

Universitatea Tehnică din Cluj - Napoca
Facultatea De Automatică și Calculatoare

ing. Viorica Rozina CHIFU

Teză De Doctorat

Compunerea Automată a Serviciilor Web

Rezumat

Coordonator Științific: Prof. dr. ing. Ioan SALOMIE

Cluj-Napoca, 2010

Listă de lucrări publicate

Lucrări Indexate DBLP

1. Cristina Bianca Pop, **Viorica Rozina Chifu**, Ioan Salomie, Mihaela Dînşoreanu, Vlad Acreţoaie, Tudor David, *Ant-based Semantic Web Service Clustering*, 4th International Symposium on Intelligent Distributed Computing - IDC'2010, to appear in **Studies in Computational Intelligence Springer-Verlag, (ISI Proceeding)**, (Tangier) Morocco, (2010).
2. Cristina Bianca Pop, **Viorica Rozina Chifu**, Ioan Salomie, Mihaela Dînşoreanu, *Immune-inspired Method for Selecting the Optimal Solution in Web Service Composition*, in the 35th International Conference on Very Large Databases, **Z. Lacroix: RED 2009, Springer-Verlag Berlin Heidelberg 2010, LNCS 6162**, pp. 1–17, (2010).
3. **Viorica Rozina Chifu**, Ioan Salomie, Alpar Köver, and Roland Vachter, *Matching Semantic Web Services Using Learning Accuracy*, in the 11th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC2009), pp. 317-324, (Timisoara) Romania, (2009).
4. Cristina Bianca Pop, **Viorica Rozina Chifu**, Ioan Salomie, Mihaela Dînşoreanu, Iulia Vartic and Monica Vlad, *Immune-inspired Web Service Composition Framework*, in the 11th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC2009), pp. 376-383, (Timisoara) Romania, (2009).
5. **Viorica Rozina Chifu**, Ioan Salomie and Emil Ştefan Chifu, *Taxonomy Learning for the Romanian Language using SOTA and Wordnet*, in the 4th International Conference on Web Information Systems and Technologies (WEBIST), vol. 2, pp. 169-174, (**ISI Proceeding**), (Funchal) Portugal, (2008).
6. **Viorica Rozina Chifu**, Ioan Salomie, Emil Ştefan Chifu and Constantin Pârţac, *Harmony - A Framework for Automatic Web Service Composition*, in the Proceedings of the 4th International Conference on Web Information Systems and Technologies (WEBIST), vol. 2, pp. 247-252, (**ISI Proceeding**), (Funchal) Portugal, (2008).
7. Emil Ştefan Chifu and **Viorica Rozina Chifu**, *Evaluating the Hyponym Attachments in an Unsupervised Taxonomy Enrichment Framework*, in the Proceedings of the 4th International Conference on Web Information Systems and Technologies (WEBIST), vol. 1, pp. 240-243, (**ISI Proceeding**), (Funchal) Portugal, (2008).
8. **Viorica Rozina Chifu**, Ioan Salomie, and Simona Manole, *Fluent Calculus Based Web Service composition*, in the Proceedings of the Second International Conference on Web Reasoning and Rule Systems (RR2008), **LNCS Springer-Verlag Berlin Heidelberg**, pp. 230-231, (**ISI Proceeding**), (Karlsruhe) Germany, (2008).
9. Ioan Salomie, **Viorica Rozina Chifu**, Ioana Harşa and Marius Gherga, *Towards Automated Web Service Composition with Fluent Calculus and Domain Ontologies*, in the **ACM** Proceedings of

the 10th International Conference on Information Integration and Web-based Applications & Services (iiWAS2008), pp. 201-207, (Linz) Austria, (2008).

10. **Viorica Rozina Chifu**, Ioan Salomie and Emil Ștefan Chifu, *Ontology Model for Traceability in Food Industry*, in the Proceedings of the International Conference on e-Business (ICE-B2007), pp. 63-68, (**ISI Proceeding**), (Barcelona) Spain, (2007).

Lucrări Indexate IEEE Xplore

11. Cristina Bianca Pop, **Viorica Rozina Chifu**, Ioan Salomie, Mihaela Dînșoreanu, Tudor David, and Vlad Acrețoaie, *Ant-inspired Technique for Automatic Web Service Composition and Selection*, 1st Workshop on Software Services in the 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC2010), accepted for publication, (Timisoara) Romania (2010).
12. **Viorica Rozina Chifu**, Cristina Bianca Pop, Ioan Salomie, Mihaela Dînșoreanu, Alpar Köver, and Roland Vachter, *Web Service Composition Technique Based on a Service Graph and Particle Swarm Optimization*, in the Proceedings of the 6th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP2010), (**ISI Proceeding**), accepted for publication, (Cluj-Napoca) Romania, (2010).
13. **Viorica Rozina Chifu**, Cristina Bianca Pop, Ioan Salomie, and Mihaela Dînșoreanu, Tudor David, and Vlad Acrețoaie, *An Ant-inspired Approach for Semantic Web Service Clustering*, in the Proceedings of the 9th Romanian Educational Network International Conference (RoEduNet 2010), pp. 145-150, (**ISI Proceeding**), (Sibiu) Romania, (2010).
14. **Viorica Rozina Chifu**, Ioan Salomie, Riger Agota and, Valentin Radoi, *A Graph Based Backward Chaining Method for Web Service Composition*, in the Proceedings of the 5th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP2009), pp. 237-244, (**ISI Proceeding**), (Cluj-Napoca) Romania, (2009).
15. Cristina Bianca Pop, **Viorica Rozina Chifu**, Ioan Salomie, and Mihaela Dînșoreanu, *Optimal Web Service Composition Method Based on an Enhanced Planning Graph and Using an Immune-inspired Algorithm*, in the Proceedings of the 5th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP2009), pp. 291-298, (**ISI Proceeding**), (Cluj-Napoca) Romania, (2009).
16. **Viorica Rozina Chifu**, Ioan Salomie and Emil Ștefan Chifu, *Automatic Web Service Composition Using OWL-S and Fluent Calculus*, in the Proceedings of the IEEE International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC08), pp. 271-276, (**ISI Proceeding**), (Vienna) Austria, (2008).
17. Emil Ștefan Chifu and **Viorica Rozina Chifu**, *A Neural Model for Unsupervised Named Entity Classification*, in the Proceedings of the IEEE International Conference on Intelligent Agents, Web Technologies and Internet Commerce - IAWTIC08, pp.1073-1078, (**ISI Proceeding**), (Vienna) Austria, (2008).

18. **Viorica Rozina Chifu**, and Ioan Salomie, *Fluent Calculus-Based Web Service composition – From OWL-S to Fluent Calculus*, in the Proceedings of the 4th IEEE International Conference on Intelligent Computer Communication and Processing, pp. 161-168, (**ISI Proceeding**), (Cluj-Napoca) Romania, (2008).
19. Ioan Salomie, **Viorica Rozina Chifu**, Ioana Giurgiu, and Mihai Cuibus, *SAWS: A Tool for Semantic Annotation of Web Services*, in the Proceedings of the 16th IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2008), pp. 387-391, (**ISI Proceeding**), (Cluj-Napoca) Romania, (2008).
20. **Viorica Rozina Chifu**, Ioan Salomie, and Emil Ștefan Chifu, *Ontology Enhanced Description of Traceability Services*, in I.A. Letia (ed.), in the Proceedings of the IEEE 3rd International Conference on Intelligent Computer Communication and Processing (ICCP 2007), pp. 1-8, (**ISI Proceeding**), (Cluj-Napoca) Romania, (2007).
21. Emil Ștefan Chifu and **Viorica Rozina Chifu**, *Evaluation of an Unsupervised Ontology Enrichment Framework*, in I.A. Letia (ed.), in the Proceedings of the IEEE 3rd International Conference on Intelligent Computer Communication and Processing (ICCP 2007), pp. 225-228, (**ISI Proceeding**), (Cluj-Napoca) Romania, (2007).

Lucrări Indexate ACM

22. **Viorica Rozina Chifu**, Ioan Salomie, and Emil St. Chifu, *Taxonomy Learning for Semantic Annotation of Web Services*, in the Proceedings of the 11th WSEAS International Conference on Computers, (**ISI Proceeding**), pp. 299-305, (Aghios Nikolaos) Greece, (2007).

Lucrări Publicate în Reviste

23. **Viorica Rozina Chifu**, Ioan Salomie, Riger Agota and, Valentin Radoi, *Web Service Composition Approach Based On a Graph of Service Cells*, in the Proceedings of the 2th Knowledge Engineering: Principles and Techniques Conference (KEPT2009), , in Studia Informatica - Special Issue on Knowledge Engineering: Principles and Techniques, pp. 308-311, (Cluj-Napoca) Romania, (2009).
24. **Viorica Rozina Chifu**, Ioan Salomie, *A Fluent Calculus Approach to Automatic Web Service Composition*, in the **Advances in Electrical and Computer Engineering Journal**, Issue 3/2009, **ISI Journal**, pp. 75 – 83, (Suceava) Romania, (2009).
25. Ioan Salomie, **Viorica Rozina Chifu**, Sergiu Nedevschi, and Emil Ștefan Chifu, *Ontology Learning for Traceability in the Food Industry*, in the Automation, Computers, Applied Mathematics (ACAM) journal, pp. 94-101, indexed in Mathematical Reviews, (Cluj-Napoca) Romania, (2008).
26. Ioan Salomie, Cornel Rat, Sorin Suci, Cristina Pop, **Viorica Rozina Chifu**, Ioan Dragan, Livia Ardelean, Ion Carja, *A Case-study in the Archival Domain, Distributed, Agent-based, Semantic Reasoning Engine for Ontologies*, in the Automation, Computers, Applied Mathematics (ACAM) journal, (Cluj-Napoca) Romania, (2008).

