Data, znalosti, reprezentace znalostí: produkční systémy, rámce a scénaře, sémantické sítě a predikátová logika. Metody využívání znalostí. Znalostní a expertní systémy, typy, základní architektury, princip jejich činnosti a návrhu. Inferenční síť.

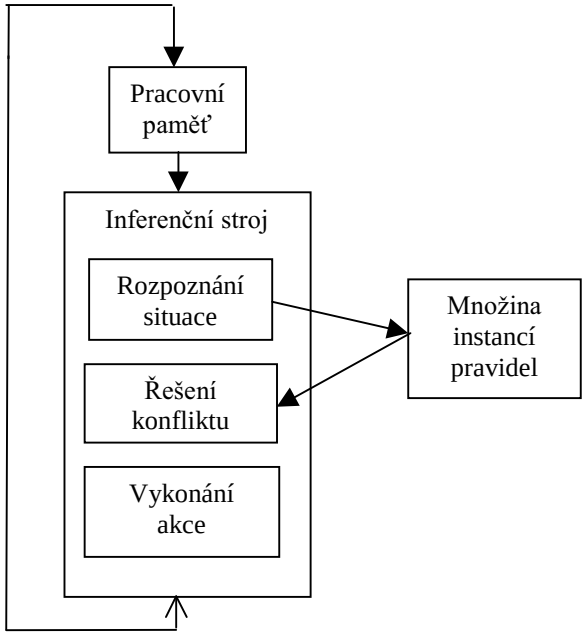
Data - elementární (do jisté míry), ověřitelná fakta. Jedná se o libovolnou representaci faktografických údajů. Data odrážejí současný stav světa na úrovni instancí. Data jsou velice objemná a často se mění, např. hodnoty získané pomocí senzorů.

Informace – informaci chápeme jako vhodně interpretovaná data, např. zpracování senzorických dat za účelem vytvoření modelu světa.

Znalosti – relace mezi jednotlivými množiny dat, která slouží zpravidla pro vyvozování dalších informací. Mají obecnější charakter než data. Obsahují informace o chování abstraktních modelů světa (representovaných daty), např. co informace získaná ze senzorických dat představuje.

Reprezentace znalostí – znalosti je třeba nějakým způsobem reprezentovat, proto následuje jednoduchý popis nejpoužívanějších a základních typů reprezentace.

Produkční systémy – je tvořen třemi částmi, viz. obr. č.1.



Obrázek č.1

1. *Soubor produkčních pravidel* tvaru *Situace → Akce*. Situační část pravidla popisuje podmínku, za které ,může být pravidlo použito. Tato pravidla jsou uložena v paměti produkčních pravidel.
2. *Pracovní paměť (báze dat)*,ve které jsou uložena jak počáteční data, tak i data odvozená. Tato část paměti popisuje okamžitý stav řešené úlohy.
3. *Inferenční stroj* porovnává data v pracovní paměti s produkčními pravidly, vybírá pravidla vhodná k vykonání a provede příslušnou akci.

Produkční systémy pracují obvykle v cyklu *Rozpoznávání situace → Vykonání akce*. Krok rozpoznání situace porovnává levou stranu pravidel s aktuálním stavem báze dat a jsou vybrána všechna aplikovatelná pravidla. Mají-li pravidla proměnné ⇒ *instance pravidel*. Volba pravidla, které je provedeno ⇒ *řešení konfliktu* :

1. Výběr pravidla, které kombinuje větší počet podmínek kladených na pracovní paměť.
2. Vyloučení pravidel, která byla v předchozím kroku aplikována na stejná data.
3. Preference novějších dat, nová dat ⇒ jejich zpracování je důležitější.

Aplikace pravidla ⇒ modifikace pracovní paměti ⇒ nové podmínky pro první krok dalšího cyklu. Tento způsob práce produkčního systému se nazývá *přímé řetězení*. Některé inferenční stroje ⇒ vyhledávají pravidla v opačném směru ⇒ inference je spuštěna cílovým stavem ⇒ *zpětné řetězení*.

