

**UNIVERSITATEA DIN CRAIOVA  
FACULTATEA DE ECONOMIE ȘI ADMINISTRAREA AFACERILOR  
ȘCOALA DOCTORALĂ DE ȘTIINȚE SOCIO-UMANE  
DOMENIUL: CIBERNETICĂ ȘI STATISTICĂ ECONOMICĂ**

**REZUMATUL TEZEI DE DOCTORAT**

**SISTEME MULTI AGENT  
CU APLICAȚII ÎN E-BUSINESS**

**Coordonator științific,  
Prof. univ. dr. Vasile GEORGESCU**

**Doctorand,  
Adrian Victor SĂNDIȚĂ**

**Craiova  
2015**

## CUPRINS

<b>1. Cuprinsul tezei de doctorat</b>	<b>3</b>
<b>2. Cuvinte cheie</b>	<b>5</b>
<b>3. Motivația și importanța cercetării</b>	<b>5</b>
<b>4. Prezentarea sintetică a capitolelor cuprinse în teza de doctorat</b>	<b>6</b>
<b>5. Sinteza concluziilor cercetării științifice</b>	<b>11</b>
<b>6. Bibliografie selectivă</b>	<b>16</b>

# 1. CUPRINSUL TEZEI DE DOCTORAT

<b>INTRODUCERE</b>	<b>5</b>
<b>1 SISTEME MULTI-AGENT</b>	<b>10</b>
1.1. Agenții și caracteristicile acestora	10
1.2. Utilizarea agenților	13
1.2.1 Reguli de utilizare	15
1.2.2 Comunicarea între agenți	16
1.3. Criterii de clasificare a agenților	17
1.4. Mediul în care acționează agenții	19
1.5. Comportamentul agenților	21
1.6. Tipuri de agenți	24
1.6.1. Agenți cu arhitectură BDI	24
1.6.2. Agenți cu structură stratificată	26
1.6.3. Agenți inteligenți	28
1.6.4. Agenți cu arhitectură reactivă	30
1.6.5. Agenți cu arhitectură hibridă	32
1.6.6. Coordonarea în sistemele multi agent	33
1.7. Concluzii	33
<b>2. PLATFORME ORIENTATE CĂTRE MAS</b>	<b>36</b>
2.1. Standarde FIPA	36
2.1.1. Specificații FIPA de implementare	36
2.1.2. Ciclul de viață al standardelor FIPA	37
2.1.3. Specificațiile pentru managementul agenților	39
2.2. Cougaar	42
2.2.1. Modelul Cougaar	43
2.2.2. Comunități Cougaar	44
2.3. JADE	45
2.3.1. Caracteristicile arhitecturii JADE	46
2.3.2. Implementarea JADE	48
2.4. Concluzii	51
<b>3. TEHNICI DE INTELIGENȚĂ COMPUTAȚIONALĂ</b>	<b>53</b>
3.1. Definirea conceptelor	53
3.2. Rețele neuronale	55
3.2.1. Avantajele rețelelor neuronale	58
3.2.2. Modelul neuronului artificial	60
3.2.3. Arhitectura rețelelor neuronale	62
3.2.4. Elementele unei rețele neuronale	65
3.2.5. Perceptronul	67
3.2.6. Instruirea perceptronilor	69
3.3. Mașini cu vector suport	70
3.4. Concluzii	75
<b>4. SISTEM MULTI-AGENT DE TRANZAȚIONARE AUTOMATĂ FOLOSIND TEHNICI DE INTELIGENȚĂ COMPUTAȚIONALĂ</b>	<b>77</b>
4.1. Obiectivele studiului	77
4.2. Construcția sistemului	77
4.2.1. Market Maker	77
4.2.2. Arena Gateway	79
4.2.3. Sistemul multi-agent proiectat	86
4.2.4. Analiza performanțelor sistemului de agenți în condiții de volatilitate	97

4.3. Datele utilizate -----	101
4.4. Instruirea supervizată a agenților unei platforme MaS-----	105
4.5. Rezultatele obținute în faza de estimare și predicție utilizând operatori Rapidminer -----	116
4.6. Îmbunătățirea performanțelor predictive ale rețelelor neuronale prin aplicarea strategiei de instruire cu oprire timpurie-----	144
4.7. Instanțierea automată a agenților predictivi prin integrarea tehnicilor de inteligență computațională într-un sistem multi-agent -----	151
CONCLUZII ȘI DEZVOLTĂRI ULTERIOARE -----	153
BIBLIOGRAFIE-----	158

## 2. CUVINTE CHEIE

- Agenți inteligenți;
- Bursa de Valori București (BVB);
- Fundația pentru Agenți Fizici Inteligenți (FIPA)
- Instruire supervizată;
- Inteligență computațională;
- Market Maker;
- Mașini cu vector suport (SVM);
- Platforme orientate agent;
- Rețele neuronale (NN);
- Sisteme multi-agent (MaS);
- Tranzacționare automată.

## 3. MOTIVAȚIA, OBIECTIVELE ȘI IMPORTANȚA CERCETĂRII

Schimburile electronice joacă un rol tot mai important pe piețele financiare, iar studiul mecanismelor de tranzacționare utilizate pentru valori mobiliare financiare reprezintă cheia înțelegerii lor. Cele mai multe burse de valori importante utilizează mecanismul formatorilor de piață (Market Maker) pentru a asigura lichiditatea tranzacțiilor și pentru a oferi o piață calitativ mai bună și mai eficientă (Kim A.J., 2002). Fiecare emitent are unul sau mai mulți formatori ce acționează pentru a furniza lichiditate pentru acțiunile emitentului în cauză. Responsabilitatea formatorului de piață este de a stabili volumele și prețurile la care este dispus să vândă, respectiv să cumpere (Stone P. și Sherstov A.A., 2005). Formatorii de piață reprezintă un avantaj major pentru piața pe care acționează. Traderii care doresc să cumpere sau să vândă acțiunile unui emitent nu au nevoie să se aștepte unul pe celălalt pentru a realiza tranzacții dorite. În plus, existența unui *Market Maker* descurajează traderii care încearcă să manipuleze piața prin introducerea de ordine care să determine fluctuațiile de preț în direcția dorită (Nevmyvaka Y. et al., 2005).

