



# Inteligentné monitorovanie a modifikovanie výkonu informačných systémov (2)

Príspevok opisuje štruktúru a princípy nového typu architektúry informačných systémov, vychádzajúcu zo štandardnej komponentovej architektúry. Uvedená architektúra poskytuje možnosť monitorovania a modifikácie výkonu týkajúceho sa vykonávania operácií poskytovaných komponentmi známymi z modelovacieho jazyka UML. Na to sa vo vykonávacom prostredí pre túto architektúru používajú inteligentné agenty na báze platformy JADE-LEAP.

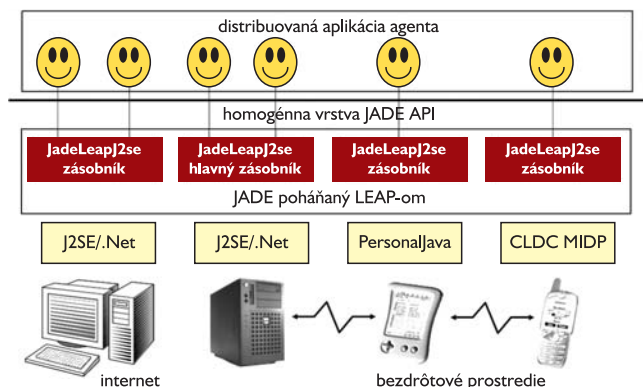
## Agenty a agentové systémy

Návrh a budovanie vysokokvalitných industriálnych softvérových systémov je náročné. Agentovo orientované techniky a prístupy [3] reprezentujú nové, „excitované“ prostriedky analýzy, návrhu a budovania komplexných softvérových systémov s komplexným a sofistikovaným správaním. Tiež majú potenciál značne zlepšiť súčasné praktiky softvérového inžinierstva a rozšíriť oblasť praktických aplikácií. Využitím agentovo orientovaných prístupov by sme v budúcnosti mohli vytvárať také systémy, ktoré budú schopné meniť samy seba (a to napr. učením, adaptabilitou a pod.) a ich návrh a konfigurácia bude viac-menej otázkou zadefinovania požadovaných vlastností namiesto práčného návrhu, resp. implementácie či oboch.

Používanie pojmu „agent“ nie je zjednotené (podobne ako pojmu „umelá inteligencia“), a teda môže byť definovaný rôzne, napr.: Agenty sú nezávislé softvérové entity prejavujúce sa autonómnym a proaktívnym správaním orientovaným na ciele. Medzi základné vlastnosti agenta patria autonómnosť, interaktívnosť, učenie, sociálne schopnosti a proaktivita, čiže schopnosť cielovo orientovaného správania. Agenty poskytujú softvérovým návrhárom a vývojárom možnosť štrukturovať aplikácie na autonómne a komunikujúce komponenty a ponúkajú novú a často vhodnejšiu cestu pre vývoj komplexných výpočtových systémov, obzvlášť v otvorených a dynamických prostrediach. Inteligentné agenty (Intelligent Agents) majú (oproti štandardným) nasledujúce základné vlastnosti: adaptívnosť, inteligencia, racionálnosť a transparentnosť spolu so zodpovednosťou.

## Java Agent Development Framework

Java Agent Development Framework (JADE) [2] je podporná technológia, middleware pre vývoj a vykonávanie distribuovaných multiagentových aplikácií typu peer-to-peer, založených na paradigme agentov,



Obr.1 Vykonávacie prostredie pre JADE-LEAP [1]

ktoré môžu pracovať a interagovať v sieťovom tradičnom aj bezdrôtovom prostredí. Platforma JADE je Open Source projekt vyvinutý organizáciou TILAB – Telecom Italia Lab (predtým CSELT). V JADE môžu byť inteligencia, iniciatíva, informácia, zdroje, resp. ovládanie úplne distribuované na mobilné terminály, rovnako aj počítače v „pevnej“ sieti. Aplikáčne prostredie sa môže podľa potrieb a požiadaviek vyvíjať dynamicky vzhľadom na jednotlivé agenty, ktoré sa objavujú/opúšťajú systém. Agenty platformy JADE sa vyznačujú nasledujúcimi vlastnosťami:

- sú aktívnymi, voľne spriahnutými entitami schopnými asynchrónnej komunikácie,
- vykonávajú určité akcie a ich komunikácia zahŕňa aj sémantický význam.

