



ROMÂNIA
Universitatea „Dunărea de Jos” din Galați



MINISTERUL
EDUCAȚIEI
NAȚIONALE

C2628 / 9.04.2014

C ă t r e

Universitatea „Dunărea de Jos” din Galați vă face cunoscut că, în data de 18.06.2014, ora 9,30 în Sala Consiliului de Administrație – fosta Sală a Senatului, va avea loc susținerea publică a tezei de doctorat intitulată: **”CONTRIBUȚII LA PROBLEMATICA ACHIZIȚIEI DE CUNOȘTINȚE PENTRU AGENȚI INTELIGENȚI”**, elaborată de doamna/domnul **LAZĂR VERONICA (JÂȘCANU)**, în vederea conferirii titlului științific de doctor în domeniul de doctorat **Știința calculatoarelor**.

Comisia de doctorat are următoarea componență :

- | | |
|----------------------------------|---|
| 1. Președinte | Conf.univ.dr.ing. Emilia PECHEANU
Universitatea „Dunărea de Jos” din Galați |
| 2. Conducător de doctorat | Prof.univ.dr.ing.Luminița DUMITRIU
Universitatea „Dunărea de Jos” din Galați |
| 3. Referent oficial | Prof.univ.dr.ing. Ștefan TRĂUȘAN-MATU
Universitatea „Politehnica” din București |
| 4. Referent oficial | Prof.univ.dr.ing. Costin BĂDICĂ
Universitatea din Craiova |
| 5. Referent oficial | Conf.univ.dr.ing. Cornelia TUDORIE
Universitatea „Dunărea de Jos” din Galați |

Cu această ocazie vă transmitem rezumatul tezei de doctorat, și vă invităm să participați la susținerea publică. În cazul în care doriți să faceți eventuale aprecieri sau observații asupra conținutului lucrării, vă rugăm să le transmiteți în scris pe adresa universității, str. Domnească nr. 47, 800008 Galați, Fax 0236 / 461353, e-mail rectorat@ugal.ro.



Rector,

Prof. univ. dr. ing. **Iulian Gabriel BÎRSAN**

**Universitatea „Dunărea de Jos” din Galați
Școala doctorală de Inginerie**



REZUMAT TEZĂ DE DOCTORAT

CONTRIBUȚII LA PROBLEMATICA ACHIZIȚIEI DE CUNOȘTINȚE PENTRU AGENȚI INTELIGENȚI

**Doctorand
LAZĂR VERONICA (JÂȘCANU)**

Conducător științific,

Prof. Univ. Dr. Ing. Dumitriu Luminița

Referenți științifici

Prof. Univ. Dr. Ing. Trăușan-Matu Ștefan

Prof. Univ. Dr. Ing. Bădică Costin

Conf. Univ. Dr. Ing. Tudorie Cornelia

Seria I2: Calculatoare și tehnologia informației Nr. 2

**GALAȚI
2014**

Seriile tezelor de doctorat susținute public în UDJG începând cu 1 octombrie 2013 sunt:

Domeniul **ȘTIINȚE INGINEREȘTI**

Seria I 1: **Biotehnologii**

Seria I 2: **Calculatoare și tehnologia informației**

Seria I 3: **Inginerie electrică**

Seria I 4: **Inginerie industrială**

Seria I 5: **Ingineria materialelor**

Seria I 6: **Inginerie mecanică**

Seria I 7: **Ingineria produselor alimentare**

Seria I 8: **Ingineria sistemelor**

Domeniul **ȘTIINȚE ECONOMICE**

Seria E 1: **Economie**

Seria E 2: **Management**

Domeniul **ȘTIINȚE UMANISTE**

Seria U 1: **Filologie- Engleză**

Seria U 2: **Filologie- Română**

Seria U 3: **Istorie**

Respectul și mulțumirile mele se îndreaptă către toți cei care, prin încurajări, sugestii sau sfaturi, au contribuit la realizarea acestui demers științific.

Îi port deosebită recunoștință domnului Prof. Univ. Dr. Ing. Bumbaru Severin, distins OM, pentru încrederea acordată, pentru insistențele, discuțiile purtate și mai ales răbdarea de care a dat dovadă în decursul timpului, pentru îndrumarea de specialitate în planificarea tezei, susținerea examenelor și referatelor.

De asemeni, adresez sincere mulțumiri doamnei Prof. Univ. Dr. Ing. Dumitriu Luminița, în calitate de conducător științific al lucrării și îndrumător moral, pentru încurajarea și sprijinul real oferit.

Doresc să mulțumesc, în egală măsură, tuturor colegilor mei din Departamentul de Calculatoare și Tehnologia Informației, al Facultății de Automatică, Calculatoare, Inginerie Electrică și Electronică, pentru observațiile constructive și criticile bine venite, primite pe tot parcursul desfășurării stagiului de doctorat.

Mulțumesc echipei de dezvoltare și implementare a modulului de argumentare din cadrul proiectului European TWIRL, pentru colaborarea apropiată și șansa oferită de a participa ca și consultant în analiza și dezvoltarea componentei comerciale.

Țin să mulțumesc în mod deosebit familiei pentru înțelegerea și încurajarea fără de care această teză nu ar fi fost posibilă.

Cuprins

	în rezumat	în teză
1. Stadiul actual	1	1
1.1. Achiziția de cunoștințe	1	1
1.1.1. Procesarea textului	---	5
1.1.2. Identificarea opiniilor și a sentimentelor	---	6
1.1.3. Procesarea imaginilor și a secvențelor video	---	6
1.1.4. Analiza rețelelor sociale	---	7
1.1.5. Rețele semantice	---	9
1.1.6. Sisteme multi-agent	---	10
1.2. Argumentare	2	11
1.2.1. Argumentare în sisteme multi-agent	2	13
1.2.2. Concepte de bază ale argumentării	2	14
1.2.3. Argumentarea logică	3	16
1.2.4. Argumentarea practică	---	18
1.2.5. Cadrul abstract propus de Dung	4	19
1.2.6. Sistemul de argumentare bipolar propus de Cayrol	5	21
2. Contribuții privind utilizarea argumentării pentru achiziția de cunoștințe în sisteme bazate pe agenți inteligenți	7	25
2.1. Conceperea unei propuneri de proiect	8	28
2.2. Crearea unei baze de cunoștințe pentru colectarea cunoașterii de la experți	9	30
2.3. Identificarea surselor de cunoaștere care vor fi consultate	10	32
2.4. Utilizarea de metode eficiente pentru capturarea, structurarea și validarea cunoașterii	11	34
2.5. Desfășurarea etapei de achiziție astfel încât să fie maximizată utilizarea resurselor disponibile	15	42
2.6. Diseminarea bazei de cunoștințe către entitățile interesate	15	42
2.7. Concluzii	---	45
2.7.1. Contribuții	---	46
2.7.2. Direcții viitoare de cercetare	---	46
3. Contribuții privind reprezentarea cunoștințelor în ARGKAMAS	18	48
3.1. Reprezentarea cunoștințelor în sistemul ARGKAMAS	18	48
3.1.1. Reprezentarea cunoștințelor agentului AgSB	18	49
3.1.2. Reprezentarea cunoștințelor agentului Ag	19	53
3.2. Reprezentarea influențelor dintre cunoștințe	21	56
3.2.1. Reprezentarea influențelor agentului AgCIB	22	57
3.2.2. Reprezentarea influențelor interne ale agentului Ag	23	60
3.3. Achiziția de cunoștințe noi	24	62
3.3.1. Selecția topicului și declanșarea dezbaterii	24	62
3.3.2. Participarea la dezbateri	25	65
3.3.3. Determinarea gradului de interes <i>DegInt</i>	26	66
3.3.4. Determinarea gradului de încredere <i>DegConf</i>	27	68
3.4. Paralelă cu achiziția clasică	30	74
3.5. Concluzii	---	80
3.5.1. Contribuții	---	80
3.5.2. Direcții viitoare de cercetare	---	81
4. Contribuții privind modelarea agenților inteligenți argumentativi în vederea achiziției de cunoștințe	31	82
4.1. Comportamentul la nivel de sistem	31	82
4.2. Modelarea agenților	34	87
4.2.1. Agentul bază de date semantică AgSB	34	87
4.2.2. Agentul bază de date de influențe comune AgCIB	---	95
4.2.3. Agentul comun Ag	---	99
4.2.4. Agentul registru de dezbateri AgDTR	---	107

4.2.5. Agentul arbore de dezbatere <i>AgDT</i>	---	110
4.3. Concluzii	---	114
4.3.1. Contribuții	---	114
4.3.2. Direcții viitoare de cercetare	---	114
5. Contribuții experimentale și aplicative	42	115
5.1. Comparație cu modelul propus de Cayrol	42	117
5.2. Rezultate experimentale	43	119
5.3. Concluzii	45	126
5.3.1. Contribuții	---	127
5.3.2. Direcții viitoare de cercetare	---	127
6. Concluzii, contribuții și direcții viitoare de cercetare	46	128
6.1. Contribuții	47	130
6.2. Direcții viitoare de cercetare	48	131
Listă lucrări publicate și prezentate	50	133
Bibliografie selectivă	53	137
Anexe	55	144
Anexa 1	55	144
Anexa 2	56	145

Contents

	în rezumat	în teză
1. State of the Art	1	1
1.1. Knowledge acquisition	1	1
1.1.1. Text mining	---	5
1.1.2. Opinion and sentiment analysis	---	6
1.1.3. Image and video processing	---	6
1.1.4. Social Networks Analysis	---	7
1.1.5. Semantic Web	---	9
1.1.6. Multi-agent systems	---	10
1.2. Argumentation	2	11
1.2.1. Argumentation in multi-agent systems	2	13
1.2.2. Basic concepts for argumentation	2	14
1.2.3. Logical argumentation	3	16
1.2.4. Practical argumentation	---	18
1.2.5. Dung's abstract argumentation framework	4	19
1.2.6. Cayrol's bipolar argumentation framework	5	21
2. Contributions concerning the use of argumentation for knowledge acquisition in multi-agent systems	7	25
2.1. Designing a project proposal	8	28
2.2. Creating a knowledge base for collecting knowledge from experts	9	30
2.3. Identifying the knowledge sources that will be consulted	10	32
2.4. Using effective methods for capturing, structuring and validating knowledge	11	34
2.5. Conducting the acquisition process so as to maximize the use of available resources	15	42
2.6. Disseminating the knowledge base to interested entities	15	42
2.7. Conclusions	---	45
2.7.1. Contributions	---	46
2.7.2. Future research directions	---	46
3. Contributions concerning the knowledge representation in ARGKAMAS	18	48
3.1. Knowledge representation in ARGKAMAS system	18	48
3.1.1. Agent <i>AgSB</i> knowledge representation	18	49
3.1.2. Agent <i>Ag</i> knowledge representation	19	53
3.2. Representation of influences	21	56
3.2.1. Agent <i>AgCIB</i> influences representation	22	57
3.2.2. Agent <i>Ag</i> internal influences representation	23	60
3.3. Acquisition of new knowledges	24	62
3.3.1. Topic selection and debate starting	24	62
3.3.2. Participation at the debate	25	65
3.3.3. Degree of interest <i>DegInt</i> determination	26	66
3.3.4. Degree of confidence <i>DegConf</i> determination	27	68
3.4. Comparison with classical knowledge acquisition	30	74
3.5. Conclusions	---	80
3.5.1. Contributions	---	80
3.5.2. Future research directions	---	81
4. Contribution concerning the modeling of argumentative intelligent agents for knowledge acquisition	31	82
4.1. System-level behavior	31	82
4.2. Modeling agents	34	87
4.2.1. Semantic base agent <i>AgSB</i>	34	87
4.2.2. Common influences base <i>AgCIB</i>	---	95
4.2.3. Common agent <i>Ag</i>	---	99
4.2.4. Debates registry agent <i>AgDTR</i>	---	107
4.2.5. Debate tree agent <i>AgDT</i>	---	110

4.3. Conclusions	---	114
4.3.1. Contributions	---	114
4.3.2. Future research directions	---	114
5. Experimental and applicative contributions	42	115
5.1. Comparision with Cayrol's model	42	117
5.2. Experimental results	43	119
5.3. Conclusions	45	126
5.3.1. Contributions	---	127
5.3.2. Future research directions	---	127
6. Conclusions, contributions and future research directions	46	128
6.1. Contributions	47	130
6.2. Future research directions	48	131
List of published and presented papers	50	133
Selective Bibliography	53	137
Annexes	55	144
Annex 1	55	144
Annex 2	56	145

Introducere

„Problema cea mai dificilă în dezvoltarea oricărui sistem bazat pe cunoștințe este extinderea bazei de cunoștințe prin achiziția de noi cunoștințe sau prin învățare. În inteligența artificială se face distincție între achiziție – introducerea de noi cunoștințe în baza de cunoștințe – și învățare de exemplu, deducerea unor noi cunoștințe din o serie de cunoștințe existente, prin generalizare.”

(Trăușan-Matu Ș., Inteligența Artificială)

Societatea umană este un sistem complex. Cunoașterea umană este o creație a societății, iar la rândul ei, aceasta creează societatea. Dezvoltarea cunoașterii unui om are loc prin interacțiuni sociale, încă de la cea mai fragedă vârstă. Interacțiunile sociale au multiple dimensiuni, o componentă esențială fiind argumentarea. Prin intermediul argumentării, oamenii își consolidează și transmit cunoașterea. Argumentarea este un instrument esențial pentru achiziția de cunoștințe, indiferent de tipul entităților implicate.

Ideea centrală a tezei constă în aplicarea de mecanisme specifice de argumentare agenților inteligenți, în scopul achiziției de noi cunoștințe. Clasic, problema achiziției de cunoștințe constă în formalizarea cunoașterii expertului și transpunerea acesteia într-un sistem informatic. Istoric vorbind, au fost specificate și implementate numeroase sisteme și metodologii care permit achiziția de cunoaștere de la un expert uman. Achiziția de cunoștințe este o primă etapă a dezvoltării unui sistem informatic inteligent. În cadrul sistemelor multi-agent, agenții dispun de cunoaștere definită apriori sau obținută prin metode clasice de knowledge discovery pe seturi de date disponibile sistemului.

Teza identifică problematica dificultății extinderii bazei de cunoștințe prin metodele clasice și propune o abordare bazată pe argumentație, abordare care permite dezvoltarea continuă a acesteia fără intervenția unui expert uman.

Tematica abordată de către teză este una de mare actualitate în contextul actual al dezvoltării internetului. IOT – *Internet of Things* – sau internetul dispozitivelor, și prelucrarea cantităților mari de date (BigData), necesită mai mult ca oricând existența de metode de achiziție automată de cunoaștere. Modelarea sistemelor se realizează folosind arhitecturi distribuite, iar noțiunea de agent și sistem multi-agent a devenit din ce în ce mai prezentă în cadrul platformelor software. Descrierea semantică a informației a ajuns la un nivel la care se pot implementa asistenți personali care pot oferi răspunsuri contextuale și pot interacționa cu agentul uman într-un mod natural. Teza transpune aspecte ale argumentării într-un formalism care permite agenților să-și îmbogățească cunoașterea în mod autonom, prin intermediul dezbaterilor. Sistemul multi-agent propus este compatibil cu arhitecturile moderne de aplicații distribuite și cu reprezentările semantice ale informațiilor.

Teza introduce o perspectivă novatoare asupra dezvoltării cognitive a agentului. În cadrul tezei este definit un mecanism de achiziție de cunoaștere prin argumentare, inspirat din mecanismul interacțiunilor din cadrul societății umane. Au fost studiate și integrate în sistemul multi-agent aspectele care influențează declanșarea și participarea la o dezbateri. De asemenea, au fost elaborați algoritmi necesari pentru calculul *DegInt* – Gradul de Interes și *DegConf* - Gradul de Încredere referitor la un argument. Parametrii definiți pentru un argument permit agentului participarea la noi dezbateri, fiind posibilă astfel o actualizare permanentă a bazei de cunoștințe proprii agentului. Sunt specificați un număr de patru agenți specializați: *AgSB* – Agentul Baza de Date Semantică, *AgCIB* – Agentul Baza de Date de Influențe Comune, *AgDT* – Agentul Arbore de Dezbateri, *AgDTR* – Agentul Registru de Dezbateri și un al cincilea agent, agentul comun *Ag*. Fiecare agent este conceput ca un sistem multi-agent. Sunt descrise mecanismele prin care o cunoaștere similară a mai multor agenți devine cunoaștere comună, fiind inclusă în baza de date de influențe comune a *AgCIB*. Sistemul multi-agent propus în teză

împreună cu interacțiunile și algoritmi specifici a fost numit *ARGKAMAS – Argumentation based Knowledge Acquisition in Multi-Agent Systems*.

La acest moment, sistemul propus în teză, reprezintă o abordare unică a achiziției de cunoștințe prin argumentare în sisteme multi-agent.

A doua contribuție esențială a tezei constă în corelarea algoritmului de argumentare cu etape specifice de achiziție clasică. Astfel, agenții devin pe rând experți și ingineri de cunoștințe, interacțiunile fiind asemănătoare cu cele din lumea reală. Definirea unor etape ale procesului de achiziție permite dezvoltări ulterioare ale algoritmilor numai pentru acestea, fără a afecta celelalte etape.

În cadrul tezei, argumentarea se realizează exclusiv pentru achiziția de cunoștințe. Se realizează o delimitare clară de alte mecanisme de argumentare care au ca scop învățarea, negocierea, etc. Agentul nu aplică nici un alt fel de algoritm specific de knowledge discovery pentru filtrare, generare de reguli de asociere, identificare de clustere, etc. Cunoașterea achiziționată poate fi folosită în cadrul altor argumentări în forma în care a fost dobândită. Analiza și interpretarea noilor cunoștințe prin metode de knowledge discovery rămâne ca direcție viitoare de cercetare.

În cadrul tezei, este definită o metodologie și un algoritm specific prin care un agent se poate documenta în mod automat prin interacțiuni sociale bazate pe argumentare.

Să considerăm studiul de caz următor:

Agentul primește o cerere de documentare despre limbajul de programare *Python*. În urma analizei cererii primite, agentul selectează diferite teme de dezbatere despre limbajul *Python*. Printre temele alese poate fi spre exemplu, popularitatea limbajului *Python*. Agenții din sistemul *ARGKAMAS* decid dacă participă la dezbatere în funcție de domeniile personale de interes și de cunoștințele pe care le dețin. Analiza arborelui de argumentare permite atât agentului care a inițiat dezbaterea cât și celor care au participat să-și modifice interesul și încrederea în cunoștințele proprii. Astfel, agentul care a inițiat dezbaterea (inginerul de cunoștințe) va afla că *Popularitatea limbajului Python NU este Scăzută* și despre această cunoaștere (credință) are un interes ridicat și o încredere mare. Această cunoaștere nu exista anterior argumentării în baza de cunoștințe a agentului, deci *se poate afirma că aceasta a fost achiziționată de către agent în mod automat*. Cunoașterea a fost asimilată în urma unui proces de justificare și argumentare la care au participat numeroși agenți din diverse domenii și cu diverse interese. Cu această cunoaștere nou dobândită, agentul poate participa la rândul său la alte dezbateri.

Datorită faptului că platforma descrisă în teză este una de tip multi-agent, agenții sunt descriși în termeni de Credințe, Dorințe și Intenții (BDI – Beliefs, Desires, Intentions). Astfel, o piesă de cunoaștere, în vocabularul specific sistemelor multi-agent reprezintă o credință a agentului. Etapele specifice achiziției clasice de cunoștințe sunt transpuse în etapele algoritmului de argumentare, iar pașii specifici pentru argumentare sunt translați în credințe, dorințe și intenții între agenții sistemului. Această abordare, permite în final o implementare practică a platformei *ARGKAMAS*.

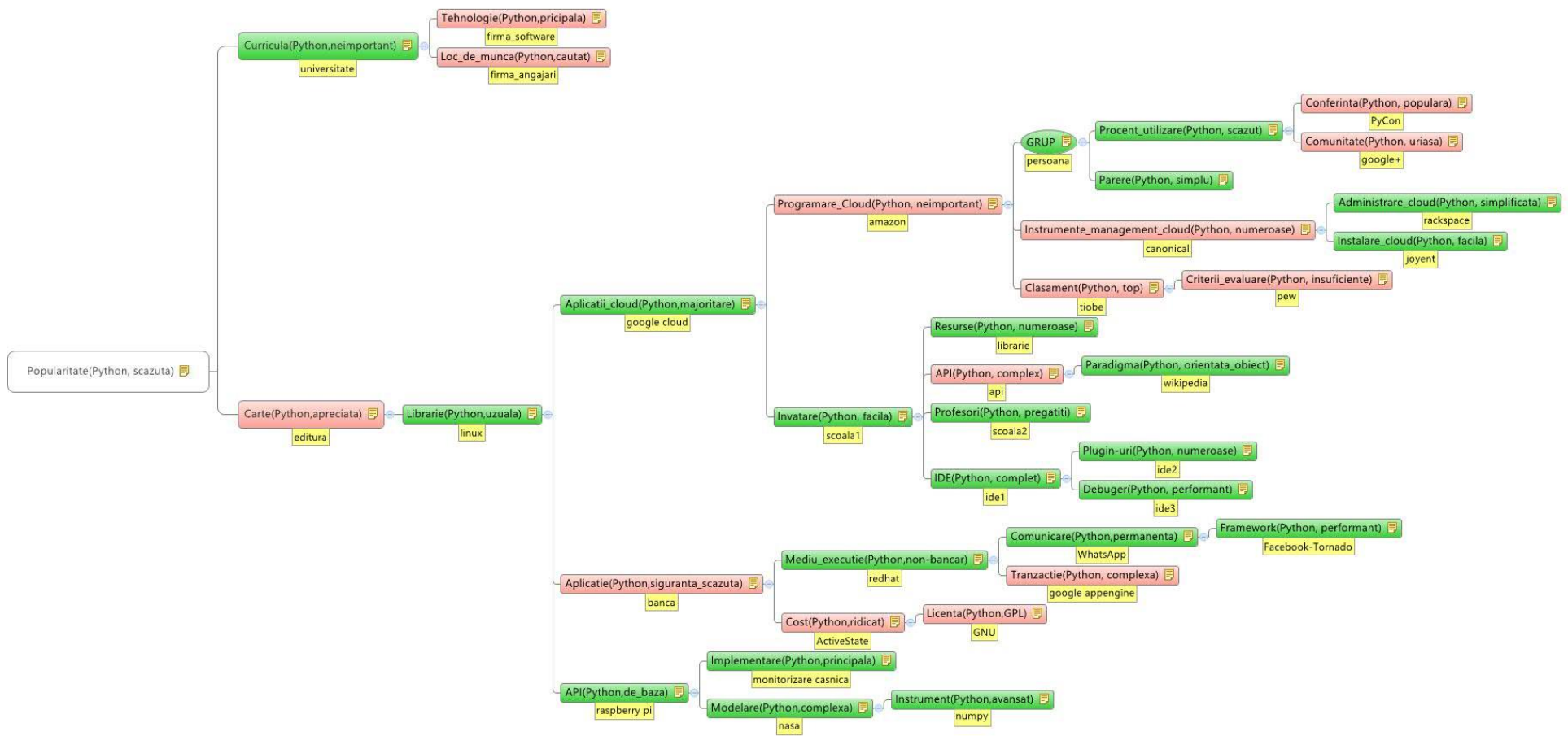
Teza a fost dezvoltată având în permanență în vedere sisteme reale funcționale. Astfel, modelul bazei de cunoștințe folosește și extinde modelul semantic implementat de către *Freebase.com*, componentă a *Google Knowledge Graph*. Sistemul propus de *Freebase* este în primul rând unul practic, dovadă fiind baza de cunoaștere care conține la acest moment un număr de 43 de milioane de subiecte cu peste 2,5 miliarde de conexiuni între ele. Modelul de reprezentare a cunoașterii este folosit ca parte a credinței unui agent comun *Ag*.

De asemenea, în ultimele 18 luni, a existat o colaborare foarte apropiată cu echipa de dezvoltare din cadrul proiectului European ITEA2 TWIRL în scopul dezvoltării modulului de argumentare, modul specific proiectului. Proiectul TWIRL are o durată de 24 de luni, urmând

a se finaliza la sfârșitul acestui an. TWIRL (Twinning virtual World on-line Information with Real world off-Line data source) își propune crearea unei platforme deschise care să permită achiziționarea, procesarea, îmbogățirea și interconectarea datelor provenite din diverse surse de informații. Consorțiul românesc participă în cadrul proiectului cu o platformă educațională de tip second-screen, în cadrul căreia componenta de dezbateri este una foarte importantă. Chiar dacă algoritmi pentru argumentare implementați în cadrul proiectului sunt mult mai simpli decât cei dezvoltați în cadrul tezei, a fost avută în permanență în vedere posibilitatea integrării rezultatelor tezei în platforma TWIRL, astfel ca rezultatele teoretice să producă efecte economice.

Teza propune un sistem care reprezintă o nișă de cercetare pentru domeniul argumentării în sistemele multi-agent. Sistemul este unul cu un grad ridicat de complexitate și fiecare componentă, algoritm și etapă poate fi optimizată sau descrisă teoretic cu diverse teorii din domeniul inteligenței artificiale. Teza demonstrează că sistemul propus este unul fezabil și de perspectivă.

În figura următoare este prezentat un arbore de argumentare despre popularitatea limbajului *Python*, dezbateri la care au participat un număr de 36 de agenți.



Arborele de argumentare pentru aflarea popularității limbajului *Python*

Structura tezei

În cadrul *capitolului 1* este descris stadiul actual al cercetărilor în domeniul achiziției de cunoștințe, cu o scurtă trecere în revistă a metodelor de procesare a textului, a opiniilor și sentimentelor, procesarea imaginilor și a secvențelor video, analiza rețelelor sociale, rețele semantice și a diverselor roluri ale agenților inteligenți în sisteme multi-agent. De asemenea, în *capitolul 1*, este descrisă teoria argumentării în sisteme multi-agent, concepte de bază ale argumentării și componentele esențiale ale acesteia: argumentul și contra-argumentul, precum și teoriile propuse în argumentarea logică și practică, argumentarea abstractă și argumentarea bipolară.

În cadrul *capitolului 2* sunt descrise contribuțiile privind utilizarea argumentării pentru agenți inteligenți prin etape specifice procesului de achiziție de cunoștințe și acțiunile pe care agenții le realizează în fiecare dintre ele, corelate cu momente din algoritmul de dezbatere, dezbatere ce are loc între agenții implicați. Cele șase etape de achiziție de cunoștințe sunt: conceperea unei propuneri de proiect, crearea unei baze de cunoștințe pentru colectarea cunoașterii de la experți, identificarea surselor de cunoaștere care vor fi consultate, utilizarea de metode eficiente pentru capturarea, structurarea și validarea cunoașterii, desfășurarea etapei de achiziție astfel încât să fie maximizată utilizarea resurselor disponibile, diseminarea bazei de cunoștințe către entitățile interesate.

Capitolul 3 descrie contribuții privind achiziția de cunoștințe prin argumentare, prin detalierea reprezentării cunoștințelor agenților *AgSB* – Agentul Baza de Date Semantică și *Ag* – Agentul Comun, în sistemul propus în teză - *ARGKAMAS*, detalierea reprezentării influențelor dintre cunoștințe prin reprezentarea influențelor agentului *AgCIB* – Agentul Baza de Date de Influențe Comune și prin reprezentarea influențelor interne ale agentului comun *Ag*, achiziția de noi cunoștințe prin selecția topicului și declanșarea dezbaterii, participarea la dezbatere și la finalizarea ei, determinarea gradelor *DegInt* – Gradul de Interes și *DegConf* – Gradul de Încredere, necesare pentru transformarea nodului rădăcină în credință. În cadrul aceluiași *capitol 3* este realizată și o paralelă cu achiziția clasică între diversele problematici specifice fiecărei etape și algoritmi implementați.

Capitolul 4 cuprinde contribuții privind agenții inteligenți argumentativi, comportamentul la nivel de sistem a sistemului *ARGKAMAS*, ce este organizat pe trei niveluri de complexitate: nivelul superior alcătuit din agenții principali, nivelul intermediar ce descrie sistemul multi-agent care alcătuieste fiecare agent principal și nivelul inferior care conține agenți *unități funcționale* și *unități de legătură*. De asemenea, în cadrul *capitolului 4* este prezentată modelarea agenților principali propuși – *AgSB* – Agentul Baza de Date Semantică, *AgCIB* – Agentul Baza de Date de Influențe Comune, *Ag* – Agentul Comun, *AgDTR* – Agentul Registru de Dezbateri și *AgDT* – Agentul Arbore de Dezbatere.

Capitolul 5 descrie contribuții experimentale și aplicative, ce au avut ca scop validarea algoritmului de calcul pentru determinarea gradului de încredere *DegConf*.

Capitolul 6 cuprinde concluzii generale, enumeră contribuțiile originale și direcțiile viitoare de cercetare.

Introduction

„The most difficult problem in the development of any knowledge-based system is expanding the knowledge base through the acquisition of new knowledge or learning. In artificial intelligence, there is a difference between acquisition - the introduction of new knowledge in the knowledge base - and learning for example, the deduction of new knowledge from a range of existing knowledge, through generalization”

(Trăușan-Matu Ș., Inteligența Artificială)

The human society is a complex system. The knowledge is a creation of the society, and in turn, it creates the society. A man's knowledge development occurs through social interaction, since their infancy. Social interactions have multiple dimensions and argumentation is an essential piece from the puzzle. Through argumentation, people reinforce and convey knowledge. Argumentation is an essential tool for knowledge acquisition, regardless of the type of entities involved.