Sémantické sítě - znalosti jsou reprezentovány pomocí objektů (entit) binárních či víceargumentových relací mezi nimi. Víceargumentové relace je možné vždy rozložit na množinu binárních. Sémantické sítě mají přirozenou grafovou reprezentaci, každý uzel grafu odpovídá jistému objektu a každá hrana odpovídá binární relaci. Lze výhodně vyjadřovat vztahy množinové inkluze a příslušnosti k množině. Rozlišuje se mezi typem (auto Škoda favorit) a instancí (např. můj favorit, SPY ADU 93 37), viz. obr. č.2. Vztahy množinové inkluze a příslušnosti ke třídě umožňují efektivně reprezentovat taxonomické (hierarchické) uspořádání objektů ⇒ *specializace a generalizace*. Specializace – informace od obecnějšího typu ke speciálnějšímu, generalizace naopak. Relace (viz. obr. č.2 konec dokumentu):

1. *Is – a* – definuje relaci specializace.
2. *Has – a* – přiřazuje vlastnosti.

Objekty vzniklé specializací mohou dědit vlastnosti od objektů obecnější úrovně ⇒ rychlé vybavování informace a úspora místa. Postup odvozování je naznačen na obr. č.3.

Škoda Favorit je osobní auto vozidlo.	Škoda Favorit má čtyřdobý benzínový motor karburátor Jirkov kola převodovku spalovací motor motorový agregát brzdy

Obrázek č.3

Uvažujme obr. č.4 (konec dokumentu). Postupně lze odvodit, že instance *můj favorit* má karburátor *Pierburg* a pomocí hierarchie (dědění) má karburátor *Jikov* ⇒ nelze reprezentovat *default* hodnoty ⇒ použití *rámců*.

Rámce – sestávají se z položek, které slouží k popisu jednotlivých vlastností (PASCAL – record, C – struct). V průběhu užívání nabývají tyto položky konkrétních hodnot. Jako příklad obr. č. 5.

JMÉNO RÁMCE: Škoda Favorit
Položky:
ČÍSLO RÁMCE: 1
IS-A: osobní auto
MOTOR: čtyřdobý benzínový
PŘEVODOVKA: manuální
KARBURÁTOR: Jikov
... atd.

Obrázek č.5

Popis položky se skládá z *jména* (např. motor) u z *hodnoty* (např. čtyřdobý benzínový). Jména položek odpovídají hranám v sémantické síti a hodnoty pak uzlům sítě. Položky dělíme na *fasety* ⇒ odstranění nekonzistenci při dědění (*default* hodnoty), např. u rámce, který bude popisovat *můj vůz Škoda*, nebude faseta *hodnota* položky *karburátor* zadána a faseta *předpokládaná hodnota* zdědí obsah fasety *předpokládaná hodnota* položky *karburátor* hierarchicky nadřazeného rámce *Škoda Favorit*. Případě potřeby se místo obsahu fasety *hodnota* použije obsah fasety *předpokládaná hodnota*. *Iff* chceme zaznamenat, že *můj vůz* má karburátor *Pierburg*, prostě uložíme tento údaj do fasety *hodnota* položky *karburátor*. Rozdílnost faset není na závadu. V případě přiřazení fasetě *hodnota* zpočátku ⇒ faseta *předpokládaná hodnota* by nejspíše zůstala nepřiřazena. Ke každé fasetě může být přidružen jeden nebo několik démonů ⇒ *spící* procedura, která se vyvolává při určité

události. Nejužívanější démoni : *když je třeba, když se vyplňuje, když se maže*, lze definovat i další ⇒ např. kontrola hodnot položek.

Scénáře – (skripty) vyjadřují standardní posloupnost akcí, složitější rámcové struktury k popisu příčinných nebo měnících se událostí. Kromě samotných datových položek obsahují informaci, jak s touto informací zacházet. Je zde možné shledat určitý vztah k *objektům*, avšak metody jsou vyjádřeny pomocí objektů a znalostí.