Formatorii de piață beneficiază la rândul lor de poziția pe care o dețin. Deși nu direct, un *Market Maker* este avantajat pentru că activitatea desfășurată îi permite să cumpere ieftin și să vândă scump (Feng Y. et al., 2004). Formatorul de piață introduce două cotații: una de vânzare și una de cumpărare, astfel încât ceilalți traderi pot să cumpere, respectiv să vândă la prețurile stabilite de *Market Maker*. Evident, prețul de vânzare este mai mare decât cel de cumpărare, iar diferența dintre cele două prețuri poartă numele de *spread*. Pentru a crește lichiditatea pieței, spreadul trebuie să fie cât mai mic. Pe de altă parte, pentru ca șansele de câștig ale formatorului de piață să crească, valoarea spreadului trebuie să fie mare.

Beneficiile automatizării acestei activități sunt maxime: o mașină instruită corespunzător poate reacționa întotdeauna mult mai rapid la schimbările din piață decât un trader uman, iar deciziile pe care le poate lua nu sunt afectate de inconsecvența care caracterizează în multe cazuri deciziile umane.

Obiectivul principal al cercetării din cadrul tezei de doctorat a fost reprezentat de studiul necesar proiectării, realizării și implementării în mediul real al unui sistem multi-agent de tranzacționare care să automatizeze activitatea realizată de un *Market Maker* în cadrul Bursei de Valori București. Deciziile de tranzacționare automată a sistemului multi-agent realizat au la bază predicțiile privind evoluția trendului, emise de agenți inteligenți ce folosesc în analiză metode din domeniul inteligenței computaționale: rețele neuronale și mașini cu vector suport (Sândiță și Durac, 2015).

## 4. PREZENTAREA SINTETICĂ A CAPITOLELOR CUPRINSE ÎN TEZA DE DOCTORAT

Teza de doctorat este alcătuită din introducere, patru capitole, o parte dedicată concluziilor și dezvoltărilor ulterioare și se încheie cu bibliografia.

### Capitolul 1. Sisteme multi-agent

Termenul de agent are înțelesuri multiple care depind de contextul în care se utilizează. Wooldridge (1997) definește un agent inteligent ca un sistem care are următoarele patru proprietăți: *autonomie* (agenții funcționează fără intervenția directă a utilizatorilor și au controlul asupra acțiunilor și a stărilor lor interne), *abilitate socială* (agenții sunt capabili să coopereze cu utilizatorii sau alți agenți în vederea realizării sarcinilor), *reactivitate* (agenții percep mediul și răspund în timp util la schimbările care apar) și *pro-activitate* (agenții nu acționează pur și simplu ca răspuns la mediul în care se află, ei sunt capabili să aibă un comportament obiectiv prin luarea de inițiative).

Un agent este o entitate software care caută în mod activ modalități de a-și finaliza sarcinile. Agenții inteligenți au capacitatea de a dobândi cunoștințe prin intermediul proceselor de soluționare a problemelor. Agenții software, se concentrează pe interacțiuni și colaborări pentru a-și atinge obiectivele într-un context care se schimbă într-o manieră de obicei neprevăzută. Agenții sunt folosiți atunci când complexitatea sistemelor software clasice de mari dimensiuni ridică probleme de design pe care tehnologia convențională nu reușește să le rezolve sau atunci când modularitatea, viteza de execuție, fiabilitatea, coordonarea și comunicarea între componente sunt caracteristici ale sistemului ce urmează a fi implementat. Într-un sistem dinamic distribuit, agenții cu capacitate de auto-reglare pot simplifica designul sistemului, care poate fi extrem de complicat în arhitecturile tradiționale, bazate pe modele orientate obiect.

Modelarea orientată pe agenți reprezintă o abordare neconvențională a designului sistemului, inclusiv în definirea componentelor și în integrarea sistemului. Autonomia este o proprietate distinctivă a unui agent și presupune capacitatea agentului de a supraviețui într-un mediu în schimbare. Un agent are capacitatea de a percepe factorii de mediu și de a lua decizii cu privire la modul în care trebuie să reacționeze în consecință. Adaptabilitate presupune existența capacităților de învățare necesare agentului pentru adaptarea deciziilor în funcție de experiența din trecut. Mai mult decât atât, un design orientat agent ar trebui să dezvolte metode și tehnici care îl fac să reacționeze corect atunci când apar evenimente neașteptate.

Inteligența Artificială Distribuită (*Distributed Artificial Intelligence – DAI*) este un subdomeniu al inteligenței artificiale care se ocupă cu rezolvarea problemelor în care agenții interacționează în scopul de a rezolva o problemă comună (Green et al. 1997). Agenții pot moșteni potențiale beneficii atât din partea *DAI*: modularitate, viteză, fiabilitate, cât și din partea inteligenței artificiale: operare la nivel de cunoștințe, întreținere, platformă reutilizabilă, independență (Nwana, 1996).

În proiectarea de sisteme mari și/sau complexe, un agent este o abstracție care facilitează proiectarea unor componente diferite care colaborează în rezolvarea diferitelor aspecte ale unei probleme. Fiecare agent este conceput în paradigma cea mai potrivită pentru a rezolva partea sa de o problemă. Un sistem multi-agent este folosit pentru a rezolva o problemă complexă care nu poate fi rezolvată de către o entitate unică a sistemului. Agenții au capacitatea de a procesa eficient datele locale și de a comunica cu alți agenți

atunci când este necesar, în cazul în care sarcinile cu care se confruntă sunt dincolo de domeniul lor de cunoștințe.

Sisteme multi-agent au fost utilizate într-un spectru larg de aplicații, cum ar fi e-commerce, e-learning, comunicare, data mining, simulare, robotică, sisteme de transport și de grid computing. Au fost inițiate, de asemenea, studii teoretice în specificațiile și raționamentelor sistemelor multi-agent, reprezentarea și prelucrarea cunoștințelor, precum și în științele cognitive.

Studiul sistemelor multi-agent a stat la baza cercetării noastre. Metodologia implementării agenților, proprietățile definitorii ale acestora, modul de comunicare și arhitecturile sistemelor multi-agent studiate ne-au permis realizarea sistemului multi-agent prezentat în capitolul al patrulea al lucrării.

## Capitolul 2. Platforme orientate către MaS

Platformele și instrumentele de dezvoltare multi-agent sunt componente importante care afectează difuzarea și utilizarea tehnologiilor orientate către agenți în diferite domenii. De fapt, succesul implementării sistemelor multi-agent depinde în mare măsură de disponibilitatea tehnologiei corespunzătoare, care să permită punerea în aplicare a conceptelor și tehnicilor care stau la baza sistemelor multi-agent (Ricordel și Demazeau, 2000).

