

Agenți software

Definire, clasificare și utilizare – Sabin-Corneliu Buraga

Încă din cele mai vechi timpuri, oamenii au fost fascinați de ideea de a crea entități artificiale care să îndeplinească diverse funcții. Existența automatelor poate fi urmărită de-a lungul secolelor, cei mai relevanți predecesori ai agenților software de astăzi fiind servomecanismele, dispozitivele de control și de automatizare a activităților de ghidare și supraveghiere a traficului aerian.

Inițial, termenul de agent apare la mijlocul anilor '50, propus de John McCarthy, apoi de Oliver Selfridge, ambii de la MIT. Cercetările în acest domeniu sunt împărțite în două mari stadii: primul începând prin 1977, al doilea debutând în anii '90. Primul stadiu își are rădăcinile în inteligența artificială extinsă la mediile distribuite, concentrându-se în crearea agenților pe baza modelelor de raționament simbolic, cu contribuții în interacțiunea și comunicația între agenți, divizarea și distribuirea activităților, coordonarea și cooperarea, negocierea informațiilor etc. Al doilea val, având o dezvoltare explozivă în ultimii ani, studiază tipologia agenților și modul de conlucrare în rețea, punându-se mai mult accent pe mobilitate.

La cristalizarea teoriei și metodologiei agenților software au contribuit însă, alături de inteligența artificială, și programarea orientată-obiect, sistemele obiectuale concurente și proiectarea interfețelor om-calculator.

Agenții își găsesc utilizări variate în domenii precum inteligența artificială, robotica, interacțiunea om-mașină, calculul distribuit, interfețele inteligente și adaptabile, căutare pe Web, achiziția de cunoștințe și altele.

Încercare de definire

Definițiile asupra termenului de agent sunt împărțite, fiind socotiți agenți software atât asistenții „inteligenti”, cât și programele implementând funcții cognitive sau având capacitatea de migrare de la un calculator la altul.

În multitudinea de încercări de definiții pentru agenți se detașează două abordări distincte, dar înrudite, una bazată pe noțiunea de agent privit ca o atribuire de identitate comportamentală unei componente soft, cealaltă bazată pe descrierea unor atribute pe care agenții le posedă.

Agenții ca entități comportamentale. Termenul de *agent* derivă din limba latină,

provenind de la participiul verbului *agere*: a conduce, a interpreta, a face. Se așteaptă de la agenți un comportament inteligent, dar pentru aceasta aceștia trebuie să „cunoască” anumite informații despre contextul situației cu care sunt confrunțați.

Astfel, putem defini agenții software ca fiind sisteme informaționale care se comportă asemeni unor alte entități într-o manieră autonomă, execută diverse acțiuni având un anumit nivel de reacție și etalează atribute precum învățare, cooperare și mobilitate, asistând utilizatorii în activitățile lor.



Se pot descrie trei atitudini predictive pe care oamenii le au în plus față de sistemele informaționale:

- *atitudinea fizică* bazată pe caracteristici și legi fizice;
- *atitudinea de proiectare* bazată pe ceea ce se proiectează în prealabil să se realizeze;
- *atitudinea intențională* bazată pe presupunerea existenței unei agenții raționale.

Pentru sistemele naturale (situația coliziunii bilelor de biliard) oamenii pot prevedea comportamentul pe baza caracteristicilor și legilor fizice. Dacă înțelegem suficient de bine un sistem proiectat (un automobil, de pildă), atunci putem să ne imaginăm care ar fi comportamentul lui. Pentru a ne explica comportamentul mulțimilor (de oameni, animale, roboți sau agenți), ne folosim de descrierea comportamentului colectiv, al așa-numitei agenții raționale.

Astfel, agenții pot fi priviți ca sisteme intenționale. Intențiile agenților pot fi văzute ca o combinație între *decizii* și *scopuri*, punându-se bazele unei teorii a interacțiunii dintre agenți. O altă abordare formală a noțiunilor mentale asociate agenților este cea fundamentată pe logici multi-modale, în care primitivele considerate sunt *convingerile*, *dorințele* și *intențiile* - logici BDI (Belief, Desire, Intention).

Agenții ca descriere a atributelor lor. O definiție pe care cercetătorii o găsesc mai acceptabilă este următoarea: agenții sunt entități software posedând funcții comportamentale, într-o manieră autonomă și continuă în medii colective, compuse din alți agenți și procese.