Predikátová logika. Následující fakta se týkají pouze využití reprezentace. Jazyk predikátové logiky 1.řádu je tvořen množnou konstant, proměnných, množinou funkčních symbolů, množnou predikátových symbolů a kvantifikátory. Konstanty chápeme jako funkční symboly o četnosti nula.logické spojky a kvantifikátory mají vždy stejné významy bez ohledu na použití jazyka. Funkční a predikátové symboly tedy můžeme chápat jako *slova* jazyka, ze kterého s využitím proměnných, logických spojek a kvantifikátorů tvoříme *věty* = formule jazyka. Zvolme pevně nějakou množinu χ formulí jazyka, kterou jsme získali pozorováním a jenž je bezesporná (neobsahuje fakt a současně jeho negaci). Jazyk predikátové logiky 1.řádu slouží k popisu objektů a relací mezi nimi.
Funkční a predikátové symboly sami o sobě nic nepředstavují ⇒ přiřadíme je objektům nějakého světa ⇒ určíme *interpretaci jazyka*. Interpretace *M* je definována množinou (objektů) *D*, přiřazením predikátových symbolů relacím nad *D* a přiřazením funkčních symbolů funkcím definovaných nad *D*. Množina *D* se nazývá *nosič* a obsahuje všechny objekty, o kterých může jazyk vypovídat. Mějme příklad: tři krychle barvy žluté, modré a zelené, žlutá na podložce, zelená na žluté, tedy relace *podpírá* je tvořena dvojicemi <žlutá, zelená>, <zelená, modrá>. Jazyk tvoří jediný predikátový symbol *P* četnosti 2 a třemi konstantami *a,b,c*. Nosič *D* relační struktury je tedy {modrá, zelená, žlutá}. Objekty a relace přiřadíme symbolům následovně: $a \rightarrow$ *žlutá,* $b \rightarrow$ *zelená,* $c \rightarrow$ *modrá,* $P \rightarrow$ *podpírá.*
K přesnější definici sémantiky formulí jazyka predikátové logiky 1.řádu potřebujeme pojem *ohodnocení* proměnných. Ohodnocení *e* proměnných, které se vyskytují ve formuli *alfa* je funkce zobrazující jednotlivé proměnné na objekty z *D*. Lze jej přirozeně rozšířit na zobrazení všech termů do *D*. Termy jazyka zobrazíme do nosiče *D* takto:

1. Proměnné formule *alfa* přiřazujeme prvkům nosiče *D* podle ohodnocení *e*.
2. Nechť *t* je term ve tvaru $t = f(t1, ...tn)$, kde *t1, .. tn* jsou termy, jejichž zobrazení *e(ti)* do *D* již známe, a nechť funkčnímu symbolu f je přiřazena interpretací *M* funkce ϕ na *D*, potom termu *t* přiřadíme hodnotu $e(t) = \phi(e(t1), ...e(tn))$.

Pravdivost formule *alfa* v interpretaci *M* při ohodnocení *e* určujeme touto rekurzivní definicí:

1. Je-li *alfa* atomická formule tvaru *p(t1, ..tn),* kde *p* predikátový symbol a *t1, ...tn* jsou termy, a je-li predikátovému symbolu *p* přiřazena interpretací *M* relace *p* na *D*, potom řekneme, že atomická formule *alfa* je pravdivá v interpretaci *M* při ohodnocení *e*, jestliže termy *t1, ...tn* přiřadíme objektům z *D* podle uvedených bodů ohodnocení (viz. výše) a jsou-li tyto objekty prvkem relace *p*.
2. Formule *alfa* vytvořená pomocí logických spojek má pravdivostní hodnotu danou sémantikou logických spojek a pravdivostními hodnotami dílčích formulí.
3. Formule tvaru $\forall x$ *(alfa)* je pravdivá při ohodnocení *e*, je-li formule *alfa* pravdivá pro všechna ohodnocení *ei*, které je identické s *e* s výjimkou hodnot přiřazených proměnné *x*.
4. Formule tvaru $\exists x$*(alfa)* je pravdivá při ohodnocení *e*, je-li formule *alfa* pravdivá alespoň pro jedno ohodnocení *ei*, která je identické s *e* s výjimkou hodnot přiřazených proměnné *x*.

Vraťme se nyní k výše uvedenému příkladu s kostkami. Ve výše uvedené interpretaci jsou formule *P(a,b) a P(b,c)* splněny, naproti tomu formule *P(c,a)* není splněna.

