

# Modele multi-agent pentru sisteme ambientale inteligente

Manaŝe Bogdan, master IS, an 2

31 ianuarie 2011

## Abstract

Proiectarea și dezvoltarea de software pentru sistemele ambientale inteligente implică provocări dificile în funcție de diversitatea dispozitivelor și canalele de comunicație. Deoarece sistemele ambientale au cerințe imprevizibile și sunt sensibile la context, sistemele software trebuie să suporte schimbări dinamice. Caracteristicile unui sistem ambiental multi-agent orientat pe aspecte sunt prezentate ca un mecanism de raționament pentru comportamente interdependente.

## 1 Introducere

Sistemele ambientale inteligente vizionare se bazează pe mașini de calcul prevăzute cu un număr mare de procesoare și senzori de mici dimensiuni integrați în obiectele utilizate în viața de zi cu zi, acest lucru ducând la dispariția dispozitivelor tradiționale și a modului de prelucrare a informației. În cadrul sistemelor ambientale inteligente există un grad ridicat de partajare a datelor între unități larg distribuite, structurate pentru a promova colaborarea, auto-sincronizarea, răspunsuri rapide la informații noi, adaptabilitate și sustenabilitate.

Proiectarea sistemelor ambientale inteligente reprezintă o provocare pentru arhitectii software. Sistemele de acest tip implică participarea dispozitivelor heterogene, care formează un sistem deschis, dinamic în cadrul căruia resursele disponibile, contextul și activitățile se schimbă continuu. Sistemele multi-agent prezintă o abordare naturală și puternică pentru a proiecta sisteme ambientale inteligente care să funcționeze în medii complexe.

Cele mai importante caracteristici ale agenților inteligenți sunt anticiparea evenimentelor și adaptarea la schimbările mediului în care aceștia rulează. Agenții pot executa acțiuni proactive pentru a-și găsi scopurile și pentru a-și urma crezurile care țin de situațiile pe care le întâlnesc. De asemenea agenții pot comunica activ cu alți agenți pentru a atinge obiective mai ample.

În cadrul sistemelor ambientale există o nevoie clară de suport decizional pentru agenții, pentru a forma echipe de agenți și să-și asume roluri în cadrul echipelor.

## 2 Sisteme multi-agent

Un agent software reprezintă un sistem software încapsulat situat într-un mediu unde poate executa acțiuni flexibile și autonome pentru a-și atinge obiectivele de design. Scopul unui agent poate fi comun sau privat. Caracteristica cheie a unui agent software care este distinctivă față de alte paradigme de programare, este că într-un context, agenții evaluează persistent o suită de opțiuni care le sunt disponibile, apoi selectează una dintre acțiuni pe care o execută. În schimb, alte paradigme de programare sunt mult mai prescrise.

Într-un sistem ambiental, agenții sunt definiți pentru diferite roluri, cum ar fi monitorizarea activităților și intențiile utilizatorilor, monitorizarea stării în care se află alți agenți sau dispozitive, distribuirea sarcinilor computaționale între resursele disponibile și care reprezintă preferințele utilizatorilor și a altor agenți.

Agenții pot fi omogeni sau heterogeni și pot coopera sau pot fi competitivi. Agenții care reprezintă un sistem ambiental inteligent sunt heterogeni și cooperativi. Heterogenitatea rezultă din dispozitivele care simt diferite lucruri, activând diverse controllere, negociind terminarea diferitelor sarcini, și găsirea soluțiilor acceptabile pentru diferite modele. Aceștia sunt cooperativi în sensul că toți contribuie la realizarea scopului global.

Modelul BDI (crez-dorință-intenție) se poate ocupa flexibil de modelare întregului sistem bazat pe agenți într-un sistem ambiental inteligent. În acest model credințele corespund stării în care se află agentul, inclusiv caracteristicile curente ale agentului în mediul în care funcționează. Dorințele corespund efectelor pe care agentul încearcă să le producă în mediul în care funcționează. Intențiile reprezintă planurile pe care agentul le are disponibile și pe care le urmează pentru a produce efecte.

