejvyšší třídy pak budeme doporučovat pro zákazníky profesionály, tzn. kteří šijí velmi často, potřebují co nejvíce usnadnit manipulaci se strojem a jeho přípravu k šití, vyžadují maximální komfort a možnosti nastavení, a vzhledem k rozsahu výrobků vyžadují široké možnosti příslušenství.

Z výše uváděných vlastností a charakteristik zákazníka, pro kterého by byl stroj určen bychom mohli učinit následující závěr: pro doporučení cílového typu stroje je potřeba v rámci konzultace zjistit a rozlišit **úroveň šití** profi (UP) či hobby (UH); **šíři použití** stroje vysoká (ŠPV) a nízká (ŠPN); **četnost šití** nízká (ČŠN) a vysoká (ČŠV); **komfort** obsluhy a **možnosti** nastavení či rozšíření nízká (KaMN), střední (KaMS) a vysoká (KaMV). Identifikované vlastnosti a vztahy mezi nimi můžeme zaznamenat do grafu (viz Obrázek 37), který nám poslouží i pro budoucí tvorbu pravidel (vztahy reprezentované pomocí šipek mezi jednotlivými atributy). Tyto vlastnosti mimo četnosti šití budou vyhodnocovány na základě základních dotazů. Mezi základní dotazy pak zařadíme dotazy na profesionalitu výstupu, šití svatebních, plesových šatů, společenských a slavnostních oděvů, šití patchworku, šití ozdobných výšivek, vyšívání monogramů, látání, oprávek, prošívání a sešívání vícevrstvých dílů, šití z kluzkých, jemných či krčivých materiálů (atd.). Základní dotazy již na obrázku Obrázek 37 zachycené nejsou, neboť by se stal nepřehledným.

Z výše uvedeného odstavce to není tak zřejmé, nicméně při stanovování atributů v rámci přípravy projektu se nám může stát, že některé z atributů se v jistých dimenzích mohou z menší části překrývat. Proto nepodceňujte vytváření si své vlastní dokumentace již při analýze problému, ve které nejen atributy nazvete, ale i blíže popíšete, tak aby při návratu od přerušené práce nedocházelo k logickým posunům, způsobeným zapomínáním. Při práci s NEST editorem pak můžeme využít k tomuto účelu již zmiňované komentáře jednotlivých atributů.

Mějme, prosíme, neustále na paměti, že při zpracování úlohy ve znalostním inženýrství neexistuje obecně správné řešení. Každý znalostní inženýr řeší úlohu logickou úvahou tak, aby dosáhla požadovaných cílů nebo subjektivně vytyčených cílů – čili úloha bývá podmíněna účelem a logika řešení má tuto účelovost ctít. Je tedy obtížné, zůstaneme-li v rovině

našeho kurzu, vyjadřovat se k návrhům na seminární práci pokud tutor (přednášející, cvičící atp.) nezná mantinely, cíl, účel či nástin logiky problému, který student hodlá zpracovávat.



Obrázek 37: První náčrt budoucí báze znalostí a vztahů mezi některými jejími prvky.

### 5.3 Založení nové báze

Po přípravné analýze námi zvoleného problému můžeme směle spustit NEST editor a započít vlastní práci. Nejprve (např. volbou z menu Soubor/Nový nebo klávesovou zkratkou CTRL+N) založíme novou bázi.

Obrázek 38: Zadání globálních parametrů uživatelem.

Je vhodné v okně „Globální parametry“ vyvolaném předchozí akcí uživatele ihned zadat známé parametry a zvolit volitelné parametry (viz Obrázek 38). Doplníme krátký popis báze, tak aby bylo jasné k jakým účelům je báze konstruována, co je jejím cílem či výsledkem. Dále zadáme jména nebo jméno experta (kdo je to expert jsme již definovali v Kapitole 1.2) a znalostního inženýra.

Další volby, jimiž ovlivníme chování báze znalostí či spíše chod inferenčního mechanismu, jsou tyto:

- **Rozsah vah** - interval  $[-\text{rozsah vah} ; \text{rozsah vah}]$ , ve kterém budou zadávány a zobrazovány veškeré váhy v bázi. V tomto intervalu pak zadávejte i veškeré váhy při tvorbě báze. Pokud rozsah vah změníte, veškeré váhy se automaticky přepočítají. Tento parametr byste měli určitě zadat při založení nové báze, abyste mohli jednotlivé váhy zadávat ve správném intervalu. Běžně se v bázi znalostí používá rozsah vah 1 nebo 3. Je celkem jedno jaký rozsah si zvolíte systém na pozadí váhy nakonec přepočítává na rozsah 1 a pro zobrazení výsledků zase zpět do rozsahu zadaným uživatelem. Prakticky: někteří lidé jsou zvyklí lépe hodnotit nebo vyjadřovat váhy procentuálně (pak bychom preferovali rozsah vah 1). Rozsah vah 3 má tu výhodu, že respektuje obecnou schopnost lidí ještě dobře rozlišovat na škále 7 stupňů, což rozpětí  $-3$  až  $3$  v celých číslech poskytuje.
- **Globální práh platnosti kontextu** udává váhu, kterou musí překročit kontext, aby byl splněný (a aby tedy bylo použito pravidlo, u kterého je kontext uveden). U každého pravidla lze zadat individuální práh, pokud však není uveden, je použit tento globální práh.
- **Globální práh platnosti předpokladu** udává váhu, kterou musí překročit logické pravidlo (resp. jeho minimální váha), aby bylo vyhodnoceno jako splněné. Opět lze u jednotlivých pravidel nastavit individuální práh.
- **Inferenční mechanismus** – určí jakým způsobem bude defaultně přistupovat inferenční mechanismus k zpracování neurčitosti (viz také kapitola 4 Neurčitost v NEST).
- **Implicitní váha** určuje, jakou váhu mají dostat výroky, kterým nebyla váha určena jinak (například výpočtem nebo dotazem). Tuto váhu dostanou například všechny výroky odvozené od multiple atributu, kterým uživatel v dotazu nepřihradil jinou váhu. Implicitní váha může být buď neznámá, tj. váha  $[-1 ; 1]$  (upravená na rozsah příslušné báze), nebo irelevantní, tj. váha  $[0 ; 0]$ .

- **Globální priorita pravidel** určuje, v jakém pořadí bude systém zpracovávat pravidla, nastane-li situace, že jich může použít více najednou. Od začátku znamená, že systém dá přednost tomu pravidlu, které je v bázi znalostí uvedeno dříve, od konce znamená přesný opak. Minimální délka znamená, že systém použije nejprve pravidlo, které má kratší předpoklad, při maximální délce vezme nejprve pravidlo s delším předpokladem. Při volbě definovaná expertem budou brána pravidla v pořadí podle priority, kterou určil expert při tvorbě báze. Pokud tato priorita není u pravidel uvedena, systém se zachová jako při volbě od začátku.

