

Projektovanie znalostných systémov

Obsah

Obsah.....	1
Zoznam obrázkov.....	2
Znalostná úroveň.....	3
Modelovanie znalostí.....	4
Použitie znalostnej úrovni v praxi.....	4
Heuristická klasifikácia.....	5
Modelovanie procesu návrhu.....	6
Hĺbkové znalosti.....	8
Metódy návrhu znalostného systému.....	9
Klasifikácia.....	18
Konkrétne modely.....	18
Metodológia v KADS.....	25
Analýza v KADS.....	26
Dizajn.....	38
Metodológia v Common-KADS.....	42

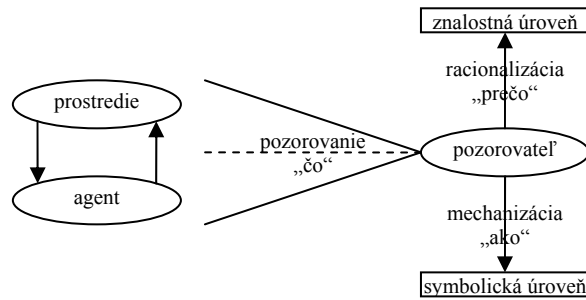
Zoznam obrázkov

Obrázok 1: Vzťah medzi znalostnou a symbolickou úrovňou	4
Obrázok 2: Závislosť medzi znalosťami a chovaním systému na znalostnej úrovni	4
Obrázok 3: Závislosť medzi znalosťami a chovaním systému na knowledge-use level	4
Obrázok 4: Vytvorenie modelu na knowledge-use level	4
Obrázok 5: Bloková schéma práce heuristickej klasifikácie	6
Obrázok 6: Stromová štruktúra obsiahnutá v pravidle	8
Obrázok 7: Príklad doménovej teórie	9
Obrázok 8: Štruktúra systému s hĺbkovými znalosťami	9
Obrázok 9: Schéma KADS	9
Obrázok 10: Schematické znázornenie inferenčného typu v KADS	10
Obrázok 11: Schematické znázornenie znalostnej role v KADS	10
Obrázok 12: Schematické znázornenie podpornej role v KADS	10
Obrázok 13: Schematické znázornenie fragmentu inferenčnej siete v KADS	10
Obrázok 14: Príklad popisu úlohovej vrstvy pomocou grafu v KADS	11
Obrázok 15: Inferenčná sieť heuristickej analýzy v KADS	11
Obrázok 16: Schéma Common-KADS	12
Obrázok 17: Schematické znázornenie inferencie v Common-KADS	12
Obrázok 18: Schematické znázornenie dynamickej znalostnej role v Common-KADS	13
Obrázok 19: Schematické znázornenie statickej znalostnej role v Common-KADS	13
Obrázok 20: Schematické znázornenie komunikačnej funkcie v Common-KADS	13
Obrázok 21: Štyri komunikačné funkcie	13
Obrázok 22: Schematické znázornenie kardinality v Common-KADS	13
Obrázok 23: Príklad inferenčnej siete v Common-KADS	14
Obrázok 24: Schematické znázornenie úlohy v Common-KADS	14
Obrázok 25: Schematické znázornenie metódy v Common-KADS	14
Obrázok 26: Activity diagram v Common-KADS	15
Obrázok 27: Príklad úlohovej vrstvy v Common-KADS	16
Obrázok 28: Schematické znázornenie konceptu v Common-KADS	16
Obrázok 29: Schematické znázornenie relácie v Common-KADS	16
Obrázok 30: Schematické znázornenie vzťahu v Common-KADS	16
Obrázok 31: Schematické znázornenie viac-árnej relácie v Common-KADS	17
Obrázok 32: Schematické znázornenie vzťahu SUBTYPE a SUPERTYPE v Common-KADS	17
Obrázok 33: Schematické znázornenie typu pravidla v Common-KADS	18
Obrázok 34: Príklad prepojenia doménovej a inferenčnej vrstvy v Common-KADS	18
Obrázok 35: Inferenčná sieť v systematickej diagnostike	19
Obrázok 36: Inferenčná sieť v lokalizácii	20
Obrázok 37: Inferenčná sieť v kauzálnom trasovaní	20
Obrázok 38: Inferenčná sieť v systematickom zjemení	21
Obrázok 39: Inferenčná sieť v korelácii	21
Obrázok 40: Inferenčná sieť v ocenení	22
Obrázok 41: Inferenčná sieť v priradení	23
Obrázok 42: Inferenčná sieť vo verifikácii	24
Obrázok 43: Inferenčná sieť v monitorovaní	25
Obrázok 44: Grafické zobrazenie metodológie návrhu znalostných systémov	26
Obrázok 45: Časové hľadisko analýzy systému	27
Obrázok 46: Celkový pohľad na analýzu v KADS	28
Obrázok 47: Schéma procesnej analýzy	29
Obrázok 48: Príklad dekompozície	30
Obrázok 49: Model práce kooperačnej analýzy	31
Obrázok 50: Príklad diagramu prenosu stavov pre analýzu užívateľských činností	33
Obrázok 51: Príklad role activity diagram	34
Obrázok 52: Príklad diagramu prenosu stavov pre analýzu systémových činností	35
Obrázok 53: Schéma práce expertíznej analýzy	36
Obrázok 54: Riešenie problému viacerými znalostnými systémami	36
Obrázok 55: Spojenie znalostných systémov na strategickej vrstve	37
Obrázok 56: Spojenie znalostných systémov na doménovej vrstve	37
Obrázok 57: Spojenie znalostných systémov na inferenčnej vrstve	37
Obrázok 58: Rozdelenie dizajnu	38
Obrázok 59: Schéma práce dizajnu	38
Obrázok 60: Schéma práce KBS dizajnu	40
Obrázok 61: Príklad priradenia návrhových metód	42
Obrázok 62: Rozdelenie metodológie Common-KADS do modelov	42
Obrázok 63: Vzťah transakcie a komunikačného plánu	43
Obrázok 64: Tabuľka špecifikácií správ	44
Obrázok 65: Schéma organizačného modelu	45

Znalostná úroveň

- na každý objekt sa môžeme pozerat' z viacerých hľadísk
 - o napríklad počítačový program
 - symbolická úroveň – tvorená registrami počítača
 - logická úroveň – tvorená sekvenčnými a klopnými obvody počítača
 - obvodová úroveň – tvorená obvody na doskách
 - prvková úroveň – tvorená tranzistormi, odpormi, ...
 - o úrovne tvoria hierarchiu
 - o jednu hierarchiu nemožno použiť pre akýkoľvek objekt
 - o pomocou niektorej úrovne vieme popísať objekt, jeho interface s okolím, ale nie okolie objektu
 - každá úroveň je realizovaná
 - o médium
 - o komponenty
 - sú to základné prvky, ktoré môžu médium meniť
 - o zákon kompozície
 - pravidlo, ako spájať základné komponenty do väčších častí
 - o zákon chovania
 - popisuje ako sa bude chovať systém, za predpokladu, že viem, ako základné komponenty pôsobia na médium a viem, ako sú komponenty medzi sebou pospájané pomocou zákona kompozície
 - napríklad pre prvkovú úroveň
 - o médium – prúd alebo napätie
 - o komponenty – tranzistory, odpory, ...
 - o zákon kompozície – ako sú komponenty spojené dokopy
 - o zákon chovania – ohmový zákon
 - médium je závislé od úrovne z ktorej sa naň pozeráme. Objekt má iné vlastnosti na iných úrovniach
 - popis úrovni
 - o autonómny
 - popis je iba na jednej úrovni
 - nepotrebujeme vedieť ako pracujú podúrovne a okolie
 - o redukcia na nižšiu úroveň
 - popis objektu na úrovne o jednu nižšie
 - redukcia na nižšiu úroveň je vždy možná
 - objekt je možné popísať aj na vyššej úrovni, avšak tento prechod nie je vždy možný
 - popis inteligentného systému
 - o na popis sa vždy snažíme vybrať najvyššiu úroveň
 - zvolíme si symbolickú úroveň
 - o médium
 - symboly
 - o komponenty
 - znalosti
 - znalosti musíme správne reprezentovať
 - predikátový počet prvého rádu
 - stromy
 - produkčné pravidlá
 - ontológia
 - znalosť sa musí vždy vyjadrovať v nejakej reprezentácii
 - zvolením správnej reprezentácie si môžeme problém značne uľahčiť
 - znalosť sa nedá od reprezentácie neďa oddeliť, dochádza k zmiešavaniu, a niekedy je reprezentácia znalosti dôležitejšia ako znalosť samotná
 - reprezentácie je presne definovaná, no znalosť nie
- transfer znalostí
 - o knowledge transfer
 - o transfer znalostí od experta do systému
 - o v praxi to prináša problémy, lebo nie vždy je možné preniesť znalosti do systému
- znalostná úroveň
 - o knowledge level
 - o A. Newell, 1982 publikoval v časopise Artificial Intelligence
 - o zaviedol ďalšiu úroveň pre popis systému
 - o na tejto úrovni sú médium znalostí
 - o inteligentný systém sa na tejto úrovni nazýva agent, ktorý sa skladá
 - ciele
 - sú znalosti o stave systému a okolia
 - telo
 - sú to znalosti, ktoré agent má
 - akcie
 - zásahy, ktoré agent vie vykonať, a tým teda môže modifikovať okolie alebo svoj vnútorný stav
 - o agent sa chová racionálne, a teda sa chce dopracovať k svojmu cieľu
 - o na telo agenta nie sú kladené žiadne obmedzenia, ani na množstvo, ani na tvar znalostí
 - o medzi akciami a cieľmi existuje vzťah, ktorý nie je explicitne vyjadrený. Tento vzťah je reprezentovaný cez telo agenta
 - o v znalostnej úrovni neexistuje zákon kompozície
 - každý inteligentný systém má práve tieto tri úrovne
 - o zákon chovania je daný racionalitou, teda agent sa snaží dosiahnuť cieľ
 - tento zákon má teda globálny charakter
 - zákon chovania sa teda rovná princípu racionality
 - ak k cieľu vedie práve jedna akcia, tak agent sa túto akciu vykoná
 - ak cieľu vedie viac akcií, tak agent nemusí vybrať optimálnu
 - zákon chovania nevieme presne predikovať

Modelovanie znalostí

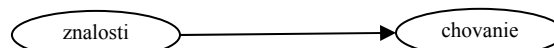


Obrázok 1: Vzťah medzi znalostnou a symbolickou úrovňou

- agent prijíma informácie z prostredia, a pritom naň aj vplýva
- pozorovateľ iba vidí, že agent asi má znalosti a volí akcie tak, aby dosiahol cieľ. Pozorovateľ nevidí znalosti agenta priamo, ale mu ich len prisudzuje, čím vnáša do procesu subjektívny prvok
 - o znalostný inžinier len pozoruje experta a prisudzuje mu znalosti
- pozorovateľ potom môže vytvoriť model na symbolickej alebo znalostnej úrovni
 - o ak vytvára model na znalostnej úrovni, tak racionalizuje chovanie agenta
 - pozorovateľ zisťuje prečo sa agent správa práve tak ako sa správa
 - o ak vytvára model na symbolickej úrovni, tak mechanizuje chovanie agenta
 - tu je už nutná reprezentácia znalostí
- poznámky k modelovaniu znalostí
 - o je to nekonečný proces
 - nevieme povedať, či model presne modeluje znalosti agenta
 - existuje nekonečne veľa aproximácií správania agenta
 - o je to chybový proces
 - v ľubovoľnom okamihu sa môže stať, že časť modelu je zlá
 - o je to inkrementálny proces
 - niečo namodelujeme, potom modelujeme inú časť, ktorá nám umožní zlepšiť prvý model, atď.

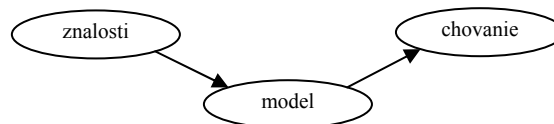
Použitie znalostnej úrovni v praxi

- použitie znalostnej úrovni má dve nevýhody
 - o keďže chýba reprezentácia znalostí, znalosti sa nedajú vyjadriť
 - pozorovateľ si teda ani nemá možnosť zapísať si znalosti
 - o znalosti nemajú hierarchiu
 - znalosti teda neposkytujú informáciu ako získať inú znalosť
- medzi znalostnú úroveň a symbolickú úroveň sa preto vložila ďalšia úroveň
 - o táto úroveň má viac pomenovaní
 - knowledge level
 - epistemologická úroveň
 - knowledge-use level
 - o je podobná ako znalostná úroveň, ale znalosti už majú svoju reprezentáciu
 - o znalosti sa začínajú deliť do skupín
 - o keďže znalosti majú svoju reprezentáciu, tak sa nedá použiť na akékoľvek znalosti
 - o odstránil sa však problém, že nevieme povedať, aké znalosti hľadáme
 - o medzi jednotlivými druhmi znalostí sa začalo používať prepojenie
 - o v znalostnej úrovni môžeme zakresliť takýto diagram



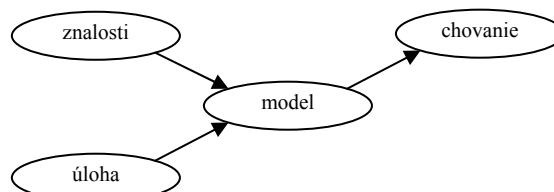
Obrázok 2: Závislosť medzi znalosťami a chovaním systému na znalostnej úrovni

- o pre knowledge-use level by sme ho mohli prekresliť



Obrázok 3: Závislosť medzi znalosťami a chovaním systému na knowledge-use level

- model je nutné vytvoriť
 - vytvorí sa úloha a potom sa vytvorí model, ktorí túto úlohu rieši
 - použitie tých istých znalostí v inom modeli by porušilo princíp racionality

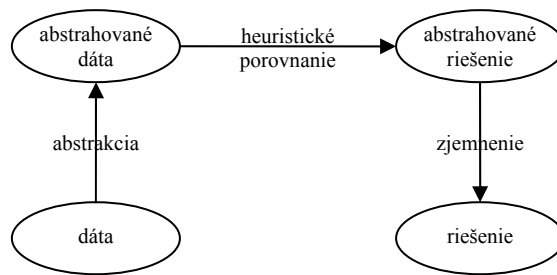


Obrázok 4: Vytvorenie modelu na knowledge-use level

- typy modelov
 - o doménový model
 - je zameraný na konkrétnu oblasť, teda na znalosti z nejakej oblasti
 - doménový model zodpovedá znalostiam
 - je ho možné použiť na rozličné ciele
 - o úlohový model
 - task model
 - hovorí akým spôsobom môžeme splniť úlohu
 - zodpovedá cieľom
 - o metódový model
 - problem solving method
 - obsahuje aj znalosti o konkrétnej oblasti, ale obsahuje aj daný cieľ

Heuristická klasifikácia

- Clancey, 1985
- študoval konkrétne expertné systémy a urobil analýzu ako systémy narábajú s bázou dát
- skúmal viacero druhov expertných systémov, ale zameriaval sa hlavne na diagnostické expertné systémy
- zistil, že v činnosti týchto expertných systémov je vzor, a tento vzor nazval inferenčný vzor
- tento inferenčný vzor je charakterizovaný tromi etapami
 - o abstrakcia
 - o heuristické porovnanie
 - o zjemnenie
- tieto tri etapy zhrnul a nazval ich heuristická klasifikácia
- zaradenie heuristickej klasifikácie do typov modelov
 - o nemôže to byť doménový model, lebo rieši úlohy z viacerých domén
 - o nemôže to byť úlohový model, lebo rieši viacero druhov úloh, nielen tú akým spôsobom môžeme danú úlohu splniť
 - o je to teda metódový model
 - používa sa pre riešenie viacerých úloh
 - používa sa pre prácu s viacerými doménami
- jednoduchá klasifikácia
 - o máme nejaký objekt, ktorý chceme klasifikovať
 - o máme systém tried
 - o chceme povedať, do ktorej triedy daný objekt patrí (s ktorou má vzťah)
 - o medzi triedami existuje nejaká hierarchia (napríklad strom)
 - ak sú triedy na jednej úrovni, potom je to jednoúrovňový strom
 - o potom však objekt klasifikujeme do hierarchie tried
 - o každá trieda má vlastnosti, ktoré musí objekt spĺňať, aby sme mohli prehlásiť, že daný objekt patrí do danej triedy
 - o objekt však môže patriť do viacerých tried, ak sa vlastnosti triedy s vlastnosťami objektu iba čiastočne kryjú
 - objekt potom patrí do koreňa týchto tried
 - o abstrakcia
 - niekedy nevieme presne povedať, aké má objekt vlastnosti
 - nameriame dáta na objekte
 - z nameraných dát zisťujeme vlastnosti objektu
 - typy abstrakcie
 - definičná
 - o je založená na podstatných a nevyhnutných vlastnostiach objektu
 - o príklad: „AK je usporiadanie pravidelné v oboch smeroch, POTOM je to matica“
 - o jedná sa vlastne o vytváranie definícií
 - kvalitatívna (kvantitatívna)
 - o podstatou je práca s numerickými hodnotami
 - o máme dané prahy
 - o podľa nameraných hodnôt a prahov dostaneme danú vlastnosť
 - o vzory a prahy môžu byť závislé od iných hodnôt
 - generalizácia
 - o predpokladáme nejakú hierarchiu tried
 - o ak je daná hierarchia tried strom, tak sa vieme v tejto hierarchii posunúť o úroveň vyššie
 - zovšeobecnenie vlastností
 - pri prechode od hodnôt k vlastnostiam môžeme použiť akýkoľvek typ abstrakcie, dokonca aj viacero druhov naraz
 - o zjemnenie
 - vykonáva sa porovnanie v vyššom uzle v hierarchii
 - postupuje sa v hierarchii tried nadol
 - zošpecifikovanie vlastností
- heuristická klasifikácia
 - o obsahuje abstrakciu a zjemnenie
 - o v hierarchii vlastností tried sú jedny vlastnosti a v hierarchii vlastností objektov sú iné vlastnosti
 - nemožno porovnať priamo vlastnosti triedy s vlastnosťami objektu
 - o heuristické porovnanie
 - využíva sa, aby sme mohli porovnať vlastnosti triedy s vlastnosťami objektu, pričom typy vlastností sú rôzne
 - vytvára asociácie z rôznych pojmových oblastí
 - porovnanie sú heuristické nehierarchické asociácie
 - asociácia nemusí byť stopercentne istá



Obrázok 5: Bloková schéma práce heuristickej klasifikácie

- heuristická klasifikácia musí obsahovať heuristické porovnanie
- heuristická klasifikácia nemusí fungovať presne podľa obrázku
- spôsoby použitia heuristickej klasifikácie
 - riadená dátami
 - data driven
 - typ, ktorý je nakreslený na obrázku
 - je založená na získavaní dát o objekte
 - riadená hypotézami
 - solution driven
 - začíname riešením, a to buď potvrdíme alebo vyvrátíme
 - väčšinou sa nezačína priamo v riešeniach, ale iba v abstrahovaných riešeniach
 - menej práce heuristickej klasifikácie
 - po úspešnom potvrdení abstrahovaného riešenia nastáva zjemnenie
 - oportunistické hľadanie
 - oportunistic search
 - snaha o skombinovanie dopredného a spätného použitia heuristickej klasifikácie
 - vychádzame z nejakého abstrahovaného riešenia
 - zistím aké dáta potrebujem, aby som ho potvrdil alebo vyvrátil
 - získané dáta využijem v abstrakcii a abstrahované dáta heuristicky porovnám s abstrahovanými riešeniami
- výhody heuristickej klasifikácie
 - dôraz dávame na znalostné štruktúry, a tak sa v nich ľahšie vieme orientovať
 - vieme zistiť, ktoré znalosti sú funkčne závislé
 - medzi rôznymi expertnými systémami je podobnosť, a tak je návrh nového expertného systému jednoduchší
- nevýhody heuristickej klasifikácie
 - keďže máme znalostné štruktúry, tieto nemusia postačovať na popis všetkého, čo sa v pravidlách používa
 - schéma heuristickej klasifikácie je veľmi obecná a tak sa v každom kroku využíva mnoho prostriedkov a mechanizmov
 - pretože je tento model veľmi všeobecný, tak ho nájdeme v ktoromkoľvek expertnom systéme a tak nemôžeme vytvárať klasifikáciu expertných systémov

