

Otázka č. 4.4.

Distribuovaná umělá inteligence (DAI) (UI2 str.142)

Technickým cílem DAI je navrhnout praktickou metodologii řešení problémů pomocí distribuovaných systémů. Vlastnosti výsledného řešení záleží jak na schopnostech zúčastněných agentů, tak na organizaci jejich spolupráce.

Nabízejí se dvě perspektivy pro pozorování světa DAI, a to

- zdola, z hlediska jednotlivého agenta, a
- shora, z hlediska vztahů ve skupině.

Použití multiagentového systému jako technického řešení nabízí tyto výhody, jakých si ceníme na týmové práci:

- zkrácení doby řešení vzhledem k možnostem paralelního postupu,
- snížení nároků na komunikaci, pokud se každý specializovaný člen kolektivu zabývá jen svou relevantní částí údajů a předává ostatním především své závěry o nich,
- zvýšení operativnosti a spolehlivosti vlivem toho, že tým může podle potřeby přibírat další specialisty, popř. se někteří členové týmu mohou vzájemně zastupovat.

Distribuované řešení úloh

- dekompozice úlohy, alokace zdrojů
- komunikace mezi moduly (jaké jazyky a protokoly mají být použity, a co, kdy a komu má být sděleno)
- koordinace a kooperace (Jaké metody slouží k odhalení nestejných názorů či rozporuplných záměrů agentů? Jak je zajištěno aby se akce jednotlivých agentů vzájemně doplňovaly? Prostředky, které dovolí agentovi uvažovat o akcích, plánech a znalostech ostatních agentů.)

Trend:

Skupina volně propojených autonomních systémů. Spolupracujících v zájmu dosažení nějakého společného cíle-multiagentové systémy.

Typy agentů:

- reaktivní - nemají svůj vlastní cíl, vždy reagují na změny podmínek okolního světa. Mají k dispozici předem známou množinu akcí, které reprezentují např. nějaké stereotypní plány. Jako typický příklad agenta tohoto typu může sloužit např. klasický expertní systém 1. generace.
- intencionální (deliberativní) – mají svůj cíl, fungují proaktivně, prioritou je splnění cíle. Zvažuje své možnosti dosáhnout nějakého cíle. Ve volbě cíle se odráží motivace agenta, který vytváří k jeho dosažení plán svých akcí. Koordinace činnosti skupiny agentů probíhá tak, že se agenti vzájemně informují o tom, čemu momentálně věří, i o tom, co plánují.
- sociální – upřednostňuje cíle celé skupiny. Pracuje s explicitními modely chování ostatních agentů, kteří s ním tvoří MAS a přizpůsobuje své plánované činnosti jiných agentů.

Definice agenta

1) obecnější [Jennings, Wooldridge]

Agent= HW nebo SW systém, který je

- autonomní
- sociální (bere ohled na cíle celé skupiny)
- reaktivní
- proaktivní (sám si vytváří cíle a sám je také vykonává)

2) specifitější

Agent= autonomní, reaktivní a proaktivní počítačový systém, jehož jednání vychází z předpokladů, že jde o systém, který je

- mobilní
- korektní
- benevolentní (není bezohledný vzhledem k ostatním)
- racionální

Metody komunikace:

- 1) Přímá – synchronní (periodicky v daný čas)
- asynchronní (typické pro agenty)
- 2) Nepřímá (např. tabule)

Přímá komunikace:

- Adresné posílání zpráv – předávání zpráv na specifickou adresu příjemce, která musí být odesílateli známa.
Výhody: snadná a efektivní implementace, snadné zajištění a kontrola bezpečnosti
Nevýhody: adresnost (jak agent dozví adresu kam posílat, problém je zvláště aktuální u budování otevřených systémů, kde se může počet agentů dynamicky měnit).
- Všeměřové vysílání zpráv (broadcasting) – zasílání zprávy všem agentům najednou
Výhody:
 - kompoziční (každý agent může být nahrazen jiným agentem podobného chování, přičemž se chování celku nezmění)
 - ideální v systémech, kde mezi sebou agent soutěží nebo spolupracují při získávání kontraktů
 - použití v adaptivních systémech odolných vůči poruchám (agenti, kteří odumřou jsou nahrazeni jinými a umožní systému pokračovat v činnosti)
- Selektivní vysílání zpráv (uvědomovací modely – acquaintance models) – nejběžnější metoda zasílá určité zprávy pouze určitým agentům => musím mít někde model sítě agentů. Tímto způsobem se redukuje komunikace.