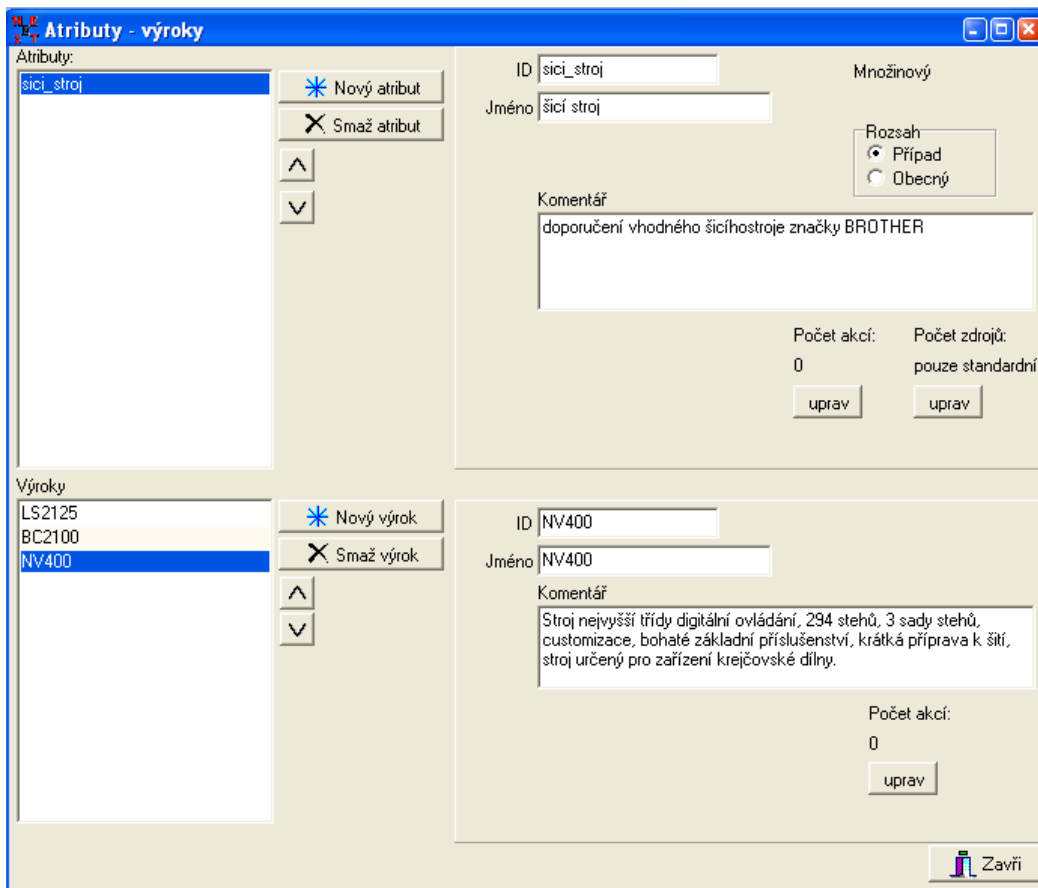
V zásadě mimo určení rozsahu vah a doplnění popisných informací (popis báze, expert, znalostní inženýr) můžeme nastavení ponechat ve výchozím stavu.

V tuto chvíli bychom (pokud samozřejmě chcete dle výkladu zpracovávat analogicky např. svůj problém) měli mít ujasněno, jakým tématem se zabýváme, identifikovali jsme cílové výroky a také některé mezilehlé a základní výroky. Stěžejní výroky a stěžejní vazby jsme si zachytili pomocí grafu. Založili jsme novou bázi, do které jsme zadali globální vlastnosti a nyní se můžeme vrhnout na vlastní konstrukci báze.

## 5.4 Konstrukce báze znalostí

Na základě grafického znázornění výroků a vztahů mezi nimi (vlastního nebo můžeme pracovat s grafem Obrázek 37) můžeme uvažovat, jaké v bázi použijeme atributy a které z v grafu zachycených výroků jim přiřadíme jakožto jejich výrokové stavy (samozřejmě, že stále máme v pozadí na zřeteli i ostatní vlastnosti v grafu nezachycené a popsané při našem modelovém případě v kapitole 5.1).

V případě doporučení šicího stroje z grafu vidíme, že máme tři konečné cílové výroky, čili toto jsou výrokové stavy příslušného atributu, který si můžeme nazvat „šicí stroj“. Konkrétní výroky tohoto atributu jsou tedy jednotlivé typy (viz Obrázek 39). Typ atributu použijeme množinový, neboť jednotlivé výroky mohou samozřejmě platit současně s rozdílnou vahou (prodávač v obchodě mimo to, že zákazníkovi něco doporučí, se také často vyjadřuje k alternativám, které též hodnotí).



**Obrázek 39: Zadávání cílových výroků pro cvičný případ doporučení šicího stroje.**

Obdobným způsobem pak zadáme i ostatní atributy tedy šíře použití (ŠP), úroveň šití (U), komfort obsluhy a možnosti (KaM) také jako množinové atributy. Atribut četnost šití (ČŠ) bud atributem numerickým, kdy budeme zjišťovat jak často uživatel bude se strojem během měsíce pracovat v rozmezí 0-31. Odpověď z intervalu 0-4 považujeme za odpověď zcela jistě ČŠ nízká, odpověď z intervalu 15-31 zcela jistě ČŠ vysoká, odpověď z intervalu 4-8 ještě ČŠ nízká (nízká s vahou nižší než 1) a odpověď z intervalu 4-15 za již vysoká (vysoká s vahou nižší než 1).

Mezi základní atributy (tedy atributy, které budou figurovat pouze v předpokladové části pravidel a z pohledu konzultujícího vystupují v roli otázek) zařadíme mimo numerického atributu další, dotazující se konzultujícího jakým způsobem bude stroj využívat, jaké produkty bude na stroji zhotovovat, či jaké šicí techniky hodlá používat. Z těchto odpovědí pak lze usuzovat na nutnost a škálu doplňků, možnosti nastavení šicího stroje a možnosti využívání škály stehů, což jsou nejvýznamnější vlastnosti odlišující jednotlivé typy strojů.



Konkrétně do báze znalostí bychom mohli zavést atributy (údaje či upřesnění uváděné v závorkách, můžeme uvádět do komentářů – poslouží pro lepší orientaci a pochopení otázky konzultující osobou při vlastní konzultaci):

- Binární: Podnikání (šití na šicím stroji Vás živí – podnikáte v oboru švadlena, krejčovská atp.)? Šití známým (šijete širšímu okruhu svých známých)? Šití pro nejužší rodinu?
- Množinové (jednotlivé položky z následujících atributů by samozřejmě bylo možno řešit jako samostatné binární atributy, nicméně výroky v attributech jsou logicky seskupeny a navíc přinášejí přehled do NEST editoru):
  - Šicí techniky (ohodnoťte vahou v přípustném rozmezí, prosím, z množiny následujících šicích technik, které budete používat): záplatování, sešívání dílů, obšívání (ubrusy, povlečení, závěsy, záclony), patchwork (někdy také quilting), prošívání, knoflíkové dírky, vyšívání monogramů, entlování se začistěním ořepků.
  - Konfekce (ohodnoťte vahou v přípustném rozmezí, prosím, z množiny následujících možných finálních produktů, které budete produkovat): drobné opravy, šití společenských oděvů a konfekce, šití domácích oděvů, šití pracovních oděvů, šití svatebních šatů, šití plesových šatů.
  - Ostatní (ohodnoťte vahou v přípustném rozmezí, prosím, z množiny následujících možných finálních produktů, které budete produkovat): výroba textilních hraček, ozdob a dekorací, ochranných pomůcek, obalů, tašek.