The thesis main idea consists in applying specific argumentation mechanisms to intelligent agents in order to acquire new knowledge. Classically, the knowledge acquisition problem comes from the knowledge formalization and implementation in a computer system. Historically, there have been specified and implemented many systems and formalisms that enable the acquisition of knowledge from an expert. The acquisition of knowledge is the first step in the development of an expert system. In multi-agent systems, the agency already has prior knowledge extracted with classical methods of knowledge discovery on large data sets.

The thesis identifies the difficulties of extending the knowledge base through classical methods and propose a novel approach based on argumentation, approach that allows self-evolvement of agents without human expert intervention.

The thesis approach is very actual in the context of the Internet development. The domains of *IOT*-Internet of Things, or Internet of devices, and the domain of big data processing, requires more than ever the existence of methods for automatic acquisition of knowledge. The systems are implemented using distributed architectures, and the notion of agent and multi-agent system became increasingly present in the software environments. The semantic web has reached a level where personal assistants can provide contextual responses and may interact with the human agent in a natural way. The thesis propose a formalism that allows agents to enrich their own knowledge through the debates, in an autonomous way. The proposed multi-agent system is compatible with modern distributed application architectures and with the semantic representation of the information.

The thesis introduces a new perspective on the cognitive development of an agent. It defines a mechanism for knowledge acquisition through argumentation, inspired by the interaction mechanisms within the human society. The starting and the participation to a debate have been studied and integrated into the multi-agent system formalism. In addition, the algorithms for calculating *DegInt*-the Degree of Interest and *DegConf*-the Degree of Confidence with respect to an argument, have been developed. The specified parameters for an argument allow the agent participation in new discussions and a continuous update of its own knowledge base. There have been specified a number of five agents: *AgSB*-Semantic Database Agent, *AgCIB*- Common Influences Database Agent, *Ag*- Regular Agent, *AgDT*- Debate Tree Agent and *AgDTR*-Debates Register Agent. At its own turn, each agent is designed as a multi-agent system. It is also described the mechanisms by which a similar knowledge of several agents becomes common knowledge that it is included into *AgCIB*. The multi-agent system proposed by the thesis, together with specific algorithms and interactions is called *ARGKAMAS* - *Argumentation based Knowledge Acquisition in Multi-Agent Systems*.

At this point, the system proposed in this thesis represents a unique approach to the acquisition of knowledge through argumentation in multi-agent systems.

The second essential contribution of the thesis lies in the linking of argumentation algorithm with classical knowledge acquisition stages. Thus, the agents become experts and knowledge engineers, the interactions between them being similar to those in the real world. The stages allow further developments of algorithms only for a specific stage, without affecting the other stages.

In the thesis, the argumentation is performed solely for the purpose of knowledge acquisition. There is a clear delimitation from other mechanisms of argumentation for learning, negotiation, etc. The agent does not use any other kind of specific knowledge discovery algorithm for filtering, generation of association rules, identification of clusters, etc. The knowledge acquired could be used within other debates in the form in which it was acquired. The analysis and interpretation of new knowledge through knowledge discovery methods remain as a future research direction.

The thesis defines a methodology and a specific algorithm for automatic documenting by social interactions based on argumentation.

Let us consider the following case study:

The agent receives a request for a documentation about the Python programming language. From the analysis of the received request, the agent selects several debate topics about the Python language. Among the topics chosen may be, for example, the popularity of the Python language. Each agent of the ARGKAMAS system decides if it participates at the debate according to personal interests and areas of expertise. The analysis of the argument tree allows agents to revise their interest and confidence in their own knowledge. Thus, the agent who initiated the debate, the knowledge engineer, will find out that the popularity of the language *Python is NOT low*, and on this knowledge, it has a high interest and great confidence. This knowledge did not exist before in the knowledge base, so we can sustain that the agent acquired it automatically. The knowledge was acquired following a process of justification and argumentation that was attended by numerous agents from various fields and with varied interests. With this newly acquired knowledge, the agent may participate in turn to other debates.

Because the platform described in this thesis is a multi-agent one, the agents are described in terms of Beliefs, Desires and Intentions (BDI). Thus, a piece of knowledge in the specific vocabulary of multi-agent systems represent a belief of the agent. Each stage of knowledge acquisition has a specific argumentation algorithm that corresponds to agent's beliefs, desires and intentions. This approach allows, in the end, a practical implementation of the ARGKAMAS platform.

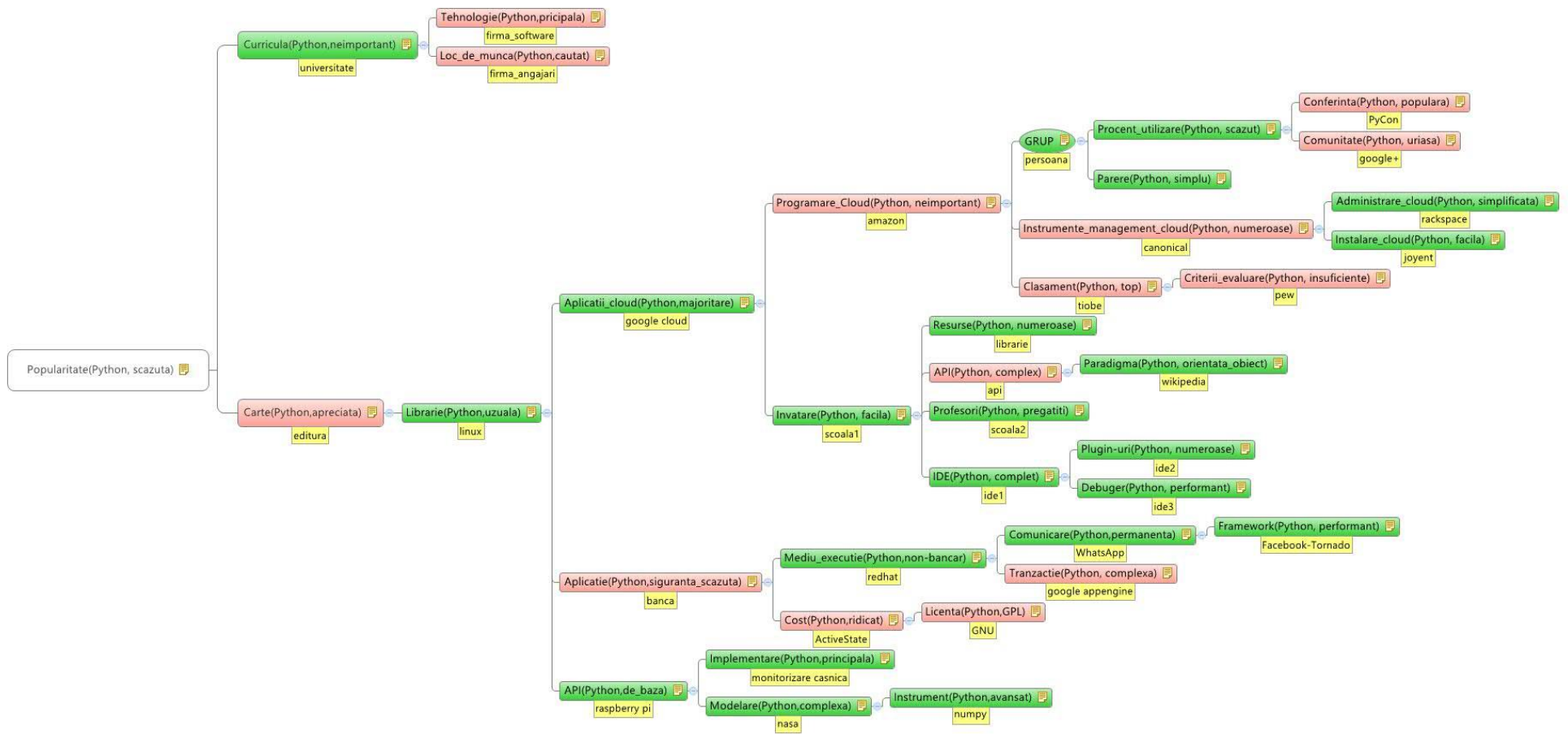
The thesis was developed taking into account real systems. Thus, the model uses the knowledge base and extend the semantic model implemented by Freebase.com, which is now a component of Google Knowledge Graph. The proposed system of Freebase is a practical one, the proof being the knowledge base that contains at the moment, a number of 43 million topics with over 2.5 billion connections between them. The knowledge representation model is part of the agent's belief.

In addition, in the last 18 months, there has been a very close collaboration with the development team of the European project ITEA2 TWIRL, to develop the argumentation module. TWIRL project has a duration of 24 months, ending this year. TWIRL (Twinning virtual World on-line Information with Real world off-line data source) aims to create an open platform that enables the acquisition, processing, enrichment and interconnection of data from various sources of information. The Romanian consortium participates in the project with a second-screen educational platform, in which the debate component is very important. Even though the algorithms for argumentation developed within the project are much simpler, the thesis results are considered for an integration with TWIRL platform, so that the theoretical results to produce economic effects.

The thesis proposes a system that represents a research niche for argumentation in multi-agent systems. The system is one with a high degree of complexity and each component,

being algorithm or knowledge acquisition stage, could be optimized or described with various theories in the field of artificial intelligence. The thesis demonstrates that the proposed system is feasible and a forward-looking one.

The following figure shows a debate tree on the popularity of the *Python* language, with 36 participant agents.



The argument tree for the popularity of the *Python* language

Thesis structure

Chapter 1 describes the current state of research in the field of acquisition of knowledge with a short review of the methods of text, opinion and feeling, image and video processing, social network analysis, semantic networks and the various roles of intelligent agents in multi-agent systems. In addition, it describes the argumentation theory in multi-agent systems, basic concepts of argumentation and its essential components: the argument and counter argument, and the theories proposed in practical and logical argumentation, abstract and bipolar argumentation.

Chapter 2 describes the contributions on the use of argumentation for intelligent agents through specific stages of knowledge acquisition process and the actions that agents carried out in each of them. The six stages of acquisition of knowledge are: designing a project proposal, the creation of a knowledge base for collecting knowledge from experts, identification of the knowledge sources that will be consulted, the use of effective methods for capturing, structuring and validation of knowledge acquisition process stage so as to maximized the use of available resources, the dissemination of the knowledge base to interested entities.

Chapter 3 describes the contributions related to the acquisition of knowledge through argumentation. The knowledge representation is detailed for *AgSB* - Semantic Database Agent and *Ag* - Common Agent. The chapter presents the formalism for representing the influences of *AgCIB* - Common Influences Agent and common agent *Ag* internal influences. It is presented the acquisition of new knowledge, through the selection of the topic and the participation to debates. There are presented the Degree of Interest - *DegInt* and the Degree of Confidence *DegConf*, required for transforming the root argument in Belief. In the same chapter, there is a parallel to classical knowledge acquisition for various issues specific to each phase and their corresponding algorithms.

Chapter 4 includes contributions related to argumentative intelligent agents, *ARGKAMAS* system-level behavior. The system has three levels of complexity: the upper level consists of main agents, the intermediate level describes the multi-agent system that accounts for each main agent and lower level contains the functional units. In addition, the chapter describes the model of *AgSB* - Semantic Base Agent, *AgCIB* - Common Influences Base Agent, *Ag* - Common Agent, *AgDTR* - Debates Register Agent and *AgDT* - Debate Tree Agent.

Chapter 5 describes the experimental and applied contributions, which were aimed at validating the algorithm for determining the degree of confidence *DegConf*.

Chapter 6 covers general conclusions, the original contributions and future research directions.

Lista figurilor

Fig. 1 Structura unui argument conform Toulmin [23]	3
Fig. 2 Caracteristicile unui arbore de argumentare conform Dung [20] și Cayrol [24]	5
Fig. 3 Evaluare graduală în cadrul propus de Cayrol [24]	6
Fig. 4 Platforma educațională de tip second-screen	8
Fig. 5 Argumentul rădăcină al dezbaterii despre popularitatea limbajului <i>Python</i>	9
Fig. 6 Dezbaterea argumentului popularitate limbaj <i>Python</i> scăzută	11
Fig. 7 Dezbaterea argumentului învățare Python facilă	12
Fig. 8 Dezbaterea siguranței scăzute a aplicațiilor scrise în limbajul <i>Python</i>	13
Fig. 9 Dezbaterea argumentului aplicații cloud majoritare	13
Fig. 10 Arborele de dezbateri pentru popularitatea limbajului <i>Python</i>	14
Fig. 11 Schema generală a interacțiunilor dintre agenți în cadrul fiecărei etape de achiziție de cunoștințe	16
Fig. 12 Parametrii de calcul pentru determinarea <i>ValSoc</i>	29
Fig. 13 Nivelurile de abstractizare ale sistemului <i>ARGKAMAS</i>	31
Fig. 14 Componentele <i>interacțiuni externe și management dezbateri</i> ale sistemului <i>ARGKAMAS</i>	32
Fig. 15 Interacțiunile dintre agenți în cadrul nivelului superior	33
Fig. 16 Schema funcțională a agentului bază de date semantică <i>AgSB</i>	34
Fig. 17 Diagrama de secvențe pentru căutarea de caracteristici <i>Crt</i>	35
Fig. 18 Diagrama de secvențe pentru calculul distanței dintre două caracteristici	36
Fig. 19 Schema funcțională a agentului intern <i>SM</i> al <i>AgSB</i>	36
Fig. 20 Schema funcțională a agentului intern <i>GM</i> al <i>AgSB</i>	37
Fig. 21 Schema funcțională a agentului intern <i>PM</i> al <i>AgSB</i>	39
Fig. 22 Schema funcțională a agentului intern <i>KM</i> al <i>AgSB</i>	40
Fig. 23 Schema funcțională a agentului intern <i>CM</i> al <i>AgSB</i>	41
Fig. 24 Aplicația Java pentru determinarea gradelor	42
Fig. 25 Arbore de dezbateri cu un argument care ATACĂ	43
Fig. 26 Analiza datelor experimentale	44
Fig. 27 Reprezentarea grafică a analizei afirmației 1	44
Fig. 28 Comparatie între algoritmi și experiment	45
Fig. 29 Evoluția mesajelor pe platforma Twitter în timpul emisiunii XFactor	47

Lista tabelelor

Tab. 1 Setul de reguli pentru calculul gradului de interes <i>DegInt</i>	26
Tab. 2 Setul de reguli pentru calculul potențialului de participare la dezbateri <i>PartPot</i>	27
Tab. 3 Setul de reguli pentru calculul parametrului convingător <i>Conv</i>	28
Tab. 4 Setul de reguli pentru calculul <i>abs_ValSoc_ArgK</i>	29
Tab. 5 Corespondența dintre etapele de achiziție de cunoștințe și etapele algoritmului	30

1. Stadiul actual

Domeniul de cercetare al tezei este unul multi-disciplinar. În cadrul tezei, achiziția de cunoștințe se realizează pentru agenții inteligenți. Agenții sunt entitățile care necesită cunoaștere, iar achiziția le oferă această cunoaștere într-o formă procesabilă. Din punct de vedere clasic, cele două domenii sunt separate în sensul că achiziția de cunoștințe implică experți și ingineri de cunoștințe pentru extragerea și reprezentarea cunoașterii, iar sistemele multi-agent procesează informația care deja există implementată în sistem, exact ca în cazul sistemelor expert.

Teza consideră sistemul multi-agent ca fiind cel care necesită cunoaștere și abordează achiziția de cunoștințe pentru agenți folosind un mecanism comun în societatea umană: argumentarea. Astfel, domeniile principale de cercetare ale tezei sunt cel al achiziției de cunoștințe, sisteme multi-agent și argumentare. Documentarea realizată concluzionează că abordarea propusă în teză este una novatoare și care permite multiple direcții viitoare de cercetare.

1.1. Achiziția de cunoștințe

Achiziția de cunoștințe și utilizarea acestora sunt mecanisme esențiale de dezvoltare ale civilizației umane. În funcție de stadiul dezvoltării tehnologice, diverși autori situează momentul de început al domeniului achiziției de cunoaștere ca fiind etapa trecerii de la forma orală de transmitere la cea scrisă [1]. Alți autori consideră că momentul de început sunt secolele 17 și 18, odată cu dezvoltarea industriei la nivel global și inventarea tranzistorului sau debutul erei societății informaționale [2]. Unii cercetători susțin că explozia demografică și răspândirea civilizațiilor este strâns legată de posibilitatea de a capta cunoașterea și de a o transmite [3]. Dacă privim achiziția de cunoștințe prin prisma informației care se păstrează, atunci se pot considera picturile rupestre ca fiind printre primele metode de păstrare a cunoașterii, unele având o vechime de 30000-40000 de ani [4], [5].

Cercetarea științifică a domeniului achiziției de cunoștințe a început să se dezvolte odată cu dezvoltarea domeniului inteligenței artificiale, începând cu anii 1980. Sistemele expert au fost cele care au necesitat cunoștințe, iar achiziția lor s-a dovedit a fi dificilă. În [6], [7] se afirmă că procesul achiziției de cunoștințe este o piedică în construirea sistemelor expert. Inginerul de cunoștințe trebuie să formalizeze cunoașterea expertului, astfel ca aceasta să poată fi prelucrată de către un sistem informatic. În cadrul achiziției clasice, rolul fundamental al procesului de achiziție de cunoștințe este acela de a extrage cunoașterea de la experți într-un format procesabil. Cunoașterea achiziționată este inclusă într-o bază de cunoștințe. Forma bazei de cunoștințe trebuie să satisfacă scopul pentru care a fost creat proiectul de achiziție de cunoaștere. De obicei, tehnologia folosită depinde de instrumentele cu care organizația și experții din organizație sunt obișnuiți și antrenați.

În anul 1986 are loc primul workshop în domeniul achiziției de cunoștințe, *KAW*, la care au fost prezentate peste 120 de lucrări științifice [8]. Ingineria cunoașterii este un subdomeniu al Inteligenței Artificiale care aplică științele cognitive în construirea mașinilor inteligente. Domeniul științelor cognitive și cel al ingineriei cunoașterii se completează reciproc, reprezentând instrumente puternice de modelare și explorare. Domeniul științelor cognitive oferă instrumente și metode de înțelegere a mecanismelor primare de formare a cunoașterii [9]. Ingineria cunoașterii se focalizează pe includerea cunoașterii umane în cadrul algoritmilor. Domeniul ingineriei cunoașterii este divizat în trei tipuri de activități: reprezentarea cunoașterii, procesarea cunoașterii și achiziția de cunoștințe [10]. Sistemele expert sunt un produs al domeniului ingineriei cunoașterii, iar rolul acestora este să rezolve probleme reale într-o manieră similară cu cea a unui expert uman. Printre sistemele expert de referință se numără și sistemele INTERNIST-1, MYCIN, ILIAD sau DXplain [11].

Unul dintre cele mai avansate sisteme expert la ora actuală este Watson, conceput și dezvoltat de către IBM [12]. Sistemul este capabil să înțeleagă textul și limbajul natural într-o măsură suficient de mare încât să participe la concursuri de tip *Trivia*, având concurenți umani. Specificarea unei structuri la scară globală [13] (<http://www.w3.org/History/1989/proposal.html>), capabilă să stocheze documente și legături între acestea, a dus la apariția celui mai mare depozit de cunoștințe cunoscut umanității (*Wikipedia* - http://en.wikipedia.org/wiki/Main_Page). În [14] se propune limbajul HTML ca fiind un mediu universal de programare al interfețelor pentru un sistem expert. Reprezentarea cunoașterii sub formă semantică, reprezintă noua formă de stocare a cunoașterii în internet. Forma semantică este procesabilă algoritmic și stă la baza unor sisteme expert care înțeleg cerința și contextul și oferă informații adaptate [15], [16], (*Apple Siri* - <http://en.wikipedia.org/wiki/Siri>), (*Google Now* - http://en.wikipedia.org/wiki/Google_Now), (*Microsoft Cortana* - http://en.wikipedia.org/wiki/Microsoft_Cortana).

1.2. Argumentare

Argumentarea este o activitate rațională verbală și socială având ca scop acceptarea de către interlocutor a unui punct de vedere controversat. Teoria argumentării este un domeniu vast, interdisciplinar, care folosește teorii din filozofie, psihologie, lingvistică și comunicare [17]. În prezent, cea mai mare colecție de argumente este baza de date *AraucariaDB* a Universității Dundee [18]. Aceasta conține un număr de aproximativ 500 de argumente prelucrate de experți și culese din diverse resurse precum ziare, reviste, rapoarte judiciare, înregistrările parlamentare și grupuri de discuții on-line din diferite țări și din diferite domenii. Analiza argumentelor și a diverselor teorii argumentative este o muncă laborioasă. Comunitatea științifică *Globalargument* a reușit să analizeze un set specific de argumente extrem de minuțios, din perspectiva tuturor teoriilor dezvoltate până în prezent (<http://projects.kmi.open.ac.uk/GlobalArgument.net/about.html>).

1.2.1. Argumentare în sisteme multi-agent

În cadrul sistemelor multi-agent, agenții trebuie să interacționeze pentru a-și îndeplini obiectivele sau pentru a-și îmbunătăți performanțele și să fie capabili să ajungă la o înțelegere chiar dacă nu au informații complete și corecte asupra propriilor decizii și preferințe. Abordarea bazată pe argumentare permite agenților schimbul de informații sub formă de argumente despre credințe și atitudini mentale. Un argument este o informație care permite unui agent să-și justifice propunerea sau să influențeze preferințele altui agent sau percepția acestuia [19].

1.2.2. Concepte de bază ale argumentării

Adesea oamenii se confruntă cu informații conflictuale, fiind forțați să se descurce cu anumite inconsistențe, în urma cărora se construiesc argumente și contra-argumente în funcție de informația disponibilă. Argumentarea implică raționament care folosește concluziile anterioare ca suport pentru dezvoltarea viitoarelor concluzii. Contra-argumentele pot fi definite drept conflicte cu suportul sau concluzia unui alt argument. Argumentarea reprezintă procesul prin care sunt construite, comparate și/sau evaluate argumentele și contra-argumentele.

În cadrul dezvoltării tezei au fost studiate în special modelele dezvoltate de Dung [20], Amgoud și Cayrol [21] și cartea sinteză *Elemente de Argumentare* realizată de Besnard și Hunter [22]. În cadrul acestei secțiuni o să prezint câteva dintre definițiile principale, așa cum se regăsesc în modelele studiate.

Un **argument** este alcătuit dintr-o mulțime de ipoteze (suport) împreună cu concluzia (consecința) obținută prin unul sau mai mulți pași de raționament. Un argument cu o concluzie ce reprezintă respingerea unei concluzii sau respingerea uneia dintre ipotezele din suportul

unui alt argument este un contra-argument și se numește **argument de respingere**. Dacă argumentul A_1 are concluzia α , atunci argumentul A_2 ce are concluzia $\neg\alpha$ este un argument de respingere a concluziei pentru argumentul A_1 . Dacă un argument A_1 are în mulțimea suport informații reprezentate prin γ și $\gamma \rightarrow \delta$, atunci argumentul A_2 este argument de respingere a suportului pentru argumentul A_1 , dacă A_2 are o concluzie de forma $\neg\gamma$ sau $\neg(\gamma \rightarrow \delta)$. Fiind dat argumentul A_1 , numim argumentul A_2 **contra-argument** pentru A_1 , dacă A_2 este un argument de respingere pentru A_1 .

Componentele structurii unui argument sunt prezentate în figura următoare [23]:

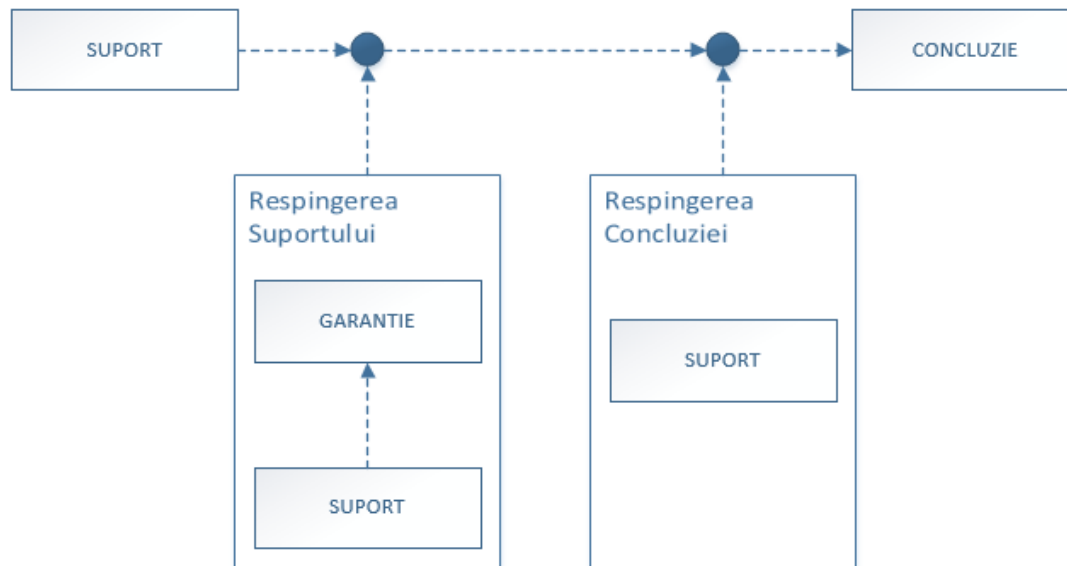


Fig. 1 Structura unui argument conform Toulmin [23]

1.2.3. Argumentarea logică

Argumentarea logică se bazează pe logica clasică, iar respingerile sau conflictele sunt între seturile de formule, ceea ce duce la noțiunea logică de inconsistență bazată pe inferența unei formule și negația ei. Natura inferenței este diversă și include: inferență analogică, inferență cauzală, inferență inductivă și inferență deductivă [22]. Din ipotezele prezentate, folosind logica clasică se deduce o concluzie validă. În această abordare, un argument este reprezentat printr-o pereche, în care primul item este o mulțime minimală de formule consistente ce verifică al doilea item al perechii. În [22] se arată că orice argument poate fi disputat de alte argumente. Un argument poate fi respins dacă unele dintre ipotezele prezentate sunt respinse. Astfel, se poate crea un arbore de argumente, ce reprezintă o sinteză a tuturor argumentelor și contra-argumentelor ce dispută argumentul din rădăcină.

Definiție [22]. Un **argument** reprezintă o mulțime alcătuită din formule și o concluzie în care formulele pot fi folosite pentru a demonstra acea concluzie. Fie Δ un set finit de formule. Numim argument perechea $\langle \Phi, \alpha \rangle$ ce verifică simultan:

1. $\Phi \not\vdash \perp$
2. $\Phi \vdash \alpha$
3. Φ este un subset minimal al lui Δ ce satisface 2.

Dacă $A = \langle \Phi, \alpha \rangle$ este un argument, spunem că A este un argument pentru α , în general ($\alpha \notin \Delta$). Mai spunem că Φ este **suport** pentru α , astfel că putem scrie: $Support(A) = \Phi$. Numim α **consecința** sau concluzia argumentului, astfel că putem scrie: $Concluzie(A) = \alpha$. Notăm cu Ω mulțimea tuturor argumentelor, fiind dată Δ .

Exemplu [22]. Fie $\Delta = \{\alpha, \alpha \rightarrow \beta, \gamma \rightarrow \neg\beta, \gamma, \delta, \delta \rightarrow \beta, \neg\alpha, \neg\delta\}$. Exemple de argumente:

$$\begin{aligned} & \langle \{\alpha, \alpha \rightarrow \beta\}, \beta \rangle \\ & \langle \{\gamma \rightarrow \neg\beta, \gamma\}, \neg\beta \rangle \\ & \langle \{\delta, \delta \rightarrow \beta\}, \beta \rangle \\ & \langle \{\neg\gamma\}, \neg\gamma \rangle \end{aligned}$$

Definiție [22]. Numim respingere a suportului pentru argumentul $\langle \Phi, \alpha \rangle$, argumentul de forma $\langle \Psi, \neg(\phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_n) \rangle$ în care $\{\phi_1, \phi_2, \dots, \phi_n\} \subseteq \Phi$ și $\Phi \cup \Psi \subseteq \Delta$.

Definiție [22]. Un argument $\langle \Psi, \beta \rangle$ este un argument de respingere a concluziei pentru argumentul $\langle \Phi, \alpha \rangle$ dacă și numai dacă $\beta \leftrightarrow \neg\alpha$ este o tautologie.

1.2.5. Cadrul abstract propus de Dung

Definiție [20]. Un **sistem de argumentare** este un cuplu $\langle \mathcal{A}, \mathcal{R} \rangle$, în care \mathcal{A} este o mulțime de argumente și \mathcal{R} o relație binară pe \mathcal{A} numită relație contrară. Fie $A_i, A_j \in \mathcal{A}$. Relația $A_i \mathcal{R} A_j$ înseamnă că A_j este contrazisă de A_i sau că A_i o contrazice pe A_j și se mai notează cu $(A_i, A_j) \in \mathcal{R}$.