Platforme software și framework-urile sunt mijloace generice esențiale în dezvoltarea sistemelor multi-agent (Odell et al., 2002). Cele mai multe oferă un mijloc de a implementa sisteme multi-agent pe diferite tipuri de sisteme de hardware și sub diverse sisteme de operare, oferind de obicei instrumente și tehnici care sprijină rularea lor și permit implementarea facilă a operațiunilor esențiale, cum ar fi comunicarea și coordonarea. Unele dintre aceste platforme și framework-uri au ca obiectiv comun furnizarea de funcționalități standardizate pentru a sprijini interoperabilitatea între diferite sisteme multi-agent. Mai mult decât atât, unele au de asemenea obiectivul de a sprijini diferite tipuri de hardware, rețele de comunicații și arhitecturi de agent (JADE, Cougaar etc.), iar altele de a implementa numai tipuri speciale de agenți, cum ar fi, de exemplu, agenții mobili (Lange și Oshima, 1998).

Limbajele și platformele orientate către agent permit cercetătorilor și dezvoltatorilor de aplicații să se concentreze mai intens asupra sarcinilor pe care agenții trebuie să le execute și să consume mai puține resurse pentru implementarea efectivă a sistemelor de agenți.

Fundația pentru Agenți Fizici Inteligenți (Foundation for Intelligent Physical Agents – *FIPA*) este o organizație internațională care s-a dedicat promovării standardelor în domeniul conceperii, implementării și utilizării agenților și sistemelor bazate pe agenți. Standardele *FIPA* reprezintă un punct de referință pentru dezvoltatorii de sisteme bazate pe agenți. Respectarea specificațiilor *FIPA* oferă câteva argumente care ar putea fi determinante în selecția platformelor pe care urmează a se implementa o anumită clasă de agenți. Astfel, compatibilitatea cu standardele *FIPA* garantează arhitectura și performanța sistemului bazată pe protocoalele puternice și pe deplin testate, care permit comunicarea între agenți, cooperarea și interoperabilitatea (*FIPA* website). Prin urmare, se preferă respectarea standardelor *FIPA* atunci când sistemele ce urmează a fi implementate sunt dinamice, flexibile și reconfigurabile. În plus, comunitatea cercetătorilor sistemelor de agenți utilizează în mod extins platformele compatibile *FIPA* și deci există un mare număr de implementări *open source* a unor clase Java, îndelung testate și bine documentate ce pot fi utile în dezvoltarea aplicațiilor. Cu toate acestea, diversitatea aplicațiilor și domeniilor în care sunt folosiți agenții determină inerent varietatea mecanismelor de abordare și rezolvare

a problemelor legate de implementare, pentru multe din acestea respectarea standardelor reprezentând o constrângere și nu un avantaj.

Astfel, platforma Cougaar descrisă în acest capitol, implementează în mod particular agenții, sarcinile pe care aceștia le pot executa precum și modalitatea de comunicare între entitățile ce activează în sistem. Modelul Cougaar adaugă sistemului tradițional serviciul *Binders*, parte a unui mecanism puternic de izolare și încapsulare pentru toate componentele (Helsinger et al., 2004). Standardizarea unor astfel de mecanisme ar fi costisitoare în raport cu aria de aplicabilitate.

Totuși, pentru foarte multe direcții de cercetare în domeniul sistemelor de agenți, compatibilitatea, portabilitatea și mediul de programare facil sunt argumente extrem de importante în favoarea alegerii unei platforme care respectă standardul FIPA, de tipul JADE.

Arhitectura JADE este bazată pe existența unor containere în care trăiesc agenții și care asigură logistica executării sarcinilor pe care agenții trebuie să le îndeplinească. Există un container principal care reține adresele și caracteristicile tuturor celorlalte containere și gestionează tabela ce reține informații despre toți agenții platformei (JADE website).

Containerele conțin și o tabelă ce reține descrierile agenților înregistrați local, iar comunicația între containere se realizează prin intermediul unui protocol specific, implementat local. Fiind folosit exclusiv pentru comunicația între agenții aceleiași platforme, fără să depășească granițele acesteia, protocolul folosit nu e constrâns să respecte standardul FIPA. De fapt, JADE folosește acest protocol și pentru a lansa comenzi specifice platformei distribuite și pentru a monitoriza la distanță starea containerelor platformei.

Spre deosebire de protocolul folosit în cadrul platformei, pentru asigurarea comunicării cu entitățile situate în afară, *Message Transport Protocol* respectă standardele FIPA. JADE suportă de asemenea interacțiuni complexe și conversații paralele multiple și oferă un set de cadre de modele de interacțiune specifice anumitor sarcini. Prin utilizarea acestor cadre (implementate ca și clase abstracte Java), programatorii se pot degreva de tratarea problemelor legate de sincronizarea activității agenților.

În plus, pentru a crește scalabilitatea sau pentru a îndeplini constrângerile unui mediu cu resurse limitate, JADE oferă posibilitatea execuției în paralel a mai multor sarcini în același fir Java.

Deși sistemul multi-agent proiectat de noi nu a fost implementat pe o platformă dedicată, studiul platformelor și limbajelor orientate către agenți ne-a permis identificarea și integrarea în proiectul nostru a mai multor elemente absolut necesare funcționării sistemului: comunicarea asincronă inter-agent, mecanismul *White și Yellow Pages* și nu în ultimul rând, mecanismul de coordonare și supervizare al agenților.

### **Capitolul 3. Tehnici de inteligență computațională**

Tehnicile de Inteligență Computațională sunt din ce în ce mai mult utilizate în rezolvarea problemelor ce nu pot fi abordate prin tehnici tradiționale sau atunci când informațiile de care se dispune sunt insuficiente pentru crearea unui model pe baza căruia să se poată dezvolta un algoritm de rezolvare.

În acest capitol au fost prezentate două din tehnicile specifice domeniului, tehnici folosite în dezvoltarea sistemului multi-agent realizat, respectiv rețelele neuronale artificiale și mașinile cu vectori suport.