Modelovanie procesu návrhu

- snažíme sa vytvoriť model, ktorý rieši návrhové problémy
- takéto úlohové modely majú často analytický charakter
- návrhová úloha
 - prehľadávame priestor riešení, ktorí je obvykle veľmi veľký (nie je často definovaný hranicami), a tak je možné hľadať riešenie prakticky kdekoľvek v priestore riešení
 - kandidátov na riešenie je veľmi veľa
 - prijateľných kandidátov vzhľadom na ich celkový počet je málo
 - optimálnych kandidátov je veľmi ťažké nájsť
 - artefakt
 - návrh má zostrojiť artefakt (vec, riešenie)
 - mal by spĺňať funkčné chovania
 - musí mať niektoré funkcie
 - má spĺňať ohraničenia
 - skladá sa z
 - komponentov
 - komponenty definujú ohraničenia úlohy
 - repertoár komponentov je daný doménovou oblasťou
 - medzi komponentmi existujú relácie
 - definujú ako sa môžu komponenty medzi sebou spájať
 - komponenty, ktoré chceme použiť môžu a nemusia existovať
 - ak sa použije neexistujúci komponent, úloha sa transformuje na návrh tohto komponentu
 - rekurzia
 - funkcie
 - definujú stav alebo stavy, ktoré artefakt dosiahne alebo sa im vyhne, pri daných počiatočných podmienkach
 - sú dané explicitne alebo implicitne
 - ohraničenia
 - požiadavky alebo hranice, ktoré má daný artefakt spĺňať
 - v rámci ohraničenia sú vlastne vyhradené funkcie (funkčné ohraničenia)
 - delenia
 - ohraničenie na samotný artefakt
 - ohraničenie na proces výroby artefaktu
 - ohraničenie na samotný proces návrhu artefaktu

- každú metódu možno navrhnuť pomocou defaultnej metódy, ale tieto nás nezaujímajú
- PCM family
 - rodina PCM metód
 - nie je to jedna metóda
 - tieto metódy majú štyri časti
 1. návrh riešenia (propose)
 2. kontrola riešenia (verify)
 3. kritika riešenia (critic)
 4. oprava riešenia (modify)
 - najjednoduchšia metóda
 - návrh riešenia
 - vytvoríme návrh riešenia
 - kontrola riešenia
 - testovanie, či sú všetky ohraničenia splnené
 - ak sú testy v poriadku, tak končíme s úspechom
 - kritika riešenia
 - snažíme sa zistiť, prečo kontrola riešenia neprešla
 - spätná diagnostika
 - ak sa nepodarí chybu skritizovať, tak končíme s neúspechom
 - modifikácia riešenia
 - zmeníme riešenie podľa toho, aká bola kritika riešenia
 - vrátime sa na kontrolu riešenia
 - metóda nemusí obsahovať všetky etapy, ani ich nemusí používať v danom poradí
 - návrh riešenia
 - je to prvotný návrh riešenia
 - metódy používané pri návrhu
 - metóda kompozície a dekompozície
 - hrubý popis
 - úlohu si dekomponujeme na menšie úlohy (artefakt rozdelíme na menšie celky)
 - vyriešime čiastkové úlohy
 - riešenia spolu skombinujeme
 - má štyri fázy
 1. dekompozícia
 - pre dekompozíciu máme mnoho možností
 - medzi jednotlivými dekompozíciami môžu existovať vzťahy
 - hierarchia dekompozícií
 - v tejto časti väčšinou nebýva neúspech
 2. vygenerovanie špecifikácií pre podúlohy
 - ideálne je, ak časť ohraničení úlohy plní pre ohraničenia podúlohy a ohraničenia pre podúlohy sú disjunktné
 - horšie je ak sa takto úloha nedá rozdeliť
 - jednotlivé špecifikácie a tým aj časti úloh sa prekrývajú
 - môžu existovať ohraničenia na spojenie jednotlivých podúloh spolu
 3. riešenie podúloh
 - riešenie podúloh sa nerozumie návrh komponentov
 - medzi jednotlivými podúlohami môžu byť závislosti
 - ak navrhnem jednu časť, potom mi môžu vzniknúť ohraničenia pre iné podúlohy
 - je to teda vybratie správneho postupu riešenia podúloh
 4. kompozícia
 - zoberieme jednotlivé podúlohy a spojíme ich spolu
 - je to jednoduchý proces, lebo dekompozícia je opak kompozície
 - ak sa proces nepodarí, môžeme prejsť na dekompozíciu, alebo na riešenie podúloh
 - mnoho úloh v praxi sa označuje ako rutinný návrh
 - expert nešpecifikuje ako urobiť dekompozíciu
 - dekompozícia vyplynie z domény
 - expert vykoná ľahko vygenerovanie špecifikácií a riešenie podúloh, lebo to má natrénované
 - vykonáva ich rutinne
 - expert nemusí robiť kompozíciu
 - metóda case base reasoning (CBR)
 - vo svojej pamäti nájdem podobné riešenie
 - podobnosť môže byť v zadaní, doméne, ...
 - vyberieme z pamäti dvojicu zadanie riešenie
 - vypočítame rozdiel medzi zadaním z dvojice a zadaním, ktoré riešime
 - riešenie upravíme podľa rozdielu
 - potrebujeme znalosti
 - ako vybrať dvojicu zadanie, riešenie
 - ako porovnať zadaná
 - ako upraviť riešenie podľa rozdielu
 - metóda splnenia ohraničení (CS)
 - constraint satisfaction
 - riešenie sa generuje tak, aby boli splnené ohraničenia
 - sú na to matematické algoritmy
 - nevyžaduje veľa znalostí
- kontrola riešenia

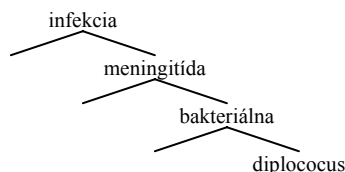
- návrh musíme overiť voči funkčným požiadavkám a ohraničeniam
- máme dve možnosti
 - vypočítame charakteristiku návrhu
 - napríklad váha artefaktu je váhou všetkých jeho častí
 - simulácia
 - zostrojíme komponent a zisťujeme, či spĺňa všetky ohraničenia
 - delenie
 - kvalitatívna
 - zistíme, či riešenie prekmitlo za požadovanú hranicu
 - kvantitatívna
 - zistíme, o koľko riešenie prekmitlo za požadovanú hranicu
- kritika riešenia
 - je to vlastne obrátená diagnostika
 - máme artefakt, ktorý nefunguje, tak ako by sme si predstavovali
 - musíme zistiť prečo artefakt nefunguje
 - nechceme zistiť ktorá časť spôsobuje chybu, ale kde sme v návrhu riešenia spravili chybu
 - na to potrebujeme zodpovedajúce znalosti
 - ideálne je ak máme znalosti, ktorá časť artefaktu za čo zodpovedá
 - (analýza závislostí) dependentis analysis
 - znalosti o chovaní artefaktu a jeho jednotlivých častí
 - teda hľadanie chyby od zlého chovania artefaktu, cez chybnú časť, až k chybe v postupe návrhu artefaktu
 - môžeme využiť heuristickú analýzu
- modifikácia riešenia
 - druhy modifikácie
 - backtracking
 - ak máme analýzu závislostí, tak sa môžeme v postupe vrátiť a vykonať danú zmenu
 - je potrebné mnoho znalostí
 - hillclimbing
 - preskúmame okolie bodu a v jeho okolí sa posunieme k najlepšiemu
 - toto sa rekurzívne opakuje
 - často sa táto metóda používa pri parametrických návrhoch
 - je potrebné navrhnuť parametre riešenia
 - nie je potrebných veľa znalostí
 - nemusíme meniť artefakt, ale môžeme meniť aj požiadavky na ohraničenia a spustiť návrh procesu odznova

Hĺbkové znalosti

- deep knowledge
- je to doménový model znalostného systému
- vychádza z myšlienky, že bežné pravidlá kódujú znalosti, ale aj to ako ich použiť
 - stratégie prehľadávania riešení
 - stratégia obmedzenia priestoru riešení
 - stratégie, ako postupovať, ak sa v systéme vyskytne neurčitosť, alebo chyba
 - príklad pravidla:

IF 1. infekcia je meningitída
 2. podtyp meningitídy je bakteriálna
 3. k dispozícii je len náhodný prejav
 4. pacient má najmenej 17 rokov
 5. pacient je alkoholik
 THEN je podozrenie na to, že diplococcus spôsobil danú meningitídu

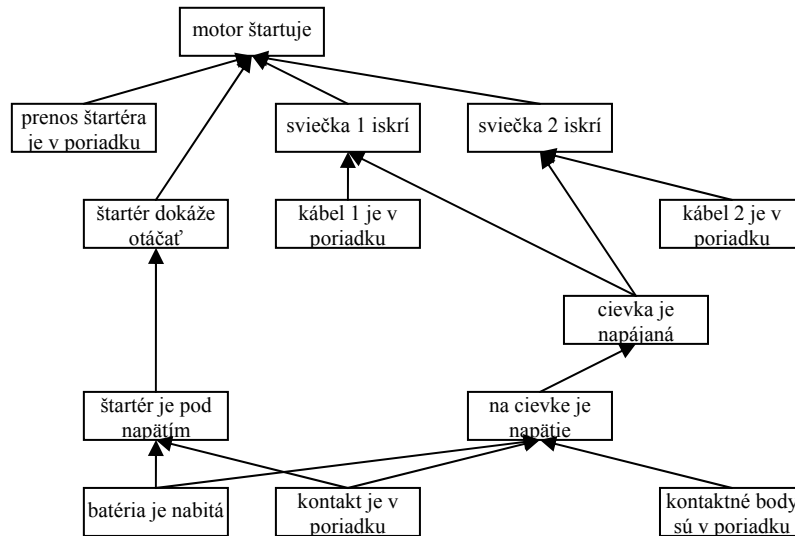
- v pravidle je zahrnutá stromová štruktúra



Obrázok 6: Stromová štruktúra obsiahnutá v pravidle

- v pravidle je obsiahnutá, aj tzv. screaming klauzula
 - 4. klauzula len zisťuje, či má zmysel položiť piatu otázku
 - vek v pravidle nemá vplyv na výsledok
- v báze znalostí (surface knowledge) sú zahrnuté
 - konkrétne znalosti – domain theory
 - stratégie – problem solving knowledge
- konkrétne znalosti a stratégie spolu vytvárajú hĺbkové znalosti
- doménová teória
 - znalosti sú zoradené v hierarchii
 - stromy
 - siete
 - celok – časť
 - môže byť využitá vo viacerých modeloch a sú teda nezávislé na použití
 - hierarchia týchto znalostí ostáva stála, len sa inak používa
 - doménová teória je teda relatívne stála

- o príklad
 - kauzálna sieť

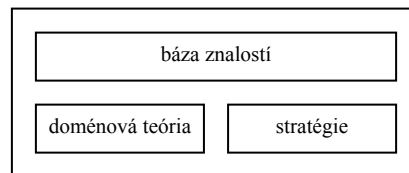


Obrázok 7: Príklad doménovej teórie

- stratégie
 - o je to spôsob práce so sieťou
 - o predpokladajme, že nevieme overiť, či je na cievke napätie, či je cievka napájaná, a či je prenos štartéra v poriadku
 - o zo siete sa dá odvodiť pravidlo

IF sviečka 1 neiskrí AND kábel 1 je v poriadku THEN skontroluj kontakty

- o spôsob použitia siete je možné zaznamenať
- o vznikol systém, ktorý má takúto štruktúru

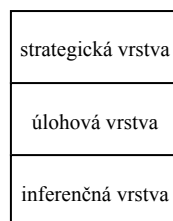


Obrázok 8: Štruktúra systému s hĺbkovými znalosťami

- systém rieši úlohu najskôr v báze znalostí (surface knowledge), pretože je to rýchlejšie
- ak sa úloha nepodarí vyriešiť, tak sa prepája na doménovú teóriu
- to čo si systém odvodí v doménovej teórii, si pridá do bázy znalostí
- ak užívateľ požiada o vysvetlenie, tak sa použije doménová teória a užívateľ dostane presnejšiu odpoveď
- takéto expertné systémy sa nazývajú expertné systémy 2. generácie
- najčastejšie sa využíva v doménovej teórii kauzálna sieť

Metódy návrhu znalostného systému

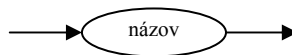
- aké modely znalostí sa využívajú v rôznych znalostných systémoch
- model typu KADS
 - o používa model typu GTS
 - generic task model
 - o je to úlohový model
 - o GTS je trojvrstvový model



Obrázok 9: Schéma KADS

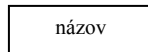
- o inferenčná vrstva
 - definuje aké inferencie je možné vykonať
 - pod inferenciou sa chápe minimálny riešiaci krok
 - inferencia je atomická činnosť a zároveň čierna skrinka, o ktorej nás nezaujíma ako funguje
 - na inferenčnej vrstve rozoznávame
 - inferenčný typ
 - znalostná rola

- inferenčná sieť (znalostná štruktúra)
- inferenčný typ
 - na vstupe má nejakú znalosť a na výstupe produkuje inú znalosť



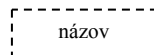
Obrázok 10: Schematické znázornenie inferenčného typu v KADS

- skladá sa
 - názov inferenčného typu
 - jeden alebo viac vstupov
 - jeden alebo viac výstupov
- znalostná rola
 - je to fragment znalostí používaný konkrétnym spôsobom
 - zvyčajne z doménových znalostí
 - limituje spôsob použitia znalosti
 - znalostná rola tvorí vstupy a výstupy inferenčných typov



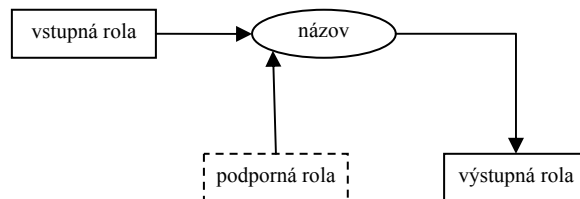
Obrázok 11: Schematické znázornenie znalostnej role v KADS

- podporná rola
 - špeciálny typ znalostnej role
 - podporná rola vždy tvorí vstup inferenčného typu
 - inferenčný typ berie ako vstup znalostnú rolu a produkuje výstupnú znalostnú rolu a vykonáva to na základe podpornej role



Obrázok 12: Schematické znázornenie podpornej role v KADS

- inferenčná sieť
 - spojenie inferenčných typov a znalostných rôl
 - fragment inferenčnej siete



Obrázok 13: Schematické znázornenie fragmentu inferenčnej siete v KADS

- popis inferenčnej vrstvy
 - presné definovanie znalostných rôl a typov
 - definícia znalostnej roly

název komentár

- definícia inferenčného typu
 - je to frame, ktorý má sloty pre vstupné, výstupné a podporné role
 - popisuje metódu ako vykonať inferenčný typ

název popis vstupné role výstupné role metódy podporné role

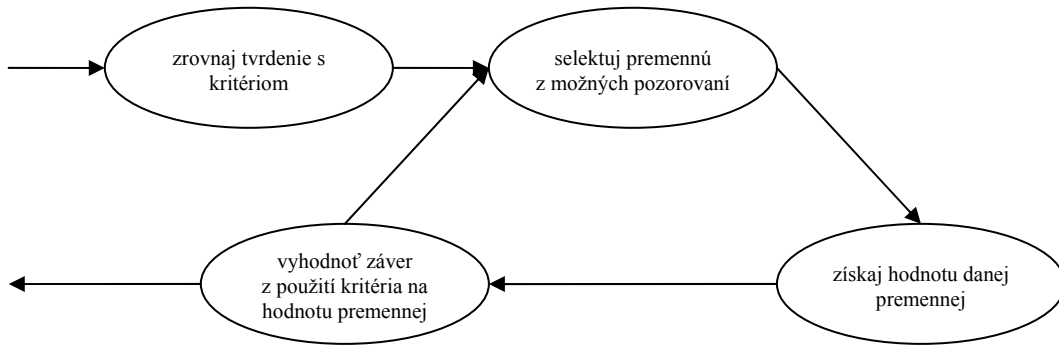
- úlohová vrstva
 - inferenčná vrstva nehovorí kedy sa inferencia vykonáva, za akých podmienok, a pod.
 - definuje použitie inferencií
 - pseudokód

```

verifikuj (tvrdenie) pomocou
zrovnaj tvrdenie s kritériom
UNTIL záver nedosiahnuteľný
    selektuj premennú z možných pozorovaní
    získaj hodnotu danej premennej
    vyhodnot' záver z použitia kritéria na hodnotu premennej
  
```

- dokazujeme tvrdenie na základe kritérií
- budem dokazovať, kým sa mi to nepodarí
 - nikdy neskončí
- vyberiem premennú (opýtam sa) a porovnam ju s kritériom
- popis úlohovej vrstvy je viazaný na inferenčnú vrstvu
 - vkladá použitie inferenčnej vrstvy do KADS
 - pomocou grafu

- používajú sa iba inferenčné typy
- príklad



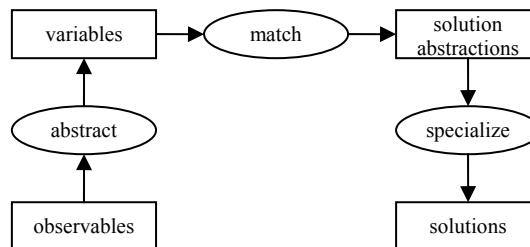
Obrázok 14: Príklad popisu úlohovej vrstvy pomocou grafu v KADS

- vyjadrenie pomocou pravidiel
 - príklad

IF nie je kritérium
THEN zrovnaj tvrdenie s kritériom

IF nebol dosiahnutý záver
THEN selektuj premennú z možných pozorovaní
získaj hodnotu danej premennej
vyhodnoť záver z použitia kritéria na hodnotu premennej

- strategická úroveň
 - nie je pre ňu definovaný žiaden formalizmus
 - rieši problémy, keď môžeme použiť aspoň dve úlohové vrstvy
 - rozhoduje, ktorá sa použije
 - rieši problémy, keď je na úlohovej úrovni viacero popisov
 - rieši problémy, keď je úlohová úroveň parametrizovaná
 - rozhoduje čo je cieľom riešenia
 - podľa toho sa vyberá daná úlohová vrstva
 - rozhoduje akým spôsobom dosiahnuť cieľ
 - sekvencia cieľov
 - vkladanie dodatočných cieľov
 - vypustenie niektorých cieľov
 - zaradenie cieľa do štruktúry
 - výber módu riešenia úlohy
 - napríklad učiace systémy
 - vyučujúci mód
 - skúšajúci mód
- je vybudovaná knižnica modelov GTS
 - každý model má dané úrovne
 - strategickú úroveň nemusí mať
 - zanalyzuje sa úloha, nájde sa príslušný model z knižnice, trochu sa upraví a použije sa
 - model je základná tehlička, ktorá sa spája do celku
 - znalostnému inžinierovi to môže pomôcť pri získavaní znalostí, lebo bude vedieť, aké znalosti má získať
- príklad heuristickej klasifikácie v KADS
 - inferenčná sieť pre heuristickú klasifikáciu v KADS môže vyzeráť takto