Organizace komunity agentů

Původní idea : množina rovnoprávně komunikujících a kooperujících entit. Každý agent pak vlastní příslušnou část globální strategie ve formě lokálních pravidel.

Později: Nestrukturovaná architektura agentové komunity není nejefektivnější. Diferenciace funkcí: někteří agenti pozorují a korigují chování ostatních – metaagent (superagent).

V důsledku toho: přechod k hierarchické struktuře komunity agentů

- může působit i více metaagentů (s překryvem)
- typická úloha metaagentů – rekonfigurace sítě agentů (např. při změně počtu agentů)

Metaagent je obecně schopen:

- sledovat chování jednoho či více agentů
- vyhodnocovat řešení konkrétní funkce zabezpečované příslušnou částí komunity
- sledovat komunikační provoz ve vymezené části komunity

Cílem činností metaagentů je:

- optimalizovat multiagentovou architekturu pro urychlení rozhodovacího procesu (koalice)
- filtrovat zprávy (cenzura)
- poskytovat uživateli „hrubé“ vysvětlení fyzikální podstaty rozhodovacího procesu (propaganda)

Agent = tělo + obal

Tělo obsahuje vlastní program

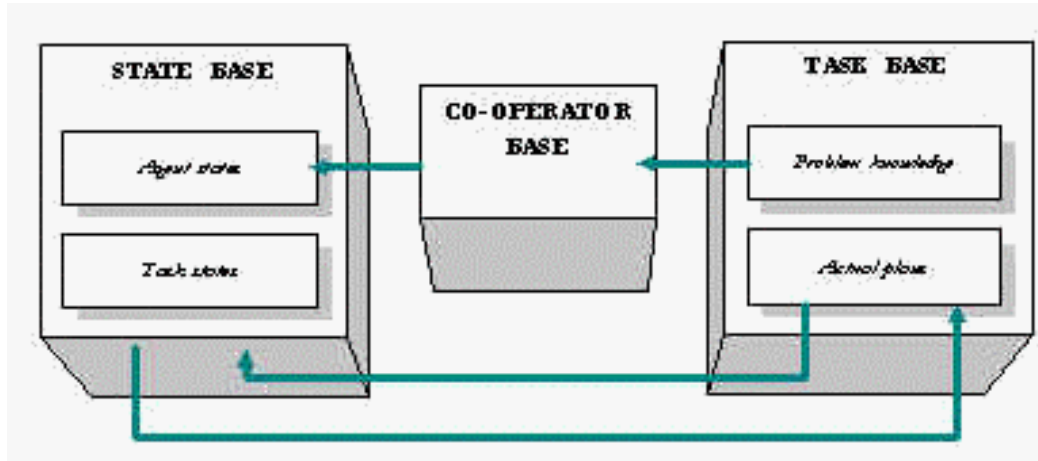
Obal:

- modely chování agentů
 - komunikační modul (vysílač a přijímač zpráv v ACL – agent communication language)
- Agentifikace-spojení obalu a těla.

TRI-BASE ACQUAINTANCE MODEL – umístěn v obalu agenta

Skládá se z :

- Cooperator base – obsahuje permanentní a semi-permanentní informace o jiných agentech
- State base – obsahuje aktuální informace o stavu všech agentů.
- Task base – obsahuje informace (znalosti), jak dekomponovat úlohu a udržuje aktuální plány činnosti.