Přehled zavedených atributů a jejich výroků je ilustrován pomocí obrázku Obrázek 40.

| Atributy   | Výroky                         | Pravidla |
|------------|--------------------------------|----------|
| sici_stroj | sici_stroj(LS2125)             |          |
| ŠP         | sici_stroj(BC2100)             |          |
| U          | sici_stroj(NV400)              |          |
| KaM        | ŠP(ŠPN)                        |          |
| ČŠ         | ŠP(ŠPV)                        |          |
| Podnikání  | U(UF)                          |          |
| Známí      | U(UH)                          |          |
| Rodina     | KaM(KaMN)                      |          |
| Techniky   | KaM(KaMS)                      |          |
| Konfekce   | KaM(KaMV)                      |          |
| Ostatní    | ČŠ(ČŠN)                        |          |
|            | ČŠ(ČŠV)                        |          |
|            | Podnikání                      |          |
|            | Známí                          |          |
|            | Rodina                         |          |
|            | Techniky (poklepáním editujte) |          |
|            | Techniky(sešívání dílů)        |          |
|            | Techniky(obšívání )            |          |
|            | Techniky(patchwork )           |          |
|            | Techniky(prošívání)            |          |
|            | Techniky(knoflíkové dířky)     |          |
|            | Techniky(vyšívání monogramů)   |          |
|            | Techniky(entlování)            |          |
|            | Konfekce(opravy)               |          |
|            | Konfekce(spol.šaty)            |          |
|            | Konfekce(dom.šaty)             |          |
|            | Konfekce(prac.šaty)            |          |
|            | Konfekce(svatebky)             |          |
|            | Konfekce(plesovky)             |          |
|            | Ostatní(hračky)                |          |
|            | Ostatní(ozdoby)                |          |
|            | Ostatní(ochran.pomucky)        |          |
|            | Ostatní(obaly)                 |          |
|            | Ostatní(články)                |          |

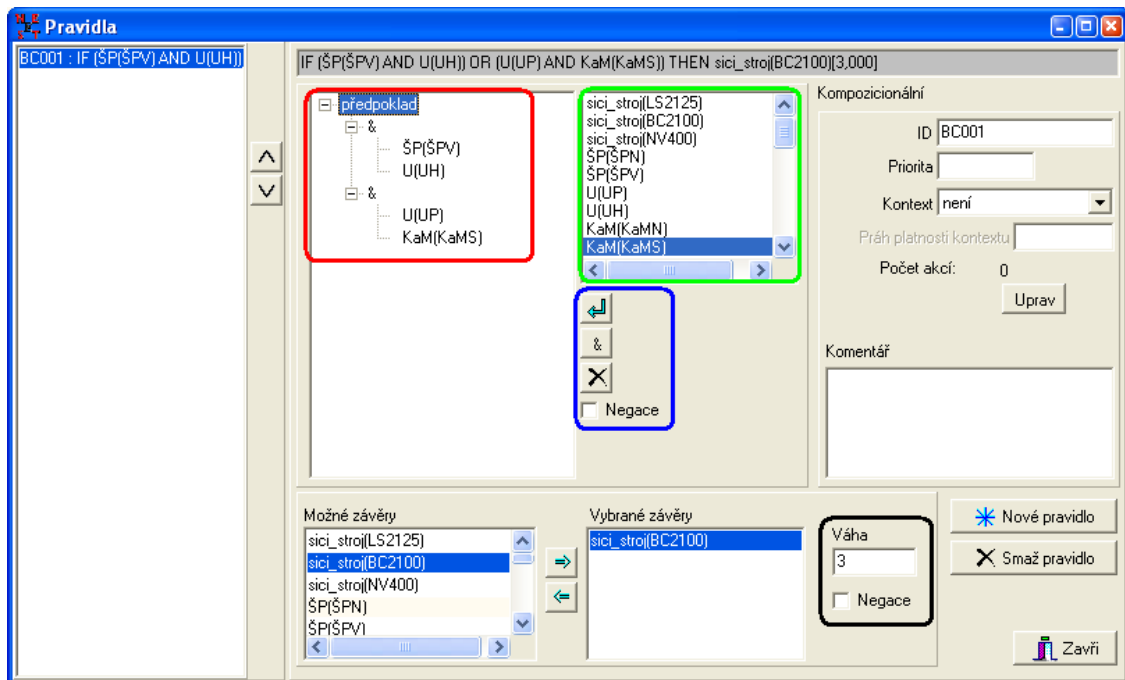
**Obrázek 40: Přehled atributů a jejich výroků pro bázi znalostí šicí stroj.**

Nyní se nacházíme v situaci, kdy jsme založili novou bázi znalostí a pomocí grafického znázornění vazeb mezi jednotlivými atributy, respektive jejich výrokovými stavy, které jsme identifikovali na základě např. analýzy cílových výroků a jejich vlastností, bychom mohli začít tvořit pravidla. Není chybou, když během další práce zjistíte, že je vhodné nebo dokonce potřebné do báze vložit nový atribut nebo doplnit dříve zapomenutý či neidentifikovaný výrokový stav (čím méně takových zásahů budete mít, tím jste pravděpodobně kvalitněji provedli analýzu problému). Celý proces vytváření a odlaďování báze znalostí je proces, kde má fungovat zpětná vazba.

Pro tvorbu vlastních pravidel již máme potřebné předpoklady splněny. Provedli jsme analýzu našeho problému a graficky jsme náš problém zachytili. Zadali jsme pomocí NEST Editoru atributy a jejich výrokové stavy. Nebrání nám nic tvořit pravidla.

Zopakujme si, že pravidlo se skládá z předpokladu, kde může figurovat kterýkoli z výrokových stavů všech typů atributů a závěru pravidla, kde mohou být pouze výrokové stavy atributů množinových a binárních.

Nová pravidla zadáváme např. vyvoláním nabídky menu **Edituj/Pravidla** nebo dvojklikem pravým tlačítkem v horní polovině sloupce pravidla, vyvoláme tak okno pro zadávání nového pravidla (viz Obrázek 29). Po stlačení tlačítka **Nové pravidlo** výběru typu pravidla (pracujeme s kompozicionálními pravidly).



**Obrázek 41: Editační okno pro editaci a vytváření pravidel (barevně jsou zatrženy některé důležité části okna).**

Okno pro editaci a zadávání pravidel je znázorněno na obrázku: Obrázek 41. V levém sloupci se nachází nebo bude nacházet seznam již existujících pravidel. V horním šedém řádku je pak zobrazeno vlastní pravidlo v IF THEN podobě (tento zápis se mění v závislosti na změnách vykonávaných pomocí ostatních ovládacích prvků okna).

Při vytváření pravidla je potřeba si rozmyslet jakým způsobem budeme značit jednotlivé pravidla (numericky, za pomoci zkratk a čísel, či zvláštními názvy objasňující vlastní vazbu). Do kolonky ID, v levé části okna pod šedým řádkem, vyplníme tedy označení pravidla. Dále bychom ještě mohli nastavit prioritu pravidla případně kontext. Také můžeme vyplnit kolonku Komentáře, do které (pokud je projekt rozsáhlejší či pravidlo složitější) vložíme nějaký blíže vysvětlující komentář.

Obrázek 41 zachycuje pravidlo reprezentované čtyřmi vazbami sbíhajícími se v atributu BC2100 (viz Obrázek 37).

Při zadávání předpokladové části pravidla budeme využívat tři části okna „Pravidla“ (zvýrazněná červenou, zelenou a modrou barvou viz opět Obrázek 41, na tyto barevně odlišené části oken se budeme v následujících odstavcích odvolávat). V předpokladové části může být pouze jeden předpoklad, logická konjunkce dvou či více předpokladů nebo logická disjunkce dvou či více předpokladů či logických konjunkcí.

Při vytváření předpokladu nejprve musíme kurzorem označit (tak, aby bylo podbarveno modře) slovo „předpoklad“ v červeném rámečku. Po té můžeme ze zeleného boxu vybrat konkrétní předpokladový výrok (předpoklad) a po stlačení první ikony z modrého rámečku (připomínající klávesu enter) vložíme tento do předpokladu pravidla. Zopakováním tohoto postupu vytvoříme předpoklad s disjunkcí dvou předpokladů.

Chceme-li vytvořit konjunkci, nejprve opět označíme slovo „předpoklad“ a poté stiskneme druhé tlačítko se shora v modrém rámečku s ikonou „&“. Výroky pro konjunkci pak vkládáme stejným způsobem jako v předešlém odstavci, jen místo slova „předpoklad“ musíme mít označený znak „&“.