Articole la Conferințe

27. **Viorica Rozina Chifu** and Ioan Salomie, *An Architecture for QoS Driven Automatic Web Service Composition*, in the Proceedings of the 2nd IEEE International Conference on Intelligent Computer Communication and Processing, pp. 121-127, (Cluj-Napoca) Romania, (2006).
28. Ioan Salomie, **Viorica Rozina Chifu**, Sergiu Nedevschi and Emil Ștefan Chifu, *Ontology Enrichment for Food Traceability Domain Using Romanian Lexico-Syntactic Patterns*, Workshop on Traceability and Systems for Traceability, pp. 28-33, Cluj-Napoca (Romania), (2008).
29. **Viorica Rozina Chifu**, Ioan Salomie, Sergiu Nedevschi and Emil Șt. Chifu, *Building a Food Industry Taxonomy by using SOTA and Extended WordNet*, Workshop on Traceability and Systems for Traceability, pp. 22-27, (Cluj-Napoca) Romania, (2008).
30. **Viorica Rozina Chifu**, Cristina Bianca Pop, Ioan Salomie, Mihaela Dinsoreanu, Ioana Avram, Anda Cipariu, *SWSDF – A Semantic Web Service Discovery Framework*, in the Proceedings of the 8th Romanian Educational Network International Conference (RoEduNet 2009), pp. 113-118, (**ISI Proceeding**), (Galați) Romania, (2009).
31. Cristina Bianca Pop, **Viorica Rozina Chifu**, Ioan Salomie, Mihaela Dînsoreanu, *Immune-inspired Method for Selecting the Optimal Solution in Web Service Composition*, in the 35th International Conference on Very Large Databases (VLDB) – the Second International Workshop on Resource Discovery (RED2009), selected to appear in **LNCS**, (Lyon) France, (2009).
32. Cristina Bianca Pop, **Viorica Rozina Chifu**, Ioan Salomie, Mihaela Dînsoreanu, Fodor Mihaly, Irina Condor, *A Bee-inspired Approach for Selecting the Optimal Service Composition Solution*, in the Proceedings of the International Conference on Development and Application Systems (DAS 2010), pp.: 420-426, (Suceava) Romania, (2010).
33. Emil Ștefan Chifu, Ioan Alfred Letia, **Viorica Rozina Chifu**, *A system to retrieve Web documents from a large collection*, in the Proceedings of the International Conference on Automation, Quality and Testing, Robotics (AQTR-THETA 13), Tome II, pp. 419-424, (Cluj-Napoca) Romania, (2002).

Cuprins

Introducere.....	1
1.1 Servicii Web Semantice.....	1
1.2 Compunerea serviciilor Web.....	2
1.2.1 Complexitatea problemei compunerii automate.....	4
1.3 Obiectivele tezei	5
1.4. Organizarea tezei.....	6
Stadiul actual al cercetării.....	10
2.1 Servicii Web semantice.....	10
2.1.1 SAWSDL.....	10
2.1.2 WSDL-S.....	11
2.1.3 OWL-S.....	12
2.1.4 WSMO.....	12
2.2 Cercetări existente în domeniul compunerii serviciilor Web.....	13
2.2.1 Tehnici de compunere bazate pe workflow.....	14
2.2.2 Tehnici de compunere folosind planificarea din inteligența artificială.....	15
2.2.3 Tehnici de compunere bazate pe metode formale.....	21
2.2.4 Tehnici de compunere ghidate ontologic.....	23
2.2.5 Tehnici de compunere bazate pe grafuri.....	24
2.2.6 Tehnici de compunere folosind calcul inspirat din natură.....	25
Tehnică de compunere a serviciilor Web folosind calculul fluenților.....	28
3.1 Calculul fluenților și FLUX.....	28
3.1.2 Calculul fluenților versus calculul situațiilor.....	32
3.2 Translatarea din ontologii OWL-S.....	33
3.2.1 Ontologia OWL-S.....	34
3.2.2 Translatarea proceselor atomice OWL-S.....	35
3.2.3 Translatarea proceselor compuse OWL-S.....	43
3.3 Translatarea din ontologii de domeniu.....	48
3.3.1 Ontologie de domeniu pentru servicii Web.....	49
3.3.2 Translatarea unei ontologii de domeniu.....	53
3.4 Compunerea serviciilor Web folosind calculul fluenților.....	57
3.4.1 Algoritmul de compunere al serviciilor Web.....	59

3.5. Prototip experimental.....	61
3.5.1 Arhitectura prototipului experimental.....	61
3.5.2 Rezultate experimentale.....	63
3.6 Comparație cu abordări similare.....	67
3.7 Concluzii.....	68
Metodă de descoperire automată a serviciilor Web.....	69
4.1 Similaritate semantică.....	70
4.1.1 Grad de potrivire semantică (<i>DoM</i>)	70
4.2 Determinarea gradului de potrivire semantică dintre două servicii.....	71
4.2.1 Codificarea descrierilor semantice ale serviciilor.....	72
4.2.2 Metrice de evaluare a gradului de potrivire semantică dintre două servicii Web.....	74
4.3 ACDOM - Algoritmul de calcul al gradului de potrivire semantică.....	79
4.3.1 Metrice de evaluare a performanțelor algoritmului ACDOM.....	84
4.4 Exemplu de calcul al DoM dintre două servicii.....	85
4.5 Prototip experimental.....	89
4.5.1 Arhitectura prototipului experimental.....	89
4.5.2 Rezultate experimentale.....	91
4.6 Concluzii.....	95
Tehnică de compunere a serviciilor Web bazată pe grafuri de celule de servicii.....	96
5.1 Concepte și definiții.....	96
5.2 Algoritmul de construire al grafului.....	105
5.3 Algoritmul de compunere.....	106
5.3.1 Etapa de inițializare.....	107
5.3.2 Etapa de căutare.....	108
5.3.3 Etapa de execuție parțială.....	110
5.3.4 Etapa de selecție și execuție.....	111
5.4 SGDL – limbaj de descriere a grafului de celule de servicii.....	112
5.5 Prototip experimental.....	114
Tehnică de compunere a serviciilor Web inspirată din sistemul imunitar.....	122
6.1 Graful de planificare îmbogățit pentru compunerea serviciilor Web	122
6.1.1 Definiții.....	123
6.1.2 Modelul de graf de planificare îmbogățit pentru compunerea serviciilor Web.....	126
6.1.3 Construirea grafului de planificare îmbogățit pentru compunerea serviciilor Web.....	127
6.1.4 Exemplu de graf de planificare îmbogățit.....	128

6.2 Metrice de evaluare.....	129
6.3 Matricea de legături semantice.....	133
6.3.1 Exemplu de matrice de legături semantice.....	133
6.4 Metoda de compunere.....	134
6.4.1 Algoritmul de compunere.....	134
6.4.2 Algoritmul de descoperire.....	136
6.4.3 Algoritmul de construire al matricei de legături semantice.....	137
6.5 Determinarea soluției optime de compunere.....	138
6.5.1 Definiții.....	138
6.5.2 Sistemul imunitar.....	141
6.5.3 Maparea principiului de selecție a clonelor la problema compunerii serviciilor Web.....	144
6.6 Metoda de selecție.....	146
6.6.1 Algoritm de selecție a soluției optime de compunere.....	146
6.6.2 Algoritm de maturizare a afinității unei soluții de compunere.....	149
6.6.3 Îmbunătățirea algoritmului de selecție a soluției optime de compunere.....	151
6.7 Prototip experimental și studiu de caz.....	152
6.7.1 Arhitectura prototipului experimental.....	153
6.7.2 Rezultate experimentale.....	155
6.8 Comparatie cu abordări similare.....	180
6.9 Concluzii.....	181
Concluzii finale.....	183
7.1 Contribuții.....	184
7.2 Dezvoltări în desfășurare	187
Bibliografie.....	189

1. Introducere

Compunerea serviciilor Web are rolul de a satisface o cerere complexă care nu poate fi îndeplinită de un singur serviciu. Necesitatea compunerii automate a serviciilor Web este justificată de omniprezența Internet, care forțează companiile să își abandoneze modelele de business moștenite și sistemele clasice învechite și să se reorganizeze ca și companii virtuale [27]. Concluzionând cele descrise mai sus, putem spune că domeniul de cercetare privind compunerea automată a serviciilor Web este un domeniu de vârf, de mare actualitate în tehnologia informațiilor, atât pe plan mondial cât și pe plan național.

Teza de față a avut ca obiectiv *cercetarea și elaborarea de tehnici de compunere automată a serviciilor Web folosind ontologii*. În cadrul acestui obiectiv mi-am propus să studiez, analizez, elaborez și validez experimental modele, metode și algoritmi referitori la descoperirea automată a serviciilor Web, compunerea automată a serviciilor Web și selectarea soluției optime de compunere. Ideile prezentate în această teză au fost publicate în 18 de lucrări științifice, opt dintre acestea fiind indexate DBLP.

Am fost membră în următoarele contracte de cercetare: 1) MAESTRO – Compunerea automată a serviciilor Web folosind ontologii (2007-2010), CNCSIS PN II IDEI 333/2007; 2) ARHINET – Sistem integrat de dezvoltare de conținut arhivistic adnotat semantic (2007 – 2010), proiect CNMP; 3) FOOD – TRACE – Sistem informatic integrat pentru asigurarea trasabilității și controlul calității în industria alimentară, (2006 – 2008), proiect CEEEX, 4) SCANURGENT – Screening, profilaxie și corecție a malformațiilor congenitale genito-urinare la copil în epoca tehnicilor terapeutice minim invazive (laparoscopie, endoscopie), (2006 – 2008), proiect CEEEX.

Am câștigat prin competiție națională un contract de cercetare pentru tineri doctoranzi intitulat *Compunerea Automată a Serviciilor Web semantice*, TD-416, CNCSIS-520, desfășurat în perioada 2007 – 2008.

2. Stadiul actual al cercetării

Acest capitol prezintă stadiul actual al cercetării în domeniul compunerii automate a serviciilor Web. Capitolul începe cu o prezentare succintă a celor patru abordări privind infrastructura serviciilor Web semantice propuse de W3C, urmată de o prezentare a cercetărilor existente în domeniul compunerii automate de servicii Web. Pe baza analizei domeniului s-au identificat șase tipuri de tehnici de compunere automată de servicii. Pentru fiecare dintre tehnicile de compunere identificate, se prezintă cercetări științifice relevante în domeniu, avantajele și dezavantajele pe care aceste abordări le oferă.

2.1. Tehnici de compunere bazate pe workflow

Tehnicile de compunere bazate pe workflow consideră un serviciu compus ca o succesiune de activități. Aceste tehnici definesc un model de proces abstract care include un set de activități și dependențele lor de date. Fiecare activitate conține o clauză de interogare care este folosită pentru a căuta servicii atomice reale care îndeplinesc acea activitate [15]. Este

necesar ca solicitantul de serviciu să specifice câteva constrângeri, inclusiv dependențele dintre servicii atomice și preferințele non-funcționale.