Cooperator base: skládá se ze dvou částí

- Collaborators section

$\langle a, address_a, Language_a, Set-of-tasks_a \rangle$

a - jméno agenta

Address_a - fyzická adresa agenta a

Language_a - komunikační formát

Set-of-tasks_a - úlohy agenta a

-subscribers section

obsahuje adresy agentů, kteří mají zájem o výsledky agenta A.

State base: skládá se ze dvou částí

-agent section poskytuje data o stavu kooperujících agentů

$\langle a, Load_a, Trust_a, Capabilities_a, Schedule_a \rangle$

a - agent

Load_a – zatížení agenta

Trust_a – míra důvěry agenta

Capabilities_a – charakteristika aktuálních zdrojů agenta

Schedule_a – rozvrh aktivit agenta

-task section spravuje znalosti o aktuálním stavu řešení

$\langle Task, Agent, Solution, Trust \rangle$

Solution popisuje stav řešení úlohy Task agentem Agent

Task Base:

$\langle T_j, A_i, S_{ij} = \{ \langle T_{jk}, A_{jk} \rangle \} \rangle$

Úloha T_j je řešitelná agentem A_i (neumí ji ale řešit celou sám) => dekompozice.

A_i dekomponuje T_j na T_{jk} , které řeší agenti A_{jk} .

Ve složitější verzi pětice:

$\langle T_j, A_i, S_{ij} = \{ \langle T_{jk}, A_{jk} \rangle \}, C_{ij}, V_{ij} \rangle$; C_{ij} ...omezení (např.zatížení), V_{ij} ...důvěra agenta i v agenta j.

Předpokládá se: (performativy)

- vysílání požadavků

ask

request

- vysílání odpovědi

reply

sorry

- vysílání zpráv redukujících objem komunikace

subscribe

advertise

(KQML performativy (standardní zprávy))

TRI-BASE model umožňuje následující objem revize struktury komunity:

- 1) Agent A odumřel, všechny pětice ve kterých figuroval jsou zrušeny.
- 2) Úloha T_j nemůže být dále řešen agentem A_i jako koordinátorem – všechny pětice, v nichž se objevuje pár $\langle T_j, A_i \rangle$, a všechny pětice, v nichž má být koord. A_i jsou zrušeny.
- 3) Narození nového agenta – do příslušných task bases se doplní pětice reprezentující schopnosti nového agenta.
- 4) Existující agent získal nové schopnosti.
- 5) Změny v odhadu zatížení.
- 6) Změny v omezeních.
- 7) Změny v měrách důvěry.

KQML (Knowledge and Query Manipulation Language)

Norma pro komunikaci ve znalostních systémech. KQML je formát zpráv a protokol pro výměnu zpráv podporující run-time výměnu a sdílení znalostí mezi Knowledge Reasoning systémy, DBMS, uživateli, agenty.

•definuje formát zpráv

•systém přenosu zpráv jako obecný rámec pro kooperaci a komunikaci v MAS.

Dále umožňuje sdílet znalosti prostřednictvím komunikačního facilitátoru – speciální třídy agentů. Facilitátor identifikuje ostatní agenty (jména, adresy), udržuje bázi nabízených služeb agenty, zajišťuje ostatní komunikační služby jako je forwarding, broadcasting, zajišťuje překlad obsahu zprávy mezi sémantikou a ontologií agentů.

Struktura KQML zprávy:

- 1) Communication wrapper – mechanismus komunikace – odesílatel, příjemce, synchronizace, identifikace zprávy, potvrzení.
- 2) Message wrapper – logika komunikace – performativy, „speech act type“, kvalifikace.
- 3) Content expression – obsah zprávy v dohodnutém jazyce a v rámci dohodnuté ontologie.

Formát zápisu:

(<Performative>

: content <statement/speechact>

: sender <name>

: receiver <name>

: language <text>

: ontology <text>

)

„Performative“ odpovídá některé ze „speechactů“ (akce prováděná ve zprávě).

Speechacts - předdefinované

- dodefinované v MAS konstruktérem.

Jazyk může být libovolný (def. KQML je dosti volná), je specializován v položce „Language“.

