



Universitatea Politehnica București
Facultatea de Automatică și Calculatoare
Departamentul de Automatică și Ingineria Sistemelor

LUCRARE DE LICENȚĂ

**Sisteme multi-agent pentru controlul și mentenanța unui sistem
de climatizare în clădirile inteligente**

Absolvent
Alexandru-Florentin Burcea

Coordonator
Sl. dr. ing. Monica Pătrașcu

București, 2014

Cuprins

1. INTRODUCERE.....	1
2. AGENȚI ÎN CLĂDIRILE INTELIGENTE.....	2
2.1. Agenți	2
2.2. Sisteme multi-agent (MAS).....	3
2.3. Smart Building	6
2.4. Modele de sisteme multi-agent implementate.....	6
2.4.1. Sistem multi-agent pentru detectarea intrușilor în rețele locale	6
2.4.2. CitySCAPE – Sistem multi-agent	8
3. STUDIU DE CAZ: SISTEM DE CLIMATIZARE.....	10
3.1. Mediul de programare: JADE.....	10
3.2. Sistemul de climatizare (Sistem HVAC).....	12
3.3. Modul de implementare.....	16
3.3.1. Agentul care simulează comportamentul camerei.....	17
3.3.2. Implementarea agentului care simulează senzorii	19
3.3.3. Implementarea setului de reguli în agentul de inferență	21
3.3.4. Implementarea agenților de execuție.....	22
3.3.5. Implementarea agentului de mentenanță	22
3.4. Scenarii de simulare	23
3.4.1. Controlul mărimilor când există câte un senzor de același tip.....	23
3.4.2. Controlul mărimilor pentru mai mult de doi senzori de același tip	26
3.4.3. Verificarea conductelor de aer și a senzorilor de viteză	29
3.4.4. Verificarea senzorilor de temperatură și a geamurilor	30
4. CONCLUZII.....	31
5. ANEXE.....	32
6. BIBLIOGRAFIE.....	39

1. INTRODUCERE

Deși în urmă cu câteva decenii singurele tehnologii întâlnite în clădiri erau liniile de comunicație telefonică, cablurile de televiziune, sistemele de încălzire sau sistemele de ventilație, în ziua de astăzi tehnologia de ultimă generație este tot mai întâlnită la fiecare pas și odată cu introducerea cuvântului de „smart” în vocabular, saltul de la automatizarea unui banal, la prima vedere, automat de cafea la automatizarea unor sisteme cu un grad de complexitate mult mai ridicat și folosind tehnologii din ce în ce mai noi precum inteligența artificială este iminent.

Principalele funcționalități ale unei clădiri de tipul *Smart Building* sunt controlul și monitorizarea componentelor non-structurale precum sistemul de climatizare, sistemul electric sau cel de securitate.

O funcție de bază a unei astfel de clădiri este reprezentată de controlul mediului în care se desfășoară activitățile ceea ce presupune atât un confort de trai sporit, cât și o productivitate sporită în funcție de scopul clădirii (birouri, locuințe etc.)

Scopul acestei lucrări este de a implementa o soluție bazată pe sisteme multi-agent pentru controlul și mentenanța unui sistem de climatizare în clădirile inteligente.

În primul capitol am prezentat scopul acestei lucrări și motivația alegerii temei.

În al doilea capitol am prezentat atât conceptele de agenți și sisteme multi-agent cât și conceptul de smart building, termenii ce stau la baza acestei lucrări, precum și exemple de sisteme multi-agent implementate.

În al treilea capitol am prezentat mediul de programare în care a fost construită aplicația și modul de funcționare al acestuia, sistemul de climatizare propus în vederea realizării aplicației precum și modurile de implementare ale unor agenți și scenarii de simulare.

În al patrulea capitol am sintetizat concluziile, contribuțiile personale și direcțiile viitoare de lucru.

2. AGENȚI ÎN CLĂDIRILE INTELIGENTE

2.1. Agenți

În (Maes, 1995) se definește că agenții sunt sisteme de calcul care populează medii complexe și dinamice, detectează și acționează autonom în acest mediu și făcând așa, realizează un set de obiective și de sarcini pentru care au fost creați.

Un agent duce la bun sfârșit activitățile sale într-o manieră flexibilă și inteligentă fără îndrumarea constantă a utilizatorului sau intervenții. Lucrând autonom, un agent ideal trebuie să fie suficient de inteligent să poată învăța din propriile experiențe (Gilbert et al., 1995). Asemenea unui agent uman care folosește ochii, nasul sau alte organe drept senzori și mâinile, picioarele sau alte părți ale corpului pe post de elemente de execuție, un agent poate avea diferiți senzori (optici, magnetici, etc.) pentru receptarea informațiilor din mediul din care face parte și diferite motoare ca elemente de execuție. (Russell & Norvig, 2003).

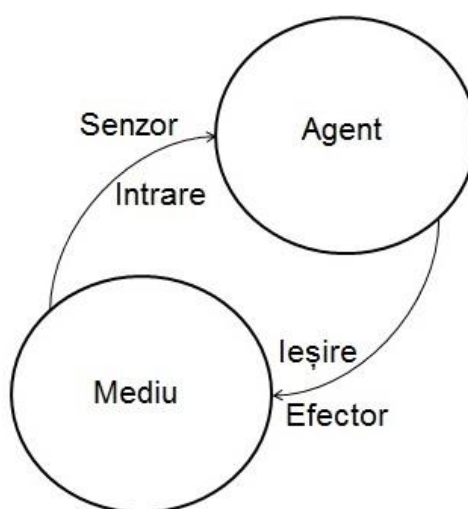


Figura 2.1.1. Un agent și mediul său

În majoritatea domeniilor de complexitate medie, un agent nu poate manipula pe deplin mediul din care face parte. În cel mai bun caz, acesta va avea un control parțial, în care acesta îl va putea influența. Astfel, din punctul de vedere al unui agent, înseamnă că aceeași acțiune poate avea efecte complet diferite sau poate să nu aibă efectul dorit chiar dacă aceasta a fost executată de două ori sub circumstanțe aparent identice. De aceea, agenții trebuie să fie pregătiți pentru un eventual eșec. Așadar, putem concluziona că, de regulă, mediile se presupun a fi nedeterminate (Wooldridge, 2002).

Pe baza proprietăților din tabelul 2.1. agenții se pot clasifica în diferite clase. Prin definiție, orice agent trebuie să satisfacă primele patru proprietăți. Adăugând alte proprietăți putem produce alte clase utile de agenți precum cei adaptivi sau mobili. De asemenea, agenții se pot clasifica și după alte criterii. De exemplu, aceștia se pot clasifica după sarcinile pe care le îndeplinesc, după arhitectura lor sau după aria și eficiența acțiunilor acestora (Franklin & Graesser, 1996).

Proprietate	Semnificație
Reactivitate	Răspunde în timp util la schimbările mediului
Autonomie	Deține control asupra propriilor acțiuni
Deliberare	Nu acționează pur și simplu ca un răspuns la mediu
Continuitate	Reprezintă un proces continuu
Comunicativitate	Comunică cu alți agenți, incluzând și oamenii
Adaptivitate	Își schimbă comportamentul pe baza experiențelor anterioare
Mobilitate	Capabil să se transporte de la o mașină la alta
Flexibilitate	Acțiuni bine definite
Caracter	Stare emoțională de tip personalitate

Tabelul 2.1.1. Proprietățile agenților (Franklin & Graesser, 1996)

Diferența dintre un agent și un mediu multi-agent este destul de simplă. De exemplu, un agent jucându-se tetris este clar un mediu cu un singur agent, în timp ce un agent jucând tenis reprezintă un mediu multiagent (Russel & Norvig, 2003).

2.2. Sisteme multi-agent (MAS)

Agenții există și acționează în medii care, de regulă, sunt atât virtuale cât și fizice. Mediul poate fi deschis sau închis și poate sau nu poate să mai conțină și alți agenți. Deși există situații în care un agent poate funcționa de unul singur în mod util, înmulțirea rețelelor de calculatoare a făcut ca această situație să devină tot mai rară și un agent trebuie să interacționeze cu alți agenți.

Sistemele multiagent reprezintă cea mai bună metodă de a descrie sau de a proiecta sistemele de calcul distribuite. Procesarea informației este omniprezentă. Procesoarele sunt înglobate peste tot în mediul înconjurător, în sistemul de climatizare al unei case, în sistemul de energie electrică din bucătărie sau în cafetiera din aceasta.

Un sistem multi-agent reprezintă un sistem cu un număr de agenți capabili să interacționeze între ei. Pentru a putea interacționa cu succes, aceștia necesită anumite abilități precum cooperarea, coordonarea sau negocierea.

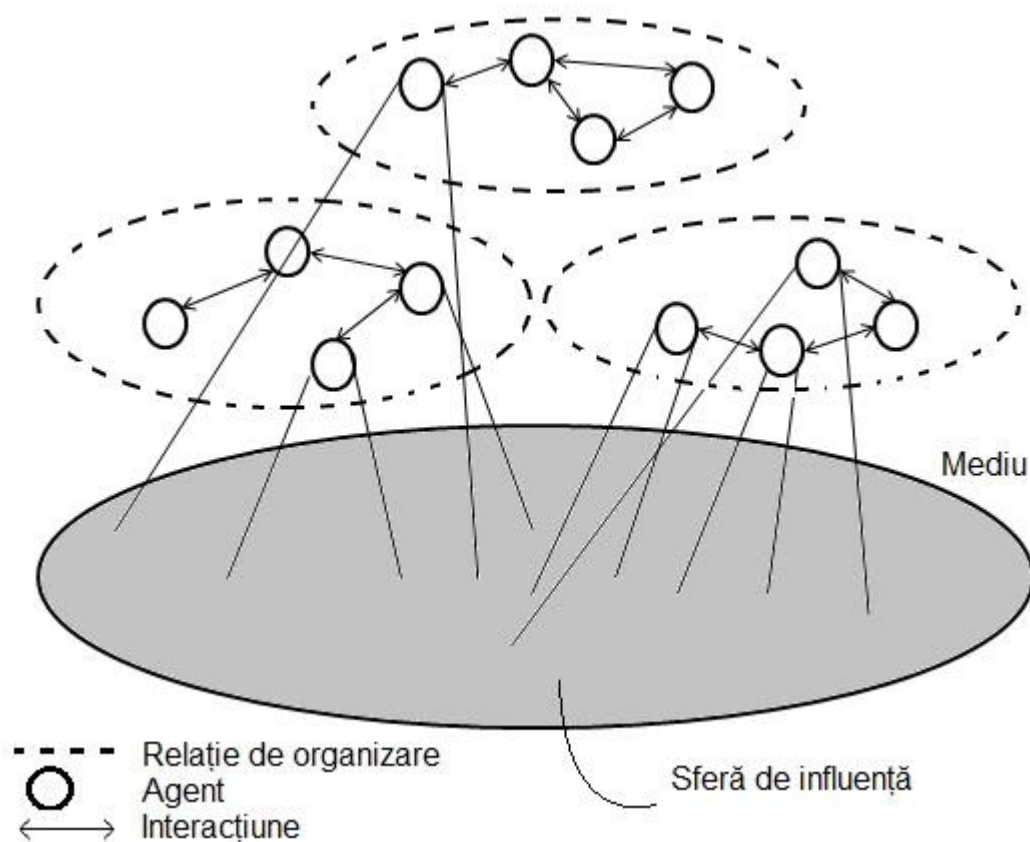


Figura 2.2.1. Structura tipică a unui sistem multi-agent (Wooldridge, 2002)

Un mediu poate fi influențat atât de unul dintre agenți, cât și de mai mulți sau nici unul, caz în care ieșirea funcției va fi independentă de acțiunile agenților. (Wooldridge, 2002).

Cum am afirmat, agenții necesită anumite abilități pentru a putea interacționa cu succes, iar în ceea ce privește cooperarea, o problemă evidentă este aceea de a ajunge la o înțelegere între agenții interesați numai de sine. Aptitudinea de a ajunge la o înțelegere (fără ajutorul unei a treia părți) este o capacitate fundamentală a agenților inteligenți și autonomi. Capacitățile de negociere și de argumentare stau la baza abilității unui agent de a ajunge la o înțelegere (Wooldridge, 2002).

Coordonarea reprezintă o caracteristică a unui sistem de agenți care execută anumite activități într-un mediu partajat. Gradul de coordonare reprezintă măsura în care aceștia evită activitatea neesențială reducând întrecerea pentru resurse, evitând livelock-ul și deadlock-ul și menținând condiții de siguranță adecvate (Weiss, 1999).

În (Andreadis et al., 2013) se specifică faptul că într-un sistem multi-agent, coordonarea se clasifică după modul în care aceasta se realizează:

- coordonare centralizată (agenți coordonativi) – în aceasta, un agent central primește datele parțiale de la restul agenților, construiește un plan pe baza acestora sau un set de reguli și rezolvă posibilele conflicte

- coordonare descentralizată (agenți autonomi) – agenții nu sunt controlați de către un agent central, deși aceștia comunică între ei pentru atingerea scopurilor și soluționarea problemelor

Totodată, modul de coordonare definește și arhitectura sistemului multi-agent:

- arhitectura centralizată – în acest tip de arhitectură se identifică un agent coordonator și unul intermediar. Agentul coordonator asigură comunicația în cadrul sistemului și este responsabil cu transmiterea mesajelor între agenți. Agentul intermediar se ocupă cu luarea deciziilor de prioritate scăzută, dar numai în situații cruciale. Printre responsabilitățile agentului intermediar se numără: interpretarea mesajelor și descompunerea sarcinilor.
- arhitectura descentralizată – este alcătuită din agenți autonomi care nu sunt controlați de un alt agent sau o persoană, comunică imediat cu alți agenți din cadrul aceluiași sistem sau altul și se dedică propriilor planuri.