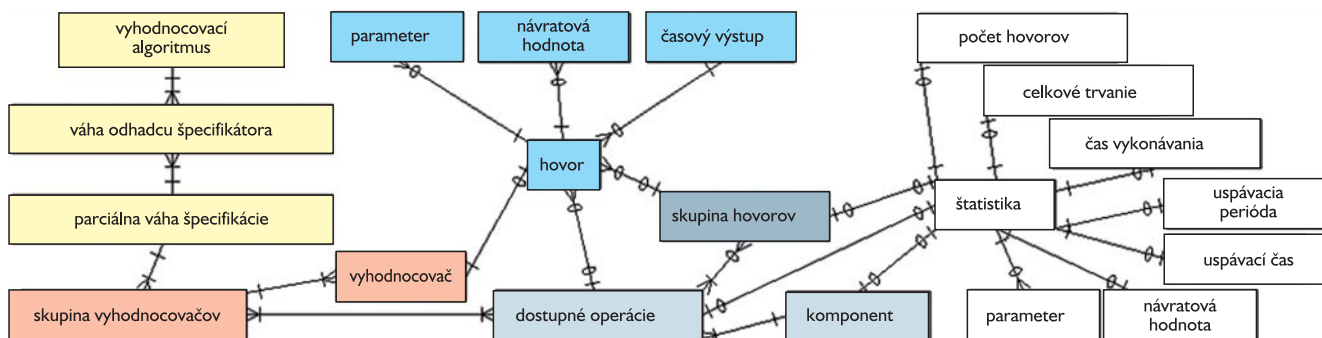
JADE-LEAP („JADE powered by LEAP – Lightweight Extensible Agent Platform“, obr. 1) použitý vo vykonávacom prostredí pre navrhnutú architektúru je modifikovanou verziou platformy JADE, ktorá umožňuje, aby boli agenty vykonávané na zariadeniach s obmedzenou výpočtovou kapacitou, napr. na mobilných telefónoch, zariadeniach prevádzkujúcich Javu alebo .NET Framework [1]. Platforma LEAP je prídavný modul k JADE, vyvíjaný prevažne v rámci projektu LEAP IST.

## Použitie agentov pri monitorovaní a modifikovaní výkonu IS

Dôvod, pre ktorý boli použité inteligentné agenty platformy JADE-LEAP, spočíva najmä v ich schopnostiach autonómne vykonávať určitú činnosť bez nutnosti zásahu a pri výborných možnostiach plánovania. Tým, že za modifikáciu a monitorovanie výkonu zodpovedajú inteligentné agenty, zjednodušuje sa návrh aj samotná prevádzka IS a otvárajú sa možnosti lepšieho konfigurovania a škálovania IS, čo vyplýva z hlavných rysov agentových paradigiem. Návrh sa zjednodušuje tým, že inteligentným JADE agentom možno ľubovoľne počas vykonávania systému (v ktorom figurujú) pridávať požadovaný počet a typ jednotlivých správanií. Preto nebudú mať prípadné zmeny vo vykonávaní agenta veľký vplyv na mnohé oblasti návrhu a implementácie, ale sa budú dotýkať len daného správania príslušných agentov. Podobne ak by sme chceli škálovať agentový systém, resp. agentovú vrstvu IS, je oveľa jednoduchšie pridávať nezávislé agenty, nakoľko tie nie sú previazané – oproti objektom tried a komponentov – rozličnými reláciami na ďalšie objekty na rôznych miestach. Ešte skôr, ako prejdeme k aplikácii agentov pri monitorovaní a modifikovaní výkonu, ozrejmime pojem „vyhodnocovač“.

## Vyhodnocovač

Šetrenie systémových prostriedkov počítačového systému hrá pri navrhnutých architektúre (Intelligent Component-Based Architecture – ICBA) kľúčovú úlohu, nakoľko bude mať vplyv na čas vykonania určitej



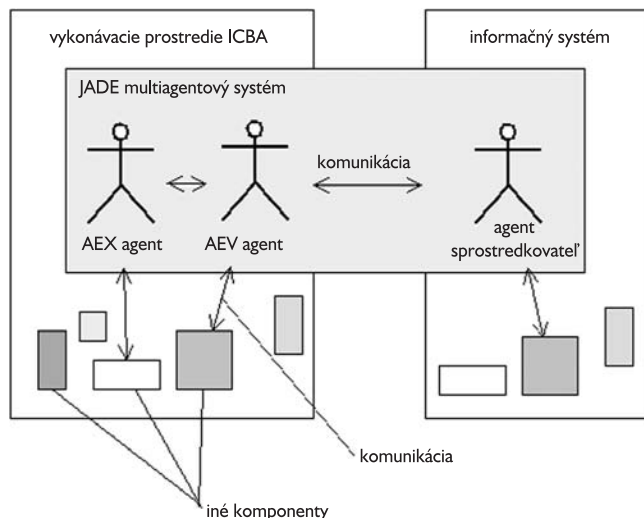
Obr.2 Entitno-relačný diagram databázy IRE