Principala calitate a rețelelor neuronale rezidă în aceea că pot înmagazina cunoștințe pe care le pot utiliza ulterior în situații noi. Achiziția cunoștințelor în rețelele neuronale se



face prin stocarea unor valori la nivelul ponderilor sinaptice, valori care depind în egală măsură de arhitectura rețelei ce le înglobează. Procesul de atribuire de valori ponderilor sinaptice dintr-o rețea neuronală poartă numele de *proces de învățare* sau *proces de antrenare*.

Așa cum am văzut, se disting două tipuri fundamentale de învățare: *învățarea supervizată* și *învățarea nesupervizată*. În cazul învățării supervizate, se prezintă rețelei o mulțime de exemple de învățare, reprezentate de perechi vectoriale intrare-ieșire, caracteristice cunoștințelor ce urmează a fi achiziționate. Valorile vectorilor de intrare sunt introduse în calcule, afectând ponderile curente ale rețelei și determinând apariția la ieșire a unor valori ce se compară cu rezultatele așteptate. În funcție de abaterile dintre valorile obținute și cele așteptate, se ajustează ponderile, astfel încât diferențele să devină nule sau cât mai mici.

Întregul proces este repetat până când se consideră că rețeaua a fost complet instruită. Algoritmul de învățare se bazează pe aceea că dacă rețeaua se comportă într-un anumit fel în situații cunoscute, își va păstra comportamentul și în situații noi, necunoscute. Datele folosite pentru învățare poartă numele de *date de antrenament*, iar cele folosite pentru urmărirea comportamentului rețelei în situații noi poartă numele de *date de test*.

În cazul învățării nesupervizate, mulțimea de date pentru învățare constă doar dintr-o mulțime de vectori de intrare care, la fel ca în cazul învățării supervizate afectează ponderile stocate în rețea. Diferența față de învățarea supervizată este că ajustarea ponderilor se face în acest caz pentru ca în cazul unor vectori de intrare apropiați să se obțină același vector de ieșire sau vectori de ieșire cu valori foarte apropiate.

O problemă care apare frecvent în cazul rețelelor cu propagare înapoi și care scade dramatic eficiența învățării este cea a minimelor locale. Pentru că procesul de instruire se oprește atunci când rețeaua ajunge într-un minim global al funcției criteriu sau într-un minim local, trebuie implementată o metodă care să determine ieșirea rețelei din punctele de minim local, scopul fiind atingerea unui minim global. De obicei, metodele folosite impun schimbarea ponderilor inițiale, a constantei de instruire sau a numărului de unități ascunse.

Oricum, dacă soluția obținută ca urmare a instruirii rețelei este acceptabilă din punctul de vedere al ratei de eroare, faptul că rețeaua a ajuns într-un punct de minim local sau într-unul de minim global nu mai este relevant.

Mașinile cu vector suport (Support Vector Machines – SVM) reprezintă o metodă eficientă în proiectarea unei rețele feedforward cu un singur strat ascuns de unități neliniare. După cum sugerează și numele, proiectarea acestui tip de rețele neuronale are la bază extragerea unui subset al datelor de antrenament ce servește ca suport vectorial și reprezintă o caracteristică stabilă a datelor (Vapnik și Chervonenkis, 1971).

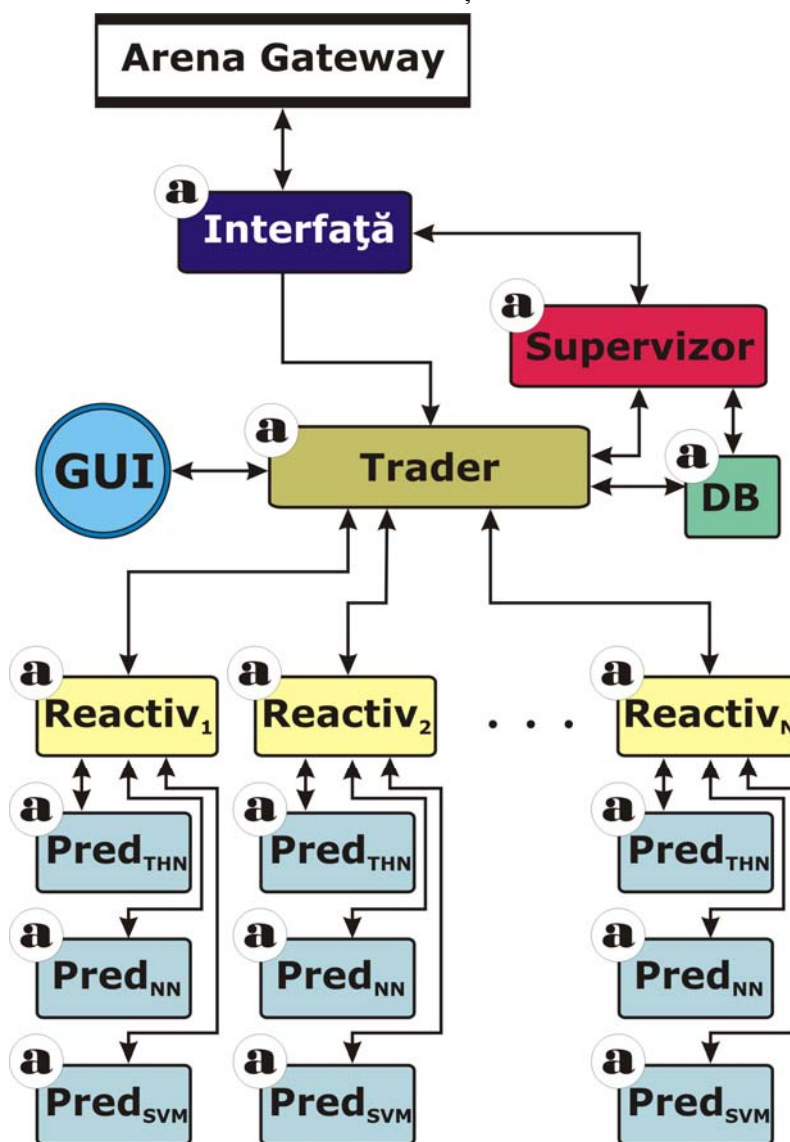
În literatura de specialitate, SVM s-au impus ca fiind cel mai utilizat algoritm datorită performanțelor foarte bune de generalizare, fundamentelor teoretice riguroase, relativ ușoara lor implementare și capacității de a oferi performanțe remarcabile în problemele de recunoaștere a formelor și regresie (Yom-Tov, 2007).