Eflow [4] este o platformă pentru specificarea și managementul serviciilor compuse. În *Eflow* un serviciu compus este modelat prin intermediul unui graf care definește ordinea de execuție a serviciilor într-un proces. Graful este creat manual, dar poate fi actualizat dinamic. Pot fi distinse trei tipuri de noduri în graf: nod de tip serviciu, nod de decizie și nod de tip eveniment. Nodurile de tip serviciu reprezintă invocarea unui serviciu atomic sau compus, nodurile de decizie specifică alternative și reguli de control ale fluxului de execuție, iar nodurile de tip eveniment permit serviciilor să trimită/recepționeze câteva tipuri de evenimente. Arcele grafului denotă dependențele de execuție dintre noduri. Deși graful este specificat manual, procesul de legare la servicii concrete este automatizat.

2.2. Tehnici de compunere care folosesc planificarea din domeniul inteligenței artificiale

O problemă de planificare în AI [8] poate fi descrisă ca o tuplă (S, S_0, G, A, Γ) , unde: S reprezintă setul de stări posibile ale lumii, $S_0, G \in S$ reprezintă starea inițială și finală a lumii, A este setul de acțiuni pe care planificatorul le execută în încercarea de a schimba o stare a lumii în alta, $\Gamma \subseteq S \times A \times S$ este setul tranzițiilor de stare. O traziție între două stări este posibilă doar atunci când precondițiile și efectele acțiunilor care trebuie executate pentru a ajunge din starea S_i în starea S_{i+1} sunt satisfăcute. În termeni de servicii Web, A este setul de servicii disponibile, Γ este de setul tranzițiilor de stare care denotă intrările și ieșirile serviciilor, $S_0, G \in S$ sunt intrările și ieșirile serviciului compus solicitat. Pornind de la aceste premise, în literatură au fost propuse diverse tehnici de planificare din domeniul AI pentru rezolvarea problemei compunerii serviciilor Web. Aceste tehnici de compunere pot fi clasificate în:

- Tehnici de compunere care folosesc calculul situațiilor;
- Tehnici de compunere care folosesc planificare bazată pe reguli;
- Tehnici de compunere care folosesc alte tehnici de planificare din AI;
- Tehnici de compunere care folosesc demonstratoare de teoreme.

SWORD [13] este un prototip experimental care folosește tehnici de planificare bazate pe reguli pentru construirea unui serviciu compus. Spre deosebire de alte abordări existente [16], [17], SWORD nu a fost dezvoltat pornind de la standardele de descriere a serviciilor Web (ex: WSDL, DAML-S). În SWORD, serviciile sunt specificate prin intermediul precondițiilor și efectelor lor folosind modelul *entitate – relație*. Un serviciu Web este reprezentat printr-o clauză Horn care denotă postcondiția care este satisfăcută atunci când precondiția este adevărată. Pentru a crea un serviciu compus, e necesar ca solicitantul de serviciu să specifice intrările și ieșirile serviciului compus solicitat. Pe baza acestora, folosind un sistem expert bazat pe reguli, se va genera un plan de compunere. Important de menționat este faptul că o astfel de abordare poate genera uneori rezultate nesigure atunci când postcondiția nu este determinată în mod unic de către o precondiție. Autorii

argumentează că astfel de rezultate nesigure pot fi evitate doar atunci când precondițiile sunt dependente funcțional de postcondițiile unui serviciu.

2.3. Tehnici de compunere bazate pe metode formale

O metodă formală este un set de unelte și notații, cu o semantică formală, folosite pentru specificarea neambiguă a cerințelor unui sistem, care admite demonstrarea de proprietăți ale acelei specificații și demonstrarea corectitudinii unei implementări în raport cu acea specificație [1]. În contextul compunerii serviciilor Web, proprietatea demonstrată este că o implementare corectă din punct de vedere funcțional satisface o specificație particulară.

În [2] și [3], serviciile Web sunt reprezentate cu ajutorul sistemelor de tranziție. Se consideră că serviciile schimbă mesaje ținând seama de o topologie de comunicare predefinită, exprimată ca un set de canale. Sistemului de sinteză al compunerii îi sunt furnizate două intrări: (i) comportamentul serviciului dorit (setul tuturor conversațiilor posibile) descris de un sistem de tranziție etichetat și (ii) template-ul de compunere (ex: modalitatea de înlănțuire, setul de servicii componente și mesajele interschimbate între servicii). Ieșirea sistemului de sinteză al compunerii este un mediator (reprezentat ca un sistem de tranziție etichetat) care orchestrează execuția serviciilor componente în așa fel încât mesajele interschimbate între servicii să se conformeze cu specificația serviciului dorit. Autorii codifică problema compunerii ca o formulă în logică dinamică propozițională φ și reduc generarea mediatorului la abilitatea de a satisface formula φ . Un dezavantaj al abordării prezentate în [2] și [3] este că nu se iau în considerare constrângeri non-funcționale în procesul de compunere.

2.4. Tehnici de compunere ghidate ontologic

În tehnicile de compunere ghidate ontologic sunt folosite descrieri ontologice ale serviciilor, în termeni de concepte și relații, pentru compunerea automată/semi-automată a serviciilor Web.

În [19] autorii propun o abordare de compunere dinamică în care serviciile Web sunt descrise semantic în DAML-S. Informația semantică din DAML-S este convertită în triplete RDF și încărcată într-o bază de cunoștințe. Prototipul experimental dezvoltat are două componente de bază: un motor de inferență și un modul de compunere. Motorul de inferență, reprezentat de un raționator OWL construit în Prolog, stochează informațiile despre serviciile existente în baza de cunoștințe și are capacitatea de a descoperi serviciile care se potrivesc cu o cerere. Modulul de compunere asigură comunicarea între utilizatorul uman și motorul de inferență. Modulul de compunere îi permite utilizatorului să creeze un *workflow* de servicii prin prezentarea alternativelor disponibile la fiecare pas. Dezavantajul abordării propuse este că sistemul nu e scalabil.

2.5. Tehnici de compunere bazate pe grafuri

În contextul compunerii serviciilor Web, grafurile au fost folosite pentru a modela dependențele fluxurilor de *control* și de *date* dintre diverse funcționalități furnizate de unul sau mai multe servicii.

Hashemian și Mavaddat [10] propun o abordare bazată pe grafuri pentru compunerea serviciilor Web. Autorii modelează serviciile Web și cererea de serviciu folosind automate de I/O și reprezintă dependențele de date dintre serviciile componente folosind un graf de dependențe. Nodurile grafului corespund intrărilor și ieșirilor serviciilor, iar muchiile reprezintă însăși serviciile Web asociate. Compunerea se realizează în două etape: (i) în prima etapă sunt descoperite serviciile care participă la compunere, (ii) în a doua etapă se crează graful de dependențe pe baza serviciilor descoperite. Dezavantajul abordării propuse este că nu ține cont de faptul că în timpul execuției serviciului compus unul sau mai multe dintre serviciile care fac parte din soluție pot fi indisponibile.

2.6. Tehnici de compunere folosind calcul inspirat din natură

Tehnicile de calcul inspirate din natură se aplică în procesul de selecție a soluției optime de compunere și pot fi clasificate în: (i) tehnici de compunere inspirate din comportamentul furnicilor; (ii) tehnici de compunere inspirate din comportamentul colectiv al păsărilor; (iii) tehnici de compunere inspirate din principiile sistemului imunitar; (iv) tehnici de compunere inspirate din comportamentul albinelor.

În [25] autorii propun un algoritm inspirat din procesele sistemului imunitar pentru selecția soluției optime de compunere dintr-un set de soluții disponibile. Autorii propun o tehnică de compunere bazată pe workflow, în care workflow-ul reprezintă un plan abstract căruia îi sunt asociate servicii concrete corespunzătoare fiecărui task din workflow. Fiecare serviciu are asociată o valoare globală de QoS. Algoritmul de selecție propus constă din doi pași: (i) operația de selecție imunitară și (ii) operația de selecție a clonelor. Operația de selecție imunitară se referă la: (i) măsurarea afinității – se măsoară similaritatea dintre anticorpi și (ii) măsurarea concentrației – se măsoară diversitatea populației de anticorpi. O particularitate a acestei abordări constă în utilizarea unui vaccin care furnizează informație euristică (se selectează întotdeauna cele mai bune servicii din setul de servicii candidate) care accelerează convergența către soluția optimă.

3. Tehnică de compunere a serviciilor Web folosind calculul fluenților

Acest capitol prezintă o tehnică originală de compunere automată a serviciilor Web folosind calculul fluenților [21, 22, 23, 24, 25]. Tehnica propusă constă dintr-o metodă de traducere a descrierilor semantice de servicii în calculul fluenților și o metodă de compunere a serviciilor Web folosind calculul fluenților. Serviciile Web descrise semantic în OWL-S sau SAWSDL [12] sunt traduse în calculul fluenților în axiome pentru condițiile acțiunii și

axiome de actualizare a stării. Baza de cunoștințe rezultată în urma procesului de traducere este folosită de metoda de compunere în vederea identificării unei soluții de compunere care satisface o cerere de serviciu compus. Metoda de compunere propusă definește problema compunerii serviciilor Web ca o problemă de planificare în calculul fluenților.

Pentru validarea abordării propuse s-a dezvoltat un prototip experimental care integrează metoda de traducere și algoritmul de compunere. Prototipul experimental a fost testat și evaluat pe un set de scenarii din domeniul planificării unei călătorii și din domeniul planificării de evenimente sociale.

3.1. Traducerea proceselor atomice OWL-S în calculul fluenților

O operație a unui serviciu Web este modelată în OWL-S ca un proces atomic care are intrări, ieșiri, precondiții și efecte. Intrările reprezintă informația pe care utilizatorul trebuie să o furnizeze pentru a se putea invoca serviciul. Ieșirile sunt rezultatele pe care utilizatorul se așteaptă să le obțină după execuția serviciului. Precondițiile sunt constrângeri ce trebuie îndeplinite pentru a se putea invoca serviciul. Efectele sunt expresii logice care devin adevărate dacă serviciul a fost invocat cu succes.

Un proces atomic OWL-S, PA_{OWL-S} este tradus în calculul fluenților într-o acțiune, $a_{PA_{OWL-S}}$.

Traducerea precondițiilor și intrărilor unui proces atomic. În calculul fluenților o precondiție a unei acțiuni a se exprima astfel:

$$Poss(a, z) \equiv \pi(x, z)$$

În formula anterioară, $\pi(x, z)$ reprezintă un fluent. În abordarea pe care o propunem, atât precondițiile cât și intrările unui proces atomic PA_{OWL-S} sunt traduse în calculul fluenților în precondițiile acțiunii $a_{PA_{OWL-S}}$. În calculul fluenților, facem distincție între precondițiile provenite din intrările procesului atomic PA_{OWL-S} și precondițiile provenite din precondițiile procesului PA_{OWL-S} .