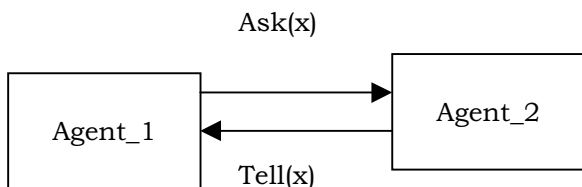
Ontology obsahuje doplňkovou sémantickou informaci.

Př. Performative – ASK, REGISTER, ACHIEVE, TELL, EVALUATE REPLY, SORRY

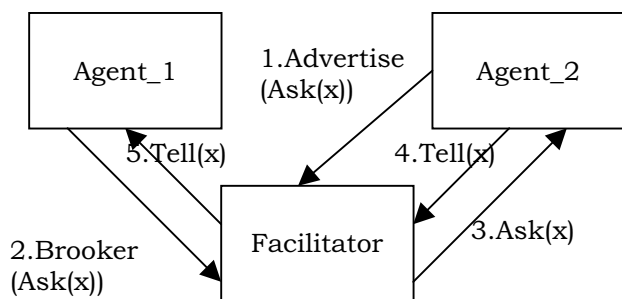
Dialog v KQML

- přímý
- nepřímý - (facilitator) – broker performativa
 - recommend performativa
 - subscribe performativa

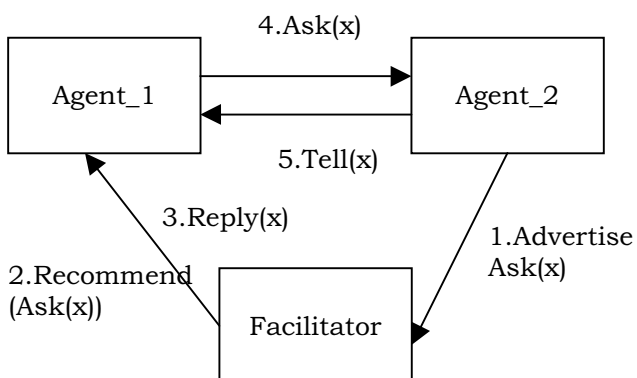
Přímá komunikace



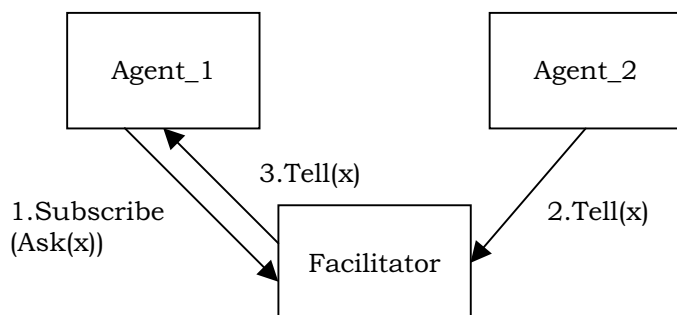
Broker



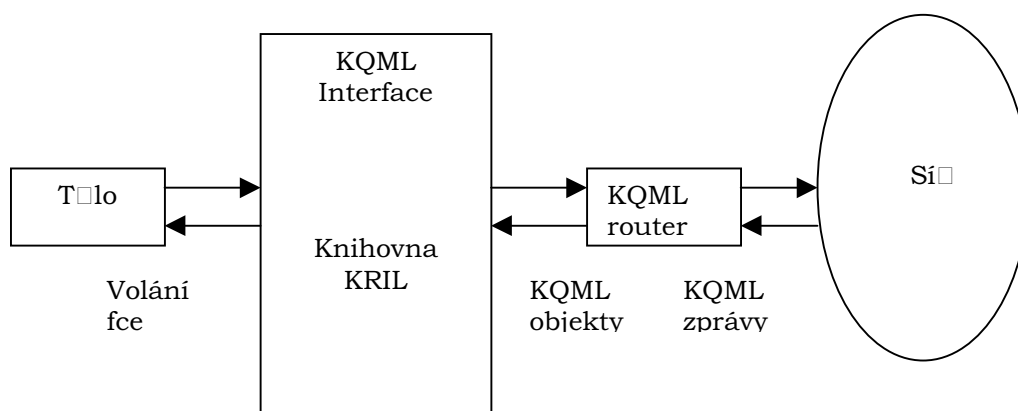
Recommend



Subscribe



Standardní architektura agenta komunikujícího v KQML



Základní principy kooperace a koordinace

Koordinace

- vnáší do jednání komunity agentů řád
- vytváří společenské normy systému

Kooperace

Princip organizace činnosti skupiny agentů vybraných pro realizaci dané úlohy, které zajistí její úspěšné dokončení.