Říkáme, že formule *alfa* je splněna v interpretaci *M*, je-li tam pravdivá při libovolném ohodnocení *e*. Ohodnocení *e* je zjevně důležité jen pro volné proměnné, nemá-li formule volné proměnné, je její pravdivost na *e* nezávislá. Je-li formule *alfa* splněna v interpretaci *M*, píšeme $M \models$ *alfa*. Jestliže je formule splněna ve všech interpretacích říkáme, že je logicky pravdivá. Jestliže je formule *alfa* splněna ve všech interpretacích, ve kterých je splněna ve všech interpretacích, ve kterých je splněna formule *beta*, říkáme, že *alfa* logicky (je logickým důsledkem) z *beta* a píšeme *beta* $\models$ *alfa*.

Je-li *M* interpretací, ve které jsou splněny všechny formule (speciální axiomy) teorie χ, říkáme, že *M* je modelem teorie χ a píšeme $M \models \chi$. Teorie χ je splnitelná, pokud má model. Platí-li pro nějakou formuli *alfa* a všechny modely $M \models \chi$, že *alfa* je splněna v modelu *M*, tj. $M \models$ *alfa*, říkáme, že *alfa* logicky vyplývá z teorie χ a píšeme $\chi \models$ *alfa*.

Úplnost – iff $\chi \vdash$ *alfa*, potom $\chi \models$ *alfa* $\Rightarrow$ libovolná formule, kterou formálně dokážeme z teorie χ , z této teorie také logicky vyplývá, jestliže platí opak $\Rightarrow$ úplnost.

Expertní systém.-systém snažící se rozhodnout jako expert.Počítačové programy,simulující rozhodovací činnost experta při řešení složitých úloh a využívající vhodně zakódovaných, explicitně vyjádřených speciálních znalostí, převzatých od experta, s cílem dosáhnout ve zvolené problémové oblasti kvality rozhodování na úrovni.

Znalostní systém. – není odlišný od expertního systému, snad pouze bází znalostí, která není natolik expertní, proto v další výklad bude věnován oběma, ale přesto: softwarový systém, který podporuje proces lidského rozhodování. Filosofickým základem je imitace expertního rozhodování, t.j. vhodná representace rozhodovacích znalostí a rozhodovacího procesu. V zásadě se jedná o znalosti experta, můžeme ovšem vhodným způsobe, manipulovat i se znalosti získanými automatizovanou analýzou.

Upozornění! – následující text se nepokouší popsat matematiku s kterou expertní systémy pracují a to z důvodu pouhého seznámení s expertními systémy.

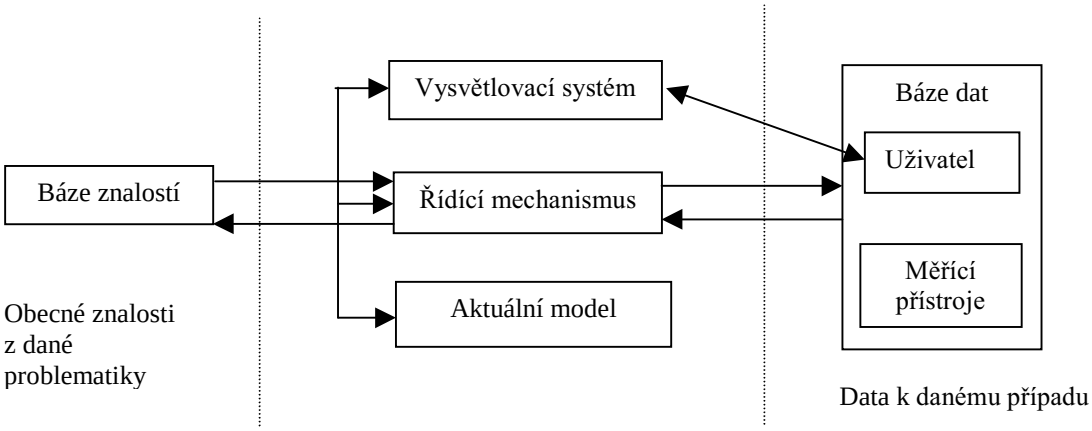
Charakteristické vlastnosti:

1. Oddělení báze znalostí od odvozovacího mechanismu, báze znalostí → systém pravidel → rámce, sémantické sítě.
2. Heuristické znalosti a jejich reprezentace → pomáhají řešit úlohy, experti si je většinou nedovolí publikovat.
3. Data k řešenému příkladu – báze dat, dodávána obvykle v dialogovém režimu.
4. Nejistota ve znalostech, nejistota v bázi dat.
5. Vysvětlovací schopnosti (proč jsem udělal co).

