

CAPITOLUL IX

APLICAȚII ALE AGENȚILOR INTELIGENȚI ȘI A SISTEMELOR MULTIAGENT

În secțiunile ce urmează vom prezenta câteva domenii de utilizare ale agenților inteligenți și a sistemelor multiagent în anumite probleme din lumea reală.

9.1. Agenți Inteligenți în comerț electronic (*e-commerce*)

În ultima vreme, agenții inteligenți sunt deosebit de utili în domeniul afacerilor electronice (*e-business*), cu subdomeniile acesteia: comerț electronic (*e-commerce*), servicii bancare electronice (*e-banking*), învățământ electronic (*e-learning*), jocuri de noroc electronice (*e-gambling*), franciză electronică (*e-franchise*).

Din 1993 este operațional sistemul FAIS, care se referă la domeniul tranzacțiilor financiare. Sistemul a fost proiectat și implementat ca un sistem cooperativ incluzând oameni și agenți software.

Wenger și Probst au analizat posibilul rol și impact al paradigmei bazate pe agenți în domeniul general al furnizării serviciilor financiare, discutând diverse modalități în care agenții inteligenți ar putea fi folosiți pentru: operațiuni financiare, tranzacții, segmentul de piață.

Comerțul electronic se referă la partajarea informațiilor despre afaceri, gestionarea relațiilor de afaceri și conducerea tranzacțiilor de afaceri prin intermediul rețelelor de comunicare. Comerțul electronic include relații între companii (*business to business*), relații între clienți (*customer to customer*) precum și relații între companii și clienți (*business to customer*).

Tehnologia agenților inteligenți este folosită în diversele etape ale experienței de vânzare:

- **Identificare.** Agenții monitorizează o mulțime de senzori sau fluxuri de date și efectuează anumite acțiuni atunci când apare o anumită condiție prespecificată. Spre exemplu, există agentul **Eyes** al site-ului **Amazon.com**, care monitorizează cataloagele de cărți de vânzare și notifică cumpărătorii atunci când anumite evenimente sunt interesante pentru cumpărător (când o anumită carte din categoria **X** devine disponibilă).
- **Operații de brokering.** Agenții sunt folosiți pentru a indica cumpărătorilor (clienților) acele produse care satisfac cel mai bine cerințele lor.
- **Negocieri.** Agenții sunt folosiți pentru negocierea prețurilor și altor aspecte ale tranzacțiilor.
- **Plată și livrare.** Agenții sunt folosiți pentru a monitoriza diferitele opțiuni de plată și livrare ale produselor.

- **Evaluarea serviciilor.** Agenții sunt folosiți pentru a evalua serviciile de producție, serviciile oferite consumatorilor precum și o evaluare a experienței de cumpărare și decizie.

În continuare vom da câteva exemple de agenți folosiți în comerț electronic: **PersonalLogic** (permite cumpărătorilor să găsească cele mai potrivite produse - <http://www.personallogic.com>), **BergainFinder** (primul agent de cumpărare virtual pentru interogare prețuri de CD-uri muzicale - <http://bf.cstar.ac.com/bf>), **Kashab** (un sistem tranzacțional on-line, multi-agent - <http://kashab.media.mit.edu>).

9.2. Agenți Inteligenți în servicii bancare electronice (*e-banking*)

În prezent, metodele de plată electronică joacă un rol foarte important în toate formele de business on-line. În ciuda dezvoltării sistemelor bancare electronice, care permit un anumit grad de automatizare, interfețele Web necesită totuși un grad destul de ridicat de interacțiune umană. Un mediu bazat pe agenți care furnizează servicii bancare electronice suportă diferite tranzacții în medii software.

Tranzacțiile și ordinele de plată, spre exemplu, pot fi efectuate electronic, dar numai dacă omul introduce codul corect sau dacă apasă butonul corect de pe interfața grafică specifică.

Analizii prezic faptul că în viitorul apropiat oamenii nu vor mai merge la bănci și nici nu se vor loga la Internet pentru a-și rezolva problemele bancare. Ca o alternativă, oamenii vor delega gestionarea conturilor bancare, plăților, investițiilor, asigurărilor unui **asistent financiar electronic**.