Chceme-li mít v předpokladové části obsaženou negaci nějakého výroku, tento výrok, pokud už jsme je vložili a vidíme ho v červeně označené části okna, označíme a zatrhneme políčko „Negace“. Před výrokem v červeně ohraničené části okna se objeví znaménko záporu a v šedém řádku NOT.

Chceme-li některý z výroků (nebo konjunkci) z předpokladové části smazat, pak jej označíme a myší klikneme na poslední tlačítko v modře označené části s ikonou křížku.

V závěrové části pravidla může být opět jediný výrok nebo skupina výroků, jimž navíc přiřazujeme váhu jejich platnosti. Závěr(y) vybíráme ze sloupečku „Možné závěry“ a přesouváme je do sloupečku „Vybrané závěry“ pomocí tlačítka (tlačítek mezi oběma sloupci) mezi sloupci. Váhu pak po označení vlastního závěru vepíšeme vždy jako kladnou hodnotu boxu „Váha“. Má-li být výsledná váha záporná, zatrhneme volbu negace (viz černá část okna „Pravidla“).

## **5.5 Ladění báze**

Ladění báze znalostí je dalším stěžejním bodem při tvorbě fungujícího projektu. Pokud jste přesvědčeni, že ve vaší bázi jsou zachyceny téměř všechny nebo všechny výroky a atributy, a pokud jste již všechny výroky použili v alespoň jednom pravidle, můžete se zabývat laděním.

Pro odladování je potřeba mít připraveny tzv. vzorové odpovědi pro jednotlivé cílové výroky. V zásadě existuje pro každý cílový výrok vzorová konzultace, respektive soubor odpovědí, který by měl systém vyhodnotit s vysokou vahou pro daný cílový výrok (na rozdíl od ostatních cílových výroků). Je to vlastně jakýsi kontrolní soubor odpovědí – kontrolní ve smyslu kontroly smysluplnosti báze znalostí. Kontrolní či vzorové odpovědi vám poskytne expert nebo je musíte identifikovat sami na základě analýzy svého problému částečně už vyplývající z počáteční analýzy při tvorbě grafu.

Při odladování s úspěchem použijete v první řadě možnosti uložení odpovědi do externího souboru, abyste nemuseli po každé úpravě vah či přidání nového pravidla pracně procházet znova a znova konzultaci a odpovídat na všechny dotazy. Dále pak jistě využijete možnosti ukládat si výsledky konzultací do např. textového souboru, pomocí nichž pak můžete srovnávat, jak se která úprava projevila na výsledcích a fungování báze.

Je zřejmé, že není možné, aby člověk přehledně zachytil do grafu o třech cílových výrocih, čtyřech až šesti výrocih mezilehlých a deseti či více výrocih základních, což samo o sobě předpokládá několik desítek jednoduchých pravidel, přehledně veškeré vazby (jednoduchá pravidla, složitá pravidla s konjunkcí a pravidla s negativními váhami). Sestavení báze znalostí začíná po zadání atributů a výroků konstrukcí základních pravidel (většinou jednoduchého typu – v předpokladu i závěru s jedním pozitivním výrokem či pozitivní vahou). Takto vytvoříme základní kostru báze a zajistíme její jakousi základní funkčnost, ale pravděpodobně zjistíme, že systém dostatečně nerozlišuje mezi cílovými výroky – váhy cílových výroků jsou v malém rozpětí (všechny nebo některé z nich). Váhy se liší např. v setinách nebo několika málo desetínách, a to i v případě, že použijete kontrolní či vzorové odpovědi. Tento fakt nemusí a většinou ani neznamenaá fundamentální chybu v logice zpracování.

Nyní je potřeba se zamyslet nad složitějšími pravidly a zabývat se hodnotami vah v závěrech jednotlivých pravidel. Například pokud v grafu máme situaci, kdy do jednoho cílového výroku směřuje více vazeb z kupříkladu tří výroků mezilehlé roviny, můžeme zkonstruovat tři jednoduchá pravidla reprezentující každou z těchto vazeb (nebo jedno pravidlo s logickou disjunkcí pokud v závěru chceme použít stejnou váhu) protože platnost, kteréhokoliv předpokladu bude mít vliv na platnost závěru. Nicméně většinou z praxe platí jistý synergický efekt, kdy výsledné působení jednotlivých faktorů se nerovná jejich prostému součtu, ale je o něco vyšší. Tento jev můžeme zachytit pomocí jednoho či více pravidel, kdy do předpokladu dáme konjunkci dvou či více výroků a v závěru ohodnotíme výrok vyšší vahou.

Dalším krokem je zamyšlení se nad použitím negací (jestliže jedním pravidlem definujeme, že když platí předpoklad, pak platí s jistou vahou závěr, neznamenaá to automaticky, že systém ví, jaký platí vztah v případě neplatnosti toho předpokladu).

Konstrukce pravidel s negací v předpokladu – zamyšlení, zda-li neexistuje také znalost popisující situaci, kdy víme, že některý z výroků je nepravdivý nebo konzultující jeho platnost ohodnotil zápornou vahou.

Konstrukce pravidel s negací v závěru – zamyšlení se, zda-li platnost či neplatnost některého výroku (např. základního) nemá negativní vliv na platnost hierarchicky výše položeného výroku (např. v mezilehlé rovině).

Poslední možností je pak bedlivě zanalyzovat výši vah v jednotlivých výrocih.

V průběhu ladění je také dobré si občas zkontrolovat graf báze, zda-li nevznikla v projektu nadbytečná – duplicitní pravidla. Zbytečný výrok figuruje v jediném pravidle v předpokladové části a zároveň k výroku v závěru pravidla nevede žádné další pravidlo, takový to výrok můžeme zrušit a platnost výroku ze závěru nadbytečného pravidla nyní zadá uživatel při konzultaci.

System sám si hlídá, aby nějaký výrok neobjevil v předpokladu pravidla a později v navazujícím pravidle v závěru. Rekurze či cyklus v bázi znalostí není možná.

V tuto chvíli jsme si prošli téměř veškeré možnosti systému NEST a NEST editor, které využijete nebo můžete využít při tvorbě svého vlastního projektu. Také jste byli upozorněni na některé postupy při vlastní tvorbě a chybí nám ještě upozornit na často opomíjenou, ale někdy velice praktickou část každé práce, a to je tvorba dokumentace.

## **5.6 Tvorba dokumentace**

Tvorba dokumentace je často opomíjenou a ne příliš populární stránkou při zpracovávání projektů, a to nejen na půdě školy, ale leckdy i v podnikové praxi. Je potřeba si však uvědomit, že dobře provedená dokumentace může zásadním způsobem ovlivnit zaškolení nových pracovníků do problému, přechod či zapojení dalších nových pracovníků do řešitelského týmu, nemluvě o situaci, kdy z nějakého důvodu jediný či hlavní řešitel práci opustí a my projekt musíme dotáhnout do cíle nebo později upravovat. V případě úprav projektu po delší době dokumentace dobře poslouží i samotnému autorovi pro rychlejší opětovné proniknutí do problému.

V dokumentaci je potřeba vždy představit a vymezit problém a zhodnotit jeho řešení, případně uvést možné slabiny systému a doporučení dalšího rozvoje.

Zůstaneme-li na naší akademické půdě, je tedy nutno objasnit problémovou oblast, typ řešeného problému, uvést, co je cílem projektu, jaké jsou cílové výroky. Charakterizujeme obecné vlastnosti cílových výroků mající vliv na průběh konzultace. Dále uvedeme globální parametry řešení: tedy zdroj znalostí, řešitelský tým atd. Metody získávání znalostí v řešené oblasti můžeme dále rozvést, aby bylo možno posoudit jejich správnost, opodstatněnost a relevantnost. Také je vhodné dokumentovat řešení pomocí grafu báze znalostí. V profesionálně tvářící se dokumentaci bychom jistě uvedli výčet všech atributů i výroků s komentářem, aby byl jasný jejich smysl a použití. Také bychom uvedli výčet pravidel a okomentovali bychom mechanismus či způsob stanovení vah.