Expertní systémy dělíme následovně:

1. *Diagnostické* –uvažuje konečný počet hypotéz a na základě dat rozhoduje, která z hypotéz je nejpravděpodobnější.
2. *Plánovací* – generuje plán, konečnou posloupnost kroků, snaží se převést počáteční stav do cílového.
3. *Hybridní (smíšené)* – např. monitorovací systémy, zjištěna chyba $\Rightarrow$ naplánuj odstranění, kombinace výše dvou.

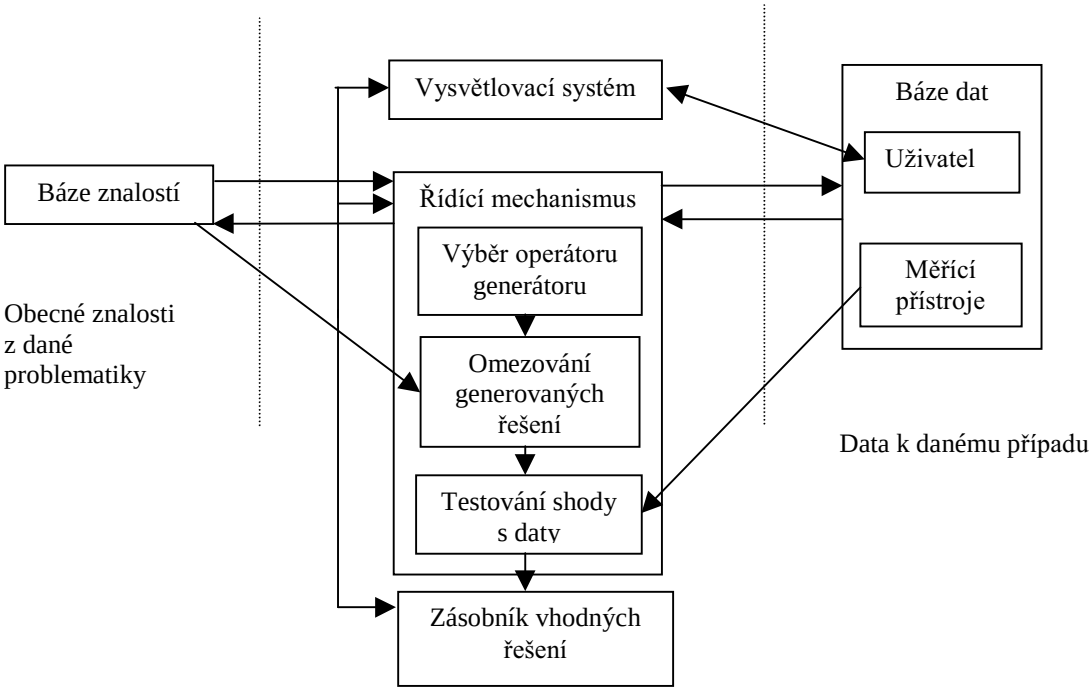
Diagnostický expertní systém - viz. obr. č. 6.



Obrázek č.6

Postup vyhodnocování: báze znalostí (počáteční stav)→ řídící mechanismus (výběr dotazu) → báze dat (dotaz uživateli) → řídící mechanismus → aktuální model → báze znalostí and aktuální model → odbozvovací mechanismus (dotaz) → .....

Plánovací expertní systém – viz obr. č.7.



Obrázek č.7

Řídící mechanismus s využitím báze znalostí a báze dat:

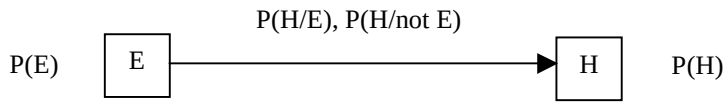
1. Ovlivňuje výběr přípustných operátorů.
2. Omezuje generativní schopnost generátoru použitím znalostí z báze znalostí (např. apriorním zamítnutím některých dílčích posloupností kroků apod.)
3. Řídí testování shody vygenerovaných řešení s daty z báze dat, a tím utváření zásobníku potenciálních řešení, včetně ohodnocení jejich vhodnosti.

