

CAPITOLUL I

AGENȚI ȘI AGENȚI INTELIGENȚI

1.1 Inteligența Artificială

Domeniul **Inteligenței Artificiale**, una dintre cele mai noi discipline apărute în domeniul informaticii, urmărește să înțeleagă entitățile inteligente. Dar, spre deosebire de filozofie și psihologie, care deasemenea se ocupă de studiul inteligenței, Inteligența Artificială are ca scop *construirea* de entități inteligente, precum și înțelegerea lor.

Inteligența Artificială (IA) încercă să înțeleagă entitățile inteligente. Așadar, un motiv pentru care se studiază este învățarea despre noi înșine. Spre deosebire de filozofie și psihologie care sunt confruntate și ele cu inteligența, IA se ocupă cu construirea *entităților inteligente* și înțelegerea acestora. Entitățile inteligente sunt interesante și folositoare în multe domenii.

Inteligența Artificială poate fi considerată ca fiind acel domeniu al informaticii ce are ca obiectiv proiectarea sistemelor înzestrate cu anumite proprietăți pe care în mod obișnuit le asociem inteligenței umane. Deci, unul dintre scopurile fundamentale ale cercetărilor în domeniul Inteligenței Artificiale este reproducerea de către calculator a raționamentelor umane.

Așa cum nimeni nu poate să prevadă viitorul în amănunt, așa și calculatoarele înzestrate cu un grad de inteligență umană (sau mai mult decât atât) vor avea un mare impact în viața cotidiană și în viitorul curs al civilizației.

Inteligența Artificială a cunoscut o dezvoltare rapidă, ajungând în ziua de azi să cuprindă o varietate de subdomenii, printre care se numără atât subdomenii generale cum ar fi percepția și gândirea logică, învățarea, înțelegerea limbajului, cât și subdomenii particulare precum demonstrarea teoremelor, scrierea poeziilor și diagnosticarea bolilor. Aplicații din acest domeniu se regăsesc frecvent în viața cotidiană, în științele tehnice și economice, în științele naturii, și în științele umane.

1.1.1 Istoric al Inteligenței Artificiale

Figura 1.1 prezintă evenimentele majore care avut loc în domeniul Inteligenței Artificiale.

În anul 1941 o invenție a revoluționat fiecare aspect legat de stocarea și procesarea informației. Această invenție, dezvoltată atât în SUA cât și în Germania a fost calculatorul electronic.

Inovația din anul 1949, primul calculator care stoca programele, a făcut mult mai ușoară munca de introducere a unui program, iar progresele în teoria calculatoarelor conduc spre Știința Calculatoarelor, eventual spre Inteligența Artificială. Odată cu inventarea metodelor electronice de procesare a datelor, a apărut un mediu care face posibilă apariția IA.

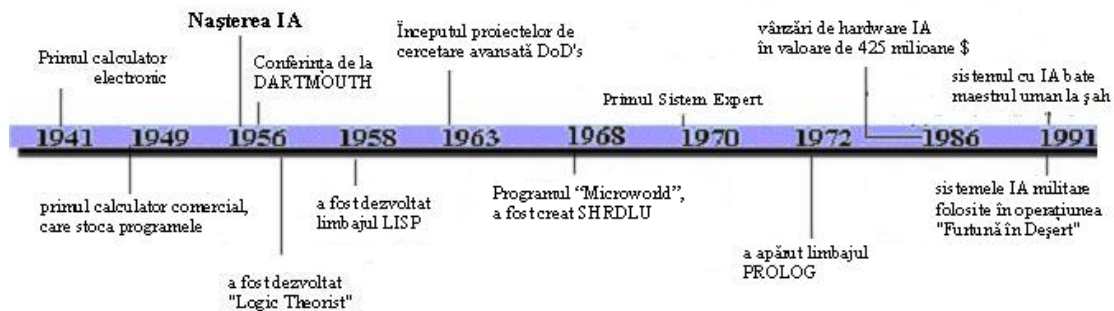


Figura 1.1 Evenimente majore în Inteligența Artificială

Momentul de naștere al Inteligenței Artificiale este, însă, legat de anul 1956, când la reuniunea de la Dartmouth College a fost prezentat programul de demonstrare a logicii propozițiilor "Logic Theorist" de către Newell, Shaw și Simon, iar termenul de inteligență artificială a fost "inventat" cam în același an de către John McCarthy. Programul "Logic Theorist" este considerat de mulți ca fiind primul program de IA.

După conferința de la Dartmouth, au fost format centre de cercetare în Inteligența Artificială la Carnegie Mellon și MIT, și astfel au apărut noi provocări în domeniu: în primul rând cercetări legate de crearea de sisteme care pot eficient rezolva probleme, limitând căutarea, așa cum a fost Logic Theorist, iar în al doilea rând crearea de sisteme care pot învăța din experiență.

În 1957, a fost testată prima versiune a unui nou program numit *General Problem Solver* (GPS). În timp ce erau create multe programe în domeniul IA, John McCarthy dezvoltă una din cele mai mari realizări în istoria Inteligenței Artificiale. În anul 1958, McCarthy anunța limbajul LISP, care foarte curând a fost adoptat ca limbaj de majoritatea dezvoltatorilor de IA.

În anul 1963 MIT a primit 2,2 milioane dolari din partea guvernului Statelor Unite ale Americii pentru a fi folosite în cercetări legate de Inteligență Artificială.

În anul 1968 este creat programul SHRDLU. SHRDLU a fost parte a unui proiect "Microworld", care consta în cercetare și programare în medii (lumi) mici (cum ar fi cele cu număr limitat de forme geometrice). Următorul progres a fost în anul 1970, când au apărut sistemele expert, sisteme capabile să efectueze expertize în anumite domenii (să facă previziuni economice, să ajute doctorii la diagnosticare, etc). În anul 1970 apare un alt rezultat al dezvoltării în domeniul Inteligenței Artificiale, și anume limbajul PROLOG.

În anul 1986 vânzările de hardware și software IA în Statele Unite ale Americii au ajuns la 425 milioane de dolari. Sistemele expert erau deosebit de căutate datorită eficienței lor în diverse domenii. Companii mari precum DuPont, General Motors și Boeing se bazau din ce în ce mai mult pe sisteme expert.

Perioada 1980-1990 nu a fost foarte prolifică pentru industria IA. În anii 1986-1987 cererea de sisteme cu IA era în scădere. În ciuda unor evenimente descurajatoare și eșecului unor proiecte, Inteligența Artificială a început ușor să-și revină. Japonia a dezvoltat o nouă tehnologie, și anume logica fuzzy (abilitatea de a lua decizii în condiții incerte). Rețelele neuronale au fost, deasemenea, reconsiderate, ca fiind posibile căi de a realiza IA.

Domeniul militar a fost cel care a testat calculatoare cu IA în timpul operațiunii "Furtună în Deșert".

În prezent, cererea mare de tehnologie bazată pe Inteligența Artificială conduce inevitabil la noi progrese în domeniu. Cert este că Inteligența Artificială afectează și va continua să afecteze viețile noastre.

1.1.2 Problematika Inteligenței Artificiale

Una dintre întrebările la care IA ar trebui să răspundă este: cum este posibil ca un creier mic și încet, biologic sau electronic, să perceapă, să înțeleagă, să prezică și să manipuleze o lume mult mai mare și mai complicată decât el însuși. Toți cercetătorii ar trebui să se privească în oglindă pentru a vedea un exemplu de *sistem inteligent*.

Studiul inteligenței este una dintre cele mai vechi preocupări ale omenirii. De peste 2000 de ani, filosofii încearcă să înțeleagă cum învățarea, memorarea și raționarea pot sau trebuie făcute. Apariția calculatoarelor la începutul anilor '50 a revoluționat speculația cu privire la capacitățile mentale într-un experiment real și o disciplină teoretică.

IA are o mare varietate de subdomenii, de la arii generale ca și percepția și raționalitatea logică, până la teme specifice ca și jocul de șah, demonstrarea teoremelor matematice, scrierea de poezii și diagnosticarea bolilor. Adesea, cercetători din alte domenii folosesc IA, unde găsesc instrumentele și vocabularul pentru a sistematiza și automatiza temele intelectuale la care au lucrat toată viața lor. Similar, cercetătorii IA pot alege să aplice metodele lor fiecărei arii din intelectul uman. În acest sens putem spune că IA este într-adevăr un domeniu universal.

