



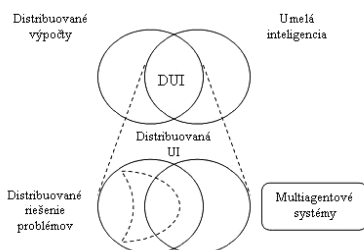
# Multiagentové systémy zamerané na riadenie mobilných robotických systémov

Článok sa zaoberá prehľadom rýchlo sa rozvíjajúcej oblasti multiagentových systémov. Vzhľadom na obšírnosť problematiky sú informácie zovšeobecňované tak, aby poskytli hlavne základné údaje týkajúce sa motivácie a koncepcie agentových a multiagentových systémov so zameraním na aplikáciu riadenia mobilných robotických systémov. Orientuje sa najmä na analýzu agentových architektur a vhodnej architektúry pre úlohy spojené s riadením pohybových systémov.

## Pozadie vzniku MAS

Multiagentové systémy vznikli ako rozšírenie oblasti distribuovanej umelej inteligencie, ktorá tvorí prienik medzi *distribuovanými výpočtami* (*distributed computing*) a umelou inteligenciou. Oblasť *distribuovaných výpočtov* existuje tak dlho, ako možno rozdeliť riešenie jedného výpočtového problému na viac procesorov. Pôvodne tieto procesory spoločne využívali dáta spojené s riešeným problémom a hlavnou úlohou bolo zabezpečiť ich paralelizáciu a synchronizáciu [9].

Ďalšie rozšírenie distribuovaných výpočtov úzko súvisí s rozvojom umelej inteligencie, ktorá aplikovala distribučný prístup na problémy riadenia spoločne využívaného viacerými procesormi. Výsledkom tejto fúzie bola oblasť nazvaná *distribuovaná umelá inteligencia*. Rozdiel od distribuovaných výpočtov je práve v zameraní na riešenie problémov, komunikáciu a koordináciu oproti nízkoúrovňovej paralelizácii a synchronizácii. Rozvoj manažmentu informácií neskôr spôsobil ďalšie rozdelenie distribuovanej umelej inteligencie na *paralelnú umelú inteligenciu*, *distribuované expertné systémy*, neskôr aj na *distribuované zdroje znalostí* (*distributed knowledge bases*) a *distribuované riešenie problémov* (*distributed problem solving*) [3]. V súčasnosti sa však najčastejšie možno stretnúť s delením distribuovanej umelej inteligencie na dve triedy: *distribuované riešenie problémov* a *multiagentové systémy* (obr. 1).



Obr.1 Taxonómia oblastí zahrňajúcich multiagentové systémy

Kým distribuované riešenie problémov súvisí s manažmentom informácií, oblasť multiagentových systémov je zameraná na manažment správania. V distribuovanom riešení problémov sa zvyčajne požadujú vysoké nároky na vzájomnú kompatibilitu jednotlivých riešiacich entít. Na druhej strane v multiagentových systémoch môžu byť agenti z hľadiska vlastností úplne odlišné a zvyčajne majú aj tendenciu vzájomne interagovať. Ďalšou dôležitou špecifickou vlastnosťou multiagentových systémov je, že správanie jedného agenta je spravidla om-

noho jednoduchšie, ako komplex vzájomných interakcií všetkých agentov v systéme. Práve tieto interakcie správania sú základným kameňom oblasti multiagentových systémov. Hranica medzi týmito dvoma odvetvami však, v konečnom dôsledku, vôbec nie je jednoznačná.

Podľa [4] možno vyčleniť základné oblasti, ktoré z historického a formálneho hľadiska priamo súvisia s multiagentovými systémami:

- *Distribuované výpočty*: Rozdielne procesory spoločne využívajú dáta, ale nie riadenie. Riešenie nízkoúrovňovej paralelizácie a synchronizácie.
- *Distribuovaná umelá inteligencia*: Riadenie podľa distribúcie dát. Riešenie problémov, komunikácia a koordinácia.
- *Distribuované riešenie problémov*: Dekompozícia úloh (spoločné využívanie úloh) a/alebo syntéza riešení (zdieľanie výsledkov): manažment informácií.
- *Multiagentové systémy*: Koordinácia správania alebo manažment správania. Zvyčajne neexistujú žiadne garancie o ostatných agentoch. Zložitosť vzájomných interakcií agentov je omnoho väčšia ako zložitosť správania jednotlivcov.