Negocierea și cooperarea reprezintă tot o formă de coordonare, prima fiind specifică agenților competitivi sau interesați numai de sine, iar cea de-a doua este specifică agenților care nu sunt adversari.

Protocoalele de comunicație permit agenților să interschimbe și să înțeleagă mesajele, iar protocoalele de interacțiune permit agenților să aibă conversații.

Protocoalele de comunicație sunt de obicei specificate pe niveluri. Cel mai de jos nivel al protocolului specifică metoda de interconectare, nivelul mediu specifică structura și sintaxa informației care se transferă, iar nivelul superior specifică semantica informației. Semantica nu se referă doar la conținutul mesajului, ci și la tipul acestuia.

La începutul anilor 1990, s-a format Knowledge Sharing Effort (KSE) cu scopul de a dezvolta protocoale pentru interschimbarea informațiilor între sistemele informatice autonome. Din aceasta au derivat următoarele două limbaje de comunicare:

- The Knowledge Query and Manipulation Language (KQML) care reprezintă un limbaj pentru comunicarea între agenți. Acesta definește o structură de tip "anvelopă" pentru mesaje în care agentul formulează explicit ce efecte poate avea mesajul. Așadar, KQML nu se preocupă cu conținutul mesajului.
- The Knowledge Interchange Format (KIF) este un limbaj care a fost explicit proiectat pentru a permite reprezentarea anumitor cunoștințe legate de domeniul de conversație. A fost inițial destinat formării conținutului mesajelor de tipul KQML. (Wooldridge, 2002)

Într-un mediu cu resurse limitate, agenții trebuie să își coordoneze activitățile pentru a-și vedea propriul interes și pentru a putea satisface scopurile grupului. Acțiunile mai multor agenți trebuie să fie coordonate deoarece între aceștia se stabilesc anumite relații de dependență. Un exemplu de coordonare este furnizarea informațiilor la timp asigurând sincronizarea între agenți (Weiss, 1999).

Independență	Nu există dependență între agenți
Unilaterală	Un agent depinde de altul, dar nu și viceversa
Reciprocă	Amândoi agenții depind unul de celălalt pentru același scop
Bilaterală	Agenții depind unul de celălalt pentru scopuri diferite

Tabelul 2.2.1. Relații de dependență (Wooldridge, 2002)

2.3. Smart Building

În (Pătrașcu & Drăgoicea, 2014), termenul de Smart Building (SB) se definește ca fiind un set cuprinzător de abordări, tehnologii, instrumente, metode și mijloace construit ca să sporească conștiința afacerilor și a cetățenilor europeni în ceea ce privește termenii de mediu, siguranță și confortul traiului. Acesta poate fi abordat în stiluri diferite, pe de o parte, prin legislație, inițiativele cetățenilor și ale diferitelor corporații care vor să implementeze o mai bună separare și instalare a resurselor de energie regenerabilă și pe de altă parte, prin controlul și monitorizarea mai bună a siguranței și a energiei în clădire. Totodată, termenul de Smart Building se referă la un model care în ultimii ani a fost într-o continuă dezvoltare ce încearcă să definească noi soluții pentru optimizarea energiei. Acestea sunt proiectate pentru a funcționa mai eficient și pentru a putea interacționa cu sistemele din jur.

Acest tip de clădiri asigură informații despre o clădire sau un spațiu în interiorul acesteia, pentru a permite proprietarului sau locatarului clădirii să gestioneze clădirea sau spațiul. SB furnizează cele mai eficiente abordări din punctul de vedere al costului referitor la arhitectura și implementarea sistemelor tehnologice ale clădirii. Modul tradițional de a proiecta și construi o clădire este proiectarea, instalarea și operarea fiecărui sistem în parte. (Sinopoli, 2010).

2.4. Modele de sisteme multi-agent implementate

2.4.1. Sistem multi-agent pentru detectarea intrușilor în rețele locale

De vreme ce este imposibil de prezis și identificat în prealabil toate punctele slabe ale unei rețele, iar pătrunderea intrușilor într-un sistem nu poate fi întotdeauna prevenită, sistemele pentru detecția intrușilor sunt componente esențiale care asigură securitatea unui sistem de rețea.

Pentru a fi eficiente în ceea ce privește realizarea funcțiilor lor, sistemele pentru detecția intrușilor trebuie să fie precise, adaptive și extensibile. Având în vedere aceste cerințe stricte și nivelul ridicat de vulnerabilitate al rețelelor din prezent, proiectarea unui astfel de sistem a devenit o cerință foarte dificilă. Deoarece sistemele distribuite pentru detecție au numeroase dezavantaje precum rezultatele fals pozitive sau eficiența scăzută în ceea ce privește detecția, cu ajutorul agenților autonomi în (Sen, 2010) s-a proiectat un nou sistem distribuit care să fie capabil să identifice eventualele breșe de securitate, să reducă numărul alarmelor false și să îmbunătățească performanțele comparativ cu sistemele deja existente.

Sistemul utilizează supravegherea coordonată cu ajutorul agenților implementați și calculul distribuit pentru a identifica semnele premature unui atac și pentru a recunoaște cazurile care pot apărea înainte ca un atac să aibă loc.

Principalele scopuri ale sistemului:

- Detecția activităților nedorite – pe lângă capacitatea de a răspunde unui atac, sistemul trebuie să fie capabil să alerteze administratorul de fiecare dată când recepționează orice semn care poate preceda un atac.
- Identificarea și izolarea gazdelor compromise – de asemenea, sistemul trebuie să fie capabil să localizeze și să izoleze nodurile compromise

În ceea ce privește arhitectura sistemului, agenții sunt văzuți ca fiind autonomi, reflexivi, cooperativi și deliberativi și sunt responsabili cu achiziționarea și analizarea datelor precum și crearea mecanismului de inferență adecvat.

Agenții sunt grupați în câteva subdomenii. Cei care fac parte din același subdomeniu trebuie să comunice frecvent și în mod activ, în timp ce comunicația dintre agenții care fac parte din domenii adiacente este rară. Totodată aceștia au cunoștințe legate de arhitectura rețelei și de modele de atacuri de bază.

Sistemul propus conține trei tipuri de agenți:

- Agenții care monitorizează sistemul – responsabili cu achiziționarea, transformarea și distribuirea datelor la cerere. Publică detaliile variabilelor, pe care le monitorizează, care ulterior pot fi utilizate de către alți agenți.
- Agenții care monitorizează breșele – se abonează la datele publicate de către agenții care monitorizează sistemul, dar și la datele publicate de către agenți de același tip.
- Agenții de registru – stochează numele și locațiile agenților pentru facilitarea comunicației.

Fiecare agent este alcătuit din șase module:

- Modulul de percepție – responsabil cu achiziționarea datelor de audit sau a datelor rețelei subdomeniului din care face parte agentul.
- Modulul de dezbatere – responsabil cu analizarea datelor achiziționate de modulul de percepție.
- Modulul de comunicație – permite unui agent să comunice părerile, deciziile și cunoștințele sale altor agenți din același subdomeniu.
- Modulul de acțiune – ia decizii corespunzătoare când se detectează o breșă. În momentul în care un agent sesizează că un nod depășește pragul unui atac cunoscut, declanșează o alarmă specifică atacului respectiv și o comunică administratorului sistemului. De asemenea, dacă agentul nu recunoaște tipul atacului, va declanșa o alarma ce indică o "situație anormală" pe care administratorul o va interpreta confirmând atacul sau respingând alarma în cazul în care aceasta se dovedește a fi falsă

- Modulul de actualizare – în cazul în care administratorul confirmă o ”situație anormală”, aceasta va fi recunoscută drept ”atac cunoscut” și va fi tratată ca atare în cadrul viitoarelor monitorizări.
- Modulul de supervizare – modulul central care coordonează sarcinile și interacțiunile dintre celalalte module

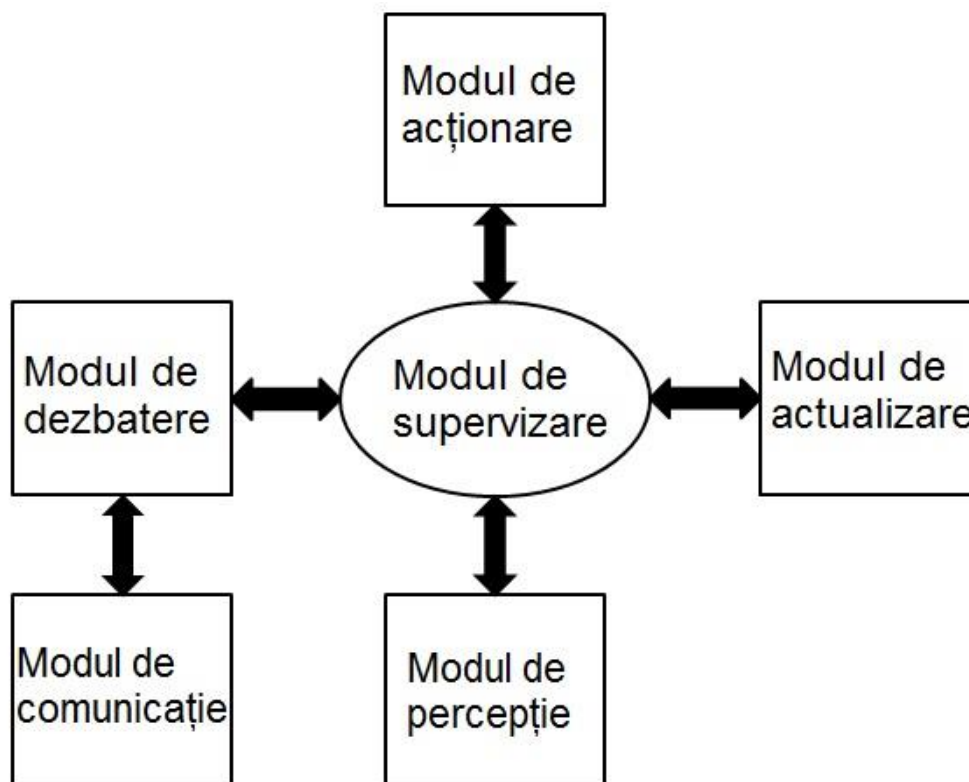


Figura 2.4.1.1. Arhitectura unui agent dintr-un nod (Sen, 2010)

2.4.2. CitySCAPE – Sistem multi-agent

În (Pătrașcu et al., 2014) este prezentată crearea în timp real a unei bucle de control pentru o intersecție cu cale ferată în oraș, cu ajutorul sistemelor multi-agent.

Un sistem urban este atât mare cât și complex, pentru a putea fi proiectat cu ajutorul unor mijloace convenționale. Dat fiind scopul acestui sistem, de a găzdui și proteja viața umană, cerințele pentru dezvoltarea unei asemenea arhitecturi de control pentru un oraș, trebuie să aibă integrată și componenta socială.

A fost propusă o arhitectură denumită CitySCAPE care este văzută ca o structură ierarhică, ce implementează componente descentralizate la nivelurile inferioare și componente centralizate la nivelurile superioare care se ocupă cu integrarea subsistemelor în sistem. Această arhitectură este compusă din inSCAPE (rețea integrată CitySCAPE) care asigură integritatea structurală și eSCAPE (emergent CitySCAPE) care asigură siguranța oamenilor.

La termenul de eSCAPE s-a ajuns pe baza proprietăților emergente ale sistemelor adaptive complexe. Rolul său este de a modela, controla și monitoriza siguranța oamenilor. Subsistemul gestionează comportamentul uman și sistemul social atât în modul de funcționare normal cât și în situațiile în care este nevoie de un răspuns urgent.

eSCAPE este reprezentat de un sistem multi-agent scalabil care a fost construit pe niveluri, fiecare nivel reprezentând un țesut, iar fiecare agent o celulă. Celula minimală are patru componente:

- un dispozitiv senzorial, D_S , care poate citi și procesa date din mediul exterior cu ajutorul senzorilor sau traductoarelor
- un efector, D_A , care poate acționa și produce schimbări asupra mediului
- un dispozitiv de inferență, D_I , capabil să ruleze algoritmi de inferență
- un agent nucleu, N , care coordonează celula

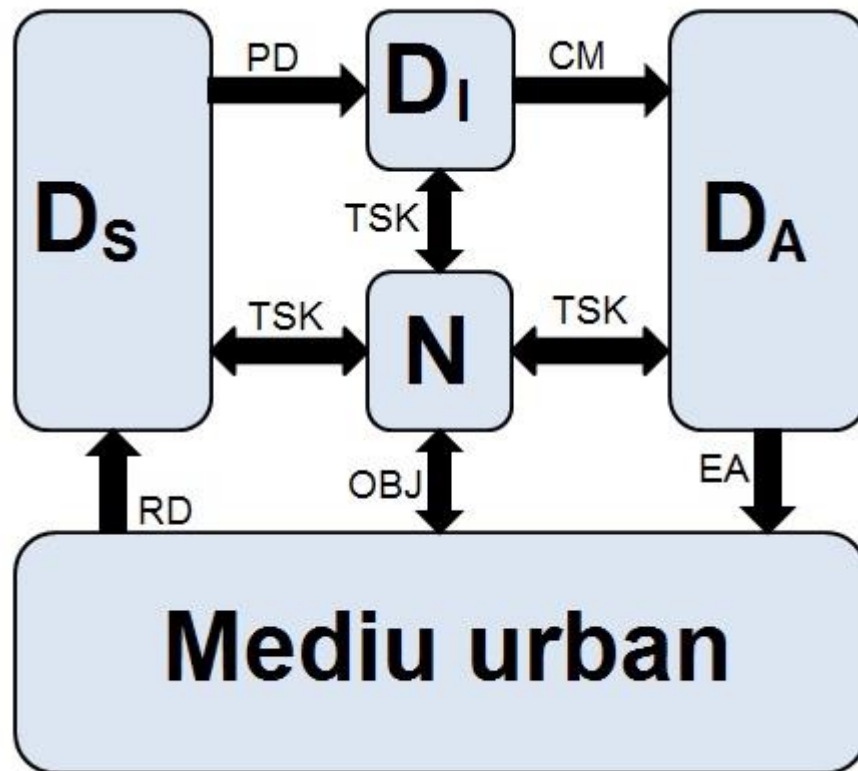


Figure 2.4.2.1. Celulă minimală

- RD – date recepționate
- PD – date procesate
- CM – comenzi calculate
- EA – acțiuni executate
- OBJ – obiective cu un grad mai ridicat de abstractizare
- TSK – sarcinile potrivite obiectivelor

În lucrarea de față, voi aplica acest concept pentru controlul și mentenanța sistemului de climatizare dintr-un Smart Building, accent asupra componentelor D_S , D_I și D_A , luând în considerare comunicația dintre acestea și interacțiunea lor cu mediul.