O precondiție a unui proces atomic PA_{OWL-S} este tradusă în calculul fluenților astfel:

$$Poss(a_{PA_{OWL-S}}, z) \equiv Holds(\phi_i, z)$$

unde :

- ϕ_i – este un fluent reprezentând precondiția procesului atomic PA_{OWL-S} ;
- $a_{PA_{OWL-S}}$ – reprezintă acțiunea obținută din traducerea procesului atomic PA_{OWL-S} .

O intrare a unui proces atomic PA_{OWL-S} este tradusă în calculul fluenților astfel:

$$Poss(a_{PA_{OWL-S}}, s) \equiv Knows(\phi_i, z)$$

unde:

- φ_i – este un fluent reprezentând intrarea procesului atomic PA_{OWL-S} ;
- $a_{PA_{OWL-S}}$ – reprezintă acțiunea obținută din translatarea procesului atomic PA_{OWL-S} .

Pe baza formulelor anterior menționate, axioma pentru condițiile acțiunii $a_{PA_{OWL-S}}$ care exprimă toate condițiile ce trebuie satisfăcute pentru a se putea executa acțiunea $a_{PA_{OWL-S}}$ este:

$$Poss(a_{PA_{OWL-S}}, z) \equiv Holds(\varphi_1, z) \wedge Holds(\varphi_2, z) \wedge \dots \wedge Holds(\varphi_n, z) \wedge \\ Knows(\phi_1, z) \wedge \dots \wedge Knows(\phi_n, z)$$

Translatarea efectelor și ieșirilor unui proces atomic. În calculul fluentilor [Thi04], toate efectele unei acțiuni sunt codificate într-o axiomă de actualizare a stării. O axiomă de actualizare a stării descrie efectele unei acțiuni ca diferența dintre două stări: starea de dinaintea execuției acțiunii și starea rezultată în urma execuției acțiunii:

$$Poss(a(x), z) \wedge \Delta(x, State(z)) \supset State(Do(a(x), z)) = State(z) \circ v^+ - v^-$$

unde:

- $State(Do(a, z))$ – este starea rezultată în urma execuției acțiunii a în starea z ;
- $State(z)$ – este starea de dinaintea execuției acțiunii a în starea z ;
- v^- – sunt efectele negative (efectele care devin false în urma execuției acțiunii a în starea z);
- v^+ – sunt efectele pozitive (efectele care devin adevărate în urma execuției acțiunii a în starea z);
- $\Delta(x, State(z))$ – sunt constrângeri suplimentare prin care consecința dintr-o axiomă de actualizare a stării definește relația dintre o stare și starea sa succesoare.

În abordarea pe care o propunem, atât ieșirile cât și efectele unui proces atomic PA_{OWL-S} sunt translate, în calculul fluentilor, în efectele acțiunii $a_{PA_{OWL-S}}$ rezultate în urma translatării procesului PA_{OWL-S} .

O ieșire/un efect al unui proces atomic PA_{OWL-S} e translatată/traslatat în calculul fluentilor astfel:

$$State(Do(a_{PA_{OWL-S}}, z)) = State(z) \circ f_{eff/out_i}$$

unde:

- f_{eff/out_i} – este un fluent reprezentând o ieșire sau un efect al procesului atomic PA_{OWL-S} ;
- $a_{PA_{OWL-S}}$ – reprezintă acțiunea obținută din translatarea procesului atomic PA_{OWL-S} .

Dacă procesul atomic PA_{OWL-S} are mai multe ieșiri/efecte, formula se generalizează astfel:

$$State(Do(a_{PA_{OWL-S}}, z)) = State(z) \circ f_{out_1} \circ \dots \circ f_{out_i} \circ f_{eff_1} \dots \circ f_{eff_j}$$

Condiții suplimentare, $\Delta(x, State(z))$, sunt translatate automat din parametrii de intrare ai procesului atomic PA_{OWL-S} astfel:

$$Poss(a_{PA_{OWL-S}}, z) \wedge Knows(\varphi_1, z) \wedge \dots \wedge Knows(\varphi_n, z) \supset State(Do(a_{PA_{OWL-S}}, z))$$

unde:

- φ_i – este un fluent reprezentând o intrare a procesului atomic PA_{OWL-S} ;
- $a_{PA_{OWL-S}}$ – reprezintă acțiunea obținută din translatarea procesului atomic PA_{OWL-S} .

Pe baza formulelor anterior menționate se poate exprima axioma de actualizare a stării pentru acțiunea $a_{PA_{OWL-S}}$.

$$Poss(a_{PA_{OWL-S}}, z) \wedge Knows(\varphi_1, z) \wedge \dots \wedge Knows(\varphi_n, z) \supset \\ State(Do(a_{PA_{OWL-S}}, z)) = State(z) \circ f_{out_1} \circ \dots \circ f_{out_i} \circ f_{eff_{e_1}} \circ \dots \circ f_{eff_{e_j}}$$

3.2. Algoritmul de compunere a serviciilor Web

Algoritmul de compunere dezvoltat primește ca intrare o cerere de serviciu compus reprezentată ca un set de fluenți și returnează o secvență de *stagii* reprezentând planul de compunere. Planul de compunere obținut poate fi salvat ca o acțiune complexă $a_{complex}$ în calculul fluenților. Acțiunea complexă $a_{complex}$ constă dintr-un set de *stagii* înlănțuite secvențial. Un *stagi* este la rândul său o acțiune complexă alcătuită dintr-un set de acțiuni elementare care se execută în paralel. Setul de fluenți de intrare este obținut în urma translatării intrărilor serviciului compus solicitat și reprezintă starea inițială z_0 din care pornește algoritmul. Setul de fluenți obținut în urma translatării ieșirilor serviciului compus solicitat reprezintă starea finală z_g (starea scop) în care algoritmul încearcă să ajungă.

La fiecare pas i algoritmul selectează, din setul tuturor acțiunilor disponibile, doar acțiunile care se pot executa în starea curentă z_i . O acțiune a_j se poate executa în starea curentă z_i , dacă axioma pentru condițiile acțiunii a_j este satisfăcută în starea z_i . Starea curentă z_i se actualizează conform axiomelor de actualizare a stării pentru acțiunile selectate la pasul i . Starea nouă z_{i+1} se obține din vechea stare z_i la care se adaugă efectele acțiunilor care au fost executate la pasul i . Toate acțiunile selectate la pasul i corespund stagiului i . Algoritmul se termină, fie când s-a ajuns în stare finală z_g , fie când se ajunge într-o stare în care nu există nici o acțiune care poate fi selectată. Dacă algoritmul a ajuns într-o stare în care nu mai poate fi selectată nici o acțiune, dar care nu coincide cu starea finală z_g , înseamnă că nu există soluție de compunere care să satisfacă cererea dată. Dacă algoritmul a ajuns în starea finală z_g am obținut un plan de compunere P (un plan P corespunde unei soluții de compunere) pentru cererea dată. În acest moment se face o rafinare a planului de compunere P . Prin rafinare se urmărește eliminarea din planul de compunere a acțiunilor a

căror execuție produce efecte care nu sunt solicitate de utilizator, sau nu sunt precondiții pentru alte acțiuni selectate în orice pas intermediar al planului.

3.3. Concluzii

Contribuțiile originale ale acestui capitol sunt enumerate în cele ce urmează: (i) metoda de translatăre din descrieri semantice de servicii (OWL-S, SAWSDL [12]) în calculul fluenților; (ii) metoda de compunere automată a serviciilor Web folosind calculul fluenților; (iii) prototipul experimental pentru validarea/testarea abordării propuse. Avantajul metodei de compunere propuse este strâns legat de mecanismul de raționare bazat pe progresie din calculul fluenților. Într-o abordare bazată pe progresie efortul computațional este același pe parcursul execuției, independent de numărul de acțiuni executate.

4. Metodă de descoperire automată a serviciilor Web

Acest capitol prezintă o metodă de descoperire automată a serviciilor Web. Metoda evaluează gradul de potrivire semantică (*DoM*) dintre două servicii pe baza unui set de metrice originale. În abordarea noastră, am considerat că un serviciu Web este descris semantic printr-un set de concepte, aparținând unei ontologii $\mathcal{I}$. Setul de concepte asociat unui serviciu reprezintă descrierea semantică a parametrilor de intrare/ieșire ai serviciului. Deoarece conceptele, care descriu semantic serviciile, pot avea asociate proprietăți, în evaluarea *DoM*-ului dintre două servicii am considerat atât ierarhia de concepte, cât și ierarhia de proprietăți.

4.1. Metrice de evaluare a gradului de potrivire semantică dintre două servicii

Pentru evaluarea gradului de potrivire semantică dintre două servicii S_1 , S_2 am folosit principiul clasic al decompoziției unei probleme. Am considerat că procesul de evaluare a gradului de potrivire semantică dintre două servicii poate fi descompus în:

- 1) Evaluarea gradului de potrivire semantică dintre setul de concepte care descriu semantic intrările serviciului S_1 și setul de concepte care descriu semantic intrările serviciului S_2 ;
- 2) Evaluarea gradului de potrivire semantică dintre setul de concepte care descriu semantic ieșirile serviciului S_1 și setul de concepte care descriu semantic ieșirile serviciului S_2 ;
- 3) Evaluarea gradului de potrivire semantică dintre serviciul S_1 și serviciul S_2 pe baza gradelor de potrivire semantică evaluate la pașii 1) și 2).

La rândul lor, procesele descrise la pașii 1) și 2) pot fi descompuse în:

- 1.1) Evaluarea gradului de potrivire semantică dintre un concept c_i^1 care descrie semantic o intrare/ieșire a serviciului S_1 și un concept c_j^2 care descrie semantic o intrare/ieșire a serviciului S_2 pe baza ierarhiei de concepte;

1.2) Evaluarea gradului de potrivire semantică dintre setul de proprietăți asociate conceptului c_i^1 și setul de proprietăți asociate conceptului c_j^2 pe baza ierarhiei de proprietăți;

1.3) Combinarea rezultatelor de la pașii 1.1) și 1.2) în vederea obținerii unui scor global care reflectă gradul de potrivire dintre cele două concepte.