Fie $A \in \mathcal{A}$. Mulțimea $\{A_i \in \mathcal{A} | A_i \mathcal{R} A\}$ este notată cu $\mathcal{R}^-(A)$, iar mulțimea $\{A_i \in \mathcal{A} | A \mathcal{R} A_i\}$ este notată cu $\mathcal{R}^+(A)$.

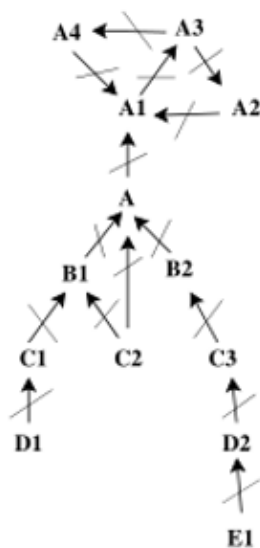
Definiție [20]. Fie \mathcal{G} , graful de relații contrare asociat sistemului de argumentare $\langle \mathcal{A}, \mathcal{R} \rangle$. Definim:

- un argument $A \in \mathcal{A}$ pentru care $\mathcal{R}^-(A) = \emptyset$ se numește **frunză** a grafului \mathcal{G} ,
- o **cale** de la A la B este o succesiune de argumente $C = A_1 - \dots - A_n$ astfel încât $A = A_1, A_1 \mathcal{R} A_2, \dots, A_{n-1} \mathcal{R} A_n, A_n = B$. Lungimea acesteia este $n - 1$ și este dată de numărul de arce ce constituie calea. Această lungime va fi notată cu l_C . Mulțimea de căi de la A la B va fi notată cu $\mathcal{C}(A, B)$.

Definiție [20]. Fie $A \in \mathcal{A}$.

- atacurile directe ale lui A sunt elementele mulțimii $\mathcal{R}^-(A)$,
- atacurile indirecte ale lui A sunt elementele A_i definite prin $\exists C \in \mathcal{C}(A_i, A)$ astfel încât $l_C = 2k + 1$, cu $k \geq 1$,
- suporturile directe ale lui A sunt atacurile directe ale elementelor mulțimii $\mathcal{R}^-(A)$,
- suporturile indirecte ale lui A sunt elementele A_i definite prin $\exists C \in \mathcal{C}(A_i, A)$ astfel încât $l_C = 2k$, cu $k \geq 2$.

Definiție [20]. Fie $A \in \mathcal{A}$. O ramură de atac (respectiv, de apărare) pentru A este o cale în \mathcal{G} de la o frunză către A de lungime impară (pară). Se afirmă că A este rădăcina unei ramuri de atac, respectiv, de apărare [24]:



În graful \mathcal{G} se remarcă:

- o cale de la $C2$ către A de lungime 2 ($C2 - B1 - A$),
- două circuite $A1 - A3 - A2 - A1$ și $A1 - A3 - A4 - A1$, fiecare de lungime 3, care nu sunt izolate (se remarcă faptul că $A1 - A3 - A2 - A1 - A3 - A4 - A1$ nu este un circuit conform definiției),
- circuitele anterioare sunt interconectate (în $A1$ și $A3$),
- căile $D1 - C1 - B1$ și $C3 - B2 - A$ sunt independente, astfel că $D1 - C1 - B1 - A$ și $C3 - B2 - A$ sunt dependente în rădăcină și că $D1 - C1 - B1 - A$ și $C2 - B1 - A$ sunt dependente,
- $D1, C2, E1$ sunt frunze ale \mathcal{G} ,
- $D1 - C1 - B1 - A$ este o ramură de atac pentru A , astfel că $C2 - B1 - A$ este o ramură de apărare pentru A ,
- $B1$ și $B2$ sunt doi atacanți direcți ai lui A ,
- $C1, C2$ și $C3$ sunt trei apărători direcți ai lui A ,
- $D1$ și $D2$ sunt cei doi atacanți indirecti ai lui A ,
- $E1$ este singurul apărător indirect al lui A .

Fig. 2 Caracteristicile unui arbore de argumentare conform Dung [20] și Cayrol [24]

Generalitatea și flexibilitatea cadrului abstract propus de Dung a dus la numeroase extinderi și propuneri de noi modele bazate pe acesta.

1.2.6. Sistemul de argumentare bipolar propus de Cayrol

Cayrol [24] introduce relația de interacțiune de tip **sprijin/suport** între argumente, prin definirea unui sistem de argumentare bipolar, în opoziție cu sistemul unipolar propus de Dung [20]. Cadrul descris de către Cayrol [24] este mai general, definind argumente „pro” și „contra”.

Fie \mathcal{L} un limbaj logic și \vdash relația de inferență asociată limbajului \mathcal{L} .

Definiție [24]. Se numește **argument**, cuplul (S, C) , pentru care S este un suport (mulțimea de formule consistente a lui \mathcal{L}) și C este o formulă consistentă a lui \mathcal{L} ce respectă constrângerile următoare: $S \vdash C$ și S este minimă pentru a obține pe C , adică $\forall \Psi_i \in S, S \setminus \psi_i \not\vdash C$.

Fie $A \in \mathcal{A}, B \in \mathcal{A}$.

- Relația $\mathcal{R}_{atac} A B$ este reprezentată grafic prin $A \rightarrow B$,
- Relația $\mathcal{R}_{sprijin} A B$ este reprezentată grafic prin $A \rightarrow B$,
- Mulțimea $\{A_i \in \mathcal{A} | A_i \mathcal{R}_{atac} A\}$ este notată cu $\mathcal{R}_{atac}^-(A)$, iar mulțimea $\{A_i \in \mathcal{A} | \mathcal{R}_{atac} A_i\}$ este notată cu $\mathcal{R}_{atac}^+(A)$,
- Mulțimea $\{A_i \in \mathcal{A} | A_i \mathcal{R}_{sprijin} A\}$ este notată cu $\mathcal{R}_{sprijin}^-(A)$, iar mulțimea $\{A_i \in \mathcal{A} | \mathcal{R}_{sprijin} A_i\}$ este notată cu $\mathcal{R}_{sprijin}^+(A)$.

Sistemul de argumentare $\langle \mathcal{A}, \mathcal{R}_{atac}, \mathcal{R}_{sprijin} \rangle$ definește un graf orientat \mathcal{G} , numit **graful interacțiunilor bipolare**. Un argument $A \in \mathcal{A}$ pentru care $\mathcal{R}_{atac}^-(A) = \emptyset$ și $\mathcal{R}_{sprijin}^-(A) = \emptyset$ se numește **frunză**.

Definiție [24]. Fie $A \in \mathcal{A}$ cu $\mathcal{R}_{atac}^-(A) = \{B_1, \dots, B_n\}$ și $\mathcal{R}_{sprijin}^-(A) = \{C_1, \dots, C_p\}$.

O **evaluare** este o funcție $v: \mathcal{A} \rightarrow V$ cu

$$v(A) = g(h_{sprijin}(v(C_1), \dots, v(C_p)), h_{atac}(v(B_1), \dots, v(B_n)))$$

în care **funcția** $h_{atac}: V^* \rightarrow \mathcal{H}_{atac}$ evaluează **calitatea atacului** asupra argumentului A , iar **funcția** $h_{sprijin}: V^* \rightarrow \mathcal{H}_{sprijin}$ evaluează **calitatea sprijinului** asupra argumentului A , iar

funcția $g: \mathcal{H}_{atac\acute{a}} \times \mathcal{H}_{sprijin\acute{a}} \rightarrow V$, notată cu $g(x, y)$ este crescătoare în x și descrescătoare în y .

Definiție [24]. Fie A și B două argumente. Dacă $v(A) < v(B)$ atunci între cele două argumente există relația de ordonare $A < B$.

Fie $V = [-1, 1]$ un interval de numere reale,

- $\mathcal{H}_{atac\acute{a}} = \mathcal{H}_{sprijin\acute{a}} = [0, \infty)$ interval de numere reale,
- $h_{sprijin\acute{a}}(x_1, \dots, x_n) = h_{atac\acute{a}}(x_1, \dots, x_n) = \sum_{i=1}^n \frac{x_i+1}{2}$,
- $g(x, y) = \frac{1}{1+y} - \frac{1}{1+x}$

Astfel, $\alpha = 0$ și $\beta = \infty$ și $g(\alpha, \alpha) = 0$, iar $h_{atac\acute{a}}(x_1, \dots, x_n, x_{n+1}) \geq h_{atac\acute{a}}(x_1, \dots, x_n)$ deoarece $\frac{x_{n+1}+1}{2} \geq 0$ când $x_{n+1} \in [-1, 1]$. Relația de ordonare se respectă și pentru funcția $h_{sprijin\acute{a}}$.

$h_{sprijin\acute{a}}(0) = h_{atac\acute{a}}(0) = \alpha$, α fiind valoarea minima a intervalului $[0, \infty)$, iar β valoarea maximă a acestuia. Se verifică că $g(\alpha, \beta) = g(0, \infty) = -1 = V_{min}$ și $g(\beta, \alpha) = g(\infty, 0) = 1 = V_{max}$.

În figura următoare sunt prezentate câteva exemple de evaluare graduală conform modelului propus de Cayrol [24]:

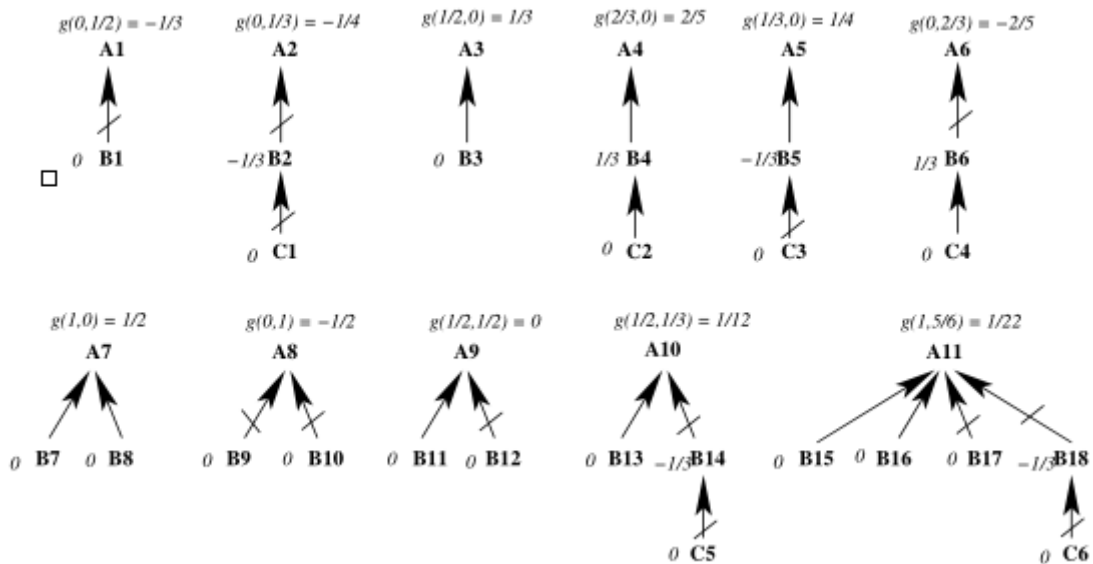


Fig. 3 Evaluare graduală în cadrul propus de Cayrol [24]

2. Contribuții privind utilizarea argumentării pentru achiziția de cunoștințe în sisteme bazate pe agenți inteligenți

Argumentarea este fundamentală pentru acumularea de cunoștințe și dezvoltarea individului în societate. Argumentarea este de diferite tipuri și poate fi realizată la diverse niveluri, însă în forma ei primară, argumentarea se folosește pentru achiziția de noi cunoștințe. Cu cât un individ are un nivel cultural mai ridicat, cu atât argumentările sunt mai elaborate, iar scopul primar de achiziție de cunoaștere este mascat de actul argumentării în sine.

În cadrul tezei, susțin ideea că un individ se poate dezvolta și poate fi un membru activ și funcțional al societății datorită argumentării. Chiar dacă gradul de cultură este redus, baza de cunoștințe este suficient de dezvoltată pentru a participa la activitățile grupului social din care face parte. Achiziția de cunoștințe se realizează prin conversațiile zilnice despre evenimentele cotidiene ale comunității. Conversațiile arareori au caracter pur informativ și chiar și acestea se transformă cu mare ușurință în dezbateri. Consider că modelul propus în teză reprezintă o nișă și că există numeroase direcții și perspective viitoare de cercetare. Organizarea achiziției de cunoștințe într-un mod similar cu cea condusă de un inginer de cunoștințe, pe etape clare permite realizarea unei argumentări structurate și sistematice care are ca scop final realizarea unei priviri de ansamblu cât mai complete asupra subiectului dezbaterii. Imbricarea domeniilor argumentării, achiziției de cunoștințe și sistemelor multi-agent permite direcții de cercetare interdisciplinare și evoluția anumitor direcții fără a afecta celelalte direcții.

În cadrul scenariului de argumentare voi prezenta un arbore de dezbateri, etapele și modul în care acesta a fost construit. Un astfel de scenariu este necesar pentru a motiva introducerea termenilor din capitolul următor. Deoarece teza implementează achiziția de cunoștințe pentru agenți inteligenți, scenariul va fi descris dintr-o perspectivă algoritmică și care să permită formalizarea termenilor.

Dezbaterile au loc între agenți. Acesta este contextul general în care se desfășoară dezbaterile. Orice agent Ag poate propune un argument care să fie dezbătut. Acest argument reprezintă argumentul inițial, sau rădăcina arborelui de discuții.

Să ne imaginăm un scenariu în care un tânăr pasionat de tehnologie urmărește documentarul *Download: The True Story of the Internet* (<http://www.imdb.com/title/tt1684716/>). Documentarul este urmărit folosind un televizor SmartTV și o tabletă pe care este instalată o aplicație de tip second-screen sincronizată cu emisiunea TV. Platforma deține o suită de informații suplimentare relative la emisiunea curentă. La momente specifice de timp, informația suplimentară este afișată pe aplicația mobilă, iar pe aplicația de pe SmartTV este afișată o notificare.

O astfel de platformă este dezvoltată în cadrul proiectului European ITEA2 TWIRL (<https://itea3.org/project/twirl.html>), proiect în cadrul căruia am avut rolul de consultant în analiza și dezvoltarea componentei de dezbateri (în cadrul **Anexei 2** există confirmarea colaborării în cadrul proiectului european). Proiectul este în desfășurare, având termen de finalizare sfârșitul acestui an. În cadrul acestei teze nu este prezentat proiectul european decât ca un argument pentru fezabilitatea soluției propuse și pentru potențialul abordării de a produce efecte economice.

În figura următoare este prezentată o captură a ambelor aplicații, cea de pe SmartTV și cea mobilă. Se observă că în timp ce se urmărește un documentar despre Delta Dunării, pe tabletă se afișează informații suplimentare despre fauna acesteia.



Fig. 4 Platforma educațională de tip second-screen

În cazul scenariului propus în teză, agentul personal al tânărului pasionat de tehnologie primește o cerere de documentare de forma următoare:

DEBATE(Python)

În momentul primirii cererii de documentare, agentul se transformă într-un inginer de cunoștințe. Sarcina acestuia este de a interoga experții în domeniu, de a extrage cunoașterea acestora relevantă la topicul dezbaterii și de a prelucra astfel încât să poată oferi un răspuns corespunzător tânărului pasionat de tehnologie.

O primă contribuție a tezei constă în definirea unui număr de șase etape specifice pentru procesul achiziției de cunoaștere și corelarea acestora cu diverse momente din algoritmul de dezbateri. În continuare vor fi prezentate cele șase etape și acțiunile pe care agentul le întreprinde în fiecare dintre acestea.

2.1. Conceperea unei propuneri de proiect

Primirea unei cereri de documentare este necesară dar nu este suficientă pentru declanșarea procesului de dezbateri. Chiar și pentru un om ar fi dificil să înceapă o dezbateri dispunând doar de acest topic. Topicul este mult prea vag. *Python* poate fi un șarpe, o formație de comedie stand-up, un limbaj de programare, numele unei rachete aer-aer, o creatură din mitologia greacă, un revolver, titlul unui film sau numele unui roller-coaster din parcul de distracții din Efteling, Olanda.

Așa cum a fost precizat și în *Introducere*, în cadrul tezei am conceput un sistem care are potențial de a fi implementat într-un sistem comercial. În cadrul primei etape de achiziție, este foarte important pentru agent să poată interoga o bază de informații din care să selecteze argumentul care declanșează dezbateri. Modelul pentru stocarea bazei de informații se bazează pe cel implementat de Freebase, care este o bază de date colaborativă pentru stocare structurată a cunoașterii umane [25], [26].

În cadrul tezei a fost esențial faptul că Freebase poate fi extinsă deoarece în forma actuală nu conține proprietăți specifice care pot fi argumentate. Extinderea Freebase este naturală

Contribuții privind utilizarea argumentării pentru achiziția de cunoștințe în sisteme bazate pe agenți inteligenți

sistemului, rațiunea acestuia fiind tocmai să permită colaborarea și extinderea bazei de cunoaștere în funcție de necesități.

O contribuție importantă adusă de teză constă în introducerea unui domeniu nou de forma:

/debates/general_debates/debatable cu valoarea */type/enumeration*

Introducerea acestui domeniu face accesibilă Freebase de către agenții inteligenți din sistemul ARGKAMAS. Mai multe detalii vor fi prezentate în capitolul următor.

În acest moment, este explicabil de ce agentul personal al tânărului pasionat de tehnologie descoperă mai multe perspective pentru topicul *Python*. Pentru a se putea decide care perspectivă prezintă interes, agentul își consultă baza internă de domenii de interes. Domeniul de interes al agentului este reprezentat de un set de structuri ce cuprind domenii și categorii similare celor din Freebase. Deoarece reprezentarea este similară, agentul poate comunica direct cu informațiile stocate în Freebase. Agentul *Ag* va realiza o comparație între domeniile sale de interes și perspectivele definite în Freebase referitor la topicul *Python* și le va selecta pe cele care îndeplinesc criteriul de selecție. Pentru fiecare perspectivă aleasă, agentul va declanșa dezbateri pentru toate proprietățile argumentative. Detalii referitoare la algoritmul de selecție vor fi prezentate în capitolul următor.

La finalul *Etapei 1*, agentul va dispune de o listă de argumente care trebuie dezbătute.

2.2. Crearea unei baze de cunoștințe pentru colectarea cunoașterii de la experți

Disponând de lista de argumente de dezbătut, agentul va planifica o serie de dezbateri referitoare la topicul *Python*. Setul de dezbateri va fi trimis către *AgDTR* - Agentul Registru de Dezbateri, care are ca scop configurarea și managementul dezbaterilor înregistrate. Pentru fiecare argument de dezbătut, va fi creat un agent *AgDT* – Agent Arbore de Dezbateri, care va fi responsabil de toate interacțiunile din arborele de dezbateri al argumentului respectiv.

În figura următoare este reprezentat argumentul rădăcină al uneia dintre dezbaterile înregistrate în *AgDTR*:

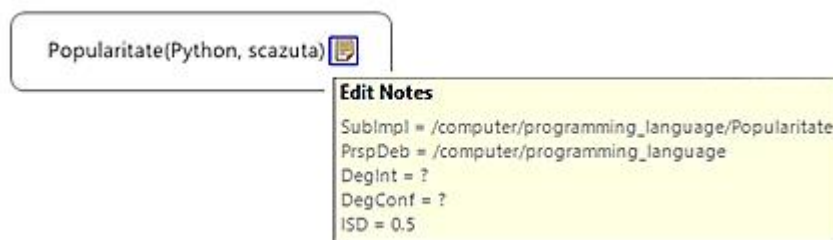


Fig. 5 Argumentul rădăcină al dezbaterii despre popularitatea limbajului *Python*

Se remarcă în figura anterioară faptul că argumentul este descris de o serie de parametri. La acest moment ne interesează doar *PrspDeb* – Perspectiva Dezbaterii și *ISD*- Interesul agentului pentru declanșarea unei dezbateri.

Perspectiva dezbaterii *PrspDeb*, are un rol esențial în cadrul dezbaterii. Aceasta asigură perspectiva topicului și permite agenților să aleagă dacă participă sau nu la dezbateri. În cazul scenariului din teză, se remarcă faptul că topicul *Python* are o perspectivă din zona limbajelor de programare:

PrspDeb = /computer/programming_language

În cadrul sistemului ARGKAMAS, perspectiva principală este notată prin *PrspCtx* – Perspectiva Contextuală a dezbaterii, iar fiecare domeniu sau categorie relativă la topic prin *PrspCnx* – Perspectiva Conexă a perspectivei principale.

Tot în descrierea argumentului rădăcină există și parametrul *ISD*, care reprezintă interesul agentului pentru declanșarea dezbaterii. Acest parametru se deduce prin compararea domeniilor de interes ale agentului, cu perspectivele unui topic. Parametrul *ISD* este luat în considerare atât la declanșarea arborelui de dezbateri cât și la determinarea *DegInt* – gradului de interes pentru conținutul dezbaterii. O prezentare detaliată a parametrilor și a algoritmilor va fi realizată în capitolul următor.

2.3. Identificarea surselor de cunoaștere care vor fi consultate

În momentul în care managementul dezbaterii a fost preluat de către *AgDT*, acesta difuzează în sistem argumentul rădăcină. Difuzarea este mecanismul prin care toți agenții din sistem sunt anunțați de topicul dezbaterii și perspectiva din care are loc aceasta. În cazul scenariului din teză, perspectiva topicului *Python* este cea a limbajelor de programare. Forma sub care este publicat un argument este următoarea:

$$ArgPub = \{PrspDeb, Attr\}$$

în care:

- *PrspDeb* – reprezintă perspectiva din care are loc dezbateri,
- $Attr = SubImpl(Topic, Pvaloare) \equiv /domeniu/categorie/Pnume(Topic, Pvaloare)$ – reprezintă atributul publicat în sistemul multi-agent pentru a putea fi argumentat de alți agenți interesați.

În momentul în care un argument este publicat de către *AgDT*, topicul devine vizibil pentru toți agenții din sistem. În acest moment, agenții pot verifica dacă dispun de cunoaștere despre topicul respectiv. Deoarece modelul utilizat pentru descrierea agenților este BDI (Belief, Desire, Intention), noțiunea de cunoaștere pentru agent este reprezentată prin credința *Bel*. O credință despre un topic este reprezentată astfel:

$$Bel = \{SubImpl(Topic, Pvaloare), PrspDeb, DegInt, DegConf\}$$

Odată identificată o credință despre un topic, trebuie identificată o regulă de influență între argumente. Spre exemplu, între *Moneda(X, puternica)* și *Credit(X, riscant)* există o influență. Dar și între *Guvern(X, instabil)* și *Credit(X, riscant)* de asemenea există o influență. În mod intuitiv, un argument poate fi influențat de multe alte atribute. Influențele dintre atribute alcătuiesc *IB* - Baza de Influențe. Este natural că un argument influențează un alt argument într-o măsură mai mare sau mai mică. Influența reprezintă susținere sau atac.

Există două tipuri de baze de influențe:

- *IIB* – Baza Internă de Influențe – fiecare agent conține o bază proprie de influențe. Există și este naturală situația în care doi agenți diferiți dețin reguli de influență diferite despre aceleași argumente,
- *AgCIB* – Agentul Bază de Date de Influențe Comune – este un agent specializat care conține toate influențele de tip lege. Dacă în cazul agenților, influențele între aceleași argumente puteau să fie diferite, în cazul *AgCIB* influențele nu sunt contestabile. Sunt echivalente legilor fizicii, economice etc. Spre exemplu, este un fapt unanim acceptat că un *Guvern(X, instabil)* susține un *Credit(X, riscant)*. Se observă că influențele agentului *AgCIB* se referă la un topic general *X*, adică sunt valabile indiferent de topic. Evident, trebuie să existe o legătură între subiectul implicit *SubImpl* și topicul *X*.

2.4. Utilizarea de metode eficiente pentru capturarea, structurarea și validarea cunoașterii

Pentru a putea participa la argumentare, agentul consultă mai întâi *IIB* - baza internă de influențe și dacă nu identifică o influență între argumentul publicat și argumentul din baza proprie de cunoștințe, va consulta *AgCIB* în încercarea de a identifica o influență. Dacă în niciuna dintre baze nu se reușește identificarea unei influențe, atunci agentul nu poate participa la dezbateri.

În cadrul tezei va fi prezentat un posibil algoritm pentru completarea bazei de influențe dar această problemă este complexă și reprezintă o direcție viitoare de cercetare.

La dezbateri despre popularitatea limbajului *Python* au participat agenți cu diverse domenii de interese precum: universitate, firma software, firma angajări, editura, sistem de operare linux, Google cloud, banca, hardware Raspberry Pi, Amazon Servicii Web, școală, persoana, NASA, conferința PyCon etc. Se remarcă diversitatea agenților care conțin credințe despre topicul *Python*. Cu cât un topic poate fi prezent în mai multe domenii, cu atât forma arborelui va fi mai complexă.

Un agent *Ag* poate declanșa un număr foarte mare de arbori de dezbateri (în funcție de numărul de proprietăți de tip *debatable* pe care le identifică) însă este foarte posibil ca topicul să nu se regăsească în foarte multe domenii, așa că arborii pot fi mici ca dimensiune, iar generarea rezultatelor finale să nu dureze foarte mult.

Popularitatea

/computer/programming_language/Popularitate(Python,scazuta)

este dezbătută de două argumente, unul de susținere și unul de atac, ca în figura următoare:

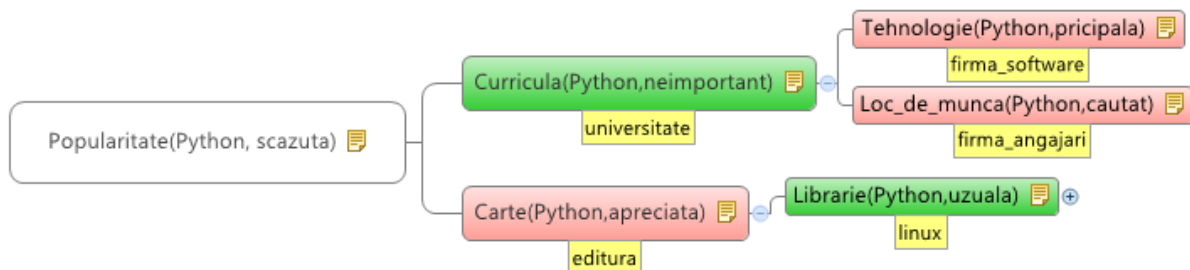


Fig. 6 Dezbateri argumentului popularitate limbaj *Python* scăzută

Argumentul */education/university/Curricula(X, neimportant)* al agentului *Ag(universitate)* susține popularitatea scăzută cu un grad $DegInfl = 0.6$. Adică, faptul că studenții nu studiază limbajul Python are un impact mediu asupra popularității acestuia. Continuând pe această ramură, agentul *Ag(firma_software)* atacă argumentul curricula este neimportantă prin argumentul */business/business_operation/Tehnologie(X, principala)* cu un grad destul de mic $DegInfl = 0.25$. Adică, faptul că într-o anumită firmă de software limbajul Python este principalul limbaj nu influențează foarte mult curricula dintr-o universitate. Pe de altă parte, dacă piața locurilor de muncă cere ca angajații să cunoască limbajul, atunci este posibil ca acest aspect să influențeze curricula universitară.

Agentul *Ag(firma_angajari)* atacă argumentul curricula este neimportantă cu argumentul */business/industry/Loc_de_munca(X, cautat)* având un grad mare de influență $DegInfl = 0.8$.

Argumentul */book/publishing_company/Carte(X, apreciata)* al agentului *Ag(editura)* atacă faptul că popularitatea limbajului este scăzută cu un grad mare de influență $DegInfl = 0.9$. Cu alte cuvinte, deoarece cărțile despre limbajul Python sunt foarte apreciate, aprecierea fiind

calculată în funcție de număr de vânzări și rating, popularitatea nu are cum să fie scăzută. Cărțile despre limbaj sunt apreciate și datorită faptului că librăria Python este uzuală în cadrul sistemelor de operare Linux. Argumentul $/computer/operating_system/Librarie(X,uzuala)$ al agentul $Ag(linux)$ susține cu un grad mare $DegInfl = 0.77$.

În figura următoare este prezentat arborele de dezbateri pentru argumentul $Invatare(Python, facila)$.