V dokumentaci bychom měli, zdůraznit k jakému účelu je projekt nebo řešení vhodné a upozornit na limity řešení, případně na možnosti dalšího rozvoje.

Nástroje systému NEST využitelné pro tvorbu dokumentace:

- Statistika – NEST (Báze/Statistika).

- Graf – NEST i NEST editor.
- Globální parametry, atributy, výroky, pravidla a komentáře – NEST editor. (Soubor/Export báze).

## **SHRNUTÍ KAPITOLY**



T této kapitole jsme si ukázali, jak postupovat při samotném uchopení problému a zorientování se, nalezení atributů, započítání vlastní práce v NEST editoru a tvorby pravidel. Dále jsme se věnovali možnostem při ladění báze znalostí a možnostem výstupu použitelných pro dokumentační účely.

---

## **POZNÁMKY**



## LITERATURA

- [1] Berka, P., Jirků, P., Vejnarová, J. *Expertní systémy*. 1. vyd. Praha, VŠE 1998. 160 s.
- [2] Berka, P. *Vybrané znalostní systémy SAK, SAZE, KEX*. Skripta VŠE, 2000.
- [3] Berka, P. *Dobývání znalostí z databází*. 1. vyd. Praha, Academia, 2003. ISBN 80-200-1062-9.
- [4] Berka, P., Laš, V., Svátek, V. Nový expertní systém NEST. Sborník konference *Znalosti 2004*, Ostrava 2004, s. 256–267. ISBN 80-248-0456-5.
- [5] Berka, P., Laš, V., TOMEČKOVÁ, Marie. AtherEx: An Expert System for Atherosclerosis Risk Assessment. Aberdeen 23.07.2005 – 25.07.2005. In: MIKSCH, Silvia, HUNTER, Jim, KERAVALOU, Elpida (ed.). *Artificial Intelligence in Medicine*. Berlin : Springer-Verlag, 2005, s. 79–88. ISBN 3-540-27831-1.
- [6] Clarke, B., Fokoué, E., Zhang, H. H. *Principles and theory for data mining and machine learning*. Springer, 2009.
- [7] Dvořák, J. *Expertní systémy*. Skripta. VUT Brno. 2004
- [8] Ferjenčík, J., Ivánek, J., Švenda, J. Stručná informace o systému automatických konzultací SAK. In: *Informatika a umělá inteligence*. Bratislava, DT ČSVTS 1985.
- [9] Giarratano, J., Riley, G. *Expert Systems – Principles and Programming*. PWS Publishing Company, Boston 1993. ISBN: 0-534-95053-1.
- [10] Hájek, P. *Combining Functions for Certainty Factors in Consulting Systems*. Int. J. Man-Machine Studies 22, 1985
- [11] Ivánek, J., Ferjenčík, J., Švenda, J. *Znalostní inženýrství - východiska a perspektivy*. SIVO 2092. Praha, ÚVTEI/ÚTEIN 1986, 74s.
- [12] Ivánek, J. *Ziskávání znalostí - určující faktor rozvoje informačních technologií*. SIVO 2175. Praha, ÚVTEI/ÚTEIN 1988, 102s.
- [13] Ivánek, J., Ferjenčík, J., Stejskal, B., Švenda, J. *Základy matematické informatiky - III. Znalostní inženýrství*. Praha, Skripta VŠE 1987, 114s.
- [14] Ivánek, J.: *An expert system recommending suitable mathematical decision method*. Computers and Artificial Intelligence, 5, 1986, s.241-251.
- [15] Ivánek, J., Švenda, J., Ferjenčík, J. *Inference in Expert Systems Based on Complete Multi-valued Logic*. Suppl. Journal Kybernetika, Vol. 25, 1989, p.25-32.
- [16] Ivánek, J. *Representation of expert knowledge as a fuzzy axiomatic theory*. International Journal of General Systems, 20, 1991, s. 55-58.
- [17] Ivánek, J., Stejskal, B.: Automatic Acquisition of Knowledge Base from Data without Expert: ESOD (Expert System from Observational Data). In: *Proc. COMPSTAT'88 Copenhagen*. Physica-Verlag Heidelberg 1988, s.175-180
- [18] Ivánek, J. Realizace konzultačního systému SAK-OPTIMALI pro výběr matematické metody rozhodování. In: *Expertní systémy – principy, realizace, využití*. Praha, ČSVTS FEL ČVUT 1984, s.160-172.
- [19] Ivánek, J., Kempný, R., Laš, V. *Znalostní inženýrství*. Skripta. OPF SU. 2007.

Klikněte sem a zadejte text.

[20] Jiroušek, R. *Metody reprezentace a zpracování znalostí v umělé inteligenci*. 1. vyd. Skripta VŠE, 1995. 103 s.

## SHRNUTÍ STUDIJNÍ OPORY

Cílem tohoto studijního textu bylo seznámit čtenáře se základy oboru věnujícího se expertním systémům, který se poprvé objevil především v souvislosti se širším vědním oborem Umělá inteligence. V souvislosti s pojmem expertní systém se dále čtenář seznámil s možnostmi návrhu a tvorby expertního systému v rámci disciplíny nazývané Znalostní inženýrství, která se věnuje možnostem získávání, reprezentace a využití znalostí. Dále byl v textu představen systém NEST, který umožňuje vytvářet určitý typ expertních systémů a v rámci tohoto nástroje byla taktéž nastíněna práce s neurčitostí, která je nedílnou součástí většiny expertních systémů.

Klikněte sem a zadejte text.

# **PŘÍLOHA Č. 1 VZOROVÁ SEMINÁRNÍ PRÁCE**

**Slezská univerzita v Opavě**

**Obchodně podnikatelská fakulta v Karviné**



Expertní systémy  
**ES pro výběr ubytování**

SEMINÁRNÍ PRÁCE

Bc. Michael Kubát  
ZS 2013/2014

Klikněte sem a zadejte text.

## ÚVOD

Ve své seminární práci do předmětu Expertní systémy bych rád zpracoval ES, který bude doporučovat především budoucím studentům OPF, zda si zvolit koleje Vyhlídka, ubytovnu Kosmos nebo privát ve vybraných částech Karviné, na základě např. způsobu dopravy, umístění vyučování, sportovního nebo kulturního vyžití apod.

Toto téma jsem si zvolil, protože považuji za užitečné, vytvořit nástroj, který by pomohl vybrat ubytování pro studenty, kteří pochází ze vzdálenějších míst a o Karviné mnoho nevědí.