Definițiile IA

De-a lungul anilor s-au enunțat diferite definiții ale IA. Aceste definiții variază de-a lungul a două dimensiuni principale care se adresează: *proceselor de gândire și raționament și comportamentului*.

Până în prezent s-au făcut multe încercări de definire a IA. Aceste definiții pot fi grupate în 4 categorii:

1. Sisteme care gândesc ca oamenii. În acest sens o definiție e dată de Haugeland în 1985: "IA este studiul modului în care calculatoarele pot fi făcute să gândească – mașini ce gândesc-".
2. Sisteme care gândesc rațional. O definiție sugestivă e dată de Winston în 1992: "IA este studiul modalităților de a face capabil un calculator să perceapă, să raționeze și să acționeze inteligent".
3. Sisteme care acționează ca oamenii. Kurzweil, în 1990 definea IA astfel: "Arta de a crea calculatoare capabile să efectueze acțiuni ce necesită inteligență atunci când sunt executate de oameni". Rich și Knight, în 1991 spuneau că: "IA este studiul modului în care putem determina calculatoarele să facă lucruri la care, în prezent, omul este mai bun".
4. Sisteme care acționează rațional. Definiția sugestivă în acest sens e dată de Luger și Stubblefield în 1993: "IA este ramură a științei calculatoarelor ce se ocupă cu automatizarea comportamentului inteligent".

Aceste definiții deschid două direcții: 1 și 2 se referă la *procesele cognitive* (gândire și raționament), în timp ce 3 și 4 se referă la *comportament*.

Deasemenea, 1 și 3 definesc succesul în termeni de performanță *umană*, pe când 2 și 4 fac referire la conceptul *ideal* de inteligență, numit *raționalitate*. În acest ultim sens, un sistem va fi numit rațional dacă execută ceea ce trebuie.

Cele două direcții generează 2 întrebări de bază. În realizarea sistemului de IA ne interesează gândirea sau comportamentul? Vrem ca sistemul să modeleze activitatea umană, sau să-și creeze propriul mod de acțiune conform unui standard

ideal? În orice caz, ipoteza de bază este că inteligența se referă în principal la acțiuni *raționale*. În cazul ideal, un sistem (agent) inteligent va alege cea mai bună acțiune posibilă într-o anumită situație.

Să studiem, deci, mai îndeaproape, care sunt caracteristicile pe care trebuie să le aibă o mașină pentru a se putea spune despre aceasta că este dotată cu „*inteligență artificială*”.

Inteligența umană

După cum se observă, din definițiile date, una dintre direcțiile propuse este cea în care se spune că o mașină este dotată cu inteligență dacă manifestă caracteristici umane în ceea ce privește comportamentul sau gândirea. Rămâne de văzut cum poate fi realizată oricare dintre aceste manifestări și ceea ce este mai important „*Cum știm dacă am reușit sau nu?*”. Pentru a putea răspunde la această întrebare au fost elaborate o serie de teste la care sunt supuse mașinile considerate inteligente și o serie de metode prin care mașinile sunt verificate acestea pentru a demonstra că posedă Inteligență Artificială, noțiune definită prin enunțurile date anterior.

Comportament uman

În 1950, *Allan Turing* a creat *testul Turing* pentru a testa dacă o mașină se comportă inteligent. Pentru desfășurarea testului este nevoie de doi oameni și de mașina care este evaluată. Un om este interogatorul, conectat la un terminal într-o cameră separată. Al doilea om este și el conectat la un alt terminal. Ambele terminle comunică cu mașina care urmează să fie testată. Interogatorul pune întrebări și primește răspunsuri de la oricare dintre cei doi subiecți (om sau mașină) prin terminal. El poate pune întrebări atât omului cât și mașinii, dar nu îi cunoaște decât ca fiind subiectul A și subiectul B.

Testul este conceput ca să determine care dintre cei doi (A și B) este omul și care mașina. Scopul mașinii este să-l păcălească pe interogator în așa fel încât acesta să creadă că ea este omul, sau cel puțin să nu poată face diferența dintre om și ea. Mașina are voie să facă orice pentru a reuși. Dacă reușește putem trage concluzia că mașina testată poate gândi.

Cea mai serioasă problemă, de care se lovesc cei care concep și dezvoltă astfel de mașini considerate inteligente, s-a dovedit a fi cantitatea mare de cunoștințe (informație) pe care mașina ar trebui să o dețină. Deci, pentru ca o mașină să poată rezolva probleme cu un grad rezonabil de complexitate are nevoie de o cantitate de informație care nu poate fi neglijată de proiectanții săi.

Revenind la *testul Turing*, acesta presupune că mașina care îl va trece deține un imens bagaj de cunoștințe, dacă avem în vedere că aceasta este în competiție cu un om. Astfel ajungem la concluzia că pentru a trece acest test o mașină necesită următoarele calități:

- cunoașterea reprezentațiilor – pentru a putea stoca informații înaintea și în timpul interogatoriului;
- gândire în timp real – pentru a răspunde întrebărilor și a trage noi concluzii, în timp util, folosind informațiile stocate
- învățare – pentru a se putea adapta noilor circumstanțe și pentru a putea detecta și extrapola tipare.

Cu toate aceste calități mașina (computerul) tot nu ar putea să treacă *testul Turing*, decât dacă îngustăm problema și dorim cunoștințe limitate la un domeniu

specific, abia atunci realizarea unei mașini care să treacă testul devine posibilă. Deci, *testul Turing* așa cum este el acum s-ar putea să nu fie trecut încă mult timp.

În *testul Turing* nu se pune problema interacțiunii fizice dintre interogator și cel interogată deoarece, simularea fizică a unei persoane nu este necesară pentru a simula inteligența acesteia. Dacă luăm în considerare și existența unei interacțiuni fizice între interogator și mașină, atunci testul poartă denumirea de „*Testul Turing Total*”.

Dacă avem în vedere definiția conform căreia o mașină posedă Inteligență Artificială dacă se comportă ca și un om (definițiile 1 și 3), atunci trebuie să avem în vedere faptul că un om are un bagaj foarte mare de cunoștințe, deci vom avea de rezolvat următoarele probleme:

- un sistem de Inteligență Artificială trebuie să conțină multe cunoștințe dacă vrem să rezolve și altceva decât probleme triviale;
- pe măsură ce cantitatea de cunoștințe crește, devine din ce în ce mai greu de accesat lucrurile dorite, deci vor fi adăugate cunoștințe în plus, dar acum sunt deja mai multe cunoștințe, deci vom adăuga din nou altele, etc.

Gândire umană

Metoda prin care am putea obține o mașină inteligentă ar fi să facem o mașină să gândească la fel ca și un om. Pentru a putea face un asemenea lucru ar trebui, în primul rând, să știm cum gândește un om. Cei mai în măsură să ne ajute în rezolvarea acestei dileme sunt psihologii. Pentru a desluși misterele funcționării minții umane psihologii ne spun că există două metode: autoanaliza și experimentele psihologice. Aplicând aceste metode putem deduce teorii în legătură cu gândirea oamenilor. Teoriile deduse vor fi puse în practică cu ajutorul programelor implementate, iar dacă datele de intrare, cele de ieșire precum și sincronizarea comportamentului se potrivește cu cel uman, atunci avem dovada că mecanismul programului seamănă cu cel uman.

Există mai multe motive care fac o gândire asemănătoare cu cea a oamenilor de dorit la mașinile presupuse a fi inteligente. În primul rând este foarte practic, având în vedere că vor rezolva probleme reale, cu care se confruntă oamenii. Este util ca omul să înțeleagă modul în care mașina a ajuns la rezultatul pe care îl propune ca fiind cel corect, deoarece de multe ori omul trebuie să știe el să rezolve problemele nu să fie rezolvate de un robot. De asemenea, se presupune că aceste mașini vor acționa în locul oamenilor, deci este foarte posibil să interacționeze și cu oameni, iar pentru a colabora cu succes mașina ar trebui să gândească la fel ca și un om. Sunt, în mod evident, câteva caracteristici ale gândirii umane pe care nu ar fi bine să le regăsim în gândirea mașinii inteligente, și anume: viteza redusă de efectuare a calculelor algebrice și predispoziția omului spre a face greșeli.