Ideea de bază este că entități software vor acționa în folosul oamenilor și/sau furnizorii de servicii pot automatiza anumite afaceri electronice sau activități comerciale. Ca urmare, se poate imagina o instituție virtuală (**IV**) care oferă servicii bancare agenților care accesează serviciile unei piețe bancare. Este vorba de fapt despre un agent (instituție bancară virtuală) care oferă servicii agenților care accesează piața.

Aceste servicii pot fi de două feluri:

- plăți electronice care să permită agenților să facă/primească plăți;
- gestionare de conturi pentru crearea/păstrarea/închiderea de conturi bancare.

Rolurile agenților

Sunt trei tipuri principale de agenți într-o aplicație de e-banking:

- **Agent Bancar (AB)** - acționează în favoarea băncii. Poate fi considerat ca fiind interfața între **IV** și lumea exterioară;
- **Agent Personal (AP)** - poate fi considerat ca un asistent personal care acționează în scopul unui utilizator și folosește serviciile oferite de piața bancară;
- **Agent de Asigurare (AA)** - reprezintă o afacere de asigurare, oferind vânzarea unor polițe de asigurare (spre exemplu, plățile cu credit card-uri).

În continuare, vom considera un scenariu de plată cu card de credit.

Scenariu

AP-X (o instanță a agentului AP) cere o poliță de asigurare lui AA-T de 500 RON. AP-X își trimite informațiile despre cardul de credit agentului AA-T care cere instituției IV să transfere 500 RON din contul lui AP-X în contul lui AA-T. Atât AP-X cât și AA-T au conturi cu AB (același agent bancar gestionează ambele conturi) și sunt informați despre tranzacție.

Sarcinile agenților în cazul plății cu card de credit sunt următoarele:

- **faza de negociere** - se termină cu succes dacă cei doi agenți cad de acord asupra poliței de asigurare, incluzând și prețul. Când se ajunge la o înțelegere, AP-X (cumpărător) va plăti pe AA-T (vânzător) și vânzătorul va trebui să furnizeze serviciul cumpărătorului. Plata se va realiza folosind serviciile lui AB;
- **începerea procedurii de plată** - AA-T ia informațiile necesare despre contul lui AP-X și cere lui AB să transfere suma în contul său. AB determină tipul transferului (în cazul nostru intrabancar);
- **efectuarea plății** - AB verifică dacă în contul lui AP-X sunt suficienți bani pentru a face plata. În caz afirmativ, se face plata și AP-X și AA-T vor fi notificați despre tranzacție. Dacă nu sunt suficienți bani în cont, ambii agenți vor fi notificați.

În Figura 9.1 sunt descrise interacțiunile între cei trei agenți în scenariul descris anterior.

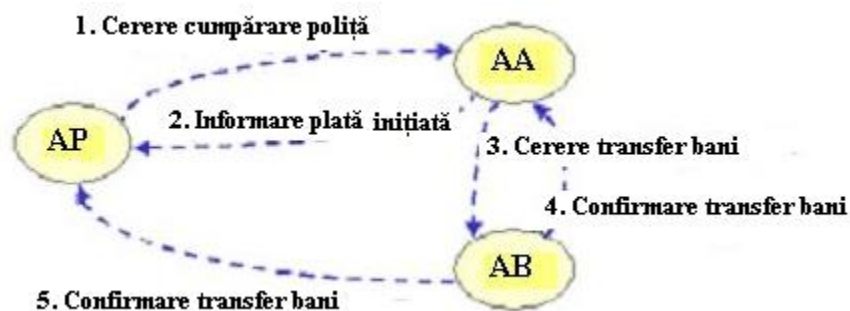


Figura 9.1 Interacțiunile între agenții AP-X, AA-T și AB

9.3. Agenți Inteligenți în Data Mining Distribuit

Data Mining (DM) sau descoperirea de cunoștințe în bazele de date (*Knowledge Discovery in Databases - KDD*) este procesul căutării unor informații semnificative în

volume mari de date, analiza explorării, prin mijloace automate sau semi-automate, unor mari cantități de date în scopul descoperirii unor șabloane, reguli sau tendințe semnificative.