În realizarea sistemului de agenți prezentat în capitolul următor s-a folosit învățarea supervizată, datele de antrenament și cele de test fiind preluate din sistemul de tranzacționare al Bursei de Valori București.

## Capitolul 4. Sistem multi-agent de tranzacționare automată folosind tehnici de inteligență computațională

Acest capitol prezintă în detaliu arhitectura, modul de implementare și funcționare al sistemului multi-agent ce a constituit de fapt obiectul de studiu al cercetării. Acesta este primul și deocamdată singurul sistem funcțional de tranzacționare automată realizat în România, dedicat activității desfășurată de Market Makeri la Bursa de Valori București. Sistemul funcționează în timp real, este conectat la Bursa de Valori București printr-o interfață dedicată, Arena Gateway și permite comunicația bidirecțională printr-un flux continuu de date cu serverele de tranzacționare.

În cadrul capitolului este descrisă activitatea unui Market-Maker, regulile după care aceasta se desfășoară, constrângerile sistemului și mecanismul global prin care se realizează conexiunea și comunicarea cu serverele de tranzacționare ale Bursei de Valori București.



Sistemul de agenți

În figura de mai sus este prezentată arhitectura sistemului multi-agent realizat. Sistemul este conectat la sistemul de tranzacționare al BVB prin Arena Gateway. Cuplarea bidirecțională la aceasta se realizează prin intermediul *agentului de Interfață* care transmite mai departe datele primite *agentului Trader* și *agentului Supervisor*. *Agentul Trader* și *agentul Supervisor* sunt conectați la o bază de date MySQL prin intermediul unui agent

specializat, *agentul DB*. Pentru fiecare emitent pentru acțiunile căruia desfășoară activitatea de Market Maker, *agentul Trader* este conectat la un *agent Reactiv* care primind de la acesta informațiile despre piață, le comunică *agenților Pred* din subordine și decide pe baza analizei pe care o realizează aceștia dacă ordinele introduse pe simbolul respectiv urmează să fie anulate sau nu. Capitolul conține o prezentare detaliată a funcțiilor și modului de lucru al fiecărui agent al sistemului MaS. Este abordată de asemenea structura ierarhică și modul de comunicare al agenților din sistem.

Capitolul conține de asemenea o secțiune dedicată prezentării comportamentului și performanțelor sistemului înregistrate în condiții de volatilitate ridicată, generată de evenimente imprevizibile.

Sistemul include trei agenți specializați în analiza și predicția trendului: unul care folosește algoritmi specifici analizei tehnice, unul care utilizează rețele neuronale și un al treilea care folosește mașinile cu vector suport. La sfârșitul capitolului este expus modul în care sunt instruiți agenții predictivi și sunt date detalii despre datele folosite pentru antrenarea acestora. În secțiunile precedente a fost exemplificată utilizarea a două dintre cele mai performante tehnici de inteligență computațională din punctul de vedere al acurateții predictive: rețelele neuronale și mașinile cu vectori suport. Implementarea acestora s-a bazat pe două tipuri diferite de instrumente software: limbajul MATLAB și pachetul RapidMiner.

Capitolul se încheie cu o scurtă prezentare a beneficiilor ce se pot obține prin combinarea la nivel funcțional a două arhitecturi software complementare, ambele proiectate în limbajul Java: platforma multi-agent JADE și pachetul RapidMiner, ce pune la dispoziție o paletă cvasi-completă de metode data-mining, inclusiv din categoria celor bazate pe inteligență computațională.

## 5. SINTEZA CONCLUZIILOR CERCETĂRII ȘTIINȚIFICE

Rezultatele obținute în implementarea sistemului multi-agent de tranzacționare automată prezentat în capitolul patru al lucrării, demonstrează fără echivoc faptul că arhitecturile bazate pe agenți pot constitui baza realizării de aplicații fiabile într-un domeniu extrem de dinamic așa cum este cel al tranzacțiilor financiare.

Agenții inteligenți au capacitatea de a dobândi cunoștințe prin intermediul proceselor de soluționare a problemelor. Studiul comportamentului social al agenților în știința cognitivă este o parte importantă din domeniul agenților inteligenți.

Agenții software se concentrează pe interacțiuni și colaborări pentru a-și atinge obiectivele într-un context care se schimbă într-o manieră de obicei neprevăzută. Caracteristicile ce sugerează folosirea agenților în sistemele complexe sunt în primul rând adaptabilitatea, autonomia și colaborarea. Autonomia este o proprietate distinctivă a unui agent și presupune capacitatea agentului de a supraviețui într-un mediu în schimbare. Un agent are capacitatea de a percepe factorii de mediu și de a lua decizii cu privire la modul în care trebuie să reacționeze în consecință. Adaptabilitate presupune existența capacităților de învățare necesare agentului pentru adaptarea deciziilor în funcție de experiența din trecut. Colaborarea dintre agenți face ca aceștia să poată fi concepuți pentru a aborda diversele aspecte ale rezolvării unei probleme, fiecare agent fiind conceput în paradigma cea mai potrivită pentru a rezolva partea sa din sarcinile generale ale sistemului. Coordonarea comportamentelor agenților independenți reprezintă și ea o parte centrală proiectării sistemului multi-agent.

Agenții au capacitatea de a procesa eficient datele locale și de a comunica cu alți agenți atunci când este necesar și în cazul în care sarcinile cu care se confruntă sunt dincolo de domeniul lor de cunoștințe. Sisteme multi-agent sunt utilizate într-un spectru larg de aplicații, cum ar fi e-commerce, e-learning, comunicare, data mining, simulare, robotică, sisteme de transport și grid computing.

Produsă de Bursa de Valori București și pusă la dispoziția intermediarilor care acționează pe piața de capital din România, *Arena Gateway* este o aplicație complexă care facilitează transferul de mesaje între sistemul central al bursei și aplicațiile dedicate ale participanților. BVB oferă intermediarilor un sistem format din servere de tranzacționare cuplate prin VPN securizat, bazat pe adrese fixe, la mai multe puncte de acces numite Gateway. Prin socket-uri, un Gateway se conectează la aplicația utilizatorului, negociază conectarea la sistem și asigură fluxul bidirecțional de date necesar tranzacționării. Acesta oferă servicii de cerere/răspuns, servicii bazate pe evenimente, precum și conectivitate. Folosind un sistem de mesaje XML transmise prin rețea, *Arena Gateway* primește comenzi și cereri de la aplicațiile dedicate aparținând participanților, le trimite sistemului central al bursei și furnizează răspunsuri și date din piață solicitantului.