3. STUDIU DE CAZ: SISTEM DE CLIMATIZARE

3.1. Mediul de programare: JADE

JADE (Java Agent Development Framework) reprezintă un software implementat în limbajul de programare Java. Datorită specificațiilor FIPA (Foundation for Intelligent Physical Agents) și al unui set de instrumente care suporta atât procedurile de implementare, cât și cele de debugging, JADE constituie un limbaj eficient pentru dezvoltarea aplicațiilor de tip multi-agent (JADE, 2014).

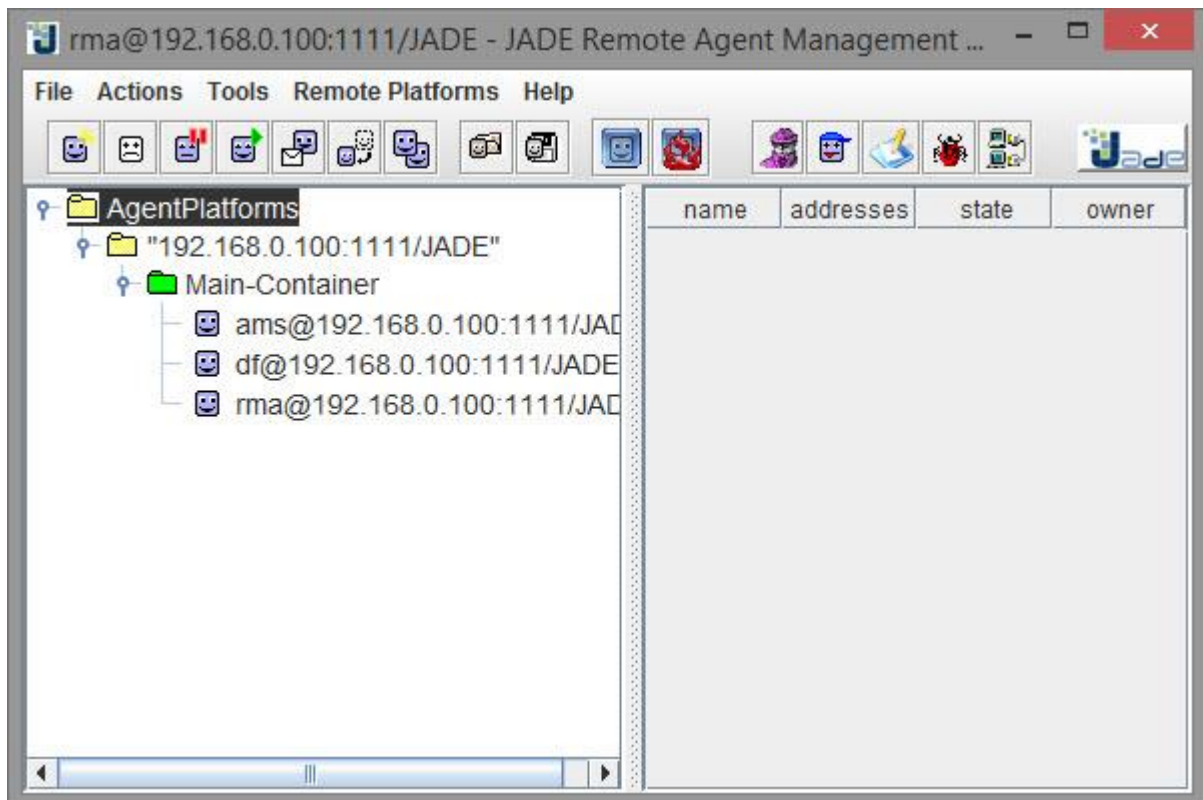


Figura 3.1.1. JADE – Agent Platform

Consola de management se poate porni din fereastra de comandă a sistemului de operare sau cu ajutorul unui program specializat în limbaj Java precum Eclipse sau NetBeans unde nucleele JADE-ului vor fi încărcate sub forma unor biblioteci de unde se vor importa ulterior pachetele necesare creării agenților doriți. În momentul în care platforma agent pornește, se vor crea automat trei agenți (JADE, 2014):

- AMS (Agent Management Service) – reprezintă expertul în cadrul platformei și este singurul agent care este capabil să pornească sau oprească alți agenți, precum și oprirea întregii platforme.
- DF (Directory Facilitator) – acționează ca o revistă de anunțuri unde agenții publică serviciile pe care le pot efectua, dar și găsi alți agenți care pot executa serviciile de care aceștia au nevoie.
- RMA (Remote Management Agent) – reprezintă însuși interfața JADE.

Agenții FIPA există fizic doar în cadrul platformei agent și utilizează toate resursele puse la dispoziție de aceasta pentru a-și atinge scopul. În acest context, agentul are un ciclu de viață ce trebuie gestionat de către platformă.

Proprietățile ciclului de viață:

- Un agent este fizic integrat în cadrul platformei, așadar ciclul său de viață este legat și depinde de platformă.
- Modelul ciclului de viață este independent de la o aplicație la alta și definește doar stările și tranzițiile prin care acesta trece.
- Agentul descris în modelul ciclului de viață este perceput ca o instanță. (are nume unic și se execută independent)
- Orice agent are un singur ciclu de viață la un moment dat într-o platformă.

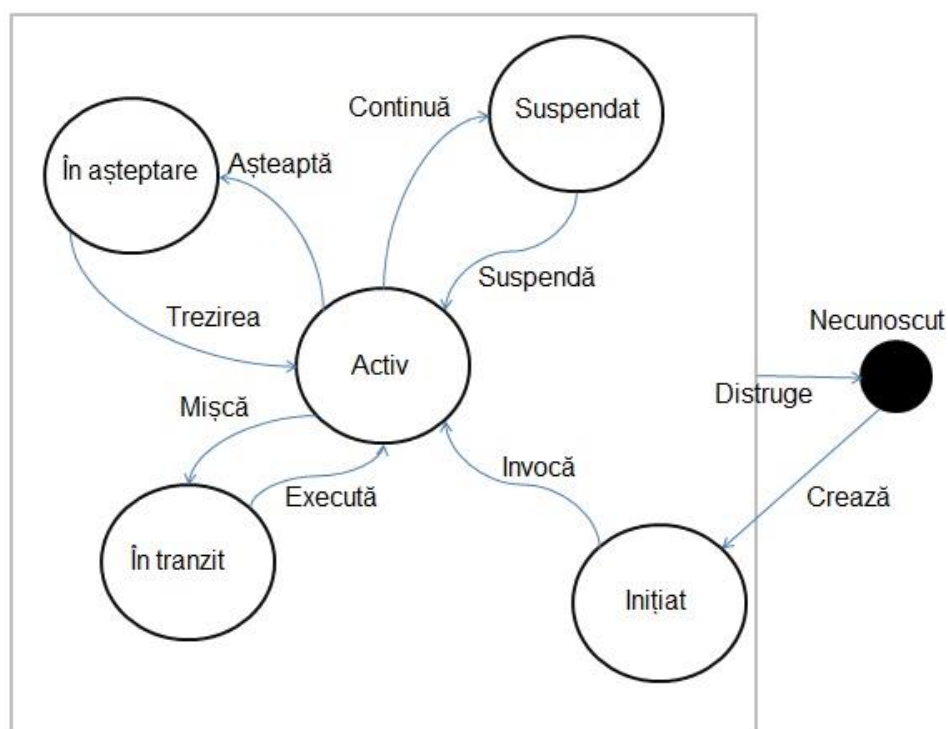


Figura 3.1.2. Ciclul de viață al agentului AMS (FIPA, 2014)

- Activ – serviciul de transport mesaje (MTS) transmite mesaje către agenți.
- Inițiat/Suspendat/În așteptare – MTS pune mesajele în așteptare până când agentul se întoarce în starea *Activ* sau trimite mesajele către alt agent dacă a fost setat.
- În tranzit – similar celor trei stări precedente cu mențiunea că doar agenții mobili pot intra în această stare.
- Necunoscut – MTS pun mesajele în coadă sau le respinge în funcție de cerințele de transport ale mesajului.
- Crează – crearea sau instalarea unui agent nou.
- Invocă – se invocă un agent nou.
- Distruge – distrugerea forțată a unui agent (aceasta nu poate fi inițiată decât de către AMS și nu poate fi ignorată de agent).
- Suspendă – suspendă un agent (se poate iniția de către AMS sau agent).
- Continuă – aduce un agent din starea de suspendat (executată doar de AMS).
- Așteaptă – pune un agent în așteptare (inițiată doar de agent).
- Trezire – trezește un agent din starea de așteptare.
- Mișcă – agentul intră într-o stare tranzitorie (inițiată doar de agent).
- Execută – agentul revine din starea tranzitorie (FIPA, 2014)

3.2. Sistemul de climatizare (Sistem HVAC)

Sistemul de climatizare reprezintă o tehnologie utilizată atât în interiorul unei case sau clădiri, cât și în cel al unei mașini pentru mărirea gradului de confort.

Sistemul este alcătuit din trei subsisteme: sistemul de încălzire care este reprezentat de încălzitor sau radiator, sistemul de răcire reprezentat de aerul condiționat și sistemul de ventilație reprezentat de ventilatoare.

Cu ajutorul sistemului HVAC se pot controla patru mărimi: temperatura, umiditatea, concentrația de dioxid de carbon din aerul respirat, precum și presiunea. Pentru reglarea mărimilor, cele trei subsisteme pot lucra atât concomitent cât și succesiv.

Din punctul de vedere al temperaturii, niciun mediu nu poate fi judecat asemenea de toată lumea, deoarece nivelul de confort diferă de la om la om, de la regiune la regiune sau de la țară la țară. Însă, constanța temperaturii este importantă pentru stabilirea nivelului de confort. Aceasta nu trebuie să varieze într-o singură zonă și să nu se schimbe brusc sau radical. (FHC, 2014)

Umiditatea reprezintă prezența vaporilor de apă în aer și afectează atât confortul cât și sănătatea oamenilor. Dacă umiditatea relativă scade sub 32%, anumiți oameni încep să simtă disconfort la nivelul ochilor, nasului, urechilor sau gâtului. Nu doar că acest lucru este iritant, el poate duce și la sângerări ale nasului sau inflamarea sinusurilor. Totodată, sub pragul de 30%, virușii se răspândesc mult mai ușor. Dacă umiditatea depășește pragul de 60%, clădirile sunt predispuse unor atacuri microbiologice precum mușgaiul sau igrasia care pot fi dăunătoare sănătății omului. Așadar, un interval bun de păstrare a umidității este de 40%-60% (Arundel et al., 1985).

Concentrația de dioxid de carbon din aer se măsoară în părți pe milimetru (ppm), iar conform ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers), concentrația optimă de dioxid de carbon în aerul dintr-o încăpere se află sub 1000 ppm, dar dacă nivelul depășește pragul de 600, aerul poate avea un miros urât (CDLC, 2014).

În cazul în care se dorește reglarea temperaturii, este suficient ca sistemul de încălzire să fie pornit în cazul în care se dorește o temperatură mai ridicată sau sistemul de răcire să fie pornit în cazul în care se dorește o temperatură mai joasă. Însă procesele de umidificare și dezumidificare a aerului reprezintă două procese mai complexe decât cele de creștere, respectiv scădere a temperaturii.

Atât procesul de umidificare a aerului, cât și cel de dezumidificare se poate realiza în două moduri: umidificare sau dezumidificare prin răcire sau încălzire. (Khemani, 2009)

Dezumidificarea prin răcire este cel mai cunoscut mod de eliminare a vaporilor de apă din aer. Acest proces se realizează prin răcirea bruscă în interiorul conductelor a aerului umed, iar apa rezultată este reținută în sistemul de climatizare într-un container special. Așadar, din moment ce aerul este sub temperatura sa de condens, se va produce și o ușoară răcire a aerului.

Dezumidificarea prin încălzire se face cu ajutorul unor substanțe chimice precum oxidul de aluminiu, care emană căldură constant și are tendința de a absorbi umezeala. De aceea, aceste substanțe chimice se mai numesc și substanțe higroscopice. Prin urmare, la finalul procesului, aerul dezumidificat va avea o temperatură mai ridicată.