DM este considerat ca un domeniu de cercetare în cel puțin trei domenii: statistică, Inteligență Artificială și baze de date. Câteva dintre cele mai populare tehnici și instrumente folosite în DM sunt:

- 1) **Metode statistice.** Unul dintre principalele scopuri ale DM, și mai general ale inferenței statistice, este, deseori, extragerea informațiilor cauzale din date. În acest sens, regresia liniară, regresia multiplă și regresia neliniară sunt câteva dintre metodele de modelare folosite ca metode statistice în DM.
- 2) **Metode ale Inteligenței Artificiale.** Cele mai importante metode de IA folosite în DM sunt cele ale Inteligenței Computaționale (*Computational Intelligence*) și ale Învățării Automate (*Machine Learning*). Dintre metodele Inteligenței Computaționale folosite în DM amintim: rețele neuronale, sisteme fuzzy, calculul probabilistic, calculul evolutiv. Domeniul Învățării Automate constă în metode care să permită calculatorului să învețe relații din anumite seturi de date. Dintre cele mai populare metode ale Învățării Automate folosite în DM amintim mașinile cu suport vectorial și arborii de decizie.
- 3) **Metode specifice bazelor de date.** Cercetătorii din domeniul bazelor de date au dezvoltat o serie de metode și mecanisme care să fie folosite în sarcini de DM. În special decoperirea regulilor de asociere a fost folosită în domeniul DM și este foarte potrivită pentru seturi mari de date.

Data Mining Distribuit (*Distributed Data Mining - DDM*)

DDM se referă la descoperirea șabloanelor în seturi de date distribuite. Seturile de date sunt stocate în baze de date locale, găzduite de calculatoare locale, care sunt conectate printr-o rețea de calculatoare. DM are loc la nivel local și apoi la nivel global unde rezultatele locale (obținute în urma DM local) sunt combinate pentru a obține rezultatele globale.

Figura 9.2 prezintă o arhitectură de Data Mining Distribuit.

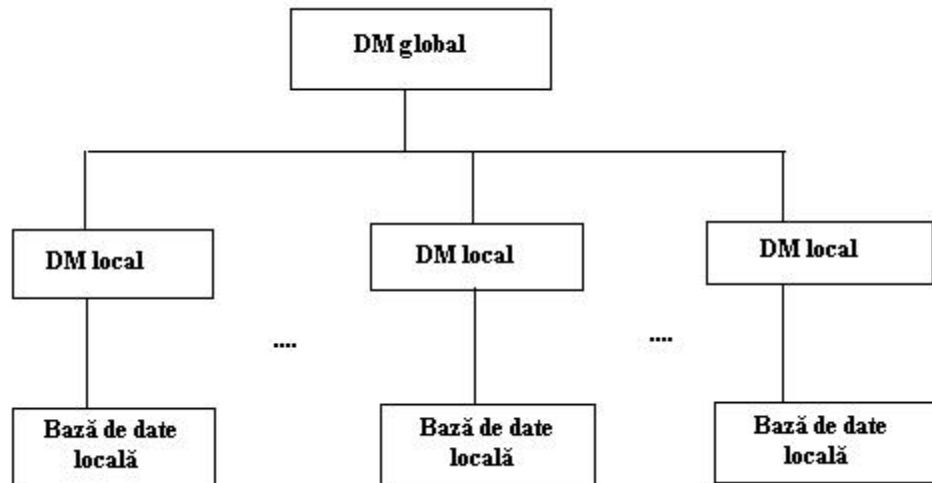


Figura 9.2 Arhitectura unui sistem de DDM

Model bazat pe agenți pentru DDM

Cele două metode de bază dezvoltate pentru arhitecturi de DDM sunt: modelul client-server și modelul bazat pe agenți.

Modelul bazat pe agenți pentru o arhitectură de DDM poate fi împărțit în: sisteme care folosesc agenți mobili și sisteme care folosesc agenți staționari.