Sistemul multi-agent realizat se conectează la BVB prin *Arena Gateway* permițând astfel transferul bidirecțional de date între agenții specializați și serverele de tranzacționare.

La nivelul agenților, corpul mesajului este interpretat ca un text formatat XML. Structura mesajului XML este descrisă în întregime folosind un fișier schemă XML, furnizat de bursă împreună cu aplicația ce permite conectarea. Fiecare mesaj de ieșire va fi analizat și validat folosindu-se fișierul schemă. În cazul în care se detectează un mesaj care nu respectă constrângerile schemei, *Arena Gateway* trimite un mesaj de eroare către sistemul multi-agent și ignoră mesajul.

Arhitectura sistemului multi-agent implementat are o structură complexă, constituită din mai multe tipuri de agenți ce execută sarcini specifice și care interacționează. Sistemul permite scalarea facilă precum și adăugarea de noi metode de analiză a datelor în timp real. Deși autentificarea agenților în sistem se realizează prin folosirea adreselor de IP ale calculatoarelor gazdă, din motive ce țin de confidențialitatea datelor a fost preferată implementarea sistemului într-o rețea locală.

Cuplarea bidirecțională la *Arena Gateway* se realizează prin intermediul *agentului de Interfață* care transmite mai departe datele primite *agentului Trader* și *agentului Supervizor*. *Agentul Trader* și *agentul Supervizor* sunt conectați la o bază de date MySQL prin intermediul unui agent specializat, *agentul DB*. Pentru fiecare emitent pentru acțiunile căruia desfășoară activitatea de Market Maker, *agentul Trader* este conectat la un *agent Reactiv* care primind de la *Trader* informațiile despre piață, le comunică *agenților Pred* din subordine și decide pe baza analizei acestora dacă ordinele introduse pe simbolul respectiv urmează să fie anulate sau nu.

Interfață grafică cu care este dotat sistemul de agenți permite urmărirea evoluției cotațiilor, modificarea setărilor și comportamentul de ansamblu al sistemului. Plasat de obicei pe stația de lucru a traderului uman, permite acestuia supervizarea facilă a întregului proces. Evenimentele critice sunt semnalate auditiv și prin mesaje transmise în sistem. *Agentul Trader* cunoaște în orice moment starea pieței prin actualizarea permanentă a listelor ce conțin ordinele active din piață pentru emitenții selectați de utilizator. Lista ordinelor se actualizează automat pe baza datelor primite de la *agentul de Interfață*, lista fiind reîncărcată numai atunci când, *agentul de Interfață* semnalează întreruperea comunicației cu serverul de tranzacționare și retransmite datele. Pe baza semnalelor primite

de la agenții Reactivi, *agentul Trader* gestionează operațiunile pentru simbolurile Market Makerului.

Au fost definite trei moduri în care *agentul Trader* acționează ca Market Maker: *Real*, *Undercover* și *Fake*. Pentru fiecare simbol gestionat se poate stabili unul din cele trei moduri de lucru. În modul *Real*, ordinele lansate automat sunt introduse în sistemul BVB, cu toate consecințele privind portofoliul și fondurile bănești ce decurg din această acțiune. Pentru a lucra în modul real, participantul la sistemul de tranzacționare trebuie să ceară și să primească acceptul BVB, ca urmare a semnării unui contract și a asumării unor obligații. În modul real, perechile de ordine sunt introduse în sistem printr-o singură comandă.

În modul *Undercover*, se poate testa capacitatea participantului de a acționa ca Market Maker, fără a se înregistra ca atare la BVB și fără a fi obligat să respecte condițiile impuse de bursă în ceea ce privește spread-ul, timpul în care ordinele sunt active în piață și cantitatea prevăzută în ordin. Acest mod a fost utilizat în faza a doua de testare. Modul de lucru *Fake* a fost folosit intensiv în prima fază de teste pentru că nu implică riscuri financiare și nici nu presupune existența în contul participantului a acțiunilor ce pot fi vândute. Diferența majoră față de primele două moduri de lucru este dată de faptul că în acest caz ordinele nu sunt introduse în piața reală, introducerea lor fiind doar simulată. Sistemul de agenți se comportă la fel ca în cazul ordinelor reale, cu diferența că potențialele tranzacții nu sunt reale, ci simulate de sistem. Astfel, o tranzacție de cumpărare se consideră a fi efectuată dacă nivelul celui mai bun preț de vânzare a ajuns sau a scăzut sub prețul ordinului de cumpărare sau dacă a fost efectuată o tranzacție în piața reală la un preț mai mic sau egal cu cel al ordinului. Analog, un ordin de vânzare se consideră a fi executat dacă s-a efectuat o tranzacție în piața reală la un preț mai mare sau egal cu cel al ordinului sau dacă cel mai bun preț de cumpărare a atins sau depășit nivelul de preț al ordinului de vânzare.

Așa cum am afirmat mai sus, sistemul de agenți proiectat și implementat funcționează independent și nu necesită intervenția operatorului decât atunci când apar disfuncționalități grave hardware (întreruperea comunicațiilor, pene de curent etc.). Totuși, pentru a minimiza riscurile inerente folosirii unei aplicații automate de tranzacționare și pentru a respecta procedurile legale de gestiune a riscurilor, *agentul de Interfață* este dotat cu un sistem complex de afișare a informațiilor și comunicare cu traderul uman a operațiunilor efectuate. Dacă dorește, traderul uman poate invalida deciziile luate de sistem și poate modifica în timp real parametri acestuia pentru ca sistemul să acționeze în sensul dorit. Cu toate acestea, sistemul s-a dovedit a fi foarte stabil, până în acest moment nefiind semnalată niciodată intervenția operatorului.