## Multiagentové systémy - úvod

Jednou zo súčastí distribuovanej umelej inteligencie sú multiagentové systémy, ktoré sú definované ako množina presne určených, od seba nezávislých prvkov (agentov), ktorých štruktúra je navrhnutá tak, aby tieto agenti vzájomnou interakciou a komunikáciou dosiahli určitý dopredu definovaný cieľ alebo správanie. Pôvodne sa multiagentové systémy používali najmä na skúmanie a modelovanie vzťahov medzi živými systémami. Takéto modely sa využívali najmä pre potreby ekonomiky, sociológie, ale aj v iných vedných odboroch.

V súčasnosti sa však okrem spomínaných vedných odborov začali multiagentové systémy úspešne používať aj na modelovanie správania inteligentných systémov, čo ich priamo predurčuje na nasadenie v oblasti automatického riadenia autonómnych a mobilných systémov, pričom takto navrhnuté riadiace systémy majú tendenciu splniť náročné požiadavky kladené na inteligentné správanie pri minimalizácii nárokov na ľudskú obsluhu alebo dokonca úplné vylúčenie človeka ako supervízora [10, 11].

Použitie modelu multiagentového systému ako algoritmu riadenia napríklad v riadiacej jednotke mobilného autonómneho systému umožňuje splniť aj náročné požiadavky na vysokú schopnosť samostatného



rozhodovania v reálnom prostredí aplikáciou sady výpočtovo nenáročných rozhodovacích algoritmov [10]. Ich nenáročnosť na výpočtový výkon nám potom dovoľuje využitie menej výkonných riadiacich jednotiek, čo sa priaznivo odrazí na ich energetickej náročnosti, rozmeroch, ale aj na ich cene.

Na druhej strane modelovanie multiagentových systémov na výkonnejšej výpočtovej technike dovoľuje modelovať aj vyššie formy rozhodovacích algoritmov a správania v reálnom čase. Pritom môžeme využiť prirodzenú distribuovanosť multiagentového systému tak, že spracúvanie jednotlivých častí (jednotlivých agentov alebo ich skupín) rozdelíme medzi viacero fyzických počítačov. Musíme však zabezpečiť ich vzájomnú komunikáciu, napr. prostredníctvom lokálnej siete, čo umožní efektívne využiť výpočtový výkon viacerých jednotiek súčasne, takže nie sme obmedzovaní výkonom výpočtovej techniky.

Ďalšou výhodou plynúcou z prirodzenej distribuovanosti multiagentového systému je jeho vysoká modularita, ktorá umožňuje budovať takéto riadiaci systém postupným pridávaním nových modulov na základe jeho skutočnej reakcie v reálnom prostredí, a to bez nutnosti výrazne zasahovať do už existujúcej štruktúry agentov.

### Definícia pojmu agent

V súčasnosti neexistuje presná všeobecná definícia pojmu agent. Poznáme viacero pokusov o definíciu od rôznych autorov, avšak ani jedna sa neujala ako platná a uznávaná definícia tohto pojmu. Najčastejšie citovaná definícia Wooldridgea a Jenningsa [5, 6], ktorá rozlišuje agenty na tzv. silné a slabé, pričom slabý agent je definovaný ako:

Hardvérový alebo (častejšie) softvérový počítačový systém, ktorý spĺňa nasledujúce požiadavky:

- *autonómnosť*: agent koná bez akéhokoľvek priameho ľudského alebo iného zásahu a má kontrolu nad svojimi akciami a vnútorným stavom,
- *sociálna schopnosť*: agenty interagujú s inými agentmi (prípadne človekom) prostredníctvom nejakého komunikačného jazyka,
- *reaktivita*: agenty vnímajú svoje okolie (to môže predstavovať fyzický svet, používateľa – prostredníctvom grafického rozhrania, kolektív agentov, internet alebo ich kombináciu) a v príhodnom čase reagujú na zmenu vo svojom okolí,
- *pro-aktivita*: konanie agentov nie je jednoduchá reakcia na stav okolia, ale môže byť aj cieľovo orientované s prevzatím iniciatívy.