Umidificarea prin încălzire se face prin trecerea aerului uscat pe deasupra unui jet de apă care are o temperatură mai ridicată decât cea a aerului. Astfel, vaporii de apă sunt preluați de aerul uscat. Umidificare prin încălzire se face și prin amestecarea aerului uscat cu abur. Datorită faptului că temperatura apei este mai ridicată decât cea a aerului, se va produce o ușoară încălzire a aerului umidificat.

Umidificarea prin răcire se face prin trecerea aerului uscat pe deasupra unui jet de apă care are o temperatură mai joasă decât cea a aerului. În momentul în care aerul trece, particulele de apă din jet se evaporă și ajută la creșterea umidității aerului. Datorită faptului că temperatura apei este mai joasă decât cea a aerului, se va produce o ușoară răcire a aerului umidificat.

În practică, reglarea mărimilor cu ajutorul sistemului de climatizare se poate face atât manual cât și automat cu ajutorul senzorilor și a reglatoarelor.

Fiecare tip de senzor are un tip de semnal la ieșire:

- Senzorii analogici sunt folosiți pentru monitorizarea continuă a schimbărilor, iar ca ieșire pot trimite un semnal între 0V – 10V sau 4mA – 20mA
- Senzorii digitali indică poziția de deschis (1) sau închis (0)

Cu ajutorul sistemelor multi-agent am simulat controlul și mentenanța unui sistem de climatizare într-o clădire inteligentă.

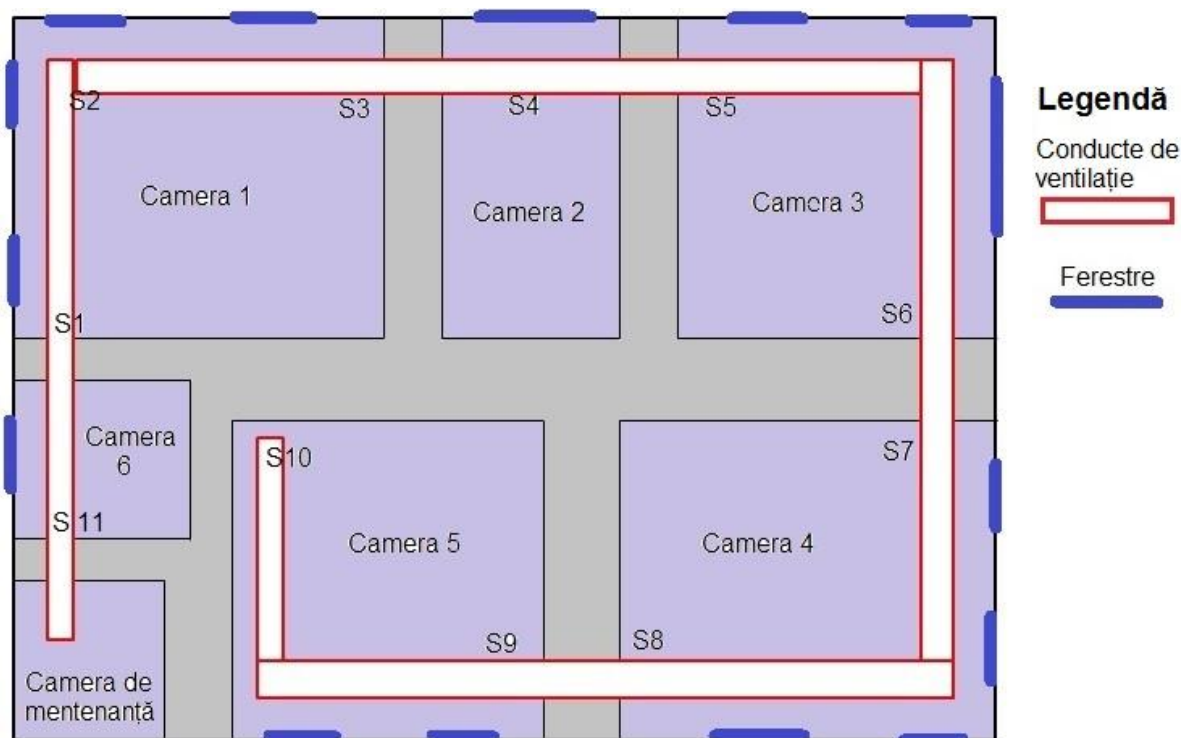


Figura 3.2.1. Exemplu organizare sistem HVAC pe un etaj

Pentru a demonstra funcționalitatea sistemului am propus un etaj cu șase camere în care se dorește menținerea temperaturii între 23 și 29 de grade Celsius, a umidității între 40 și 60%, și a nivelului de dioxid de carbon din aer sub 600 ppm.

Dacă există vreun sensor defect de viteză a aerului sau nu se află între valorile normale, atunci sistemul de climatizare se închide de urgență și se solicită verificarea conductelor de aer și/sau a senzorilor.

Dacă există vreo fereastră spartă în cameră, atunci sistemul de climatizare va rula în modul *standby* (ventilația pornită, fără controlul mărimilor pentru a nu face risipă de energie).

În cazul în care este vorba despre o cameră mică precum camera numărul șase care conține câte un singur senzor analog, atunci controlul este relativ simplu. În schimb, cu cât numărul de senzori analogici crește, cu atât lucrurile se complică, lucru ce va fi detaliat în secțiunea 3.3.3.

În funcție de dimensiunea camerei, numărul senzorilor poate varia, însă aceștia se presupun a fi plasați echidistant și de o manieră firească (se exclud cazurile în care un senzor de temperatură este lângă un calorifer, iar altul lângă un geam).

Fiecare S1÷ S11 reprezintă un grup de patru senzori analogici și sunt produși de către Carlo Gavazzi (Gavazzi, 2014):

- Senzor pentru temperatură și umiditate
 - Senzor pentru viteză a aerului
 - Senzor pentru concentrația de dioxid de carbon din aer
- iar fiecare fereastră are asociat un senzor pentru detecția geamurilor sparte.

Senzorul de temperatură și umiditate

- Senzorul de temperatură funcționează între -5 grade Celsius și 50 grade Celsius.
- Senzorul de umiditate funcționează între 10% RH și 95% RH.
- Valoarea ieșirii se află în intervalul 0V – 10V.

-

Senzorul de viteză a aerului

- Funcționează între 0 m/s – 20 m/s.
- Valoarea ieșirii se află în intervalul 0V – 10V.

Senzorul de dioxid de carbon

- Funcționează între 0 ppm – 2000 ppm.
- Valoarea ieșirii se află în intervalul 0V – 10V.

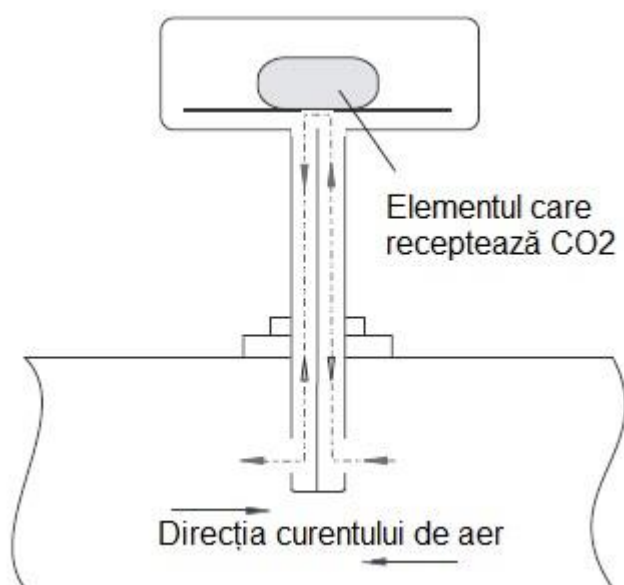


Figura 3.2.2. Principiul de funcționare al senzorului de CO2 (Gavazzi, 2014)

Senzorul pentru detecția geamurilor sparte

- Senzorul detectează un semnal audio care corespunde frecvenței de sparge a ferestrelor (SGD, 2014).
- Acesta returnează la ieșire 0 (geam intact) sau 1 (geam spart)

3.3. Modul de implementare

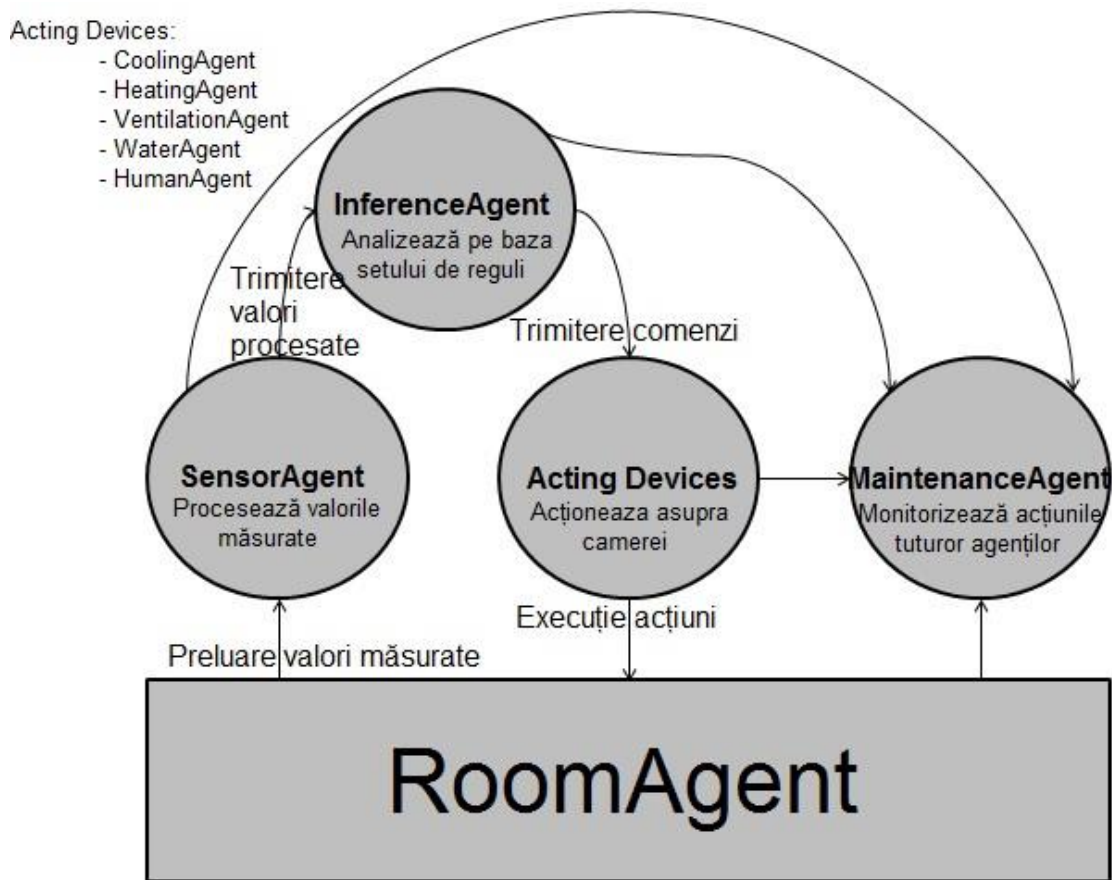


Figura 3.3.1. Sistemul multi-agent propus

Agentul cameră simulează comportamentul camerei. Inițial se introduc valori de la tastatură pentru numărul de senzori din cameră și valorile acestora. Valorile vor fi transmise ulterior agentului senzor care le prelucrează și trimite agentului senzor. Agentul senzor trimite agentului de inferență noile valori calculate unde vor fi analizate pe baza unui set de reguli. După ce se analizează regulile, acest agent ia deciziile corespunzătoare pentru a controla mărimile camerei, dacă acestea se pot controla, sau alertează echipa de mentenanță (HumanAgent) și camera de mentenanță (MaintenanceAgent) că variabilele nu au putut fi controlate și de ce. În urma cererii intervenției echipelor de mentenanță, se va răspunde sistemului cu deplasarea acestora în clădire.

Transmisia între agenți se face cu ajutorul agentului AMS. Fiecare agent în parte apelează rutina *AMSService.search* care presupune căutarea printre toți agenții activi în cadrul platformei. În momentul în care unui agent i se comunică faptul că un alt agent cu care dorește să comunice este online, atunci transmisia se poate face. Așadar, fiecare agent în parte trebuie să știe numele agentului căruia dorește să îi transmită informații.