*Agenții Reactivi* sunt agenți de decizie care primesc în mod continuu prin intermediul *agentului Trader* informații despre operațiunile efectuate în piață și iau decizia anulării ordinelor pentru simbolurile gestionate de sistemul de agenți. Pentru fiecare simbol din listă, *agentul Trader* alocă un *agent Reactiv* și structura subordonată acestuia, formată din *agenții Predictivi*. În momentul de față există trei tipuri de astfel de agenți: *Pred-THN*, care folosește pentru predicție un model specific analizei tehnice, *Pred-NN* care utilizează pentru predicție un model generat cu ajutorul rețelelor neuronale și *Pred-SVM* ce se bazează pe mașini cu vector suport. *Agenții Predictivi* sunt alimentați în timp real cu date din piață de către *agentul Reactiv* în subordinea căruia se află. Modelele pe baza cărora *agenții Predictivi* *Pred-NN* și *Pred-SVM* furnizează predicțiile sunt actualizate la sfârșitul fiecărei săptămâni cu datele ultimei perioade de tranzacționare și sunt încărcate în format XML în sistem atunci când acesta nu este conectat la BVB. O dezvoltare ulterioară a sistemului ar

putea fi realizată prin extinderea numărului de agenți predictivi care să raționeze pe baza altor modele.

În acest moment sistemul funcționează pentru cinci din cei mai importanți emitenți listați la Bursa de Valori București: Fondul Proprietatea S.A., SIF Oltenia S.A., OMV Petrom S.A., C.N.T.E.E. Transelectrica S.A. și Banca Transilvania S.A. Adăugarea sau eliminarea unui emitent se face printr-o simplă selecție din listă, singurul impediment fiind acela că sistemul trebuie să fie deconectat de la serverele de tranzacționare în momentul realizării selecției. Evident, adăugarea unui nou emitent trebuie precedată de antrenarea agenților predictivi pe baza datelor de tranzacționare ale noului emitent. În timpul testelor a fost simulată monitorizarea simultană a zece emitenți, fără ca acest lucru să ducă la scăderea performanțelor sistemului. De altfel, niciunul din intermediarii ce acționează ca Market Maker nu face acest lucru pentru mai mult de cinci emitenți, din cauza condițiilor destul de restrictive pe care trebuie să le îndeplinească în ceea ce privește capacitatea financiară și portofoliul minim pe care trebuie să îl dețină.

Pentru fiecare emitent, ca variabile exogene au fost folosite volumul de tranzacționare și valorile a doi din indicii bursieri reprezentativi pentru emitent, respectiv cei în care ponderea emitentului este cea mai mare (Săndiță și Matei, 2015).

Așa cum am arătat în capitolul patru, instruirea supervizată a agenților are scopul de a-i dota cu capacitatea de realiza predicții la capătul unui proces de instruire supervizată. Instruirea supervizată presupune învățarea din exemple, prin colectarea datelor dintr-o bază de date de instruire, scopul final fiind acela de a obține inductiv un model de predicție validat ce poate fi folosit ulterior de un agent specializat. Tipul modelului de predicție corespunde naturii neliniare, dinamice și nestaționare a proceselor stocastice modelate, natură de altfel intrinsecă seriilor de timp financiare. Prin urmare, am considerat ca fiind potrivite modelele dinamice discrete de tip *NARX*. Fiecare agent este conectat la un set de variabile de intrare. Când valorile acestor variabile se îmbătrânesc, ajungând pe portul său de intrare, agentul se activează și procesează semnalul de intrare prin utilizarea modelului de predicție cu care este înzestrat. Pe baza acestuia, agentul va infera o valoare de ieșire, reprezentând predicția, transferată apoi agentului de la nivelul superior care o va folosi pentru a lua o decizie.

Modelul este livrat *off line*, folosindu-se formatul și sintaxa XML. Agenții citesc fișierul script ce descrie modelul și dispun de un interpretor și de un modul de execuție pentru calculul predicției pe baza modelului și utilizarea acesteia în scopul luării unei decizii. Integrarea MaS cu clase și operatori RAPIDMINER asigură operaționalizarea fazelor de achiziție a datelor și scrierea lor în baza de date MySQL a sistemului, identificarea și estimarea modelului *NARX*, utilizând, după caz, tehnici de învățare supervizată bazate pe rețele neuronale sau pe mașini cu vector suport.

Respectând principiul parcimoniei, modelele obținute trebuie să conțină doar strictul necesar modelării. Pentru a evita supra-ajustarea care apare atunci când o rețea neuronală se specializează excesiv, a fost folosită tehnica opririi timpurii, tehnică ce presupune împărțirea datelor disponibile în trei subseturi: subsetul de instruire, subsetul de validare și subsetul de testare. În cazul rețelelor neuronale, subsetul de instruire este utilizat pentru calcularea gradientului și actualizarea ponderilor rețelei. Subsetul de validare este utilizat doar în procesul de instruire pentru a decide asupra momentului opririi acestuia, prin monitorizarea erorii pentru subsetul de validare în timpul procesului de instruire. Eroarea asociată subsetului de validare scade în mod normal în timpul fazei inițiale de instruire, la fel ca eroarea pentru subsetul de instruire.

Când eroarea pentru subsetul de validare crește pentru un număr specificat de iterații, procesul de instruire este oprit, iar ponderile returnate sunt acelea ce corespund erorii minime de validare. În acest fel se selectează modelul de rețea cu cea mai bună performanță pentru subsetul de validare. Evident, pentru a nu obține rezultate viciate, subsetul de date de test este ales dintr-o mulțime de date de intrare care nu au fost folosite în timpul procesului de antrenament sau validare. Această strategie a fost implementată cu ajutorul aplicației MATLAB ce dispune de funcțiile specifice necesare instrumentării acestei strategii.

În capitolul al doilea al tezei am analizat două din cele mai utilizate platforme pentru agenți: Cougaar și JADE. Cougaar, implementează un model complex de agenți, sarcinile pe care aceștia le pot executa precum și modalitatea de comunicare între entitățile ce activează în sistem. Modelul Cougaar implementează un mecanism specific, foarte puternic de izolare și încapsulare pentru toate componentele. Impedimentele majore care m-a făcut să renunț la folosirea acestei platforme pentru implementarea sistemului de agenți au fost tocmai elementele prezentate drept calitatea esențială a platformei, respectiv supra specializarea în toleranța la erori și generalitatea, elemente ce ar fi complicat nepermis sistemul.

În ceea ce privește platforma JADE, respectarea standardelor FIPA și mecanismul de comunicare între agenți determină reacții destul de lente ale sistemului la evenimentele ce se petrec foarte rapid în piață. Și în acest caz, generalitatea sistemului diminuează performanțele în ceea ce privește viteza de procesare, fapt ce nu poate fi acceptat într-un sistem de tranzacționare automată.