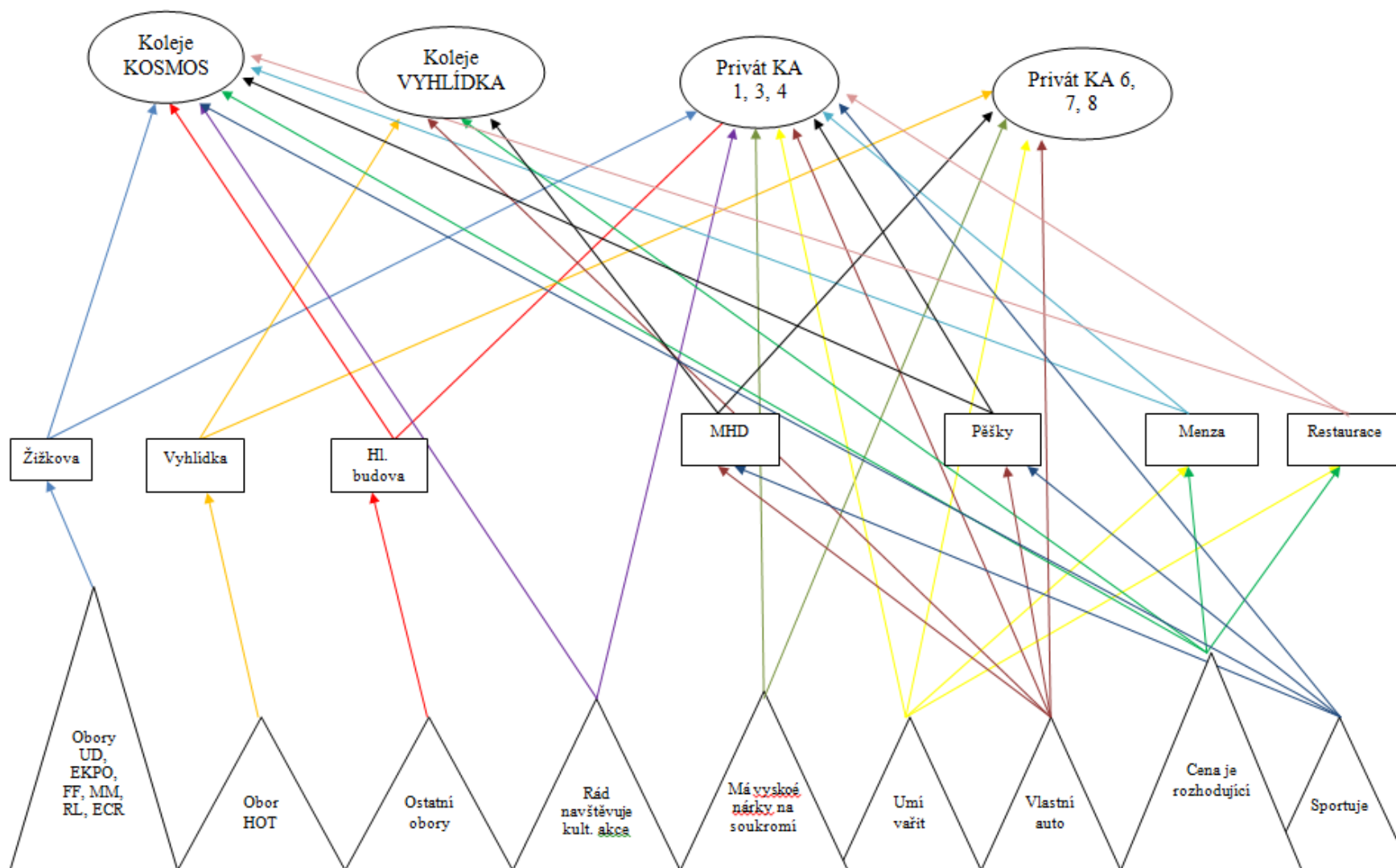
## 1 NÁVRH BÁZE

V následující tabulce jsem vypsali vlastnosti jednotlivých diagnóz. Podle nich potom budu stanovovat váhy jednotlivých pravidel.

**Tabulka 1: Vlastnosti diagnóz (jednotlivých možností ubytování v Karviné)**

| Typ                | Koleje Vyhlídka         | Koleje Kosmos                             | Privát Karviná 1, 3, 4  | Privát Karviná 6, 7, 8               |
|--------------------|-------------------------|---|---|--------------------------------------|
| <b>Cena</b>        | 2500,-/měsíc            | 2200,- /měsíc                             | Záleží na počtu obyvatel  | Záleží na počtu obyvatel             |
| <b>Vzdálenosti</b> | Součást budovy Vyhlídka | Blízko hl. budově a Žižkova               | Blízko hl. budově   | Blízko Vyhlídce, případně Žižkova    |
| <b>Vyžití</b>      | Blízko Inso-mnie        | Blízko bazén, kurty, fotbal. hřiště, kino | Blízko náměstí, fotbal. hřiště, park, tenisové kurty, zimní stadion | Blízko Inso-mnie, bazén, hala házené |
| <b>Doprava</b>     | BUS, možnost parkování  | Hl. budova i Žižkova pěšky                | Pěšky, možnost parkování  | Pěšky, možnost parkování             |

Na další straně pak následuje diagram návrhu báze.

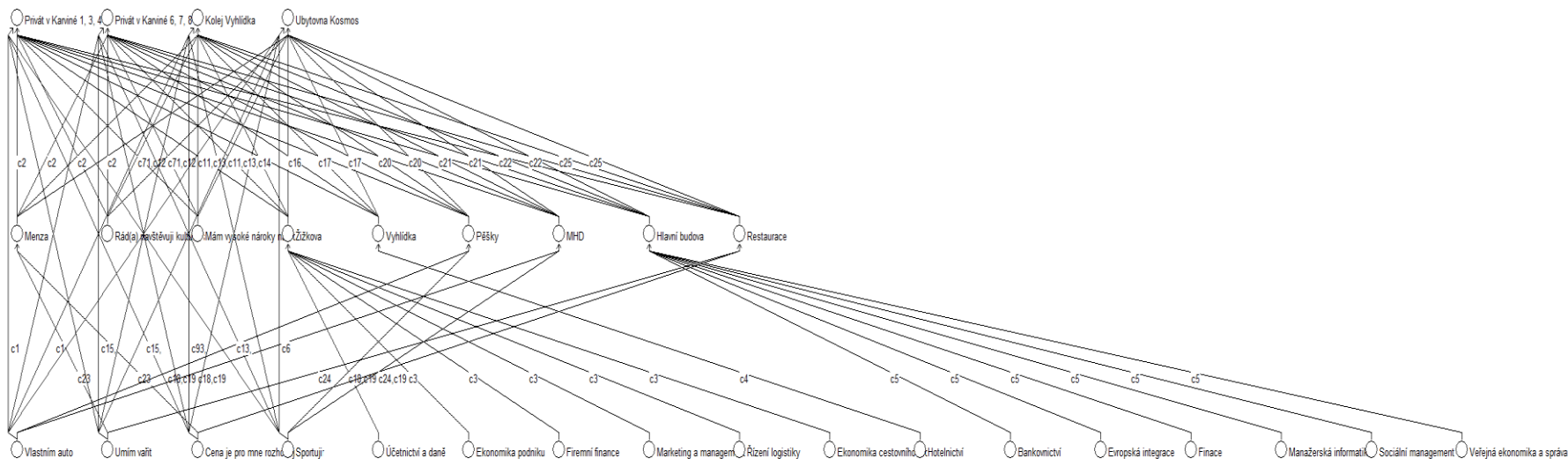


Obrázek 1: Diagram návrhu báze (koncept báze znalostí)

Klikněte sem a zadejte text.

## 2 STATISTIKY

Na Obrázku 1 je graf báze, který popisuje reálné vazby mezi dotazy, mezilehlými výroky a diagnózami, které jsou dány jednotlivými pravidly. Oproti návrhu báze jsou zde zobrazeny i pravidla s negací a také pravidla, která obsahují v předpokladu konjunkci.