Pri definícii *silného* agenta sa pridávajú k týmto vlastnostiam aj pojmy charakterizujúce „mentálne“ a „emocionálne“ stavy, ako sú napr. vedomosť, viera, zámer. Z veľkého počtu definícií bola sformulovaná v práci [7] nasledujúca koncepcia autonómneho agenta:

Autonómny agent je systém, ktorý je časťou prostredia, ktoré vníma, na ktoré pôsobí a v ktorom existuje v priebehu času, pričom vykonáva vlastné úlohy, a tak ovplyvňuje to, čo bude vnímať v budúcnosti.

Musíme však pripomenúť, že funkcionálnosť agenta má svoje ohraničenia a každý agent je schopný len takého správania, ktoré môže dosiahnuť prispôbením svojich senzorov (napr. kamery na robotoch alebo príkazy na zistenie obsahu adresárov pri softvérových agentoch) a aktuátorov (napr. kolesá, ramená pri robotoch a príkaz na zmazanie súboru pri softvérových agentoch).

Agent je niečo, čo neustále vníma svoje prostredie a na základe toho volí akcie, ktoré má v tomto prostredí vykonať, aby sa dosiahol stanovený cieľ, a následne tieto akcie vykonáva.

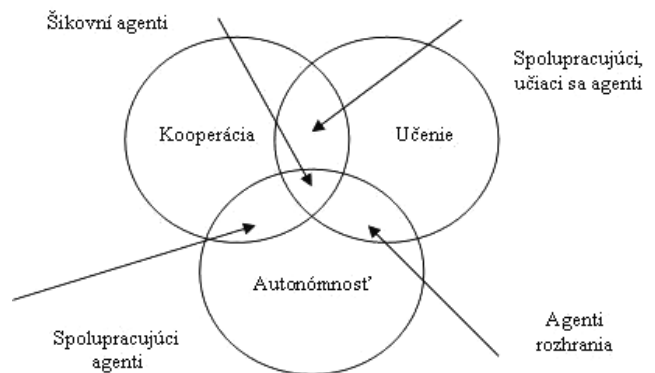
Pod prostredím si môžeme predstaviť nielen okolie mobilného robota, ale aj niečo, k čomu môže agent pristupovať bez zablokovania riadenia. Pod interakciou agenta s prostredím si môžeme predstaviť prijímanie signálov zo senzora (vstup), alebo vysielanie signálov na aktuátor (výstup), priamu komunikáciu agenta s inými agentmi (príjem a vysielanie správy), alebo nepriamu komunikáciu cez určitý proces, ktorý predsta-

vuje prostredie, cez ktoré si agenti vymieňajú správy (čítanie, zápis). Je viacero názorov na to, ktorá architektúra komunikácie a aký typ agentov je výhodnejší, či sú to už deliberatívne agenty s komunikáciou cez prostredie alebo s priamou komunikáciou, prípadne reaktívne agenty s komunikáciou cez prostredie alebo s priamou komunikáciou.

### Typy agentov

Na základe uvedených vlastností možno vymedziť rôzne typy agentov. Zrejme najčastejšie uvádzané rozdelenie agentov (obr. 2) je opísané v práci H. S. Nwanu [8]. Na základe tohto rozdelenia sa špecifikuje niekoľko základných typov agentov, ktoré sú definované vzhľadom na kombinácie jednotlivých vlastností. Uvedená práca vyčleňuje sedem typov agentov:

- kolaboratívne (spolupracujúce) agenty,
- mobilné agenty,
- agenty rozhrania (interfejsové agenty),
- informačné/internetové agenty,
- reaktívne agenty,
- hybridné agenty,
- šikovné, tzv. smart agenty.