Păstrarea securității implică faptul că, în unele situații, utilizatorul poate exploata date sensibile, care nu ar trebui să părăsească site-ul proprietarului. În aceste situații, există opțiunea de a folosi modelul bazat pe agenți mobili, în care algoritmul de *mining* și parametrii relevanți ai acestuia sunt trimiși spre site-ul de date iar la sfârșitul procesului agentul mobil este distrus.

În modelul bazat pe agenți, fiecare bază de date locală (*site* de date) are asociat unul sau mai mulți agenți care procesează datele locale și comunică rezultatele celorlalți agenți sau unui agent central supervisor care controlează comportamentul agenților locali. Agenții pot lucra în paralel și pot și să partajeze informația pe care au obținut-o la un moment dat.

Ca urmare, arhitectura unui sistem de DM folosind agenți inteligenți conține câțiva agenți locali corespunzători bazelor de date locale și un agent central corespunzător bazei de date centrale. Într-o astfel de arhitectură, fiecare agent procesează datele din baza sa de date locală și comunică rezultatul agentului central, iar apoi rezultatele explorării vor fi combinate de agentul central.

Figura 9.3 prezintă arhitectura unui sistem de DM folosind agenți inteligenți.

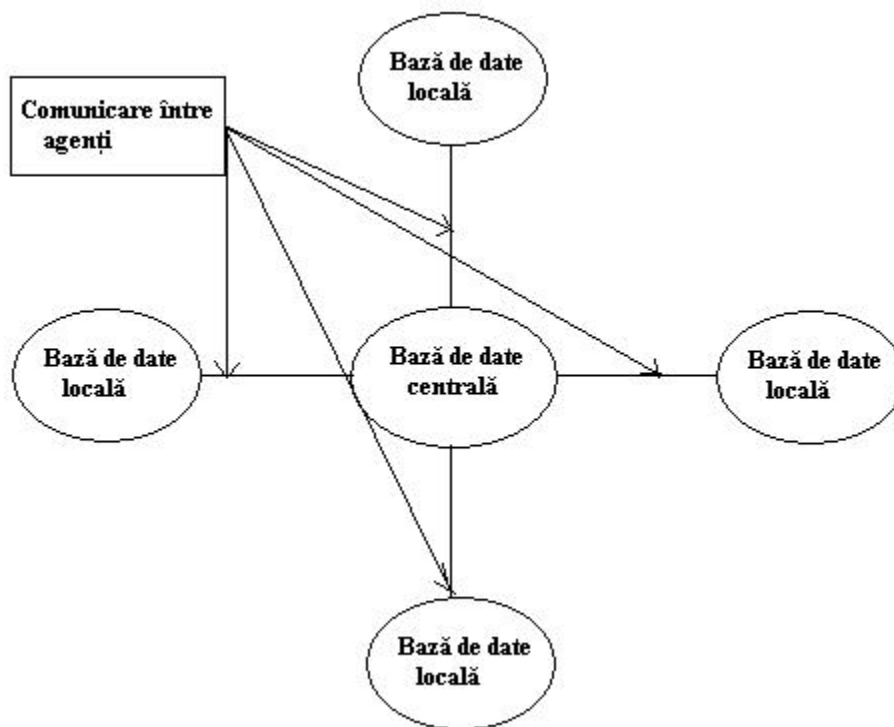


Figura 9.3 Arhitectura unui sistem de DDM folosind agenți inteligenți

Algoritmii de DDM pot fi aplicați în probleme din lumea reală din diferite domenii, cum ar fi: clasificarea de texte, detectarea atacurilor în rețele, detectarea fraudelor de credit, explorarea datelor financiare de pe dispozitive mobile, analiza diagnosticelor medicale, comerț electronic (stabilirea profilului cumpărătorilor).

9.4. Agenți Web de Informații

Datorită naturii dinamice a World Wide Web, a creșterii rapide a cantității de date precum și eterogenității serviciilor, extragerea informațiilor utile din cantități mari de date stocate a devenit o sarcină care nu poate fi efectuată doar de utilizatori.

Conform metaforei “Information Superhighway” a lui Etzioni, un *agent de informație* ar putea fi:

- conducătorul de pe bancheta din spate care face sugestii la orice moment;
- un conducător de taxi care ne conduce la destinație;

- un consilier ale cărui cunoștințe ne scutesc de a aborda personal anumite sarcini.