Prázdný expertní systém – bez problémově závislých částí, bez báze znalostí a bez báze dat.

Problémově orientovaný expertní systém – prázdný + báze znalostí a báze dat, řeší nějaký problém.

Uzavřený expertní systém – kompletně vyvinuté, nelze je nijak upravovat (báze znalostí), např. lékařské.

Princip činnosti a inferenční síť – většina diagnostických systémů pracuje s pravidly. Práce s neurčitostí poté rozlišuje jednotlivé modely (jak vyjadřovat pravděpodobnost předpokladu, jak přepočítávat pravděpodobnost závěru, atd.). Vyjděme z obr. č.8.



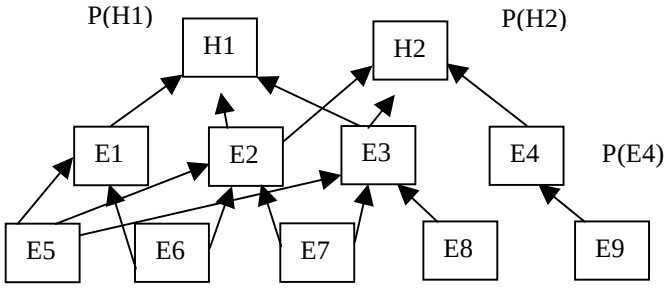
Obrázek č.8

P(E) – pravděpodobnost předpokladu,
P(H) – pravděpodobnost hypotézy,
P(H/E) – jakou pravděpodobnost nabude hypotéza H, je-li E splněno,
P(H/ not E) – jakou pravděpodobnost nabude hypotéza H, není-li E splněno.

Pravidlo: *IF E THEN H WITH THEN ACCEPT H WITH P(H/E) ELSE ACCEPT H WITH P(H/not E).*

Definujme nyní pojem *šance* : *O(H) = P(H)/(1-P(H)), O(H/E) = P(H/E)/(1-P(H/E)*.

Inferenční síť. Způsob úpravy jednotlivých pravděpodobností předpokladů a finálních hypotéz. Předpoklad může býti i závěrem. Uvažujme obr. č.9.

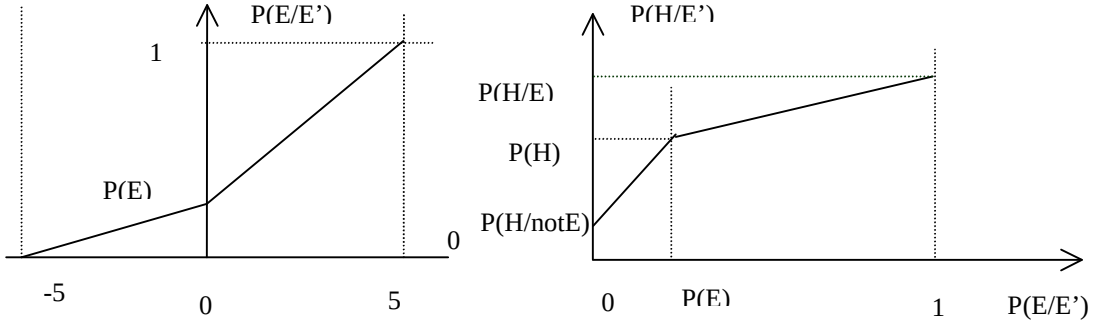


Obrázek č.9

Počáteční nastavení pravděpodobností je nastaveno dle experta. Poté vybereme hypotézu s největší P(H), např. H2. Poté vyberu pravidlo, které mi tuto hypotézu nejvíce upřesní dle *skórovací funkce*., pomocí něhož určím předpoklad, např. E3. To se stává závěrem o podobným způsobem určím předpoklad na nejnižší úrovni, např. E7. Poté položím dotaz. Dle odpovědi upravím hypotézu E7, E2, poté H1 a H2. Nyní se změnily pravd. Hypotéz H1 a H2. Toho si však nevšímám a pokračuji dalším cyklem, popsaným výše s H2. Na konci (všechny dotazy) uvedu pořadí hypotéz dle výsledných pravděpodobností.