Cerința de a asigura autonomia și continuitatea este dată de dorința programatorilor ca agenții proiectați să fie capabili să acționeze într-un mod flexibil și, pe cât posibil, inteligent, adaptându-se situațiilor fără aport uman. În mod ideal, un agent ar trebui să poată învăța din propria lui experiență și să poată dezvolta tehnici de comunicare și de cooperare cu alți agenți similari sau să poată manifesta mobilitate.

Se pot enumera o serie de atribute (caracteristici) care descriu agenții:

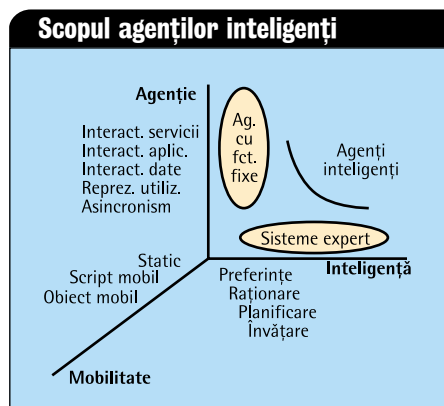
- *reacția*: abilitatea de a se comporta în mod selectiv
- *autonomia*: comportament direcționat spre scop, independent de utilizator
- *colaborarea*: posibilitatea de a lucra împreună cu alți agenți pentru îndeplinirea unui scop comun
- *comunicarea*: abilitatea de a comunica cu persoane sau alți agenți prin intermediul unui limbaj de nivel înalt
- *inferența*: posibilitatea de a opera conform unei specificații abstracte, utilizând cunoștințe anterioare despre scopul urmărit și situațiile survenite
- *continuitatea*: persistența în timp a identității și stării
- *personalitatea*: capacitatea de a manifesta atribute umane „credibile”, ca de exemplu emoție
- *adaptabilitatea*: posibilitatea de a învăța și de a se dezvolta ținând cont de experiența acumulată și de versatilitate în rezolvarea unor situații inedite
- *mobilitatea*: abilitatea de a migra de la sine de pe o platformă pe alta

Clasificări ale agenților

Luând în considerație cele descrise mai sus, există mai multe modalități de clasificare a agenților, prin prisma inteligenței artificiale, a calculului distribuit sau a tipurilor de software. Se oferă și o taxonomie a agenților.

Din prisma inteligenței artificiale. În cadrul comunității specialiștilor în inteligența artificială, agenții au fost caracterizați pe baza măsurării capabilității de rezolvare a problemelor. Astfel, un agent *reactiv* este unul care posedă reacții la modificări ale mediului sau ale mesajelor receptate de la alți agenți. Un agent *intențional* este capabil să-și elaboreze comportamentul în funcție de „intenții” și de „convingeri”, să creeze planuri de acțiune și să le execute. Un agent *social* urmărește diverse modele inspirate din societatea umană și interacționează cu alți agenți, în cadrul agenției.

Agenții inteligenți pot fi descriși în termenii unui spațiu tridimensional definit prin dimensiunile de *agenție*, *inteligență* și *mobilitate* (vezi figura de mai jos).

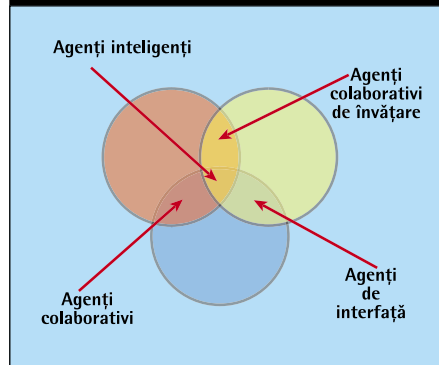


Agenția reprezintă gradul de autonomie și autoritate de care se bucură un agent și poate fi măsurată calitativ de natura interacțiunilor dintre agent și alte entități prezente în sistem. Un agent poate rula în mod asincron. Gradul agenției crește dacă agentul îl reprezintă pe utilizator într-un anumit mod. Un agent mai avansat poate interacționa cu datele, cu aplicațiile, cu serviciile ori cu alți agenți. Agenția poate fi privită ca areal de operare al agenților.