Obrázok 15: Inferenčná sieť heuristickej analýzy v KADS

- k danej sieti patrí dokument, ktorý popisuje dané inferenčné typy a znalostné roly
- ak chceme daný model použiť pre konkrétny prípad musíme k nemu pripojiť ešte doménovú vrstvu
 - obsahuje poznatky z danej oblasti
 - prepojenie doménovej vrstvy sa deje na základe prepojenia znalostí z doménovej vrstvy so znalostnými rolami
 - znalostné role teda musia obsahovať smerník na znalosti z doménovej oblasti
- okrem prepojenia je možné definovať aj názvy znalostných rôl podľa danej domény
 - užívateľky prijateľnejšie
- popis inferenčného typu

názov: abstract
 popis:
 vstupné role: observables
 výstupné role: variables
 metóda: vyberieme si konkrétnu metódu na abstrakciu z ponúkaných
 podporné role: sú závislé od vybranej metódy

- popis úlohovej vrstvy
 - riadenie dátami

```

HC(+Observables, -Solutions) by
  obtainData(for Observables)
  abstract(+Observables, -Variables)
  match(+Variables, -Solution abstractions)
  specialize(+Solution abstractions, -Solutions)
  
```

- riadenie cieľmi

```

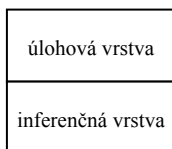
HC(+Observables, -Solutions) by
  specialize(-Solutions, -Solution abstraction)
  match(-Solution abstractions, -Variables)
  abstract(-Variables, +Observables)
  obtainData(for Observables)
  
```

- + znamená, že v čase vyhodnocovania daný výraz poznáme
- - znamená, že v čase vyhodnocovania daný výraz nepoznáme

- strategická vrstva
 - mnoho modelov v KADS ju nemá
 - je na nej napísané, ktorú metódu z úlohovej vrstvy použiť
 - v našom prípade či riadenie dátami, alebo riadenie cieľmi
 - napríklad: ak sa dajú získať všetky vstupy, potom použiť riadenie dátami
 - v strategickvej vrstve môže byť napísané do akej hĺbky robiť špecifikáciu
 - napríklad: hĺbka špecifikácie sa mení podľa druhu aplikácie

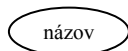
- model typu Common-KADS

- bol následníkom KADS
- snažil sa vybrať dobré veci z modelu typu KADS a zlé odstrániť
- celý model postavil na modernejšiu úroveň
- veľkým rozdielom od KADS je to, že neobsahuje strategickú vrstvu
 - strategická vrstva obsahovala len slovné popisy
 - ak bolo potrebné použiť nejakú stratégiu, tak sa presunula do úlohovej vrstvy
 - model je teda realizovaný na dvoch vrstvách



Obrázok 16: Schéma Common-KADS

- inferenčná vrstva
 - skladá sa z
 - inferencia
 - používa sa namiesto označenia inferenčný typ
 - označuje sa



Obrázok 17: Schematické znázornenie inferencie v Common-KADS

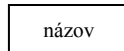
- typicky obsahuje pamäť
 - je rozdiel medzi spustením inferencie a jej znovuspustením
 - spustenie vygeneruje znalostnú rolu a pri tej istej vstupnej znalostnej role vygeneruje tú istú výstupnú znalostnú rolu
 - znovuspustenie vygeneruje znalostnú rolu, ale pri tej istej vstupnej role vyprodukuje inú výstupnú znalostnú rolu
- definícia inferencie

```

INFERENCE: názov inferencie
ROLES
  INPUT: vstupné dynamické znalostné role
  OUTPUT: výstupné dynamické znalostné role
  STATIC: statické znalostné role
DESCRIPTION: popis
END INFERENCE
  
```

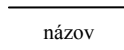
- znalostná rola
 - delia sa do dvoch kategórií
 - dynamické znalostné role

- znalostné role, ktoré sa za behu menia
- typicky je to vstup alebo výstup inferencie
- dynamická rola nemusí byť prepojená s doménovou vrstvou
- označuje sa



Obrázok 18: Schematické znázornenie dynamickej znalostnej role v Common-KADS

- statické znalostné role
 - znalostné role, ktoré sa za behu nemenia
 - typicky sú to znalosti, ktoré používam na riešenie problému
 - statická rola musí byť vždy prepojená s doménovou vrstvou
 - označuje sa



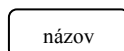
Obrázok 19: Schematické znázornenie statickej znalostnej role v Common-KADS

- definícia znalostnej role

```

KNOWLEDGE ROLE: náзов
TYPE: static, dynamic
DOMAIN MAPPING: mapovanie na doménovú vrstvu
DESCRIPTION: popis
END KNOWLEDGE ROLE
  
```

- komunikačné funkcie
 - používajú sa na komunikáciu systému s okolím
 - označuje sa



Obrázok 20: Schematické znázornenie komunikačnej funkcie v Common-KADS

- zaviedol štyri predefinované komunikačné funkcie
 - rozdeľujú sa podľa toho
 - či znalosti vlastní systém, alebo vonkajší agent
 - či sa komunikácia inicializuje zvonku, alebo zvnútra systému

	interná inicializácia	externá inicializácia
interná znalosť	obtain	receive
externá znalosť	present	provide

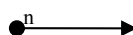
Obrázok 21: Štyri komunikačné funkcie

- definícia komunikačnej funkcie

```

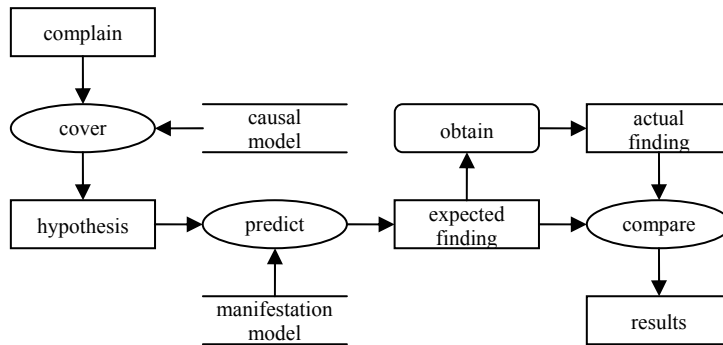
TRANSFER FUNCTION
TYPE: obtain, receive, present, provide
ROLES
INPUT: vstupné znalostné role
OUTPUT: výstupné znalostné role
DESCRIPTION: popis
END TRANSFER FUNCTION
  
```

- kardinalita
 - každá inferencia nemusí produkovať iba jednu znalostnú rolu, ani nemusí mať na vstupe iba jednu znalostnú rolu
 - pomocou kardinality rozlíšili, či má inferencia iba jednu alebo viac vstupných alebo výstupných rôl
 - môže sa pridávať aj číslo na označenie kardinality
 - označuje sa:



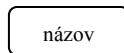
Obrázok 22: Schematické znázornenie kardinality v Common-KADS

- príklad inferenčnej siete



Obrázok 23: Príklad inferenčnej siete v Common-KADS

- predpokladajme, že príde požiadavka, že je nejaký problém
- na základe inferencie cover zistíme hypotézu, ktorá môže byť za daný problém zodpovedná, ktorú sa následne snažíme dokázať, alebo vyvrátiť
- ak je príčinou nejaká hypotéza, po predikcii získame predpokladaný záver
- to je príčinou interného volania funkcie obtain, aby sa získali aktuálne závery
- aktuálne a predpokladané závery sa porovnávajú a tým potvrdia alebo vyvrátia danú hypotézu
- informačnú vrstvu je možné anotovať
 - ukážka príkladu
 - lepšia zrozumiteľnosť
- úlohová vrstva
 - bola podstatne hlbšie prepracovaná ako inferenčná vrstva
 - zavádzajú sa dva pojmy
 - úloha
 - cieľ, ktorý chcem dosiahnuť
 - označuje sa



Obrázok 24: Schematické znázornenie úlohy v Common-KADS

- snaha je o rozloženie úlohy na
 - úlohy
 - komunikačné funkcie
 - inferencie
- úloha sa rozkladá dovedy, kým sú všetky úlohy rozdelené iba na komunikačné funkcie a inferencie
- definícia úlohy

TASK: názov

GOAL: popis cieľa

ROLES

INPUT: znalostná rola, komunikačná funkcia, úloha (popis)

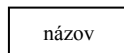
OUTPUT: fault_category, evidence

SPECIFICATION: popis

DESCRIPTION: popis

END TASK

- úloha má dva typy znalostných rôľ
 - fault_category
 - hypotéza vysvetľuje sťažnosť a súčasne je vysvetlená pozorovanými faktami
 - evidence
 - je množina pozorovaných javov
- fault_category ani evidence nenájdeme na inferenčnej vrstve
- na úlohovej vrstve môže byť rola, ktorá nie je definovaná na inferenčnej vrstve
- na inferenčnej vrstve môže byť rola, ktorá nie je definovaná na úlohovej vrstve
- ak má rola z úlohovej vrstvy protejšok v inferenčnej vrstve, potom ho má aj v doménovej vrstve
- znalostná rola z inferenčnej vrstvy má vždy protejšok v doménovej vrstve
- znalostná rola z inferenčnej vrstvy má protejšok v úlohovej vrstve ako úlohu, alebo pomocnú rolu
- metóda
 - spôsob, akým chcem daný cieľ dosiahnuť
 - označuje sa



Obrázok 25: Schematické znázornenie metódy v Common-KADS

- definícia metódy

TASK METHOD: názov metódy

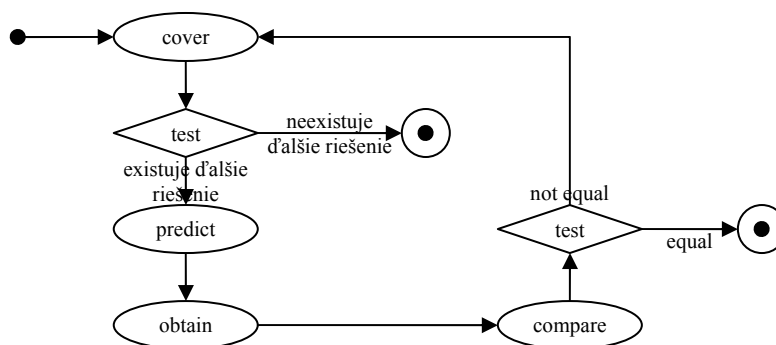
REALIZE: názov úlohy
 DECOMPOSITION
 INFERENCE: zoznam inferencií
 TRANSFER FUNCTION: zoznam komunikačných funkcií
 TASKS: zoznam úloh
 ROLES
 INPUT: vstupné role
 OUTPUT: výstupné role
 INTERMEDIATE: všetky role, ktoré používame okrem
 vstupných a výstupných
 CONTROL STRUCTURE:
 DESCRIPTION: popis
 END TASK METHOD

- o kontrolná štruktúra

```

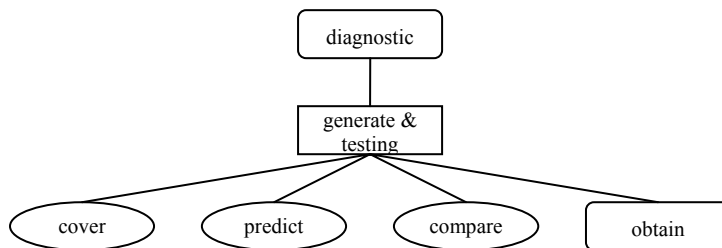
WHILE NEW_SOLUTION cover(complain → hypothesis)
DO
  predict(hypothesis → expected finding)
  obtain(expected finding → actual finding)
  evidence = evidence ADD actual finding
  compare(expected finding + actual finding → result)
  IF result == equal
    THEN „break for loop“
  END IF
END WHILE
IF result == equal
  THEN fault_category = hypothesis
  ELSE „no solution“
END IF
  
```

- presne definujú spôsob použitia znalostných rôl
- NEW_SOLUTION znamená, že funkcii priradujeme pamäť, to znamená, že keď sa systém vráti znovu k volaniu tejto funkcie, tak bude pokračovať od bodu, kde skončila
- namiesto NEW_SOLUTION je možné použiť aj HAS_SOLUTION, pričom v tomto prípade sa zisťuje, či existuje hypotéza, ktorú by dokázala vygenerovať
- v tele môžeme volať inferencie, komunikačné funkcie, alebo podúlohu
- v tele kontrolnej štruktúry môžeme priamo manipulovať so znalostnými rolami
 - znalostné role môžeme meniť
 - vieme zistiť, čo je v danej znalostnej role
 - vieme informáciu zo znalostnej role čítať
 - vieme zmeniť obsah znalostnej role
- v kontrolných štruktúrach sa používajú cykly vetvenia, a pod.
- okrem textového popisu sa znalostné štruktúry znázorňujú aj graficky
 - nazýva sa activity diagram
 - takýto diagram nerozlišuje medzi inferenciami, komunikačnými funkciami a podúlohami
 - nepoužíva znalostné role



Obrázok 26: Activity diagram v Common-KADS

- príklad úlohovej vrstvy



Obrázok 27: Príklad úlohovej vrstvy v Common-KADS

- doménová vrstva
 - model musíme napojiť na konkrétnu doménovú oblasť
 - doménová vrstva sa delí na dve časti
 - doménová schéma (domain scheme)
 - popis znalostí z doménovej oblasti
 - hovorí o tom, aké znalosti sú v modeli použité a aké štruktúry sa na ich reprezentáciu používajú
 - definujú sa znalostné typy
 - vytvárame model znalostí podobne ako vytvárame tabuľky v databázach
 - skladá sa z troch základných častí
 - koncept (concept)
 - definujeme triedu objektov s podobnými vlastnosťami
 - objekty vlastne potom chápeme ako jeden typ
 - objekt je charakterizovaný atribútmi
 - atribúty nadobúdajú hodnoty
 - hodnoty sú z definovaných typov hodnôt
 - textový popis

```

CONCEPT palivo
  ATTRIBUTES
    oktánové číslo: OKT_CIS
  DESCRIPTION: popis
END CONCEPT

```

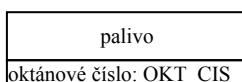
```

VALUE_TYPE OKT_CIS
  VALUE_LIST(91, 95, 98)
  TYPE: nominal
END VALUE_TYPE

```

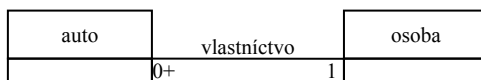
- typ hodnoty môže byť aj ordinálny, potom je definované aj poradie prvkov v zozname
- typ hodnoty nemusí byť daný iba zoznamom hodnôt, ale aj intervalom a pod.
- existujú preddefinované typy hodnôt, ako napríklad INTEGER, a pod.

- grafický zápis



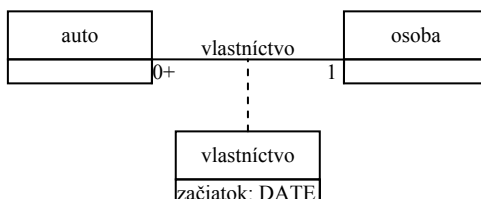
Obrázok 28: Schematické znázornenie konceptu v Common-KADS

- relácie (relations)
 - definujú vzťahy medzi konceptmi
 - môžu byť binárne, alebo viac-árne
 - grafický zápis



Obrázok 29: Schematické znázornenie relácie v Common-KADS

- relácia má svoj názov
- definuje sa kardinalita relácie
- môžu sa používať aj orientované vzťahy
- vzťahu môžeme pridať atribúty



Obrázok 30: Schematické znázornenie vzťahu v Common-KADS

- textový zápis

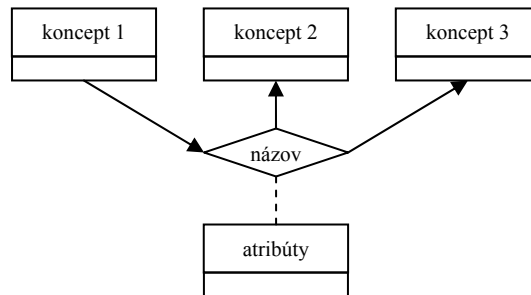
```

BINARY_RELATION: vlastníctvo
  ARGUMENT 1: auto
  CARDINALITY: ANY
  ARGUMENT 2: osoba
  CARDINALITY: 1
  ATTRIBUTES
    začiatok: DATE
  DESCRIPTION: popis
END BINARY_RELATION

```

- ak by sme mali orientovanú reláciu, často sa uvádza aj časť INVERSE, ktorá predstavuje inverznú reláciu

- grafické znázornenie n-árnej relácie



Obrázok 31: Schematické znázornenie viac-árnej relácie v Common-KADS

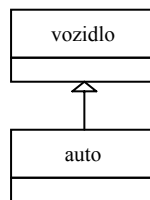
- vzťah typu SUBTYPE a SUPERTYPE
 - mohli by sa vytvárať ako relácie
 - zapisujeme ich priamo do konceptu
 - alebo do atribútov relácií
 - takto je možné pridať atribút do konceptu, zmeniť jeho hodnotu, alebo zmeniť jeho typ
 - textový zápis

```

CONCEPT názov
  SUBTYPE OF: nadradený koncept
  ATTRIBUTES: atribúty
  DESCRIPTION: popis
END CONCEPT

```

- grafický zápis



Obrázok 32: Schematické znázornenie vzťahu SUBTYPE a SUPERTYPE v Common-KADS

- typy pravidiel (rule types)
 - používajú sa na vyjadrenie vzťahu medzi jednotlivými atribútmi konceptov
 - takto vlastne definujeme iba to, že pravidlo daného typu môže existovať, jeho konkrétny výskyt je zadaný v znalostnej báze
 - textový zápis

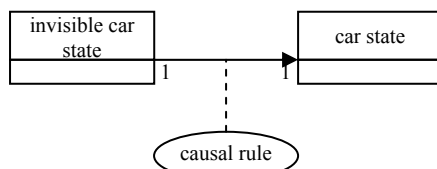
```

RULE_TYPE causal rule
  ANTECEDENT: invisible car state
  CARDINALITY: 1
  CONSEQUENT: car state
  CARDINALITY: 1
  CONNECTION SYMBOL: causes
  DESCRIPTION: popis
END RULE_TYPE

```

- v častiach ANTECEDENT a CONSEQUENT sa neuvádzajú priamo atribúty, ale iba koncepty, a potom daný typ pravidla platí pre všetky atribúty daných konceptov
- CONNECTION SYMBOL označuje typ pravidla, napríklad kauzálny vzťah

- grafický zápis



Obrázok 33: Schematické znázornenie typu pravidla v Common-KADS

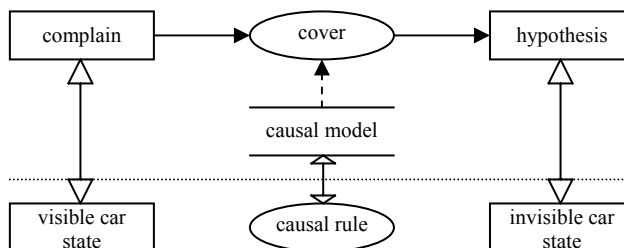
- znalostná báza (knowledge base)
 - obsahuje konkrétne inštancie znalostných rôl, obsiahnutých v doménových schémach
 - podobne ako konkrétne záznamy v databáze
 - je to vlastne zbierka konkrétnych faktov
 - textový zápis

```

KNOWLEDGE_BASE car_network
USES:
    causal rule FROM názov
    manifest rule FROM názov
EXPRESIONS:
    battery.state == low causes engine.status = not start
DESCRIPTION: popis
END KNOWLEDGE_BASE