Obr.2 Taxonómia typov agentov na základe vlastností podľa Nwanu

### Architektúry agentov

Základnou jednotkou pri tvorbe multiagentových systémov sú, samozrejme, agenty. Vymenujeme teda a opíšeme základné typy architektúr agentov. Jednou z možných definícií architektúry je tá, ktorá hovorí, že architektúra je špeciálnou metodológiou tvorby agentov. Špecifikuje, ako môže byť agent dekomponovaný na konštrukciu tvorenú množinou častí, modulov, a ako by mali tieto moduly vzájomne interagovať. Úplná množina modulov a ich interakcií by mala poskytnúť odpoveď na otázku, ako senzorové údaje a aktuálny vnútorný stav agenta determinujú akcie agenta. Architektúra zahŕňa techniky a algoritmy podporujúce túto metodológiu.

Typy architektúr na tvorbu agentov možno rozdeliť na štyri základné skupiny:

- Deliberatívne architektúry (klasický prístup): *najprv myslí, potom konaj*.
- Reaktívne architektúry (alternatívny prístup): *nemysli, konaj*.
- Hybridné architektúry: *mysli a konaj nezávisle, paralelne*.
- Behavior-based architektúry: *vymysli spôsob, ako niečo vykonať*.

Opíšeme si prvé tri typy architektúr na tvorbu agentov.

#### Deliberatívne architektúry

Tieto architektúry používajú centralizovaný vnútorný model sveta, ktorý slúži na interpretáciu dostupných senzorických informácií a odvodenie adekvátnej akcie. Akcia, resp. sekvencia akcií je typicky produkovaná plánovacím systémom hľadajúcim možné sekvencie stav – akcia vo vnútornom modeli sveta.



Deliberatívne architektúry sú vysoko efektívne na generovanie strategických plánov a akcií, ale bohužiaľ len v prípadoch, keď sú dostupné všetky informácie potrebné na odvodenie aktuálneho stavu vo vnútornom modeli, a tiež musí byť k dispozícii dostatočný čas na odvodenie akcie. V prípade takejto koncepcie návrhu agenta treba riešiť dva základné problémy:

1. problém transformácie reálneho sveta do jeho symbolickej reprezentácie v primeranom čase,
2. problém samotnej symbolickej reprezentácie zložitých informácií o entitách reálneho sveta a procesoch, pričom treba riešiť spôsob manipulácie s týmito informáciami tak, aby výsledky takéhoto uvažovania spĺňali prinajmenšom časové požiadavky.

V prípade agentov situovaných v reálnom prostredí (napr. v prípade mobilných robotov) je však typická vysoká miera šumu a neurčitosti senzoričných údajov v neustále sa meniacom dynamickom prostredí [12]. Dynamické a z hľadiska snímaných údajov často neúplné reálne prostredie svojimi kontinuálnymi zmenami spôsobuje potrebu preplánovania aktuálnej sekvencie akcií, čo zvyšuje hlavne časovú komplexitu odvodenia adekvátnej akcie. Deliberatívne (tiež známe ako *planner-based*) architektúry boli často kritizované pre ich slabú schopnosť pracovať v reálnom prostredí s vysokou komplexitou práve pre neschopnosť generovania adekvátnych akcií v reálnom čase.

### Reaktívne architektúry

Problémy symbolickej umelej inteligencie a nevýhody deliberatívnych architektúr viedli k otázkam variability pôvodných prístupov, na základe ktorých bola vyvinutá architektúra známa ako *reaktívna*. Pre reaktívnu architektúru je typická absencia centrálného, symbolicky reprezentovaného modelu sveta a symbolického uvažovania. Priamym protikladom deliberatívnych modelov bola práca R. Brooks [9], ktorý vyvinul alternatívnu, tzv. *subsumpčnú* architektúru. Navrhol tri základné tézy tejto architektúry:

- Inteligentné správanie možno získať bez explicitných symbolických reprezentácií v zmysle klasickej umelej inteligencie.
- Inteligentné správanie možno získať bez explicitného abstraktného uvažovania v zmysle klasickej umelej inteligencie.
- Inteligencia je vlastnosťou komplexných systémov, ktorá vzniká emergenciou.