Metricile pe care le propunem, pentru evaluarea gradului de potrivire semantică dintre două servicii, extind/adaptează acuratețea învățării (eng. learning accuracy) LA [8]. Acuratețea învățării este o metrică folosită pentru evaluarea performanțelor metodelor de învățare și populare automată a ontologiilor. Formula cu care se calculează acuratețea învățării [8] este:

$$LA(p, c) = \frac{\delta(\text{root}, a) + 1}{\delta(\text{root}, a) + \delta(a, c) + \delta(a, p) + 1}$$

unde root reprezintă nodul rădăcină în ontologia Γ , a este *hipernimul* comun cel mai apropiat al nodurilor p și c , iar $\delta(c_1, c_2)$ denotă distanța dintre conceptele c_1 și c_2 în ontologia Γ . Distanța δ se calculează ca fiind numărul arcelor dintre conceptele c_1 și c_2 situate pe aceeași ramură în ontologia Γ .

4.2. Rezultate experimentale

Metoda propusă a fost testată și evaluată pe colecția OWLS–TCv3 propusă de [20] pentru evaluarea performanțelor algoritmilor de calcul al gradului de potrivire semantică. OWLS–TCv3 conține: (i) un set de ontologii pentru șapte domenii particulare (ex: *education*, *medical care*, *food*, *travel*, *communication*, *economy*, *weapons*), (ii) un set de 1007 de servicii Web descrise semantic în OWL–S și (iii) un set de 28 de interogări. O interogare este descrisă, în OWLS–TCv3, ca un proces atomic OWL–S. Experimentele au fost efectuate pe setul de interogări definit în OWLS–TCv3. Tabelul 4.1 prezintă rezultatele experimentale obținute pentru interogare *university_lecturer-in-academia_service.owls* din domeniul *education*.

Tabelul 4.1: Rezultatele experimentale obținute pentru interogarea *university_lecturer-in-academia_service.owls* din domeniul *education*

Servicii Regăsite	DoM
<i>university_lecturer-in-academiaCurrentSemester_service.owls</i>	1
<i>university_lecturer-in-academia_Recommendservice.owls</i>	1
<i>university_lecturer-in-academia_service.owls</i>	1
<i>university_senior-lecturer-in-academia_service.owls</i>	0.9697
<i>higher-educational-organization_lecturer-in-academia_MostUsedservice.owls</i>	0.9412
<i>higher-educational-organization_lecturer-in-academia_service.owls</i>	0.9412
<i>educational-organization_lecturer-in-academia_service.owls</i>	0.875
<i>learning-centred-organization_lecturer-in-academia_service.owls</i>	0.8
<i>organization_lecturer-in-academia_service.owls</i>	0.7143
<i>university_research-fellow-in-academia_service.owls</i>	0.6667

Pentru măsurarea acurateții metodei de descoperire am folosit două metrici: *precision* și *recall*. Am definit *precision* ca fiind numărul serviciilor relevante care au fost descoperite pentru o interogare, raportat la numărul total al serviciilor descoperite pentru acea interogare. *Recall*-ul a fost definit ca fiind numărul serviciilor relevante care au fost descoperite pentru o interogare, raportat la numărul total al serviciilor disponibile care sunt relevante pentru acea interogare.

În graficul din figura 4.1 sunt prezentate valorile pentru *precision* și *recall*, calculate pe baza rezultatelor experimentale obținute prin testarea metodei de descoperire pe setul de interogări definit în OWLS-TCv3. Valorile medii pentru *precision* și *recall* calculate pentru întregul set de interogări sunt: $precision_{mean} = 0,801$, $recall_{mean} = 0,959$.

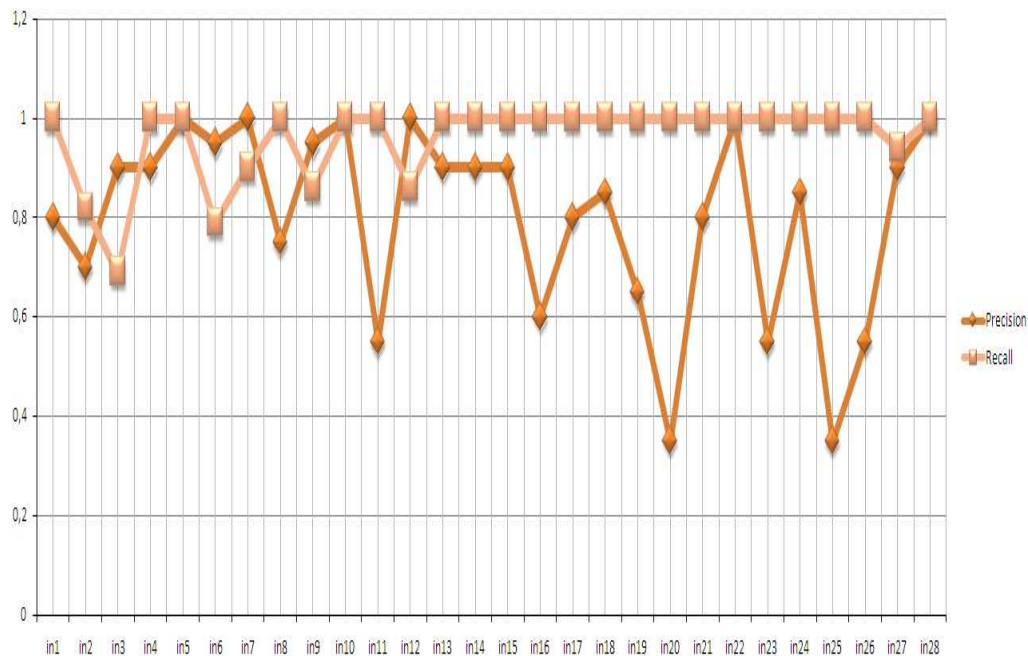


Figura 4.1: *Precision* și *recall* calculate pentru setul de interogări definit în OWLS-TCv3

4.3. Concluzii

Contribuțiile originale ale acestui capitol sunt: (i) setul de metrici propuse pentru evaluarea gradului de potrivire semantică dintre două servicii; (ii) algoritmul de descoperire automată a serviciilor Web care are la bază setul de metrici propuse; (iii) prototipul experimental pentru testarea/validarea abordării propuse. Rezultatele experimentale obținute au demonstrat validitatea abordării propuse.

Principalele avantaje ale metodei de descoperire prezentate sunt enumerate în cele ce urmează:

1. Metricile de evaluare a gradului de potrivire semantică dintre două servicii furnizează valori continue în intervalul $[0, 1]$. O astfel de abordare este utilă atunci

- când se lucrează cu seturi mari de servicii deoarece ne oferă posibilitatea de a face o diferențiere fină între serviciile candidate care satisfac cererea de serviciu solicitată.
2. O abordare în care, în evaluarea gradului de potrivire semantică dintre două servicii, se ia în considerare atât ierarhia de concepte, cât și ierarhia de proprietăți, furnizează rezultate mai exacte comparativ cu abordările care evaluează *DoM* pe baza ierarhiei de concepte.
 3. Metoda propusă evaluează gradul de potrivire semantică dintre două servicii într-o manieră eficientă. Eficiența derivă din codificarea descrierilor semantice de servicii ca un set de intervale. Setul de intervale asociat fiecărui concept se calculează pe baza schemei de etichetare propusă de Argawal [35]. Codificarea descrierilor semantice ale serviciilor ca un set de intervale permite verificarea relațiilor de subsumare dintre concepte din ontologie în timp constant, independent de dimensiunea arborelui ontologic.

5. Tehnică de compunere a serviciilor Web bazată pe grafuri de celule de servicii

Acest capitol prezintă o tehnică originală de compunere automată a serviciilor Web. Tehnica propusă constă din construirea unui graf orientat aciclic și căutarea acelor subgrafuri care reprezintă soluții de compunere pentru o cerere dată. Mulțimea nodurilor grafului este reprezentată de mulțimea operațiilor serviciilor disponibile, iar mulțimea arcelor grafului este reprezentată de mulțimea relațiilor de similaritate semantică dintre servicii. Pentru verificarea gradului de similaritate semantică dintre două servicii am considerat trei tipuri de potrivire semantică: *exact*, *plug-in* și *subsume* [14]. Gradul de similaritate semantică dintre două servicii se calculează pe baza relațiilor de subsumare existente între conceptele care descriu semantic ieșirile unui serviciu și conceptele care descriu semantic intrările celui alt serviciu. Graful astfel construit se actualizează în mod dinamic.

Determinarea subgrafurilor care reprezintă soluțiile de compunere pentru cererea dată este un proces de raționare și căutare bazată pe înlănțuire înapoi, similar cu comportamentul unui motor de raționare Prolog. Din totalitatea soluțiilor de compunere găsite, metoda selectează soluția care satisface cel mai bine preferințele și constrângerile impuse de utilizator.

Tehnica de compunere propusă are la bază un formalism original bazat pe grafuri de celule de serviciu. Considerăm că o astfel de abordare de compunere în care graful de celule de servicii este construit în background este o alegere potrivită și naturală pentru problema compunerii serviciilor. Argumentăm această afirmație în cele ce urmează. Fiecare serviciu poate primi/furniza date de la/la un set de servicii conform cu semanticile asociate intrărilor și ieșirilor serviciului, iar stabilirea conexiunilor între servicii (construirea grafului) se face în funcție de gradul de potrivire semantică dintre servicii – dacă gradul de potrivire semantică dintre două servicii este mai mare decât zero, atunci se stabilește o conexiune între cele două servicii. Deoarece în timp, valorile gradelor de potrivire semantică dintre servicii nu se modifică, o abordare în care calcularea gradelor de potrivire semantică și stabilirea conexiunilor se face într-un proces care lucrează în background și este invizibil utilizatorului este fezabilă. Avantajul unei astfel de abordare

este o îmbunătățire a timpului de execuție deoarece problema compunerii se reduce la o problemă de căutare a unei căi într-un graf.

În abordarea propusă am tratat problema compunerii serviciilor Web din punct de vedere practic și funcțional. Am identificat cerințele cheie pe care trebuie să le îndeplinească un sistem care compune automat servicii Web și am găsit un mod consistent și direct de a le rezolva. Scopul a fost de a crea o soluție fezabilă care poate lucra în lumea reală. Am încercat să păstrăm un nivel de echilibru între gradul de automatizare al procesului și gradul de control/confort pe care utilizatorul îl simte atunci când folosește sistemul. Acestea au fost premisele care au stat la baza dezvoltării prototipului experimental – am dezvoltat un sistem care lucrează împreună cu utilizatorul, acționând ca un consultant care recomandă acțiuni, dar care îi lasă utilizatorului libertatea de a lua deciziile finale.

Contribuțiile originale ale acestui capitol sunt: (i) definirea formală a abordării de compunere propuse; (ii) metoda de compunere automată a serviciilor Web bazată pe graful de celule de serviciu, (iii) specificarea și integrarea constrângerilor și preferințelor utilizatorului în procesul de compunere, (iv) definirea unui limbaj formal pentru stocarea grafului de celule de serviciu, numit *SGDL* (în eng. *service graph description language*).