Inteligența Artificială studiază cunoașterea și gândirea și prin acest lucru se relaționează științe precum psihologia, lingvistica, antropologia, filozofia. Însă cercetătorii din domeniul Inteligenței Artificiale, spre deosebire de cei din alte domenii precum filozofia, psihologia, sociologia nu încercă doar să înțeleagă entitățile inteligente, ci, în același timp, încearcă să construiască entități inteligente, încearcă să „*fabriceze inteligență*”.

Omul este considerat a fi ființa cea mai inteligentă, astfel se poate concluziona că a modela gândirea umană înseamnă a crea inteligență.

Inteligența rațională

Omul este, în principiu, o ființă rațională, dar, din păcate, este supus greșelii. De aceea, dacă dorim să construim o mașină care să gândească sau să se comporte ca și un om, ar trebui să înlăturăm această problemă, prentu că nu am vrea ca mașina construită să greșească.

Există axiome și reguli în logică, astfel încât pornind de la ipoteze adevărate, pe baza acestora, cu ajutorul axiomelor și regulilor să ajungem doar la concluzii corecte. Pentru ca, pornind de la premise corecte, mașina să ia doar decizii corecte ea trebuie să fie dotată cu inteligență bazată pe logică, iar o astfel de inteligență este inteligența rațională.

Comportament rațional

O direcție a cercetărilor din Inteligența Artificială este cea îndreptată spre demonstrarea teoremelor, jocuri, sau rezolvarea de probleme care necesită expertiza umană într-un anumit domeniu.

Dacă execută acțiuni foarte importante, adică sunt într-o poziție de mare răspundere (ex. pilot automat al unui avion), mașinile inteligente trebuie să aibă un comportament rațional. Comportament rațional înseamnă să te porți în așa fel, încât, pe baza unor cunoștințe (păreri), să îți atingi scopurile.

S-a reușit implementarea de programe care să demonstreze teoreme matematice complicate și care să descopere chiar concepte matematice noi. De asemenea putem observa o creștere a complexității jocurilor pe calculator cu fiecare joc nou apărut.

Un comportament rațional este caracterizat de acțiuni care își ating scopul prestabilit, acestea fiind consecința unei decizii luate ca și concluzie a unei gândiri logice. Un astfel de comportament este necesar pentru a avea siguranța că decizia luată este cea corectă. Acesta este preferat pentru mașini ce îndeplinesc sarcini statice, situație în care avem mai mare nevoie de siguranță în detrimentul flexibilității.

Gândire rațională

Filozoful grec *Aristotel* este primul care a încercat să codifice *gândirea corectă* ca fiind un incontestabil proces al rațiunii. Vestitele lui silogisme formează tipare în așa fel încât, pentru premise corecte să rezulte concluzii corecte. Exemplu: „*Socrates este om. Toți oamenii sunt muritori. Din cele două rezultă că și Socrates este muritor*”. Aceste legi ale gândirii se presupune că guvernează operațiile minții și inițiază logica.

Allan Newell și Herbert Simon, două nume semnificative în domeniul Inteligenței Artificiale, au dat, în anul 1976, definiția unui sistem fizic de simboluri. „*Un sistem fizic de simboluri este format dintr-o mulțime de entități numite simboluri, care sunt șabloane fizice ce pot apărea drept componente ale altor tipuri de entități numite structuri (sau structuri simbolice). Astfel, o structură simbolică este compusă dintr-un număr de instanțe (particularizări) ale simbolurilor legate într-un mod fizic. În orice moment de timp, sistemul va conține o colecție de astfel de structuri simbolice. Pe lângă aceste structuri, sistemul conține de asemenea o colecție de procese care operează asupra expresiilor pentru a produce alte expresii: procese de creare, modificare, reproducere și distrugere. Un sistem fizic de simboluri este o mașină care produce în timp o colecție de structuri simbolice care evoluează. Un*

astfel de sistem există într-o lume de obiecte mai cuprinzătoare decât cea a expresiilor simbolice”.

Pornind de la această definiție Newell și Simon au făcut următoarea ipoteză, numită ipoteza sistemului fizic de simboluri: „*Un sistem fizic de simboluri are capacitățile necesare și suficiente pentru a produce acțiuni general inteligente”.*

Deși nu a fost demonstrată, această teorie pare foarte plauzibilă, deoarece standardul raționalității este bine definit și complet general.

O mare problemă este faptul că, prin intermediul notațiilor logice, trebuie să transformăm cunoștințele preluate din lumea reală în informație standardizată, care să poată fi apoi stocată și prelucrată. Acest lucru este o mare problemă, deoarece informații de bun simț, precum faptul că un obiect nu poate fi în două locuri de-o dată sau faptul că nu trebuie să IAD drumul unui pahar de sticlă pe care îl ai în mână pentru că se va sparge, nu pot fi ușor modelate într-un program.

În concluzie, putem spune că problemele de Inteligență Artificială sunt diverse și complexe, necesitând un volum mare de cunoștințe pentru a rezolva probleme banale pentru oameni.

O altă direcție în eforturile de a crea inteligență este aceea de a studia ipoteze privind natura inteligenței și felul în care ea se manifestă în mod natural. Astfel, înțelegând principiile care stau la baza comportamentului și gândirii inteligente vom putea face ca acestea să se manifeste și la o mașină. O analogie reușită este cea dintre mașini care zboară și păsări (sau insecte), care poate fi aplicată și în cazul mașinilor care sunt inteligente, analogia fiind făcută cu oamenii. Precum avioanele au preluat de la păsări doar principiile aerodinamicii, la fel și mașinile vor manifesta doar inteligență, fără a prelua nimic altceva din tot ceea ce ține de oameni.

Scopul Inteligenței Artificiale este de a înțelege principiile care fac posibilă manifestarea inteligenței în medii atât naturale cât și artificiale.

1.1.3 Direcții de cercetare în Inteligența Artificială

Se disting două direcții fundamentale de cercetare în Inteligența Artificială. Pe de o parte se urmărește realizarea de mașini capabile să reproducă funcțiile cognitive ale minții umane, așa numita *modelare cognitivă*. În acest sens, *psihologia cognitivă* este una din științele cognitive căreia Inteligența Artificială îi preia metodele, în domenii cum ar fi prelucrarea limbajului natural, învățarea, percepția. Astfel de mașini inteligente au fost numite în literatura de specialitate “creiere artificiale”, ideea de bază fiind construirea de mașini inteligente pornind de la modele ale activității neuronale. Astfel de modele sunt rețelele neuronale artificiale. Aceste direcții de studiu în Inteligența Artificială definesc domeniul *Inteligenței computaționale* (Calcul inteligent), caracteristica sa principală fiind reprezentarea numerică a cunoașterii.

Pe de altă parte, se urmărește realizarea de programe care să permită prelucrarea simbolică a informațiilor (cunoștințelor) folosind metodele logicii, așa numita *abordare logico-simbolică* (gândirea rațională). Tradiția logico-simbolică a Inteligenței Artificiale urmărește realizarea sistemelor inteligente utilizând mecanisme de deducție pentru obținerea unei soluții a problemei, bazate pe descrierea problemei în formalismul logic. O direcție de cercetare în acest sens urmărește dezvoltarea unor *sisteme expert*, programe de Inteligență Artificială bazate pe cunoaștere, ce rezolvă probleme care în mod normal necesită experiența unui specialist. Sistemele expert manipulează descrieri simbolice ale cunoștințelor și raționează în general euristic asupra acestor descrieri simbolice. Modul euristic de raționament în astfel de sisteme se impune datorită faptului că se utilizează și cunoștințe imprecise, incomplete sau

uneori ambigue. Acest domeniu al Inteligenței Artificiale tradiționale e bazat pe reprezentarea simbolică a cunoașterii și inferențe logice între cunoștințe.