**Obrázek 3: Graf báze (generováno programem NEST)**



Obrázek 4 pak shrnuje statistiky báze.

| Globální vlastnosti                 |   | Atributy        |    | Výroky             |    |
|-------------------------------------|---|-----------------|----|--------------------|----|
| JmenoLabel                          | Baze_Kubat.xml  | Celkem          | 15 | Celkem             | 30 |
| Popis báze                          | ES, který doporučuje studentům, zda si zvolit kolej Vyhlička, Kosmos nebo privát v určité části Karviné, na základě např. způsobu dopravy, umístění vyučování, sportovního nebo | Binární         | 13 | Binární            | 13 |
| Expert                              | Michael Kubát   | Jednoduchý      | 1  | Jednoduchý         | 13 |
| Znalostní inženýr                   | Michael Kubát   | Množinový       | 1  | Množinový          | 4  |
| Datum                               | 6.11.2013   | Numerický       | 0  | Numerický          | 0  |
| Rozsah vah                          | 3   | dotaz           | 7  | dotaz              | 19 |
| Globální práh platnosti kontextu    | 0,000   | mezilehlý       | 7  | mezilehlý          | 7  |
| Globální práh platnosti předpokladu | 0,767   | cíl             | 1  | cíl                | 4  |
| Inferenční mechanismus              | Standardní  | osamocený       | 0  | osamocený          | 0  |
| Implicitní váha                     | Irelevantní   | Zdroje          | 0  | Akce               | 0  |
| Globální priorita                   | Od začátku  | Akce            | 0  |                    |    |
|                                     |   | Pravidla        |    | Další              |    |
|                                     |   | Celkem          | 25 | Kontexty           | 0  |
|                                     |   | Apriorní        | 0  | Integritní omezení | 0  |
|                                     |   | Logické         | 0  |                    |    |
|                                     |   | Kompozicionální | 25 |                    |    |
|                                     |   | Akce            | 0  |                    |    |

Obrázek 4: Statisticky (generováno programem NEST)

### 3 PŘÍKLADY DIAGNÓZ

Po prvním návrhu báze znalostí v systému NEST jsem začal testovat kvalitu vytvořené báze. Pro každou ze čtyř diagnóz jsem si vytvořil tzv. typického uživatele diagnózy, což je vlastně nějaká konkrétní sada odpovědí na dotazy v bázi, pro kterého by měla vyjít zvolená diagnóza s nejvyšší vahou, a váhy všech ostatních diagnóz by měly vyjít nižší. Samozřejmě, že takto mi to napoprvé nevyšlo a bylo potřeba několikrát bázi přepracovat, a to hlavně upravit váhy jednotlivých pravidel, ale taky přidat pravidla s negacemi v předpokladu i závěru a také občas přidat konjunkci do předpokladu některých pravidel.

#### 3.1 Diagnóza Privát Karviná 1, 3, 4

Jedná se o diagnózu, při které systém doporučí vybrat privát v Karviné 1 (Fryštát), Karviné 2 (Ráj) nebo Karviné 4 (Darkov).

##### 3.1.1 TYPICKÝ UŽIVATEL

Podle mých znalostí a zkušeností bych toto bydlení doporučil studentovi, který je:

- Budoucí student Manažerské informatiky
- Vlastníci automobil
- Sportující
- Rád navštěvující kulturní akce
- Který umí vařit
- Mající vysoké nároky na soukromí

- Cena je pro něj rozhodující faktorem
- Vymyslel jsem si tedy, jak by takový student zřejmě na dotazy dané mým expertním systémem odpovídal (viz spodní část Obrázku 5).

| Výroky                   |            |          |         |      |
|--------------------------|------------|----------|---------|------|
| Jméno                    | Min váha < | Max váha | Stav    | Druh |
| Privát v Karviné 1, 3, 4 | 2,990      | 2,990    | finální | cíl  |
| Privát v Karviné 6, 7, 8 | 2,849      | 2,849    | finální | cíl  |
| Ubytovna Kosmos          | 1,227      | 1,227    | finální | cíl  |
| Kolej Vyhlička           | -2,419     | -2,419   | finální | cíl  |

| Atributy |                      |   |                               |
|----------|----------------------|---|-------------------------------|
| +        | Jaký obor chcete st  | s | Manažerská informatika: 3,000 |
| +        | Vlastním auto        | b | 3,000                         |
| +        | Sportuji             | b | 3,000                         |
| +        | Rád(a) navštěvuji kt | b | 3,000                         |
| +        | Umím vařit           | b | 3,000                         |
| +        | Mám vysoké nároky    | b | 3,000                         |
| +        | Cena je pro mne roz  | b | 3,000                         |

**Obrázek 5: Nahoře jsou výsledné váhy pro typického uživatele Diagnózy 1 a níže jsou odpovědi tohoto typického uživatele (generováno programem NEST)**

Pomocí finální verze mého ES jsem pro tohoto studenta dostal váhu 2,99 u diagnózy Privát v Karviné 1, 3, 4 a pro ostatní diagnózy jsem dostal nižší váhy (jak je vidět na Obrázku 5), což mi potvrdilo, že pro typického uživatele této diagnózy mám bázi nastavenou správně. Je také vidět vysoká váha u diagnózy Privát v Karviné 6, 7, 8. Toto však odpovídá skutečnosti, že tyto diagnózy jsou si velmi podobné a pro studenta, pro kterého je vhodné doporučit Privát v Karviné 1, 3, 4 je také často možné doporučit i Privát v Karviné 6, 7, 8. Pro ostatní diagnózy už je váha výrazně nižší, což odpovídá výraznému rozdílu mezi těmito typy bydlení (ubytovna, kolej) a bydlením na privátu v Karviné 1, 3, 4.

## 3.2 Diagnóza Privát Karviná 6, 7, 8

Tato diagnóza doporučuje vybrat privát v Karviné 6 (Nové Město), Karviné 7 (Mizerov) nebo Karviné 8 (Hranice).

### 3.2.1 TYPICKÝ UŽIVATEL

Podle mých znalostí a zkušeností bych toto bydlení doporučil studentovi, který je:

- Budoucí student Hotelnictví
- Nevlastníci automobil
- Nesportující
- Který nerad navštěvuje kulturní akce
- Který neumí vařit
- Mající vysoké nároky na soukromí
- Cena je pro něj irelevantní

Vymyslel jsem si opět, jak by takový student zřejmě na dotazy dané mým expertním systémem odpovídal (viz spodní část Obrázku 6).

| Výroky                   |            |          |         |      |
|--------------------------|------------|----------|---------|------|
| Jméno                    | Min váha < | Max váha | Stav    | Druh |
| Privát v Karviné 6, 7, 8 | 2,857      | 2,857    | finální | cíl  |
| Privát v Karviné 1, 3, 4 | 2,708      | 2,708    | finální | cíl  |
| Kolej Vyhlídka           | -1,003     | -1,003   | finální | cíl  |
| Ubytovna Kosmos          | -2,399     | -2,399   | finální | cíl  |

| Atributy |                                       |   |                    |
|----------|---------------------------------------|---|--------------------|
| +        | Jaký obor chcete studovat             | s | Hotelnictví: 3,000 |
| +        | Vlastním auto                         | b | -3,000             |
| +        | Sportuji                              | b | -3,000             |
| +        | Rád(a) navštěvuji kulturní akce       | b | -3,000             |
| +        | Umím vařit                            | b | -3,000             |
| +        | Mám vysoké nároky na soukromí         | b | 3,000              |
| +        | Cena je pro mne rozhodujícím faktorem | b | irrelevant         |

**Obrázek 6: Nahoře jsou výsledné váhy pro typického uživatele Diagnózy 2 a níže jsou odpovědi tohoto typického uživatele (generováno programem NEST)**

Pomocí mého ES jsem pro tohoto studenta dostal váhu 2,857 u diagnózy Privát v Karviné 6, 7, 8 a pro ostatní diagnózy jsem dostal nižší váhy (jak je vidět na Obrázku 6), což mi opět potvrdilo, že pro typického uživatele této diagnózy mám bázi nastavenou správně. Je také opět vidět vysoká váha u diagnózy Privát v Karviné 1, 3, 4. Toto zase odpovídá skutečnosti, že tyto diagnózy jsou si velmi podobné a pro studenta, pro kterého je vhodné doporučit Privát v Karviné 6, 7, 8 je také možno doporučit i Privát v Karviné 1, 3, 4. Pro ostatní diagnózy už je váha výrazně nižší, což odpovídá výraznému rozdílu mezi těmito typy bydlení a bydlením na privátu v Karviné 6, 7, 8.