```

- prepojenie medzi doménovou a inferenčnou vrstvou
 - príklad



Obrázok 34: Príklad prepojenia doménovej a inferenčnej vrstvy v Common-KADS

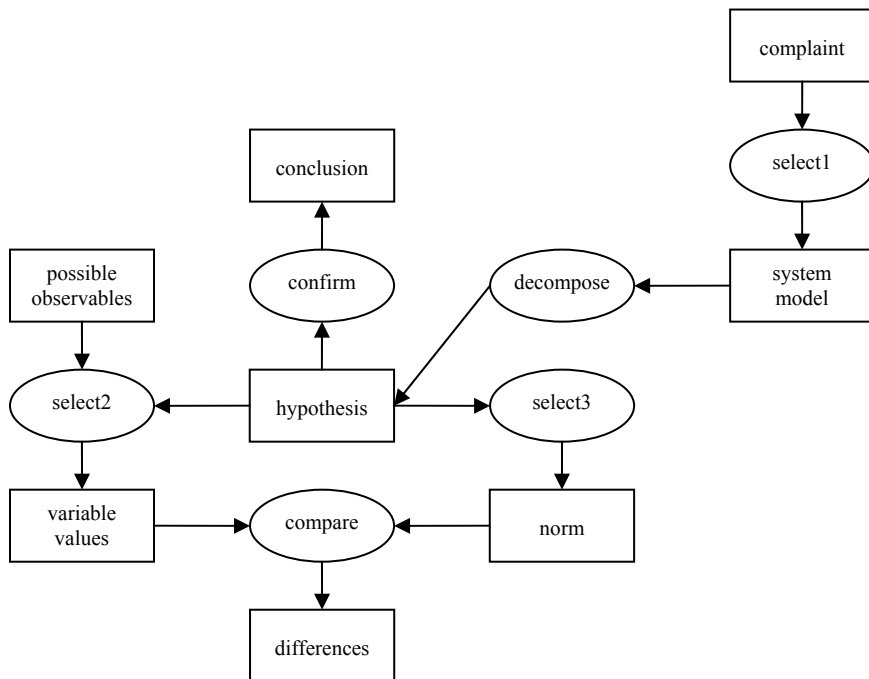
Klasifikácia

- klasifikácia sa často používa ako základná inferencia
- na vstupe klasifikácie je objekt s vlastnosťami
- chceme ho zatriediť do niektorej z tried
- objekt môže patriť do viacerých tried
- môže sa vyjadrovať, na koľko daný objekt patrí do určitej triedy
- metódy klasifikácie
 - lineárne metódy
 - sekvenčne prechádzame všetky triedy
 - porovnávame vlastnosti triedy s vlastnosťami objektu
 - vyhlasujeme, či daný objekt do triedy patrí alebo nie
 - asociácie
 - hľadáme vzťahy medzi vlastnosťami objektu a vlastnosťami triedy
 - na tento typ klasifikácie sa používajú heuristické pravidlá
 - diferenčné metódy
 - keďže triedy majú vlastnosti, potom existujú vlastnosti, ktoré ich navzájom od seba existujú
 - diferenčné vlastnosti potom slúžia na to, aby sme na základe nich vedeli rozhodnúť, do ktorej triedy daný objekt patrí alebo nepatrí
 - hierarchická klasifikácia
 - triedy sú usporiadané do stromu
 - listy sú najšpecifickejšie triedy a koreň je najvšeobecnejšia trieda
 - zistí sa, ktorá vlastnosť je splnená a podľa nej sa prehľadáva už iba podstrom, alebo podstromy
 - kombinácia svedectiev
 - každá vlastnosť je dôležitá s nejakou hodnotou
 - získame príspevky vlastností pre danú triedu
 - jednotlivé príspevky skombinujeme a dostaneme na koľko daný objekt patrí do danej triedy
 - vzdialenostná klasifikácia
 - každý objekt je bod v priestore
 - počíta sa vzdialenosť objektu od triedy
 - objekt patrí do tej triedy, ku ktorej má najmenšiu vzdialenosť

Konkrétne modely

- dané modely sú popísané pomocou modelu KADS
- jednotlivé modely sú uložené v stromovej štruktúre
 - v koreni sa táto štruktúra delí na
 - modely pre analýzu
 - modely pre syntézu
 - modely pre modifikáciu

- modely sa nenachádzajú iba v listových uzloch, ale v každom uzle stromu
- systematická diagnostika
 - je to model pre analýzu
 - diagnostika je hľadanie chyby, alebo príčiny
 - príčiny nie sú dané vymenovaním, ale systém obsahuje model systému
 - príčina stavu sa odvádza z modelu
 - inferenčná vrstva



Obrázok 35: Inferenčná sieť v systematickej diagnostike

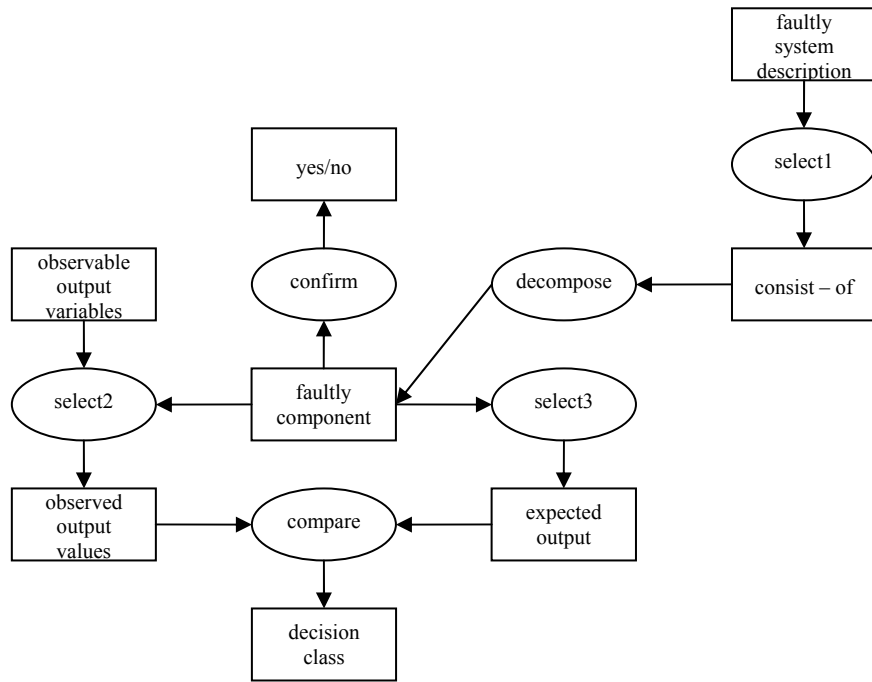
- príde sťažnosť v complaint
- select 1 vyberie vhodný systémový model
 - môžeme mať viacero systémových modelov
 - model musíme vybrať jednoznačne
- model sa dekomponuje na jednotlivé časti
- jedna z častí sa stáva hypotézou
 - snažíme sa zistiť, ktorá časť spôsobila problém
- na základe čiastkového modelu selektujeme všetky pozorovania o danej časti, ktoré nám k nej niečo povedia
- získame hodnoty premenných relevantných pre danú časť systému
- select 3 predikuje na základe hypotézy, aké by mali byť hodnoty premenných relevantných pre danú časť
- compare hodnoty porovná a vypočíta rozdiel
 - môžeme si zvoliť, či bude compare porovnávať na správnosť, alebo nesprávnosť hodnôt
 - ak compare povie, že v danej časti chyba nenastala, potom sa overuje ďalšia hypotéza
 - ak sme objavili chybnú časť, potom ju podelíme na menšie časti a vykonávame daný proces odznova
- confirm zisťuje, či sa dá ešte daná časť rozdeliť
- conclusion je záver, či sa má ešte daná časť rozdeľovať
 - ak áno, tak sa hypotéza stáva systémovým modelom
 - nemusíme vždy ísť až na atomickú úroveň
- každý inferenčný typ má aj podporné znalostné role
- úlohová vrstva

```

SG(+complaint, +possible observables, -hypothesis)
select1(+complaint, -system model)
REPEAT
    decompose(+system model, -hypothesis)
    WHILE (počet možných hypotéz > 1)
        select2(+possible observables, +hypothesis, -variable values)
        select3(+hypothesis, -norm)
        compare(+variable values, +norm, -differences)
        system model ← hypothesis
    UNTIL confirm(+hypothesis)
  
```

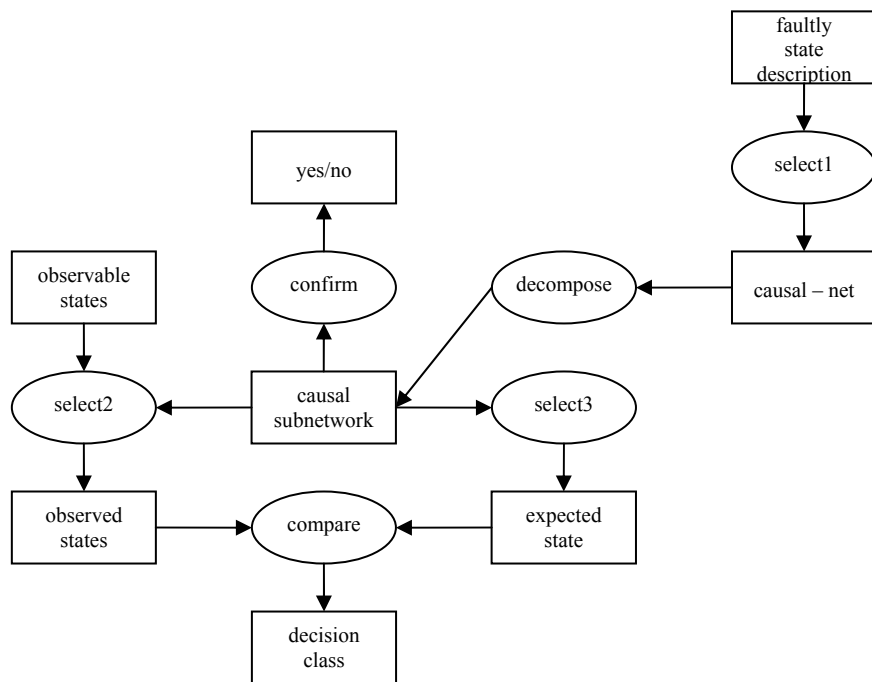
- strategická vrstva
 - nie je uvedená
 - mohla by sa týkať napríklad toho, či používať linku od hypothesis k select2
- lokalizácia
 - jeden z typov systematickej diagnostiky
 - menia sa len názvy rôl
 - ak sa zariadenie dá rozdeliť na súčiastky
 - ak hovoríme, že táto súčiastka je chybná, vytvárame vlastne priestorovú lokalizáciu
 - merania sa musia dať vykonať na každej úrovni

- o inferenčná sieť



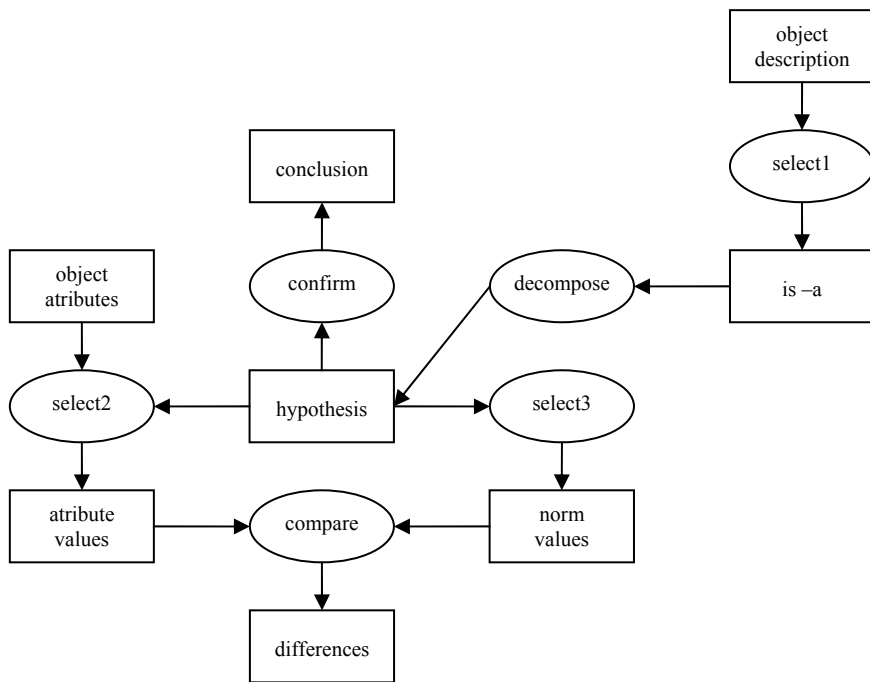
Obrázok 36: Inferenčná sieť v lokalizácii

- kauzálne trasovanie
 - o jeden z typov systematickej diagnostiky
 - o používa sa kauzálna sieť
 - každý bod kauzálnej siete predstavuje stav systému
 - systém sa nedekomponuje, ale postupuje sa v stavoch tak, aby sme vyriešili problém
 - o inferenčná sieť



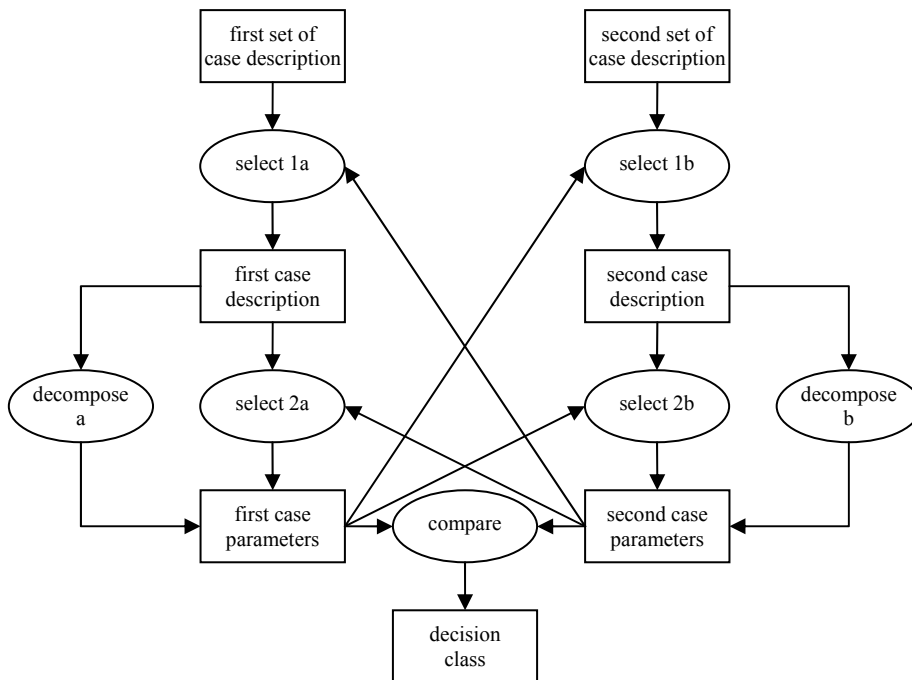
Obrázok 37: Inferenčná sieť v kauzálnom trasovaní

- systematické zjemnenie
 - o systematic refinement
 - o používa sa v heuristickej klasifikácii
 - o na reprezentáciu znalostí sa používa model typu is - a
 - triedy sa rozdeľujú na všeobecné a tie sa špecifikujú
 - o inferenčná sieť



Obrázok 38: Inferenčná sieť v systematickom zjemnení

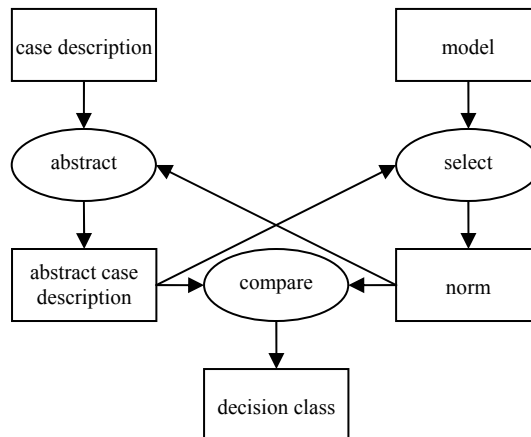
- korelácia
 - o inferenčná sieť



Obrázok 39: Inferenčná sieť v korelácií

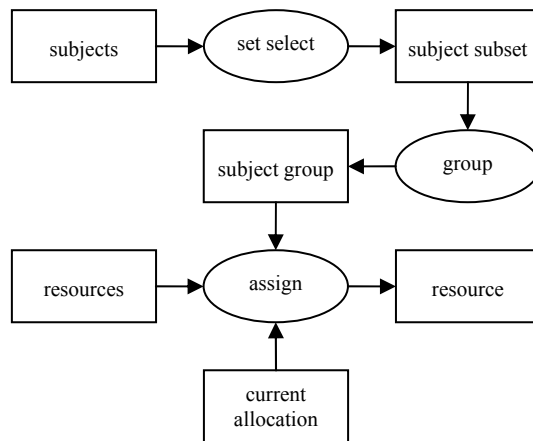
- o inferencia compare je centrálna časť systému
- o bočné časti sú postrádateľné
 - takýto model je veľmi všeobecný
 - pre konkrétnu oblasť je potrebné vždy niečo odstrániť
- o cieľom je porovnať dva javy, ktoré nemusia byť v zmysle jednoznačnej klasifikácie, ale porovnanie môže byť zložitejšie
 - napríklad na koľko sa dané javy zhodujú
 - počíta sa vzťah medzi dvomi javmi
- o decision class
 - na základe našej problematiky si definujeme výstupy
- o spôsoby porovnania
 - porovnanie objektu s modelom
 - jedna vetva predstavuje objekt
 - druhá vetva predstavuje model
 - porovnanie dvoch objektov
 - obe vetvy predstavujú objekt
- o first case description

- popis objektu, ktorý porovnávam s niečím iným
 - first case parameters
 - vybrané atribúty z popisu objektu
 - select 2a
 - ak je popis objektu daný vymenovaním, potom len selektujeme atribúty
 - riadenie cieľom
 - vyberieme čo chceme preskúmať
 - z reálneho objektu potom vyberieme atribúty, ktoré potrebujeme
 - riadenie dátami
 - vyberieme všetky atribúty na popis objektu
 - decompose
 - ak popis nie je daný vymenovaním, potom musíme popis objektu dekomponovať
 - v schéme pre konkrétny problém môže byť iba inferencia select 2a, alebo iba inferencia decompose a
 - first set of case description
 - množina popisov
 - označuje viacero popisov toho istého javu
 - ak existuje len jeden popis javu, potom schéma začína na first case description
 - úlohová vrstva
 - križne linky závisia od typu riadenia
 - dva typy riadenia
 - riadenie cieľom
 - riadenie dátami
 - strategická vrstva
 - definuje, ktorý typ riadenia použijeme
 - závisí od úlohy
- ocenenie
 - inferenčná sieť



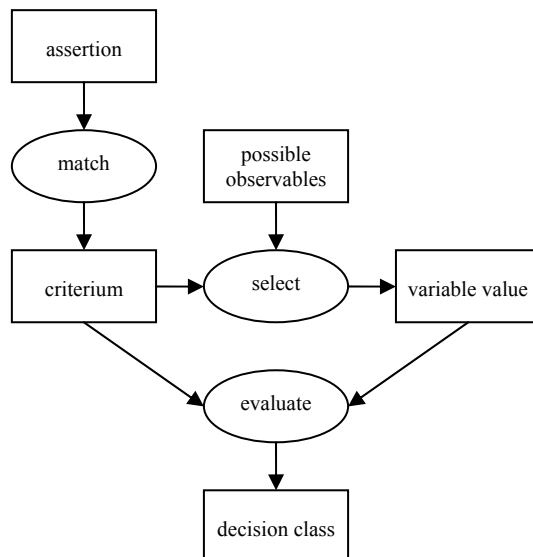
Obrázok 40: Inferenčná sieť v ocení

- anglicky: assesment
- je to špecifický tvar korelácie
- z popisu k abstraktnému popisu sa prechádza pomocou abstrakcie
- model
 - je to popis modelu
 - popis je na úrovni, s ktorou vieme priamo pracovať a teda nemusíme robiť jeho abstrakciu
- výsledok selekcie modelu je norma
- norma sa porovnáva s abstraktným popisom a dostávame výstup
- dva typy riadenia
 - riadenie cieľmi
 - riadenie dátami
- takýmto spôsobom môžeme porovnať iba model s objektom, lebo na strane modelu nemáme abstrakciu
- príklad
 - uchádzanie o zamestnanie
 - inzerát je modelom, ktorý je vlastne daný zoznamom atribútov
 - prax by mala byť najmenej päť rokov
 - uchádzač poslal dopis, b ktorom sa uchádza o miesto
 - z dopisu zistíme, akú prax mal uchádzač
 - niektorú prax mu uznáme niektorú nie
 - na základe praxe a iných údajov rozhodneme, či uchádzača prijmeme alebo nie
 - ak nie je jedna vlastnosť splnená, ešte to neznamená automatické neprijatie
- čím je viac abstraktný model, tým musí byť vyššia inferencia abstrakcie
- priradenie
 - inferenčná sieť