Unii agenți pot fi compuși din subagenți, aceștia pot fi agenți negociatori, brokeri sau agenți care se folosesc de serviciile altor agenți. Agenții care au capacități de comunicare, specifică modul de lucru în cadrul rețelei de comunicare și se ocupă de asemenea de dispozitivele care se află în raza lor de ascultare și care pot comunica cu succes. Problemele de protocol, cum ar fi responsabilitatea pentru agenții logici să expedieze anumite pachete de date, sunt manipulate de către agenți însărcinați cu alegerea, aplicarea și modificarea protocoalelor după cum este necesar. Agenții trebuie să înțeleagă sintaxa, să parcurgă și să proceseze mesajul pentru a acționa conform informației primite.

Agenții pot fi mobili, având posibilitatea să se deplaseze în cadrul platformelor suportate. În cadrul unui sistem ambiental inteligent, agenții mobili pot modela preferințele agenților brokeri, utilizatorilor și agenților care se ocupa cu furnizarea datelor. Acești agenți pot răspunde adaptiv la condițiile dinamice, folosind mai multe opțiuni disponibile pentru a realiza o comandă sau o cerere.

Mobilitatea este o preocupare intersectorială care nu este prezentă în majoritatea sistemelor multi-agent, dar poate fi o unealtă puternică sau chiar necesară în domeniu sistemelor ambientale inteligente.

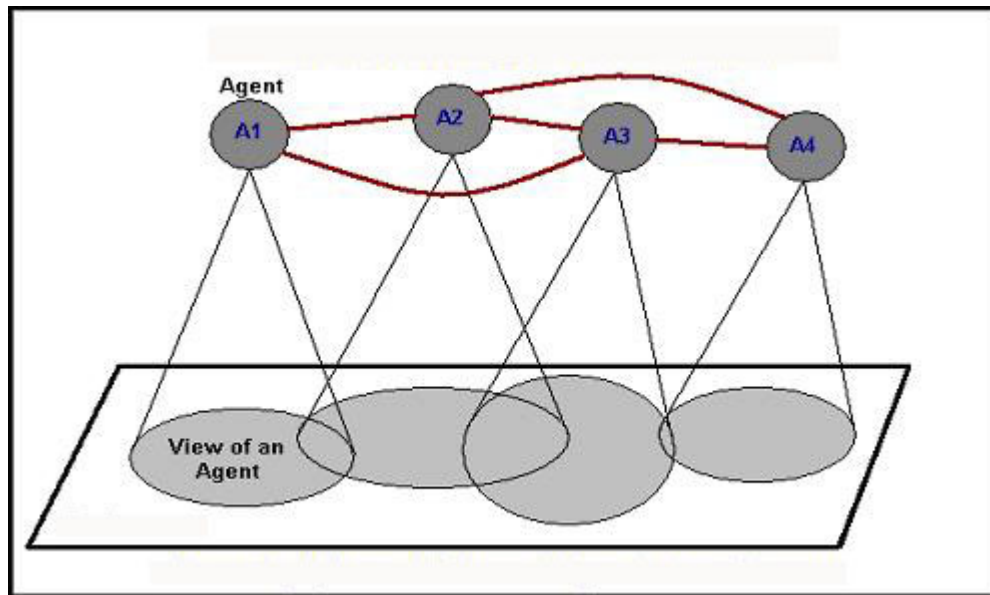


Figure 1: Relațiile organizaționale între agenți

### 3 Probleme intersectoriale

Necesitatea de a gestiona problemele complexe, în timp ce se dorește crearea unui sistem software care să fie flexibil, adaptabil și care să evolueze, este esențială în sistemele ambientale inteligente. Cu toate că modularitatea este esențială, unele cerințe și proprietăți nu pot fi modularizate deoarece acestea sunt legate de unele probleme intersectoriale. Problemele intersectoriale sunt dificile, deoarece sunt distribuite în mod inerent și fac celălalte componente software să fie mai puțin independente. Într-un sistem multi-agent, de exemplu, comunicarea securizată între agenții care reprezintă entități decizionale reprezintă o preocupare importantă privind funcțiile lor specifice. În ingineria software, este cunoscut faptul că, problemele intersectoriale pot duce cu ușurință la dublificarea codului în cadrul mai multor componente. De astfel duplicarea codului duce la scăderea calității software, aceasta fiind dificil de rezolvat la nivelul arhitectural.