Brooksovú subsumpčnú architektúru aplikovanú na reálne mobilné roboty možno opísať ako hierarchiu úlohovo orientovaných modelov, resp. vzorcov správania. Každá úroveň hierarchie reprezentovala typ správania istej komplexity, ktorá sa znižovala s úrovňou v hierarchii. Na najnižšej úrovni boli najjednoduchšie modely (napr. vyhýbanie sa prekážkam). Finálne systémy boli v zmysle náročnosti a počtu výpočtových operácií extrémne jednoduché bez explicitnej reprezentácie a uvažovania. Brooks však napriek tomu demonštroval také úlohy vykonávané robotmi, kde by bolo pôsobivé, ak by tieto úlohy boli riešené symbolickým systémom.

Reaktívne agenty pracujú na princípe podnet – reakcia, teda konajú akoby inštinktívne. Neustále skenujú svoje okolie a keď nastane udalosť, ktorú majú v sebe napevno definovanú, tak vykonajú určitú akciu. Výsledkom tohto procesu sú relatívne jednoduché agenty schopné spolupráce s inými agentmi na nízkej úrovni.

### Hybridné architektúry

Oba opísané typy architektúr majú svoje výhody aj nevýhody. Hybridné architektúry vznikli ako spôsob spojenia pozitívnych vlastností deliberatívnych a reaktívnych architektúr. Cieľom je kombinácia reakcie reaktivity a racionálnych rozhodnutí deliberatívnej časti v reálnom čase.

Ak výstupy oboch podsystémov nespôsobujú vzájomné konflikty, nie je potrebná ich koordinácia. Na druhej strane, aby mohli oba komponenty vzájomne využívať svoje vlastnosti, treba zabezpečiť ich vzájomnú interakciu.

## Základná klasifikácia MAS

Základná klasifikácia pre multiagentové systémy je z hľadiska ich homogenosti a heterogenosti.

V *homogénnych multiagentových systémoch* sú všetky agenty úplne alebo skoro rovnaké. Napriek tomu, že každý agent je v podstate celkom jednoduchý, tak z globálneho hľadiska všetky agenty vytvárajú zložitý model správania. Špeciálne reaktívne agenty sú vhodné na tento druh multiagentových systémov, lebo pracujú na jednoduchom princípe podnet – reakcia. V homogénnych multiagentových systémoch vytvára množstvo nízkoúrovňových interakcií, ako napríklad to, že agent môže vnímať iného agenta alebo objekt a môže s ním aj komunikovať, v kombinácii zložitý systém správania.

*Heterogénne multiagentové systémy* pozostávajú z odlišných typov jednotlivých agentov. V takýchto systémoch každý agent alebo skupiny agentov majú špecializovanú úlohu. Nevýhodou takýchto heterogénnych systémov je, že každý agent má odlišné správanie a je oveľa zložitejšie vytvoriť komunikačný jazyk, ktorému by rozumel každý agent.

### Štruktúra multiagentového systému

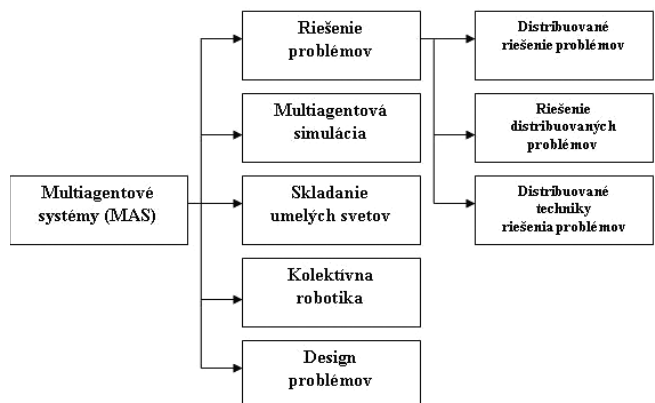
Postavenie agentov v organizácii a vzájomné vzťahy medzi nimi sú veľmi dôležité v MAS. Ako budú jednotlivým agentom pridelené úlohy a aká bude redundancia agentov, je dôležité pri navrhovaní architektúry a štruktúry systému.