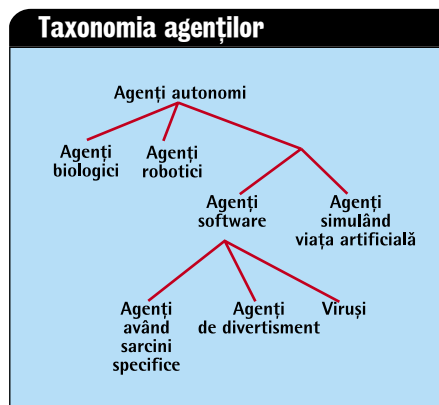
Inteligența reprezintă gradul de raționare și de auto-învățare, adică abilitatea de a accepta și înțelege scopurile utilizatorului și de a le îndeplini, independent de acesta. Cerința minimă este posibilitatea agentului de a opera cu un set de preferințe ale utilizatorului. Nivelele superioare de inteligență includ modelele de raționament, posibilități de *învățare* și *adaptare* la mediu etc.

Mobilitatea furnizează gradul de migrare a agenților în rețea. *Scripturile mobile* pot fi concepute pe o mașină și trimise altui calculator spre execuție. *Obiectele mobile* sunt transportate din gazdă în gazdă în timpul execuției, acumulând diverse date despre mediul în care operează și pe care pot să-l modifice.

Clasificarea agenților după atributele primare



Clasificare pe baza atributelor primare. În anul 1996 se propune o tipologie bazată pe atributele primare ale agenților, putând fi clasificați conform:



- mobilității: *statistici* sau *mobili*
- prezenței unui model de raționament simbolic: *deliberativi* sau *reactivi*
- atributelor ideale sau primare: *autonomi*, *cooperativi*, *de învățare*. Se pot descrie patru tipuri de agenți: *colaborativi*, *colaborativi de învățare*, *de interfață* și *inteligenți* (vezi figura următoare)
- rolurilor: informativi sau distribuți pe Internet
- filosofilor hibride combinând mai multe abordări: *agenți mixti*
- atributelor secundare: versatili, continui temporal, emoționali etc.

Conform acestei clasificări, agenții pot fi divizați în șapte categorii: *colaborativi*, *de interfață*, *mobili*, *informativi/Internet*, *reactivi*, *hibridi* și *inteligenți*

În raport cu alte tipuri de software. Cea de a treia încercare de clasificare divide agenții în funcție de structurile de control, de mediul de operare (bază de date, sistem de fișiere, rețea, Internet), de limbajul de programare sau de aplicații. Taxonomia generală a agenților este ilustrată în figura cu același titlu.

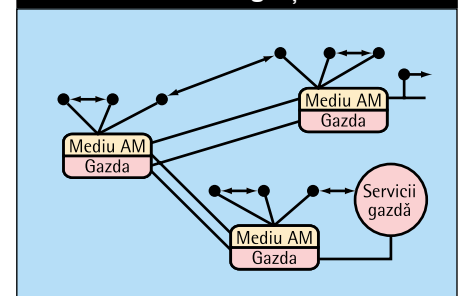
Utilizări

În ultima perioadă au apărut diverse abordări în implementarea agenților, utilizările lor urmărind mai ales rațiuni de natură practică: simpli-

ficarea sistemelor distribuite de calcul și depășirea limitărilor interfețelor-utilizator actuale.

Simplificarea calculului distribuit. Agenții mobili. La sfârșitul anilor '80 se punea în discuție nevoia de *interoperabilitate inteligentă* a sistemelor software. În prezent și în viitor, mediile distribuite sunt și vor fi compuse din entități eterogene, care vor rula pe platforme multiple, având diverse configurații. În majoritatea cazurilor, componentele acestor medii nu comunicau și nu cooperau între ele decât într-un mod primitiv (prin intermediul transferului de fișiere, a interogărilor de baze de date etc.). Actualmente pe Web sistemele pot comunica instituind conexiuni particulare *ad-hoc*, respectând diverse protocoale de comunicație bazate pe standarde precum TCP/IP, IIOP sau ODBC, care oferă un nivel de bază de încapsulare a conectivității între sisteme și servicii.