În Figura 3.3.2. se poate observa că toți agenții sunt online, iar numele fiecăruia este:

- CoolingAgent are numele cool;
- HeatingAgent are numele heat;

- VentilationAgent are numele vent;
- MaintenanceAgent are numele cont;
- InferenceAgent are numele ia;
- WaterAgent are numele water;
- HumanAgent are numele team;
- SensorAgent are numele sens;
- Numele lui RoomAgent variază în funcție de numărul camerei (room1 ÷ room6);

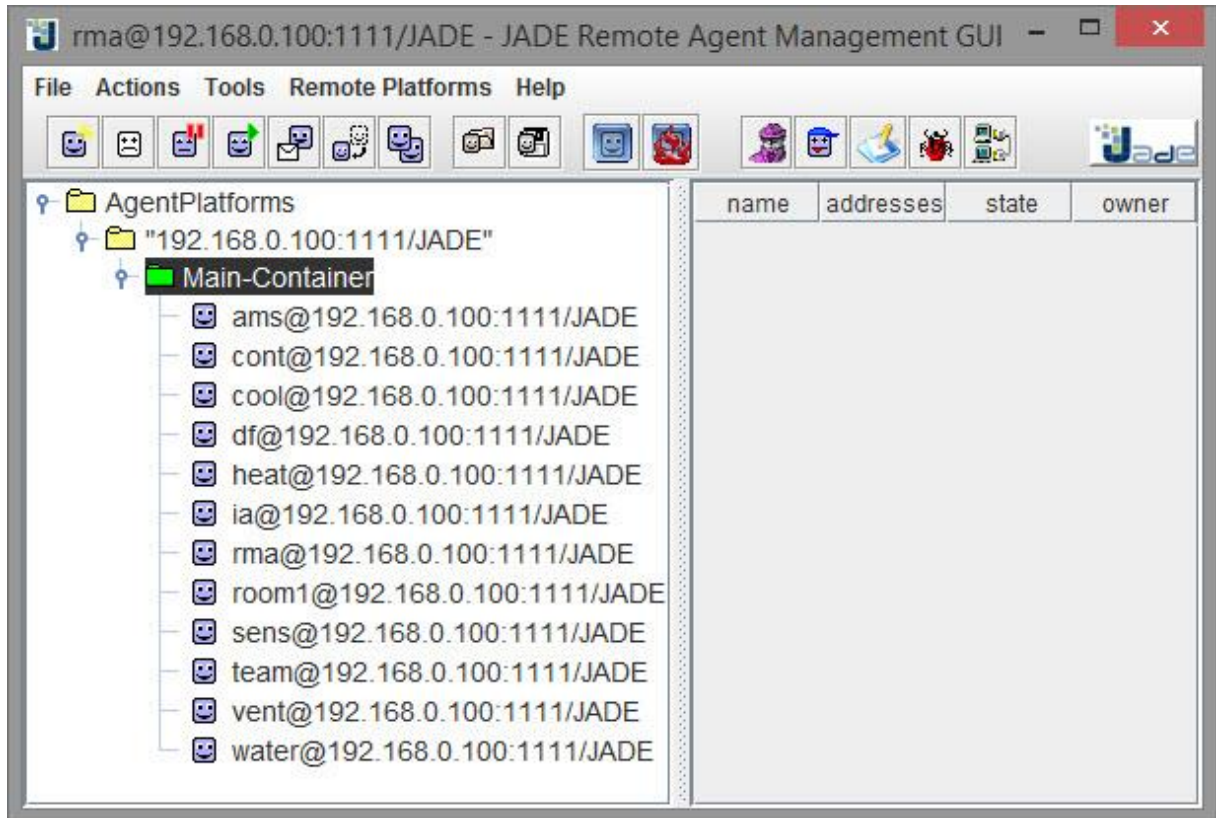


Figura 3.3.2. Platforma

3.3.1. Agentul care simulează comportamentul camerei

Dacă după prima analiză a mărimilor agentul de inferență nu detectează nicio eroare, atunci controlul are loc prin transmiterea anumitor mesaje către agenții de execuție prin care li se va spune cum să acționeze asupra camerei și să o afecteze. La baza controlului mărimilor din cameră stă un set de reguli bazat pe mesajele primite de la agenții de execuție cu mențiunea că:

- Procesul de umidificare a aerului s-a făcut prin încălzire, iar cel de deumidificare s-a făcut prin răcire.
- La fiecare iterație pentru controlul dioxidului de carbon, se va scădea concentrația cu 50ppm.
- La fiecare iterație pentru controlul temperaturii se va crește sau scădea temperatura cu 0.5 grade Celsius.
- La fiecare iterație pentru controlul umidității se va crește sau scădea umiditatea cu 1% și temperatura va scădea sau crește cu 0.1 grade Celsius.
- Subsistemul apă ajută strict pentru scăderea, respectiv creșterea umidității.

Ventilație	Încălzire	Răcire	Apă	Concentrația CO2	Temperatura	Umiditatea
OFF	OFF	OFF	OFF	-	-	-
idle	OFF	OFF	OFF	-	-	-
idle	ON	OFF	OFF	-	+0.5	-
idle	OFF	ON	OFF	-	-0.5	-
ON	ON	OFF	OFF	-50	+0.5	-
ON	OFF	ON	OFF	-50	-0.5	-
ON	OFF	OFF	OFF	-50	-	-
ONd	OFF	ONd	ONd	-	-0.1	-1
ONd	OFF	ONtd	ONd	-	-0.6	-1
ONd	ON	ONd	ONd	-	+0.4	-1
ONcd	OFF	ONd	ONd	-50	-0.1	-1
ONcd	OFF	ONtd	ONd	-50	-0.6	-1
ONcd	ON	ONd	ONd	-50	+0.4	-1
ONh	ONh	ON	ONh	-	-0.4	+1
ONh	ONh	OFF	ONh	-	+0.1	+1
ONh	ONth	OFF	ONh	-	+0.6	+1
ONch	ONh	ON	ONh	-50	-0.4	+1
ONch	ONh	OFF	ONh	-50	+0.1	+1
ONch	ONth	OFF	ONh	-50	+0.6	+1

Notății:

- Idle – modul *standby* pentru ventilator;
- ON – pornit;
- ONh – pornit pentru umidificare;
- ONd – pornit pentru deumidificare;
- ONth – pornit pentru umidificare și încălzire;
- ONtd – pornit pentru deumidificare și răcire;
- ONcd – pornit pentru deumidificare și reglare concentrație de CO2;
- ONch – pornit pentru umidificare și reglare concentrație de CO2;
- OFF – oprit;

Dacă nu există decât un singur senzor de fiecare tip, operațiile de adunare sau scădere se aplică asupra valorii mărimii după care este iar transmisă la senzor și bucla se reia până când se ajunge la regula idle-OFF-OFF-OFF, aceasta fiind regula care returnează mărimea optimă.

Dacă există doi sau mai mulți senzori de același tip, atunci se face media lor și dacă se execută o operație de scădere, atunci senzorii care au valoarea peste medie se modifică primii, iar în cazul în care se execută o operație de adunare, atunci senzorii cu valoarea mai mică decât media se vor actualiza primii. În acest mod, se evită obținerea unor diferențe prea mari între senzori, caz în care la iterația următoare, agentul de inferență ar sesiza asta.

3.3.2. Implementarea agentului care simulează senzorii

Pentru simularea senzorilor a fost creat un singur agent denumit SensorAgent. Acest agent este în strânsă legătură cu RoomAgent și InferenceAgent dar și cu agenții de tip „acting devices”, deoarece în momentul în care preia valorile măsurate din cameră, preia și numărul acestora. Așadar, agenții care acționează precum elementele de execuție, vor ști asupra cărei camere se acționează.

La citirea valorilor măsurate din cameră (se dau de la tastatură la pornirea agentului ce simulează comportamentul camerei), agentul senzor va scala fiecare valoare conform specificațiilor senzorilor din capitolul 3.2. și se va trimite valoarea în tensiune agentului de inferență care rulează mai multe seturi de reguli ce vor fi explicate în secțiunea următoare.

Dacă ne aflăm în cazul camerei 6, atunci se va introduce câte un singur senzor din fiecare, valorile de intrare vor fi:

```
Insert number of CO2 sensors from room: 1
CO2 sensors range is 0 to 2000 ppm!
Insert CO2 sensor 1 value: 300
Insert number of temperature sensors from room: 1
Temperature sensors range is -5 to 55 Celsius!
Insert temperature sensor 1 value: 20
Insert number of humidity sensors from room: 1
Relative humidity sensors range is 10 to 95%!
Insert humidity sensor 1 value: 32
Insert number of air velocity sensors from room: 1
Air velocity sensors range is 0 to 20 m/s!
Insert air velocity sensor 1 value: 10
Insert number of broken glass detectors from room: 1
Broken glass detectors values: 1 - glass is broken or 0 - glass is OK
Insert broken glass detector 1 value: 0
```

Aplicația returnează	Interpretare
Primit de senzor: C0 300.0 Primit de senzor: T0 20.0 Primit de senzor: H0 32.0 Primit de senzor: AV0 10.0 Primit de senzor: M0 0.0	SensorAgent preia valoarea concentrației de CO2, a temperaturii, a umidității, a vitezei aerului din conducte și starea ferestrei din cameră.
Trimis de senzor: C0 1.5 Trimis de senzor: T0 4.5454545 Trimis de senzor: H0 2.5882354 Trimis de senzor: A0 5.0 Trimis de senzor: M0 0.0	SensorAgent trimite valorile agentului de inferență după ce au fost procesate și scalate pentru a putea reprezenta valorile în tensiune.

În cazul în care este necesară folosirea mai multor senzori, cum este cazul camerei 1, atunci transmisia mesajelor se va face secvențial. De asemenea, trebuie menționat faptul că aplicația funcționează pentru orice număr de senzori de orice tip.

Pentru valorile de intrare:

```

Insert number of CO2 sensors from room: 3
CO2 sensors range is 0 to 2000 ppm!
Insert CO2 sensor 1 value: 500
Insert CO2 sensor 2 value: 530
Insert CO2 sensor 3 value: 560
Insert number of temperature sensors from room: 3
Temperature sensors range is -5 to 55 Celsius!
Insert temperature sensor 1 value: 16
Insert temperature sensor 2 value: 17
Insert temperature sensor 3 value: 15
Insert number of humidity sensors from room: 3
Relative humidity sensors range is 10 to 95%!
Insert humidity sensor 1 value: 51
Insert humidity sensor 2 value: 54
Insert humidity sensor 3 value: 49
Insert number of air velocity sensors from room: 3
Air velocity sensors range is 0 to 20 m/s!
Insert air velocity sensor 1 value: 10
Insert air velocity sensor 2 value: 11
Insert air velocity sensor 3 value: 9
Insert number of broken glass detectors from room: 4
Broken glass detectors values: 1 - glass is broken or 0 - glass is OK
Insert broken glass detector 1 value: 1
Insert broken glass detector 2 value: 0
Insert broken glass detector 3 value: 1
Insert broken glass detector 4 value: 0

```

Aplicația returnează	Interpretare
Primit de senzor: C0 500.0 Primit de senzor: C1 530.0 Primit de senzor: C2 560.0 Primit de senzor: T0 16.0 Primit de senzor: T1 17.0 Primit de senzor: T2 15.0 Primit de senzor: H0 51.0 Primit de senzor: H1 54.0 Primit de senzor: H2 49.0 Primit de senzor: AV0 10.0 Primit de senzor: AV1 11.0 Primit de senzor: AV2 9.0 Primit de senzor: M0 1.0 Primit de senzor: M1 0.0 Primit de senzor: M2 1.0 Primit de senzor: M3 0.0	SensorAgent preia valorile concentrației de CO2, ale temperaturii, ale umidității, ale vitezei aerului din conducte și stările ferestrelor din cameră rând pe rând. După ce s-au primit toate valorile trimise de un senzor, se trece la următorul tip de senzor până când toate datele au fost citite.
Trimis de senzor: C0 2.5 Trimis de senzor: C1 2.65 Trimis de senzor: C2 2.8 Trimis de senzor: T0 3.8181818 Trimis de senzor: T1 4.0 Trimis de senzor: T2 3.6363637 Trimis de senzor: H0 4.8235292 Trimis de senzor: H1 5.1764708 Trimis de senzor: H2 4.5882354 Trimis de senzor: A0 5.0 Trimis de senzor: A1 5.5 Trimis de senzor: A2 4.5 Trimis de senzor: M0 1.0 Trimis de senzor: M1 0.0 Trimis de senzor: M2 1.0 Trimis de senzor: M3 0.0	SensorAgent trimite secvențial valorile agentului de inferență după ce acestea au fost procesate și scalate pentru a putea reprezenta valorile în tensiune.

3.3.3. Implementarea setului de reguli în agentul de inferență

Velocitatea aerului este cea mai importantă mărime din sistem, deoarece de aceasta depinde funcționarea corectă a conductelor de aer. În cazul în care există defecțiuni la nivelul conductei sau la nivelul senzorilor de aer și nu se intervine la timp, consecințele pot fi uriașe. De aceea acesta se verifică prima în agentul de inferență și în cazul în care sunt probleme legate de aceasta, întregul sistem de climatizare se va opri din funcționare, iar personalul calificat înștiințat de eroare.

Ca importanță urmează senzorul pentru detecția ferestrelor care nu poate fi controlat în mod manual ci doar analizat și interpretat. Dacă unul din senzorii digitali returnează 1 (geam spart), atunci agentul de inferență semnalizează care senzor a declanșat alarma și în ce cameră, pentru ca personalul calificat să poată remedia problema în cel mai scurt timp cu putință.

Până la remedierea problemei, sistemul de climatizare rulează în modul standby (ventilație la nivel scăzut sau idle), dacă și numai dacă nu s-a detectat și o problemă în legătură cu conductele de aer sau senzorii asociați. În acest caz, sistemul se va opri complet datorită priorității.

Dacă în cazul în care niciunul din cele două cazuri nu au apărut, atunci se verifică și ceilalți senzori. Dacă există doar câte unul din fiecare, atunci se realizează controlul.

Dacă există doi sau mai mulți senzori analogici de același tip, se va calcula diferența dintre senzorii receptați de SensorAgent în prealabil. Dacă diferența se află în limitele impuse, atunci se poate continua cu controlul mărimilor, altfel se afișează eroarea întâlnită și sistemul de climatizare intră în modul standby sau se oprește.