Deși sistemul multi-agent de tranzacționare automată nu a fost realizat pe o platformă standardizată, studiul platformelor orientate către agenți și a performanțelor acestora a determinat implementarea în sistemul realizat a unor mecanisme similare celor instalate pe platforme. Ne referim aici la modul de identificare al agenților printr-un mecanism similar cu White Pages și comunicarea între agenți prin mesaje asincrone. De asemenea, agenții predictivi utilizează fire de execuție diferite pentru a-și îndeplini sarcinile.

Sistemele multi-agent pot beneficia de dezvoltarea și rafinarea tehnicilor de inteligență computațională pentru extragerea inductivă a cunoștințelor din bazele de date, generarea de modele predictive și instanțierea automată a unor agenți predictivi capabili să utilizeze aceste modele pentru a realiza previziuni. Așa cum am încercat să detaliem în cele prezentate anterior, modelul predictiv poate fi generat la capătul unui proces de instruire supervizată, utilizând rețelele neuronale sau mașinile cu vectori suport. RapidMiner este perfect adaptat pentru a îndeplini această activitate. Avantajul său esențial este că modelul obținut prin instruire supervizată poate fi descris și salvat în cod XML. Un agent dotat cu un analizor sintactic poate apoi să interpreteze codul XML și să execute modelul respectiv, îndeplinind astfel o funcție predictivă. O astfel de abordare permite instanțierea automată a unei întregi clase de agenți predictivi în cadrul sistemului multi-agent. Am realizat astfel clase de agenți predictivi ce interpretează și execută modele bazate pe rețelele neuronale și mașinile cu vectori suport. Atât interpretorul modelului în format XML, cât și executivul (modulul de predicție) pot fi proiectate să realizeze sarcini generice. Astfel, un agent predictiv dotat cu un interpretor și un executiv orientat către rețele neuronale va fi capabil să analizeze sintactic orice model neuronal și se efectueze predicții pe baza lui.

În acest mod, integrarea într-o platformă comună a sistemelor JADE și RapidMiner, oferă premiza implementării în sistem și a altor modele de predicție. Un astfel de sistem integrat este flexibil prin natura sa, permițând instanțierea automată a agenților predictivi și utilizarea lor în cadrul unor diverse procese decizionale. Agentul astfel construit își poate schimba funcția predictivă prin simpla schimbare a modelului pe care îl procesează, dat

fiind că modelul reprezintă o funcție exterioară agentului (de tipul *plug-in*, folosit în platforma Cougaar). Evident, schimbarea funcției predictive prin adăugarea unui alt model va trebui însoțită de o eventuală ajustare a formatului structurilor de date ce vor fi transferate pentru procesare noului model.

În momentul de față, sistemul de tranzacționare automată este instalat și funcționează pentru un singur intermediar. Sperăm ca la începutul anului viitor să reușim implementarea lui în alte două locații pentru a-i putea urmări comportamentul când se află în concurență cu el însuși. Dotat încă din versiunile inițiale cu un mecanism de jurnalizare a activității, mecanism ce permite studiul off-line al comportamentului fiecărui agent din sistem, am putea astfel evalua mai bine reacțiile agenților la stimulii externi.

## 6. BIBLIOGRAFIE SELECTIVĂ

- Feng Y., Yu R., Stone P., *Two Stock-Trading Agents: Market Making and Technical Analysis*, Agent Mediated Electronic Commerce V: Designing Mechanisms and Systems, Volume 3048 of Lecture Notes in Artificial Intelligence, Springer Verlag, 2004;
- FIPA website, [www.fipa.org](http://www.fipa.org), 22 iul. 2015;
- Green S., Hurst, L., Nangle, B., Cunningham, D. P., Somers, F., Evans, D. R., *Software agents: A review* (Tech. Rep. No.TCS-CS-1997- 06). Trinity College Dublin, Broadcom Éireann Research Ltd. 1997;
- Helsingier A., Thome M., Wright T., *Cougaar: A Scalable, Distributed Multi-Agent Architecture*, Proc. of the IEEE International Conference on Systems, Man and Cybernetics, The Hague, 2004;
- JADE website, <http://jade.tilab.com>, 22 iul. 2015;
- Kim A.J., Christian R. Shelton C.R., Poggio T., *Modeling Stock Order Flows and Learning Market-Making from Data*, AI Memo 2002-009, 2002;
- Lange D., Oshima M., *Programming and Deploying Java™ Mobile Agents with Aglets™*. Addison-Wesley, 1998;
- Nevmyvaka Y., Sycara K., Seppi D.J., *Electronic Market Making: Initial Investigation*, Artificial Intelligence in Economics and Finance, World Scientific, 2005;
- Nwana H. S., *The potential Benefits of Software Agent Technology to BT*, Internal Technical Report, Poject NOMADS, Intelligent Systems Research, AA&T, BT Labs, UK, 1996;
- Odell J. J., Parunak H., Bernhard Bauer B., *Representing Agent Interaction Protocols in UML*, Volume 1957 of the series Lecture Notes in Computer Science, pp 121-140, 2002;
- Ricordel P.M., Demazeau Y., *From Analysis to Deployment: a Multi-Agent Platform Survey*, ESAW:pp. 93-105, 2000;
- Săndiță A.V., Durac C., *Market Maker și Sisteme Multi-Agent în cadrul Bursei de Valori București*, Finanțe, provocările viitorului, nr. 17, Universitatea din Craiova, 2015;
- Săndiță A.V., Matei Gh., *Tranzacționare algoritmică la Bursa de Valori București*, ANNALS of the University of Petrosani, ECONOMICS, vol. XV, 2015;



- Stone P., Sherstov A.A., *Three Automated Stock-Trading Agents: A Comparative Study*, Agent-Mediated Electronic Commerce VI, Theories for and Engineering of Distributed Mechanisms and Systems, Volume 3435 of the series Lecture Notes in Computer Science, pp 173-187, 2005;
- Vapnik, V., Chervonenkis, A., *On the uniform convergence of relative frequencies of events to their probabilities*. Theory of Probability and its Applications, 16(2):264-280, 1971;
- Wooldrige M., *Agent – based software engineering*, IEE Transactions of Software engineering, February, 144(1), 26 – 37, 1997;
- Yom-Tov, E., *A distributed sequential solver for large-scale SVMs*, L. Bottou, O. Chapelle, D. DeCosta, and J. Weston, editors, Large-Scale Kernel Machines, pp.139–154, Cambridge: MIT Press, 2007.