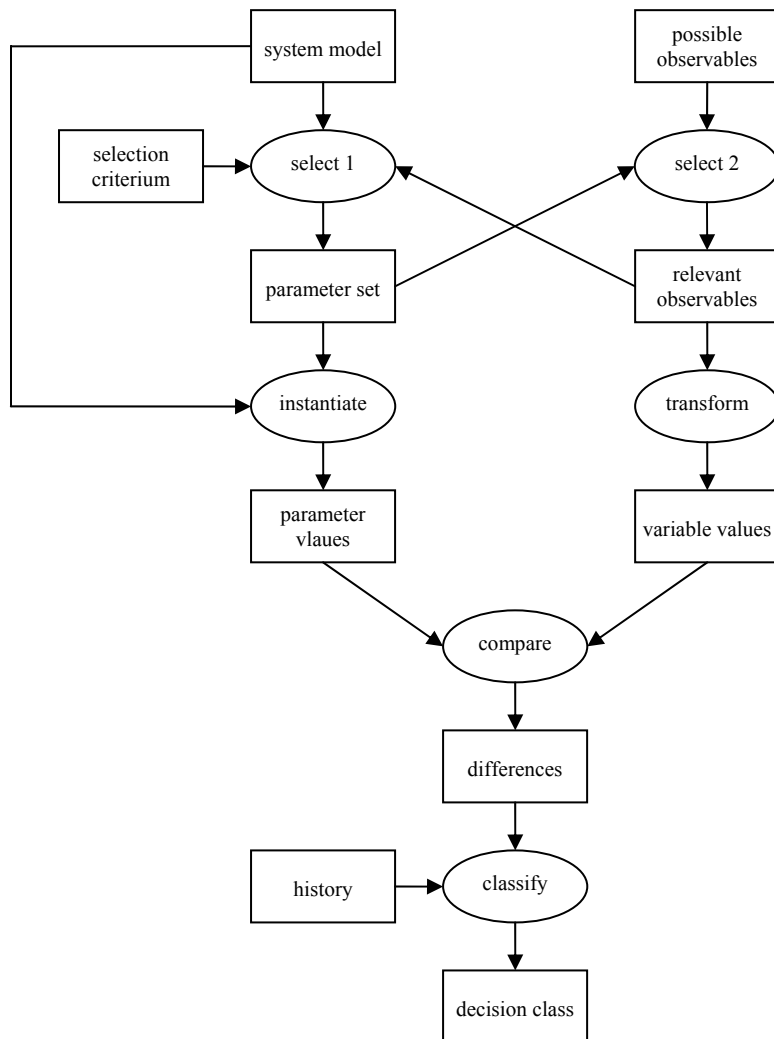
Obrázok 41: Inferenčná sieť v priradení

- je to syntetická metóda
 - skupina metód PCM
 - splnenie ohraničení
 - priradenie
- cieľom priradenia je, že chceme porovnať objekty z dvoch množín
 - 1. množina: subjekty
 - 2. množina: zdroje
- chceme priradiť zdroje subjektom tak, aby boli subjekty spokojné
 - vedie to na úlohy, kde máme napríklad rozdeľovať ľudí do miestnosti
- zo subjektov vyberieme podmnožiny
- v rámci podmnožín vytvoríme skupiny
- skupiny objektov s prostriedkami priradíme do výstupov
- current allocation
 - poskytuje nám informácie o tom, ktoré zdroje sú vyčerpané
- subjekty
 - sú všetci ľudia
 - vyberieme z nich nejakú skupinu
 - napríklad výskumníkov
 - z výskumníkov formujeme skupiny
 - napríklad na základe toho, či sú alebo nie sú fajčiari
 - resources
 - voľné miestnosti
 - current allocation
 - obsadené miestnosti
 - výstupom je konkrétna miestnosť, kam skupinky zaradiť
- systém funguje vtedy, ak sú medzi skupinkami závislosti
 - je potrebné najskôr vybrať nejakú skupinu pred inou
- zdroje sú zdieľateľné
 - preto formujeme skupinky, ktorým priradíme jednotlivé zdroje
 - formovanie skupín je možné vykonať na základe všetkých možných rozkladov
 - môžeme sa viac priblížiť k optimálnemu riešeniu
- úlohová vrstva
 - zložitosť závisí od toho ako budeme vytvárať skupinky
 - naraz
 - po častiach
 - zložitosť je závislá od backtracku
- strategická vrstva
 - neexistuje, lebo táto forma vyhodnocovania je jednoznačná
- verifikácia
 - inferenčná vrstva



Obrázok 42: Inferenčná sieť vo verifikácii

- cieľom je zistiť, či tvrdenie je pravdivé alebo nie
 - tvrdenie
 - nevieme ho hneď porovnať
 - je to len nejaké tvrdenie, ktoré nám priamo nepotvrzuje fakt
 - z tvrdenia musíme získať kritérium
 - vykonáva sa na základe zrovnania
 - príklad
 - tvrdenie
 - dopravný pás ide
 - kritérium
 - pás je na valcoch, a teda tie sa musia otáčať nejakou rýchlosťou
 - ak kritérium nie je zhodné s našim pozorovaním, potom musíme heuristicky rozhodnúť, či je kritérium splnené
 - možné pozorovania
 - zistenie otáčok valca
 - vyhodnotenie
 - či sa pás pohybuje, alebo nie
 - nedokazujeme priamo, či je splnené tvrdenie, ale či sú splnené kritériá
 - úlohová vrstva
 - možné použitie riadenia
 - dátami – z nameraných dát zistím, či je tvrdenie splnené
 - cieľom – podľa toho, čo na dokázanie tvrdenia potrebujeme, zistíme dáta
- monitorovanie
- inferenčná sieť



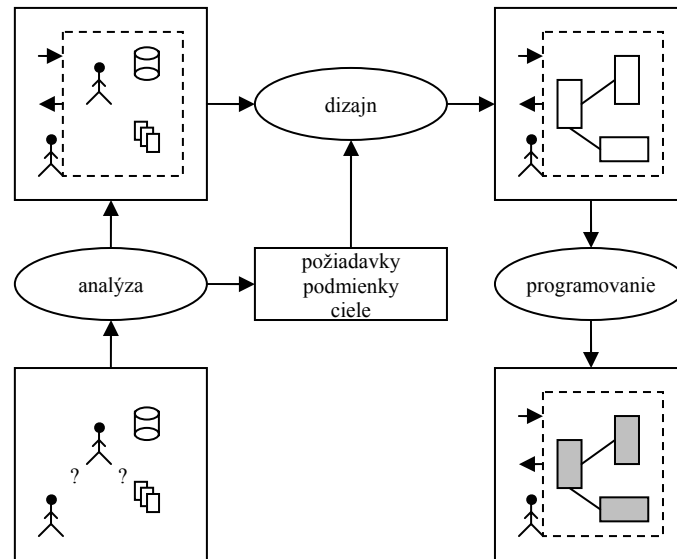
Obrázok 43: Inferenčná sieť v monitorovaní

- na jednej strane sú reálne dáta, na druhej model
- ak vznikne rozdiel medzi reálnymi dátami a tými, ktoré majú byť podľa modelu, potom rozdiel klasifikujeme
- select 2
 - zo všetkých pozorovaní vyberiem relevantné a transformujem ich na nejaké hodnoty
 - hodnoty sú z nejakých intervalov
 - môže to byť abstrakcia
- typy riadenia
 - riadenie dátami
 - riadenie cieľom
- select 1
 - z modelu vyberieme premenné
- selection criterium
 - používa sa, ak sa používa riadenie cieľmi
- instantiate
 - získanie hodnôt daných premenných
- compare
 - porovnanie nameraných hodnôt a tých, ktoré by mali byť
 - získame rozdiel
- classify
 - ak nejaký rozdiel vznikol, potom ho klasifikujeme
 - ma klasifikáciu môžu byť potrebné historické dáta
 - nie sú povinné
 - ak je diferencia malá, nemusíme ju klasifikovať
 - z klasifikácie zistíme decision class

Metodológia v KADS

- KADS – knowledge analysis and desing system
- návrh systémov sa tvorí v štyroch fázach
 - návrh systému
 - cieľ, čo chceme pomocou znalostného systému realizovať
 - vytvorenie analýzy systému
 - je to analýza daného problému a systému
 - definovanie vstupov a výstupov systému
 - definovanie interakcií v systéme

- dizajn systému
 - navrhnutie znalostného systému na báze modulov
- naprogramovanie systému
 - naprogramovanie jednotlivých modulov systému



Obrázok 44: Grafické zobrazenie metodológie návrhu znalostných systémov

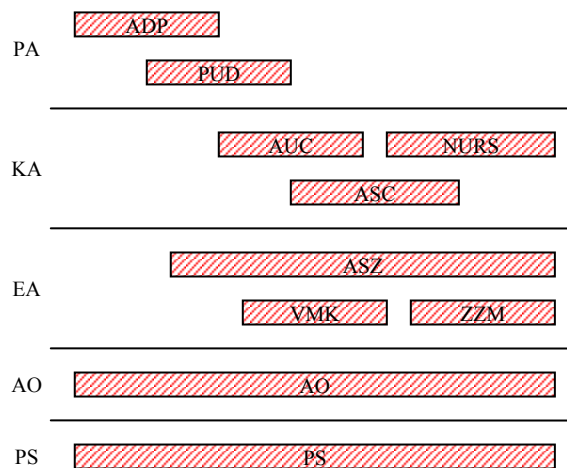
- v KADS
 - návrh systému
 - KADS predpokladá, že vieme aký systém chceme vybudovať
 - analýza systému
 - primárna oblasť záujmu KADS
 - dizajn systému
 - vedľajšia oblasť záujmu KADS
 - vykonáva ju
 - programovanie systému
 - KADS sa programovaním systému nezaobera

Analýza v KADS

- každá fáza sa delí na stupne
- stupne sa delia na aktivity
- aktivita
 - konkrétna oblasť činnosti systému
 - aktivita vždy niečo vyprodukuje
- štúdia
 - hľadanie oblasti, kde by sa dala použiť technológia znalostných systémov
 - v štúdiu sa teda oblasť pôsobenia znalostného systému čiastočne analyzuje
- vstup
 - organizačné procesy
 - sú to výstupy štúdie, ktorých rozsah je definovaný
 - myšlienka, kde by bolo možné daný systém nasadiť
- výstup
 - špecifikácie
 - definícia porozumenia organizačných procesov
 - definícia zasadenia systému do prostredia
 - dajú nám odpoveď na to, čo bude systém vykonávať
 - definujú vzťahy medzi systémom a okolím
- pozostáva z 5 stupňov
 - procesná analýza (PA)
 - cieľom je štúdium organizácie procesov z informačného hľadiska
 - informácie
 - tok informácií
 - na čo sa informácie používajú
 - začína sa na vrchnom pohľade na systém, a ten sa dekomponuje až na nejakú úroveň
 - všetkým procesom musíme priradiť vlastníkov
 - distribuovanie procesov medzi agentov
 - delí sa na dve aktivity
 - analýza a dekompozícia procesov (ADP)
 - priradenie úloh a dát (PUD)
 - výsledkom procesnej analýzy je procesný model
 - kooperačná analýza (KA)
 - vychádza z procesnej analýzy
 - využije sa distribúcia procesov medzi agentmi a definuje sa komunikácia medzi agentmi
 - delí sa na tri aktivity
 - analýza užívateľských činností (AUC)
 - komunikácia celého systému s užívateľom

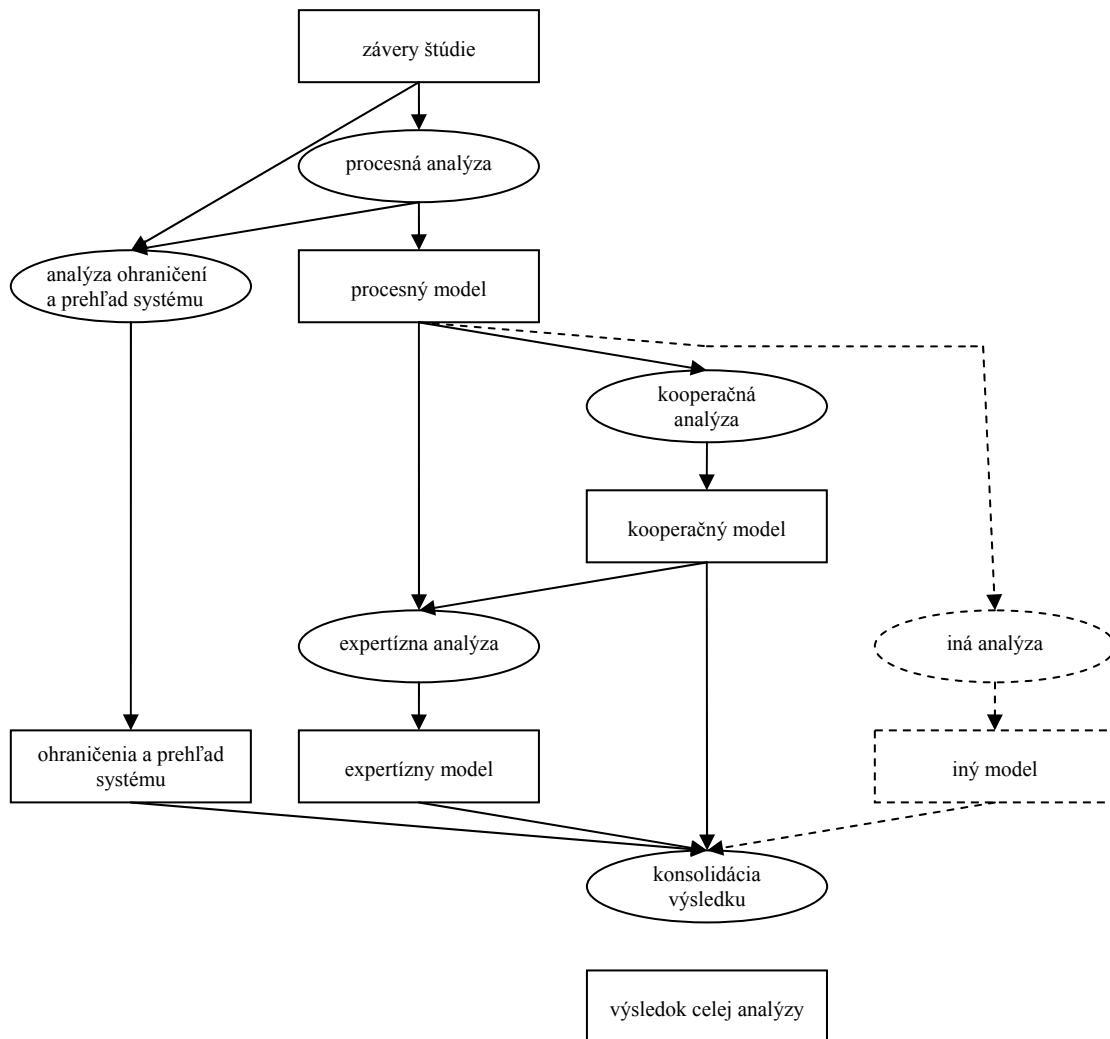
- komunikácia medzi externými agentmi a budúcim systémom
 - analýza systémových činností (ASC)
 - systém rozdelím na menších agentov
 - definujeme spoluprácu interných agentov
 - návrh užívateľského rozhrania systému (NURS)
 - výsledkom je kooperačný model
- expertízna analýza (EA)
 - zo všetkých úloh, ktorí sa v systéme vykonávajú vyberieme tie, ktoré majú charakter znalostného systému
 - vyberieme prácu agenta
 - snažíme sa modelovať túto prácu pomocou znalostného systému
 - delí sa na tri aktivity
 - analýza statických znalostí (ASZ)
 - definícia doménovej vrstvy
 - definícia znalostí potrebných pre splnenie úlohy
 - vybratie modelu z knižnice (VMK)
 - zostrojenie znalostného modelu (ZZM)
 - na základe analýzy statických znalostí a vybratia modelu z knižnice, zostrojíme konkrétny model znalostného systému
 - výsledkom je expertný model
- analýza ohraničení (AO)
 - má len jednu aktivitu
 - analýza ohraničení (AO)
 - zbieranie všetkých ohraničení, podmienok a požiadaviek na systém
 - možnosť zberu počas behu návrhu systému
 - možnosť zistenia od experta
- prehľad systému (PS)
 - má len jednu aktivitu
 - prehľad systému (PS)
 - cieľ je koordinačný
 - udržiava sa zjednodušený model systému
 - tento model slúži ako referenčný bod pre validáciu návrhov
 - v každom kroku sa vie, ako bude systém v konečnej fáze vyzerat'

- časové hľadisko analýzy systému



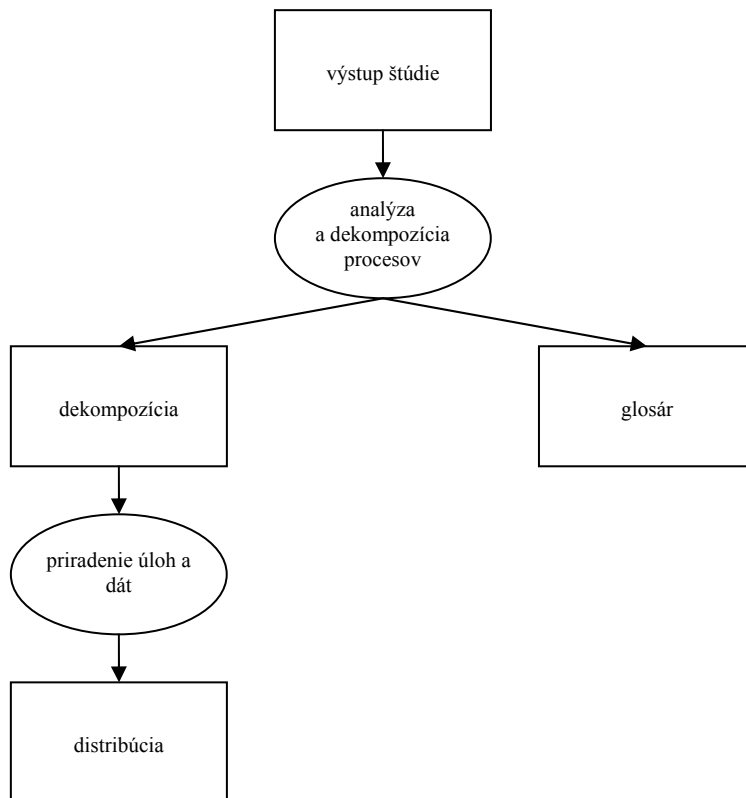
Obrázok 45: Časové hľadisko analýzy systému