- Diferența maximă acceptată dintre doi senzori de viteză a aerului este de 5 m/s. Cazul în care această diferență este mai mare, se afișează senzorii cu probleme și se oprește sistemul HVAC. Personalul calificat are obligația de a verifica atât senzorii pentru înlăturarea senzorului defect, cât și conductele de aer pentru a nu predispuce clădirea și oamenii în pericol.
- Atât senzorii de umiditate, cât și cei de temperatură și pentru detecția concentrației de dioxid de carbon se presupun a fi instalați într-o manieră cât mai eficientă și se dorește evitarea cazului în care un senzor de temperatură este lângă geam, iar altul lângă o sursă de căldură.
- Diferența maximă acceptată dintre doi senzori de umiditate relativă este de: 6%.
- Diferența maximă acceptată dintre doi senzori de temperatură este de: 3 grade Celsius.
- Diferența maximă acceptată dintre doi senzori pentru detecția concentrației de dioxid de carbon din aer este de: 75 ppm.

Pentru a evita situația neplăcută în care se returnează o alarmă fals pozitivă, adică se face controlul pentru mai multe mărimi, acestea sunt în regulă la prima verificare, iar la una din următoarele iterații nu mai sunt deoarece a început procesul de control, agentul care simulează camera va micșora valoarea celui mai mare senzor dacă este necesară scăderea concentrației de dioxid de carbon din aer, micșorarea temperaturii sau dezumidificarea; iar cazul în care este necesară umidificarea sau creșterea temperaturii, senzorii cu valorile cele mai

mici vor fi primii actualizați. În acest mod controlul se realizează într-un mod eficient și elimină eventualele alarme de tipul fals pozitiv.

3.3.4. Implementarea agenților de execuție

În vederea realizării controlului și mentenanței într-o clădire inteligentă, cu ajutorul sistemelor multi-agent am propus cinci agenți de execuție, patru automați care se ocupă strict de controlul în clădire, iar unul uman care se va ocupa de mentenanță.

Agenții automați sunt VentilationAgent, CoolingAgent, HeatingAgent și WaterAgent și comunică în mod direct cu agentul care a simulat camera, pe baza deciziilor luate de către agentul de inferență. Cei patru agenți își modifică starea, dacă este nevoie și comunică agentului cameră în ce stare au intrat pentru ca acesta să fie ulterior capabil să returneze valorile optime ale sistemului.

Al cincelea agent reprezintă personalul clădirii care se ocupă cu mentenanța în clădire, așadar acesta nu primește niciun mesaj până când o eroare nu a fost detectată.

3.3.5. Implementarea agentului de mentenanță

Maintenance.Agent poate fi văzut drept un agent nucleu deoarece acesta are legături directe cu fiecare agent în parte, fiind informat de situația fiecăruia.

- RoomAgent are datoria de a înștiința camera de control momentul în care măsurile au fost controlate și au ajuns la valorile optime.
- SensorAgent are datoria de a comunica numărul camerei din care au fost citiți senzorii.
- După depistarea unei defecțiuni, atât camera de control cât și personalul de mentenanță sunt informați pentru a fi la curent cu problema ivită și pentru a o rezolva. Totodată, MaintenanceAgent este informat de valorile senzorilor din camera.
- În momentul în care se depistează o eroare, agentul uman confirmă camerei de control că a recepționat-o și este în curs de remediere.

3.4. Scenarii de simulare

3.4.1. Controlul mărimilor când există câte un senzor de același tip

Valorile măsurate de senzori se introduc de la tastatură. Pentru a se putea realiza controlul în cazul în care există un singur senzor pe tip, aceștia trebuie să se afle în limitele cerute, altfel controlul nu va putea fi realizat.

```
Insert number of CO2 sensors from room: 1
CO2 sensors range is 0 to 2000 ppm!
Insert CO2 sensor 1 value: 700
Insert number of temperature sensors from room: 1
Temperature sensors range is -5 to 55 Celsius!
Insert temperature sensor 1 value: 14
Insert number of humidity sensors from room: 1
Relative humidity sensors range is 10 to 95%!
Insert humidity sensor 1 value: 64
Insert number of air velocity sensors from room: 1
Air velocity sensors range is 0 to 20 m/s!
Insert air velocity sensor 1 value: 10
Insert number of broken glass detectors from room: 1
Broken glass detectors values: 1 - glass is broken or 0 - glass is
OK
Insert broken glass detector 1 value: 0
```

După ce agentul de inferență a trecut o data prin setul de reguli, determină că umiditatea și concentrația de CO₂ sunt peste valorile admise, iar temperatura este sub valoarea optimă, așadar începe controlul.

Aplicația returnează	Semnificația
22:21:02.179: Watering system is working to decrease humidity in room two! ← 1 iterație 22:21:02.182: Ventilation system is working at a higher capacity to decrease CO2 concentration and to decrease humidity in room two! 22:21:02.184: Cooling system is at minimum power in room two to decrease humidity! Temperature is too low and humidity is too high! 22:21:02.184: Heating system is increasing temperature in room two! Temperature is too low and humidity is too high! 22:21:02.251: Watering system is working to decrease humidity in room two! ← 2 iterații 22:21:02.253: Heating system is increasing temperature in room two! Temperature is too low and humidity is too high! 22:21:02.256: Cooling system is at minimum power in room two to decrease humidity! Temperature is too low and humidity is too high! 22:21:02.256: Ventilation system is working at a higher capacity to decrease CO2 concentration and to decrease humidity in room two! 22:21:02.329: Watering system is working to decrease humidity in room two! ← 3 iterații 22:21:02.332: Heating system is increasing temperature in room two! Temperature is too low and humidity is too high! 22:21:02.333: Ventilation system is working at a higher capacity to decrease CO2 concentration and to decrease humidity in room two! 22:21:02.334: Cooling system is at minimum power in room two to decrease humidity! Temperature is too low and humidity is too high!	Valoarea optimă pentru concentrația de CO ₂ este mai mica decât 600 ppm, iar la fiecare iterație de control al acesteia, concentrația scade cu 50 ppm. Așadar, în cazul prezentat, valoarea optimă se va atinge după trei iterații, iar ventilatorul va funcționa la o capacitate redusă până ce umiditatea va avea la rândul ei, o valoare optimă.

Aplicația returnează	Semnificația
22:21:02.405: Watering system is working to decrease humidity in room two! ← 4 iterații 22:21:02.408: Heating system is increasing temperature in room two! Temperature is too low and humidity is too high! 22:21:02.409: Cooling system is at minimum power in room two to decrease humidity! Temperature is too low and humidity is too high! 22:21:02.410: Ventilation system is decreasing humidity in room two! 22:21:02.474: Watering system is working to decrease humidity in room two! ← 5 iterații 22:21:02.477: Ventilation system is decreasing humidity in room two! 22:21:02.478: Cooling system is at minimum power in room two to decrease humidity! Temperature is too low and humidity is too high! 22:21:02.480: Heating system is increasing temperature in room two! Temperature is too low and humidity is too high!	Având în vedere faptul că valoarea umidității trebuie sa fie mai mica decât 60, această mărimeși va atinge valoarea optimă după încă două iterații, ieșirea sa fiind 59
22:21:02.542: Watering system is off in room two! 22:21:02.545: Ventilation system is working on standby mode in room two! ← 6 iterații 22:21:02.545: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:02.547: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:02.617: Watering system is off in room two! 22:21:02.619: Ventilation system is working on standby mode in room two! ← 7 iterații 22:21:02.619: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:02.620: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:02.686: Watering system is off in room two! 22:21:02.691: Ventilation system is working on standby mode in room two! ← 8 iterații 22:21:02.692: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:02.692: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:02.763: Watering system is off in room two! 22:21:02.766: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:02.767: Ventilation system is working on standby mode in room two! ← 9 iterații 22:21:02.768: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:02.832: Watering system is off in room two! 22:21:02.835: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:02.836: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:02.837: Ventilation system is working on standby mode in room two! ← 10 iterații 22:21:02.914: Watering system is off in room two! 22:21:02.917: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:02.917: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:02.919: Ventilation system is working on standby mode in room two! ← 11 iterații 22:21:02.988: Watering system is off in room two! 22:21:02.990: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:02.992: Ventilation system is working on standby mode in room two! ← 12 iterații 22:21:02.992: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:03.060: Watering system is off in room two! 22:21:03.063: Heating system is increasing temperature in	Dacă în cazul reglării concentrației de dioxid de carbon și a umidității a fost ușor de urmărit modalitatea în care se face controlul, în cazul temperaturii nu se mai poate proceda asemănător deoarece dezumidificarea se face cu ajutorul răcirii, așadar temperatura va fi influențată de scăderea umidității. Dar din moment ce umiditatea s-a realizat după 5 iterații, înseamnă ca temperatura a scăzut cu 0.5 grade în tot acest timp. Deci putem pleca de la premisa că în cameră au fost detectate 13.5 grade, și nu 14. Având în vedere că o iterație va crește temperatura cu 0.5 grade, pentru realizarea controlului vor fi necesare 19 iterații

Aplicația returnează	Semnificația
room two! Temperature is too low and humidity is ok 22:21:03.065: Ventilation system is working on standby mode in room two! ← 13 iterații 22:21:03.066: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:03.136: Watering system is off in room two! 22:21:03.138: Ventilation system is working on standby mode in room two! ← 14 iterații 22:21:03.140: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:03.141: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:03.209: Watering system is off in room two! 22:21:03.212: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:03.214: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:03.214: Ventilation system is working on standby mode in room two! ← 15 iterații 22:21:03.285: Watering system is off in room two! 22:21:03.288: Ventilation system is working on standby mode in room two! ← 16 iterații 22:21:03.288: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:03.289: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:03.355: Watering system is off in room two! 22:21:03.358: Ventilation system is working on standby mode in room two! ← 17 iterații 22:21:03.359: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:03.360: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:03.428: Watering system is off in room two! 22:21:03.431: Cooling system is off in room two! Temperature is too low and humidity is ok! 22:21:03.433: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:03.434: Ventilation system is working on standby mode in room two! ← 18 iterații 22:21:03.497: Watering system is off in room two! 22:21:03.500: Ventilation system is working on standby mode in room two! ← 19 iterații 22:21:03.501: Heating system is increasing temperature in room two! Temperature is too low and humidity is ok 22:21:03.501: Cooling system is off in room two! Temperature is too low and humidity is ok!	

După cum se poate observa, după cea de-a 19-a iterație, sistemul își va fi atins valorile optime.

```

22:21:03.566: Ventilation system is working on standby mode in room
two! CO2 concentration and humidity are ok!
22:21:03.566: Heating system is off in room two! Temperature and
humidity are ok!
22:21:03.566: Cooling system is off in room two! Temperature and
humidity are ok!
22:21:03.591: System's parameters in room two have reached their
optimal values!
22:21:03.591: CO2 sensor 1 value is: 550.0 ppm.
22:21:03.591: T sensor 1 value is: 22.999998 Celsius degrees.
22:21:03.591: H sensor 1 value is: 59.0 % relative humidity.

```

3.4.2. Controlul mărimilor pentru mai mult de doi senzori de același tip

În ceea ce privește urmărirea controlului mărimilor pentru doi sau mai mulți senzori de același timp, este dificilă deoarece controlul se face în funcție de media senzorilor, dacă și numai dacă diferență maximă dintre ei nu depășește limitele impuse. Pentru a simula un astfel de control, au fost introduse următoarele valori. Iar pe baza log-urilor voi explica modul în care s-a realizat controlul concentrației de dioxid de carbon și al umidității.

```
Insert number of CO2 sensors from room: 3
CO2 sensors range is 0 to 2000 ppm!
Insert CO2 sensor 1 value: 720
Insert CO2 sensor 2 value: 750
Insert CO2 sensor 3 value: 735
Insert number of temperature sensors from room: 3
Temperature sensors range is -5 to 55 Celsius!
Insert temperature sensor 1 value: 31
Insert temperature sensor 2 value: 32
Insert temperature sensor 3 value: 31
Insert number of humidity sensors from room: 3
Relative humidity sensors range is 10 to 95%!
Insert humidity sensor 1 value: 26
Insert humidity sensor 2 value: 28
Insert humidity sensor 3 value: 27
Insert number of air velocity sensors from room: 3
Air velocity sensors range is 0 to 20 m/s!
Insert air velocity sensor 1 value: 8
Insert air velocity sensor 2 value: 11
Insert air velocity sensor 3 value: 9
Insert number of broken glass detectors from room: 4
Broken glass detectors values: 1 - glass is broken or 0 - glass is
OK
Insert broken glass detector 1 value: 0
Insert broken glass detector 2 value: 0
Insert broken glass detector 3 value: 0
Insert broken glass detector 4 value: 0
```

După cum se poate observa, au fost introduși câte trei senzori pentru detecția concentrației de dioxid de carbon și pentru umiditate.

După ce agentul senzor citește valorile din cameră, acestea sunt trimise agentului de inferență unde vor fi verificate valorile senzorilor (diferența dintre oricare doi senzori să nu depășească un anumit prag). După cum am menționat și în secțiunea 3.3.3., diferența maximă admisibilă dintre doi senzori pentru CO₂ este de 75 ppm, iar cea dintre doi senzori de umiditate este de 6. Se poate observa foarte ușor că aceste condiții se respectă, deci senzorii sunt funcționali.

Odată ce agentul de inferență a stabilit starea senzorilor (defecți sau nu), acesta va calcula media senzorilor de același tip, și în funcție de rezultat se va transmite un mesaj agenților efectuatori.