Cele două direcții de cercetare, deși par a fi complet separate, interacționează totuși, printr-o serie de noțiuni, metode și rezultate. Deși nu este ceva obișnuit, ar fi o cale de cercetare interesantă o abordare de sinteză între cele două direcții, de exemplu un sistem care combină o rețea neuronală și un sistem expert, cu alte cuvinte un sistem inteligent care învață și raționează asemenea oamenilor.

1.2 Conceptul de agent și agent inteligent

Calculatoarele nu știu ce trebuie să facă: fiecare acțiune efectuată de un calculator trebuie să fie anticipată, plănuită și codată explicit de un programator. Dacă un program întâlnește vreodată o situație pe care programatorul nu a anticipat-o, atunci, de obicei, rezultatul nu este prea frumos – în cel mai bun caz, o blocare a calculatorului, iar în cel mai rău caz, pierderi de vieți.

În cele mai multe cazuri, ne bucurăm să acceptăm calculatoarele ca pe niște servitori ascultători, preciși și fără imaginație. Pentru multe aplicații (ca de exemplu calculul salariilor), este pe deplin acceptabil. Totuși, pentru un număr tot mai mare de aplicații, avem nevoie de sisteme care să poată *decide singure* ceea ce trebuie să facă, pentru a satisface obiectivele pentru care au fost scrise. Asemenea sisteme sunt cunoscute sub numele de *agenți*. Agenții care trebuie să opereze într-un mod robust, în medii care se schimbă rapid, sunt imprevizibile sau deschise, unde există o mare posibilitate ca acțiunile să *eșueze*, sunt *agenți inteligenți*, sau *agenți autonomi*. Iată câteva exemple de domenii de aplicații recente pentru agenți inteligenți:

- Atunci când o sondă spațială își face zborul lung de la pământ la alte planete, o echipă de la sol este în mod normal obligată să monitorizeze în mod continuu progresele sondei și să decidă cum să se procedeze în situații neașteptate. Acest lucru este costisitor și, dacă este nevoie să se ia decizii *rapid*, este pur și simplu impracticabil. Din aceste motive, organizații precum NASA investighează serios posibilitatea realizării unor sonde mai autonome, dându-le mai multe abilități și responsabilități de decizie.
- Căutarea pe internet a răspunsului la o anumită întrebare poate fi un proces lung și obositor. De ce nu s-ar da voie unui program – unui agent – să facă căutările pentru noi? Agentul ar primi o întrebare care ar necesita sintetizarea părților de informație din mai multe surse de informație de pe internet. Eșecul ar apărea atunci când o anumită resursă nu ar fi disponibilă (poate din cauza unei căderi de rețea), sau atunci când din acea resursă nu s-au putut obține informații.

1.2.1 Ce sunt agenții?

O modalitate evidentă de a deschide acest capitol ar fi prezentarea definiției termenului *agent*. În mod surprinzător, acest lucru nu se întâmplă: nu există o definiție universal acceptată a termenului agent, existând chiar și în prezent dezbateri și controverse asupra acestui subiect. În esență, există un consens general care se referă la faptul că *autonomia* este noțiunea centrală a agentului, dar mai departe există foarte puține concordanțe. O parte a dificultății este faptul că agenților le sunt asociate atribute care au o importanță diferită pentru fiecare domeniu. Astfel, pentru unele

aplicații, capacitatea agenților de a *învăța* din experiențele lor este de o importanță extremă, în timp ce pentru alte aplicații, învățarea este nu numai neimportantă, ci și nedorită.

Cu toate acestea, o oarecare definiție este totuși necesară – altfel, cuvântul și-ar pierde înțelesul. Deci, un **agent** este o *entitate computațională, ca de exemplu un program software sau un robot care își poate percepe mediul său înconjurător prin senzorii săi și poate acționa asupra acestuia prin acțiuni. Un agent e capabil de acțiuni autonome în mediul său, în scopul satisfacerii obiectivelor pentru care a fost proiectat*. Din acest punct de vedere, autonomia este privită ca abilitatea agenților de a acționa fără intervenția oamenilor sau a altor sisteme: ei au control atât asupra stării lor interne cât și asupra comportamentului lor.

Există câteva lucruri de menționat despre această definiție. În primul rând, definiția se referă la „agenți”, și nu la „agenți inteligenți”. În al doilea rând, definiția nu spune nimic despre ce tip de *mediu* este ocupat de agent. Aceasta se face în mod intenționat: agenții pot ocupa diverse tipuri de mediu, așa cum se va vedea în continuare.

Un agent uman are ochi, urechi și alte organe pentru simțuri (senzori) și picioare, mâini, gură și alte părți ale corpului ca efectori. Un agent robotic poate substitui camera de luat vederi și lentilele infraroșii ca senzori și motoarele ca efectori. Un agent generic este prezentat în Figura 1.2. Scopul programatorului este de a proiecta agenți care se comportă bine în mediul lor.

Interacțiunea între un agent și mediul în care acesta se situează este o interacțiune continuă.

Una din sarcinile unui *agent* este de a asista utilizatorul realizând sarcini în locul acestuia, sau învățându-l pe utilizator ce trebuie să facă. Cu alte cuvinte, agentul trebuie să satisfacă cu succes sarcinile pe care a fost delegat să le facă.

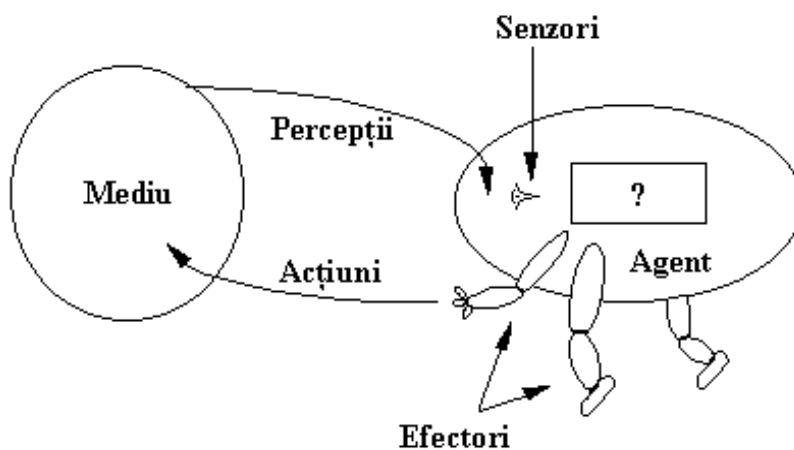


Figura 1.2 Agent interacționând cu mediul prin intermediul senzorilor și efectorilor

Figura 1.2 prezintă imaginea abstractă a unui agent. În această diagramă, se poate vedea ieșirea generată de agent pentru a afecta mediul său. În cele mai multe domenii de o complexitate rezonabilă, un agent nu va avea control *absolut* asupra mediului său. El va avea în cel mai bun caz un control *parțial*, în sensul că îl poate *influența*. Din punctul de vedere al agentului, acest lucru înseamnă că o aceeași acțiune efectuată de două ori în aparent aceleași circumstanțe, poate avea efect total

diferit, și în particular, poate să *nu* aibă efectul dorit. De aceea, aflat în medii netriviiale, agentul trebuie să fie pregătit să dea greș. Putem rezuma această situație spunând ca mediile sunt nedeterminate.

Mai jos sunt două definiții ale agenților date de personalități în domeniu:

Russel și Norvig:

*“Prin **agent** înțelegem orice lucru care poate **percepe** mediul înconjurător prin intermediul **senzorilor** și poate **acționa** asupra acestuia cu ajutorul **efectorilor**.”*

Smith, Cypher și Spohrer:

*“Dați-ne voie să definim un **agent** ca fiind o entitate software persistentă dedicată realizării unui scop special. **Persistența** distinge agenții de subrutine; agenții au propriile lor idei despre modul de finalizare a sarcinii și propriile planificări. **Scopul special** îi distinge de aplicațiile multifuncționale; agenții sunt de felul lor mult mai mici.”*

Un agent este caracterizat prin:

- partea de *arhitectură*, așa numitul *comportament* al agentului – acțiunea efectuată în urma unei secvențe perceptuale;
- partea de *program*, așa numita parte internă a agentului.