- celkový pohľad na analýzu



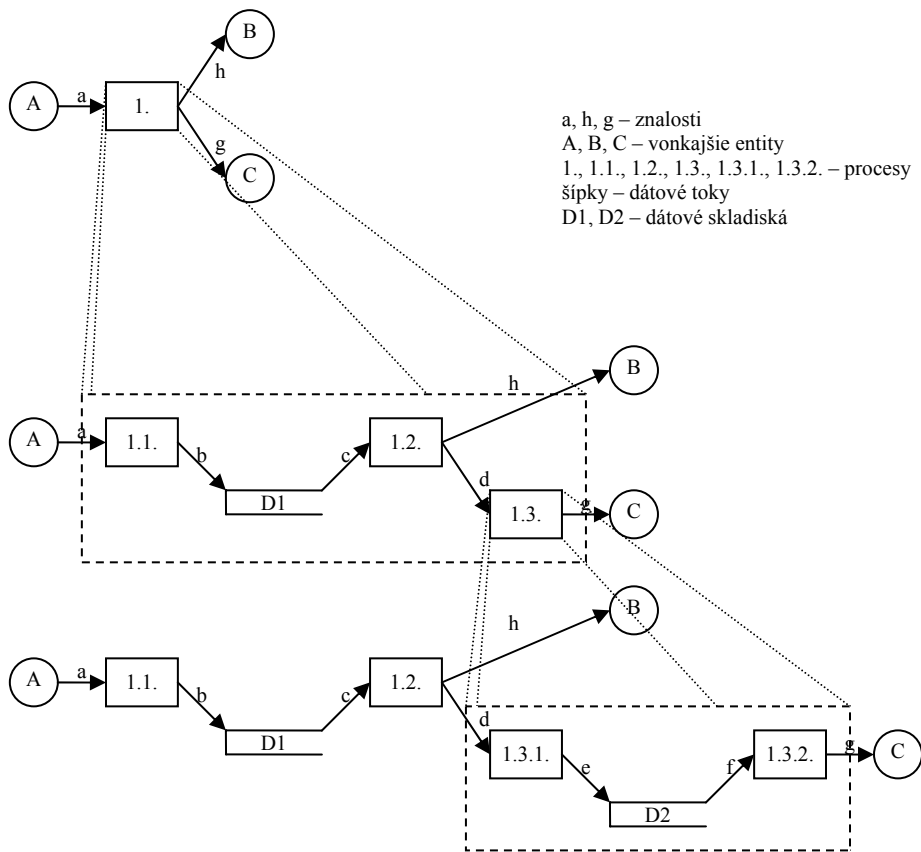
Obrázok 46: Celkový pohľad na analýzu v KADS

- model nemusí obsahovať iba znalostné modely
 - napríklad databázový model
 - ostatné modely sa vyznačujú prerušovane
 - iná analýza vygeneruje iný model
- procesná analýza
 - je to analýza a dekompozícia procesov
 - sú dva prístupy
 - analýza skutočného stavu
 - analyzuje sa daný systém tak ako je
 - analýza zmeneného stavu
 - systém sa umiestni do skutočného prostredia a analyzuje sa
 - väčšinou sa vykonáva iba analýza zmeneného stavu
 - vykonávať obe analýzy je časovo a finančne náročné
 - celý proces sa rozbieha na menšie procesy
 - rozbieha sa až na atomickú úroveň
 - agent musí vedieť daný proces realizovať
 - výstupom
 - dekompozícia
 - obrázky dekompozície
 - glosár
 - definícia všetkých typov dát
 - analýza a dekompozícia procesov
 - je stanovenie vlastníctva procesov
 - definovanie, kto je zodpovedný za prijímanie a spracovávanie dát
 - priradenie úloh a dát
 - rozdelenie procesov medzi agentov
 - výstupom je distribúcia, ktorá je doplnkom dekompozície



Obrázok 47: Schéma procesnej analýzy

- výstup procesnej analýzy
 - dekompozícia
 - distribúcia
 - glosár
- dekompozícia
 - nedá sa presne povedať, kedy skončiť dekompozíciu
 - zakresľuje sa pomocou diagramov dátových tokov
 - vyvíjali sa ešte pre obdobím objektového programovania
 - graficky znázorňujú závislosti
 - sú dostatočne jednoduché, aby ich pochopil aj užívateľ
 - definujú
 - proces – označujú sa štvorčekom
 - dátový tok – označujú sa šípkami
 - dátové skladiská – označujú sa neuzavretými obdĺžnikmi
 - na najvrchnejšej úrovni je zakreslený celý systém ako jeden proces
 - daný proces sa rozkladá na menšie procesy
 - dohoda o označovaní procesov
 - procesy sa značia číslami
 - ak sa proces x rozkladá na procesy, potom podprocesy sú označené x.1., x.2., ...
 - príklad



Obrázok 48: Príklad dekompozície

- glosár
 - popisuje
 - každý proces
 - každý dátový tok
 - každé dátové skladisko
 - popis je špecifikovaný v nejakej forme, napríklad BNF

a := b	a je definované ako b
A B	A alebo B
A + B	A a B
A*	ľubovoľný výskyt A
A ⁺	výskyt A minimálne raz
...	

- príklad

meno := titul |* + krstne + priezvisko
 titul := MUDr. | Ing. | ...
 krstne := veľké + malé⁺
 ...

- ak nevieme rozhodnúť komu daný proces patrí
 - zvolenie jedného vlastníka a vygenerovanie ohraňení
 - vytvoriť dekompozíciu procesu

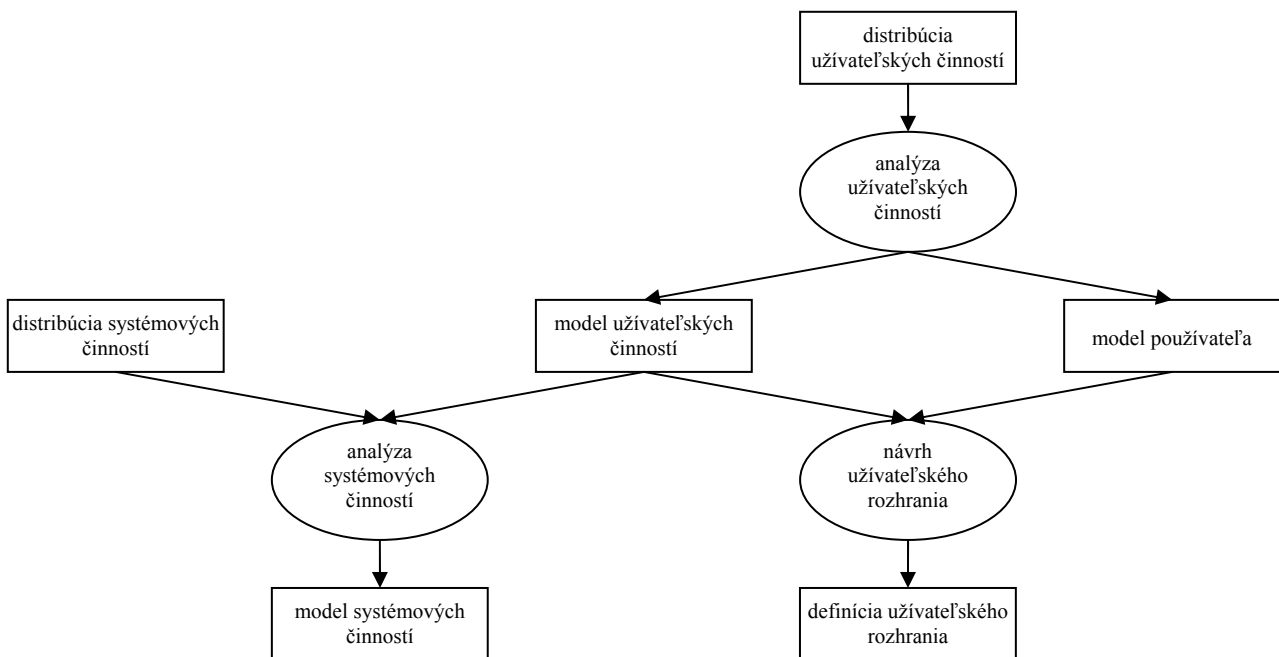
○ distribúcia

- v tejto časti hovoríme, ktorý proces komu patrí
- proces musí mať vlastníka
- dátový tok nemá vlastníka
- dátové skladisko musí mať vlastníka
- možnosti vykonania distribúcie
 - označiť v dekompozícii farbami vlastníkov
 - každý agent bude mať priradenú inú farbu
 - jeden agent môže vlastniť i viac procesov
 - vyznačením oblastí
 - nie je veľmi vhodné, ak sa oblasti križujú
 - označenie identifikátorom
 - najčastejšie sa používa pre počítačové spracovanie
 - nemá takú výpovednú hodnotu ako vyznačenie farbami

- kooperačná analýza

- definuje komunikáciu medzi agentmi
- výsledkom je návrh rozhraní, ktoré sa budú používať v dizajne
- delí sa na

- analýza užívateľských činností
 - je to popis rozhrania medzi systémom a vonkajšími agentmi
 - vonkajší agent je vo väčšine prípadov užívateľ
 - môže to byť aj softwarový agent (databáza)
 - skúmame činnosť vonkajšieho agenta vzhľadom na ostatných agentov (systém)
 - zaujímame sa o
 - stavy v ktorých sa vonkajší agent nachádza
 - udalosti generované v stavoch
 - zmenu stavu
 - systém v tejto časti vystupuje ako nedeliteľný celok
 - vytvára povinne model užívateľských činností a nepovinne model užívateľa
 - analýza systémových činností
 - systém delíme na vnútorných agentov
 - analyzujeme to, ako agenti musia spolu kooperovať, aby splnili požiadavky na daný systém ako celok
 - definovanie užívateľského rozhrania
 - vychádza z analýzy užívateľských činností
 - v tejto fáze sa navrhne aj to, ako bude rozhranie navonok vyzerať
- výstupom týchto troch činností je kooperačný model
- model práce kooperačnej analýzy



Obrázok 49: Model práce kooperačnej analýzy

- analýza užívateľských činností
 - potrebujeme zistiť, aké dáta tečú, kedy tečú, závislosť medzi dátami a riadiace toky
 - vzorové komunikácie
 - schéma pre riešenie úloh

```

WHILE NOT exit DO
  CASE USER event
    exit:
    request help:
      SYSTEM provide help
    specify problem:
      UNTIL solved DO
        SYSTEM provide question
        USER provide answer
      LOOP
      SYSTEM provide solution
  LOOP

```

- ak príde od užívateľa udalosť
 - exit – cyklus v ďalšom cykle skončí, a tým aj celý program
 - request help – systém poskytne užívateľovi pomoc
 - specify problem – systém ide riešiť problém
 - systém užívateľovi kladie otázky
 - na základe odpovedí užívateľa mu poskytne riešenie
- schéma pre samoučiaci sa systém

```

WHILE NOT exit DO
  CASE USER event
    exit:

```



```

request help:
    SYSTEM provide help
specify problem:
    UNTIL solved DO
        SYSTEM provide question
        USER provide answer
    LOOP
    SYSTEM provide solution
    IF solution wrong
    THEN
        USER provide correction
        SYSTEM update knowledge base

```

LOOP

- ak systém poskytol špatné riešenie, potom sa prevedie korektúra jeho bázy znalostí
- schéma pre výukový systém

WHILE NOT exit DO

CASE USER event

exit:

request help:

SYSTEM provide help

request problem:

SYSTEM provide problem

UNTIL solved OR abort DO

CASE USER event:

abort:

ask question:

SYSTEM provide answer

specify solution:

IF correct

THEN

SYSTEM acknowledge

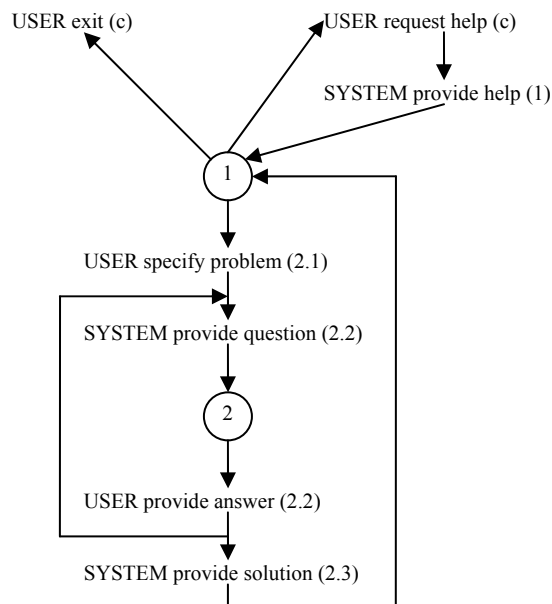
ELSE

SYSTEM explain solution

LOOP

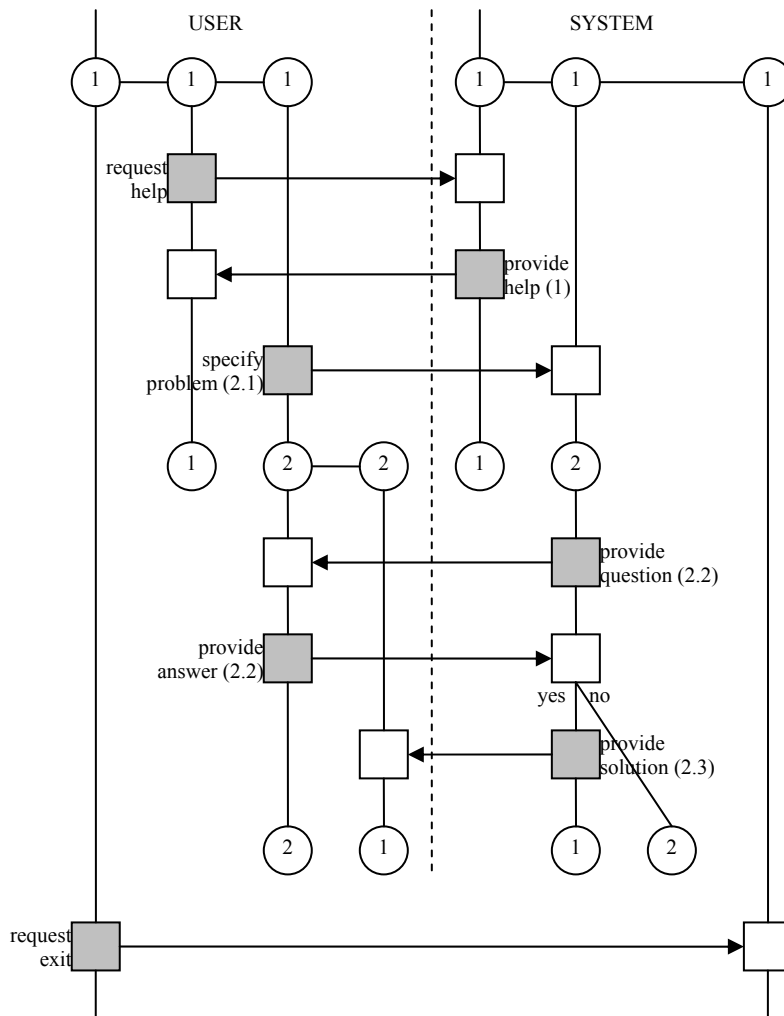
LOOP

- systém zadá problém užívateľovi
- užívateľ sa snaží riešiť daný problém
- systém užívateľa nakoniec vyhodnotí
- ak užívateľ potrebuje ďalšie informácie, tak mu ich systém poskytne
- užívateľ poskytne riešenie
 - ak je správne, systém ho pochváli
 - ak je nesprávne, vysvetlí mu dané riešenie
- diagramy prenosov stavov
 - len popis analýzy užívateľských činností je nedostatočný
 - musíme definovať
 - užívateľské stavy
 - užívateľské možné akcie v každom stave
 - spúšťajú akcie v systéme
 - informačné toky medzi užívateľom a systémom
 - vrátane toho aká je to informácia a kto ju riadi
 - možné akcie systému v každom stave
 - sú spustené užívateľovými akciami
 - môžu vyvolať i ďalšie užívateľove stavy
 - prechody medzi jednotlivými stavmi
 - krížová väzba na procesný model
 - ak máme software, tak ním vieme skontrolovať tento model s procesným modelom
- príklad



Obrázok 50: Príklad diagramu prenosu stavov pre analýzu užívateľských činností

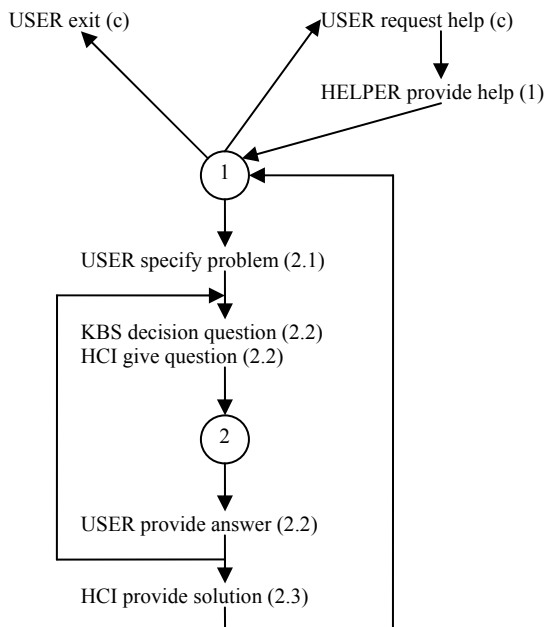
- 1 a 2 označujú stavy užívateľa
 - stav 1 označuje čakanie na užívateľovu hlavnú akciu
 - stav 2 označuje stav, keď užívateľ zadal otázku a systém ju rieši
 - užívateľ v stave 1 môže požiadať o pomoc, skončiť program, alebo položiť otázku
 - ak užívateľ položí otázku
 - vykonáva sa slučka, v ktorej sa počíta výstup
 - systém sa môže pýtať ďalšie otázky
 - ak systém poskytne odpoveď vracia sa užívateľ do stavu 1
- riadiace toky sa označujú (c)
- neriadiace toky sa označujú číslom
- proces sa označuje číslom
- role activity diagram
 - príklad



Obrázok 51: Príklad role activity diagram

- každý agent je zakreslený zvlášť
- každý stav musí byť definovaný aj v USER aj v SYSTEM časti
 - ak by nebol, potom by to znamenalo, že agent komunikuje sám so sebou
- udalosti sú značené šedými štvorcami
- čakanie na odpoveď je značené bielymi štvorcami
 - z obrázku vidíme, že užívateľ môže reagovať tromi udalosťami
- každá transakcia je značená
 - šipkou – označuje sa ňou smer toku dát
 - zafarbenie – označuje sa ním z koho popudu transakcia vyšla
- slučka sa realizuje prechodom z rovnakého stavu do rovnakého stavu
- sekvencia sa realizuje prechodom z jedného stavu do druhého stavu
- ak by boli v systéme i iní agenti (napríklad databáza), potom analogicky vykonáme model spolupráce s týmto agentom
- model používateľa
 - ak by mal systém pracovať s viacerými používateľmi, potom je nutné vytvoriť modely používateľov
 - komunikácia musí byť prispôbená každému používateľovi
 - obsahuje
 - znalosti používateľa
 - ako vie dať znalosti používateľ využívať
 - ako užívateľ komunikuje so systémom
 - je to len textový popis
 - má tri časti
 - charakteristiky užívateľa
 - intelektuálne
 - fyzické (kvôli hendikepom)
 - emočné
 - popis činnosti práce užívateľa
 - popis práce
 - na čo bude výsledky využívať
 - kompetencie
 - kontext práce používateľa
 - pracovné miesto používateľa (hardwarová konfigurácia)
- analýza systémových činností
 - to isté ako analýza užívateľských činností
 - systém v tomto prípade berieme ako súhrn vnútorných agentov