Arhitectura de bază a unui sistem de agenți mobili



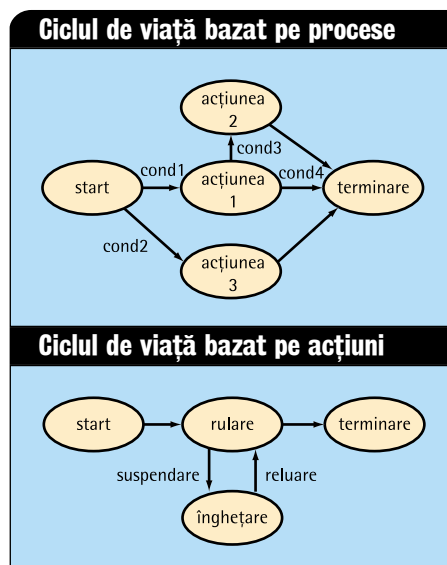
Se observă în mod evident o tranziție de la rețele locale izolate, controlate de sisteme de operare ca Novell sau Windows, la sisteme de calcul intranet sau Internet, la nivel planetar. Aceasta conduce la proliferarea serviciilor de rețea independente de sistemul de operare, precum servicii de nume, de directoare sau securitate (implementate prin tehnologii ca Java, ActiveX sau, mai nou, beneficiind de serviciile platformei Microsoft NET). Soluția se întrevede în utilizarea *programării orientate-agent*.

Astfel, o primă direcție este utilizarea agenților ca administratori de resurse. Un nivel ridicat de interoperativitate în Internet va necesita colectarea de informații despre capabilitățile fiecărui sistem, pentru asigurarea planificării securității, alocarea resurselor, execuția și monitorizarea activităților și, probabil, intervenția automată pentru configurare și administrare. Toate aceste funcții pot fi îndeplinite de un agent mobil sau de o agenție de astfel de agenți. Arhitectura de bază poate fi ilustrată de figura 4:

Un *agent mobil* este o entitate software care rulează într-un mediu software, conținând un model agent, un model al ciclului de viață, un model computațional, un model de securitate, un model de comunicație și un model de navigare.

Mediul agenților mobili reprezintă un sistem software distribuit peste o rețea compusă

din calculatoare eterogene. Sarcinile lui primare sunt să ofere un mediu de execuție pentru agenții mobili, implementând majoritatea modelelor care apar în definiția agentului. De asemenea, mediul pune la dispoziție servicii de suport pentru interactivitatea agenților mobili cu mediul și pentru accesarea altor sisteme (orientate sau nu agent).



După cum se observă din figura de mai sus, mediul agenților mobili (AM) este construit peste sistemul gazdă pe care vor evolua agenții, traversând mediile. Comunicațiile pot avea loc între agenți locali sau aflați la distanță ori între un agent mobil și serviciile oferite de gazdă.

Modelul agent. Acest model definește structura internă a unui agent inteligent ca parte a unui agent mobil, în esență specificând caracteristicile de autonomie, auto-în-vățare și cooperare, plus natura reactivă și proactivă a agenților.

Modelul implică o arhitectură software paralelă: fiecare agent are o viață autonomă, putând face parte însă din grupuri independente sau organizate într-o manieră ierarhică.

Modelul ciclului de viață. Modelul definește diferite stări de execuție ale unui agent mobil și evenimentele care determină tranzițiile de stare. Acest model este înrudit cu modelul computațional care descrie cum are loc execuția. În prezent, modelele cele mai importante ale ciclului de viață sunt cele bazate pe *procese persistente* (model adoptat de Telescript și AgentTcl) sau pe *acțiuni* (cum ar fi apleții creați de IBM). Baza formală a acestor modele este dată de teoria automatelor sau a rețelelor Petri.

Modelul computațional. Modelul de calcul specifică semantica execuției agentului mobil, atunci când este în stare de rulare. Calculul are loc în cadrul unui mediu și este facilitat fie de un procesor real, fie de unul abstract așa cum se regăsește în mașina virtuală Java (JVM -

Java Virtual Machine). Modelul computațional definește un set de instrucțiuni primitive specificând abilitățile de calcul ale agenților (e.g. instrucțiuni de manipulare a datelor și instrucțiuni de control al firelor de execuție).