Programul agentului este o funcție ce realizează corespondența dintre percepțiile pe care agentul le primește din mediu și acțiunile sale. Acest program trebuie să fie compatibil cu arhitectura agentului. **Arhitectura** realizează interfața între percepția dată de senzori și program, rulează programul și asigură efectuarea acțiunilor alese pe măsură ce acestea sunt generate. Arhitectura poate fi realizată folosind diferite tehnici cum ar fi scenariile, cadrele.

Sarcina Inteligenței Artificiale este legată de realizarea părții de program a agentului: o funcție care implementează modul de trecere de la percepție la acțiune.

Toate programele agent vor avea același schelet - primesc percepții de la un mediu și generează acțiuni. Fiecare program agent va folosi structuri de date interne care vor fi reactualizate pe măsură ce apar noi percepții. Cu aceste structuri de date vor lucra procedurile de luare a deciziilor pentru a genera acțiuni posibile.

Problema cheie a unui agent este aceea de a decide *care* dintre acțiuni ar trebui aplicată pentru a îndeplini obiectivele proiectate. *Arhitecturile agent* sunt de fapt arhitecturi software pentru sisteme care iau decizii și care sunt integrate într-un mediu.

Complexitatea procesului de luare de decizii poate fi afectată de un număr de proprietăți ale mediului. Mediul în care este situat un anumit agent (fie că este sau nu inteligent) poate fi clasificat în funcție de anumite proprietăți ale sale:

- *accesibil / inaccesibil*: un mediu accesibil este un mediu în care agentul poate obține informații complete, corecte și actualizate despre stările mediului (spre exemplu lumea reală este un mediu inaccesibil);
- *determinist / nedeterminist*: un mediu determinist este acel mediu în care o anumită acțiune a agentului are un singur efect garantat (nu există nici o incertitudine despre starea care va rezulta în urma efectuării unei anumite acțiuni);
- *episodic / nonepisodic*: într-un mediu episodic, performanța agentului depinde de un număr discret de episoade (agentul va decide ce acțiuni să efectueze doar pe

baza episodului curent, fără a fi necesar raționament despre interacțiunile dintre episodul curent și alte episoade);

- *static / dinamic*: un mediu static este un mediu care se presupune că rămâne neschimbat exceptând efectuarea acțiunilor agentului. Pe de altă parte, un mediu dinamic este un mediu în care există și alte procese operând asupra sa (spre exemplu lumea reală este un mediu dinamic);
- *discret / continuu*: un mediu este discret dacă există un număr fix, finit de acțiuni și percepții în mediu. Spre exemplu, un agent care joacă șah percepe un mediu discret, pe când un agent care conduce un taxi percepe un mediu continuu.

Două exemple de agenți ar putea fi următoarele:

- Orice sistem de *control* poate fi văzut ca un agent. Un exemplu simplu (și foarte folosit) este termostatul. Termostatul are un senzor ce detectează temperatura camerei. Acest senzor este direct încorporat în mediu (de pildă, în cameră), și produce ca ieșire unul din două semnale: unul care indică faptul că temperatura este prea scăzută, iar altul care indică faptul că temperatura este bună. Acțiunile disponibile unui termostat sunt: “încălzire ON” și “încălzire OFF”. Acțiunea “încălzire ON” va avea în general efectul de a ridica temperatura camerei, dar acesta nu poate fi un efect garantat – dacă ușa de la cameră este deschisă de exemplu, atunci pornirea încălzirii poate să nu aibă efect. Componenta care ia decizia (extrem de simplă), aflată de regulă în partea electromecanică a termostatlui, implementează următoarele reguli:

prea rece → “încălzire ON”

temperatură bună → “încălzire OFF”

Sistemele de control mai complexe au desigur structuri de decizii considerabil mai bogate. Exemplele includ probe spațiale autonome, sisteme de control ale reactoarelor nucleare etc.

- Cei mai mulți “daemoni” software (precum procesele ce rulează în background sub sistemul de operare UNIX), care monitorizează mediile software și efectuează acțiuni pentru a-l modifica, pot fi văzuți ca agenți. Un exemplu ar putea fi programul XWindows **xbiff**. Acest utilitar monitorizează încontinuu sosirea tuturor mesajelor din poșta electronică a utilizatorului, și indică printr-o iconiță GUI dacă sunt mesaje necitite sau nu. Dacă termostatul din exemplul anterior era situat într-un mediu fizic – lume fizică – programul **xbiff** este situat într-un mediu software. El primește informații despre acest mediu prin îndeplinirea unor funcții software (executând programe sistem ca *ls*, de exemplu), iar acțiunile pe care le face sunt acțiuni software (schimbă o iconiță de pe ecran, sau execută un program). Componenta care ia decizii este la fel de simplă ca în exemplul termostatlui.

Pentru a rezuma, agenții sunt doar sisteme de calcul capabile să facă acțiuni autonome, în unele medii, pentru a realiza obiectivele dorite. În mod obișnuit, un agent va simți mediul său (prin senzori fizici în cazul agenților situați în părți ale lumii reale, sau prin senzori software în cazul agenților soft), și va avea la dispoziție un repertoriu de acțiuni care pot fi efectuate pentru a modifica mediul, care poate să răspundă nedeterminist la execuția acțiunilor respective.

1.2.2 Agenți Inteligenți

Nu suntem obișnuiți să ne gândim la termostate sau la “daemoni” UNIX ca la agenți, și în nici un caz ca la agenți inteligenți. Deci, când se consideră că un agent este *intelligent*?

Toate problemele inteligenței artificiale pot fi reunite sub conceptul de *agent intelligent*. Inteligența artificială se ocupă cu descrierea și construirea agenților care primesc percepții de la mediul înconjurător și execută acțiuni.

Agenții inteligenți reprezintă un important domeniu de cercetare și dezvoltare al Inteligenței Artificiale, domeniu ce oferă o nouă metodă de rezolvare a problemelor și un nou mod de interacțiune între calculator și utilizator.

Un **agent** este **intelligent** dacă agentul este capabil de acțiuni *autonome* și *flexibile* în mediul în care este situat, în scopul satisfacerii obiectivelor pentru care a fost proiectat. Prin *flexibilitate* se înțeleg următoarele trei caracteristici:

- *reactivitate (reactivity)*: agenții inteligenți au abilitatea de a-și percepe mediul și să răspundă oportun la schimbările ce apar, în scopul atingerii obiectivelor pentru care au fost proiectați;
- *pro-activitate (proactiveness)*: agenții inteligenți sunt capabili să dea dovadă de comportament specific, așa numitul “*comportament orientat spre scop*”, luând inițiative în vederea atingerii obiectivelor;
- *abilitate socială (social ability)*: agenții inteligenți sunt capabili să interacționeze cu alți agenți (posibil umani), în vederea atingerii obiectivelor.

În continuare, vom analiza pe rând aceste trei caracteristici care fac ca un agent să fie *intelligent*.

Mai întâi să considerăm *pro-activitatea*: comportament orientat spre scop. Nu este greu să se construiască un sistem care să prezinte un comportament orientat spre scop – o facem de fiecare dată când scriem o procedură în PASCAL, o funcție în C, sau o metodă în JAVA. Când scriem o asemenea procedură, o descriem prin *ipoteza* pe care se bazează (formal, precondiția sa), și prin *efectul* pe care îl are dacă ipoteza este validă (formal, postcondiția sa). Efectul procedurii este *scopul* acesteia: ce anume dorește programatorul ca acea procedură să realizeze. Dacă precondiția este adevărată în momentul în care este apelată procedura, atunci se presupune că procedura se va executa corect: se va termina, iar la terminare, postcondiția va fi adevărată, iar astfel scopul va fi atins. Acesta este comportamentul orientat spre scop: procedura este doar un plan sau o rețetă pentru a atinge scopul. Acest model de programare este bun pentru multe medii. De exemplu, funcționează bine când se consideră *sisteme funcționale* – acele sisteme care doar iau un x ca intrare și produc o anumită funcție $f(x)$ ca ieșire. Compilatoarele sunt un exemplu clasic de sisteme funcționale.