Modely expertních systémů: rozlišujeme dle práce s neurčitostí.

1. *Pseudobayesovský model (PROSPECTOR, FELEXPERT)* – využívá následujících dvou vztahů:
O(H/E) = LAMBDA * O(H) a O(H/not E) = not LAMBDA * O(H); kde *LAMBDA* je míra postačitelnosti a *not LAMBDA* míra nezbytnosti.



Obrázek č.10

Levá část obrázku znázorňuje přepočet vstupní informace uživatele (v rozsahu –5 až 5 ⇒ jak moc je pravda) a apriorní pravděpodobnosti předpokladu na aposteriorní pravděpodobnost předpoklad. Pravá část znázorňuje přepočet pravd.předpokladu na pravd. hypotézy. Poté lze napsat:

O(H/E1,E2,..,EN) = LAMBDA1*LAMBDA1*,....,*LANBDAN*O(H),

O(H/not E1,not E2,..,not EN) = not LAMBDA1*not LAMBDA1*,....,*not LANBDAN*O(H),

uvažujeme-li vliv N předpokladů (pravidel) na hypotézu. Pro samotný výpočet používáme efektivních hodnot, využijeme vztahu pro šanci a efektivní síly pravidla s využitím obr. č.10.

LAMBDA‘ = P(H/E‘) / (1-P(H/E‘) * (1-P(H)) /P(H), potom

O(H/E1‘,E2‘,....En‘) = LAMBDA1‘ * LAMBDA2‘ *.....* LAMBDAN‘ * O(H).

2. *Model práce s neurčitostí u systémů Mycin a Emicyn*

MB(H/E) – míra důvěry,
MD(H/E) –míra nedůvěry,
CF(H/E) = MB(H/E) – MD(H/E) –certainly factor.

Přepočet předpokladu na závěr, známe-li CF(E/EE’) díky odpovědi na dotaz:

MB(H/E’) = MB(H/E)*max{0, CF(E/EE’)},

MD(H/E') = MD(H/E)*max{0, CF(E/EE')}.

Pro kombinaci pravidel použijeme následujících vztahů:

MB(H/E1 & E2) = MB(H/E1) + MB(H/E2) - MB(H/E1) * MB(H/E2),

MD(H/E1 & E2) = MD(H/E1) + MD(H/E2) - MD(H/E1) * MD(H/E2),

CF = (MB – MD) / (1 – min {MB,MD}).

3. *Fuzzy set approach*

MFV – míra příslušnosti k množině pravdivých tvrzení,

CF –confidency factor – váha pravidla.

Přepočet předpokladu na hypotézu:

MFV hypotézy = MFV předpokladu * CF.

Vliv několika pravidel:

CF = CF1 + CF2 – CF1*CF2.

Základní architektury. Velká báze znalostí. Nutné dekompozice:

1. *Blackboard (tabule)* – viz. obr. č. 11.- sdílená struktura, kam se zapisují výsledky, všichni z ní můžou číst, ale pouze jeden zapisovat, používají se dvě strategie: *daemon* – málokdy v činnosti, ale přijde-li specifická akce, tak ji zachytí a rychle reaguje, *agenda* – kancelářská práce, pracuje většinu času, úprava zásobníků.
2. *Expertní systémy 2. Generace* – integrace znalostí v objektově orientovaném jazyce, bylo velmi obtížné, proto ⇒ multiagentní systémy.
3. *Multiagentní systémy* – jednotlivé moduly pracují samostatně a v jistém jazyce si posílají zprávy (komunikace), v případě zneschopnění jednoho agenta ⇒ nic se neděje, ostatní ho nahradí, problémy koordinace a komunikace.

Návrh expertních systémů. Tvorba expertních systému se skládá:

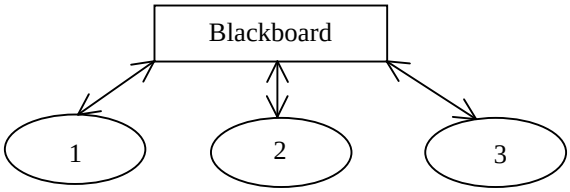
1. Konceptualisace znalostí – identifikace problému, specifikace formy znalostí, modelu uvažování.
2. Získávání znalostí – znalosti je nutné získat od experta.
3. Formalizace znalostí – reprezentace znalostí.
4. Implementace – báze znalostí a inferenčního mechanismu.
5. Testování a ladění.
6. Údržba znalostí a znalostního systému (pro znalostní systém).
7. Případný návrat do jednotlivých stupňů problému.