Extrémní strategie koordinace

- každý agent je informován o všem => přetíží se komunikační kanál
- centrální „vševědoucí“ agent => přetížen je tento agent

Vždy nutno volit rozumný kompromis.

Při naprosté „spolehlivosti“ agentů „veškeré koordinační mechanismy mohou být redukovány na závazky (commitments) a různé typy odpovídajících úmluv (conventions)“.
(Jennings 1996)

Závazek = příslib agenta vykonat určitou činnost v určitém čase. K přijetí závazku dochází za určitých podmínek, tyto podmínky se mohou měnit.

Agent může upustit od závazku (1) cíl vyřešen, (2) cíl nedosažitelný, (3) ztráta motivace.

V úmluvách se stanoví podmínky, za kterých je nutné dodržovat závazky, popř. i způsob dosažení zvolených cílů.

Koordinace:

- Vytvoření kostry plánu řešení.
- Přijetí závazků a úmluv.

Vznik kostry plánu

- Analýza cílů, dekompozice na podcíle a elementární úlohy.

Cíle souvisejí - přímo (tedy navazují)

- nepřímo (některé podcíle sdíleny).

- Výsledná struktura – pseudoles skupiny AND-OR stromů => kostra plánu.
- Identifikace cílů společných pro více agentů.
- Identifikace cílů vyžadující kooperaci více agentů (nelze zvládnout koordinaci).

Postup procesu koordinace

- 1) Vytvoření grafu cílů včetně vazeb mezi podcíli
- 2) Rozdělení úloh tvořících kostru plánu (graf cílů) mezi jednotlivé agenty.
- 3) Řízení postupu prohledávání grafu cílů (vytváření lok. plánů).
- 4) Postupné plnění navržených cílů.
- 5) Zveřejnění informace o úspěšném průchodu grafem cílů.

Typy závazků a smluv

- Smlouva (úmluva) říká, kdy je možné závazek přehodnotit (příp. od něj ustoupit).
- Sociální smlouva říká kdy a jak se oznámí odstoupení od závazku.
- Sociální smlouva pro společnou akci by měla zajišťovat vzájemnou podporu činnosti agentů.

Kooperace

Kooperace – nekomunikativní

- komunikativní – deliberativní (úvahová) → PGP (Partial Global Planning)
- dohadovací (vyjednávací) → Contract Net
- Vícestupňové vyjednávání

PGP

- 1) Vytvoření lokálních plánů pro každého agenta.
- 2) Komunikace, výměna plánů mezi agenty.
- 3) Tvorba PGP
- 4) Modifikace a optimalizace PGP

Výsledný PGP zahrnuje:

- globální cíl
- plán aktivit
- graf řešení (solution construction graph) – struktura komunikace a kooperace
- stav realizace, stav agentové komunity.

Výhody PGP:

- velmi dynamický charakter
- nízká redundance (žádná úloha se neřeší duplicitně)

Nevýhoda: obtížná realizace bez centrálního plánovače.

Contract Net protokol

- 1) Agent hledající pomoc při řešení se dočasně stává tzv. MANAGEREM. Formuluje podúlohu, o jejíž řešení chce požádat dalšího agenta.
- 2) Manažer otvírá nabídkové řízení včetně termínu přijímání nabídek.
- 3) Dostupní agenti vyhodnotí požadavek.
- 4) Vhodní agenti postoupí manažerovi nabídky pro splnění podúlohy.
- 5) Po vypršení termínu manažer vyhodnotí nabídky.
- 6) Manažer udělí kontrakt nejvhodnějšimu agentovi.
- 7) Je zahájena realizace kontraktu, v jejímž průběhu spolu manažer a kontrahovaný agent komunikují.