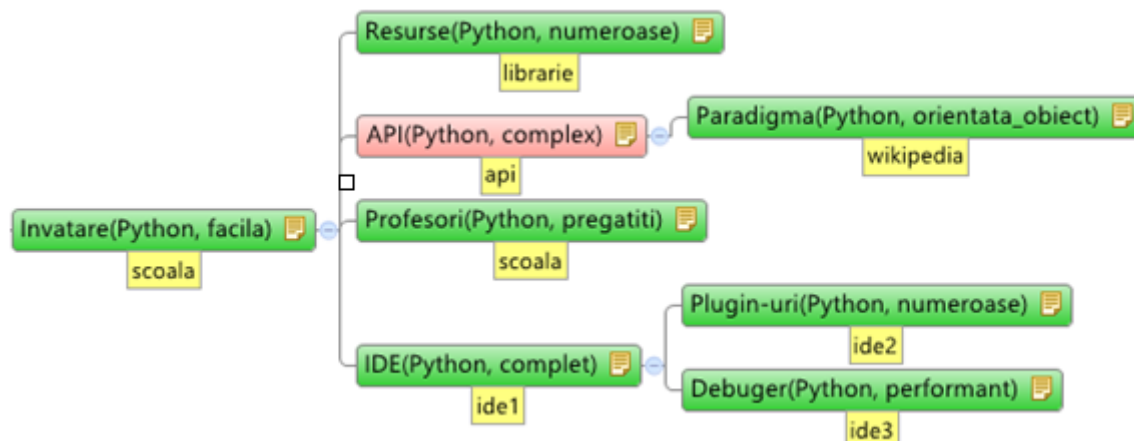


Fig. 7 Dezbateri argumentului învățare Python facilă

Faptul că există resurse numeroase, profesori pregătiți și medii de dezvoltare complete permit o învățare facilă a oricărui limbaj. Argumentul $/library/public_library/Resurse(X, numeroase)$ al agentului $Ag(librarie)$ are un grad mare de susținere $DegInfl = 0.8$. Argumentul $/education/school/Profesori(X, pregatiti)$ al agentului $Ag(scoala)$ are un grad mare de susținere $DegInfl = 0.8$. Argumentul $/computer/programming_language/IDE(X, complet)$ al agentului $Ag(ide1)$ are un grad mediu de susținere $DegInfl = 0.55$. Din punct de vedere al regulilor de influență, faptul că un mediu este complet nu înseamnă că limbajul este facil de învățat. Spre exemplu, limbajul Scala dispune de medii de dezvoltare complete dar limbajul în sine are o curbă mare de învățare.

Argumentul $/computer/programming_language/Plugin_uri(X, numeroase)$ al agentului $Ag(ide2)$ are un grad mic de susținere $DegInfl = 0.3$. Argumentul $/computer/programming_language/Debugger(X, performant)$ al agentului $Ag(ide3)$ are un grad mare de susținere $DegInfl = 0.77$. Argumentul $/computer/programming_language/API(X, complex)$ al agentului $Ag(api)$ are un grad mare de atac $DegInfl = 0.8$. Cu alte cuvinte, dacă API-ul unui limbaj este complex, există o curbă de învățare mare pentru acesta. Acesta este susținut de argumentul $/computer/programming_language_paradigm/Paradigma(X, orientata_obiect)$ al agentului $Ag(Wikipedia)$ cu un grad foarte mic de susținere $DegInfl = 0.15$. Acest aspect este corect, deoarece complexitatea API-ului nu este datorată faptului că paradigma de programare este orientată obiect.

În figura următoare argumentul

$/business/business_operation/Aplicatie(X, siguranta_scazuta)$

al agentului $Ag(banca)$ atacă cu un grad foarte mic de influență faptul că librăria Python este uzuală. Este normal ca gradul să fie mic deoarece chiar dacă pentru mediul bancar aplicațiile nu sunt scrise în limbajul Python, există numeroase alte scenarii în care acesta se folosește și este foarte popular.

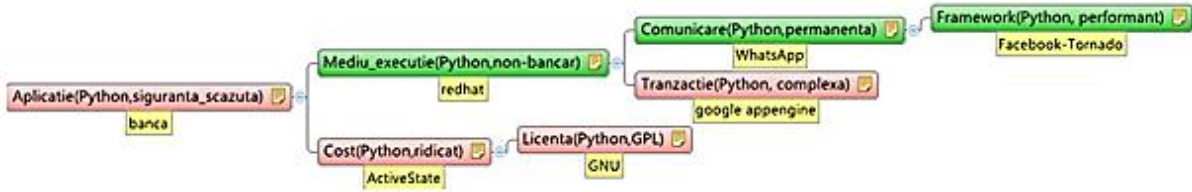


Fig. 8 Dezbaterea siguranței scăzute a aplicațiilor scrise în limbajul Python

Argumentul $/computer/software/Mediu_executie(X, non_bancar)$ a agentului $Ag(RedHat)$ are un grad mare de susținere $DegInfl = 0.7$. Argumentul $/computer/computer_platform/Tranzactie(X, complexa)$ al agentului $Ag(Google AppEngine)$ atacă faptul că mediul de execuție al limbajului Python este unul non-bancar cu faptul că limbajul permite realizare de tranzacții complexe, specifice unui mediu bancar. Gradul de atac este mediu și are valoarea $DegInfl = 0.67$. Argumentul $/computer/software/Comunicare(X, permanenta)$ al agentului $Ag(WhatsApp)$ are un grad mare de susținere $DegInfl = 0.9$. Cu alte cuvinte, o aplicație care necesită comunicare permanentă este un mediu de execuție non-bancar. Argumentul $/computer/software/Framework(X, performant)$ al agentului $Ag(Facebook Tornado)$ susține faptul că un framework performant este necesar unei aplicații pentru comunicare. Gradul de susținere este foarte mare și are valoarea $DegInfl = 0.95$.

Agentul companiei ActiveState atacă faptul că aplicațiile scrise în Python au o siguranță scăzută deoarece produsele sale garantează siguranța prin costul ridicat al acestora. Argumentul $/business/business_operation/Cost(X, ridicat)$ are un grad foarte mic de atac $DegInfl = 0.13$. Acest aspect este corect, deoarece faptul că produsele unei companii costă mult nu presupune că toate aplicațiile scrise în Python au un grad crescut de siguranță. Agentul GNU atacă costul ridicat al produselor software care folosesc componente protejate de licența liberală GPL. Argumentul $/computer/software_license/Licenta(X, GPL)$ al agentului $Ag(GNU)$ are un grad mare de atac $DegInfl = 0.7$.

Agentul $Ag(google cloud)$ susține faptul că librăria Python este uzuală deoarece majoritatea aplicațiilor din acesta sunt implementate în Python. Argumentul $/computer/computer_platform/Aplicatii_cloud(X, majoritare)$ are un grad de susținere foarte mic $DegInfl = 0.2$. Gradul mic de susținere este corect din punct de vedere al unei reguli de influență generală, care nu se referă la un anumit limbaj. Faptul că într-un cloud ponderea aplicațiilor scrise într-un anumit limbaj este mai mare nu înseamnă că acel limbaj este o librărie uzuală.

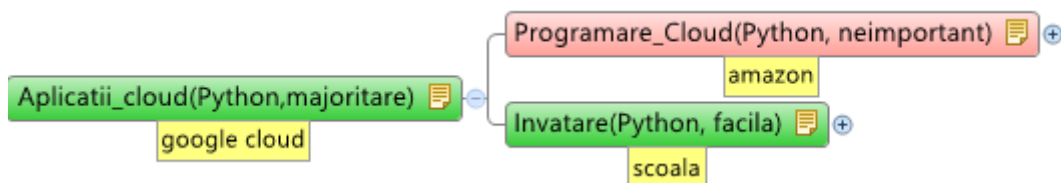


Fig. 9 Dezbaterea argumentului aplicații cloud majoritare

Agentul $Ag(scoala)$ susține că datorită faptului că învățarea limbajului Python este facilă, aplicațiile din cloud scrise în acest limbaj sunt majoritare. Argumentul $/education/school/Invatare(X, facila)$ are un grad de susținere mare $DegInfl = 0.75$. Agentul $Ag(amazon)$ consideră că limbajul de programare dintr-un cloud este neimportant și că oricând ponderea aplicațiilor scrise în diverse limbaje se poate schimba. Argumentul $/business/industry/Programare_Cloud(X, neimportant)$ are un grad de atac foarte mic $DegInfl = 0.2$. Cu alte cuvinte, argumentul agentului $Ag(amazon)$ are o influență slabă care nu atacă prea tare părintele.

Arborele de dezbateră pentru scenariul propus despre $Popularitate(Pyhton, scazuta)$, va avea următoarea formă:

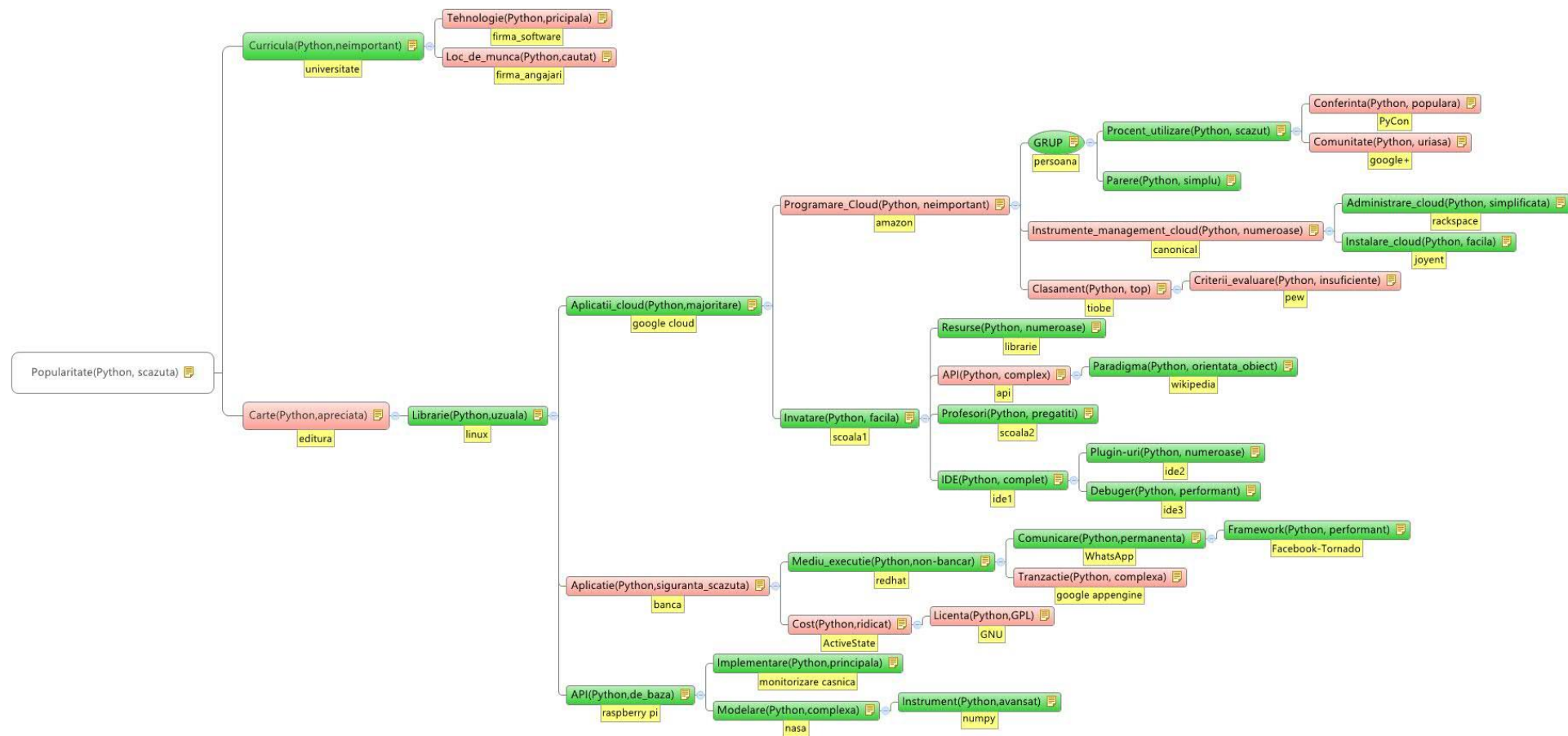


Fig. 10 Arborele de dezbatere pentru popularitatea limbajului Python

2.5. Desfășurarea etapei de achiziție astfel încât să fie maximizată utilizarea resurselor disponibile

Fiecare agent Ag poate impune agentului registru de dezbateri $AgDTR$ o anumită strategie de participare a agenților la dezbateri și implicit o modalitate de finalizare a acesteia. În funcție de strategia aleasă, rezultatele pot fi foarte diferite între ele. Acest mecanism este similar celui din societatea umană în care orice dezbateri depinde de o serie de factori intrinseci și extrinseci care pot conduce la rezultate diferite. Spre exemplu, dacă discută doi șoferi de la o firmă de transport internațional sau doi economiști despre situația exportului unei țări este posibil să se ajungă la rezultate diferite. Rezultatele finale sunt influențate nu numai de pregătirea subiecților dar și de aria de cuprindere, de timpul alocat dezbaterii, de numărul de argumente, de numărul de schimbări de topic, de distanța față de perspectiva dezbaterii $PrspDeb$ etc.

În cadrul tezei, am definit o strategie de participare liberă. O dezbateri se încheie atunci când toți agenții posibili au participat la dezbateri. Este suficient să se identifice un topic și o influență pentru a putea participa. Mai multe detalii despre posibile strategii de participare și finalizare a unei dezbateri vor fi prezentate în capitolul următor.

2.6. Diseminarea bazei de cunoștințe către entitățile interesate

Din momentul în care agentul arbore de dezbateri $AgDT$ decide că dezbateri s-a finalizat, acesta mai are de realizat următoarele acțiuni:

- să calculeze $DegInt$ – gradul de interes și $DegConf$ – gradul de încredere în argumentul rădăcină,
- să anunțe agenții participanți că dezbateri s-a finalizat și că parametrii au fost calculați,
- să transmită credința către agentul registru de dezbateri $AgDTR$. La rândul său, acesta va aștepta ca toate dezbaterile unui agent Ag să se finalizeze și va trimite rezultatul cumulat către Ag .

Ca și contribuție în cadrul tezei, am definit cei doi parametri care caracterizează o credință: $DegInt$ și $DegConf$. Datorită acestor parametri agentul înțelege semnificația unui argument pentru el și condițiile în care poate participa la o dezbateri.

$$Credinta = Argument + \{DegInt, DegConf\}$$

În care credința reprezintă un argument pentru care agentul are un interes $DegInt$ și o încredere $DegConf$.

Parametrul $DegInt$, gradul de interes, reprezintă interesul manifestat de agent pentru dezbateri respectivă. Spre exemplu, este posibil ca de la bun început agentul să nu fie foarte interesat (parametrul ISD) dar dacă dezbateri conține foarte multe argumente din zona domeniului lui de interes, este posibil ca în final să devină totuși interesat de dezbateri.

Parametrul $DegConf$, gradul de încredere, reprezintă încrederea agentului în argument și depinde de toate argumentele din dezbateri. În funcție de încrederea agentului într-un argument, acesta poate sau nu să participe în dezbateri viitoare.

În momentul când calculul celor doi parametri a fost finalizat, $AgDT$ anunță toți agenții participanți de acest aspect.

În figura următoare este prezentată schema generală a interacțiunilor dintre agenții sistemului $ARGKAMAS$, în cadrul fiecărei etape de achiziție de cunoștințe.

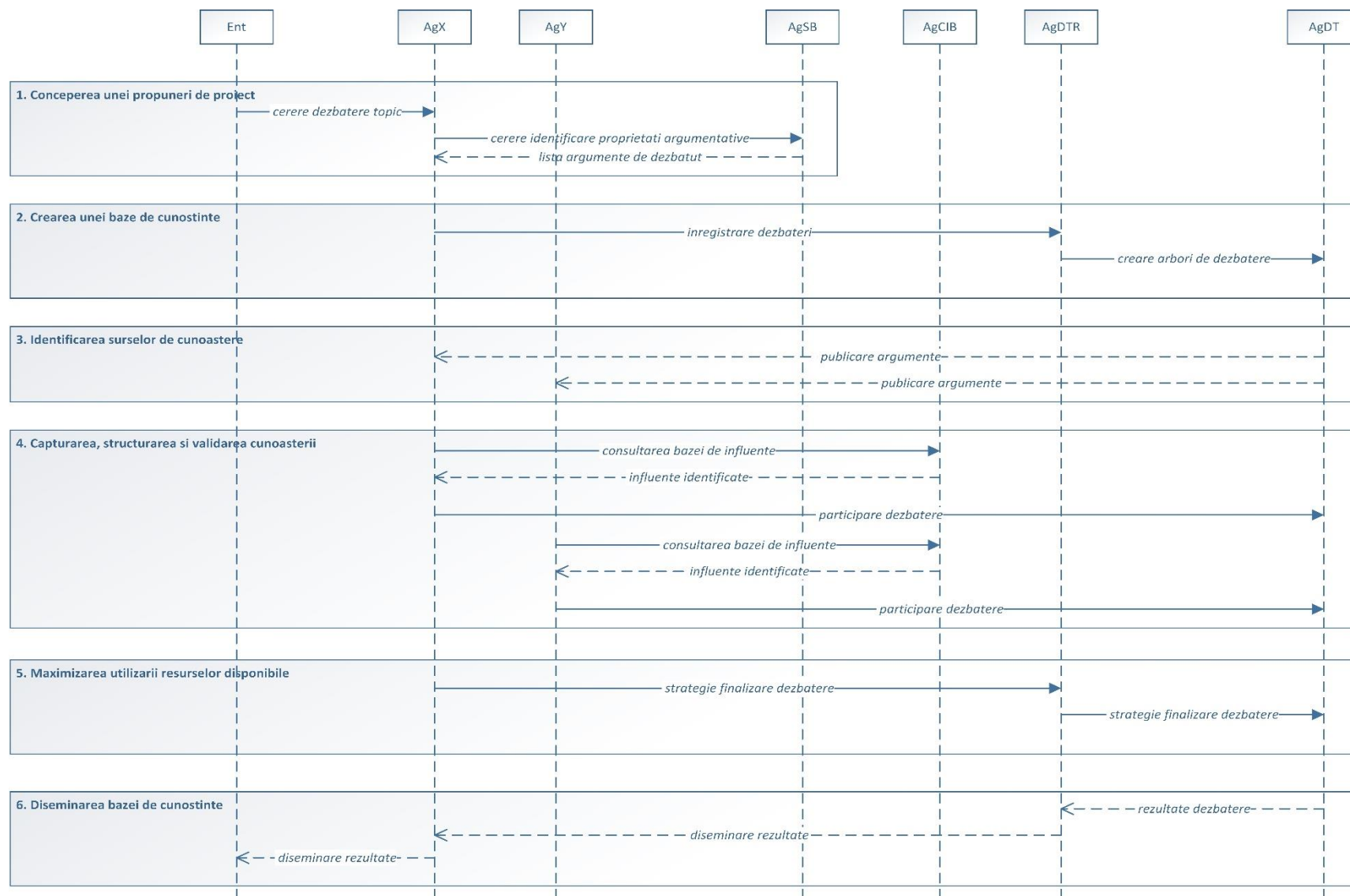


Fig. 11 Schema generală a interacțiunilor dintre agenți în cadrul fiecărei etape de achiziție de cunoștințe

Contribuții privind utilizarea argumentării pentru achiziția de cunoștințe în sisteme bazate pe agenți inteligenți

În schema anterioară se remarcă faptul că în cadrul primei etape de achiziție se realizează identificarea topicurilor pentru dezbateri. Entitatea externă *Ent* trimite o cerere de documentare către un agent *Ag* iar acesta consultă agentul bază semantică *AgSB* referitor la existența de proprietăți argumentabile.

În cadrul celei de-a doua etape de achiziție, agentul *Ag* trimite o cerere de declanșare de dezbateri pentru proprietățile argumentabile identificate, către agentul registru de dezbateri *AgDTR*. Acesta crează un nou agent *AgDT* pentru fiecare proprietate argumentabilă.

În cadrul celei de-a treia etape de achiziție, agenții arbore de dezbateri *AgDT* publică argumentele de dezbătut în cadrul sistemului *ARGKAMAS*.

În cadrul etapei patru de achiziție, agenții *Ag* identifică relații de influență între argumente și dacă acestea există, participă la dezbateri.

În cadrul celei de-a cincea etape, agentul registru de dezbateri *AgDTR* maximizează utilizarea resurselor disponibile prin stabilirea unei strategii de finalizare a dezbaterii. Strategia este trimisă către agenții *AgDT*.

Etapa a șasea de diseminare presupune finalizarea dezbaterii și informarea agentului *Ag* care a inițiat procesul de argumentare despre rezultate. Agentul *Ag* va informa în mod specific entitatea externă *Ent* despre rezultatul documentării.

3. Contribuții privind reprezentarea cunoștințelor în ARGKAMAS

Contribuția esențială a tezei constă în introducerea argumentării ca instrument principal de achiziție de cunoștințe pentru agenții inteligenți. În cadrul sistemelor multi-agent argumentarea este folosită cu precădere pentru negociere sau învățare, folosirea acesteia ca instrument exclusiv de achiziție reprezentând o direcție nouă de cercetare.

În cadrul acestui capitol este formalizat sistemul ARGKAMAS de achiziție de cunoștințe prin argumentare. Sistemul este alcătuit din cinci tipuri de agenți, fiecare având un rol bine stabilit. Pe tot parcursul elaborării tezei, am dorit ca modelul să aibă potențial de a fi implementat într-o soluție comercială și să producă efecte economice. Din acest motiv, baza de date semantică extinde modelul implementat de Freebase, iar sistemul poate fi implementat folosind paradigma de programare bazată pe agenți. Agenții care vor fi descriși în acest capitol sunt:

- *AgSB* – Agentul Baza de Date Semantică,
- *AgCIB* – Agentul Baza de Date de Influențe Comune,
- *Ag* – Agent Comun,
- *AgDT* – Agentul Arbore de Dezbatere,
- *AgDTR* – Agentul Registru de Dezbateri.

3.1. Reprezentarea cunoștințelor în sistemul ARGKAMAS

În cadrul sistemului ARGKAMAS sunt definite două tipuri de cunoștințe: una semantică ce permite crearea unui univers de entități și descrieri ale acestuia și una de tip credință care permite agentului să se exprime cu privire la entitățile din univers. Modelul este similar cu cel al societății umane, în sensul că fiecare individ are propria opinie despre evenimentele și entitățile din universul în care există.

3.1.1. Reprezentarea cunoștințelor agentului AgSB

Pentru a putea declanșa dezbateri un agent *Ag* trebuie să identifice proprietățile argumentabile despre o anumită entitate. În continuare, entitățile vor purta denumirea de topicuri, în mod similar cu noțiunea de topic definită de Freebase. Un topic poate reprezenta entități fizice, creații media, clasificări, concepte abstracte etc. Un topic poate face referință către alte topicuri din diferite domenii de informații. Freebase a fost conceput astfel încât să încurajeze extinderea structurii și a conținutului bazei semantice.

Agentul *AgSB* reprezintă o bază de date semantică, descrisă prin intermediul a două noțiuni: structură ierarhică și topic. În cadrul tezei vom reinterpretă structura principală a acesteia astfel încât să putem defini noțiunile cu care agenții din sistemul ARGKAMAS vor lucra. Se va păstra din Freebase structura ierarhică pentru domenii, categorii și proprietăți. Astfel un domeniu reunește mai multe categorii, iar o categorie este definită prin mai multe proprietăți.

Pentru a permite agenților să argumenteze, în cadrul tezei a fost completat modelul semantic cu o nouă structură de forma:

/debates/general_debates/debatable

cu valori de tipul */type/enumeration*. Acesta este definit de către Freebase și permite definirea unor seturi de valori. Noua structură definește un domeniu de tip */debates*. În cadrul acestui domeniu vor fi definite diferite tipuri de categorii de dezbateri. Momentan este definită doar categoria */debates/general_debates* pentru dezbateri cu caracter general care conține o singură proprietate *../debtable*. Spre exemplu, dacă se dorește definirea proprietății argumentabile */finante/banci/Credit* aceasta trebuie să fie de tipul */debates/general_debates/debtable* și să aibă drept valoare, de pildă, mulțimea

{*avantajos, ieftin, flexibil, riscant, neriscant*}. Astfel, pentru dezbaterile referitoare la *Credit*, se poate alege valoarea *riscant*. Evident, un agent poate declanșa cinci dezbateri, câte una pentru fiecare dintre cele cinci valori.

Definiție 3.1 Se numește **perspectivă** a unui topic o structură ierarhică de forma:

$$Prsp = /domeniu/categorie$$

în care domeniul *domeniu* reunește un număr de categorii *categorie*.

Noțiunea de *perspectivă* este esențială în cadrul procesului de argumentare. Un topic poate fi descris din diverse perspective, și în funcție de acestea, dezbaterile pot implica diverși agenți. O dezbateră are întotdeauna o perspectivă principală prin prisma căreia are loc argumentația.

Definiție 3.2 Se numește **perspectivă contextuală sau principală**, acel context în care are loc dezbaterile, permițând descrierea unui topic dintr-un anumit punct de vedere.

Definiție 3.3. Se numește **subiect** al unui topic o structură ierarhică de forma:

$$Sub = /domeniu/categorie/Pnume$$

în care:

- *Pnume* – reprezintă numele unei proprietăți argumentabile. Așa cum a fost precizat anterior, o categorie poate conține oricâte proprietăți argumentabile.

Noțiunea de *subiect* este specifică unui argument. Fiecare argument din baza de credințe a unui agent se referă la un anumit subiect.

Definiție 3.4. Se numește **caracteristică** a unui topic o structură ierarhică de forma:

$$Crt = /domeniu/categorie/Pnume/Pvaloare$$

în care:

- *Pvaloare* – reprezintă una dintre valorile setului definit pentru o proprietate argumentabilă.

Din punct de vedere funcțional, în cadrul sistemului multi-agent, agentul bază de date semantică *AgSB* îndeplinește două roluri esențiale:

- comunică agenților interesați perspectivele principale și proprietățile argumentabile,
- poate calcula distanța dintre două perspective. Scopul calculării distanței va fi descris în capitolul următor

3.1.2. Reprezentarea cunoștințelor agentului *Ag*

Agentul *Ag* este cea mai activă entitate din sistemul ARGKAMAS. Acesta primește cererea de argumentare, identifică proprietățile argumentabile, înregistrează dezbaterile și cumulează rezultatele obținute. De asemenea, acumulează cunoașterea rezultată în urma dezbaterii și o refolosește în cadrul altor argumentări.

Agentul *Ag* este alcătuit din două componente principale:

- baza de cunoștințe,
- baza de influențe internă – va fi descrisă în subcapitolul următor.

Baza de cunoștințe a agentului Ag este la rândul ei alcătuită din două tipuri de cunoștințe:

- domeniile de interes,
- credințe, dorințe și intenții.

Definiție 3.5 Domeniul de interes este un set finit de structuri având forma:

$$DomInt = \{Struct_1, \dots, Struct_n\}$$

în care $Struct_k = (/domeniu \vee /domeniu/categorie) \subseteq AgSB$.

Domeniul de interes este definit de către entitatea ce emite cererea de argumentare. Acesta poate fi modificat ulterior în funcțiile de noile interese. Existența unui domeniu de interes este necesară pentru a limita acumularea de cunoștințe de către agent, pentru a asigura o utilizare eficientă a resurselor sistemului ARGKAMAS.

Al doilea tip de cunoștințe este reprezentat de credințe, dorințe și intenții. Credințele agentului Ag reprezintă cunoașterea persistentă pe care o poate reutiliza. Dorințele și intențiile sunt cunoștințe volatile, care apar ca urmare a unor evenimente și dispar în momentul în care au fost finalizate. Acestea vor fi prezentate în cadrul algoritmilor de dezbateră.

Definiție 3.6 Gradul de Interes $DegInt$ al agentului pentru o credință reprezintă măsura în care credința prezintă interes pentru agent.

În cadrul sistemului ARGKAMAS, gradul de interes $DegInt$ este o variabilă lingvistică fuzzy cu termenii $\{mic, mediu, mare\}$ definiți pe intervalul $[0,1]$.

Definiție 3.7 Gradul de Încredere $DegConf$ al agentului pentru o credință reprezintă măsura în care acesta are încredere în credința respectivă.

Gradul de încredere $DegConf$ este de asemenea o variabilă lingvistică fuzzy având termenii $\{mic, mediu, mare\}$ definiți pe intervalul $[0,1]$.

Datorită celor două mărimi, agentul este capabil să raționeze despre propriile cunoștințe și astfel să poată alege dacă participă sau nu într-o dezbateră. Cu toate că cele două măsuri sunt definite în mod similar, semnificația acestora este diferită. Există cunoștințe de care agentul este foarte sigur, are o mare încredere în ele, dar are un interes scăzut deoarece nu fac parte din domeniul său de interes.

În cadrul tezei, definirea celor două mărimi este cea mai valoroasă contribuție, deoarece a permis construirea întregului sistem ARGKAMAS într-un mod coerent. Teza definește o nișă de cercetare în cadrul căreia se pot studia în amănunt efectele *interesului* și *încrederii* asupra comportamentului agentului.

Definiție 3.8 Un argument reprezintă o proprietate argumentabilă despre un topic. Acesta are următoarea formă extinsă:

$$/domeniu/categorie/Pnume(Topic, Pvaloare)$$

în care:

- $Topic \in AgSB$, reprezintă un topic din baza semantică a agentului $AgSB$,
- $Pnume$ este o proprietate de tip $/debates/general_debates/debatable$,
- $Pvaloare$ este una dintre valorile din setul de valori argumentabile ale $Pnume$.

Forma compactă de reprezentare a unui argument este următoarea:

$$SubImpl(X, Pvaloare)$$

în care $SubImpl$ reprezintă subiectul implicit al argumentului despre topicul X și este echivalent cu $/domeniu/categorie/Pnume$.