În cazul prezentat, media senzorilor de CO₂ va fi de 735, iar cea a senzorilor de umiditate va fi de 27. Prin urmare, sistemul va trebui să crească umiditatea din camera respectivă, și să scadă concentrația de dioxid de carbon. Odată ce agenții de acționare încep să execute comanda dată de agentul de inferență, nivelul de CO₂ va scădea în cameră, iar umiditatea va crește.

După cum am menționat anterior, scăderea concentrației de CO₂ din aer se face progresiv, cu -50 ppm pe iterație, iar creșterea umidității cu 1 procent pe iterație. Senzorii din cameră pot recepta diferite valori în funcție de poziționarea acestora, de aceea am decis ca odată ce procesul de control începe senzorii sunt afectați pe rand. De exemplu, dacă senzorul x de temperatură afișează 30 grade iar senzorul y afișează 31 grade și procesul de răcire începe, senzorii nu pot fi afectați simultan decât dacă scăderea de temperatură se face brusc, ceea ce nu este cazul, deci senzorul y va detecta primul o scădere a temperaturii. Asemeni exemplului, sunt afectați și senzorii din sistemul multi-agent implementat. Mai întâi scade senzorul 2 de CO₂, apoi 3, s.a.m.d.

De asemenea, folosind acest algoritm, se evită și transmiterea unor valori greșite. De exemplu, dacă diferența dintre primul senzor și al doilea era de cel puțin 40 și se analiza mai întâi primul senzor, atunci agentul de inferență ar fi detectat o diferență mai mare de 75, însemnând că nu se poate realiza controlul, iar sistemul de climatizare ar fi intrat în avarie.

```
17:12:16.598: Watering system is working to increase humidity in room one!
17:12:16.600: Ventilation system is working at a higher capacity to decrease CO2
concentration and to increase humidify the air in room one!
17:12:16.602: Heating system is at minimum power to increase humidity in room one!
Temperature is too high and humidity is too low!
17:12:16.603: Cooling system is decreasing temperature in room one! Temperature is
too high and humidity is too low!
17:12:16.734: Watering system is working to increase humidity in room one!
17:12:16.739: Heating system is at minimum power to increase humidity in room one!
Temperature is too high and humidity is too low!
17:12:16.740: Ventilation system is working at a higher capacity to decrease CO2
concentration and to increase humidify the air in room one!
17:12:16.740: Cooling system is decreasing temperature in room one! Temperature is
too high and humidity is too low!
17:12:16.866: Watering system is working to increase humidity in room one!
17:12:16.869: Ventilation system is working at a higher capacity to decrease CO2
concentration and to increase humidify the air in room one!
17:12:16.871: Cooling system is decreasing temperature in room one! Temperature is
too high and humidity is too low!
17:12:16.871: Heating system is at minimum power to increase humidity in room one!
Temperature is too high and humidity is too low!
17:12:16.993: Watering system is working to increase humidity in room one!
17:12:16.996: Heating system is at minimum power to increase humidity in room one!
Temperature is too high and humidity is too low!
17:12:16.997: Cooling system is decreasing temperature in room one! Temperature is
too high and humidity is too low!
17:12:16.997: Ventilation system is working at a higher capacity to decrease CO2
concentration and to increase humidify the air in room one!
17:12:17.120: Watering system is working to increase humidity in room one!
17:12:17.124: Heating system is at minimum power to increase humidity in room one!
Temperature is too high and humidity is too low!
17:12:17.124: Cooling system is decreasing temperature in room one! Temperature is
too high and humidity is too low!
17:12:17.125: Ventilation system is working at a higher capacity to decrease CO2
concentration and to increase humidify the air in room one!
17:12:17.250: Watering system is working to increase humidity in room one!
17:12:17.254: Heating system is at minimum power to increase humidity in room one!
Temperature is too high and humidity is too low!
17:12:17.254: Cooling system is decreasing temperature in room one! Temperature is
too high and humidity is too low!
17:12:17.254: Ventilation system is working at a higher capacity to decrease CO2
concentration and to increase humidify the air in room one!
17:12:17.379: Watering system is working to increase humidity in room one!
17:12:17.382: Heating system is at minimum power to increase humidity in room one!
Temperature is too high and humidity is too low!
17:12:17.383: Cooling system is decreasing temperature in room one! Temperature is
too high and humidity is too low!
17:12:17.383: Ventilation system is increasing humidity in room one!
17:12:17.509: Watering system is working to increase humidity in room one!
```

17:12:17.512: Heating system is at minimum power to increase humidity in room one!
 Temperature is too high and humidity is too low!
 17:12:17.513: Cooling system is decreasing temperature in room one! Temperature is
 too high and humidity is too low!
 17:12:17.513: Ventilation system is increasing humidity in room one!
 17:12:17.635: Watering system is working to increase humidity in room one!
 17:12:17.638: Heating system is at minimum power to increase humidity in room one!
 Temperature is too high and humidity is too low!
 17:12:17.640: Cooling system is decreasing temperature in room one! Temperature is
 too high and humidity is too low!
 17:12:17.640: Ventilation system is increasing humidity in room one!
 17:12:17.762: Watering system is working to increase humidity in room one!
 17:12:17.765: Cooling system is decreasing temperature in room one! Temperature is
 too high and humidity is too low!
 17:12:17.766: Heating system is at minimum power to increase humidity in room one!
 Temperature is too high and humidity is too low!
 17:12:17.767: Ventilation system is increasing humidity in room one!
 17:12:17.909: Watering system is working to increase humidity in room one!
 17:12:17.914: Heating system is at minimum power to increase humidity in room one!
 Temperature is too high and humidity is too low!
 17:12:17.914: Cooling system is decreasing temperature in room one! Temperature is
 too high and humidity is too low!
 17:12:17.915: Ventilation system is increasing humidity in room one!
 17:12:18.069: Watering system is working to increase humidity in room one!
 17:12:18.072: Ventilation system is increasing humidity in room one!
 17:12:18.073: Heating system is at minimum power to increase humidity in room one!
 Temperature is too high and humidity is too low!
 17:12:18.074: Cooling system is decreasing temperature in room one! Temperature is
 too high and humidity is too low!
 17:12:18.229: Watering system is working to increase humidity in room one!
 17:12:18.233: Ventilation system is increasing humidity in room one!
 17:12:18.233: Heating system is at minimum power to increase humidity in room one!
 Temperature is too high and humidity is too low!
 17:12:18.234: Cooling system is decreasing temperature in room one! Temperature is
 too high and humidity is too low!
 17:12:18.385: Watering system is working to increase humidity in room one!
 17:12:18.390: Cooling system is off in room one! Temperature is ok and humidity is
 too low!
 17:12:18.390: Heating system is at minimum power to increase humidity in room one!
 Temperature is ok and humidity is too low!
 17:12:18.391: Ventilation system is increasing humidity in room one!

Mărimile optime sunt calculate:

17:12:18.540: Ventilation system is working on standby mode in room
 one! CO2 concentration and humidity are ok!
 17:12:18.540: Cooling system is off in room one! Temperature and
 humidity are ok!
 17:12:18.540: Heating system is off in room one! Temperature and
 humidity are ok!
 17:12:18.589: System's parameters in room one have reached their
 optimal values!
 17:12:18.589: CO2 sensor 1 value is: 570.0 ppm.
 17:12:18.589: CO2 sensor 2 value is: 600.0 ppm.
 17:12:18.589: CO2 sensor 3 value is: 585.0 ppm.
 17:12:18.589: T sensor 1 value is: 29.000002 Celsius degrees.
 17:12:18.589: T sensor 2 value is: 28.900003 Celsius degrees.
 17:12:18.589: T sensor 3 value is: 29.000002 Celsius degrees.
 17:12:18.589: H sensor 1 value is: 40.0 % relative humidity.
 17:12:18.589: H sensor 2 value is: 40.0 % relative humidity.
 17:12:18.589: H sensor 3 value is: 40.0 % relative humidity.

3.4.3. Verificarea conductelor de aer și a senzorilor de viteză

În cazul în care diferența maximă admisă dintre senzorii de viteză a aerului depășește pragul de 5 m/s, agentul uman va fi alertat că există o defecțiune a senzorilor și va trebui să meargă în camera în care s-a depistat problema și să verifice senzorii care au declanșat alarma.

Totodată, sistemul de climatizare se va opri de urgență pentru evitarea unui caz nefericit în care senzorul care s-a defectat să afișeze o valoare „bună” (de exemplu 12 m/s), iar cel care merge corect să înregistreze o valoare dăunătoare sistemului.

De aceea, la declanșarea unei alarme pentru senzori defecți de viteză a aerului, sistemul de climatizare se va opri de urgență pentru evitarea eventualelor probleme, iar personalul calificat are obligația să verifice atât senzorii, cât și conductele de aer.

Sistemul de climatizare va reporni numai după ce HumanAgent-ul comunică agentului Maintenance că totul este în regulă cu conductele de aer, iar senzorii defecți au fost înlocuiți.

Intrările sistemului:

```
Insert number of CO2 sensors from room: 1
CO2 sensors range is 0 to 2000 ppm!
Insert CO2 sensor 1 value: 500
Insert number of temperature sensors from room: 2
Temperature sensors range is -5 to 55 Celsius!
Insert temperature sensor 1 value: 18
Insert temperature sensor 2 value: 19
Insert number of humidity sensors from room: 1
Relative humidity sensors range is 10 to 95%!
Insert humidity sensor 1 value: 30
Insert number of air velocity sensors from room: 2
Air velocity sensors range is 0 to 20 m/s!
Insert air velocity sensor 1 value: 1
Insert air velocity sensor 2 value: 12
Insert number of broken glass detectors from room: 1
Broken glass detectors values: 1 - glass is broken or 0 - glass is
OK
Insert broken glass detector 1 value: 0
```

Ieșirile sistemului:

```
17:15:29.360: In room two, the maximum range (5 m/s) between air
sensor 1 and air sensor 2 has been exceeded. Current range: 11.0
17:15:29.363: team@192.168.0.100:1111/JADE is going to room two to
check the air ducts and broken air sensors!
17:15:29.379: HVAC System has been shut down!
17:15:29.379: System's parameters cannot reach their optimal value!
```

3.4.4. Verificarea senzorilor de temperatură și a geamurilor

În cazul în care agentul de inferență detectează atât o eroare legată de senzorii de temperatură, cât și una legată de starea ferestrelor comunică agentului uman să se deplaseze imediat în camera cu pricina și să rezolve problemele, agenților, care controlează mărimile în cameră, să oprească activitatea pentru a nu face risipă de energie (VentilationAgent poate rămâne idle sau continua lucrul în modul standby, care nu necesită un consum ridicat de energie) și agentului din camera de control pentru a fi la curent cu starea din camera respectivă.

Agentul de inferență comunică atât camera în care s-a produs defecțiunea cât și senzorul care a generat-o.

Intrările sistemului:

```
Insert number of CO2 sensors from room: 2
CO2 sensors range is 0 to 2000 ppm!
Insert CO2 sensor 1 value: 720
Insert CO2 sensor 2 value: 760
Insert number of temperature sensors from room: 3
Temperature sensors range is -5 to 55 Celsius!
Insert temperature sensor 1 value: 34
Insert temperature sensor 2 value: 38
Insert temperature sensor 3 value: 30
Insert number of humidity sensors from room: 2
Relative humidity sensors range is 10 to 95%!
Insert humidity sensor 1 value: 18
Insert humidity sensor 2 value: 20
Insert number of air velocity sensors from room: 1
Air velocity sensors range is 0 to 20 m/s!
Insert air velocity sensor 1 value: 10
Insert number of broken glass detectors from room: 2
Broken glass detectors values: 1 - glass is broken or 0 - glass is
OK
Insert broken glass detector 1 value: 1
Insert broken glass detector 2 value: 0
```

Ieșirile sistemului:

```
17:18:15.893: In room four, the maximum range (3 Celsius degrees)
between temperature sensor 1 and temperature sensor 2 has been
exceeded. Current range: 7.27273
17:18:15.902: In room four, the maximum range (3 Celsius degrees)
between temperature sensor 1 and temperature sensor 3 has been
exceeded. Current range: 7.272725
17:18:15.910: In room four, the maximum range (3 Celsius degrees)
between temperature sensor 2 and temperature sensor 3 has been
exceeded. Current range: 14.545455
17:18:15.916: team@192.168.0.100:1111/JADE is going to room four to
check the broken sensors!
17:18:15.932: team@192.168.0.100:1111/JADE is going to room four to
check the windows!
17:18:15.933: In room four the window associated to sensor 1 is
broken!
17:18:15.941: HVAC System starts working in standby mode!
17:18:15.941: System's paramters cannot reach their optimal value!
```


4. CONCLUZII

Constanta evoluție ne determină să ne adaptăm la noi concepte precum ideea de „smart” sau la noi tehnologii precum inteligența artificială. În paralel, noile tehnologii trebuie să ajute la mărirea nivelului de confort din viața de zi cu zi, cu accent asupra asigurării unei tehnologii care să planifice o parte din activitățile casnice (automatizarea unui apartament sau case în ceea ce privește sistemul electric, sistemul de climatizare, sistemul de irigație etc.) sau la creșterea productivității la locul de muncă. Marile companii se întrec în ceea ce privește confortul la locul de muncă deoarece creșterea productivității atrage după sine, implicit, și o creștere a veniturilor.

În vederea soluționării problemei, au fost implementați nouă agenți cu ajutorul cărora se vor putea asigura condiții optime de locuit sau muncit în clădire. Principala sarcină a acestora este reprezentată de controlul temperaturii, al umidității și al concentrației de dioxid de carbon, dar și verificarea și mentenanța senzorilor asociați și a întregului sistem de climatizare.