Komunikační moduly expertních systémů:

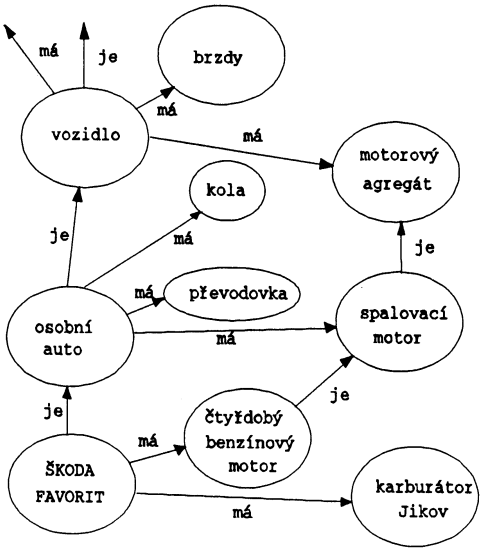
1. Hladký dialog – pouze kladení otázek.
2. Poskytuje relevantní informace, stupně: WHAT – co považuješ za nejdůležitější, WHY – proč kladeš tuto otázku, WHAT-IF – co se stane když.
3. Komunikace systému v přirozeném jazyce.

Nástroje pro realizaci expertních systémů:

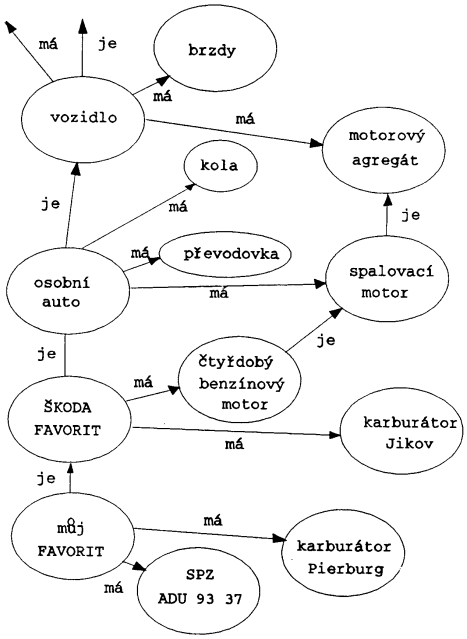
1. Nástroje pro konstrukci pravidlových systémů.
2. Jazyky pro tvorbu systémů založených na tvorbě rámců.
3. Nástroje pro vývoj systémů založených na logickém programování.
4. Jazyky pro objektově orientované programování.
5. Vysoce obecné nástroje – konstrukce ES libovolného typu.



Obrázek č.11



Obrázek č.2



Obrázek č.3

<u>*Metody využíváni znalostí*</u> – velice rozsáhlý a nespecifický pojem, proto některé obory, kde jsou znalosti využívány:

1. *Objektové programování*, jež vychází z principů sémantický sítí.
2. *Dokazování vět* – značné využití rezoluce, kde se snažíme dokázat že množina formulí tvořená modelem světa a negací dokazované formule je sporná.
3. *Plánování matematických důkazů a důkazů pro kvalitu a testování softwaru.*
4. *Logické programování* (Prolog, Eclipse) – využívá rezolučního principu.
5. *Expertní, znalostí a produkční systémy* využívají bázi znalostí pro samotné řízení algoritmu expertízy.
6. *Strukturální rozpoznávání* pro definici primitiv a jejich relací.
7. *Plánování* pro popis současného modelu světa a vztahů mezi jednotlivými elementy tvořících tento svět.
8. *Distribuovaná umělá inteligence* – viz. Proplant → dekompoziční pravidla (PPA), expertní systémy (PMA) atd.
9. *Počítačové vidění* – např. reprezentace obrazu (kvadrantové stromy).

a mnohé další.