služby IS (v užšom zmysle slova operácie komponentu) pre ostatných používateľov a v konečnom dôsledku vyústi do reakčného času celého IS. ICBA sa zameriava na šetrenie centrálnej procesorovej jednotky, keďže je jedným z kľúčových systémových prostriedkov. Vyhodnocovač (Evaluator) je v kontexte ICBA proces tvorený konečnou postupnosťou určitých krokov a vyhodnocujúci pomer medzi časom vykonania  $t_E$  a časom  $t_S$ , počas ktorého je vlákno vykonávajúce požadovanú operáciu, resp. skupinu operácií suspendované. Tieto hodnoty tvoria základ modifikácie výkonu inteligentnými agentmi vo vykonávacom prostredí ICBA – na ich základe sa vypočíta konkrétna hodnota pre periódu suspendovania  $P_{sus}$  a čas  $t_{sus}$ , počas ktorého bude vlákno suspendované. Vyhodnocovač je aplikovaný do správania (behaviours) inteligentných agentov typu AEV (o ktorých bude reč v ďalšej sekcii), nakoľko za vyhodnotenie  $t_E$ ,  $t_S$ , a výpočet  $P_{sus}$  a  $t_{sus}$  zodpovedajú práve ony. Každé správanie AEV agenta „obsahuje“ iba jeden vyhodnocovač (presnejšie kód, ktorý iniciuje jeho vykonanie) a ďalej už nič. Vyhodnocovač je v podstate vykonávateľná softvérová jednotka, ktorá vyhodnocuje percentuálny pomer medzi  $t_E$  a  $t_S$  na základe údajov nachádzajúcich sa v databáze vykonávacieho prostredia ICBA (obr. 2), pričom nemusi brať do úvahy všetky monitorované veličiny, resp. faktory, ale iba ich časť. Táto databáza obsahuje tabuľky, ktorých údaje sa vzťahujú na:

- jednotlivých používateľov – žiadateľov,
- vyhodnocovače, ktoré sú používané AEV agentmi,
- údaje o jednotlivých volaniach určitej operácie, resp. skupiny operácií,
- dostupné operácie na určitých komponentoch a zostavené skupiny operácií,
- štatistiky – asociované s používaním, resp. „vyťažovaním“ jednotlivých komponentov, dostupných operácií a skupín operácií.

**Aplikácia agentov**

Pri realizovaní ICBA sa používa multiagentový systém (Multiagent System for ICBA – MASI) premostujúci vykonávacie prostredie a IS, z ktorého migrujú žiadosti o vykonanie určitej operácie, resp. skupiny operácií (obr. 3). MASI je realizovaný na báze JADE-LEAP pre platformu



Obr.3 Multiagentový systém premostujúci IS a vykonávacie prostredie ICBA

.NET. Dôvod, pre ktorý sa používa platforma JADE-LEAP pre .NET, spočíva v skutočnosti, že AEV agenti používajú na realizáciu svojho správania vyhodnocovače, ktoré sú implementované prostredníctvom technológie WWF [4], vzhľadom na jej schopnosť výstižne vyjadriť pomocou pracovného toku určitý proces, resp. postupnosť aktivít vrátane ich typov a spôsobu ich vykonávania. Tento multiagentový systém vytvára pre každého používateľa trojicu nasledujúcich inteligentných agentov (pričom prvý z nich sa nachádza na strane IS, ostatné na strane vykonávacieho prostredia ICBA):

- Mediator Agent – je vytvorený pre každého používateľa IS a slúži ako prostredník medzi IS a vykonávacím prostredím ICBA.
- Agent for Evaluation (AEV), resp. AEV agent – komunikuje s agentom Mediator a prijíma od neho požiadavky na vykonanie. Po analýze požiadavky sa tomuto agentovi vložia, resp. zaradia do jeho plánovača správania iniciujúce vykonanie príslušných typov vyhodnocovačov (v závislosti od typu požiadavky). Po vyhodnotení  $t_E$  a  $t_S$  sa vypočítajú hodnoty periódy suspendovania  $P_{sus}$  a času  $t_{sus}$ . Tie sa odovzdávajú nasledujúcemu typu agenta.
- Agent for Execution (AEX), resp. AEX agent – prijíma vypočítané hodnoty  $P_{sus}$  a  $t_{sus}$  od AEV agenta. Na základe týchto hodnôt sa bude modifikovať výkon (kap. 5) a po vykonaní sa v databáze IRE doplnia a aktualizujú údaje súvisiace s vykonaním operácie, resp. GOO. Potrebné inštancie, resp. popisovače statických metód, preberá od IRE.

**Literatúra**

(vybrané tituly)

[1] CAIRE, G., PIERI, F.: Leap User Guide, TILAB, 20. 2006  
 [2] Java Agent Development Framework (JADE) Website. <http://jade.tilab.com/>, 2007.  
 [3] JENNINGS, N. R., WOOLDRIDGE, M.: Agent-Oriented Software Engineering. In: Handbook of Agent Technology (ed. J. Bradshaw) AAAI/MIT Press.  
 [4] Windows Workflow Foundation Website. <http://wf.netfx3.com>, 2006

Ing. Miroslav Beličák

Univerzita Karlova v Praze  
 Matematicko-fyzikální fakulta  
 Katedra softwarového inženýrství  
 Malostranské nám. 25  
 118 00 Praha 1, ČR  
 e-mail: Miroslav.Belicak@gmail.com