6. Tehnică de compunere a serviciilor Web inspirată din sistemul imunitar

Acest capitol prezintă o tehnică originală de compunere automată a serviciilor Web, care combină o metodă de compunere a serviciilor bazată pe un graf de planificare îmbogățit cu o metodă de selecție a soluției optime de compunere inspirată din principiile sistemului imunitar. Metoda de compunere propusă îmbogățește graful de planificare clasic [18] cu conceptele *cluster de servicii*, *cluster de literal*, *legătură de similaritate semantică* și *matrice de legături semantice*. Pe baza grafului de planificare îmbogățit și a matricei de legături semantice, metoda de selecție determină, din setul soluțiilor disponibile, soluția optimă de compunere. Selecția soluției optime de compunere se face în funcție de preferințele utilizatorului cu privire la: (i) valorile individuale ale atributelor de QoS pentru soluția de compunere, (ii) valoarea globală de QoS a soluției de compunere, (iii) gradul de potrivire semantică dintre serviciile implicate într-o soluție de compunere.

6.1. Formalizarea problemei

Fie Γ ontologia folosită pentru descrierea semantică a serviciilor Web și formularea cererii utilizatorului.

Definiția 6.1 Definim o *cerere de serviciu compus* ca o triplă:

$$scr = (in, out, w)$$

unde:

- $in = \{in_i | in_i \in \Gamma\}$ – este setul de concepte care descrie semantic intrările serviciului compus solicitat;

- $out = \{out_i \mid out_i \in I\}$ – este setul de concepte care descrie semantic ieșirile serviciului compus solicitat;
- $w = (w_{QoS_{ind}}, w_{QoS_{glob}}, w_{sem}) \mid w_{QoS_{ind}}, w_{QoS_{glob}}, w_{sem} \in [0, 1]$ – este un set de ponderi furnizate de utilizator.

Notă:

1. $w_{QoS_{ind}} = \{w_{QoS_1}, \dots, w_{QoS_n}\}$ – reprezintă preferințele utilizatorului cu privire la valori individuale ale atributelor de QoS (ex: *preț*, *timp de răspuns*, etc.) pentru soluția de compunere.
2. Ponderile $w_{QoS_{glob}}, w_{sem}$ reprezintă preferințele utilizatorului cu privire la:
 - 2.1 valoarea globală de QoS a soluției de compunere;
 - 2.2 valoarea scorului de similaritate semantică a soluției de compunere.

Definiția 6.2: Definim un *serviciu Web* ca o triplă:

$$ws = (f, in, out)$$

unde:

- $f \in I$ – reprezintă un concept ontologic care descrie semantic operația asociată serviciului;
- $in = \{in_i \mid in_i \in I\}$ – reprezintă setul de concepte care descrie semantic setul parametrilor de intrare ai operației serviciului;
- $out = \{out_i \mid out_i \in I\}$ – reprezintă setul de concepte care descrie semantic setul parametrilor de ieșire ai operației serviciului.

Definiția 6.3: Definim un *serviciu Web virtual*, vs ca un serviciu fără o funcționalitate asociată. Un *serviciu virtual* vs poate avea fie intrări, fie ieșiri.

Remarcă 6.1: Considerăm două servicii virtuale $vs_{out} = (\emptyset, \emptyset, scr.in)$ și $vs_{in} = (\emptyset, scr.out, \emptyset)$.

Notă:

1. vs_{out} este un serviciu virtual care nu are funcționalitate și intrări, dar are ieșiri. Ieșirile serviciului vs_{out} reprezintă intrările extrase din cererea de serviciu Web compus scr specificată de utilizator;
2. vs_{in} este un serviciu virtual care nu are funcționalitate și ieșiri, dar are intrări. Intrările serviciului vs_{in} reprezintă ieșirile extrase din cererea de serviciu Web compus scr specificată de utilizator;

Definiția 6.4: Definim un *cluster de servicii* ca fiind:

$$sc = \{ws_k \mid \forall ws_i, ws_j \in sc, (ws_i.f \equiv ws_j.f \vee ws_i.f \subseteq ws_j.f \vee ws_i.f \supseteq ws_j.f) \wedge (\forall ws_i.in_m \in ws_i.in, \forall ws_j.in_n \in ws_j.in \rightarrow ws_i.in_m \equiv ws_j.in_n \vee ws_i.in_m \subseteq ws_j.in_n \vee ws_i.in_m \supseteq ws_j.in_n)\}$$

Notă: Un *cluster de servicii* grupează concepte care descriu semantic un set de servicii care au aceeași funcționalitate și același set de parametri de intrare și care sunt în relații de tipul: *exact* ($\equiv$), *plugin* ($c_i \subseteq c_j$) și *subsume* ($c_i \supseteq c_j$).

Definiția 6.5: Definim un *literal* ca fiind:

$$l = \{l_i \mid \forall vs_{in}, vs_{out}, ws_j \in WS, l_i \in \{vs_i.in, vs_o.out, ws_j.in, ws_j.out\}\}$$

Notă: Un literal este un concept care descrie semantic o intrare/ieșire a unui serviciu generic ws_i sau a unui serviciu virtual vs_{in}, vs_{out} .

Definiția 6.6: Definim un *cluster de literal* ca fiind:

$$lc = \{l_k \mid \forall l_i, l_j \in lc, l_i \equiv l_j \vee l_i \subseteq l_j \vee l_i \supseteq l_j\}.$$

Notă: Un *cluster de literal* grupează concepte care descriu semantic același parametru de intrare/ieșire al unor servicii Web. Între două concepte care descriu semantic intrările/ieșirile a două servicii care fac parte dintr-un cluster pot exista următoarele relații în ontologia Γ : *exact* ($\equiv$), *plugin* ($c_i \subseteq c_j$) și *subsume* ($c_i \supseteq c_j$).

Definiția 6.7: Definim un *graf de planificare îmbogățit pentru compunerea serviciilor Web* ca un set de tuple:

$$GPI = \{(A_i, L_i)\}$$

unde:

- i – reprezintă nivelul în GPI ;
- $A_i = \{sc_j^i \mid sc_j^i \in i\}$ – reprezintă setul de *clusteri de servicii* de pe nivelul i din GPI ;
- $L_i = L_{i-1} \cup \{lc_m^i \mid lc_m^i \in i\}$ – reprezintă setul de *clusteri de literal* de pe nivelul i din GPI .

Notă: (A_0, L_0) este un caz particular în care $A_0 = \emptyset$ și $L_0 = vs_{in}.in$, iar (A_n, L_n) reprezintă ultimul nivel din graful de planificare îmbogățit care are proprietatea că $vs_{out}.out \subseteq L_n$.

Definiția 6.8: Definim o *soluție de compunere* care satisface o cerere de serviciu scr ca fiind:

$$sol = \{\forall i \in GPI \wedge \forall sc_j^i \in i (\exists!) ws_{j_k}^i \in sc_j^i \mid ws_{j_k}^i \in sol\}$$

unde:

- i – reprezintă nivelul i din GPI ;
- sc_j^i – este clusterul j de pe nivelul i din GPI ;
- $ws_{j_k}^i$ – este serviciul k din clusterul j de pe nivelul i din GPI .

6.2. Construirea grafului de planificare îmbogățit pentru compunerea serviciilor Web

Construirea *grafului de planificare îmbogățit pentru compunerea serviciilor Web* este un proces iterativ în care, la fiecare iterație, se adăugă un nou nivel la graful de planificare îmbogățit curent. Serviciile Web semantice, care sunt considerate pentru extinderea grafului de planificare îmbogățit, sunt furnizate prin procesul de descoperire. Procesul de descoperire identifică serviciile pentru care există un grad de potrivire semantică (eng. *degree of match*) (*DoM*) între setul de concepte ontologice care descriu intrările serviciilor și setul de literală de pe nivelul curent din graful de planificare.

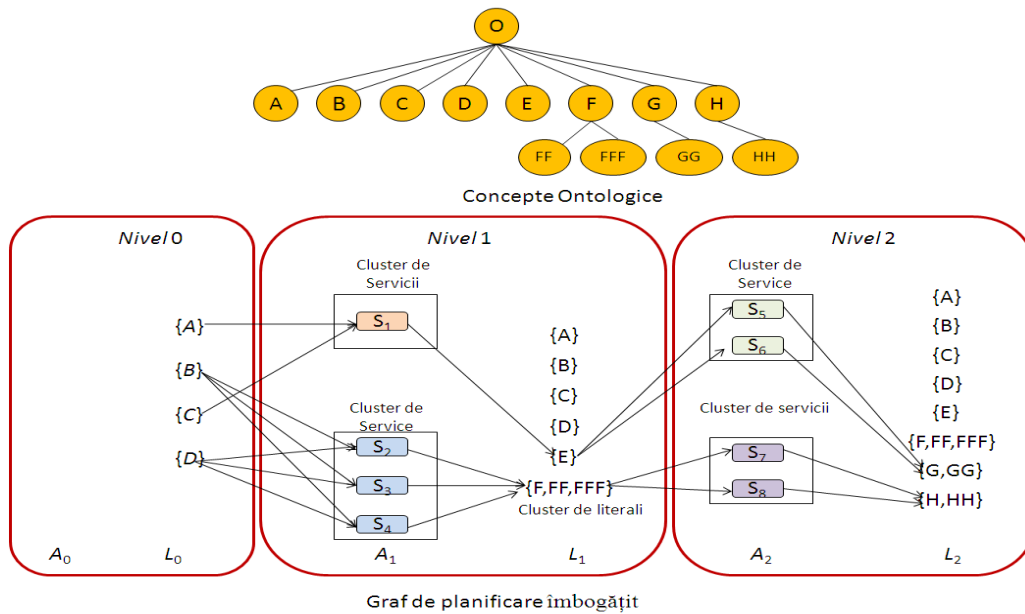


Figura 6.1: Exemplu de graf de planificare îmbogățit

6.3. Matricea de legături semantice

Matricea de legături semantice se construiește iterativ în timpul procesului de compunere. În construirea matricei se ține cont de informația semantică colectată în timpul procesului de descoperire și în timpul construirii grafului de planificare îmbogățit. Liniile și coloanele matricei sunt etichetate cu servicii din *graful de planificare îmbogățit*. O coloană și o linie din *MLS* care au același indice sunt etichetate cu același serviciu. O celulă a matricei mls_{ij} păstrează informații cu privire la *legătura de similaritate semantică* stabilită între serviciul de pe linia i și serviciul de pe coloana j din matrice.