Agenții inteligenți pentru Internet sunt numiți în general *agenți de informații*.

Tehnologia agenților de informații este un subdomeniu al tehnologiei generale a agenților software inteligenți, fiind o tehnologie interdisciplinară care folosește metode și mecanisme din Inteligența Artificială, baze de date și sisteme bazate pe cunoștințe, sisteme distribuite de informații, regăsirea informației (*information retrieval*) și interacțiune între utilizator și calculator (*human computer interaction*).

Scopul acestei tehnologii este de a dezvolta entități computaționale software autonome (agenți inteligenți) care trebuie să acceseze surse de informații eterogene și distribuite geografic, așa cum sunt în Internet. Acești agenți efectuează *căutări active*, *regăsire*, *analizare* și *manipulare* a informațiilor eterogene, precum și *vizualizarea* acestora.

Agenții de informații caută în mod activ informații pentru utilizator; ei pot scana în baze de date on-line, biblioteci de documente, sau prin directoare în scopul căutării unor documente care pot fi de interes pentru utilizator. Fiind niște entități inteligente, agenții de informații pot ține la curent utilizatorul în legătură cu orice dezvoltare a unui anumit domeniu (spre exemplu dacă într-o bibliotecă de documente a apărut un document care îl interesa pe utilizator).

Agenții de informații sunt folosiți aproape în toate aplicațiile legate de Web. Spre exemplu, un agent numit Softbot este implementat pentru a acționa ca un asistent personal a utilizatorilor Web care caută anumite informații. Agentul va descoperi cea mai bună metodă de căutare și cea mai bună sursă de date pentru a satisface cerințele utilizatorului. Alte exemple de agenți de informații sunt: **BargainFinder** (un agent de cumpărare a compact discurilor), **Maxims** (un agent pentru filtrare e-mail-uri, **MCL** (un agent pentru fixarea de întâlniri), **NewT** (un agent pentru filtrarea știrilor).

O altă categorie de agenți de informații sunt *agenții de informații adaptivi*. Aceștia se pot adapta la schimbările din mediile de informații în care acționează. În această categorie de agenți adaptivi ar putea intra asistenții personali pe Web care au capacitatea de a învăța.

Adaptarea unui agent la schimbările care apar în mediul său poate fi realizată în mod izolat sau în colaborare cu alți agenți, folosind metode de învățare pentru un singur agent, respectiv mai mulți agenți. Cele mai cunoscute tipuri de metode de învățare sunt (așa cum se va vedea în Capitolul VIII): *învățarea supervizată*, *învățarea nesupervizată* și *învățarea prin întărire*.

O altă categorie de agenți de informații sunt *agenții de informații mobili*. Aceștia se deplasează autonom pe Web de la un site la altul pentru a-și efectua sarcina sau pentru a interoga diferite servere de date. Agenții mobili de informații se folosesc îndeosebi în domeniul serviciilor complexe de telecomunicații.

Un agent de informație poate fi implementat ca, apelat de sau incorporat în orice aplicație web client server. Cele mai des utilizate tehnici pentru a implementa astfel de aplicații sunt: applet-uri Java, scripturi, ActiveX-uri independente de platformă, CGI-uri, servlet-uri Java. Pentru acces la baze de date se folosesc interfețe de programare generice, cum ar fi: JDBC (Java Database Connectivity), cu limbajul de interogare incorporat SQLJ și ODBC (Open Database Connectivity).

În ultimii ani au fost propuși multe tipuri de agenți Web, de la agenți specifici anumitor domenii (*domain-dependant agents*): agenți în comerț electronic, agenți pentru asistarea călătoriilor, agenți pentru căutare de informații până la agenți specifici funcției pe care o efectuează (*function-dependant agents*): agenți de negociere, agenți cooperativi, agenți pentru rezolvare de probleme, etc.

Ceea ce este important de menționat este faptul că uneori multe dintre aplicațiile web (Internet) care folosesc agenți folosesc doar metafora **agent**, fără a fi, de fapt, într-adevăr bazate pe agenți.