În prezentele simulări s-a demonstrat că reglarea mărimilor ce caracterizează un sistem de climatizare într-o clădire se face într-un mod rapid și eficient, iar eventualele erori sunt semnalate și se așteaptă soluționarea acestora. Așadar, sistemele multi-agent reprezintă o soluție ce asigură o bună funcționalitate a sistemului de climatizare din punctul de vedere al controlului și mentenanței.

Soluția prezentată poate fi îmbunătățită prin adăugarea unor altor tipuri de senzori și implementarea acestora, precum senzori magnetici care detectează nivelul de ocupanță în cameră și ajustează sistemul de ventilație corespunzător sau senzori optici la nivelul conductelor pentru a determina gradul de murdărie și a păstra conductele de ventilație într-o stare cât mai bună și a evita eventualele mirosuri neplăcute.

5. ANEXE

Modul de transmisie – recepție dintre agenții RoomAgent și SensorAgent pentru temperatură

Transmiterea mesajului:

```
System.out.print("Insert number of temperature sensors from room: ");
Scanner n2 = new Scanner(System.in);
m2 = Integer.parseInt(n2.next());
vT = new float[m2];
sT = new String[m2];
System.out.println("Temperature sensors range is -5 to 55
Celsius!");
for (int i = 0; i < m2; i++) {
    System.out.print("Insert temperature sensor " + (i+1) + " value: ");
// 23-29 C normal
    Scanner s = new Scanner(System.in);
    vT[i] = Float.parseFloat(s.next());
}
msg.addReceiver(sensor);
msg.setConversationId("T");
msg.setContent(Integer.toString(m2));
send(msg);
msg.clearAllReceiver();
for(int j = 0; j < m2; j++) {
    sT[j] = Float.toString(vT[j]);
    msg.addReceiver(sensor);
    msg.setConversationId("T"+j);
    msg.setContent(sT[j]);
    send(msg);
    msg.clearAllReceiver();
}
```

Recepția mesajului:

```
if("T".equalsIgnoreCase(received.getConversationId())) {
    n2 = Integer.parseInt(received.getContent());
    vT = new float[n2];
    sT = new String[n2];
}
for (int i = 0; i < n2; i++) {
    if (received.getConversationId() == null ? ("T"+i) == null :
received.getConversationId().equals("T"+i)) {
        //          System.out.println("Primit de senzor: " +
received.getConversationId() + " " + received.getContent());
        vT[i] = (((Float.parseFloat(received.getContent()) +5) * 10) /
55);
        if(i == (n2-1)) {
            ok2 = true;
        }
    }
}
```

La recepția mesajului, agentul scalează valorile primite pentru a le putea trimite ulterior agentului de inferență.

Recepția și verificarea valorilor de către agentul de inferență

Recepția mesajului transmis de către agentul senzor:

```
if("T".equalsIgnoreCase(received.getConversationId())) {
    n2 = Integer.parseInt(received.getContent());
    vT = new float[n2];
}
for (int i = 0; i < n2; i++) {
    if (received.getConversationId().equals("T"+i)) {
        //          System.out.println("Am receptionat: " +
received.getConversationId() + " " + received.getContent());
        vT[i] = Float.parseFloat(received.getContent());
        if(i == (n2-1)) {
            ok2 = true;
        }
    }
}
```

Verificarea senzorilor pentru detecția erorilor:

```
//Check temperature sensors and get MA if no errors;
if(ok2) {
    defect2 = false;
    if (vT.length == 1) {
        m2 = vT[0];
        go2 = true;
    }
    else {
        for(int i = 0; i < n2; i++) {
            s2 = s2 + vT[i];
            for(int j = i+1; j < n2; j++)
                if(Math.abs(vT[i] - vT[j]) > 0.54) {
                    msg.addReceiver(ment);
                    msg.setConversationId("Temperature sensors not
working properly!");
                    msg.setContent(Integer.toString(i+1) + " " +
Integer.toString(j+1) + " " + Float.toString((Math.abs(vT[i]-vT[j]))*10));
                    send(msg);
                    msg.clearAllReceiver();
                    go2 = false;
                    defect2 = true;
                    // System.out.println("Defectiune senzori temperatura
" + Math.abs(vT[i] - vT[j]))
                }
            }
        }
        if(!defect2) {
            m2 = s2 / n2;
            go2 = true;
        }
        // System.out.println("Medie temperatura: " + m2);
    }
    ok2 = false;
    s2 = 0;
}
```

Prezentarea regulii care determină scăderea concentrației de CO2 din aer, creșterea temperaturii și scaderea umidității

```
if (VentONcd != null && HeatON != null && CoolONd != null &&
WaterONd != null) {
    for(int i = 0; i < m1; i++)
        suma1 = suma1 + vC[i];
    medie1 = suma1 / m1;
    for(int i = 0; i < m1; i++) {
        if(vC[i] >= medie1)
            vC[i] = vC[i] - 50;
        ok1 = true;
    }
    for(int i = 0; i < m2; i++)
        suma2 = suma2 + vT[i];
    medie2 = suma2 / m2;
    for(int i = 0; i < m2; i++) {
        if(vT[i] <= medie2)
            vT[i] = vT[i] + 0.4f;
        ok2 = true;
    }
    for(int i = 0; i < m3; i++)
        suma3 = suma3 + vH[i];
    medie3 = suma3 / m3;
    for(int i = 0; i < m3; i++) {
        if(vH[i] >= medie3)
            vH[i] = vH[i] - 1;
        ok3 = true;
    }
    if (ok1 && ok2 && ok3) {
        // se realizează transmisia înapoi către
    }

    ok1 = false;
    ok2 = false;
    ok3 = false;
    suma1 = 0;
    suma2 = 0;
    suma3 = 0;
    medie1 = 0;
    medie2 = 0;
    medie3 = 0;
    VentONcd = null;
    HeatON = null;
    CoolONd = null;
    WaterONd = null;
}
```

Transmisia – recepția dintre RoomAgent și MaintenanceRoom în cazul în care mărimile au fost controlate și afișarea acestora

Transmisia efectuată de către RoomAgent:

```
if (VentIdle != null && HeatOFF != null && CoolOFF != null) {
    msg.addReceiver(ment);
    msg.setConversationId("C");
    msg.setContent(Integer.toString(m1));
    send(msg);
    msg.clearAllReceiver();
    for(int j = 0; j < m1; j++) {
        sC[j] = Float.toString(vC[j]);
        msg.addReceiver(ment);
        msg.setConversationId("C"+j);
        msg.setContent(sC[j]);
        send(msg);
        msg.clearAllReceiver();
    }
    msg.addReceiver(ment);
    msg.setConversationId("T");
    msg.setContent(Integer.toString(m2));
    send(msg);
    msg.clearAllReceiver();
    for(int j = 0; j < m2; j++) {
        sT[j] = Float.toString(vT[j]);
        msg.addReceiver(ment);
        msg.setConversationId("T"+j);
        msg.setContent(sT[j]);
        send(msg);
        msg.clearAllReceiver();
    }
    msg.addReceiver(ment);
    msg.setConversationId("H");
    msg.setContent(Integer.toString(m3));
    send(msg);
    msg.clearAllReceiver();
    for(int j = 0; j < m3; j++) {
        sH[j] = Float.toString(vH[j]);
        msg.addReceiver(ment);
        msg.setConversationId("H"+j);
        msg.setContent(sH[j]);
        send(msg);
        msg.clearAllReceiver();
    }
}
```

Recepția realizată de MaintenanceAgent și afișarea valorilor:

Primirea valorilor concentrației de carbon din aer:

```
if("C".equals(received.getConversationId())) {
    n1 = Integer.parseInt(received.getContent());
    vC = new float[n1];
    sC = new String[n1];
    //          System.out.println("Se   primește:   "   +
received.getConversationId() + " " + received.getContent());
}
for (int i = 0; i < n1; i++) {
    if (received.getConversationId().equals("C"+i)) {
        //          System.out.println("OK   Value   carbon:   "   +
received.getConversationId() + " " + received.getContent());
        vC[i] = Float.parseFloat(received.getContent());
        if(i == (n1-1)) {
            ok1 = true;
        }
    }
}
}
```

Primirea valorilor temperaturii:

```
if("T".equals(received.getConversationId())) {
    n2 = Integer.parseInt(received.getContent());
    vT = new float[n2];
    sT = new String[n2];
    //          System.out.println("Se   primește:   "   +
received.getConversationId() + " " + received.getContent());
}
for (int i = 0; i < n2; i++) {
    if (received.getConversationId() == null ? ("T"+i) == null :
received.getConversationId().equals("T"+i)) {
        //          System.out.println("OK   Value   temp:   "   +
received.getConversationId() + " " + received.getContent());
        vT[i] = Float.parseFloat(received.getContent());
        if(i == (n2-1)) {
            ok2 = true;
        }
    }
}
}
```

Primirea valorilor de umiditate:

```
if("H".equals(received.getConversationId())) {
    n3 = Integer.parseInt(received.getContent());
    vH = new float[n3];
    sH = new String[n3];
    //      System.out.println("Se   primește:   "   +
received.getConversationId() + " " + received.getContent());
}
for (int i = 0; i < n3; i++) {
    if (received.getConversationId() == null ? ("H"+i) == null :
received.getConversationId().equals("H"+i)) {
        //      System.out.println("OK   Value   umiditate:   "   +
received.getConversationId() + " " + received.getContent());
        vH[i] = Float.parseFloat(received.getContent());
        if(i == (n3-1)) {
            ok3 = true;
        }
    }
}
```

Afișarea valorilor optime:

```
if(ok1 && ok2 && ok3) {
    System.out.println(data.format(cal.getTime()) + ": System's
parameters in room " + roomNumber + " have reached their optimal
values!");
    for (int i = 0; i < n1; i++)
        System.out.println(data.format(cal.getTime()) + ": CO2 sensor
" + (i+1) + " value is: " + vC[i] + " ppm.");
    for (int i = 0; i < n2; i++)
        System.out.println(data.format(cal.getTime()) + ": T sensor " +
(i+1) + " value is: " + vT[i] + " Celsius degrees.");
    for (int i = 0; i < n3; i++)
        System.out.println(data.format(cal.getTime()) + ": H sensor " +
(i+1) + " value is: " + vH[i] + " % relative humidity.");
    ok1 = false;
    ok2 = false;
    ok3 = false;
}
```


6. BIBLIOGRAFIE

Andreadis G., Klazoglou P., Niotaki K., Bouzakis K. 2013. *Classification and Review of Multi-Agents Systems in the Manufacturing Section*. International Symposium on Intelligent Manufacturing and Automation.

Arundel A., Sterling E.M., Sterling T.D. 1985. *Criteria for Human Exposure to Humidity in Occupied Buildings*. ASHRAE Transaction, vol. 91, part. 1.

CDCL. 2014 – *Carbon Dioxide Comfort Levels*. Available at http://www.engineeringtoolbox.com/co2-comfort-level-d_1024.html accesat la data de 12.06.2014.

FHC. 2014. *Fundamentals of HVAC Controls*. Available at <http://www.cs.berkeley.edu/~culler/cs294-f09/m197content.pdf> accesat la data de 03.05.2014.

FIPA. 2014. *FIPA Agent Management Specification*. Available at http://www.fipa.org/specs/fipa00023/XC00023H.html#_Toc526742635 accesat la data de 18.04.2014.

Franklin S., Graesser A. 1996. *Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents*.

Gavazzi. 2014. *Environmental and Wind Sensors from Carlo Gavazzi*. Available at: https://www.gavazzionline.com/environmental_sensors.htm, accesat la data de 12.06.2014.

JADE. 2014. *JADE – Java Agent Development Framework*. Available at <http://jade.tilab.com/> accesat la data de 05.03.2014.

Khemani H. 2009. *Psychrometric Processes: Cooling & Humidification and Heating & Humidification* available at <http://www.brighthubengineering.com/hvac/41447-psychrometric-processes-cooling-heating-and-humidification/> accesat la data de 05.06.2014.

Khemani H. 2009. *Psychrometric Processes: Cooling & Dehumidification and Heating & Dehumidification* available at <http://www.brighthubengineering.com/hvac/41505-psychrometric-processes-cooling-heating-and-dehumidification/> accesat la data de 05.06.2014.

Maes P. 1995. *Artificial Life Meets Entertainment: Lifelike Autonomous Agents*.

Pătraşcu M., Drăgoicea M. 2014. *Integrating agents and services for control and monitoring: Managing emergencies in smart building*. Studies in Computational Intelligence 544 PP. 209 – 224.

Pătraşcu M., Drăgoicea M., Ion A. 2014. *Emergent Intelligence in Agents: a Scalable Architecture for Smart Cities*. International Conference on System Theory, Control and Computing, Sinaia, România.

Russel S., Norvig P. 2003. *Artificial Intelligence: A Modern Approach*.

Sen J. 2010. *An Agent-Based Intrusion Detection System for Local Area Networks*. International Journal of Communication Networks and Information Security.

SGD. 2014. *DS1101i Series Glassbreak Detectors*. Available at http://resource.boschsecurity.com/documents/DS1101i_Series_Data_sheet_enUS_2636471179.pdf accesat la data de 15.06.2014.

Sinopoli J. 2010. *Smart Building Systems for Architects, Owners and Builders*.

Weiss G. 1999. *MultiAgent Systems: A Modern Approach to Distributed Artificial Intelligence*.

Wooldridge M. 2002. *An Introduction to MultiAgent Systems*. Liverpool, UK.