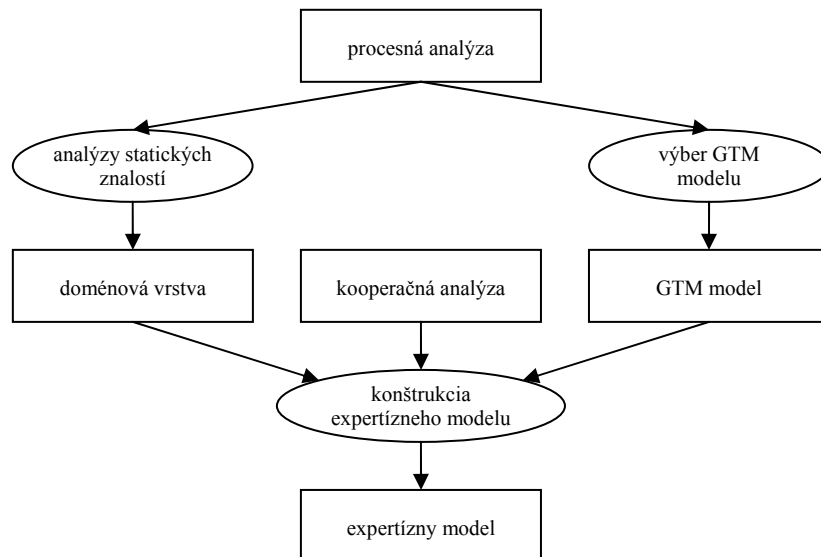
- vychádza z komunikácie medzi užívateľom a systémom s tým, že systém nahradíme vnútornými agentmi
 - tam kde sa vykonávala činnosť systémom, nahradíme činnosť agentom
 - tam kde sa vykonávala činnosť systémom, nahradíme činnosť viacerými agentmi
- je na návrhárovi ako vytvorí vnútorných agentov
- niektorý bežne používaný agenti
 - znalostný agent (využíva sa stále)
 - agent pre rozhranie
 - agent pre úložiská
- mení sa vlastne obrázok diagramu prenosu stavov



Obrázok 52: Príklad diagramu prenosu stavov pre analýzu systémových činností

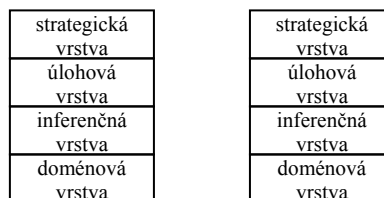
- HELPER – agent poskytujúci pomoc
- KBS – agent zodpovedný za prácu so znalosťami
- HCI – agent zodpovedný za rozhranie medzi užívateľom a systémom
- SYSTEM nahradíme za agenta alebo viacerých agentov
- môžeme zmeniť USER za HCI
- môžeme pridávať i ďalších agentov
- systém môže vykonávať viacero operácií za sebou
 - je možné v tomto prípade rozdeliť systém na viac stavov
- vychádzame z toho čo už máme
 - nahradíme SYSTEM za agentov
 - nahradíme USER za komunikačných agentov
 - môžeme rozšíriť systém o viac agentov
- definícia užívateľského rozhrania
 - zoberieme výstup analýzy užívateľských činností, model užívateľa (ak bol vytváraný) a ohraničenia týkajúce sa komunikácie a začneme navrhovať rozhranie
 - KADS nedáva presný návod ako postupovať pri definícii užívateľského rozhrania
 - zvolíme spôsob komunikácie
 - formy komunikácie
 - textová komunikácia
 - komunikácia pomocou formulárov
 - komunikácia výberom z menu
 - grafická komunikácia
 - o tom, ktorý typ komunikácie vybrať rozhoduje
 - samotná aplikácia
 - teda aké dáta je potrebné prenášať
 - ak je prenášaných dát málo, potom je vhodné zvoliť textovú komunikáciu
 - podobné pravidlá
 - užívateľ
 - užívateľ má určité schopnosti
 - užívateľ je schopný s niečím pracovať
 - komunikáciu teda musíme prispôbiť užívateľovi
 - ohraničenia
 - organizácie majú už zabehaný štandard práce, ktorý nemôžeme systémom narušiť
- navrhujeme dátové toky
 - každý dátový tok namapujeme na nejakú štruktúru, teda ju reprezentujeme
- zoskupíme komunikačné prvky do skupín
 - zoskupujeme riadiace prvky do skupín
 - ak prvky spolu nejako súvisia, je dobré ich zoskupiť do nejakého celku

- prvky začneme rozmiestňovať na obrazovku
 - začiatok návrhu dizajnu
 - nadefinujeme akcie na ovládanie prvkami
 - zostrojenie prototypu rozhrania
 - typy prototypov
 - náčrty na papier
 - plne funkčné rozhranie
 - existujú obrazovky komunikácie, ale ešte neexistuje vnútorná realizácia znalostného systému
 - prototyp ukážeme užívateľovi
 - užívateľ nám poskytne reakciu, či sa mu návrh komunikácie páči
 - hodnotenie prototypu
 - subjektívne
 - ohodnotí užívateľ
 - objektívne
 - sledovanie, ako by užívateľ pracoval s daným rozhraním
 - z tohto vypracovať analýzu a tak zmeniť užívateľské rozhranie
 - vypracovanie finálnych záverov
 - vytvára výstup tejto aktivity
- expertízna analýza
- je to začiatok návrhu znalostnej časti
 - analyzujeme znalostný proces
 - má tri aktivity
 - analýza statických znalostí
 - vytvorenie doménovej vrstvy
 - výber GTM modelu
 - vybratie vytvoreného modelu GTM z knižnice a jeho úprava
 - vytvorenie vlastného modelu, s možným využitím určitých častí z GTM knižnice
 - návrh expertízneho modelu
 - vychádza z procesnej a kooperačnej analýzy



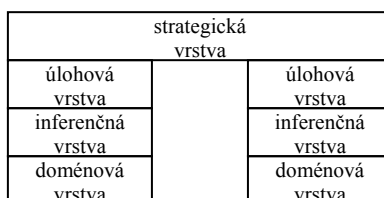
Obrázok 53: Schéma práce expertíznej analýzy

- ak riešime úlohu tak na jej riešení sa môže podieľať viac systémov
 - môže sa teda stať, že úlohu budú riešiť i viaceré znalostné systémy



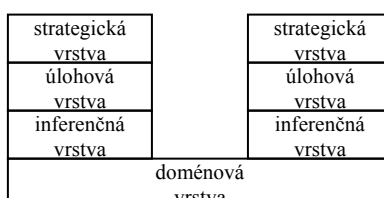
Obrázok 54: Riešenie problému viacerými znalostnými systémami

- spôsoby integrácie znalostných systémov
 - snahou je teda tieto expertné systémy spojiť, keďže majú riešiť ten istý problém
 - spojenie na strategickej úrovni



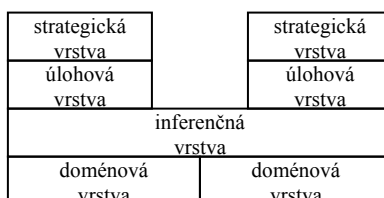
Obrázok 55: Spojenie znalostných systémov na strategickej vrstve

- v strategických vrstvách jednotlivých znalostných systémov boli rôzne strategické znalosti
- tieto znalosti spojíme a pridáme znalosť, ktorá bude rozhodovať, kedy sa ktorá časť použije
- je to najjednoduchšia metóda
- spojenie na doménovej úrovni



Obrázok 56: Spojenie znalostných systémov na doménovej vrstve

- používa sa, ak výsledná implementácia predpokladá, že doménové vrstvy budú spolu na jednom fyzickom mieste
- používa sa, ak existuje spojenie medzi doménovými vrstvami
- spojenie na inferenčnej úrovni



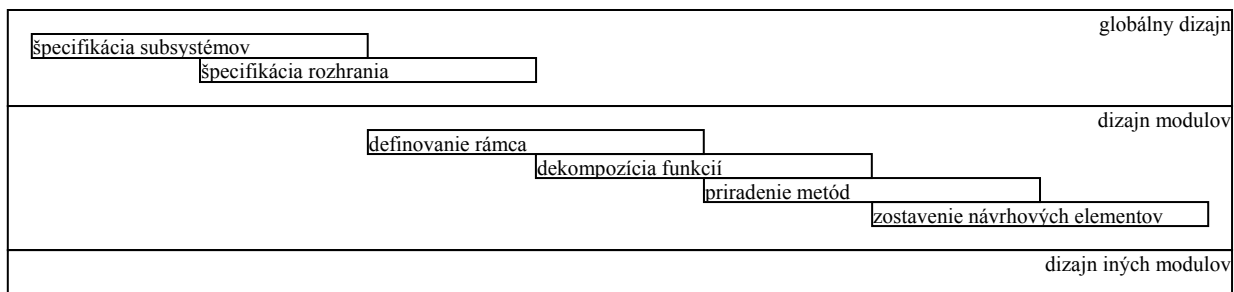
Obrázok 57: Spojenie znalostných systémov na inferenčnej vrstve

- používa sa vtedy, ak majú znalostné systémy určité časti na inferenčnej vrstve rovnaké
 - spoločné body sú najčastejšie pri znalostných rolách
 - ak majú znalostné roly rovnaké, potom je spojenie jednoduché
 - ak nie sú znalostné roly celkom rovnaké, tak sa ich musíme pokúsiť nejako zjednotiť
 - rozdelenie jednej znalostnej roly na dve, pričom jedna z nich je znalostná rola z druhého systému
 - spojenie znalostných rôľ cez inferenciu
 - v tomto prípade sa však z časti spoja aj doménové vrstvy
- analýza ohraničení
- má len jednu priebežnú aktivitu
 - počas celej analýzy v pozadí beží analýza ohraničení
 - ak v analýze narazíme na ohraničenie, tak ho pošleme do analýzy ohraničení, kde sa spracuje
 - výsledkom je udržiavanie platného zoznamu ohraničené, ktoré by mali byť splnené
 - ohraničenia musia byť splnené počas dizajnu systému
 - ohraničenia, sú akékoľvek ohraničenia, okrem ohraničené na funkčnosť systému, teda na to ako má systém fungovať
 - typy ohraničené
 - environmentálne ohraničenia
 - ohraničenia na okolie v ktorom bude systém zasadený
 - vychádzajú z toho, že
 - už existujú nejaké klientské rozhrania
 - užívatelia sú s týmito rozhraniami naučení pracovať
 - užívatelia majú svoj štýl práce
 - podmienky pre inštaláciu a pod.
 - technické ohraničenia
 - všetky ohraničenia z technického hľadiska, ktoré vzniknú počas analýzy
 - rozhranie musí byť nejako konkrétnejšie navrhnuté
 - ohraničenie na to, na akom stroji bude náš systém fungovať
 - predpisy
 - ohraničenia na to, ako možno daný systém do spoločnosti nasadiť
- prehľad systému
- má len jednu priebežnú aktivitu
 - počas celej analýzy v pozadí beží prehľad systému
 - cieľom je budovať stručnú predstavu o výslednom systéme
 - táto predstava slúži ako referenčný bod
 - obsahuje všetky ohraničenia na funkcie systému
 - všetky ciele, ktoré by mal systém spĺňať
 - priebežná predstava o štruktúre systému
 - priebežná predstava o informačných požiadavkách systému

- požiadavky na zabezpečenie patričných znalostí

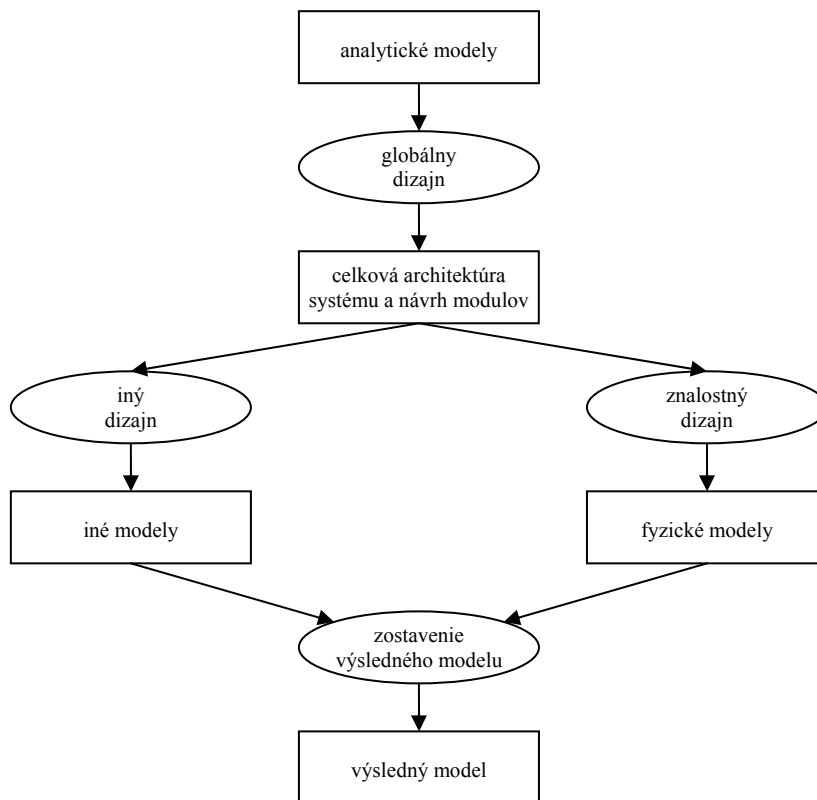
Dizajn

- cieľom je zobrať výsledky analýzy a pretransformovať ich do podoby vhodnej pre realizáciu
- má dve hlavné fázy
 - o globálny dizajn
 - vytvorenie celkovú architektúru systému
 - rozdeliť systém na moduly
 - delí sa na
 - špecifikácia subsystémov
 - špecifikácia rozhrania
 - o dizajn modulov
 - zostrojenie konkrétneho znalostného modulu
 - návrh ostatných modulov, iných ako znalostných nie je riešený v KADS
 - delí sa na
 - dekompozícia funkcií
 - o výsledkom je funkčný modul
 - priradenie metód
 - o realizácia funkcií pomocou metód
 - o výsledkom je behaviorálny model
 - zostavenie návrhových elementov
 - o spojenie funkcií dohromady
 - o výsledkom je fyzický model
 - definovanie rámca
 - všetky tri metódy pracujú v nejakom rámci, ktorý musíme definovať



Obrázok 58: Rozdelenie dizajnu

- schéma práce dizajnu



Obrázok 59: Schéma práce dizajnu

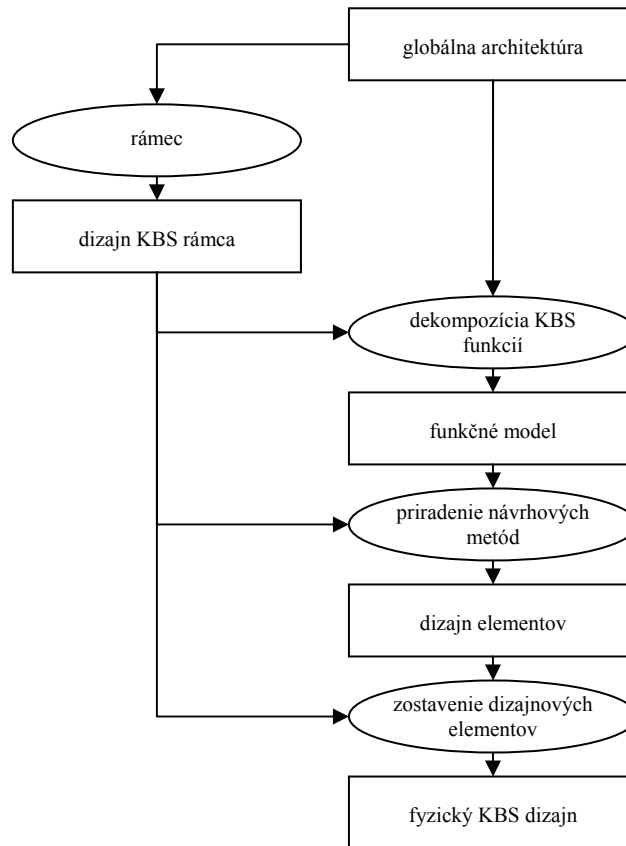
- globálny dizajn
 - o špecifikácia subsystémov

- pracuje na úlohe funkčnosti systému
 - jeho úlohou je zobrať všetky funkcie systému a distribuovať ich do modulov
 - rozdelenie systému do modelov definujúcich architektúru systému
- špecifikácia rozhrania
 - definujeme rozhranie modulov
 - dostali sme vlastne čierne skrinky, ktoré potrebujeme navrhnuť
- vychádzame z procesnej analýzy, ktorá hovorí čo ktorý proces vykonáva
- vychádzame z komunikačnej analýzy, ktorá hovorí ako agenti medzi sebou komunikujú
- vychádzame z celkového prehľadu systému, ktorý obsahuje predbežnú štruktúru systému
- dva spôsoby globálne dizajnu
 - zdola – nahor
 - vychádzame z funkcií, ktoré systém má mať
 - podobné funkcie zhlukujeme
 - vznik modulov, ktoré sú definované skupinami funkcií
 - zhora – nadol
 - vychádzame z toho, že predbežne stanovíme aké moduly budú v systéme
 - funkcie sa pokúsime do týchto modulov rozdeliť
 - funkcie teda zhlukujeme podľa cieľa
- závislosti
 - keďže sú funkcie systému medzi sebou závislé, tak vzniká aj závislosť
 - závislosť vo vnútri modulu
 - závislosť medzi modulmi
 - snažíme sa maximalizovať závislosť vo vnútri modulu a minimalizovať medzi modulmi
- faktor riadenia
 - hovorí o tom, ako bude v systéme realizované riadenie
 - delenie riadené
 - centralizované
 - existencia špecifického modulu pre účely riadenia
 - tento modul riadi činnosť ostatných modulov
 - decentralizované
 - riadenie je rozosiata medzi moduly
 - typ riadenia je závislý od
 - bohatosti strategickej vrstvy
 - ak nie je zložitá – decentralizované riadenie
 - ak je zložitá – centralizované riadenie
 - ohraničení
 - nástroj v ktorom systém vyvíjame môže mať už zabudovaný určitý typ riadenia
 - užívateľského rozhrania
 - systém nejako komunikuje s užívateľom
 - všetky funkcie, ktoré slúžia na komunikáciu sa vyčlenia do jedného modulu
 - existujú nástroje pre návrh užívateľského rozhrania
- označenie tokov – šípkami
 - čiarkovaná – dátové toky
 - plná – riadiace toky
- definovanie rozhrania
 - ak definujeme ako subsystémy medzi sebou komunikujú, potom môžeme jednotlivé subsystémy vybrať jednotlivo a jednotlivo ich popisovať
 - každý modul navrhne potom vlastnou metodikou
 - pre každú atomickú štruktúru sa vytvára tabuľka, ktorá obsahuje
 - iniciator – z koho popudu vznikla komunikácia
 - request – popis toho o čom v danom rozhraní ide, typ aktivity
 - sources – zdrojový agenti
 - destinations – cieľový agenti
 - informations – definuje dátový obsah, môže byť aj prázdny
 - recipient – definuje kto získava riadenie po skončení komunikácie
 - result – očakávaný výsledok, popis aký bude stav po dokončené transakcie

- KBS dizajn

- v tejto časti zoberieme samotný model a ten navrhujeme
- sú to vlastne znalostný agenti
- výsledkom je špecifikácia agenta
- má dve hlavné aktivity
 - definícia rámca
 - definujeme rámec, v ktorom budeme modul vyvíjať
 - rámec nám definuje, v čom budeme programovať a tento programovací balík nám poskytuje možnosti pre konkrétnu implementáciu
 - definovanie implementačnej platformy
 - berie do úvahy všetky ohraničenia z analýzy
 - berie do úvahy všetky ohraničenia z prostredia do ktorého bude systém zasadený
 - je to slovný popis, čo musíme dodržať
 - dizajn modulu
 - obsahuje zoznam funkcií, ktoré má realizovať
 - má tri časti
 - funkcie dekomponujeme na menšie aktivity
 - rozhodneme sa ako tieto aktivity budeme implementovať
 - dizajn elementov
 - dizajnové elementy zoskupíme do modulov
 - modul sa vlastne rozdelí na podmoduly a tie sa skladajú z elementov