Definiția 6.9: (*Legătură de similaritate semantică*):

Definim o *legătură de similaritate semantică* între două servicii ws_i, ws_j ca fiind $sl_{ij} = \{(ws_j.out_k, ws_p.in_l) \mid DoM(ws_j.out_k, ws_p.in_l) > 0, ws_j.out_k \in ws_j.out, ws_p.in_l \in ws_p.in\}$

Pentru evaluarea gradului de potrivire semantică dintre două servicii am adaptat și extins metricile *precizie* și *completitudine* (eng. *precision*, *recall*) din domeniul regăsirii informațiilor [26]. Am considerat că există un grad de *DoM* între un concept ontologic ($ws_i.out_k$) care descrie semantic un parametru de ieșire al unui serviciu ws_i și un concept ontologic ($ws_j.in_l$) care descrie semantic un parametru de intrare al unui serviciu ws_j dacă între cele două concepte există următoarele relații [14]: (i) cele două concepte sunt echivalente – potrivire *exactă*; (ii) conceptul $ws_i.out_k$ este subsumat de conceptul $ws_j.in_l$, – potrivire de tip *plugin*; (iii) conceptul $ws_i.out_k$ subsumează conceptul $ws_j.in_l$ – potrivire de tip *subsume*; (iv) nu există nici un tip de relație între conceptul $ws_i.out_k$ și conceptul $ws_j.in_l$ – potrivire de tip *failed*.

6.4. Metoda de selecție a soluției optime de compunere

Metoda de selecție a soluției optime de compunere propusă în această capitole se bazează pe o versiune adaptată și îmbogățită a algoritmului CLONALG [6, 7], algoritmul inspirat din principiul de selecție a clonelor în sistemul imunitar. Metoda de selecție folosește graful de planificare îmbogățit pentru compunerea serviciilor Web *GPI*, matricea de legături semantice *MLS* și o funcție de fitness *QF* pentru identificarea și ordonarea soluțiilor de compunere valide.

Am definit funcția de fitness *QF* ca fiind:

$$QF(sol) = \frac{w_{QoS_{glob}} * QoS(sol) + w_{Sem} * Sem(sol)}{w_{QoS_{glob}} + w_{Sem}}$$

unde:

- $QoS(sol)$ – reprezintă valoarea globală de *QoS* a soluției de compunere, *sol*;
- $Sem(sol)$ – reprezintă scorul de similaritate semantică al soluției de compunere, *sol*;
- $w_{QoS_{glob}}, w_{Sem} \in [0,1]$ – sunt ponderi furnizate de utilizator. Aceste ponderi reflectă preferințele utilizatorului cu privire la valoarea globală de *QoS* și la scorul de similaritate semantică al soluției de compunere, *sol*.

Aplicarea metodei de selecție pe o tuplă de forma (*GPI*, *MLS*, *QF*) necesită următoarele inițializări: definirea unei formule pentru determinarea numărului n_c de clone care vor fi generate pentru fiecare soluție de compunere și definirea unei formule care va calcula rata de mutație pe baza valorii *afinității*.

Pentru calcularea numărului de clone care vor fi generate pentru fiecare soluție de compunere am folosit formula [CZ02]:

$$n_c = round(\beta * n)$$

unde:

- β – este un factor de multiplicitate;
- n – este numărul total de soluții de compunere obținute până în acel moment.

Am definit rata de mutație m_r ca fiind:

$$m_r(sol) = round \left(\frac{\alpha * |sol|}{n-i+1} \right)$$

unde:

- α – este un factor de multiplicitate;
- $|sol|$ – reprezintă numărul de servicii Web conținute în soluția de compunere sol ;
- n – este numărul total de soluții de compunere obținute până în acel moment;
- i – este indexul corespunzător fiecărei soluții de compunere ($i=1$ corespunde soluției de compunere cu cea mai mare afinitate).

Valorile pe care le pot lua factorii de multiplicitate β și α depind de domeniul problemei pe care o rezolvă. În abordarea de selecție propusă în acest capitol am considerat $\alpha = \beta = 1$.

Rezultatul procesului de selecție este un set ordonat de soluțiilor de compunere. Ordonarea soluțiilor de compunere se face pe baza criteriilor non-funcționale (ex: *preț*, *timp de execuție*, *calitate semantică*, etc.) specificate de utilizator pentru serviciul compus solicitat.

6.5. Prototip experimental

Pentru validarea abordării propuse s-a dezvoltat un prototip experimental. Prototipul experimental a fost testat și evaluat pe un set de scenarii din domeniul planificării unei călătorii și al planificării de evenimente sociale. Considerăm pentru exemplificare scenariul clasic al planificării unei călătorii (tabel 6.1).

Tabel 6.1: Cererea utilizatorului pentru scenariul din domeniul trip planning

Intrări	Ieșiri	QoS Golbal	QoS	Calitate Sem.
<i>Country, DestinationCity, SourceCity, StartDate, NumberOfPerson, NumberOfRooms, NumberOfDays, Hotel, SingleRoom</i>	<i>FlightPrice, FlightConfirmation, Accommodation Confirmation</i>	0.25	<i>Av</i> = 0.15 <i>Rel</i> = 0.25 <i>Cost</i> = 0.45 <i>RT</i> = 0.15	0.75

Pentru cererea prezentată în tabelul 6.1, algoritmul de selecție identifică, pe baza *grafului de planificare îmbogățit* și a *matricei de legături semantice* un set de 20 soluții de compunere valide ordonate după valoarea afinității (tabelul 6.2).

Tabel 6.2: Soluții de compunere ordonate după valoarea afinității

Nr. Sol	Afinitate	Scorul de QoS	Scorul calității semantice	Soluție
1	5.223	3.251	5.881	{ [S ₁₁₂ , S ₁₂₃]; [S ₂₁₁ , S ₂₂₃] }
2	5.216	2.956	5.97	{ [S ₁₁₂ , S ₁₂₁]; [S ₂₁₁ , S ₂₂₁] }
3	5.214	3.176	5.894	{ [S ₁₁₂ , S ₁₂₁]; [S ₂₁₁ , S ₂₂₃] }
4	5.184	3.206	5.844	{ [S ₁₁₂ , S ₁₂₃]; [S ₂₁₂ , S ₂₂₃] }
5	5.177	2.911	5.933	{ [S ₁₁₂ , S ₁₂₁]; [S ₂₁₂ , S ₂₂₁] }
6	5.176	3.131	5.858	{ [S ₁₁₂ , S ₁₂₁]; [S ₂₁₂ , S ₂₂₃] }
7	5.159	3.079	5.852	{ [S ₁₁₂ , S ₁₂₁]; [S ₂₁₁ , S ₂₂₂] }
8	5.121	3.034	5.816	{ [S ₁₁₂ , S ₁₂₁]; [S ₂₁₂ , S ₂₂₂] }
9	5.117	3.031	5.812	{ [S ₁₁₂ , S ₁₂₃]; [S ₂₁₁ , S ₂₂₁] }
10	5.115	3.154	5.768	{ [S ₁₁₂ , S ₁₂₃]; [S ₂₁₁ , S ₂₂₂] }
11	5.078	2.986	5.776	{ [S ₁₁₂ , S ₁₂₃]; [S ₂₁₂ , S ₂₂₁] }
12	5.076	3.109	5.731	{ [S ₁₁₂ , S ₁₂₃]; [S ₂₁₂ , S ₂₂₂] }
13	5.054	3.083	5.711	{ [S ₁₁₁ , S ₁₂₃]; [S ₂₁₁ , S ₂₂₃] }
14	5.047	2.788	5.8	{ [S ₁₁₁ , S ₁₂₁]; [S ₂₁₁ , S ₂₂₁] }
15	5.045	3.008	5.724	{ [S ₁₁₁ , S ₁₂₁]; [S ₂₁₁ , S ₂₂₃] }
16	4.999	3.045	5.65	{ [S ₁₁₂ , S ₁₂₂]; [S ₂₁₁ , S ₂₂₃] }
17	4.989	2.911	5.682	{ [S ₁₁₁ , S ₁₂₁]; [S ₂₁₁ , S ₂₂₂] }
18	4.984	3.038	5.632	{ [S ₁₁₁ , S ₁₂₃]; [S ₂₁₂ , S ₂₂₃] }
19	4.977	2.743	5.722	{ [S ₁₁₁ , S ₁₂₁]; [S ₂₁₂ , S ₂₂₁] }
20	4.975	2.963	5.646	{ [S ₁₁₁ , S ₁₂₁]; [S ₂₁₂ , S ₂₂₃] }

6.5. Concluzii

Contribuțiile originale ale capitolului sunt: (i) modelul de graf de planificare îmbogățit, (ii) metricile de evaluare a gradului de potrivire semantică dintre două servicii, (iii) matricea de legături semantice, (iv) metoda de selecție a soluției optime de compunere bazată pe principiile sistemului imunitar. Modelul de graf de planificare îmbogățit extinde graful de planificare clasic cu conceptele de *cluster de literal*, *cluster de servicii* și *legătură de similaritate semantică*. Metricile de evaluare a gradului de potrivire semantică dintre două servicii au fost dezvoltate pornind de la definiția clasică a metricilor *precizia* și *completitudine* din domeniul *regăsirii informațiilor*. Matricea de legături semantice păstrează informații cu privire la legăturile de similaritate semantică stabilite între serviciile din graf. Metoda de selecție a soluției optime de compunere selectează soluția optimă de compunere luând în considerare atât atributele de QoS ale serviciilor cât și calitatea semantică a legăturilor dintre serviciile care fac parte din soluție.

Rezultatele experimentale obținute au arătat că o astfel de abordare hibridă care combină planificarea din domeniul AI cu un algoritm inspirat din principiile sistemului imunitar poate fi aplicată cu succes în rezolvarea problemei selecției soluției optime de compunere. Argumentăm această afirmație prin următoarele: graful de planificare îmbogățit împreună cu matricea de legături semantice furnizează spațiul de căutare pentru algoritmul de selecție. Am ales să adaptăm algoritmul CLONALG, deoarece acesta converge mai bine spre optimul global [6,7] (comparativ cu algoritmi genetici care pot stagna într-un optim local). Convergența este asigurată prin introducerea în mod iterativ a noi soluții candidate care largesc spațiul de căutare. Un alt avantaj al algoritmului CLONALG este că fundamentele matematice care stau la baza algoritmului sunt foarte simple.