### 3.3 Diagnóza Kolej Vyhlídka

Tato diagnóza doporučuje vybrat kolej Vyhlídka v Karviné 6 (Nové Město).

#### 3.3.1 TYPICKÝ UŽIVATEL

Podle mých znalostí a zkušeností bych toto bydlení doporučil studentovi, který je:

- Budoucí student Hotelnictví
- Vlastní automobil
- Nesportující
- Který nerad navštěvuje kulturní akce
- Který umí vařit
- Nemající vysoké nároky na soukromí
- Cena je pro něj rozhodujícím faktorem

Vymyslel jsem si opět, jak by takový student zřejmě na dotazy dané mým expertním systémem odpovídal (viz spodní část Obrázku 7).

| Výroky                   |            |          |         |      |
|--------------------------|------------|----------|---------|------|
| Jméno                    | Min váha < | Max váha | Stav    | Druh |
| Kolej Vyhlídka           | 2,997      | 2,997    | finální | cíl  |
| Ubytovna Kosmos          | 2,962      | 2,962    | finální | cíl  |
| Privát v Karviné 6, 7, 8 | 1,424      | 1,424    | finální | cíl  |
| Privát v Karviné 1, 3, 4 | 1,375      | 1,375    | finální | cíl  |

| Atributy                               |   |                    |
|--|---|--------------------|
| + Jaký obor chcete studovat            | s | Hotelnictví: 3,000 |
| + Vlastním autem                       | b | 3,000              |
| + Sportuji                             | b | -3,000             |
| + Rád(a) navštěvuji kulturní akce      | b | -3,000             |
| + Umím vařit                           | b | 3,000              |
| + Mám vysoké nároky na soukromí        | b | -3,000             |
| + Cena je pro mě rozhodujícím faktorem | b | 3,000              |

**Obrázek 7: Nahore jsou výsledné váhy pro typického uživatele Diagnózy 3 a níže jsou odpovědi tohoto typického uživatele (generováno programem NEST)**

Pomocí mého ES jsem pro tohoto studenta dostal váhu 2,997 u diagnózy Kolej Vyhlídka a pro ostatní diagnózy jsem dostal nižší váhy (jak je vidět na Obrázku 7), což mi opět potvrdilo, že pro typického uživatele této diagnózy mám bázi nastavenou správně. Podobně jako u předchozích dvou diagnóz lze vidět podobnost této diagnózy s diagnózou Ubytovna Kosmos. Pro ostatní diagnózy už je váha výrazně nižší, což odpovídá skutečnému rozdílu mezi bydlením na koleji a na privátu.

### 3.4 Diagnóza Ubytovna Kosmos

Tato diagnóza doporučuje vybrat kolej Kosmos v Karviné 7 (Mizerov).

#### 3.4.1 TYPICKÝ UŽIVATEL

Podle mých znalostí a zkušeností bych toto bydlení doporučil studentovi, který je:

- Budoucího studenta Manažerské informatiky
- Nevlastníceho automobil
- Sportujícího
- Který nerad navštěvuje kulturní akce
- Který neumí vařit
- Nemajícího vysoké nároky na soukromí
- Cena je pro něj rozhodujícím faktorem

Vymyslel jsem si opět, jak by takový student zřejmě na dotazy dané mým expertním systémem odpovídal (viz spodní část Obrázku 8).

| Výroky                   |            |          |         |      |
|--------------------------|------------|----------|---------|------|
| Jméno                    | Min váha < | Max váha | Stav    | Druh |
| Ubytovna Kosmos          | 2,995      | 2,995    | finální | cíl  |
| Kolej Vyhlička           | 2,987      | 2,987    | finální | cíl  |
| Privát v Karviné 1, 3, 4 | -0,413     | -0,413   | finální | cíl  |
| Privát v Karviné 6, 7, 8 | -0,956     | -0,956   | finální | cíl  |

| Atributy                          |   |                               |
|-----------------------------------|---|-------------------------------|
| + Jaký obor chcete studovat       | s | Manažerská informatika: 3,000 |
| + Vlastním auto                   | b | -3,000                        |
| + Sportuji                        | b | 3,000                         |
| + Rád(a) navštěvuji kulturní akce | b | -3,000                        |
| + Umím vařit                      | b | -3,000                        |
| + Mám vysoké nároky na bydlení    | b | -3,000                        |
| + Cena je pro mne rozhodující     | b | 3,000                         |

**Obrázek 8: Nahoře jsou výsledné váhy pro typického uživatele Diagnózy 4 a níže jsou odpovědi tohoto typického uživatele (generováno programem NEST)**























Pomocí mého ES jsem pro tohoto studenta dostal váhu 2,995 u diagnózy Ubytovna Kosmos a pro ostatní diagnózy jsem dostal nižší váhy (jak je vidět na Obrázku 8), což mi opět potvrdilo, že pro typického uživatele této diagnózy mám bázi nastavenou správně. Opět se ukázala podobnost této diagnózy s diagnózou Kolej Vyhlička. Pro ostatní diagnózy už je váha výrazně nižší, což odpovídá skutečnému rozdílu mezi bydlením na ubytovně a na privátu.

## 4 ZÁVĚR

Původní záměr, tedy vytvořit ES, který bude doporučovat především budoucím studentům OPF, zda si zvolit kolej Vyhlička, ubytovnu Kosmos nebo privát ve vybraných částech Karviné, na základě např. způsobu dopravy, umístění vyučování, sportovního nebo kulturního vyžití apod. jsem splnil. Správnost nastavení báze jsem ověřil pomocí čtyř typických uživatelů (pro každou diagnózu jeden), kdy pro každého typického uživatele vyšla s nejvyšší vahou odpovídající diagnóza a všechny zbylé diagnózy vyšly s vahou nižší.

## **POZNÁMKY**

## PŘEHLED DOSTUPNÝCH IKON

|   |                        |   |                          |
|---|------------------------|---|--------------------------|
|    | Čas potřebný ke studiu |    | Cíle kapitoly            |
|    | Klíčová slova          |    | Nezapomeňte na odpočinek |
|    | Průvodce studiem       |    | Průvodce textem          |
|    | Rychlý náhled          |    | Shrnutí                  |
|    | Tutoriály              |    | Definice                 |
|   | K zapamatování         |   | Případová studie         |
|  | Řešená úloha           |  | Věta                     |
|  | Kontrolní otázka       |  | Korespondenční úkol      |
|  | Odpovědi               |  | Otázky                   |
|  | Samostatný úkol        |  | Další zdroje             |
|  | Pro zájemce            |  | Úkol k zamyšlení         |

Pozn. Tuto část dokumentu nedoporučujeme upravovat, aby byla zachována správná funkčnost vložených maker. Tento poslední oddíl může být zamknut v MS Word 2010 prostřednictvím menu Revize/Omezit úpravy.

Takto je rovněž omezena možnost měnit například styly v dokumentu. Pro jejich úpravu nebo přidávání či odebrání je opět nutné omezení úprav zrušit. Zámek není chráněn heslem.





Název:           Expertní systémy

Autor:           **Jan Górecki**

Vydavatel:       Slezská univerzita v Opavě  
                    Obchodně podnikatelská fakulta v Karviné

Určeno:          studentům SU OPF Karviná

Počet stran:     121

Tato publikace neprošla jazykovou úpravou.