- o schéma práce KBS dizajnu



Obrázok 60: Schéma práce KBS dizajnu

- o dekompozícia KBS funkcií
 - výsledkom je funkčný model
 - zoberie všetky funkcie a rozkladá ich na jednoduchšie
 - výsledkom je hierarchia funkcií
 - strom, ktorý má koreň funkciu, ktorú chceme implementovať
 - KADS má hierarchiu funkcií
 - v hierarchii sú použité 4 typy funkcií
 - o riešenie úloh
 - delia sa na
 - inferenčné funkcie
 - kompozitné funkcie
 - o úložiskové funkcie
 - delia sa na
 - doménový model
 - dátový model
 - o strategické riadenie
 - o podporné funkcie
 - riešenie úloh
 - o majú ekvivalent v inferenčnej vrstve
 - o je to funkcia, ktorá realizuje expertné riešenie
 - o manipulujú so znalosťami
 - o inferenčná funkcia zodpovedá práve jednej inferencii
 - o kompozitná funkcia zahŕňa niekoľko inferencií
 - sekvencia inferencií
 - úložiskové funkcie
 - o sú reflexiou na doménovú vrstvu
 - o funkcie, ktoré nám umožňujú prístup k znalostiam a manipuláciu s nimi
 - o nie je to samostatná znalosť, ktorú máme uloženú, len funkcia na jej sprístupnenie
 - o doménový model obsahuje statické znalostné roly
 - o dátový model obsahuje dáta, ktoré vznikajú za behu znalostného systému
 - strategické riadenie
 - o sú reflexiou strategickkej vrstvy
 - o zvyčajne nie sú definované
 - o definujú sa ak je strategická vrstva zložitejšia
 - podporné funkcie
 - o všetky ostatné funkcie
 - každej funkcii musíme
 - stanoviť jej typ
 - definovať návaznosť – výstup jednej funkcie je vstupom druhej

- implementovať
- na definovanie typu funkcie je špeciálny rámec
 - existujú štyri typy rámcov
 - rámec pre inferenčné funkcie
 - rámec pre kompozitné funkcie
 - rámec pre úložiskové funkcie
 - rámec pre strategické a podporné funkcie
 - rámec pre inferenčné funkcie

INFERENCE FUNCTION názov

GOAL: popis cieľa
 INPUT DOMAIN ROLE: vstupná znalostná rola
 OUTPUT DOMAIN ROLE: výstupná znalostná rola
 ACTIVITES: koho aktivuje
 ACTIVITE BY: kým bola aktivovaná
 SUPPORTS: smerník na znalostnú rolu, ktorá definuje čo vlastne funkcia vykonáva
 SUPPORTED BY:
 REALIZE BY: zostáva prázdne
 SIDE EFFECTS:
 COMMENTS:

- rámec pre kompozitné funkcie

COMPOSITE FUNCTION názov

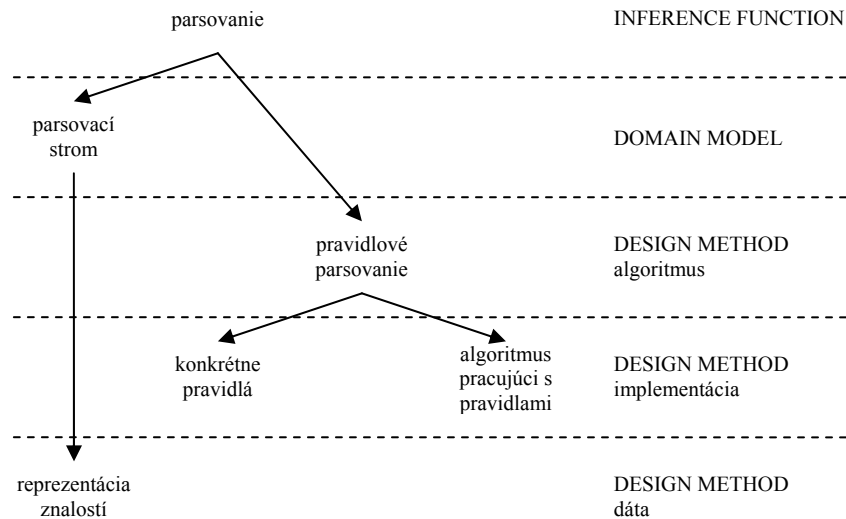
GOAL: popis cieľa
 INPUT DOMAIN ROLE: vstupná znalostná rola
 OUTPUT DOMAIN ROLE: výstupná znalostná rola
 ACTIVITES: koho aktivuje
 ACTIVITE BY: kým bola aktivovaná
 SUPPORTS: smerník na znalostnú rolu, ktorá definuje čo vlastne funkcia vykonáva
 SUPPORTED BY:
 CONTROL STRUCTURE: popis sekvencie komponentov
 REALIZE BY: zostáva prázdne
 SIDE EFFECTS:
 COMMENTS:

- rámec pre doménový model

DOMAIN MODEL názov

USE: popis čo vykonáva
 COMPOSITION: štruktúra znalostí, ktoré sú v danej funkcii
 CREATED BY: funkčné bloky, ktoré používa tento model
 USED BY: funkčné bloky, ktoré používa tento model
 ACCESSED BY: funkčné bloky, ktoré používa tento model
 DESTROYED BY: funkčné bloky, ktoré používa tento model
 REPRESENTATION: zostáva prázdne

- priradenie návrhových metód
 - vychádza z dekompozície funkcií
 - v tejto časti chceme povedať, ako bude funkcia realizovaná
 - funkčný model nahrádzame behaviorálnym modelom
 - inferenčné a kompozitné funkcie majú voľné sloty REALIZED BY a doménový model má voľný slot REPRESENTATION
 - tieto sloty sa snažíme naplniť
 - sú dve možnosti naplňania týchto slotov
 - method driven prístup
 - prístup zhora nadol
 - navrhne funkcie
 - na základe nich definujeme úložiskové funkcie
 - representation driven prístup
 - prístup zdola nahor
 - najskôr navrhne reprezentáciu znalostí
 - na základe reprezentácie navrhne funkcie, ktoré budú s nimi pracovať
 - riešiacie funkcie
 - najdôležitejšie funkcie
 - navrhne akú metódu budú vlastne reprezentovať
 - metóda je algoritmus, ktorý potrebuje dáta o vlastnej reprezentácii
 - ku každej metóde musíme povedať
 - akým algoritmom ju budeme vykonávať
 - aké dáta budeme na jej vykonávanie potrebovať
 - dizajn elementov teda je
 - algoritmus
 - dáta
 - rôzne implementácie
 - príklad

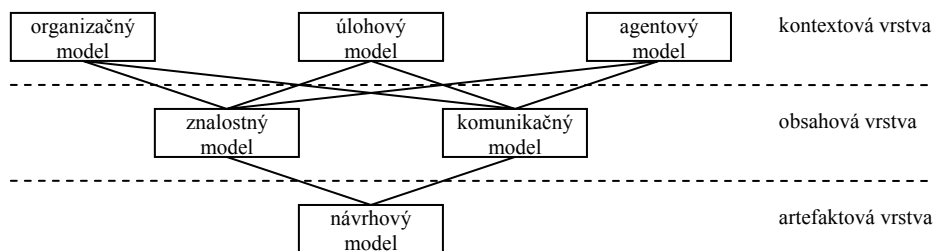


Obrázok 61: Príklad priradenia návrhových metód

- algoritmy sú uložené v knižnici
 - knižnica metód reprezentácie funkcií
 - knižnica metód reprezentácie znalostí
- zostavenie dizajnových elementov
 - zoberieme všetky dizajnové elementy a zostavíme ich do modulov
 - je na nás ako to zostavenie vykonáme
 - výstupom a teda i výstupom celého KBS dizajnu je fyzický model

Metodológia v Common-KADS

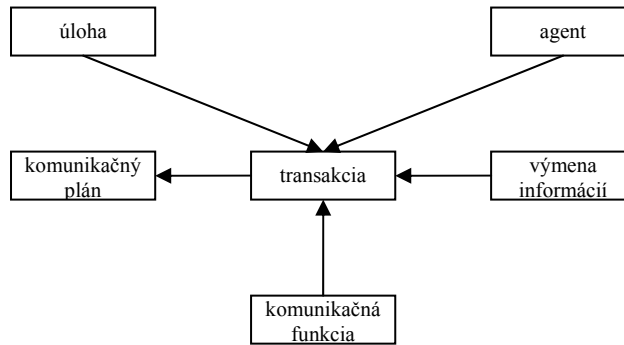
- podobne ako KADS i Common-KADS generuje modely
- KADS návrh delí na analýzu a dizajn
- Common-KADS sa snaží odpovedať na otázky
 - why
 - prečo by mal byť daný systém implementovaný
 - čo bude daný systém vykonávať
 - čo nového daný systém prinesie
 - čo nás to bude stáť
 - aký to bude mať dopad
 - what
 - čo do organizácie vložíme
 - aké znalosti použijeme
 - ako bude systém komunikovať s účastníkmi
 - how
 - ako daný systém zasadíme do organizácie
 - technické prevedenie
- toto rieši pomocou troch úrovní
 - každej otázke zodpovedá jedna úroveň
 - v každej úrovni sú modely



Obrázok 62: Rozdelenie metodológie Common-KADS do modelov

- analýza v KADS je v tomto modely rozdelená do znalostného modelu, komunikačného modelu, agentového modelu, úlohového modelu
- KADS neobsahuje organizačný model
 - vykonávame štúdiu, kde môžeme nasadiť a čo môžeme pomocou znalostného systému zlepšiť
 - na tomto modely, môže návrh skončiť zamietnutím projektu
- procesná analýza v KADS je v tomto modely rozdelená do úlohového modelu a časti organizačného modelu
- komunikačná analýza v KADS je v tomto modely rozdelená do komunikačného modelu a agentového modelu
- expertízna analýzy a dizajn sú rovnaké ako v KADS
- komunikačný model
 - vychádza sa z toho, že agent vykonáva nejakú úlohu
 - agent vyprodukuje informačný objekt, ktorý odovzdá inému agentovi
 - iný agent pozostáva z funkcie, ktorú realizuje a tá daný informačný objekt prijíma
 - teda medzi sebou komunikujú dve úlohy, patriace iným agentom

- o pokiaľ sú dané úlohy znalostné, tak s vonkajším svetom komunikujú pomocou komunikačných funkcií
- o transakcia
 - atomická funkcia komunikácie
 - je to komunikácia dvoch procesov od dvoch agentov
- o komunikačný plán
 - vzťah medzi komunikačným plánom a transakciou je typu consist of
 - komunikácia medzi dvomi procesmi môže byť zložitá



Obrázok 63: Vzťah transakcie a komunikačného plánu

- predstavuje celkovú komunikáciu medzi dvomi agentmi
- pre každého agenta vyberieme všetky procesy, ktoré používa, a vyberieme z nich komunikačné procesy (používajú informačné objekty)
- vyberieme dvoch agentov a zistíme ktoré procesy medzi sebou komunikujú
 - zistíme všetky dvojice procesov
 - každá dvojica reprezentuje transakciu
- spojením procesov ako majú nasledovať dostaneme dialógový diagram
- nad tieto procesy teda musíme navrhnuť riadiacu štruktúru
 - sekvencia
 - o postupnosť v zmysle riešenia úlohy
 - o riadenie je dané tokom dát v agentoch
 - o vhodné pri jednoduchých modeloch
 - o nie je vhodný, ak sú zainteresované v komunikácii aj vonkajšie zásahy
 - pseudokód
 - o používajú sa príkazy

SEND	poslanie informačného objektu
RECEIVE	prijatie informačného objektu
CARRY-OUT	kombinácia SEND a RECEIVE
WAIT UNTIL	čakanie pokiaľ nebude splnená podmienka
WAIT WHILE	čakanie pokiaľ bude splnená podmienka
REPEAT UNTIL	opakovanie pokiaľ nebude splnená podmienka
REPEAT WHILE	opakovanie pokiaľ bude splnená podmienka
IF THEN ELSE	vetvenie
PROCESS	zavolanie ďalšieho procesu v rámci agenta
&	AND
	OR
V	XOR
;	sekvencia

- activity diagram
 - o prechod z procesu do procesu (transakcia) je jeden stav agenta
 - o je vhodné len pre jednoduché riadenie
 - o názorné
- o navrhovanie transakcií
 - pre každú transakciu sa musí vyplniť formulár

TRANSACTION IDENTIFICATOR
 INFORMATION OBJECT
 AGENTS
 COMMUNICATION ?
 CONSTRAINTS
 SPECIFICATION EXCHANGE INFORMATION

- TRANSACTION IDENTIFICATOR
 - o identifikátor transakcie
- INFORMATION OBJECT
 - o informačný objekt
 - o procesy medzi sebou komunikujú pomocou informačných objektov
 - o zapuzdruje časť komunikácie
 - o môžu byť rôznych typov
 - čisté informačné objekty – objekty, kvôli ktorým bola vytvorená komunikácia
 - pomocné informačné objekty – nesú pomocnú informáciu
- AGENTS

- agenti, ktorý medzi sebou komunikujú
 - CONSTRAINTS
 - ohraničenia na transakciu
 - každá transakcia môže byť uskutočnená len za určitých podmienok delenie
 - podmienky na odosielateľa
 - podmienky na prijímateľa
 - ohraničenia po transakcii
 - definujú stav, ktorý nastane po transakcii
 - SPECIFICATION EXCHANGE INFORMATION
 - informačná výmena
 - ak je transakcia jednoznačná, tak sa v tomto bloku len popíše
 - ak je transakcia zložitá, vzniká výmena informácií
- výmena informácií
 - používajú sa kontrolní a riadiace prvky ako v komunikačnom pláne
 - používa sa v rámci jednej správy
 - pre každú transakciu je potrebné vyplniť tabuľku

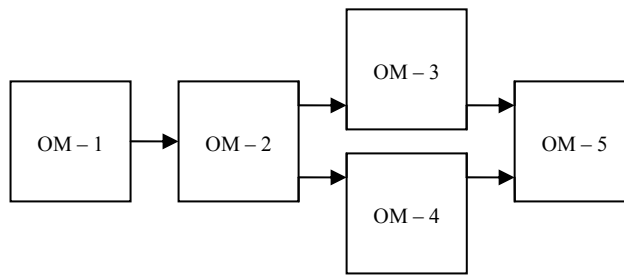
TRANSACTION
AGENT INVOLVED
INFORMATION ITEMS
MESSAGE SPECIFICATION
CONTROL OVER MESSAGE

- TRANSACTION
 - meno transakcie
- AGENT INVOLVED
 - agenti, ktorý sa na danej transakcii podieľajú
 - je definované presne kto je vysielač a kto prijímač správy
 - jeden agent môže byť zároveň vysielač i prijímač
- MESSAGE SPECIFICATION
 - správa má dve úrovne
 - obsah – obsah správy
 - intencia – ako na ňu budeme reagovať
 - vytvorila sa tabuľka

	kladný postoj	kladný postoj	kladný postoj	záporný postoj
delegácia úlohy	REQUEST	REQUIRE	ORDER	REJECT
akceptácia úlohy	PROPOSE	OFFER	AGREE	REJECT
informačná výmena	ASK	REPLY	REPORT	INFORM

Obrázok 64: Tabuľka špecifikácií správ

- každá bunka vyjadruje iný postoj
 - stĺpce v tabuľke vyjadrujú angažovanosť
 - čím je viac vpravo, tým je väčšia angažovanosť
 - v rámci správy sa musí označiť jej účel
 - PROPOSE – vyjadruje nezávislú ponuku
 - OFFER – vyjadruje závislú ponuku
 - AGREE – súhlas s ponukou
 - REPLY – odpoveď
 - REPORT – je podávaný iba tomu, kto má oň záujem
 - INFORM – informovanie všetkých
 - vznikajú typické usporiadania
 - INFORM
 - ORDER + REPORT
 - REQUIRE + (AGREE | REJECT)
 - ASK + REPLY
 - OFFER + (ORDER | REJECT)
 - PROPOSE + (REQUEST | REQUIRE | REJECT)
 - INFORMATION ITEMS
 - informačné prvky
 - informačné objekty, pričom berieme do úvahy všetky (čisté aj pomocné)
 - pre každý definujeme typ, formu a médium
- organizačný model



Obrázok 65: Schéma organizačného modelu

- je potrebné vyplniť 5 formulárov, ktoré na sebe závisia
- je to predbežná štúdia na miesto nasadenia znalostného systému
- na základe tejto analýzy sa rozhodne o nasadení, respektívne nenasadení tohto systému do praxe
- OM – 1
 - snažíme sa vytypovať zaujímavé miesto v organizácii na aplikáciu znalostného systému
 - výsledkom je niekoľko možných nasadení, ktoré špecifikujeme v OM – 2
 - popis

PROBLEMS AND OPPORTUNITIES
ORGANIZATIONAL CONTEXT
SOLUTIONS

- PROBLEM AND OPPORTUNITIES
 - problémy a príležitosti na nasadenie systému do praxe
- ORGANIZATIONAL CONTEXT
 - kontext organizácie, kde chceme znalostný systém nasadiť
- SOLUTIONS
 - možné riešenia

- OM – 2
 - robí sa analýza organizácie voči danej aplikácii
 - popis

STRUCTURE
PROCESS
PEOPLE
RESOURCES
KNOWLEDGE
CULTURE & POWER

- STRUCTURE
 - štruktúra organizácie, ktorá sa podieľa na tom, kde chceme znalostný systém nasadiť
- PROCESS
 - procesy v organizácii, s ktorými bude systém komunikovať
 - bližšie popísané v OM – 3
- PEOPLE
 - ľudské zdroje
- RESOURCES
 - ostatné zdroje
- CULTURE & POWER
 - kultúra v organizácii
 - vzťahy v organizácii

- OM – 3
 - detailne rozpísané procesy
 - výsledkom je do akej miery by sme vedeli na danú úlohu použiť znalostný systém
 - popis

NUMBER
TASK
PERFORMED BY
WHERE
KNOWLEDGE ASSET
INTENSIVE
SIGNIFICANCE

- NUMBER
 - identifikátor úlohy
- TASK
 - názov úlohy
- PERFORMED BY
 - kým je úloha uskutočnená
 - ak človekom, tak sa vzťahuje k ľudským zdrojom
 - ak počítačom, tak sa vzťahuje k iným zdrojom
- WHERE
 - umiestnenie v organizácii
- KNOWLEDGE ASSET

- zoznam použitých znalostí na vyriešenie úlohy
 - INTENSIVE
 - označenie nakoľko je daná úloha pre nás dôležitá
 - či sú potrebné špeciálne znalosti na vyriešenie úlohy
 - SIGNIFICANCE
 - odhad dôležitosti procesu
- OM – 4
- znalosti potrebné pre vyriešenie úlohy
 - popis

KNOWLEDGE ASSET
 POSSESSED BY
 USED IN
 RIGHT FORM
 RIGHT PLACE
 RIGHT TIME
 RIGHT QUALITY

- KNOWLEDGE ASSET
 - identifikátor znalosti
 - musíme popísať všetky znalosti z OM – 3
- POSSESSED BY
 - ktorý agent danú znalosť používa
- USED IN
 - v ktorej úlohe je potrebná daná znalosť
- RIGHT FORM
 - či má znalosť správnu formu
- RIGHT PLACE
 - či je dostupná na správnom mieste
- RIGHT TIME
 - či je dostupná v správnom čase
- RIGHT QUALITY
 - či má správnu kvalitu

- OM – 5
- vyhodnotenie, či má zmysel pokračovať v projekte
 - toto sa vykonáva z viacerých hľadísk
 - obchodné hľadisko
 - technické hľadisko
 - projektové hľadisko
 - každé hľadisko má svoje riziká
 - popis

BUSINESS FEASIBILITY
 TECHNICAL FEASIBILITY
 PROJECT FEASIBILITY
 PROPOSED ACTIONS

- BUSINESS FEASIBILITY
 - obchodná štúdia výhodnosti nasadenia znalostného systému v organizácii
- TECHNICAL FEASIBILITY
 - technická štúdia nasadenia znalostného systému v organizácii
- PROJECT FEASIBILITY
 - projektová štúdia
- PROPOSED ACTIONS
 - výsledok, zhrnutie a navrhnutie akcií