7. Concluzii finale

În această teză au fost prezentate o metodă de descoperire automată a serviciilor Web și trei tehnici originale de compunere automată a serviciilor Web: (i) o tehnică de compunere a serviciilor Web folosind calculul fluenților, (ii) o tehnică de compunere a serviciilor bazată pe grafuri de celule de serviciu și (iii) o tehnică de compunere a serviciilor inspirată din principiile sistemului imunitar. Tehnicile de compunere prezentate pot fi încadrate în clasificarea făcută în capitolul 2 astfel: (i) tehnica de compunere folosind calculul fluenților se încadrează în categoria tehnicilor de compunere a serviciilor Web folosind planificarea din domeniul AI (eng. *artificial intelligence*), (ii) tehnica de compunere bazată pe grafuri de celule de serviciu se încadrează în categoria tehnicilor de compunere bazate pe grafuri, iar (iii) tehnica de compunere a serviciilor Web inspirată din principiile sistemului imunitar este o abordare hibridă care combină o metodă de compunere a serviciilor Web inspirată din AI planning cu o metodă de selecție a soluției optime de compunere inspirată din biologie.

Contribuțiile originale introduse în capitolele 2, 3, 4, 5 și 6 sunt enumerate în continuare:

1. O analiză critică a domeniului de cercetare privind compunerea automată a serviciilor Web.
2. O tehnică originală de compunere automată a serviciilor Web folosind calculul fluenților alcătuită din:
 - 2.1. Metoda de translatăre din descrieri semantice de servicii în calculul fluenților;
 - 2.2. Metoda de compunere automată a serviciilor Web folosind calculul fluenților;
 - 2.3. Prototip experimental pentru validarea/testarea abordării propuse;
3. O metodă originală de descoperire automată a serviciilor Web alcătuită din:
 - 3.1. Setul de metrici originale pentru evaluarea gradului de potrivire semantică dintre două servicii;
 - 3.2. Algoritm de descoperire automată a serviciilor Web bazat pe metricile propuse;
 - 3.3. Prototip experimental pentru testarea/validarea abordării propuse;
4. O tehnică originală de compunere automată a serviciilor Web bazată pe grafuri de celule de servicii alcătuită din:
 - 4.1. Definirea formală a abordării propuse;

- 4.2. Algoritm de compunere automată a serviciilor bazat pe grafuri de celule de servicii;
- 4.3. Definirea unui limbaj formal pentru stocarea grafului de celule de servicii;
- 4.4. Prototipul experimental pentru validarea/testarea abordării propuse;
- 5. O tehnică originală de compunere automată a serviciilor Web folosind principii inspirate din sistemul imunitar alcătuită din:
 - 5.1. Modelul de graf de planificare îmbogățit pentru compunerea serviciilor Web;
 - 5.2. Matricea de legături semantice;
 - 5.3. Metoda de compunere automată a serviciilor Web bazată pe modelul de graf de planificare îmbogățit;
 - 5.4. Metoda de selecție a soluției optime de compunere inspirată din principiile sistemului imunitar.

7.1. Dezvoltări în desfășurare

Următoarele cercetări se află în derulare:

Pentru tehnica de compunere folosind calculul fluenților propunem extinderea calculului fluenților cu logici de descriere. Rolul acestei extinderi este de a permite raționarea cu privire la tipurile de potrivire semantică posibile între servicii.

Pentru tehnica de compunere bazată pe grafuri de celule de servicii, prezentăm câteva dintre ideile de dezvoltare: (i) considerarea precondițiilor și efectelor în descrierea semantică a serviciilor Web; (ii) îmbunătățirea algoritmului de compunere astfel încât să fie capabil să trateze și să tolereze erori de orice tip (ex: erori de tipuri de date, erori de conexiune, erori de rețea sau chiar și erori ale clientului, specifice pentru fiecare serviciu particular); (iii) dezvoltarea unui set de politici care să îi ofere utilizatorului o gamă variată de posibilități pentru exprimarea preferințelor/ constrângerilor.

Pentru tehnica de compunere inspirată din principiile sistemului imunitar prezentăm câteva dintre ideile de dezvoltare: (i) considerarea precondițiilor și efectelor în descrierea semantică a serviciilor Web; (ii) combinarea principiilor de selecție inspirate din sistemul imunitar cu tehnici de *reinforcements learning* în vederea îmbunătățirii performanțelor algoritmului de selecție a soluției optime de compunere.

Un alt aspect de cercetare care trebuie investigat este legat de ontologie. În absența unei ontologii standard pe baza căreia furnizorii de servicii să-și definească și să își înregistreze serviciile, există posibilitatea ca furnizorii din aceeași categorie să folosească același termen pentru a înțelege lucruri diferite sau termeni diferiți pentru a înțelege același lucru. Cu alte cuvinte furnizorii pot folosi ontologii diferite pentru a-și defini/înregistra serviciile. Într-o astfel de situație se pune problema cum integrăm diferite ontologii (concept cunoscut sub numele de “*alinarea ontologiilor*”) care descriu aceleași cunoștințe, dar conțin concepte și relații diferite.

8. Bibliografie Selectivă

1. Bowen, J. P., Hinchey, M. G.: *Applications of Formal Methods*, Prentice Hall PTR, Upper Saddle River, NJ, (1995).
2. Berardi, D., Calvanese, D., Giacomo, G. D., Hull, R. and Mecella M.: *Automatic Composition of Web Services in Colombo*, In the Proceedings of the 13th Italian Symposium on Advanced Database Systems, pp. 8–15, (Brixen-Bressanone), Italy, (2005).
3. Berardi, D., Calvanese, D., Giuseppe, D. G., Hull, R., and Mecella, M.: *Automatic Composition of Transition-based Semantic Web Services with Messaging*, In the Proceedings of the 31st International Conference on Very Large Databases (VLDB), pp. 613–624, (Trondheim) Norway, (2005).
4. Casati, F. Ilnicki, S., Krishnamoorthy, L. J. J., Ming-Chien Shan, V.: *eFlow: A platform for developing and managing composite e-services*, Academia/Industry Working Conference on Research Challenges (Aiworc 2000), pp.: 341–348, (Buffalo, NY) USA, (2000).
5. Christophides, V., Plexousakis, D., Scholl, M., Tourtounis, S.: *On labeling schemes for the semantic web*, In the Proceedings of the 12th international conference on World Wide Web, pp: 544 – 555, (Budapest) Hungary, (2003)
6. Castro, L., von Zuben, F.: *Learning and Optimization using the Clonal Selection Principle*, In the IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems, vol. 6, n. 3, pp. 239-251 (2002).
7. Castro, L. N., Timmis, J. I.: *Artificial immune systems as a novel soft computing paradigm*, SJournal Soft Computing - A Fusion of Foundations, Methodologies and Applications, 7(8): 526-544, (2003).
8. Hahn, U., Schnattinger, K.: *Towards text knowledge engineering*, In the Proceedings of the 10th Conference on Innovative Applications of Artificial Intelligence, pp. 524-531, (Madison, Wisconsin) USA, (1998).
9. Heuvel, W. J. van den, Maamar, Z.: *Moving toward a framework to compose intelligent Web services*, Communications of the ACM, 46(10), pp. 103-109, (2003).
10. Hashemian, S. V., and Mavaddat, F.: *A Graph-Based Approach to Web Services Composition*, In the Proceedings of the IEEE/IPSJ International Symposium on Applications and the Internet, pp.: 183–189, IEEE Computer Society, ISBN 0-7695-2262-9, (2005).
11. Hofmeyr, S.A.: *An interpretative introduction to the immune system*, in Design Principles for the Immune System and other Distributed Autonomous Systems, edited by I. Cohen and L.A. Segel, Oxford University Press: New York, (2000).

12. Kopecky, J.: *SAWSDL, Semantic Annotations for WSDL and XML Schema*, IEEE Internet Computing, 11(6): pp. 60-67, (2007).
13. Ponnekanti, R., and Fox, A.: *SWORD: A developer toolkit for Web service composition*, In the Proceedings of the 11th World Wide Web Conference, (Honolulu, HI) USA, (2002).
14. Paolucci, M., Kawamura, T., Payne, T. R, and Sycara, K. P.: *Semantic matching of web services capabilities*, In the Proceedings of the International Semantic Web Conference, pp. 333–347, (Sardinia) Italy, (2002).
15. Piccinelli, G., Williams, S. L.: *Workflow: A Language for Composing Web Services*, Conference on Business Process Management, Eindhoven, Netherlands, (2003).
16. Rao, J., Küngas, P., and Matskin, M.: *Application of Linear Logic to Web service composition*, In the Proceedings of the 1st International Conference on Web Services, (Las Vegas) USA, (2003).
17. Rao, J., Küngas, P., and Matskin, M.: *Logic-based Web services composition: from service description to process model*, In the Proceedings of the 2004 International Conference on Web Services, (San Diego) USA, (2004).
18. Russell, S., and Norvig, P., *Artificial Intelligence: A Modern Approach*, 2003, ISBN0-13-080302-2.
19. Sirin, E., Parsia, B., Wu, D., Hendler, J., and Nau, D.: *HTN Planning for Web Service Composition using SHOP*, Journal of Web Semantics, 1(4): pp. 377–396, (2004).
20. <http://www.semwebcentral.org/projects/owlstc/>
21. Thielscher, M.: *Introduction to the fluent calculus*, In Electronic Transactions on Artificial Intelligence, 2(3–4): pp. 179–192, (1998).
22. Thielscher, M.: *The fluent calculus: A specification language for robots with sensors in nondeterministic, concurrent, and ramifying environments*, In Technical Report CL-2000-01, Artificial Intelligence Institute, Dresden University of Technology, (2000).
23. Thielscher, M.: *Programming of Reasoning and Planning Agents with FLUX*, In the Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann Publishers, (2002).
24. Thielscher, M.: *Logic-based agents and the frame problem: A case for progression*, First-Order Logic Revisited: In Proceedings of the First Order Logic 75 (FOL75) Conference, Hendricks et al. (eds.) (2004).
25. Xu, J., Reiff-Marganiec, S.: *Towards Heuristic Web Services Composition Using Immune Algorithm*, , In the Proceedings of the 2008 IEEE International Conference on Web Services, pp. 238-245, (Beijing) China (2008).

26. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: *Performance measures for information extraction*, , In the Proceedings of the of DARPA Broadcast News Workshop, (Herndon, VA), Canada, (1999).
27. Sabou, M., Gracia, J., Angeletou, S., d'Aquin, M., Motta, E.: *Evaluating the Semantic Web: A Task-based Approach*, the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference, (Busan) Korea, (2007).