Definiție 3.9 Credința reprezintă argumentul pentru care un agent Ag are un anumit interes și în care are o anumită încredere. Aceasta se reprezintă astfel:

$$Bel\langle SubImpl(Topic, Pvaloare), PrspDeb, DegInt, DegConf \rangle$$

în care:

- $SubImpl = /domeniu/categorie/Pnume$ și reprezintă subiectul implicit al argumentului,
- $PrspDeb$ reprezintă perspectiva dezbaterii în care a fost dobândită credința respectivă. Dacă credința nu a fost dobândită în urma unei dezbateri anterioare ci provine din tehnici specifice de knowledge discovery, atunci $PrspDeb = null$,
- $DegInt$ reprezintă gradul de interes pentru argumentul respectiv,
- $DegConf$ reprezintă gradul de încredere în argumentul respectiv.

În cadrul sistemului ARGKAMAS este permis ca un agent să achiziționeze cunoștințe atât din surse externe cât și ca urmare a unei dezbateri. Cele două tipuri de cunoștințe pot coexista în cadrul aceleiași baze de cunoștințe. Spre exemplu, un agent poate primi informații de la un senzor despre starea vremii. Pentru agent, informațiile nu sunt nici controlabile și nici discutabile și deci trebuie luate ca atare. Din acest motiv, $DegInt = DegConf = 1$. Credința în cazul unui argument primit din sursă externă are următoarea formă:

$$Bel\langle SubImpl(Topic, Pvaloare), null, 1, 1 \rangle$$

Așa cum a fost precizat anterior, o credință reprezintă un argument pentru care agentul are un interes $DegInt$ și o încredere $DegConf$.

$$Credinta = Argument + \{DegInt, DegConf\}$$

3.2. Reprezentarea influențelor dintre cunoștințe

În cadrul sistemului ARGKAMAS legăturile există și sunt definite pentru argumente și nu pentru topicuri. Pentru a evita problema complexității, descrierea legăturilor se face între două argumente, în sensul că un argument are un anumit efect asupra altui argument. Dacă există mai multe argumente cu efecte asupra altui argument, pentru fiecare în parte se va descrie legătura și efectul acesteia. În acest fel se pot descrie modele suficient de complexe.

Definiție 3.10 Efectul unui argument asupra altui argument se numește **influență**.

Pentru a putea participa la o dezbateră, agentul Ag trebuie să identifice o *influență* între o credință proprie și o credință din arborele de dezbateră. Există două posibilități de identificare: consultarea bazei interne de influențe IIB sau consultarea bazei comune de influențe $AgCIB$. Fiecare agent dispune de o bază internă de influențe care este consultată într-o primă instanță. Dacă nu se identifică o influență în cadrul bazei interne, atunci agentul apelează la baza comună de influență furnizată de $AgCIB$.

Deoarece, agentul consultă mai întâi baza internă de influențe, influențe similare se pot regăsi și în baza internă a agentului și în baza agentului $AgCIB$. Acesta este un caz particular, deoarece presupune faptul că agentului $AgCIB$ i-a fost introdusă influența din exterior, iar agentul Ag a dobândit o influență similară ca urmare a unei dezbateri.

3.2.1. Reprezentarea influențelor agentului $AgCIB$

Agentul bază de date de influențe comune $AgCIB$, conține influențe, între argumente, care au caracter de lege universală. Acestea sunt acceptate de către toți agenții și sunt folosite ca atare în dezbateri.

Definiție 3.11 Valența unei influențe reprezintă o legătură de susținere sau de atac între două argumente.

$$InflVal \in \{pro, con\}$$

în care *pro* reprezintă susținere, iar *con* presupune atac.

Definiție 3.12 Gradul de Influență $DegInfl$ reprezintă măsura în care un argument afectează un alt argument.

Gradul de influență $DegInfl$ este o variabilă lingvistică fuzzy cu termenii $\{mic, mediu, mare\}$ definiți pe intervalul $[0,1]$.

Definiție 3.13 O regulă de **influență** reprezintă legătura dintre argumente, legătură descrisă prin valență și grad de influență. O regulă de influență are următoarea formă:

$$Infl\langle Arg_i, \{Arg_{j_1}, \dots, Arg_{j_n}\}, InflVal, DegInfl \rangle$$

în care:

- Arg_i reprezintă argumentul influențat,
- $\{Arg_{j_1}, \dots, Arg_{j_n}\}$ este un set nevid de argumente care influențează pe Arg_i ,
- $InflVal \in \{pro, con\}$ și reprezintă valența pozitivă - susținere, respectiv negativă - atac a influenței,
- $DegInfl$ reprezintă Gradul de Influență al unui argument asupra altuia.

Forma argumentului este $/domeniu/categorie/Pnume_k(X, Pvaloare_k)$. Se remarcă faptul că topicul argumentului este X . Notația semnifică faptul că regula de influență este valabilă indiferent de topic, cu alte cuvinte este o regulă universală. Setul de argumente $\{Arg_{j_1}, \dots, Arg_{j_n}\}$ conține minim un argument. Setul se consideră în totalitatea lui. În cadrul setului fiecare argument are o pondere de participare $\tau_{jk} \in [0,1]$. Ponderea va fi folosită în calculul gradului de încredere $DegConf$.

Expresia regulii de influență din *definiția 3.13* se interpretează astfel: Argumentul Arg_i este influențat $InflVal$ cu un grad $DegInfl$ de către setul de argumente Arg_j .

Scopul unei dezbateri este acela de a determina valorile $DegInfl$ și $DegConf$ ale argumentului rădăcină. Există posibilitatea ca în urma dezbaterii, comunitatea să nu fie de acord cu argumentul propus. Indiferent de opinia comunității, argumentul va deveni credință pentru agentul Ag . Dacă comunitatea nu este de acord cu argumentul, atunci credința agentului va avea următoarea formă:

$$Bel(\neg \text{SubImpl}(\text{Topic}, P\text{valoare}), Prsp_{Deb}, DegInt, DegConf)$$

iar în cadrul regulii de influență, valența se schimbă din $InflVal$ în $\neg InflVal$. Agentul $AgCIB$ va verifica dacă există o influență între subiectele implicite și va furniza regula de influență cu valență schimbată. Dacă este vorba despre un set de argumente, atunci cu fiecare argument din set comunitatea NU trebuie să fie de acord, pentru ca valența să poată fi schimbată.

3.2.2. Reprezentarea influențelor interne ale agentului Ag

Pentru a participa la o dezbatere, un agent Ag va consulta mai întâi baza internă de influențe și dacă nu găsește o regulă de influență între argumente, va consulta și agentul $AgCIB$. O regulă de influență din baza internă a agentului Ag are o reprezentare similară cu cea pentru agentul $AgCIB$.

$$Infl\langle Arg_i, \{Arg_{j_1}, \dots, Arg_{j_n}\}, InflVal, DegInfl \rangle$$

în care parametrii au aceeași interpretare.

Definiție 3.14 Distanța dintre perspective $\delta(Prsp_x, Prsp_y)$ reprezintă măsură similarității dintre cele două structuri și se determină astfel:

- dacă $Prsp_1 = /domeniu_1/categorie_1$ și $Prsp_2 = /domeniu_1/categorie_1$ atunci distanța este etichetată cu $\delta(Prsp_1, Prsp_2) = d0$ și reprezintă perspective identice,
- dacă $Prsp_1 = /domeniu_1/categorie_1$ și $Prsp_2 = /domeniu_1/categorie_2$ atunci distanța este etichetată cu $\delta(Prsp_1, Prsp_2) = d1$ și reprezintă perspective diferite ale aceluiași domeniu,
- dacă $Prsp_1 = /domeniu_1/ \dots$ și $Prsp_2 = /domeniu_2/ \dots$ atunci distanța este etichetată cu $\delta(Prsp_1, Prsp_2) = d2$ și reprezintă perspective diferite din domenii diferite.

Pentru a identifica reguli de influență între argumentele din dezbatere se poate aplica criteriul *distanța dintre perspective*, și se parcurg următorii pași:

- i. după finalizarea dezbaterii, pentru fiecare argument din arborele de dezbatere (nod) se calculează distanța $\delta(Prsp_{Deb}, Prsp_x)$ dintre perspectiva argumentului rădăcină și perspectiva nodului curent,
- ii. se calculează procentul de argumente din numărul total de argumente pentru fiecare distanță,
- iii. dacă procentul depășește un anumit prag stabilit apriori, atunci se vor crea reguli de influență între argumentul rădăcină și argumentul pentru care distanța respectă pasul *ii*,
- iv. forma regulii de influență va fi următoarea: $Infl\langle Arg_i, Arg_j, InflVal, DegInfl \rangle$, în care:
 - Arg_i are forma $Prsp_{Deb}/Pnume_{Deb}(X, Pvaloare_{Deb})$,
 - Arg_j are forma $Prsp_j/Pnume_j(X, Pvaloare_j)$,
 - valența $InflVal$ - se calculează valența arcelor de la rădăcină până la argumentul curent și se stabilește dacă nodul curent SUSTINE sau ATACĂ rădăcina,
 - gradul de influență $DegInfl$ se poate calcula în funcție de procentul de argumente al sub-arborelui a cărui rădăcină este argumentul curent.

În momentul când baza internă de reguli de influență a agentului Ag acumulează noi reguli, acestea sunt transmise către agentul $AgCIB$. Dacă există un număr suficient de mare de reguli similare, primite de la diverși agenți, $AgCIB$ poate decide că regula de influență primită trebuie înglobată în baza comună de influențe și anunță agenții să-și ștergă regula pe care o aveau în baza internă.

3.3. Achiziția de cunoștințe noi

Etapile majore ale algoritmului de determinare a gradelor de interes și încredere au fost încadrate în etape din achiziția clasică de cunoștințe. Această încadrare este foarte importantă deoarece apropie algoritmul de un proces clasic de achiziție de cunoștințe, și permite unui inginer de cunoștințe să înțeleagă procesul de achiziție din sistemul ARGKAMAS și să îmbunătățească fiecare etapă separat de celelalte.

3.3.1. Selecția topicului și declanșarea dezbaterii

Procesul de achiziție de cunoștințe este declanșat la primirea unei cereri $DEBATE(Topic)$. Agentul poate primi o astfel de cerere din exteriorul sistemului ARGKAMAS sau din interior, ca urmare a unui proces de introspecție. Cererea este transformată într-o dorință de forma:

$$Des\langle DEBATE(Topic) \rangle$$

În momentul în care cererea este transformată în dorință, agentul Ag trece într-o stare de motivație și declanșează procesul de selecție a topicurilor argumentabile. Dorința este transformată în scop:

$$Des\langle DEBATE(Topic) \rangle \rightarrow Goal\langle DEBATE(Topic) \rangle$$

Agentul Ag va trimite o cerere către agentul $AgSB$ care va identifica perspectivele topicului. Răspunsul primit de la agentul $AgSB$ conține o listă de perspective pentru topicul cerut:

$$Topic \left\{ \begin{array}{l} (PrspCtx_1, \{PrspCnx_1, \dots, PrspCnx_n\}), \\ (PrspCtx_2, \{PrspCnx_1, \dots, PrspCnx_n\}), \\ \dots \\ (PrspCtx_n, \{PrspCnx_1, \dots, PrspCnx_n\}) \end{array} \right\}$$

în care:

- $PrspCtx$ reprezintă perspectiva contextuală, conform definiției 3.2
- $PrspCnx$ reprezintă perspectiva conexă, definită mai jos

Definiție 3.15 **Perspectiva conexă** $PrspCnx$ reprezintă o perspectivă asociată unei perspective contextuale.

Definiție 3.16 **Interesul pentru Declanșarea unei Dezbateri** ISD reprezintă o măsură a similarității dintre perspective, calculată între un topic pregătit în vederea dezbaterii și domeniul de interes al agentului, ce primește cererea de argumentare.

În cadrul sistemului ARGKAMAS, interesul pentru declanșarea unei dezbateri ISD este o variabilă lingvistică fuzzy cu termenii $\{mic, mediu, mare\}$ definiți pe intervalul $[0,1]$.

Pentru fiecare structură din domeniul de interes se va calcula valoarea ISD astfel:

- $Struct_k$ se va compara cu fiecare perspectivă $PrspCtx_i$
 - dacă $PrspCtx_i = /domeniu_1/categorie_1$ și $Struct_j = /domeniu_1/categorie_1$ atunci $ISD = mare$ – interesul agentului coincide cu perspectiva,
 - dacă $PrspCtx_i = /domeniu_1/categorie_1$ și $Struct_j = /domeniu_1$ atunci $ISD = mediu$ – interesul agentului cuprinde printre altele și perspectiva, deci există un interes mediu în declanșarea unei dezbateri. Spre exemplu, agentul Ag este

interesat de domeniul $/music$, iar perspectiva este despre instrumente muzicale,

- dacă $PrspCtx_i = /domeniu_1/categorie_1$ și $Struct_j = /domeniu_2/ \dots$ atunci ISD_{NU} se calculează

Pentru fiecare $PrspCtx_i$ pentru care a fost calculat ISD , se va compara fiecare structură din domeniul de interes cu perspectivele conexe $PrspCnx$ pentru a afla valoarea ISD . Se consideră că dacă perspectiva contextuală $PrspCtx$ prezintă interes atunci și perspectivele conexe $PrspCnx$ prezintă un oarecare interes:

- $Struct_k$ se va compara cu fiecare perspectivă $PrspCnx_i$
 - dacă $PrspCnx_i = /domeniu_1/categorie_1$ și $Struct_j = /domeniu_1/categorie_1$ atunci $ISD = mediu$ – interesul agentului coincide cu perspectiva conexă,
 - dacă $PrspCnx_i = /domeniu_1/categorie_1$ și $Struct_j = /domeniu_1$ atunci $ISD = mic$ – interesul agentului cuprinde printre altele și perspectiva conexă, deci există un interes mic,
 - dacă $PrspCnx_i = /domeniu_1/categorie_1$ și $Struct_j = /domeniu_2/ \dots$ atunci ISD_{NU} se calculează

Argumentul rădăcină, la finalizarea dezbaterii, se va transforma în credință. Indiferent dacă proprietatea aleasă pentru dezbatere face parte din perspectiva contextuală $PrspCtx$ sau din una conexă $PrspCnx$, pentru argumentul rădăcină perspectiva dezbaterii $PrspDeb$ va fi identică cu perspectiva contextuală $PrspCtx$.

$$PrspDeb \equiv PrspCtx$$

Astfel, agentul Ag pentru același subiect implicit $SubImpl$ poate avea credințe diferite dar din perspective diferite. În momentul în care agentul Ag a identificat setul de dezbateri, acestea sunt transformate în intenții și sunt trimise către agentul arbore registru de dezbateri $AgDTR$ pentru înregistrarea și declanșarea dezbaterii. Pentru fiecare dezbatere din set, forma intenției este următoarea:

$$Int\langle SubImpl(Topic, Pvaloare), PrspDeb, ?_{DegInt}, ?_{DegConf} \rangle$$

în care:

- $SubImpl$ reprezintă subiectul implicit și în funcție de tipul perspectivei selecționate poate să fie $SubImpl = PrspCtx/Pnume$ sau $SubImpl = PrspCnx/Pnume$,
- $PrspDeb$ reprezintă perspectiva dezbaterii și este echivalentă cu perspectiva contextuală $PrspCtx$,
- $?_{DegInt}$ – gradul de interes care trebuie determinat
- $?_{DegConf}$ – gradul de încredere care trebuie determinat

Datorită faptului că întreg procesul de dezbatere este delegat către agentul arbore registru de dezbateri $AgDTR$, agentul Ag este liber să primească alte cereri de tip $DEBATE$ și să participe la rândul lui în dezbaterile active.

3.3.2. Participarea la dezbatere

La o dezbatere poate participa orice agent Ag din sistemul ARGKAMAS inclusiv agentul care a propus dezbaterea. Fiecare dezbatere este monitorizată de un agent arbore de dezbatere $AgDT$. La apariția unui argument în arbore, acesta publică evenimentul în tot sistemul.

Definiție 3.17 Un **argument publicat** este o informație necesară unui agent pentru a se decide dacă participă la dezbatere. Forma acestuia este următoarea:

$$ArgPub\langle PrspDeb, SubImpl(Topic, Pvaloare) \rangle$$

în care:

- *PrspDeb* reprezintă perspectiva dezbaterii
- *SubImpl(Topic, Pvaloare)* reprezintă argumentul

În cadrul tezei a fost implementat un algoritm de participare liberă la o dezbateri, doar pe baza existenței unei reguli de influență.

3.3.3. Determinarea gradului de interes *DegInt*

Gradul de interes *DegInt* reprezintă interesul unui agent față de argumentul respectiv în contextul dezbaterii. Contextul dezbaterii este esențial deoarece poate modifica interesul inițial pentru argument..

Conform *definiției 3.6*, gradul de interes *DegInt* este o variabilă fuzzy. Aceasta reprezintă o variabilă de ieșire pentru un sistem fuzzy cu două variabile de intrare:

- *ISD* – reprezintă interesul pentru declanșarea unei dezbateri și este o mărime fuzzy conform *definiției 3.16*,
- *PrspDebDist* – reprezintă distanța dezbaterii față de perspectiva dezbaterii *PrspDeb* și se determină astfel:
 - pentru fiecare argument din arbore se calculează distanța $\delta(PrspDeb, Prsp_x)$ față de perspectiva argumentului rădăcină. Distanța dintre perspective este calculată conform *definiției 3.14*,
- din numărul total de argumente se calculează procentul pentru fiecare valoare a distanței dintre perspective *%d0, %d1 sau %d2*,
- formula pentru *ValPrspDebDist* este următoarea:
 - $ValPrspDebDist = 0 * (%d0) + 1 * (%d1) + 2 * (%d2) \in [0, 2]$

În cadrul sistemului ARGKAMAS, distanța dezbaterii față de perspectiva dezbaterii *PrspDebDist* este o variabilă lingvistică fuzzy cu termenii {mic, mediu, mare} definiți pe intervalul [0, 2], determinată pe baza *ValPrspDebDist*.

Setul de reguli pentru calculul gradului de interes *DegInt* este prezentat în tabelul următor:

#	IF		THEN	
	ISD	PrspDebDist	DoS	DegInt
1	mic	mic	1.00	mediu
2	mediu	mic	1.00	mare
3	mare	mic	1.00	mare
4	mic	mediu	1.00	mic
5	mediu	mediu	1.00	mediu
6	mare	mediu	1.00	mediu
7	mic	mare	1.00	mic
8	mediu	mare	1.00	mic
9	mare	mare	1.00	mediu
10				

Tab. 1 Setul de reguli pentru calculul gradului de interes *DegInt*

În tabelul anterior se remarcă influența distanței dezbaterii față de perspectiva dezbaterii *PrspDebDist* asupra interesului pentru declanșarea unei dezbateri *ISD*.

3.3.4. Determinarea gradului de încredere *DegConf*

Al doilea parametru important care trebuie determinat astfel ca argumentul rădăcină al unei dezbateri să se transforme în credință este gradul de încredere *DegConf*. Gradul de încredere este o mărime complexă în sensul că reflectă atât încrederea fiecărui agent participant în propriile argumente cât și opinia comunității în ansamblul ei.

Definiție 3.18 Validarea socială *ValSoc* reprezintă opinia comunității referitoare la un argument. Intervalul de valori pentru *ValSoc* este $[-1, 1]$.

Validarea socială se calculează conform algoritmului descris de Cayrol [24]. În cazul algoritmului descris de Cayrol nodurile arborelui de argumentare sunt transparente, adică nu introduc nici un fel de parametru în calculul recursiv. În cadrul tezei, în determinarea gradului de încredere *DegConf*, validarea socială *ValSoc* este combinată cu alți trei parametri, conferind acestuia o consistență mult mai apropiată de realitate. Parametrii care sunt implicați în determinarea gradului de încredere *DegConf* al argumentului rădăcină sunt:

- *DegInt_i* – reprezintă gradul de interes al celorlalte argumente,
- *DegConf_i* – reprezintă gradul de încredere al celorlalte argumente,
- *Infl* – o regulă de influență dintre două argumente așa cum a fost specificată în cadrul definiției 3.13,
- *ValSoc* – validarea socială așa cum a fost specificată în cadrul definiției 3.18.

Definiție 3.19 Potențialul de participare la dezbateri *PartPot* este o măsură a percepției agentului asupra argumentului pe care îl deține.

Potențialul de participare la dezbateri *PartPot* este o variabilă lingvistică fuzzy cu termenii {*mic, mediu, mare*} definiți pe intervalul $[0,1]$.

Prin intermediul setului de reguli, se poate configura relaționarea agentului față de propriile cunoștințe, cu alte cuvinte, în ce măsură potențialul de participare la dezbateri depinde mai mult de interes sau de încredere. În cadrul tezei, ponderea gradului de încredere *DegConf* este relativ mai mare decât cea a gradului de interes *DegInt*.

#	IF		THEN	
	DegConf	DegInt	DoS	PartPot
1	mic	mic	1.00	mic
2	mic	mediu	1.00	mic
3	mic	mare	1.00	mediu
4	mediu	mic	1.00	mic
5	mediu	mediu	1.00	mediu
6	mediu	mare	1.00	mediu
7	mare	mic	1.00	mediu
8	mare	mediu	1.00	mare
9	mare	mare	1.00	mare
10				

Tab. 2 Setul de reguli pentru calculul potențialului de participare la dezbateri *PartPot*

Setul de reguli pentru calculul potențialului de participare la dezbateri *PartPot* este prezentat în tabelul anterior.

Prin introducerea potențialului de participare la dezbateri *PartPot*, în determinarea gradului de încredere *DegConf* s-a redus numărul parametrilor de la patru la trei: *PartPot*, *Infl* și *ValSoc*.

Definiție 3.20 Convingător *Conv* este o măsură a caracterului unui agent.

Convingător *Conv* este o variabilă lingvistică fuzzy cu termenii {*mic, mediu, mare*} definiți pe intervalul [0,1].

Potențialul de participare la dezbatere *PartPot* și o regulă de influență *Infl* permit modelarea puterii de convingere a unui agent *Ag*.

Setul de reguli permite modelarea atitudinii agentului în dezbatere. Acesta se poate baza pe potențialul de participare la dezbatere sau pe regula de influență. În cadrul tezei, ponderea potențialului de participare la dezbatere *PartPot* este relativ mai mare decât cea a regulii de influență *Infl*. Setul de reguli pentru calculul parametrului convingător *Conv* este prezentat în tabelul următor:

#	IF		THEN	
	Infl	PartPot	DoS	Conv
1	mediu	mic	1.00	mic
2	medium	mic	1.00	mic
3	mare	mic	1.00	mediu
4	mediu	mediu	1.00	mic
5	medium	mediu	1.00	mediu
6	mare	mediu	1.00	mediu
7	mediu	mare	1.00	mediu
8	medium	mare	1.00	mare
9	mare	mare	1.00	mare
10				

Tab. 3 Setul de reguli pentru calculul parametrului convingător *Conv*

Observație:

Valența influenței dintre două argumente poate să fie: SUSȚINERE sau ATAC, iar acest aspect este reprezentat în schema următoare prin semnul \pm . Semnul NU este influențat de valoarea validării sociale. Algoritmul definit de Cayrol [24] permite ca un argument din arbore să aibă o valoare negativă și să susțină un alt argument.

În schema următoare sunt prezentați parametrii care contribuie la determinarea validării sociale.

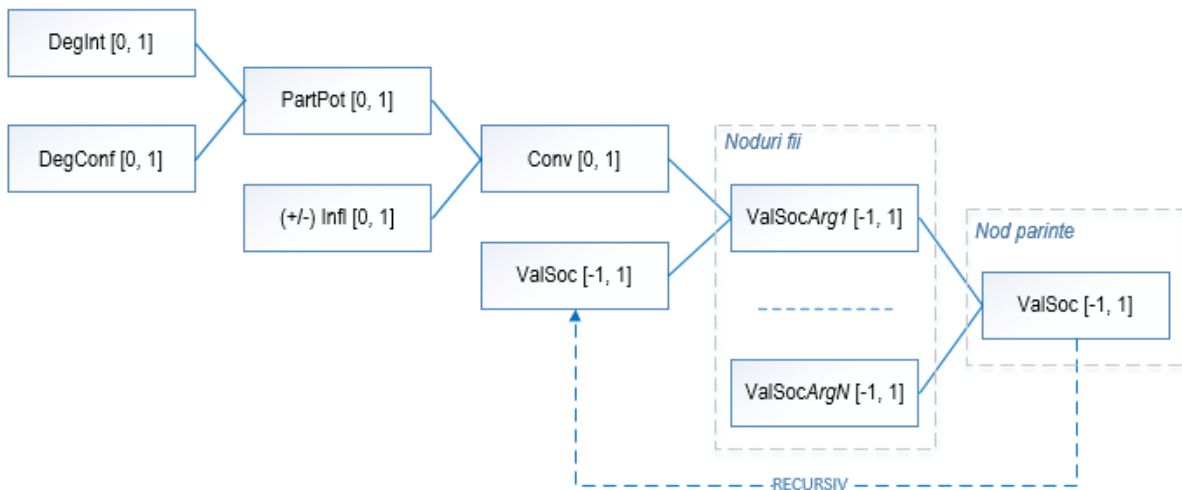


Fig. 12 Parametrii de calcul pentru determinarea *ValSoc*

În schema de mai sus se observă că validarea socială *ValSoc* este influențată de către parametrul convingător *Conv*. Acest aspect este similar cu cel întâlnit în societatea umană, în care convingerea personală poate influența opinia comunității. Spre exemplu, rolul unui discurs public este de a convinge masele de oameni. Dacă cel care susține discursul nu este sigur, nu este convingător, opinia comunității s-ar putea să fie mai puternică, iar discursul să nu-și atingă scopul. Includerea în model a acestui aspect reprezintă o contribuție importantă a tezei.

Validarea socială a argumentului $ValSoc_{ArgK}$ în valoare absolută este o variabilă lingvistică fuzzy cu termenii {mic, mediu, mare} definiți pe intervalul [0,1].

Validarea socială *ValSoc* este o valoare din intervalul [-1, 1]. Valoarea acesteia este dată de valoarea absolută $|ValSoc|$, iar valența de semnul \pm . Folosind valoarea absolută și păstrând semnul, se poate defini un sistem de inferență fuzzy ale cărui variabile lingvistice de intrare sunt *Conv* și *abs_ValSoc*.

Valoarea absolută a validării sociale *abs_ValSoc* este o variabilă lingvistică fuzzy cu termenii {mic, mediu, mare} definiți pe intervalul [0,1].

Setul de reguli permite configurarea relației dintre opinia comunității și puterea de convingere a agentului. Setul de reguli pentru determinarea valorii absolute a validării sociale a unui argument este prezentat în tabelul următor:

#	IF		THEN	
	abs_ValSoc	Conv	DoS	abs_ValSoc_ArgK
1	mic	mic	1.00	mic
2	mic	mediu	1.00	mic
3	mic	mare	1.00	mediu
4	mediu	mic	1.00	mic
5	mediu	mediu	1.00	mediu
6	mediu	mare	1.00	mare
7	mare	mic	1.00	mediu
8	mare	mediu	1.00	mare
9	mare	mare	1.00	mare
10				

Tab. 4 Setul de reguli pentru calculul *abs_ValSoc_ArgK*

Semnul valorii absolute abs_ValSoc va fi transferat către abs_ValSoc_ArgK astfel că valoarea finală $ValSoc_{ArgK}$ va aparține intervalului $[-1, 1]$.

Se remarcă faptul că validarea socială a nodului părinte, dacă acesta este un nod fiu la rândul său, va fi refolosită în mod recursiv. Odată determinată validarea socială pentru fiecare argument nod fiu, se va aplica algoritmul definit de Cayrol [24] pentru a determina validarea socială $ValSoc$ a nodului părinte. Dacă la rândul său nodul părinte este un nod fiu, atunci valoarea determinată $ValSoc$ va fi recombinaată recursiv cu parametrul convingător $Conv$.

Există situații în care un nod din arbore conține un set de argumente care influențează $Infl$ un alt argument. Conform definiției 3.13 fiecare argument din set are o pondere τ . În această situație, validarea socială pentru argumentul K se va determina ca fiind o medie ponderată:

$$ValSoc_{ArgK} = \frac{\tau_1 * ValSoc_{Arg1} + \dots + \tau_n * ValSoc_{ArgN}}{\tau_1 + \dots + \tau_n} \in [-1, 1]$$

Pentru argumentul rădăcină, gradul de încredere reprezintă valoarea absolută a validării sociale $DegConf = |ValSoc|$, iar argumentul /domeniu/categorie/Pnume($X, Pvaloare$) va SUSTINE sau va ATACA în funcție de semnul $ValSoc$.

3.4. Paralelă cu achiziția clasică

În cadrul tezei sunt definite un număr de șase etape de achiziție, etape cu caracter general și cu un grad ridicat de aplicare practică. Un sistem de achiziție care implică oameni are nevoie de rigurozitate, iar acest fapt se reflectă într-un număr mare de etape de achiziție, fiecare etapă fiind foarte bine explicată și descrisă. Teza nu lucrează cu oameni, ci cu agenți și din acest motiv numărul de etape este redus, iar specificarea fiecărei etape se reflectă în algoritm. Agenții software nu sunt atât de complecși ca cei umani și prin urmare unele etape nu au echivalent într-un proces algoritmic. Algoritmul reprezintă o specificare extrem de riguroasă a pașilor unei etape de achiziție.

În tabelul următor este prezentată maparea în cadrul etapelor de achiziție a diversilor pași ai algoritmilor.