În cadrul acestui model se pune problema programării agenților, dezvoltându-se diverse limbaje special concepute pentru dezvoltarea și implementarea sistemelor multi-agent. Astfel, sunt folosite limbaje precum Actor, Tcl/Tk, Telescript, Linda (utilizate în special pentru agenți mobili) și Agent0, Concurrent Metatem, KQML (destinate modelării agenților cognitivi).

Modelul securității. Problema securității agenților mobili se poate divide în două arii largi: protecția nodurilor de rețea de acțiunea agenților mobili distructivi și protecția agenților de gazdele pe care rulează.

Un sistem de agenți mobili este un sistem deschis, predisus la atacuri variate: acces la și/sau alterări de date de către entități neautorizate, utilizare neautorizată de diferite facilități ale agenților și interferențe în activitatea sistemului de către agenții neautorizați. Împotriva atacurilor se pot folosi tehnici precum criptografia, autentificarea, semnăturile digitale, ierarhiile de încredere.

Un agent mobil are acces la anumite resurse ale gazdei pe care rulează, putând altera activitățile altor agenți locali, propagând viruși, cai troieni sau viermi ori atacând anumite servicii publice. Soluțiile la astfel de probleme sunt rezumate de trei mecanisme: securitatea proceselor, securitatea sistemului și securitatea rețelei.

Pe partea cealaltă, un sistem gazdă poate avea informații despre un agent, putându-i modifica starea sau codul ori făcându-l neoperational. Problema critică este că agentul mobil este total neprotejat în fața sistemului pe care evoluează, fiind imposibil să se protejeze un agent de gazda sa.

Modelul comunicării. Pentru a fi efectivi, agenții mobili trebuie să comunice cu alte entități prezente în mediul de calcul: utilizatori, alți agenți (statici sau mobili), mediul de management al agenților ori alte sisteme distribuite (e.g. CORBA). Comunicațiile sunt folosite pentru accesarea serviciilor externe, în vederea cooperării și coordonării agenților sau între agenți și alte entități.

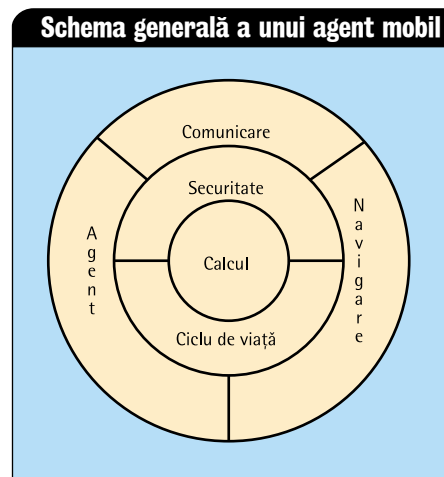
Modelul de comunicare este implementat de un protocol, ca exemple putând fi date tehnici de execuție a procedurilor la distanță (RPC - *Remote Procedure Call*) sau limbajele de comunicare inter-agenți precum KQML (*Knowledge Query Manipulation Language*) sau COOL - un limbaj de coordonare a agenților.

Comunicarea între agenți poate varia de la forme simple (exprimată prin intermediul *semnalelor* cu interpretări fixe) până la cele sofisticate (bazate pe *actele comunicării* - *speech acts*).

O formă intermediară este cea bazată pe schimbul de informații printr-o structură de tip *tablă* (*blackboard*) sau de resursă partajată, tabla fiind divizată în diverse nivele de abstractizare corespunzătoare nivelelor de rezolvare a problemelor, iar agenții pot scrie sau pot citi informațiile de interes la nivelul asociat acțiunilor lor.

Pornind de la cercetări întreprinse în domeniul limbajului natural, pentru modelarea comunicațiilor dintre agenți se folosește *teoria actelor comunicării* (*theory of speech acts*). Într-un act de comunicare se face distincția între aspectele *elocutionale* (exprimarea unei afirmații), *illocutionale* (efectul intenționat de cel care comunică) și *prelocutionale* (efectul asupra destinatarului comunicării). Aceste modele ale dialogului uman se pot utiliza și în cazul comunicării dintre agenți. Protocolul de comunicare nu desemnează un protocol de nivel scăzut (așa cum se întâmplă de regulă în cazul comunicațiilor în sistemele distribuite), ci un protocol stabilind acțiunile posibile ale agentului în fiecare moment al comunicării cu alte entități.