Je-li realizace neúspěšná -> další nabídkové řízení.

Zdokonalení:

- cílené nabídkové řízení viz. TRI-BASE
- předkládání nabídek bez předchozí výzvy
- presumpce závaznosti nabídky
- maximální doba čekání na definitivní potvrzení nabídky je omezena nebo je vyžadována velmi rychlá odpověď (i negativní).

Vícestupňové vyjednávání

p-cíle (prioritní)

s-cíle (sekundární)

- 1) Tvorba dočasného plánu (jsou uvažovány pouze p-cíle každého agenta).
- 2) Vysílání žádosti o kooperaci.
- 3) Prověření realizovatelnosti dočasného plánu (přidávání s-cílů, o než požádali jiní agenti)ů vzniká seznam aktivních cílů.

- 4) Revize dočasného plánuů změny či konflikty sdělovány relevantním agentům.
- 5) Kontrola původních plánů. → (3)

ONTOLOGIE

Ontologie je specifikace konceptualizace. Zabývají se problemaikou jak reprezentovat obecné doménové znalosti. Každý znalostní systém v sobě obsahuje jistou konceptualizaci okolního světa. Systém vykazuje ontologický závazek pokud založí své znalostní struktury a další řešení problému na tomto konceptu. Ontologie je tedy specifikace konceptu pomocí množiny definic reprezentovaných pomocí termínů jako jsou třídy, relace, sloty, metody atd. .

Sdílené ontologie se používají pro specifikaci ontologických závazků mezi skupinou agentů, tak aby mohli konzistentně komunikovat o doménových znalostech , aniž by nutně museli pracovat nad globálně sdílenou teorií. Říkáme, že se agent zaváže k ontologii jsou-li jeho pozorovatelné akce konzistentní s definicemi ontologie.

Myšlenka ontologického závazku plyne z pohledu znalostní úrovně. Znalostní úroveň je úroveň popisu znalostí agenta nezávislé na symbolické úrovni reprezentace, kterou agent interně používá. Znalosti atribuované agentům jsou identifikovatelné skrze jejich chování (akce); agent něco ví, chová-li se tak jakoby měl danou informaci a snažil se dosáhnout cíle.

Ontolingua je jazyk pro definování ontologií. Existuje spousta ontologií definovaných ve svých specifických jazycích. Ontolingua používá KIF pro definici přenositelných ontologií.

KIF – Knowledge Interchange Format

Jedná se o formát, syntaktický přepis a ne o programovací jazyk, protože neexistuje obecný překladač KIFu. Existuje mnoho translátoru KIFu do specifických programovacích jazyků. Jazyk je založen na a rozšiřuje pre-fixovou notaci jazyka predikátového logického kalkulu prvního řádu.

Má následující vlastnosti:

- jednoduchá seznamová ASCII syntaxe
- sémantika s axiomatickým charakterem umožňující reprezentovat širokou slovní zásobu objektů , funkcí a relačních vztahů pro atomy, čísla, množiny a seznamy
- podporuje reprezentovat znalosti o vlastnostech relací či funkcí
- podporuje modalitu – možnost vyjadřovat znalost o znalostech
- podporuje monotónní a nemonotónní inferenční pravidla a úplných a částečných definic.

KVALITATIVNÍ SIMULACE (UI1 str.121)

Znaménka hodnot uvažovaných veličin a jejich vývojových tendencí hrají při vysvětlování významnou roli. Proto základní obor hodnot pro kvalitativní uvažování zahrnuje vždy odpovídající základní hodnoty, tedy plus, nula a minus. Kvalitativní obor hodnot pro každou studovanou kvalitativní proměnnou by měl navíc umožňovat rozlišit významné okamžiky, kdy dochází k zásadním změnám chování odpovídající fyzikální veličiny, např. derivace proměnné, je pro danou hodnotu 0 nebo nastává změna v množině zákonů popisujících funkci studovaného systému. Hodnotám, v nichž nastává taková situace pro danou proměnnou se říká *významné hodnoty* dané proměnné. Obor hodnot kvalitativní proměnné, nazývaný někdy kvalitativní prostor, je vždy konečný a někdy zahrnuje vedle základních hodnot {plus, nula, minus} i některé další hodnoty.