Dar pentru sistemele non-funcționale, acest model simplu de programare orientată spre scop nu este acceptabil, pentru că face presupuneri importante care limitează. În particular, presupune că mediul *nu se schimbă* atâta timp cât procedura se execută. Dacă mediul se schimbă, și în particular, dacă ipoteza (precondiția) de la baza procedurii devine falsă în timp ce procedura se execută, atunci comportamentul procedurii nu se poate defini – deseori aceasta se va bloca. De asemenea, se presupune că scopul, adică motivul lansării în execuție a procedurii, rămâne valid cel puțin până la terminarea execuției. Dacă scopul nu rămâne valid, atunci pur și simplu nu are rost să se continue execuția procedurii.

În multe medii, nici una din aceste presupuneri nu este validă. Mai exact, în domeniile *prea complexe* pentru ca un agent să le observe complet, cum sunt sistemele *multiagent* (mediile sunt populate cu mai mult de un agent care le poate schimba), sau acolo unde există *incertitudine* în mediu, aceste presupuneri nu sunt valide. În asemenea medii, a executa orbește o procedură, fără a ține cont dacă ipotezele de la baza acesteia sunt adevărate, este o strategie slabă. În asemenea medii dinamice, un agent trebuie să fie *reactiv*, în modul în care tocmai a fost descris. Adică trebuie să fie responsabil de evenimentele care apar în mediul său, când aceste evenimente afectează scopul agentului sau ipotezele care stau la baza procedurilor pe care agentul le execută pentru a-și atinge scopul.

După cum s-a văzut, construirea sistemelor pur orientate spre scop nu este grea. Construirea sistemelor *pur reactive* – acele sisteme care răspund în mod continuu la mediile lor – este de asemenea ușoară. Totuși, ceea ce reiese a fi mai complicat este construirea unui sistem care să aibă un echilibru eficient între comportamentul orientat spre scop și cel reactiv. Dorim agenți care să încerce să-și îndeplinească sistematic scopurile, poate prin folosirea unor modele complexe de acțiune asemănătoare procedurilor. Dar nu dorim ca agenții noștri să continue orbește execuția acestor proceduri într-o încercare de a atinge un scop atunci când este clar că procedura nu va funcționa, sau atunci când dintr-un anumit motiv, scopul nu mai este valid. În asemenea circumstanțe, dorim ca agenții să poată reacționa la situații noi, la timp pentru ca reacția să aibă un folos. Totuși, nu dorim ca agenții să reacționeze continuu, și astfel să nu se concentreze suficient asupra scopului pentru a-l putea îndeplini.

Reflectând asupra acestui lucru, nu ar trebui să fie o surpriză faptul că a atinge un bun echilibru între comportamentul orientat spre scop și cel reactiv este dificil. Această problemă este una din preocupările principale ale programatorului de agenți.

În sfârșit, să spunem ceva despre *abilitatea socială*, ultima componentă a definiției acțiunilor flexibil autonome. Într-un fel, abilitatea socială este trivială: în fiecare zi, milioane de calculatoare din întreaga lume schimbă constant informații atât cu oamenii, cât și cu alte calculatoare. Dar abilitatea de a schimba șiruri de biți nu este tocmai o abilitate socială. Trebuie să luăm în considerare faptul că, în lume, relativ puține dintre scopurile noastre pot fi atinse fără *cooperarea* cu alte persoane, despre care nu se poate presupune că împart aceleași scopuri cu noi – cu alte cuvinte, și celelalte persoane sunt autonome, cu propriile lor scopuri de atins. Pentru a ne îndeplini scopurile în asemenea situații, trebuie să *negociem* și să *cooperăm* cu ceilalți. Este posibil să fie necesar să înțelegem și să ne gândim la scopurile altora, să facem acțiuni (de exemplu să plătim cuiva bani) pe care altfel nu am fi ales să le facem, pentru a-i determina să coopereze cu noi, și astfel să ne atingem scopurile. Acest tip de abilitate socială este mai complexă, și mai puțin înțeleasă, decât simpla capacitate de a schimba informație binară.

Un *agent rațional ideal* va fi considerat acela care acționează în sensul maximizării performanțelor sale, pe baza informațiilor pe care le are despre mediu (furnizate de secvența perceptuală) și pe baza cunoașterii proprii pe care o are încorporată în structura sa internă.

Astfel, un agent inteligent artificial va fi înzestrat cu o cunoaștere inițială și cu capacitatea de a învăța. Învățarea asigură *autonomia* agentului – capacitatea de a-și deduce comportamentul din propria sa experiență. Agenții care operează doar pe baza cunoștințelor inițiale pe care le dețin vor fi numiți *neautonomi*, datorită faptului că ei

vor acționa cu succes doar în măsura în care aceste cunoștințe rămân valabile, pierzându-și astfel flexibilitatea.

Un agent inteligent are deasemenea o *funcție de performanță (utilitate)*, care măsoară performanța acțiunilor sale. Funcția de performanță este, în mod obișnuit, unică pentru toți agenții care acționează într-un mediu dat. Performanța este de altfel o funcție care asociază unei stări un număr real, ca fiind o măsură a gradului de succes al stării (o stare pe care agentul o preferă în raport cu alte stări are funcția de performanță mai mare).

Pe baza funcției de performanță agentul e capabil să rezolve așa numitele situații conflictuale în care, la un moment dat, poate opta pentru mai multe stări (succesor), nici una dintre acestea neputând fi sigur realizabile. În astfel de situații, funcția de performanță furnizează o cale prin care succesul poate fi ponderat pe baza importanței stărilor.

Toate programele agent vor avea același schelet: primesc percepții de la un mediu și generează acțiuni (Figura 1.3). Fiecare program agent va folosi structuri de date interne care vor fi reactualizate pe măsură ce apar noi percepții. Cu aceste structuri de date vor lucra procedurile de luare a deciziilor pentru a genera acțiuni posibile.

Partea de *program* a unui agent inteligent ar putea fi schematizată sub următoarea formă:

```
funcția AGENT-PROGRAM(percepție)  
  static : memorie  
    memorie ← ACTUALIZEAZĂ-MEMORIE(memorie, percepție)  
    acțiune ← ALEGE-ACTIUNEA-OPTIMĂ(memorie)  
    memorie ← ACTUALIZEAZĂ-MEMORIE(memorie, acțiune)  
    AGENT-PROGRAM ← acțiune  
sf-AGENT-PROGRAM
```

Figura 1.3. Scheletul unui agent inteligent

Un astfel de agent program utilizează anumite structuri interne de date, care sunt actualizate pe măsură ce un nou semnal sosește. Agentul primește la intrare un singur semnal, sub forma unei percepții asupra mediului înconjurător. La fiecare apel, memoria agentului este actualizată pentru a reflecta noul semnal primit. Apoi, pe baza unor proceduri de decizie, se va alege acțiunea optimă, care apoi va fi transmisă arhitecturii agentului pentru a fi executată, apoi se memorează faptul că acțiunea a fost aleasă.

Anumiți agenți program sunt construiți astfel încât secvența perceptuală primită la intrare nu este stocată în memorie, mai ales în situații complexe în care secvențele sunt mari și este inefficientă memorarea lor.

Priviți ca și sisteme (soft) complexe care au abilitatea de a se adapta în medii dinamice și schimbătoare, *agenții* ridică o problemă fundamentală și anume modul în care își folosesc “experiența” anterioară și informațiile senzoriale primite din mediu pentru a decide următoarele mișcări, modul în care să le abordeze și să colaboreze cu alți agenți.

1.2.3 Tipuri de agenți inteligenți

În funcție de informația folosită în procesul de decizie, există mai multe tipuri de agenți inteligenți. Programele agent diferă prin modul în care iau în considerare percepțiile, acțiunile, scopurile și mediul.

Câteva clase de agenți inteligenți sunt următoarele:

- agenți reflex (agenți bazați pe reflexe simple);
- agenți care păstrează un model a mediului;
- agenți bazați pe scop;
- agenți bazați pe utilitate.