Limbajul KQML amintit mai sus se bazează pe teoria actelor comunicării, fiind propus în 1992 și standardizat în anul 1997. KQML folosește limbajul KIF (*Knowledge Interchange Format*) pentru descrierea conținutului informațional al mesajului. KIF exprimă logica cu predicate de ordin întâi, având o sintaxă apropiată de limbajul Lisp.



Modelul navigării. Acest model se concentrează pe mobilitatea agenților în vederea descoperirii surselor de destinație pe care să-și execute codul. În prezent direcțiile de cercetare, având ca suport teoretic lambda-calculul sau pi-rețelele, se concentrează asupra unor aspecte ca:

- stabilirea unor convenții de numire a entităților dintr-un sistem al agenților mobili (agenți, calculatoare, servicii și altele);
- accesul la informații privind mediile de agenți mobili aflate la distanță;

- abilitatea de a muta un agent mobil aflat în stare de suspendare pe o altă mașină în vederea rulării;
- abilitatea de a primi un agent mobil suspendat și de a-l reconstitui într-un nou mediu computațional.

Există și diverse servicii adiționale care facilitează navigarea:

- servicii director și de referință a resurselor care pot ghida agenții să descopere servicii relevante pentru activitățile care trebuie realizate (X.500, CORBA Trader, LDAP, Active Directory);
- servicii de informare asupra topologiei de rețea.

Aceste modele duc la următoarea diagramă care sumarizează structura generală a unui agent mobil:

Avantaje ale tehnologiei agenților mobili

Eficiența. Agenții mobili consumă mai puține resurse de rețea, prin migrarea codului și mai puțin a datelor (spre deosebire de paradigma RPC).

Reducerea traficului. Cele mai multe protocoale de comunicație necesită interacțiunii de date între diverse calculatoare, conducând la un trafic însemnat pe rețea. Interacțiunile în cazul agenților mobili se pot desfășura local sau prin intermediul mediului agenților.

Interacțiunea autonomă asincronă. Activitățile pot fi codate în cadrul agenților mobili și apoi executate asincron și independent de gazdă.

Interacțiunea în timp-real. Entitățile lucrând în timp-real necesită flexibilitate sporită la modificările mediului, agenții mobili constituind o alternativă viabilă.

Adaptabilitatea. Agenții mobili posedă abilități de a-și modifica starea și comportamentul în funcție de mediul unde evoluează.

Robustetea și toleranța la defecte. Un mediu colaborativ compus din agenți mobili este mult mai robust și fiabil decât alte abordări de sisteme distribuite.

Suportul pentru medii eterogene. Sistemele bazate pe agenți mobili sunt în general independente de rețea și de platforma hardware/software.

Suportul pentru comerț electronic. Agenții mobili pot fi folosiți la construirea piețelor electronice și la mijlocirea tranzacțiilor comerciale în rețea.

Implementări

Tehnologia agenților mobili a început să se dezvolte începând cu anul 1994, lista celor mai importante limbaje utilizate incluzând Java, Tcl/Tk și Obliq (și, mai nou, Visual Obliq) ca suport de dezvoltare a unor cadre de operare a agenților sau Telescript, Mole, Ara, Tacoma ca soluții academice sau comerciale, pentru a enumera doar câteva.

Probabil soluția de viitor va fi reprezentată de limbajul Java ca abordare independentă de mașină, neutră și orientată-obiect, pentru calculul distribuit. Caracteristici precum execuția de fire de execuție multiple sincronizate, capabilități de operare la nivel înalt în rețea, facilități de invocare a metodelor la distanță (RMI - Remote Method Invocation) și de serializare a obiectelor, acordă un punctaj bun în considerarea limbajului Java drept favorit în conceperea unui mediu orientat-agent.

În strânsă legătură cu Java putem menționa *agletii*, obiecte Java mobile care pot fi executate pe diverse calculatoare în Internet. Un *aglet* poate să-și întrerupă execuția, să-și salveze starea (prezervându-și mediul de lucru) și să migreze pe altă gazdă unde să-și continue activitatea din momentul opririi. *Agletii* încapsulează în ei înșiși atât codul program (instrucțiuni ale mașinii virtuale Java), cât și datele, fiind din punct de vedere conceptual agenți mobili deoarece suportă execuția autonomă și dirijarea dinamică spre gazde. Un *aglet* moștenește toate facilitățile unui sistem Java.