Nr	Etapă achiziție cunoștințe	Etapă algoritm
1	Conceperea unei propuneri de proiect	Identificarea topicului dezbaterii
2	Crearea unei baze de cunoștințe pentru colectarea cunoașterii de la experți	Configurarea sistemului pentru desfășurarea argumentării
3	Identificarea surselor de cunoaștere care vor fi consultate	Algoritmul de publicare a argumentelor
4	Utilizarea de metode eficiente pentru capturarea, structurarea și validarea cunoașterii	Algoritmul de selecție și participare la dezbateri
5	Desfășurarea etapei de achiziție de cunoștințe astfel încât să fie maximizată utilizarea resurselor disponibile	Algoritmii de finalizare a unei dezbateri
6	Diseminarea bazei de cunoștințe către entitățile interesate	Analiză arbore dezbateri și comunicare rezultate

Tab. 5 Corespondența dintre etapele de achiziție de cunoștințe și etapele algoritmului

4. Contribuții privind modelarea agenților inteligenți argumentativi în vederea achiziției de cunoștințe

Sistemul ARGKAMAS este un sistem multi-agent cu un grad ridicat de complexitate. Agenții principali ai sistemului sunt:

- *AgSB* – Agentul Baza de Date Semantică,
- *AgCIB* – Agentul Baza de Date de Influențe Comune,
- *Ag* - Agent Comun,
- *AgDTR* – Agentul Registru de Dezbatere,
- *AgDT* – Agentul Arbore de Dezbatere.

Fiecare agent principal este alcătuit la rândul său dintr-un sistem de agenți interni care definesc funcționalitatea agentului principal. Fiecare agent intern este alcătuit din agenți cu roluri simple, numiți unități funcționale și de legătură. Cele trei tipuri de agenți se constituie în trei niveluri de complexitate: superior, intermediar și inferior. Totalitatea agenților și a interacțiunilor dintre aceștia alcătuiesc sistemul ARGKAMAS.

4.1. Comportamentul la nivel de sistem

Sistemul este organizat pe trei niveluri de complexitate. Nivelul superior este nivelul cu cel mai mare grad de abstractizare și este alcătuit din agenții principali ai sistemului. Nivelul intermediar descrie sistemul multi-agent care alcătuiește fiecare agent principal. Diversele funcționalități ale unui agent sunt supravegheate de către un agent specializat, numit agent intern. În acest fel, atât adăugarea de funcționalități suplimentare cât și toleranța la erori este crescută. Mai mult, anumite funcționalități care necesită resurse de calcul complexe pot fi implementate pe sisteme distribuite, fără a necesita modificarea arhitecturii interne a agentului.

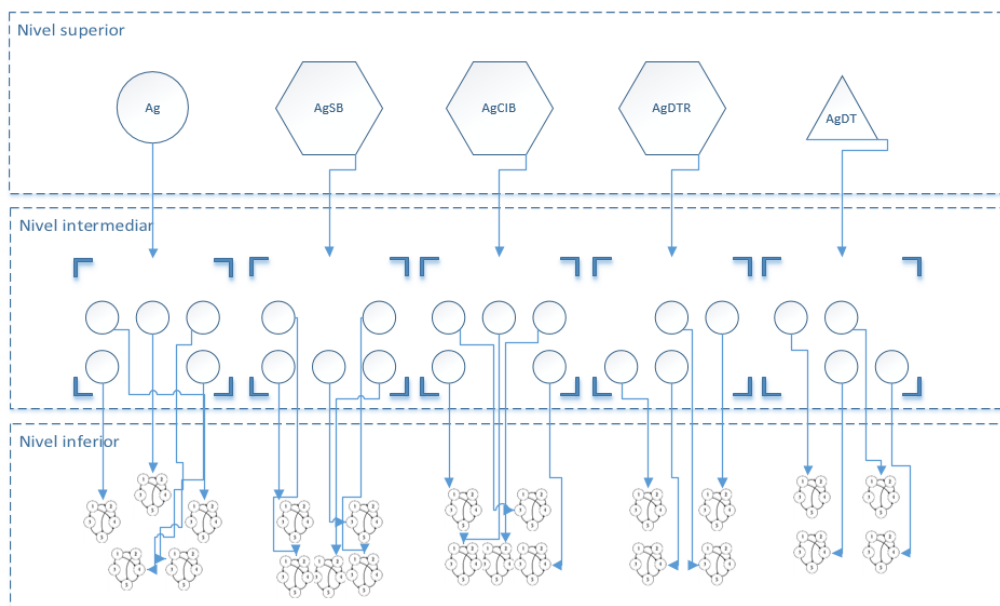


Fig. 13 Nivelurile de abstractizare ale sistemului ARGKAMAS

Nivelul inferior este tot un sistem multi-agent în care agenții îndeplinesc funcțiile de bază ale agenților interni. În cadrul acestui nivel, agenții poartă denumirea de *unități funcționale* și de *legătură*. Rolul unităților este de a prelucra și manipula informația primară a unui agent intern. Nivelul inferior este cel mai dens din punct de vedere al cantității de informație instrumentată. În figura anterioară sunt reprezentate cele trei niveluri de abstractizare. Din punct de vedere al rolurilor agenților, sistemul ARGKAMAS poate fi organizat în două componente:

- interacțiuni externe – componenta conține agenții comuni Ag care primesc cereri din exterior și comunică rezultatul dezbaterilor,
- management dezbateri – agenții $AgSB$, $AgCIB$, $AgDTR$ și $AgDT$ se ocupă de managementul dezbaterilor declanșate de agenții Ag .

În figura următoare sunt reprezentate cele două componente:

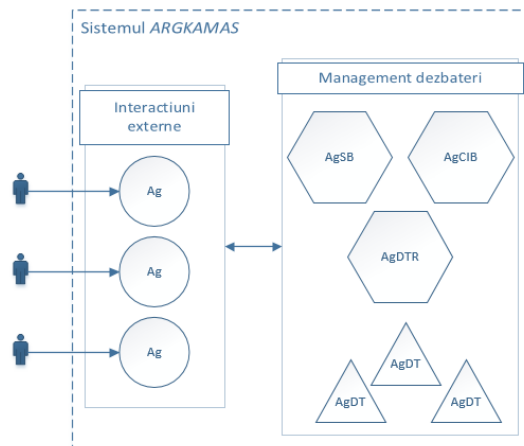


Fig. 14 Componentele *interacțiuni externe* și *management dezbateri* ale sistemului ARGKAMAS

În figura următoare sunt reprezentate interacțiunile dintre agenți în cadrul nivelului superior. Interacțiunile au loc prin intermediul mesajelor de tip *request* și *reply*. Acestea sunt transformate în cadrul comunicațiilor dintre agenții interni ai unui agent principal în mesaje de tip *ask* și respectiv *answer*.

Acest subcapitol modelează interacțiunea unei entități exterioare (agent uman) cu sistemul ARGKAMAS. Alte tipuri de interacțiuni, precum popularea bazelor de cunoștințe, reprezintă direcții viitoare de cercetare.

O entitate externă Ent emite o cerere de dezbateri pentru un anumit topic. Cererea este trimisă agentului Ag sub forma $request(debate(topic))$.

Agentul Ag trimite o cerere de tip $request(seek(topic))$ către agentul $AgSB$ pentru a identifica caracteristicile argumentabile corespunzătoare topicului. Topicurile identificate sunt trimise înapoi către agentul Ag sub forma unui răspuns $reply(seek(topic), \{Crt1, \dots\})$, în care Crt este o caracteristică argumentabilă. Forma $\{Crt1, \dots\}$ reprezintă una sau mai multe astfel de caracteristici.

În momentul primirii caracteristicilor argumentabile, agentul Ag declanșează dezbateri pentru fiecare dintre acestea prin înregistrarea lor la agentul $AgDTR$. Forma cererii este $request(register(topic, \{Crt1, \dots\}))$. Pentru fiecare dezbateri înregistrată $AgDTR$ va crea o nouă instanță a agentului $AgDT$, instanță care va superviza desfășurarea dezbaterii. Forma cererii este $request(create(AgDT(topic, Crt1)))$. La un anumit moment, în sistem pot exista milioane de agenți $AgDT$ activi.

Comportamentul agentului $AgDT$ este diferit de al celorlalți agenți în sensul că acesta cere de la fiecare Ag , prin publicarea argumentului, și așteaptă un răspuns. Cererea către agenți are forma $request(argpub(topic))$.

Pentru fiecare argument, agentul Ag încearcă să identifice credințe și influențe între acestea. Cererile de identificare de influențe către agentul $AgCIB$ au forma $request(infl(Arg1, \dots))$. Răspunsul primit de la agentul $AgCIB$ are forma $reply(infl(Arg1, \dots), \{Arg, InflVal, DegInfl\})$ și conține argumentul care este influențat, valența influenței și gradul de influență.

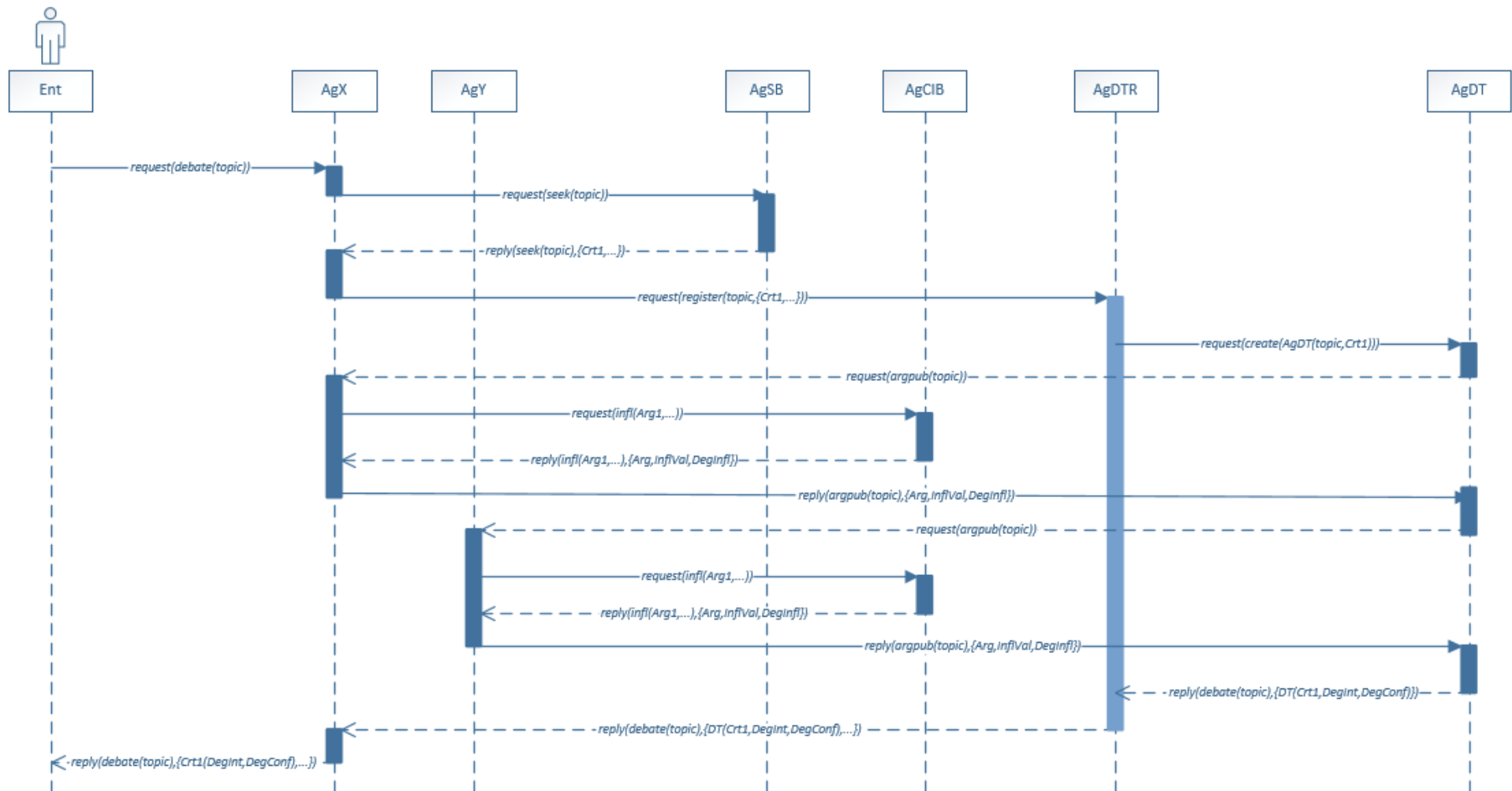


Fig. 15 Interacțiunile dintre agenți în cadrul nivelului superior

La finalizarea dezbaterii, agentul $AgDT$ trimite rezultatul către agentul $AgDTR$ care distruge agentul $AgDT$ și transmite rezultatul către agentul Ag . Acesta informează entitatea Ent , care a emis cererea inițială de dezbateri, despre rezultatul dezbaterii. Forma răspunsului este $reply(debate(topic), \{Crt1(DegInt, DegConf), \dots\})$.

Modelarea agenților este realizată în conformitate cu formalismul propus de Parsons și Sabater [27], [28]. Formalismul permite modelarea nivelurilor intermediar și inferior și poate fi transferat relativ ușor într-un sistem informatic real. De asemenea, formalismul permite extinderi viitoare ale specificațiilor agenților cu diverse alte teorii și formalisme.

4.2. Modelarea agenților

Fiecare agent principal este modelat la rândul său ca un sistem multi-agent. Totalitatea acestora formează nivelul intermediar. În cadrul sistemului multi-agent, comunicarea se realizează exclusiv prin intermediul mesajelor. Din acest motiv, fiecare agent conține un manager de relații sociale SM care are rolul de a ruta mesajele atât între agenții principali cât și între agenții care alcătuiesc structura internă a unui agent principal. Deoarece sistemele multi-agent care alcătuiesc agenții principali sunt descrise în termeni de Credințe, Dorințe și Intenții, fiecare agent principal conține un agent manager de scop GM , scopul reprezentând o dorință care trebuie realizată. În alcătuirea agentului principal se regăsește și agentul manager de planuri PM care cunoaște etapele necesare pentru îndeplinirea unui scop. Ceilalți agenți care alcătuiesc sistemul multi-agent al agentului principal îndeplinesc sarcini specifice.

4.2.1. Agentul bază de date semantică $AgSB$

Agentul $AgSB$ deține baza semantică de cunoștințe necesară unui agent Ag pentru declanșarea dezbaterilor. Din punct de vedere funcțional, agentul $AgSB$ îndeplinește două roluri esențiale:

- comunică agenților interesați perspectivele principale și proprietățile argumentabile,
- poate calcula distanța dintre două perspective.

Conform definiției 3.4 agentul $AgSB$ stochează proprietățile argumentabile sub formă de caracteristici:

$$Crt = /domeniu/categorie/Pnume/Pvaloare$$

Schema funcțională a agentului $AgSB$ este prezentată în figura următoare:

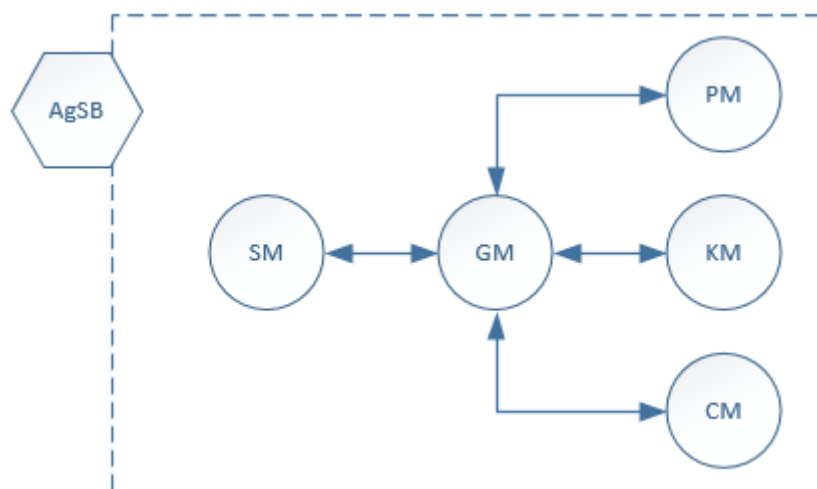


Fig. 16 Schema funcțională a agentului bază de date semantică $AgSB$

Sistemul multi-agent *AgSB* este alcătuit din următorii agenți:

- *SM* – agentul intern manager de relații sociale. Acesta primește mesajele din exterior și le transmite agentului intern *GM*,
- *GM* – agentul intern manager de scop. Acesta gestionează scopurile (dorințele) agentului *AgSB* și le monitorizează până la îndeplinire,
- *PM* – agentul intern manager de planuri. Acesta furnizează planurile necesare pentru îndeplinirea unui scop,
- *KM* – agentul intern manager de cunoștințe. Acesta stochează baza semantică de date și realizează operațiile de căutare în aceasta,
- *CM* – agentul intern manager computațional. Acesta calculează la cerere distanța dintre două caracteristici *Crt*.

În cazul în care agentul *AgSB* trebuie să caute caracteristici despre un topic, cererea este primită de către agentul intern *SM*. Acesta indexează cererea pentru a o putea identifica ulterior și apoi o transmite către agentul intern *GM*. Acesta o transformă în scop și trimite scopul către agentul *PM* pentru identificarea unui plan de realizare al scopului. După obținerea planului, acesta este transmis către agentul intern *KM* care realizează căutarea și obține rezultatele. Rezultatele sunt transmise către agentul *GM* care le trimite sub formă de răspuns către agentul *SM*. Acesta le trimite către destinatarul cererii de căutare. În figura următoare este reprezentată diagrama de secvențe pentru căutarea de caracteristici *Crt*.

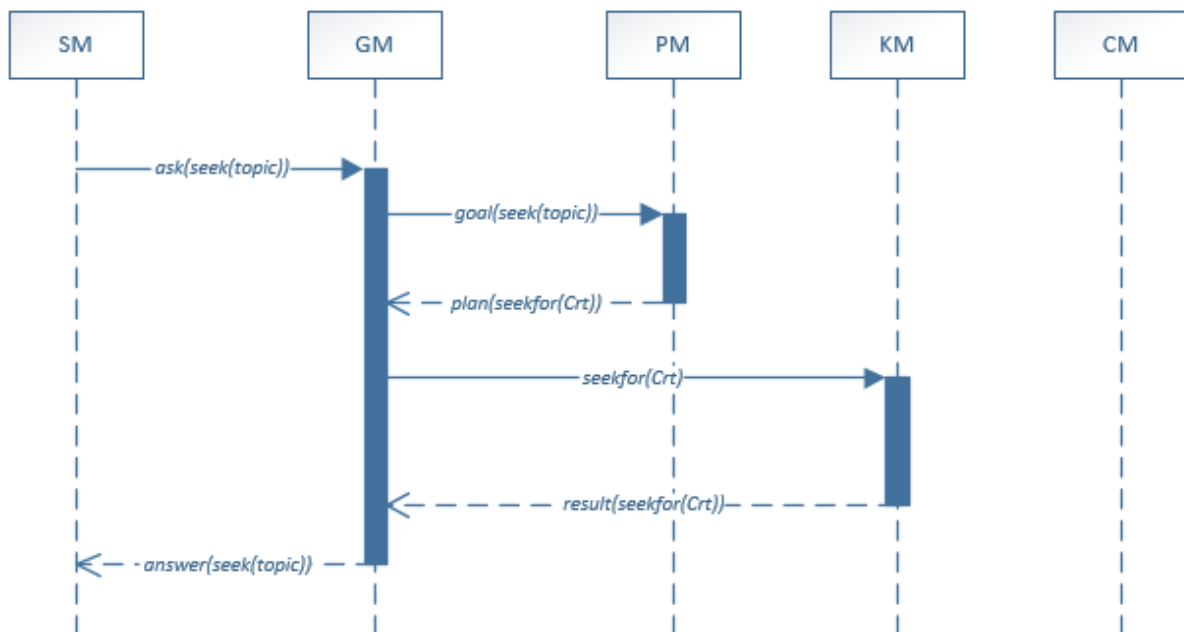


Fig. 17 Diagrama de secvențe pentru căutarea de caracteristici *Crt*

În cazul în care agentul *AgSB* primește o cerere de calcul de distanță între două caracteristici, agentul *GM* primește un plan de realizare a scopului de calcul, iar agentul *CM* va realiza calculul efectiv. În figura următoare este reprezentată diagrama de secvențe pentru calculul distanței.

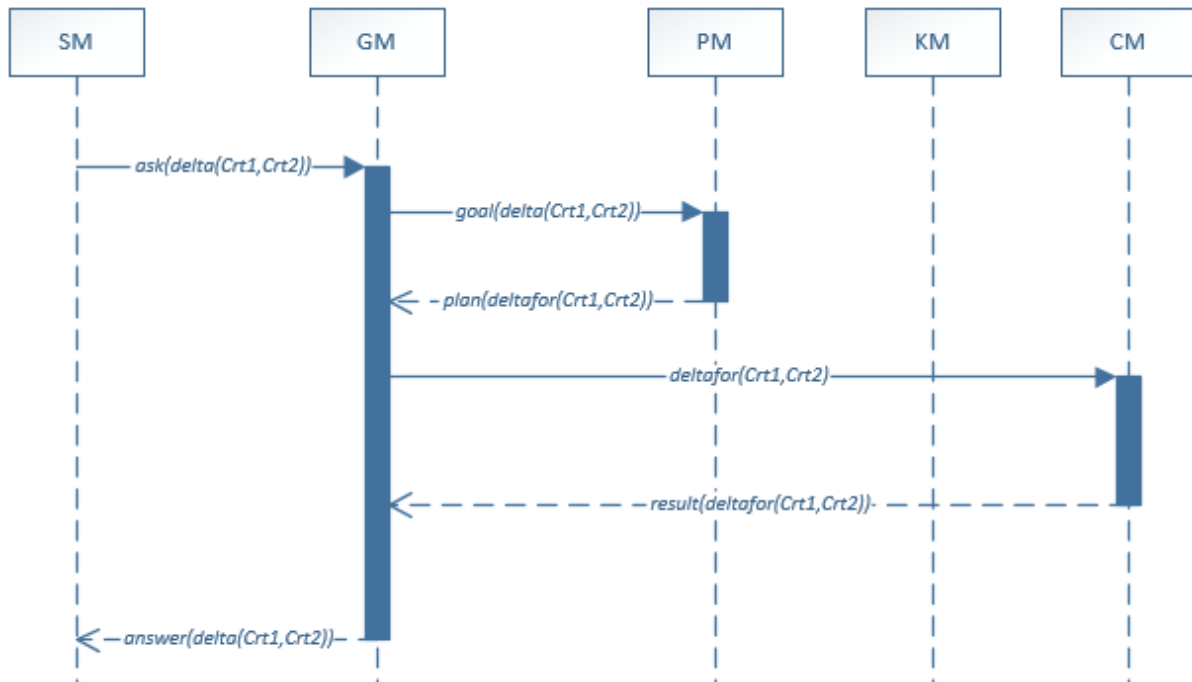


Fig. 18 Diagrama de secvențe pentru calculul distanței dintre două caracteristici

Agentul intern manager de relații sociale *SM* are rolul de a transfera mesaje atât intern cât și extern. Mesajele provenite din exterior au forma *request(mesaj)* fiind transformate în cereri interne de forma *ask(mesaj)*, iar răspunsurile interne de forma *answer(rezultat)* vor fi transformate în mesaje externe de forma *reply(rezultat)*.

Agentul intern *SM* este alcătuit din următoarele unități:

- *CU* – unitatea funcțională cu rol de primire și transmitere de mesaje interne și externe,
- *IdxG* – unitate funcțională cu rol de generator de indecși,
- *IdxDB* – unitate funcțională cu rol de bază de date pentru indecși. Un index face legătura dintre un agent comun *Ag* și *AgSB*,
- *GET* – unitate de legătură

Schema funcțională a agentului *SM* este prezentată în figura următoare:

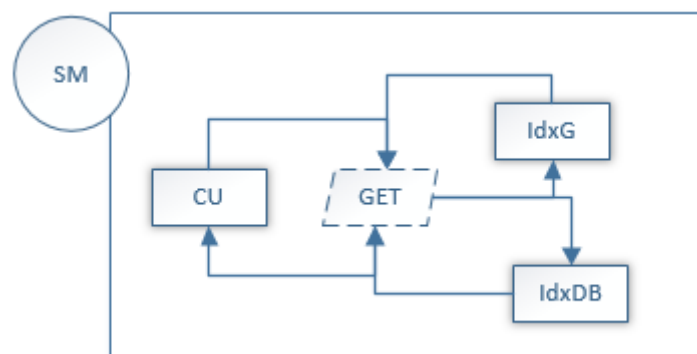


Fig. 19 Schema funcțională a agentului intern *SM* al *AgSB*

La primirea unei cereri externe de la agentul *Ag* de căutare a caracteristicilor unui topic, sub forma *request(seek(topic))*, este generat un index *idx* pentru memorarea referinței către

Contribuții privind modelarea agenților inteligenți argumentativi în vederea achiziției de cunoștințe

agentul Ag și cererea externă este transformată într-o cerere internă de tip $ask(de\ la,\ catre,\ mesaj)$. Cererea internă este transmisă către agentul intern GM .

$$GET = \frac{CU > request(Ag, AgSB/SM, seek(topic))}{\frac{IdxG: idx(seek(topic))}{CU > ask(self/SM, self/GM, ask(idx, seek(topic)))}} \quad IdxDB: set(idx, seek(topic))$$

Răspunsul are forma $result(idx, mesaj, valori)$ în care *valorile* sunt rezultatele la cererea de tip *mesaj*. Unitatea funcțională CU transformă răspunsul în mesaj extern de tip $reply(de\ la,\ catre,\ rezultat)$ și îl trimite agentului Ag care a emis cererea inițială.

$$GET = \frac{CU > answer(self/GM, self/SM, result(idx, seek(topic), \{Crt1, \dots\}))}{\frac{IdxDB: get(idx)}{CU > reply(AgSB/SM, Ag, result(seek(topic), \{Crt1, \dots\}))}}$$

În mod similar, la primirea unei cereri de calculare a distanței dintre două caracteristici, agentul SM indexează cererea și o trimite spre rezolvare către agentul GM .

$$GET = \frac{CU > request(Ag, AgSB/SM, delta(Crt1, Crt2))}{\frac{IdxG: idx(delta(Crt1, Crt2))}{CU > ask(self/SM, self/GM, ask(idx, delta(Crt1, Crt2)))}} \quad IdxDB: set(idx, delta(Crt1, Crt2))$$

La primirea rezultatului δ răspunsul este transformat în mesaj extern și este transmis către agentul Ag .

$$GET = \frac{CU > answer(self/GM, self/SM, result(idx, delta(Crt1, Crt2), \{\delta\}))}{\frac{IdxDB: get(idx)}{CU > reply(AgSB/SM, Ag, result(delta(Crt1, Crt2), \{\delta\}))}}$$

Agentul intern GM transformă cererile de căutare și calculare în scopuri. Pentru aceste scopuri trebuie să identifice un plan care să permită rezolvarea lor. Odată identificat planul, acesta este pus în aplicare, iar la obținerea rezultatului, scopul este eliminat. În figura următoare este prezentată schema funcțională a agentului intern GM :

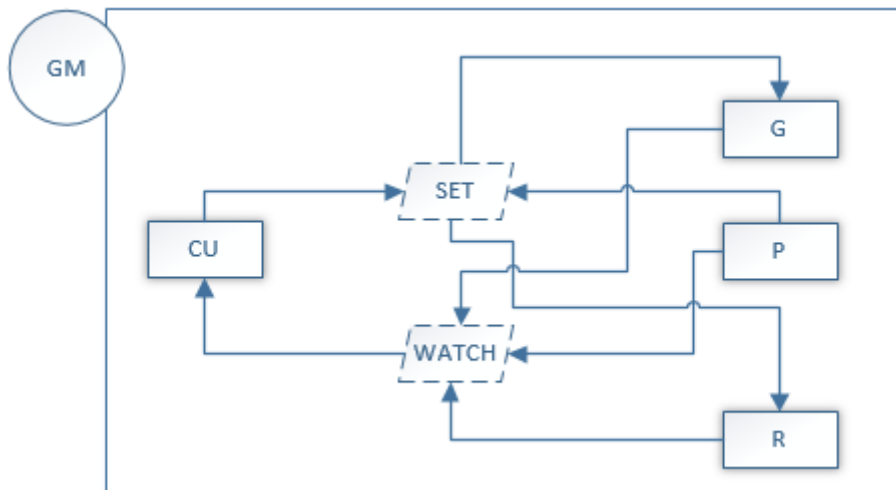


Fig. 20 Schema funcțională a agentului intern GM al $AgSB$

Agentul intern *GM* este alcătuit din următoarele unități:

- *CU* unitate funcțională cu rol de primire și transmitere de mesaje interne și externe,
- *G* – unitate funcțională cu rol de stocare a scopurilor de îndeplinit,
- *P* – unitatea funcțională pentru stocarea planurilor. Un plan reprezintă modalitatea de realizare a unui scop,
- *R* – unitatea funcțională pentru stocarea diverselor rezultate,
- *SET* și *WATCH* – unități de legătură