9.5. Algoritmi cu furnici (Ant Algorithms)

Studiul insectelor, și în particular a *insectelor sociale* a demonstrat că inteligența umană nu este singurul tip de inteligență care merită a fi studiat. În această categorie de insecte sociale intră: albine, viespi, termite, furnici.

Luat separat, fiecare individ este destul de simplu, însă putem vorbi de inteligență dacă ne gândim la un grup de insecte comunicând în scopul atingerii unui obiectiv. Inteligența unui astfel de grup se numește *swarm intelligence*.

O furnică reală poate fi modelată folosind un **agent inteligent**. Sistemele cu furnici, la fel ca și algoritmi genetici, sunt abordări bazate pe populații. Dându-se o populație de furnici (agenți), fiecare furnică găsește o soluție și apoi o comunică prin *stimergie* (*stimergergy*) (un fel de recompensă) celorlalte furnici (agenți).

Ca urmare, sistemele bazate pe furnici pot fi legate de problematica învățării automate și a sistemelor multiagent.

Algoritmii cu furnici au fost aplicați pentru rezolvarea diferitelor probleme, cum ar fi:

- Problema comis voiajorului.
- Rutarea în rețele de comunicație.
- Rutarea vehiculelor.
- Colorarea grafurilor.

9.6. Aplicații practice și industriale ale Inteligenței Artificiale Distribuite

În această secțiune vom prezenta pe scurt, fără a intra în prea multe detalii rolul Inteligenței Artificiale Distribuite în aplicații practice și industriale.

După cum se știe, agenții nu sunt o soluție universală pentru software-ul industrial. Ca și alte tehnologii, agenții sunt folosiți în special în probleme ale căror caracteristici necesită capacitățile particulare ale acestora. Agenții sunt potriviți în aplicații care sunt:

- **Modulare.**

După cum au fost definiți, agenții sunt obiecte pro-active și beneficiază de facilitățile modularității oferite de tehnologia orientată obiect (OO).

- **Descentralizate.**

Un agent este mai mult decât un obiect; este un obiect pro-activ, un proces care nu trebuie să fie invocat din exterior, ci își monitorizează în mod autonom mediul și alege acțiuni care i se par potrivite. Această caracteristică a agenților îi face potriviți pentru aplicații care pot fi descompuse în procese de sine stătătoare,

fiecare fiind capabil să efectueze lucruri utile fără o direcționare continuă din partea altor procese.

- **Schimbătoare.**

Agenții sunt potriviți problemelor modulare deoarece sunt obiecte. Ei sunt foarte potriviți problemelor descentralizate deoarece sunt obiecte pro-active. Aceste două caracteristici combinate îi fac foarte potriviți problemelor care e probabil să se schimbe frecvent. Modularitatea permite modificarea unei unități a sistemului la un moment dat, iar descentralizarea minimizează impactul pe care schimbarea unui modul o are asupra comportamentului celorlalte module.

- **Prost structurate (ill-structured).**

În urma proiectării sistemelor tradiționale rezultă arhitectura aplicației, indicând care entități interacționează cu care și specificând interfețele între acestea.

Să considerăm spre exemplu un sistem electronic care suportă operații comerciale, în care comenzile sunt disponibile oricărui licitant autorizat. Cerința ca proiectantul sistemului să specifice expeditorul și beneficiarul fiecărei tranzacții va conduce rapid la un *“blocaj prin analiză”*. Din punctul de vedere tradițional, această aplicație este prost structurată, ceea ce înseamnă de fapt că nu toate informațiile structurale necesare sunt disponibile în sistem atunci când acesta este proiectat.

Agenții, în schimb, suportă o astfel de aplicație. Distincția fundamentală în opinia agentului despre lumea înconjurătoare este între *“propria sa persoană”* și *“mediu”*. *“Propria sa persoană”* este cunoscută și previzibilă, pe când *“mediul”* se poate schimba nelimitat. Alți agenți sunt parte din acest mediu dinamic, în continuă schimbare. Depinzând de complexitatea agenților individuali, ei pot sau nu să aibă un model explicit al celorlalți. În locul specificării entităților individuale interconectate și interfețele acestora, un sistem bazat pe agenți va trebui să identifice doar clasele de entități din sistem și impactul acestora asupra mediului.