Přechod k diskrétním kvalitativním hodnotám je důležitým krokem směrem k porozumění a modelování procesu lidského uvažování. Dalším krokem musí být návrh vhodného formálního systému, který by umožňoval nahradit analytické vyjádření vazeb mezi sledovanými veličinami nějakým alternativním způsobem. I tato snaha je motivována hledáním postupů, které jsou nutné, ale i postačující pro práci s kvalitativními hodnotami.

QSIM

Soustředí se na spojitě děje. Snaží se však o podrobnější rozbor chování systému uvnitř jediné fáze chování, tj. analyzuje ty časové úseky, kdy je systém ovládnut stále tímž neměnným souborem omezujících podmínek. Pro vyřčení omezujících podmínek, které vzájemně vážou kvalitativní hodnoty pozorovaných proměnných Kuipers zavádí několik nových primitivních relací pro vazby mezi proměnnými:

$M^+(f,g)$ – binární relace „veličina f je přímo úměrná veličině g “

$M^-(f,g)$ – binární relace „nepřímá úměrnost“

Derivace(f,g) – binární relace je splněna, pokud g je derivací f

Kvalitativní stavy – kvalitativní ohodnocení všech pozorovaných proměnných nespokojuje se pouze s elementárním oborem kvalitativních hodnot $\{+,0,-\}$.

Mezní hodnoty (landmark values) – jde o hodnoty při nichž dochází v chování pozorované proměnné k nějaké kvalitativní změně.

Kvalitativní hodnota proměnné má vždy dvě složky, a to údaj o aktuální velikosti a údaj o „vývojové tendenci“ proměnné ($\{+,0,-\}$ odpovídá $\{\text{roste, je nulová, klesá}\}$).

Samotná kvalitativní velikost proměnné je udávána vůči uspořádané množině mezních hodnot. Veličina buď nabývá přímo mezní hodnoty, nebo má velikost specifikovanou jako interval mezi dvěma sousedními mezními hodnotami.

I-přechod – vývoj funkce při přechodu z intervalu do bodu.

P-přechod – přechod z bodu do stavu v intervalu, který bezprostředně následuje.

I - přechody	QS(f, t_i, t_{i+1})	QS(f, t_{i+1})
I1	$\langle l_j, 0 \rangle$	$\langle l_j, 0 \rangle$
I2	$\langle (l_j, l_{j+1}), + \rangle$	$\langle l_{j+1}, 0 \rangle$
I3	$\langle (l_j, l_{j+1}), + \rangle$	$\langle l_{j+1}, + \rangle$
I4	$\langle (l_j, l_{j+1}), + \rangle$	$\langle (l_j, l_{j+1}), + \rangle$
I5	$\langle (l_j, l_{j+1}), - \rangle$	$\langle l_j, 0 \rangle$
I6	$\langle (l_j, l_{j+1}), - \rangle$	$\langle l_j, - \rangle$
I7	$\langle (l_j, l_{j+1}), - \rangle$	$\langle (l_j, l_{j+1}), - \rangle$
I8	$\langle (l_j, l_{j+1}), + \rangle$	$\langle l^*, 0 \rangle$
I9	$\langle (l_j, l_{j+1}), - \rangle$	$\langle l^*, 0 \rangle$

Tabulka I-přechodů

Pravidla I8 a I9 upozorňují na možnost, že aktuální množina mezních hodnot není úplná a dovolují zavést novou mezní hodnotu l^* .

Viz. Umělá inteligence I - příklad na straně 138.

LITERATURA

Mařík, Štěpánková, Lažanský a kol.: Umělá Inteligence 1.a 2. díl