Cererea *ask(de la, catre, mesaj)* este transformată în scop și stocată în unitatea funcțională *G*.

$$SET = \frac{CU > ask(\overset{self}{/}_{SM}, \overset{self}{/}_{GM}, ask(idx, seek(topic)))}{G: goal(idx, seek(topic))}$$

Atunci când este primit un răspuns de la agentul intern *KM*, unitatea de legătură *SET* va trimite rezultatul către unitatea funcțională *R*.

$$SET = \frac{CU > answer(\overset{self}{/}_{KM}, \overset{GM}{/}_{R}, result(seekfor(Crt), \{Crt1, \dots\}))}{\frac{P: plan(seek(topic), \{seekfor(Crt)\})}{R: result(seek(topic), \{Crt1, \dots\})}}$$

În mod similar se desfășoară lucrurile și pentru cereri de calcul.

$$SET = \frac{CU > ask(\overset{self}{/}_{SM}, \overset{self}{/}_{GM}, ask(idx, delta(Crt1, Crt2)))}{G: goal(idx, delta(Crt1, Crt2))}$$

La primirea rezultatului calculului distanței dintre două caracteristici, de la agentul intern *CM*, acesta este stocat în unitatea funcțională *R*.

$$SET = \frac{CU > answer(\overset{self}{/}_{CM}, \overset{GM}{/}_{R}, result(deltafor(Crt1, Crt2), \{\delta\}))}{\frac{P: plan(delta(Crt1, Crt2), \{deltafor(Crt1, Crt2)\})}{R: result(delta(Crt1, Crt2), \{\delta\})}}$$

Unitatea de legătură *WATCH* monitorizează unitatea funcțională *G* pentru apariția de scopuri. Atunci când apare un nou scop, se verifică dacă există un plan pentru realizarea acestuia și dacă nu, este cerut un astfel de plan de la agentul intern *PM* al agentului principal *AgSB*. Planul obținut va fi stocat în unitatea funcțională *P*.

$$WATCH = \frac{\frac{G: goal(idx, seek(topic))}{P: \neg plan(seek(topic), \{\})}}{CU > ask(\overset{GM}{/}_{P}, \overset{self}{/}_{PM}, plan(seek(topic)))}$$

Unitatea de legătură *WATCH* monitorizează și scopurile de calcul de distanță.

$$WATCH = \frac{\frac{G: goal(idx, delta(Crt1, Crt2))}{P: \neg plan(delta(Crt1, Crt2), \{\})}}{CU > ask(\overset{GM}{/}_{P}, \overset{self}{/}_{PM}, plan(delta(Crt1, Crt2)))}$$

În momentul în care apare un plan pentru realizarea scopului, se trimite o cerere către agentul intern manager de cunoștințe *KM*, care va trimite și executa cererea. Rezultatul primit este stocat în unitatea funcțională *R*.

$$WATCH = \frac{G: goal(idx, seek(topic))}{P: plan(seek(topic), \{seekfor(Crt)\})}$$

$$CU > ask(GM/R, self/KM, seekfor(Crt))$$

Pentru calculul distanței, cererea se va trimite către agentul intern *CM*.

$$WATCH = \frac{G: goal(idx, delta(Crt1, Crt2))}{P: plan(delta(Crt1, Crt2), \{deltafor(Crt1, Crt2)\})}$$

$$CU > ask(GM/R, self/CM, deltafor(Crt1, Crt2))$$

În momentul în care unitatea de legătură *WATCH* detectează existența rezultatelor pentru scopul respectiv, trimite răspunsul către agentul intern manager de relații sociale *SM*. După trimiterea răspunsului, vor fi șterse scopul, planul și rezultatele.

$$WATCH = \frac{G: goal(idx, seek(topic))}{P: plan(seek(topic), \{seekfor(Crt)\})}$$

$$R: result(seek(topic), \{Crt1, \dots\})$$

$$CU > answer(self/GM, self/SM, result(idx, seek(topic), \{Crt1, \dots\}))$$

$$G: remove(goal(idx, seek(topic)))$$

$$P: remove(plan(seek(topic), \{seekfor(Crt)\}))$$

$$R: remove(result(seek(topic), \{Crt1, \dots\}))$$

În mod similar se desfășoară lucrurile și în momentul când există un rezultat pentru distanța dintre două caracteristici.

$$WATCH = \frac{G: goal(idx, delta(Crt1, Crt2))}{P: plan(delta(Crt1, Crt2), \{deltafor(Crt1, Crt2)\})}$$

$$R: result(delta(Crt1, Crt2), \{\delta\})$$

$$CU > answer(self/GM, self/SM, result(idx, delta(Crt1, Crt2), \{\delta\}))$$

$$G: remove(goal(idx, delta(Crt1, Crt2)))$$

$$P: remove(plan(delta(Crt1, Crt2), \{deltafor(Crt1, Crt2)\}))$$

$$R: remove(result(delta(Crt1, Crt2), \{\delta\}))$$

Agentul intern *PM* este alcătuit din următoarele unități:

- *CU* – unitatea funcțională cu rol de primire și transmitere de mesaje interne și externe,
- *P* – unitatea funcțională de stocare a planurilor,
- *GET* – unitate de legătură.

Schema funcțională a agentului *PM* este prezentată în figura următoare:

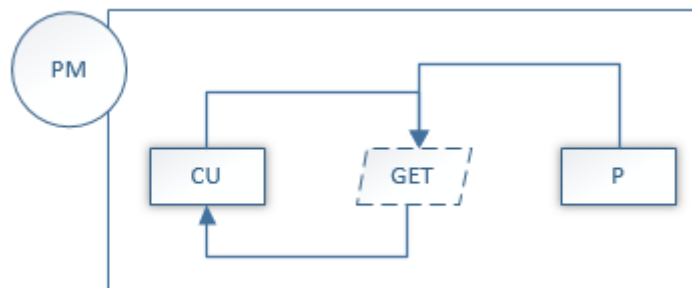


Fig. 21 Schema funcțională a agentului intern *PM* al AgSB

Rolul agentului intern PM este acela de a stoca planurile de realizare pentru diversele scopuri ale agentului principal $AgSB$. Astfel, la apariția unei cereri de plan pentru $seek(topic)$, unitatea de legătură GET va identifica planul corespunzător și-l va trimite către agentul intern GM .

$$GET = \frac{CU > ask(GM/p, self/p_M, plan(seek(topic)))}{CU > answer(self/p_M, GM/p, plan(seek(topic), \{seekfor(Crt)\}))}$$

În mod similar este trimis planul de calcul a distanței dintre două caracteristici.

$$GET = \frac{CU > ask(GM/p, self/p_M, plan(delta(Crt1, Crt2)))}{CU > answer(self/p_M, GM/p, plan(delta(Crt1, Crt2), \{deltafor(Crt1, Crt2)\}))}$$

Agentul intern KM este alcătuit din următoarele unități:

- CU – unitatea funcțională cu rol de primire și transmitere de mesaje interne și externe,
- K – unitate funcțională pentru stocarea și interogarea cunoașterii,
- GET – unitate de legătură

Schema funcțională a agentului KM este prezentată în figura următoare:

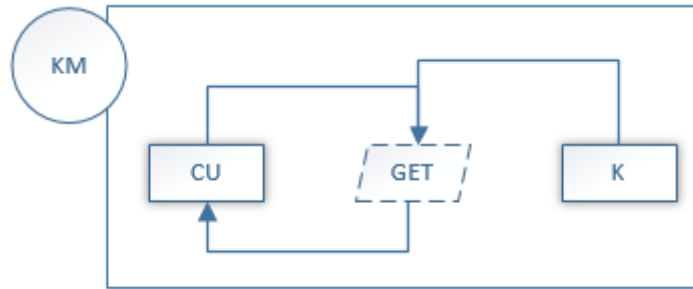


Fig. 22 Schema funcțională a agentului intern KM al $AgSB$

Rolul agentului intern KM este de a stoca și interoga baza de cunoștințe. Răspunsul la o cerere de tip $seekfor(Crt)$ este o listă de caracteristici.

$$GET = \frac{CU > ask(GM/R, self/K_M, seekfor(Crt))}{CU > answer(self/K_M, GM/R, result(seekfor(Crt), \{Crt1, \dots\}))}$$

Agentul intern CM este alcătuit din următoarele unități:

- CU – unitatea funcțională cu rol de primire și transmitere de mesaje interne și externe,
- C – unitate funcțională pentru calculul distanței dintre două caracteristici,
- GET – unitate de legătură

Schema funcțională a agentului CM este prezentată în figura următoare:

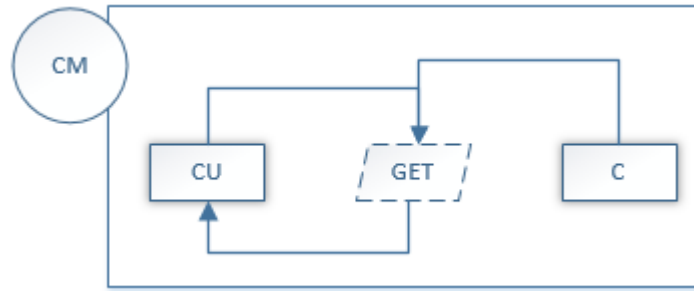


Fig. 23 Schema funcțională a agentului intern *CM* al *AgSB*

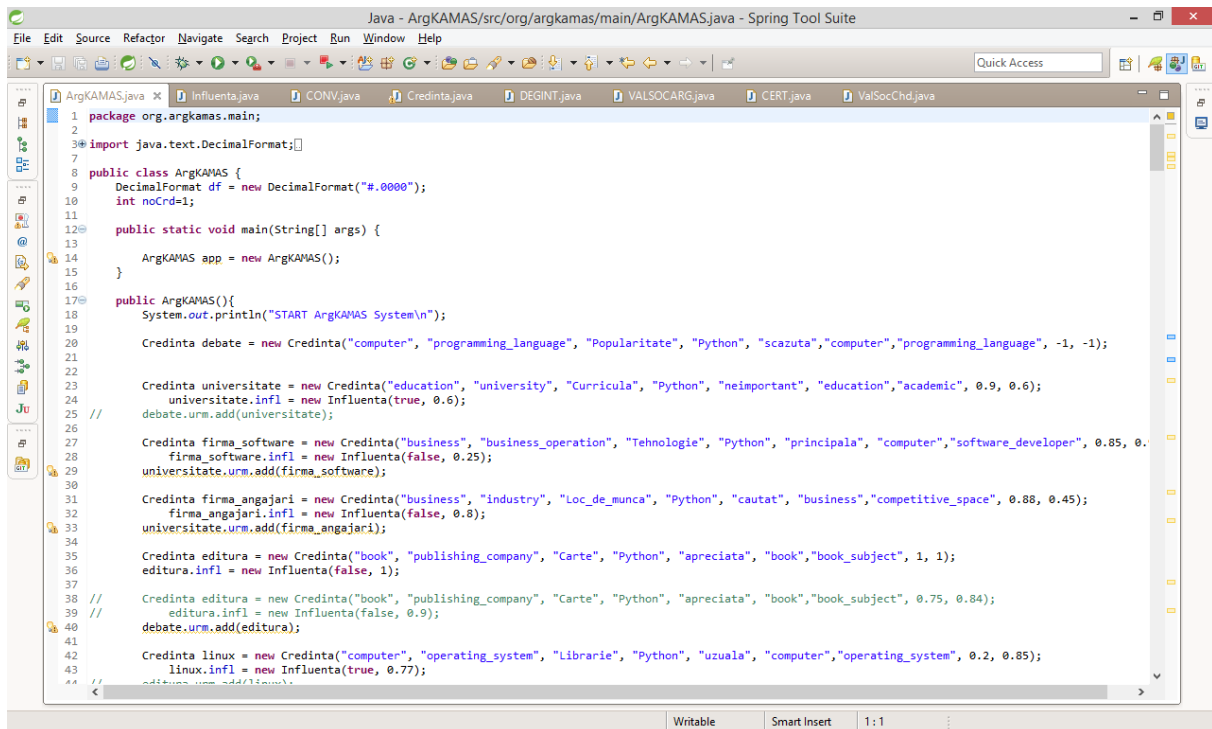
Rolul agentului intern *CM* este acela de a calcula distanța dintre două caracteristici. Răspunsul la o cerere de tip $\text{deltafor}(Crt1, Crt2)$ este distanța δ .

$$GET = \frac{CU > ask(GM/R, self/CM, \text{deltafor}(Crt1, Crt2))}{CU > answer(self/CM, GM/R, \text{result}(\text{deltafor}(Crt1, Crt2), \{\delta\}))}$$

5. Contribuții experimentale și aplicative

Scopul unei dezbateri este acela de a determina gradul de interes pentru dezbateri $DegInt$ și gradul de încredere în dezbateri $DegConf$.

Aplicația Java pentru determinarea gradelor de interes și încredere este prezentată în figura următoare:



```

1 package org.argkamas.main;
2
3 import java.text.DecimalFormat;
4
5
6
7
8 public class ArgKAMAS {
9     DecimalFormat df = new DecimalFormat("#.0000");
10    int noCrd=1;
11
12    public static void main(String[] args) {
13
14        ArgKAMAS app = new ArgKAMAS();
15    }
16
17    public ArgKAMAS(){
18        System.out.println("START ArgKAMAS System\n");
19
20        Credinta debate = new Credinta("computer", "programming_language", "Popularitate", "Python", "scazuta", "computer", "programming_language", -1, -1);
21
22
23        Credinta universitate = new Credinta("education", "university", "Curricula", "Python", "neimportant", "education", "academic", 0.9, 0.6);
24        universitate.infl = new Influenta(true, 0.6);
25        // debate.urm.add(universitate);
26
27        Credinta firma_software = new Credinta("business", "business_operation", "Tehnologie", "Python", "principala", "computer", "software_developer", 0.85, 0.25);
28        firma_software.infl = new Influenta(false, 0.25);
29        universitate.urm.add(firma_software);
30
31        Credinta firma_angajari = new Credinta("business", "industry", "Loc_de_munca", "Python", "cautat", "business", "competitive_space", 0.88, 0.45);
32        firma_angajari.infl = new Influenta(false, 0.8);
33        universitate.urm.add(firma_angajari);
34
35        Credinta editura = new Credinta("book", "publishing_company", "Carte", "Python", "apreciata", "book", "book_subject", 1, 1);
36        editura.infl = new Influenta(false, 1);
37
38        Credinta editura = new Credinta("book", "publishing_company", "Carte", "Python", "apreciata", "book", "book_subject", 0.75, 0.84);
39        editura.infl = new Influenta(false, 0.9);
40        // debate.urm.add(editura);
41
42        Credinta linux = new Credinta("computer", "operating_system", "Librarie", "Python", "uzuala", "computer", "operating_system", 0.2, 0.85);
43        linux.infl = new Influenta(true, 0.77);
44        // editura.urm.add(linux);

```

Fig. 24 Aplicația Java pentru determinarea gradelor

Rezultatele finale sunt următoarele:

- $DegInt = 0.3$
- $DegConf = 0.2959$

Forma finală a credinței noi achiziționate de către agentul Ag este:

$$Bel = \left\langle \begin{array}{l} \neg / \text{computer} / \text{programming_language} / \text{Popularitate}(\text{Python}, \text{scazuta}), \\ / \text{computer} / \text{programming_language}, \\ 0.3 \\ 0.2959 \end{array} \right\rangle$$

Cu această credință, agentul Ag va putea participa și la alte dezbateri.

5.1. Comparație cu modelul propus de Cayrol

Pentru a putea înțelege semnificația acestor valori, vom considera o situație simplificată, în care argumentul rădăcină este ATACAT de un singur argument, cu valorile gradelor de interes, încredere și influență la limită.

În figura următoare este prezentat arborele de dezbateri considerat:



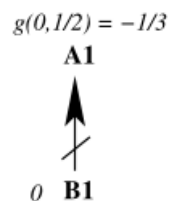
Fig. 25 Arbore de dezbateri cu un argument care ATACĂ

Considerăm cazul în care argumentul agentului $Ag(editura)$ are gradul de interes $DegInt = 0$ și gradul de încredere $DegConf = 0$. Influența are gradul $DegInfl = 0$.

Dacă toți parametrii sunt la valoarea minimă, atunci sistemul se comportă conform formalismului propus de Cayrol [24], adică:

$$DegConf = |g(0,1/2)| = 0.3333$$

Se remarcă situația similară în exemplul dat de Cayrol [24]:



Față de cadrul propus de Cayrol, modelul dezvoltat în teză este mai realist, în sensul că nodurile au informație suplimentară, fiecare agent introducând propriile influențe în sistem.

Dacă toți parametrii sunt la valoarea maximă, valoarea gradului de încredere în dezbateri al agentului Ag este:

$$DegConf = 0.4286$$

În concluzie, în condițiile existenței unui singur argument, domeniul de valori pentru gradul de încredere este situat în intervalul $DegConf \in [0.3333, 0.4286]$.

5.2. Rezultate experimentale

Experimentul desfășurat în cadrul tezei are ca scop compararea și interpretarea rezultatelor algoritmului de calcul cu cele ale chestionarului la care au participat subiecți umani. Experimentul realizat a permis înțelegerea mai bună a fenomenelor din cadrul modelului și a direcțiilor viitoare de cercetare.

În cadrul chestionarului există două tipuri de perturbații, foarte dificil de eliminat. Studiul acestor perturbații, care țin de natura umană a subiecților și posibilitățile de înglobare a lor în cadrul modelului, reprezintă direcții viitoare de cercetare. Perturbațiile identificate sunt:

- translatarea arborelui de dezbateri în limbaj natural
- interpretarea textului de către subiectul uman

Pentru realizarea experimentului, arborele de dezbateri *Popularitate(Python, scazuta)* a fost tradus în limbaj natural. Acest lucru a fost necesar deoarece s-a constatat că este imposibil pentru un subiect uman să manipuleze o formă arborescentă de o asemenea complexitate. S-a constatat că un subiect uman are dificultăți inclusiv cu arbori de mici dimensiuni, deoarece reprezentarea mentală este limitată. Traducerea în limbaj natural a fost realizată prin intermediul unor expresii care să redea cât mai exact gradul de influență dintre argumente. Totuși, percepția traducerii depinde nu numai de exactitatea traducerii ci și de felul în care este interpretată de către subiectul uman. Cei doi parametri (translatarea și interpretarea) introduc perturbații în sistem. Translatarea s-a realizat într-un limbaj cât mai simplu, iar scenariul este familiar unui student de la specializarea Calculatoare.

Datele culese în urma experimentului sunt prezentate în cadrul **Anexei 1**. Analiza realizată este prezentată în figura următoare:

Afirmația 1: Aplicațiile Python au o siguranță scăzută			
	mic	mediu	mare
Pro	6	17	8
Contra	4	14	1

Afirmația 2: Învățarea limbajului Python este facilă			
	mic	mediu	mare
Pro	1	19	26
Contra	1	3	0

Afirmația 3: Popularitatea limbajului Python este scăzută			
	mic	mediu	mare
Pro	3	5	0
Contra	7	19	16

Fig. 26 Analiza datelor experimentale

În cadrul analizei s-au cumulat opiniile *Pro* și *Contra* ale celor trei afirmații în funcție de gradul de încredere.

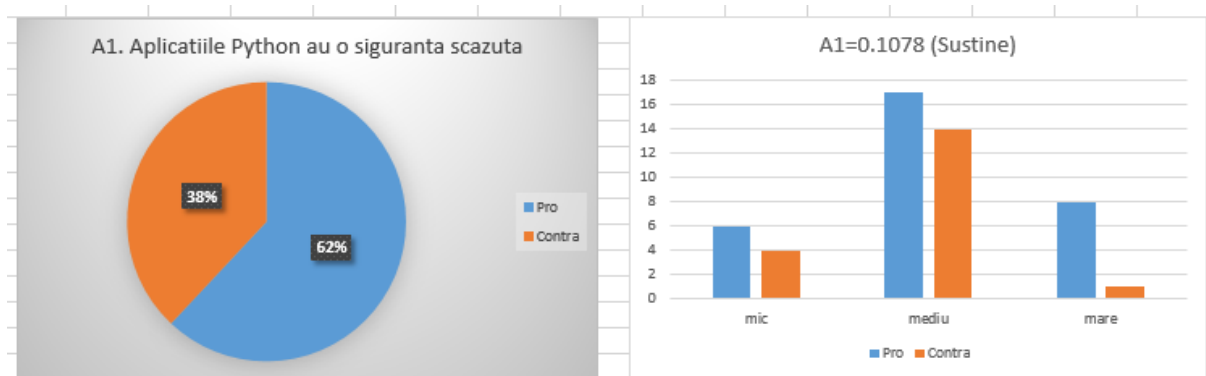


Fig. 27 Reprezentarea grafică a analizei afirmației 1

Așa cum a fost precizat, $DegConf$ reprezintă gradul de încredere în dezbateri și în final în opinia comunității. În cadrul celor două grafice sunt reprezentate proporția de voturi de susținere și de atac și distribuția acestora în funcție de gradul de încredere.

Considerăm $\delta \in [0, N]$ ca fiind diferența dintre voturile de susținere și cele de atac. Numărul de participanți la dezbateri este notat cu N . Cu cât δ este mai mic, cu atât încrederea agentului Ag în opinia comunității este mai mică.

Astfel, avem pentru fiecare grad de încredere:

$$\delta(mic) = 6 - 4 = 2$$

$$\delta(mediu) = 17 - 14 = 3$$

$$\delta(mare) = 8 - 1 = 7$$

Se calculează media ponderată MP a diferențelor δ , în care ponderea reprezintă procentul de opinii pro și contra din numărul total de opinii, pentru fiecare grad de încredere. Rezultatul obținut se reduce la intervalul $[0, 1]$.

$$MP(A1) = 2 * 0.2 + 3 * 0.62 + 7 * 0.18 = 3.52$$

$$MP_R(A1) = MP(A1)/N = 3.52/50 = 0.0704$$

Rezultatele experimentale confirmă algoritmul propus în teză. Comunitatea susține prima afirmație dar agentul are o încredere scăzută în opinia acesteia.

Reprezentarea grafică trebuie interpretată astfel: dacă un subiect uman ar privi rezultatele și graficele, ar avea încredere în opinia comunității? Răspunsul furnizat de către agentul Ag este $DegConf = 0.1078$. Cu alte cuvinte, cu cât opinia comunității este mai echilibrată, cu atât agentul are un grad mic de încredere în aceasta. Dacă un număr de cinci persoane susțin o afirmație și un alt număr de cinci persoane atacă respectiva afirmație, agentul nu poate fi sigur de afirmația respectivă. Spre deosebire de modelul Cayrol [24], algoritmul propus în teză include și alți parametri referitori la cunoașterea deținută de persoanele respective, astfel că este posibil ca rezultatul să nu fie perfect echilibrat, chiar dacă numărul de persoane care susțin și atacă este același. Ca direcție viitoare de cercetare se vor analiza limitele intervalului și variațiile de valori pentru a clarifica dacă valoarea 0.1078 este mică sau foarte mică.

Se observă că rezultatul este de SUSTINERE și este similar cu reprezentarea.

În mod similar, se pot interpreta rezultatele și pentru celelalte două afirmații.

5.3. Concluzii

Comparația dintre rezultatele execuției algoritmului și cele experimentale este prezentată în figura următoare:

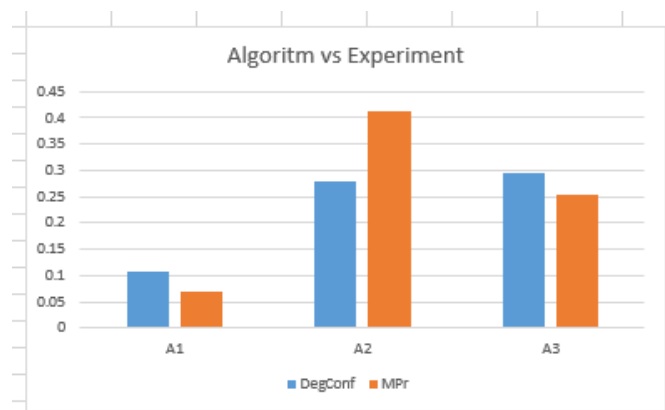


Fig. 28 Comparație între algoritm și experiment

Realizând o comparație între rezultatele experimentale și rezultatele algoritmului, se remarcă faptul că în cazul afirmației 2, gradul de încredere $DegConf$ este mai mic decât în cazul afirmației 3, dar reprezentarea grafică arată o comunitate cu pondere mai mare în cazul afirmației 2 decât al afirmației 3.

Acest fapt poate fi explicat prin existența perturbațiilor din sistem datorate traducerii arborelui în limbaj natural și a interpretării factorului uman. Rezultatele experimentale nu reflectă în mod explicit încrederea și interesul față de propriile cunoștințe și nici influența opiniilor celorlaltora vis-a-vis de propria opinie.

Realizarea experimentului a scos în evidență faptul că este dificil de analizat statistic și reprezentat grafic complexitatea gradului de încredere $DegConf$. O direcție viitoare de cercetare o reprezintă efectuarea de experimente complexe, în care să se capteze inclusiv aspecte referitoare la introspecție și autoanaliză vis-a-vis de propria cunoaștere a subiectului uman.

6. Concluzii, contribuții și direcții viitoare de cercetare

Unul dintre proiectele gigantului internet Yahoo (<http://labs.yahoo.com/project/knowledge-acquisition-and-management/>) se referă la achiziția de cunoștințe și managementul acestora în cadrul companiei. Proiectul este descris astfel:

“Pentru a susține procesele de analiză a conținutului, căutare semantică și personalizare, am construit o bază de cunoștințe unificată a tuturor entităților și conceptelor importante pentru Yahoo. Această bază de cunoștințe gestionează milioane de entități interconectate, pentru care sunt reținute aspecte și relații cheie și conexiuni cu importante baze de date externe. Platforma cuprinde o serie de servicii special concepute pentru a rezolva problemele legate de achiziția continuă de date, extragerea de informații din text și date structurate, verificarea cunoștințelor și stocarea și indexarea pe scară largă a datelor.

Rezolvarea problemelor de achiziție și gestionare de cunoștințe implică de obicei domeniile de prelucrarea datelor și ingineria cunoașterii în fluxuri de lucru complexe din care face parte inclusiv interacțiunea cu experții umani. Provocările sunt numeroase. De exemplu, cum vom dobândi, extrage și analiza entitățile și relațiile din sursele de date text și structurate în mod continuu? Cum pot fi automat aliniate, normalizate, potrivite și combinate obiectele pentru a transforma informația rigidă, incompletă, contradictorie și potențial inexactă într-un graf de cunoștințe unificate și relevante? Cum putem stoca, gestiona și interoga la scară mare grafurile de entități și relații? Domeniile de cercetare sunt complexe și includ inclusiv prelucrarea limbajului natural, extragerea de informații, analiza și prelucrarea datelor, tehnici de învățare automată, baze de date, reprezentarea cunoștințelor, regăsirea de informații și calcul de înaltă performanță.”

Yahoo Labs Project: Knowledge Acquisition and Management

Consider că descrierea proiectului Yahoo evidențiază nu numai actualitatea și complexitatea unui proces de achiziție și management, dar și faptul că tendința actuală este de a migra procesele clasice de achiziție în sisteme automate de mari dimensiuni, care achiziționează cunoaștere în mod continuu, în timp real și din fluxuri uriașe de date. La ora actuală rețelele sociale Facebook, Twitter, LinkedIn precum și Yahoo, Microsoft și Google contribuie cu numeroase platforme open-source de analiză, prelucrare, interogare și stocare a cantităților uriașe de date cu performanțe de timp real. Procesul clasic de achiziție de cunoștințe se desfășoară la o cu totul altă scară.

Expertul nu mai este o entitate individuală, ci este reprezentat de comunitate.

Analiza și extragerea de cunoaștere se realizează din perspectiva unei comunități. Compania Twitter deține instrumente prin care poate analiza opinia comunității despre anumite evenimente, show-uri televizate, campanii politice etc. Aceste instrumente devin instrumente de predicție și din punct de vedere al marketing-ului sunt extrem de valoroase. Se pot corela anumite momente, spre exemplu dintr-un show TV, cu opinia comunității. Aceste corelații prezintă informație extrem de valoroasă pentru cei direct implicați în realizarea show-ului și care au posibilitatea de acționa astfel încât să maximizeze audiența. De exemplu, în figura următoare se poate remarca evoluția mesajelor pe platforma Twitter în timpul show-ului *XFactor*. Evoluția poate fi corelată cu performanțele de scenă ale participanților.