- **Complexe.**

O măsură a complexității unui sistem este numărul de comportamente diferite pe care trebuie să le manifeste. Spre exemplu, un atelier de producție poate produce o anumită piesă în diferite moduri posibile, depinzând de mașinile folosite și de ordinea în care acestea sunt folosite. Numărul comportamentelor posibile în acest exemplu simplu este exponențial în raport cu numărul de mașini diferite din atelier.

Acest exemplu demonstrează complexitatea combinatorială: numărul interacțiunilor diferite între o mulțime de elemente crește mult mai mult decât numărul de elemente din mulțime. Prin maparea unor agenți individuali elementelor care interacționează, arhitecturile agent pot înlocui codificarea explicită a acestei mulțimi mari de interacțiuni prin generarea lor la execuție.

Oricum, arhitecturile agent proiectate corespunzător pot muta generarea spațiului combinatorial de comportamente de la momentul proiectării la momentul execuției, reducând astfel semnificativ cantitatea de software care trebuie generată și astfel și costul de construire al sistemului.

9.6.1 Unde în ciclul de viață al produselor sunt folosiți agenții?

În principiu, agenții pot suporta diferite stagii în ciclul de viață al unui sistem sau al unui produs. Spre exemplu, agenții pot fi folosiți în proiectarea unei mașini, în procesul de vânzare al acesteia sau la asigurarea service-ului când mașina se defectează.

Complexitatea din ce în ce mai mare a produselor moderne favorizează beneficiile sistemelor bazate pe agenți. Agenții pot fi folosiți în următoarele etape din ciclul de viață al produselor:

- **În proiectarea produselor.** Unul dintre sistemele de proiectare în care se folosesc agenți și care demonstrează beneficiile folosirii acestora în proiectarea produselor este sistemul **RAPPID** (Responsible Agents for Product-Process Integrated Design). În acest sistem agenții sunt proiectanți, reprezentând componente și subsisteme ale produsului.
- **În planificare și control.**
- **În controlul în timp real.**

9.6.2 Instrumente de dezvoltare a aplicațiilor bazate pe agenți

Un agent ca un sistem reactiv.

Mediul de construcție a agenților **ABE** (Agent Buiding Environment) este o bibliotecă C++ extinsă pentru construirea agenților bazați pe reguli cu raționament înainte și a fost proiectat pentru aplicații care monitorizează rețele de informații și informează utilizatorul atunci când anumite condiții sunt îndeplinite. ABE își reprezintă faptele și regulile folosind sintaxa KIF.

Agenții care se deplasează.

Anumiți agenți migrează de la un procesor la altul. Primul și unul dintre cele mai populare mecanisme pentru construirea agenților mobili este limbajul **TELESCRIPT**, care modelează lumea sub formă de locuri (procesoare) și agenți (proces). Pe de altă parte, *agletii* creați de IBM sunt obiecte Java care se mută de pe o mașină gazdă pe alta. API-ul Java Aglet (*J-APPI*) definește metodele necesare pentru a crea aglet, gestionarea mesajelor și a ciclului lor de viață.

Agenții ca membri ai unei comunități.

Mediul **ADE** (Agent Development Environment) construiește o platformă robustă în timp real pentru folosirea în domeniul industrial al tehnicilor IA. ADE furnizează o ierarhie predefinită de clase pentru agenți și părți de agenți, pentru comunicare între agenți precum, un limbaj de programare al comportamentului agenților bazat pe Rețele Petri precum și un mecanism pentru simularea de aplicații distribuite bazate pe agenți.

Un alt mediu pentru dezvoltare de sisteme multiagent în concordanță cu specificațiile FIPA este mediul **JADE** (Java Agent DEvelopment Framework). JADE permite crearea unor agenți, asignarea unor sarcini agenților, precum și comunicarea între

aceștia. JADE este scris în întregime în JAVA, permițând programatorilor să lucreze în întregime în JAVA când își dezvoltă agenții.