Agenți reflex (bazați pe reflexe simple)

Un agent bazat pe reflexe simple (agent reflex) răspunde imediat la o secvență de intrare. Un agent reflex realizează o conexiune între situația curentă (percepută) și o acțiune. Aceste conexiuni sunt memorate sub forma unor reguli de producție sau reguli condiție – acțiune.

Programul unui agent reflex este foarte simplu. El caută regula a cărei parte de condiție se potrivește cu situația curentă (cum este ea definită de percepție) și efectuează acțiunea asociată cu această regulă.

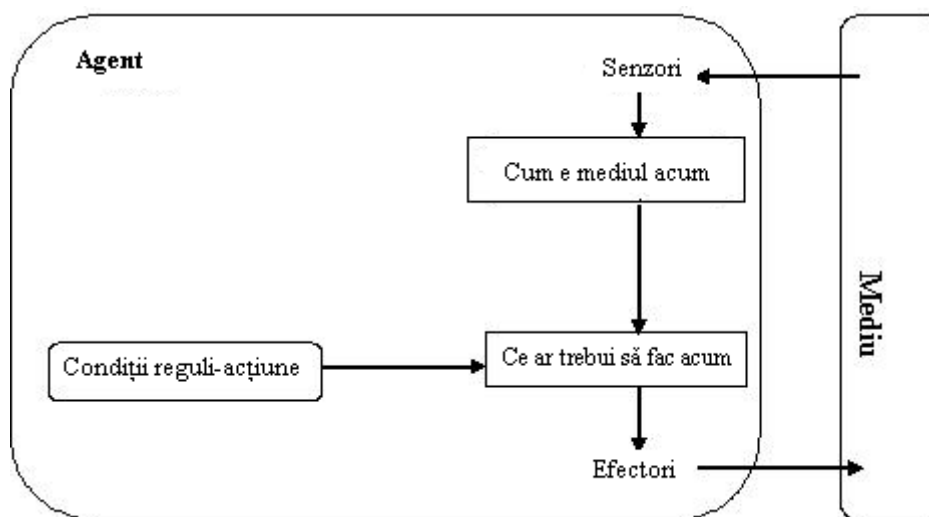


Figura 1.4. Agent reflex

Agenți care păstrează un model a mediului

Un agent uman alege o anumită acțiune în funcție de condițiile exterioare și de starea sa internă. În anumite situații un agent artificial trebuie să mențină o anumită informație asupra stării interne pentru a distinge între stări ale mediului care generează aceeași secvență perceptuală dar sunt în mod esențial diferite (necesită acțiuni diferite).

Modificarea stării interne necesită ca programul agentului să includă două tipuri de informații(cunoștințe):

- informații despre felul cum mediul (universul, domeniul de competență al agentului) evoluează independent de agent;
- informații despre felul în care acțiunile agentului afectează mediul.

Agentul este capabil să interpreteze o nouă secvență perceptuală (o nouă situație) în lumina cunoașterii pe care o are despre lume (codificată în starea internă). El folosește informația despre cum evoluează mediul pentru a păstra un model al părții necunoscute a mediului. Agentul trebuie să știe în ce fel acțiunile sale interferează cu starea mediului.

Unui agent reflex i se pot asocia stări interne. În acest caz starea curentă se combină cu starea internă pentru a genera o nouă stare curentă. Acțiunea agentului se bazează pe căutarea unei reguli a cărei condiție se potrivește cu situația curentă așa cum este definită de secvența perceptuală și de starea internă. Apoi se efectuează acțiunea asociată cu această regulă.

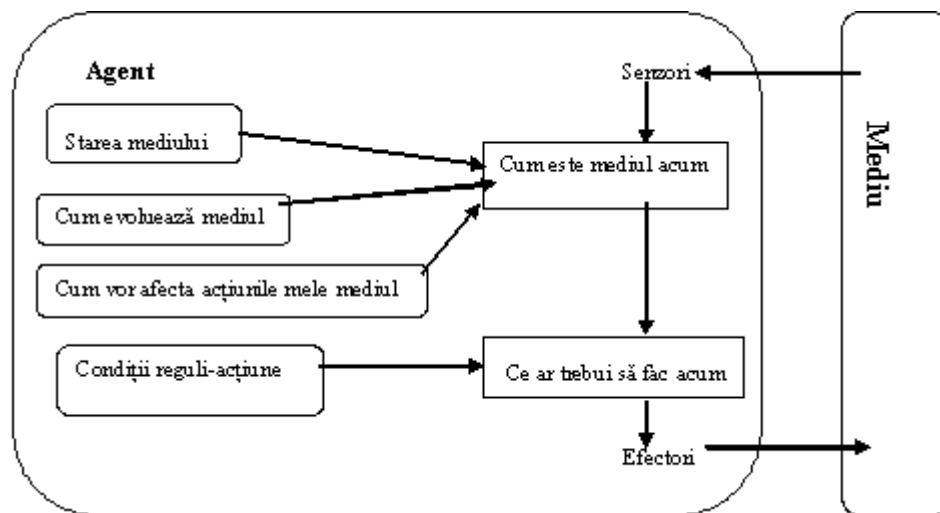


Figura 1.5. Agent care păstrează un model al mediului

Agenți bazați pe scop

Acești agenți acționează în sensul îndeplinirii scopului lor. Cunoașterea stării curente a mediului nu este întotdeauna suficientă pentru a selecta o acțiune corectă. Uneori agentul are nevoie și de o anumită informație asupra scopului, care descrie situațiile ce sunt de dorit. Programul agentului poate combina această informație cu cea despre rezultatele acțiunilor sale pentru a alege acțiunile care permit atingerea scopului propus.

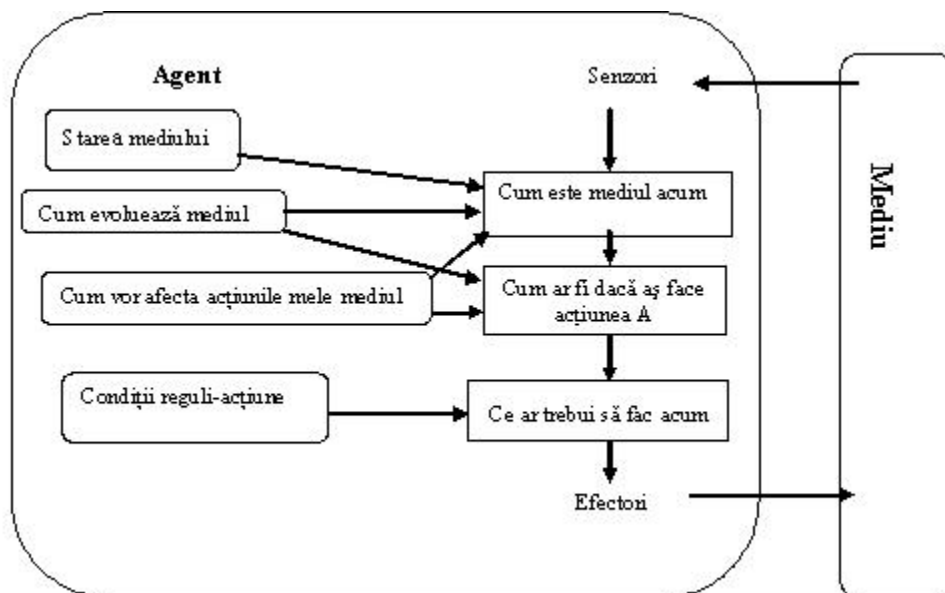


Figura 1.6. Agent bazat pe scop

Agenți bazați pe utilitate

Scopul nu este întotdeauna suficient pentru a genera cea mai bună strategie (sau comportare) a unui agent. Scopul realizează o distincție clară între stările favorabile și cele nefavorabile. Asociind fiecărei stări o măsură a performanței putem compara diferite stări, sau secvențe de stări, din punctul de vedere al satisfaciabilității sau utilității lor pentru agent.

Utilitatea este o funcție U definită pe mulțimea S a stărilor cu valori în mulțimea numerelor reale: $U : S \rightarrow \mathbf{R}$.