Alte capabilități ale *agletilor* sunt enumerate în continuare:

- o schemă globală unică de numire a agenților pentru identificarea lor în rețea (model de navigare și securitate);
- un itinerariu de migrare în rețea permițând destinații multiple și detecția automată a întreruperii conexiunii (model de navigare);
- un mecanism de colaborare și de partajare a informațiilor în mod asincron (model al comunicării);
- o schemă de transmitere a mesajelor suportând asincronismul și comunicațiile punct-la-punct sincrone între *agletii* (model al comunicării);
- un încărcător de clase de agenți de rețea permițând codului Java și a informațiilor de stare să migreze prin Internet (model de navigare);
- un context de execuție oferind un mediu uniform independent de arhitectura de sistem (model computațional).

Din păcate, *agletii* posedă un model al ciclului de viață foarte primitiv, iar divizia japoneză a IBM a renunțat în prezent la continuarea cercetărilor în acest domeniu.

Alternative

Există o serie de alternative la paradigma agenților mobili, dintre care putem enumera *evaluarea la distanță (remote evaluation)* și *codul-program la cerere (code on demand)*.

Domnul Sabin-Corneliu Buraga este doctorand în Computer Science la Universitatea „Al.I.Cuza” din Iași și poate fi contactat la adresa busaco@infoiasi.ro sau pe Web la http://www.infoiasi.ro/~busaco. ■ 87

Referințe bibliografice

- B.Alwyn - „An Overview of an Agent-Oriented Language for Concurrent and Distributed Programming”, ROSYCS'96 Proceedings, Iasi, 1996
- M.Barbuceanu, M.Fox - „Integrating Communicative Action, Conversations and Decision Theory to Coordinate Agents”, Agents'97 Proceedings, ACM, 1997
- M.Barbuceanu, R.Teigen, M.Fox - „COOL - un limbaj de coordonare a agentilor”, PC-Report, vol7/1 (64), ian.1998
- K.Bharat, L.Cardelli - „Visual Obliq - Distributed Applications In A Hypermedia Setting”, CHI'95 Video Proceedings, 1995
- J.Bradshaw - „Software Agents”, AAAI Press, 1997
- A.Cheyer, L.Julia - „InfoWiz: An Animated Voice Interactive Information System”, SRI International Artificial Intelligence Center, Menlo Park, 1999
- L.Daigle et al. - „Uniform Resource Agents”, IETF Internet Draft, 1995
- A.Florea - „Sisteme multi-agent inteligente”, PC-Report, vol7/1 (64), ian.1998
- S.Green, F.Somers - „Software Agents: A Review”: http://www.cs.tcd.ie/research_groups/aig/iag/iag.html
- K.Höök et al. - „Edited Adaptive Hypermedia: Combining Human and Machine Intelligence to Achieve Filtered Information”, Flexible Hypertext Workshop, Hypertext'97 Proceedings, ACM, 1997
- E.Horvitz - „Lumiere Project: Bayesian Reasoning for Automated Assistance”, Decision Theory Et Adaptive Systems Group, Microsoft Research. Microsoft Corp. Redmond, WA: 1998: <http://research.microsoft.com/research/dtg/horvitz/lum.htm>
- N.Jennings et al. - „A Roadmap of Agent Research and Development”, Autonomous Agents and Multi-Agent Systems, no.1, Kluwer Academic, Boston, 1998
- P.Maes - „Agents that Reduce Work and Information Overload”, Communications of the ACM 37(3), 1994
- D.Martin et al. - „The Open Agent Architecture”, Applied Artificial Intelligence Journal, vol.13, ian.-mart.1999
- C.Petrie - „Agent-Based Engineering, the Web and Intelligence”, IEEE Expert, dec.1996
- D.Weerasooriya et al. - „Design of a Concurrent Agent-Oriented Language”, in „Intelligent Agents: Theories, Architectures and Languages”, LNAI890, Springer Verlag, Amsterdam, 1995
- M.Wooldridge, N.Jennings - „Intelligent Agents: Theory and Practice”, Knowledge Engineering Review Journal, 10(2), 1995
- *** - „AgentWeb”: <http://www.cs.umbc.edu/agents>
- *** - „Microsoft Research”: <http://research.microsoft.com>