Podľa vzájomného postavenia agentov a úloh, ktoré majú plniť v rámci organizácie, rozlišujeme tieto vzťahy:

- Vzťah nezávislosti. Nezávislé agenty používajú zdroje, ktoré nie sú v konflikte, ich funkcionality je nezávislá a produkujú výstupy, ktoré vzájomne nevyužívajú na vlastnú produkciu.
- Vzťah podriadenosti. V prípade statického vzťahu, keď nadradený agent A požaduje od podradeného agenta B splnenie úlohy, nemôže agent B odmietnuť (master – slave). V prípade dynamického vzťahu môže agent A, ktorý je v nadradený, odmietnuť poskytnúť požadovanú službu podradenému agentovi B (server – klient).
- Vzťah rovnosti. Existuje medzi dvoma agentmi na rovnakej hierarchickej úrovni.
- Vzťah závislosti charakterizuje vzťah, v ktorom plnenie cieľov agenta A závisí od agenta B.
- Konfliktný vzťah. Vzniká, ak si rovnocenné agenty robia nárok na využitie rovnakého zdroja.

### Oblasti aplikácie

Oblasti aplikácie MAS je veľký počet. Na ilustráciu sú ďalej spomenuté len tie najdôležitejšie (obr. 3): riešenie problémov v najširšom zmysle slova, kolektívna robotika, multiagentová simulácia, konštrukcia umeľých svetov a genetický dizajn programov.



Obr.3 Klasifikácia rôznych typov aplikácií pre multiagentové systémy



## Záver

Článok bol zameraný na stručný prehľad a analýzu systémov s využitím multiagentového prístupu na riešenie aplikácií riadenia pohybových systémov. Formuluje základné požiadavky na takýto systém a jeho štruktúru. Takto navrhnutý systém bude vyhovovať požiadavkám na riešenia typových úloh, ich variability a rozširovania, ako aj vkladania do existujúcich aplikácií (embedded) o ďalšie stavebné prvky.

## Literatúra

- [1] Kelemen, J.: Strojovia a Agenty. Bratislava: Vydavateľstvo Archa 1994.
- [2] Capra, F.: The Turning Point-Science, Society and the Rising Culture. London: Harper Collins Publishers 1983.
- [3] Decker, K. S.: Distributed Problem Solving: A Survey. IEEE Transactions on Systems, Man and Cybernetics, 1997.
- [4] Stone, P. – Veloso, M.: Multiagent Systems: A Survey from a Machine Learning Perspective. Autonomous Robotics Vol. 8, 2000.
- [5] Wooldridge, M. – Jennings, N. R.: Agent Theories, Architectures, and Languages: a Survey. Intelligent Agents. Ed.: Wooldridge, M., Jennings, N. R. Berlin: Springer-Verlag, 1995.
- [6] Wooldridge, M. – Jennings, N. R.: Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 1995.
- [7] Franklin, S. – Graesser, A.: Is It Agent, or Just a Program? A Taxonomy for Autonomous Agents. In Proceeding of the Third International Workshop on Agent Theories, Architectures and Languages, Springer-Verlag, 1997.
- [8] Nwana, H. S.: Software Agents: An Overview. Knowledge Engineering Review, 1996.

[9] Stone, P. – Veloso, M.: Multiagent Systems: A Survey from a Machine Learning Perspective. Autonomous Robotics Vol. 8, 2000.

[10] VITKO, A. – JURIŠICA, L. – MURÁR, R.: Improving manipulation, navigation and self-diagnostic capabilities of a walking robot. In: AT&P journal plus, október 2007, s. 173 – 176.

[11] VITKO, A. – JURIŠICA, L. – MURÁR, R.: Smoothing out a Robot Path in Cluttered Environment. In: 16th Int. Conference Process Control 2007, 11 – 14. júna 2007, Štrbské Pleso, s. 134-1 – 134-4.

[12] BENICKÝ, P. – JURIŠICA, L.: Klasifikátor textúr. In: AT&P journal 6/2007, s. 73 – 75.

## Ing. Jozef Dorner

Slovenská technická univerzita v Bratislave  
Fakulta elektrotechniky a informatiky  
Ústav riadenia a priemyselnej informatiky  
Ilkovičova 3, 812 19 Bratislava  
jozef.dorner@stuba.sk

61