Funcția de utilitate permite luarea unor decizii raționale în cazul scopurilor conflictuale. În cazul în care există mai multe scopuri, dar nici unul nu poate fi atins cu certitudine, funcția de utilitate furnizează o metodă prin care probabilitatea de succes și importanța scopurilor vor putea fi comparate.

Putem admite că utilitatea unei stări măsoară gradul de satisfacție pe care o are agentul dacă atinge starea respectivă. Prin urmare un agent bazat pe utilitate tinde să-și maximizeze propria satisfacție.

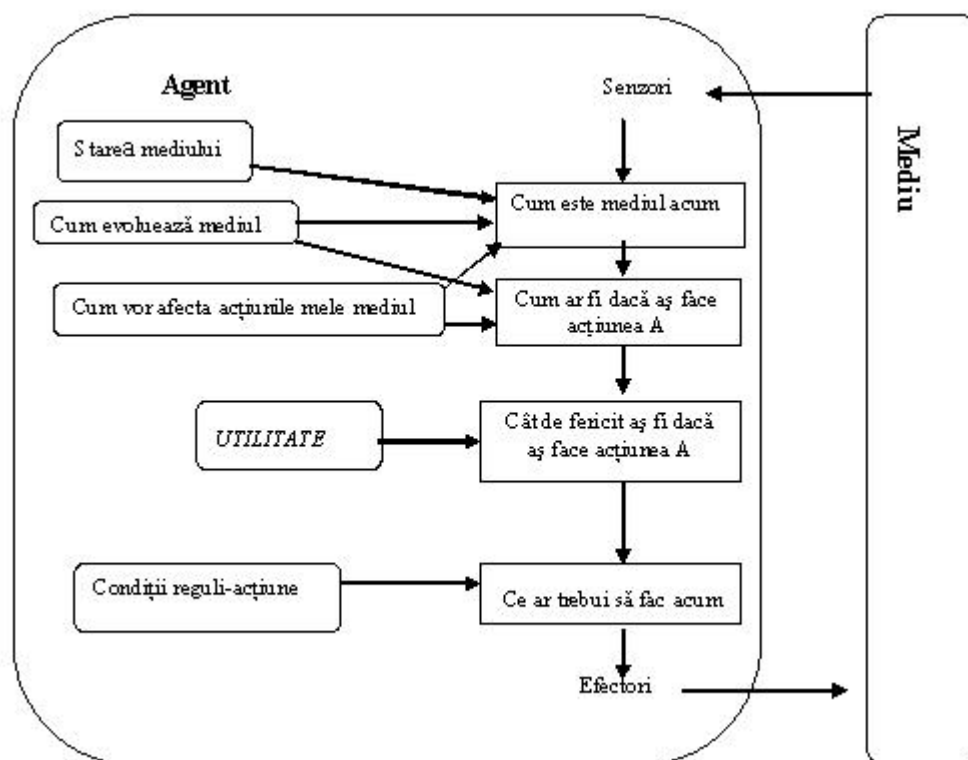


Figura 1.7. Agent bazat pe utilitate

1.3 Inteligența Artificială Distribuită

Inteligența Artificială Distribuită - IAD (Distributed Artificial Intelligence) este un subdomeniu al cercetărilor de Inteligență Artificială dedicate dezvoltării unor soluții distribuite pentru probleme complexe care se presupune că necesită inteligență.

Inteligența Artificială Distribuită se concentrează pe agenți ca “sisteme inteligente conectate” și pe inteligență ca o proprietate a sistemelor care interacționează. Ca urmare, IAD studiază procese sociale în grupuri de indivizi. Din acest punct de vedere, IAD poate fi privită ca fiind o generalizare a IA.

De ce Inteligență Artificială Distribuită?

Odată cu progresul cercetării în tehnologia informatică în ultimul deceniu, este cunoscut faptul că multe clase de probleme complexe nu pot fi rezolvate independent. Cercetările avansate în domeniul IAD au deschis căi de rezolvare a unor astfel de probleme complexe.

În general vorbind, domeniul IAD își propune să construiască entități inteligente care să interacționeze în mod productiv cu alte astfel de entități. Mai precis, IAD se ocupă cu studiul unei largi clase de probleme legate de distribuirea și organizarea (coordonarea, sistematizarea) cunoștințelor și acțiunilor în medii ce includ mai multe entități. Aceste entități, care sunt *agenții*, pot fi privite în totalitate ca o *societate*. Agenții lucrează împreună pentru a-și atinge atât propriile scopuri, cât și scopurile societății în ansamblu.

O distincție majoră în domeniul IAD este între cercetarea legată de *rezolvarea distribuită a problemelor* (RDP - *Distributed Problem Solving*) și cercetările în domeniul *sistemelor multiagent* (SMA - *MultiAgent Systems*), considerate subdomeniile de bază ale IAD.

În esență, lucrările în domeniul RDP abordează probleme și modul de a determina mai multe entități inteligente (calculatoare programate) să lucreze împreună pentru rezolvarea eficientă a acestora.

În SMA, agenții sunt *autonomi* și în mod normal *eterogeni*. Cercetările în acest subdomeniu sunt legate de organizarea și coordonarea comportamentelor inteligente într-o colectivitate de agenți autonomi - cum își organizează acești agenți cunoștințele, scopurile, abilitatea și planurile (strategiile) pentru a efectua *acțiuni* și a rezolva *probleme*. Într-un astfel de mediu, agentul poate să acționeze (lucreze) în vederea realizării unui singur scop global (comun tuturor membrilor colectivității) sau în vederea realizării unor scopuri individuale, care interacționează.

Ca și rezolvările de probleme în RDP, agenții în SMA pot partaja cunoștințe despre sarcinile și realizările lor parțiale. Spre deosebire de abordarea RDP, agenții trebuie, deasemenea, să raționeze despre procesul *coordonării* între agenți. Coordonarea este importantă în SMA; fără aceasta, beneficiile interacțiunii dispar, iar comportamentul grupului de agenți ar putea deveni haotic.

O altă diferență față de abordarea RDP, unde accentul se pune pe problema ce urmează a fi rezolvată, SMA se focalizează pe agent și caracteristicile sale în medii multiagent. Trei posibile puncte de vedere ale relației între RDP și SMA au fost identificate, fără însă a fi stabilit cu exactitate care este cel corect:

- RDP este un subdomeniu al SMA;
- SMA furnizează o alternativă de abordare pentru RDP;
- SMA și RDP sunt direcții de cercetare complementare.

O problemă centrală în IAD este cum să se permită agenților autonomi să aibă un model al altor agenți, care să le permită să raționeze despre activitățile acestora. Raționamentul despre alți agenți permite agenților să-și coordoneze activitățile în scopul producerii unor soluții elaborate, dar coerente. Coordonarea poate fi analizată sub formă a agenților efectuând planuri (strategii) independente ca să le realizeze scopurile.

Diferitele componente ale sistemului - scopuri, agenți, strategii și interdependențe - sunt asociate cu procesul de coordonare. Tabelul 1.1 prezintă aceste componente și procesele de coordonare asociate.

Componente	Procese asociate
Scopuri	Identificarea scopurilor, incluzând alegerea scopului
Agenți	Asocierea de scopuri agenților, incluzând distribuirea și negocierea scopurilor
Strategii	Asocierea de strategii scopurilor, incluzând planificarea
Interdependențe	Gestionarea interdependențelor, incluzând alocarea resurselor și sincronizarea

Tabel 1.1 Componentele coordonării

În general, se pune următoarea întrebare: *De ce să se aleagă abordarea IAD?*
Sunt patru motive principale:

- este necesară tratarea cunoștințelor distribuite, în aplicații care disparate geografic, cum ar fi rețele de senzori, controlul traficului aerian sau colaborarea între roboți. IAD poate, de asemenea, să soluționeze aplicații complexe;
- IAD ne poate ajuta în încercările de a extinde colaborarea om-mașină;
- IAD furnizează o nouă perspectivă în reprezentarea cunoașterii și rezolvarea problemelor prin intermediul formulărilor științifice și a unei originale reprezentări în practică;
- IAD poate aduce o nouă perspectivă asupra științelor cognitive și asupra Inteligenței Artificiale.