O aplicație cu numeroase probleme intersectoriale, are ca rezultat un soft care este dificil de modularizat, înțeles, reutilizat sau dezvoltat. Pentru rezolvarea acestor probleme, este util să se folosească un nivel ridicat de abstractizare pentru a face designul softului mult mai ușor de gestionat. Utilizarea unui nivel ridicat de abstractizare poate scoate în evidență unele probleme și poate oferi o perspectivă asupra modului de încapsulare ale acestora în componente separate. Astfel este posibil să se descrie cu precizie modul în care componentele interacționează și dacă sistemul se comportă conform scopului propus.

Unele sisteme multi-agent gestionează complexitatea prin separarea prob-

lemelor în agenți, obiecte și mediul în care aceștia interacționează. Unele probleme ale sistemelor, cum ar fi securitatea, nu se pot rezolva prin modularizare.

Dezvoltarea orientată pe aspecte (ASOD) abordează modularizarea problemelor intersectoriale, prin separarea problemelor în module aspect. Codul care adresează o problemă intersectorială este numit înștiințare și este menținut în interiorul aspectului. Aspectul cataloghează locurile din interiorul sistemului care au nevoie de o înștiințare și trimite o înștiințare la locațiile respective. Într-un sistem ambiental inteligent problemele sunt asociate cu interacțiunea, adaptarea, autonomia agenților, mobilitatea, capacitatea de a învăța și colaborarea.

## 4 Modele formale și agenți

Agenții dintr-un sistem multi-agent trebuie să prezinte atât comportamente reactive cât și comportamente proactive. De exemplu, un agent care monitorizează proactiv și care prezice locația și disponibilitatea resurselor prescrise, ar putea suspenda monitorizarea dacă i se cere să aloce cât mai repede resurse reactiv, ca răspuns la o schimbare percepută în preferințele și planurile utilizatorului. De asemenea unii agenți trebuie să rezolve conflicte între obiective multiple. Suspendarea activităților poate duce la pierderea unor schimbări critice în cadrul informațiilor, care la rândul lor pot produce acțiuni care nu au fost solicitate.

## 5 Suport decizional

Alocarea de sarcini este o funcționalitate critică într-o arhitectură orientată agent pentru sisteme ambinentale inteligente. Din punct de vedere cooperativ și executare a deciziilor inteligente este necesar un cadru de modelare. Modelele disponibile sunt reguli de decizie fuzzy, rețele neuronale, plase de contract și seturi rough. În continuare voi descrie un model de optimizare distribuit. Se presupune că există o colecție de obiective care trebuiesc îndeplinite și mediul în care acționează agenții care trebuie să îndeplinească obiectivele este în continuă mișcare. Scopurile sunt îndeplinite prin formarea de echipe de agenți cooperativi care sunt capabili să îndeplinească obiectivele la un nivel măsurabil de calitate. Agenții dintr-o oarecare echipă îndeplinesc un rol care contribuie la realizarea obiectivului global. Scopul modelului este de a specifica deciziile optime privind crearea echipelor și includerea agenților în echipe. Echipele pot avea un număr diferit de roluri, iar agenții sunt asociați doar rolurilor la care aceștia pot contribui.

## 6 Concluzii

Utilizarea metodelor formale are avantaje datorită suportului pentru specificații, design, validare și verificare. Designul cuprinde nevoia de adaptare dinamică și non-determinismul sistemelor ambientale inteligente. Utilizarea conceptelor

orientate obiect și metodelor formale în cadrul sistemelor multi-agent promite gestionarea facilă a complexității software.