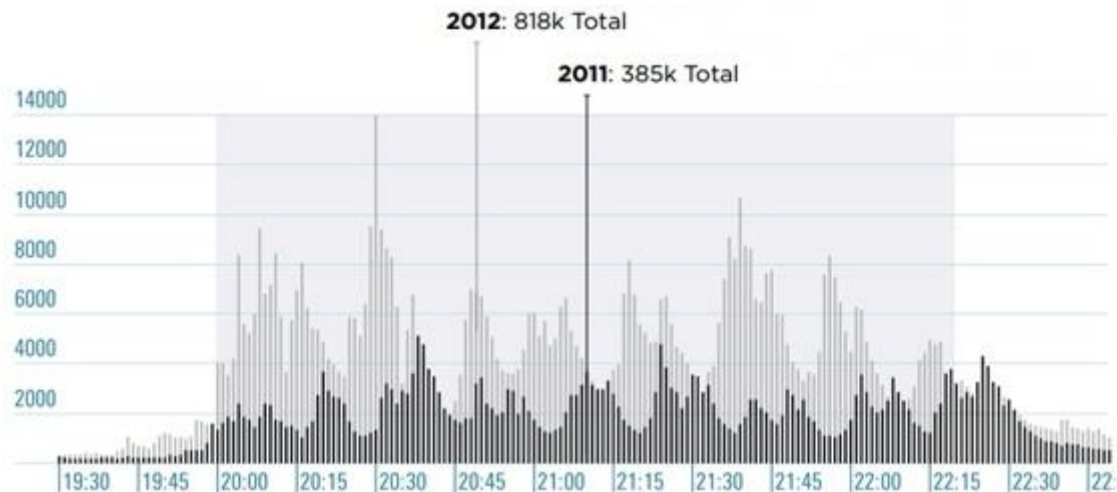


Fig. 29 Evoluția mesajelor pe platforma Twitter în timpul emisiunii XFactor

Modelul dezvoltat în cadrul tezei este actual în contextul tehnologic curent. Dezvoltarea unui sistem într-o paradigmă distribuită bazată pe agenți/actori este o soluție adoptată din ce în ce mai des de platformele comerciale. Platformele de mesagerie Facebook Chat și WhatsApp au implementate o astfel de paradigmă, folosind limbajul Erlang.

Proiectul european ITEA2 TWIRL este implementat într-o arhitectură clasică de tip Enterprise, fiind organizat în jurul unui canal de mesaje. O modelare a cunoașterii bazată pe agenți, așa cum este descrisă în cadrul tezei, ar aduce numeroase beneficii platformei din punct de vedere arhitectural.

Ca și contribuție principală, se poate afirma că este pentru prima dată când argumentarea este folosită într-un sistem multi-agent pentru achiziția de cunoștințe. Sistemul propus a fost elaborat prin prisma unei posibile implementări comerciale, astfel că descrierea și caracteristicile acestuia pot fi relativ ușor implementate cu tehnologii curente.

Consider că sistemul descris în teză deschide o nișă nouă de cercetare. Sistemul propus este vast, iar descrierea și specificația realizată reprezintă doar un punct de plecare pentru continuări și extinderi ale acestuia.

6.1. Contribuții

În continuare sunt enumerate contribuțiile principale aduse în cadrul tezei:

- identificarea și situarea problematicii achiziției de cunoștințe pentru agenți inteligenți în contextul tehnologic actual,
- introducerea argumentării ca instrument principal de achiziție de cunoștințe pentru agenții inteligenți,
- descrierea unui scenariu de argumentare – arborele de dezbatere, etapele și modul de construcție al acestuia,
- definirea unui număr de 6 etape specifice pentru procesul de achiziție de cunoaștere și corelarea lor cu momente din algoritmul de dezbatere,
- extinderea modelului Freebase pentru a putea conține proprietăți care pot fi argumentate,
- introducerea, definirea și formalizarea sistemului ARGKAMAS,
- introducerea și definirea parametrilor *PrspCtx* – Perspectiva Contextuală a dezbaterii, *PrspCnx* – Perspectiva Conexă a perspectivei principale, *PrspDeb* - Perspectiva Dezbaterii și *ISD*- Interesul agentului pentru declanșarea unei dezbateri,
- introducerea și definirea noțiunii de credință *Bel* a agentului *Ag*,

- introducerea și definirea noțiunilor de baze de influențe: *IIB* – Baza Internă de Influențe și *AgCIB* – Agentul Bază de Date de Influențe Comune,
- introducerea și definirea noțiunii de grad de influență *DegInfl*,
- introducerea și definirea celor doi parametri ce caracterizează o credință *DegInt* și *DegConf*,
- definirea algoritmului de dezbateri și a implicării fiecărui tip agent în parte,
- definirea celor două tipuri de cunoștințe din *ARGKAMAS*: cea semantică oferită de către agentul bază de date semantică *AgSB* și cea de tip credință, reprezentată intern în agentul *Ag*,
- introducerea și definirea unei reguli de influență dintre argumente,
- introducerea și definirea noțiunii de Distanța dintre perspective $\delta(Prsp_x, Prsp_y)$,
- introducerea și definirea următorilor algoritmi:
 - algoritm de selecție a topicului și declanșarea dezbaterii,
 - algoritm de participare la dezbateri,
 - algoritm pentru determinarea gradului de interes *DegInt*,
 - algoritm pentru determinarea gradului de încredere *DegConf*,
 - definirea parametrului Validare socială *ValSoc*
 - definirea parametrului Potențial de participare la dezbateri *PartPot*
 - definirea parametrului Convingător *Conv*
 - includerea în model a influenței parametrului convingător *Conv* asupra opiniei comunității, dată de validarea socială *ValSoc*
- structurarea sistemului *ARGKAMAS* pe trei niveluri,
- definirea comportamentului agenților la nivel de sistem,
- specificarea fiecărui agent principal și intermediar,
- specificarea mesajelor dintre unitățile funcționale,
- realizarea unui experiment de argumentare cu 50 de subiecți umani,
- implementarea unei aplicații pentru determinarea gradelor de interes și încredere.

6.2. Direcții viitoare de cercetare

Sistemul propus în teză reușește să modeleze doar anumite aspecte. Direcțiile de cercetare sunt numeroase, fiecare având un grad de complexitate ridicat. În continuare sunt listate câteva dintre direcțiile viitoare de cercetare identificate pe parcursul elaborării tezei. Explicații suplimentare pentru fiecare dintre acestea se regăsesc în cadrul tezei.

Printre posibilele direcții viitoare de cercetare referitoare la modelarea comportamentului psihologic al agentului sunt:

- studierea mecanismelor de introspecție – prin intermediul introspecției, agentul este capabil să declanșeze dezbateri referitoare la propriile credințe în funcție de anumite stări interioare. Spre exemplu, o stare interioară poate fi curiozitatea de a afla dacă anumite cunoștințe mai sunt valabile, sau constatarea că anumite cunoștințe nu sunt suficient argumentate,
- investigarea diverselor caracteristici psihologice ale agentului prin modificarea regulilor de inferență fuzzy, astfel încât comportamentul acestuia să fie cât mai apropiat de cel al omului. De asemenea se pot implementa și alte formalisme decât cele fuzzy,
- specificarea semantică a domeniilor de interes ale agentului, astfel încât acesta să fie capabil să-și înțeleagă și urmărească interesul în cadrul unei dezbateri,
- posibilitatea ca agentul să-și modifice singur domeniul de interes în funcție de dezbaterile la care a participat implică un nou tip de evoluție a acestuia: nu numai din punct de vedere al cunoștințelor ci și din punct de vedere al dezvoltării personale.

Printre posibilele direcții viitoare de cercetare referitoare la modelarea procesului de argumentare sunt:

- analiza evoluției arborilor de dezbateri la schimbarea topicului inițial și influența acestui aspect asupra desfășurării procesului de argumentare,
- optimizarea algoritmilor de identificare a proprietăților argumentabile în scopul reducerii numărului de dezbateri declanșate,
- optimizarea algoritmilor de difuzare a argumentelor astfel ca doar agenții interesați să fie informați în scopul participării la dezbateri. În orice sistem este de dorit a se evita o cantitate prea mare de mesaje, pentru a nu-l bloca. Se pot studia diverse mecanisme pentru ca doar agenții cu adevărat interesați de dezbateri să primească mesajele,
- problematica schimbării topicului inițial implică schimbarea tematicii în timpul dezbaterii. Este posibil ca dezbateri să ajungă în zone de foarte puțin interes pentru agent, sau, invers, agentul să constate că este interesat și de alte domenii,
- studierea diversilor algoritmi pentru finalizarea unei dezbateri. Se va studia inclusiv influența strategiilor asupra rezultatelor dezbaterii și se vor realiza studii comparative cu dezbateri între subiecți umani.

Printre posibilele direcții viitoare de cercetare referitoare la introducerea de noi cunoștințe în sistem prin metode de analiză și knowledge discovery sunt:

- studierea de diverse metode specifice de knowledge discovery care să identifice reguli de asociere între argumente, reprezintă o posibilă abordare a îmbogățirii bazelor de influențe,
- definirea de mecanisme de completare a bazelor de influențe folosind tehnici specifice de knowledge discovery care permit analiza unor seturi mari de date, precum cele specifice senzorilor sau fluxurilor de informații.

Printre posibilele direcții viitoare de cercetare referitoare la modelarea și implementarea sistemului ARGKAMAS și a agenților acestuia sunt:

- completarea modelului fiecărui agent cu diverse alte interacțiuni specifice. Modelul propus permite extinderi viitoare și adăugarea de noi roluri și funcționalități,
- implementarea unui produs software folosind o paradigmă de programare orientată agent. În cadrul tezei, sistemul a fost specificat tocmai pentru a permite o tranziție ușoară de la teorie la produs software.

Printre posibilele direcții viitoare de cercetare referitoare la configurarea experimentelor și analiza rezultatelor sunt:

- efectuarea de experimente complexe în care să fie incluse aspecte referitoare la introspecția și auto-analiza subiectului uman,
- identificarea de noi modalități de analiză și reprezentare a datelor experimentale,
- identificarea și izolarea perturbațiilor introduse în sistem de către factorul uman,
- studierea influenței nivelului educațional și cultural asupra dezbaterilor și corelarea rezultatelor cu cele ale algoritmului propus în teză.

Listă lucrări publicate și prezentate

1. Jascanu N., **Jascanu V.**, *Concept of freely mobile object*, SIMSIS 11 The 11th International Symposium on Modeling, Simulation and Systems' Identification, pp. 74-78, 2001, ISBN 973-8139-98-8
2. Nicolau F., Jascanu N., **Jascanu V.**, *Role Modeling through Aspect-Oriented Programming*. SIMSIS 12 The 12th International Symposium on Modeling, Simulation and Systems' Identification , pp. 231-237, 2004, ISBN 973-627-156-0
3. Jascanu N., **Jascanu V.**, Nicolau F., *On the role of an emotional community agent in retail markets*, SIMSIS 13 The 13th International Symposium on Modeling, Simulation and Systems' Identification, 2007.
4. Jascanu N., Bumbaru S., **Jascanu V.**, *On the emotional aspects of an electronic commerce multi-agent platform*, Proceedings of 5th European Workshop on Multi-Agent Systems, EUMAS, 2007.
5. (indexată de CiteSeerX) Jascanu N., **Jascanu V.**, Nicolau F., *A new approach to e-commerce multi-agent systems*, Proceedings of The Annals of University „Dunarea de Jos”, 2007.

Citări lucrare

- 5.1. Goncalves R. C., Zhang P., *Social Commerce: looking back and forward*, Proceedings of the American Society for Information Science and Technology, Volume 48, Issue 1, pp 1-10, ASIST 2011
 - 5.2 Huang Z., Yoon S.Y, Benyoucef M., *Adding Social Features to E-commerce*, Proceedings of the Conference on Information Systems Applied Research, EDSIG (Education Special Interest Group of the AITP), 2012, ISSN: 2167-1508.
 - 5.3. Gonçalves R.C., Zhang P., *Website features that gave rise to social commerce: a historical analysis*, Journal of Electronic Commerce Research and Applications, Volume 12, number 4, pp 260 - 279, 2013, ISSN 1567-4223, <http://www.sciencedirect.com/science/article/pii/S1567422313000306>
 - 5.4. Gatautis R., Medziausiene A., *Factors affecting social commerce acceptance in Lithuania*, The 2nd International Scientific Conference Contemporary Issues in Business, Management and Education 2013, Procedia - Social and Behavioral Sciences, Volume 110, pp 1235-1242, 2014, <http://www.sciencedirect.com/science/article/pii/S1877042813056103>
6. (indexată de Springer, Thomson WebOfScience) Jascanu N., **Jascanu V.**, Bumbaru S., *Toward Emotional E-Commerce: The Customer Agent*, Proceedings of the 12th International Conference on Knowledge-Based intelligent information and Engineering Systems, Part I (Zagreb, Croatia, September 03 - 05, 2008). I. Lovrek, R. J. Howlett, and L. C. Jain, Eds. Lecture Notes In Artificial Intelligence, vol. 5177. Springer-Verlag, Berlin, Heidelberg, 202-209. DOI= http://dx.doi.org/10.1007/978-3-540-85563-7_30

Citări lucrare

- 6.1. Efthymios A., Virvou M., *Emotional Intelligence in Multimodal Object Oriented User Interfaces*, Book Series: Studies in Computational Intelligence, Publisher: Springer Berlin/Heidelberg, Volume 226/2009, Book: New Directions in Intelligent Interactive

Multimedia Systems and Services – 2 ISBN 978-3-642-02936-3, pp. 349-359, SpringerLink Date: July 09, 2009

- 6.2. Leon S., Nikov A., *Intelligent emotion-oriented eCommerce systems*, AIKED 2010 Proceedings of the 9th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases, , pp. 202-207, ISBN 978-960-474-154-0 <http://portal.acm.org/citation.cfm?id=1808036.1808072>
- 6.3. Efthymios A., Virvou M., Kabassi K., *Recognition and Generation of Emotions in Affective e-Learning*, ICSOFT 2009 Proceedings of the 4th International Conference on Software and Data Technologies, Volume 2, pp. 273-280, 2009, ISBN 978-989-674-010-8
- 6.4 Leon S., Nikov A., Chadee D., *A Control Structure for Intelligent Emotion-Oriented eCommerce Systems*, Proceedings of the World Congress on Engineering 2010 Vol I WCE 2010, June 30 - July 2, 2010, ISBN: 978-988-17012-9-9, ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)
- 6.5. Efthymios A., Virvou M., *Object oriented architecture for affective multimodal e-learning interfaces*, International Journal of Computer Engineering & Technology Volume 4, Issue 3, pp. 171-180, August 2010, ISSN 1872-4981, <http://dx.doi.org/10.3233/IDT-2010-0078>
- 6.6. Leon S., Nikov A., *Intelligent Emotion-Oriented eCommerce Systems*, WTOS Volume 9, Issue 6, pp. 594-606, 2010, ISSN 1109-2777, <http://portal.acm.org/citation.cfm?id=1853829.1853831>
<http://www.wseas.us/e-library/conferences/2010/Cambridge/AIKED/AIKED-32.pdf>
- 6.7. Efthymios A., Virvou M., Kabassi K., *Emotion Based User Interaction in Multimedia Educational Applications*, Book: Software and Data Technologies, Volume 50, pp. 277-289, Series: Communications in Computer and Information Science, 2011, ISBN 978-3-642-20116
http://dx.doi.org/10.1007/978-3-642-20116-5_21
- 6.8. Efthymios A., Virvou M., *Multimodal object oriented user interfaces in mobile affective interaction*, Journal Multimedia Tools and Applications, Volume 59, Issue 1, pp. 41-63, ISSN 1380-7501
<http://dx.doi.org/10.1007/s11042-011-0744-y>
7. Pecheanu E., Stefanescu D., Istrate A., **Jascanu V.**, *On Modeling Adaptive Web-based Instructional Systems*, Proceedings of the 8th RoEduNet International Conference on Networking in Education and Research, pp 84-90, 2009, ISBN 978-606-8085-15-9
8. (capitol carte, indexat de Springer) Jascanu N., **Jascanu V.**, Bumbaru S., *Toward Emotional E-Commerce*, Book Agent and Multi-agent Technology for Internet and Enterprise Systems ,series Studies in Computational Intelligence, pp. 293-321, Volume 289, 2010, ISBN 978-3-642-13525-5
http://dx.doi.org/10.1007/978-3-642-13526-2_14

Citări capitol

- 8.1. Schatter G., *Affektive Agenten, Avatare, Apparate. Emotionale Empathie als Voraussetzung für überzeugende Charaktere künstlicher Subjekte* Leicht überarbeitete Fassung der 14. Buckower Mediengespräche 24. und 25. September*

2010 „*Mein Avatar und ich. Die Interaktion von Realität und Virtualität in der Mediengesellschaft*“. Kopaed München, 2011, S. 13-24, ISBN 978-3-86736-014-2.

9. (indexată de Springer, Thomson WebOfScience) **Jascanu V.**, Jascanu N., Bumbaru S., *Toward emotional E-commerce: formalizing agents for a simple negotiation protocol*, Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems, KES 2010, pp. 188-197, ISBN 3-642-15386-0, 978-3-642-15386-0,
<http://portal.acm.org/citation.cfm?id=1893893.1893920>
10. (indexată de Thomson WebOfScience) Beldiman L., **Jascanu V.**, Ifrim V., Jascanu N., *An Innovative Approach to Education-Debate Software Tools*, Proceedings of the 6th International Seminar on the Quality Management in Higher Education, QMHE, 2010, ISBN 978-973-662-566-4
11. (indexată de Thomson WebOfScience) Beldiman L., Ifrim V., Jascanu N., **Jascanu V.**, *Debate software tools used as pedagogical resources*, The 6th International Scientific Conference eLSE –eLearning and Software for Education, 2010, pp. 55-62, Unique-ID = {{ISI:000282189000006}}
12. (indexată de Springer, DL ACM) **Jascanu V.**, Jascanu N., Bumbaru S., *Formalizing Emotional E-Commerce Agents for a Simple Negotiation Protocol*, In Book *Transactions on Computational Collective Intelligence VII*, series *Lecture Notes in Computer Science*, volume 7270, pp 43-60, 2012, ISBN 978-3-642-32065-1
http://dx.doi.org/10.1007/978-3-642-32066-8_2
13. (indexată de Thomson WebOfScience) Beldiman L., **Jascanu V.**, *Towards a Second-Screen Experience in E-Learning*, Proceedings of the 9th International Scientific Conference eLearning and Software for Education, Book: Quality and efficiency in eLearning, vol 1, 2013, ISSN: 2066-026X

Bibliografie selectivă

- [1] Gaines B.R. *Human rationality challenges universal logic*. In *Logica Universalis*, vol. 4(2), SP Birkhauser Verlag Basel, 2010, pp.163-205., ISSN 1661-8297
- [2] Tilson D., Lyytinen K, Sorensen C. *Research commentary-digital infrastructures: the missing IS research agenda*.In *Journal of Information Systems Research*, vol. 21(4), 2010, pp. 748-759, ISSN 1526-5536
- [3] Korotayev A., *A compact macromodel of world system evolution*, *Journal of World-Systems Research*, vol. 11, 2005, pp.79–93, ISSN1076–156x,
- [4] Pike A.W.G., Hoffmann D.L., Garcia-Diez M., Pettitt P.B., Alcolea J., Balbin R.D., Gonzalez-Sainz C., Heras C. d. I., Lasheras J.A., Montes R., Zilhao, J., *U-series dating of Paleolithic art in 11 caves in Spain*, *Science*, vol. 336 no. 6087, 2012, pp. 1409–1413, DOI:10.1126/science.1219957
- [5] Pettitt P., *Art and the middle-to-upper paleolithic transition in Europe: comments on the archaeological arguments for an early upper paleolithic antiquity of the Grotte Chauvet art*, *Journal of Human Evolution*, Vol 55, Issue 5, 2008, pp908–917, Chronology of the Middle-Upper Paleolithic Transition in Eurasia, <http://dx.doi.org/10.1016/j.hevol.2008.04.003>
- [6] Akerkar R., Sajja P. *Knowledge-based systems*. Book. Jones & Bartlett Publishers, 2010, ISBN 978-0-7637-7647-3
- [7] Verhagen W.J., Bermell-Garcia P., van Dijk R.E., Curran R. *A critical review of Knowledge-Based Engineering: An identification of research challenges*. In *Advanced Engineering Informatics*, vol. 26(1), 2012, pp. 5-15, <http://dx.doi.org/10.1016/j.aei.2011.06.004>
- [8] Gaines B.R., *Knowledge acquisition: Past, present and future*, *International Journal of Human-Computer Studies*, vol 71, Issue 2, 2013, pp 135-156, <http://dx.doi.org/10.1016/j.ijhcs.2012.10.010>
- [9] Fresco N. *The explanatory role of computation in cognitive science*. In *Minds and Machines*, vol. 22(4), Springer Netherlands 2012, pp.353-380., ISSN 1572-8641
- [10] Ricci F., Rokach L., Shapira, B. *Introduction to recommender systems handbook* Book, Springer US, 2011, pp. 1-35. ISBN 978-0-387-85819-7
- [11] El-Kareh R., Hasan O., Schiff, G. D. *Use of health information technology to reduce diagnostic errors*. In *BMJ quality & safety*,vol. 22, 2013, pp.40-51, doi:10.1136/bmjqs-2013-001884
- [12] Gliozzo A., Biran O., Patwardhan S., McKeown K., *Semantic Technologies in IBM Watson™*, Proceedings of the 4th Workshop on Teaching NLP and CL Workshop (TNLP) at ACL 2013, Sofia, Bulgaria, 2013, p85.
- [13] Berners-Lee T., *Information Management: A Proposal*, CERN March 1989, May 1990 Geneva
- [14] Gaines B.R., Shaw M.L.G., *Human–computer interaction in online communities*, *Journal of Research and Practice in Information Technology*, vol. 33, no 1, 2001, pp.3–15
- [15] Berners-Lee T., *Linked Data*, 2006, <http://www.w3.org/DesignIssues/LinkedData.html>
- [16] Heath T., Bizer C., *Linked Data: Evolving the Web into a Global Data Space*, Synthesis Lectures on the Semantic Web: Theory and Technology, Published by Morgan & Claypool, 2011, pp. 1-136, ISBN 9781608454303 paperback, ISBN 9781608454310 ebook

- [17] Walton D., *Fundamentals of Critical Argumentation*, Cambridge University Press, 2006, ISBN 978-0-511-13969-7
- [18] Reed C., Rowe G., *Araucaria: Software for argument analysis, diagramming and representation*, International Journal on Artificial Intelligence Tools, vol. 13(04), 2004, pp. 961-979, DOI: 10.1142/S0218213004001922
- [19] Rahwan I., *Argumentation in Multi-Agent Systems*, Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS) vol. 11 (2), 2005, pp.115–125
- [20] Dung P.M., *On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games*, Artificial Intelligence, vol. 77, issue 2, 1995, pp. 321–357, DOI: 10.1016/0004-3702(94)0041-X
- [21] Amgoud L., Cayrol C., *A reasoning model based on the production of acceptable arguments*, Annals of Mathematics and Artificial Intelligence, vol. 34, issue 1-3, 2002, pp.197–215, ISSN: 1012-2443, eISSN: 1573-7470, DOI: 10.1023/A:1014490210693
- [22] Besnard P., Hunter A. *Elements of Argumentation*, MIT Press, Cambridge, 2008, ISBN 978-0-262-02643-7
- [23] Toulmin S., *The uses of Argument*, Cambridge University Press, 1958
- [24] Cayrol C., Lagasque-Schiex M. C., *Bipolarite en argumentation*, Rapport IRIT/2004-07-R, 2004
- [25] Bollacker K., Tufts P., Pierce T., Cook R., *A Platform for Scalable, Collaborative, Structured Information Integration*, Association for the Advancement of Artificial Intelligence, aaai.org, 2007
- [26] Terziyan V., Shevchenko O., Golovianko M., *An Introduction to Knowledge Computing* 2014
- [27] Sabater J., Sierra C., Parsons S., Jennings N.R., *Using multi-context agents to engineer executable agents*, Intelligent Agents VI. Agent Theories, Architectures and Languages, Volume 1757, Springer-Verlag, 2000, pp. 260–276, ISBN 978-3-540-67200-5, eISBN: 978-3-540-46467-9, ISSN: 0302-9743
- [28] Parsons S., Jennings N.R., Sabater J., Sierra C., *Agent Specification Using Multi-context Systems*, Foundations and Applications of Multi-Agent Systems: UKMAS 1996-2000, 2002, pp . 205-226

Anexe

Anexa 1. Date experimentale

Anexa 1 prezintă datele introduse de un număr de 50 de studenți din anii III și IV de la specializarea Calculatoare, a Facultății de Automatică, Calculatoare, Inginerie Electrică și Electronică, din cadrul Universității “Dunărea de Jos” Galați.

	Afirmatia 1						Afirmatia 2						Afirmatia 3						
	Valenta		Grad				Valenta		Grad				Valenta		Grad				
	da	nu	mic	mediu	mare			da	nu	mic	mediu	mare	da	nu	mic	mediu	mare		
Student1	0	1	0	1	0			0	1	0	1	0			1	0	0	1	0
Student2	1	0	0	1	0			1	0	0	0	1			1	0	0	1	0
Student3	0	1	0	1	0			1	0	0	0	1			1	0	0	1	0
Student4	0	1	0	1	0			1	0	0	0	1			1	0	0	1	0
Student5	0	1	0	1	0			1	0	0	1	0			1	0	0	1	0
Student6	0	1	0	1	0			1	0	0	1	0			0	1	1	0	0
Student7	1	0	0	1	0			1	0	0	1	0			0	1	0	0	0
Student8	0	1	1	0	0			0	1	1	0	0			0	1	1	0	0
Student9	1	0	0	1	0			1	0	0	1	0			0	1	0	1	0
Student10	1	0	1	0	0			1	0	0	0	1			0	1	0	0	1
Student11	1	0	0	0	1			1	0	0	1	0			0	1	0	0	1
Student12	1	0	0	0	1			1	0	0	1	0			0	1	0	1	0
Student13	1	0	0	1	0			1	0	0	1	0			0	1	0	1	0
Student14	0	1	0	1	0			1	0	0	0	1			0	1	0	0	1
Student15	1	0	0	1	0			1	0	0	0	1			1	0	0	1	0
Student16	0	1	1	0	0			1	0	0	0	1			0	1	1	0	0
Student17	1	0	0	0	1			1	0	0	1	0			0	1	0	1	0
Student18	1	0	1	0	0			1	0	0	1	0			0	1	0	0	1
Student19	1	0	0	1	0			1	0	0	1	0			0	1	0	0	1
Student20	1	0	0	1	0			1	0	0	1	0			0	1	0	1	0
Student21	0	1	0	1	0			1	0	0	0	1			0	1	0	1	0
Student22	1	0	0	1	0			1	0	0	0	1			0	1	1	0	0
Student23	1	0	0	1	0			1	0	0	1	0			0	1	0	0	1
Student24	0	1	0	1	0			1	0	0	0	1			0	1	0	0	1
Student25	1	0	1	0	0			1	0	0	0	1			0	1	0	0	1
Student26	1	0	1	0	0			1	0	0	0	1			0	1	0	1	0
Student27	0	1	1	0	0			1	0	0	0	1			0	1	0	1	0
Student28	0	1	1	0	0			1	0	0	0	1			0	1	1	0	0
Student29	0	1	0	1	0			1	0	0	0	1			0	1	0	1	0
Student30	0	1	0	1	0			1	0	0	0	1			0	1	0	1	0
Student31	0	1	0	1	0			1	0	0	1	0			0	1	0	0	1
Student32	1	0	0	1	0			1	0	0	0	1			0	1	0	1	0
Student33	1	0	0	1	0			1	0	0	0	1			0	1	0	0	1
Student34	1	0	0	1	0			1	0	1	0	0			0	1	1	0	0
Student35	0	1	0	0	1			1	0	0	1	0			0	1	0	1	0
Student36	0	1	0	1	0			1	0	0	0	1			0	1	0	0	1
Student37	1	0	0	0	1			0	1	0	1	0			0	1	0	1	0
Student38	1	0	0	0	1			1	0	0	1	0			0	1	0	0	1
Student39	1	0	1	0	0			0	1	0	1	0			0	1	1	0	0
Student40	1	0	0	1	0			1	0	0	1	0			0	1	0	1	0
Student41	1	0	0	0	1			1	0	0	0	1			0	1	0	1	0
Student42	1	0	0	1	0			1	0	0	0	1			0	1	0	1	0
Student43	1	0	0	0	1			1	0	0	0	1			0	1	0	1	0
Student44	0	1	0	1	0			1	0	0	0	1			0	1	0	0	1
Student45	1	0	0	0	1			1	0	0	0	1			0	1	0	0	1
Student46	1	0	1	0	0			1	0	0	1	0			0	1	0	0	1
Student47	0	1	0	1	0			1	0	0	1	0			0	1	0	1	0
Student48	1	0	0	1	0			1	0	0	1	0			0	1	0	0	1
Student49	1	0	0	1	0			1	0	0	0	1			1	0	1	0	0
Student50	1	0	0	1	0			1	0	0	0	1			1	0	1	0	0

Anexa 2. Confirmare colaborare proiect european

Anexa 2 reprezintă confirmarea colaborării în cadrul proiectului European ITEA2 TWIRL.



PARCUL PENTRU TEHNOLOGIA INFORMAȚIEI – ET. II, CAM.
209; TEL/FAX: +40-236-407030; GALAȚI – 800025 – ROMÂNIA;
office@altfactor.ro www.altfactor.ro

Confirmare,

Prin prezenta se confirmă că d-na Jâșcanu Veronica a fost implicată în cadrul proiectului European ITEA2 TWIRL – Nr. 10029 (<https://itea3.org/project/twirl.html>), în calitate de consultant pentru dezvoltarea componentei de argumentare a platformei.

Activitatea desfășurată a constat în participarea la realizarea documentației științifice și elaborarea algoritmilor specifici pentru interpretarea arborilor de argumentare.

Director General,

Comănescu Adrian



Data: 01.04.2014