

RAPORT ȘTIINȚIFIC ȘI TEHNIC

Perioada 15 Aprilie -15 Decembrie 2013

Proiect bilateral Romania-Moldova

Contract Nr. 681

Titlul proiectului: ASDEC: Argumentare structurată pentru suportul deciziilor cu constrângeri normative
Partener roman: Universitatea Tehnică din Cluj-Napoca
Partener străin: Universitatea Tehnică din Moldova
Durata proiectului bilateral: 20 luni (15-Aprilie 2013-15 decembrie 2014)



Cuprins

1	Obiective și rezultate	3
1.1	Obiectivele generale	3
1.2	Obiectivele fazei de execuție	3
1.3	Publicații	3
1.4	Livrabile	5
2	Descrierea contribuțiilor științifice	7
2.1	Suport decizional orientat e-business în cazul încălcării contractelor între companii	7
2.1.1	Motivație	8
2.1.2	Reprezentarea contractelor în logica descriptivă	8
2.1.3	Gestiunea colaborativă a contractelor	11
2.1.4	Arhitectura sistemului	12
2.1.5	Contribuții	15
2.2	Utilizarea limbajului natural în explicarea deciziilor recomandate de sistem.	16
2.2.1	Motivație	16
2.2.2	Linked Data și proiectul Bio2RDF	16
2.2.3	Accesul la date În Linked Data	17
2.2.4	Analiza interogării bazată pe sabloane structurale	18
2.2.5	Construirea interogării SPARQL	21
2.2.6	Contribuții	21

Capitolul 1

Obiective și rezultate

1.1 Obiectivele generale

Obiectivul științific general este de a dezvolta tehnici și metodologii pornind de la teoria argumentării pentru a facilita agentului uman i) înțelegerea și analiza problemei; ii) construirea și justificarea deciziei; iii) explicarea soluției altor actori. Două obiective derivă din acesta. Primul urmărește amplificarea capabilităților de raționare ale agentului decizional prin creșterea vizibilității dovezilor și argumentelor structurate și prin înțelegerea diferitelor aspecte ale problemei. Al doilea se referă la augmentarea colaborării între agenți economici din țări diferite prin oferirea de instrumente adecvate de suport a deciziilor colaborative sub constrângeri normative.

1.2 Obiectivele fazei de execuție

Obiectivele specifice sunt prezentate în tabelul 1.1, unde O1-O3 sunt planificate în primul an și privesc cercetarea fundamentală, iar O4-O7 în al doilea an, cu focalizare pe cercetarea aplicativă.

1.3 Publicații

Rezultatele obținute au fost prezentate în 7 lucrări științifice [2, 16, 8, 15, 20, 14, 11, 13]:

1. Radu Balaj and Adrian Groza. Detecting influenza epidemics based on real-time semantic analysis of Twitter streams. Modelling and Development of Intelligent Systems, 10-12 October 2013, Sibiu, Romania.
2. Gabriela Visinari and Adrian Groza. Semantic-based monitoring of e-contracts. Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013,

Plan	Obiectiv	Noutate	Activități Asociate
Iulie 2013	O1. Analiza situației și fundamentarea tehnico-științifică a sistemelor suport a deciziei în contextul teoriei argumentării.	Identificarea posibilităților de integrare a deciziilor non-monotone, normelor și teoriei argumentării.	Analiza impactului sistemelor suport ale deciziei în reingineria proceselor. Redactarea caietului de sarcini și a specificațiilor.
Sep 2013	O2. Construirea modelului argumentativ pentru suport decizional sub constrângeri normative.	Utilizarea logicii justificative non-monotone. Integrarea logicii justificative cu cea descriptivă.	Construirea nivelului logic, dialectic și procedural al modelului.
Ian 2014	O3. Dezvoltarea unui sistem pentru suport argumentativ a deciziei.	Dezvoltarea unui limbaj formal de suport decizional cu elemente argumentative. Dezvoltarea de metode și algoritmi de etichetare a lanțului argumentativ.	Identificarea instrumentației tehnologice pentru implementarea modelului. Testarea și validarea sistemului.
Mar 2014	O4. Aplicarea sistemului în eficientizarea proceselor de audit.	Formalizarea protocoalelor, dialogului și a tipurilor de dovezi schimbate în procesul de audit al proiectelor FP7.	Eficientizarea gestiunii evidențelor și a explicațiilor în procesul de audit. Dezvoltarea unui prototip pentru auditarea proiectelor FP7.
Mai 2014	O5. Aplicarea sistemului în decizii colaborative de business sub sisteme normative diferite.	Identificarea diferențelor normative în drept comercial între România și Moldova.	Ilustrarea capacităților colaborative de suport decizional pe agenți economici parteneri din Moldova și România.
Sep 2014	O6. Aplicarea sistemului pentru validarea produselor conform cu standarde de calitate.	Asigurarea suportului decizional pe baza argumentării structurate în scopul reingineriei unor procese de producție.	Specificarea formală a standardelor de calitate de tip ISO. Monitorizarea unui sistem de producție. Utilizarea datelor obținute pentru suport argumentativ în vederea etichetării produselor conform cu standardele curente.
Noi 2014	O7. Dezvoltarea unei metodologii de suport decizional sub constrângeri normative.	Reutilizarea procedurilor și modelelor decizionale în diferite domenii supuse constrângerilor normative. Reingineria proceselor economice și de producție.	Definirea particularităților deciziilor constrânse normativ. Definirea schemelor de argumentare utilizate de factorii decizionali în sisteme economice și de producție.

Tabelul 1.1: Planificarea obiectivelor specifice și caracterul inovativ al acestora.



- Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti, pages 161-164.
3. Adrian Popescu and Anca Marginean. Multi-agent system for traffic monitoring. Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013, Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti, pages 181-184.
 4. Anca Marginean and Oana Marc, Querying LinkData collection with natural language. Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013, Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti, pages 165-168.
 5. Radu Razvan Slavescu and Adrian Coapsi. Detecting Tourist Preferences for a Recommendation System. Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, 169-172.
 6. Zaporojan Sergiu, Calmicov Igor, Plotnic Constantin, Carbone Viorel. Monitoring the fabrication process of microwires. Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013, Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti, pages 173-176.
 7. Vasile Moraru, Sergiu Zaporojan. Relaxarea semidefnita pentru problema echilibrării liniilor de asamblare. Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013, Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti, pages 177-180.
 8. Oxana Hotea and Adrian Groza. Reasoning on semantic sensor streams for smart city. In International Conference on Intelligent Information Systems, 22-25 August 2013, Chisinau, Moldova, 2013, Ed. C Gaindric and S. Cojocar, ISBN 978-9975-4237-1-7, pages 219-222.

1.4 Livrabile

Livrabile corespunzătoare acestei perioade sunt:

- (D1.1) Pagina Web - disponibilă din prima lună de derulare la
<http://cs-gw.utcluj.ro/~adrian/projects/asdec>
- (D1.2) Poster de prezentare - disponibil la
http://cs-gw.utcluj.ro/~adrian/papers/2013/poster_2013.pdf



- (D1.3) Organizare seminar: Am organizat seminarul "First Workshop on Flexible Communication Between Human and Software Agents" organizat în cadrul conferinței ROCHI 2013. Seminarul a avut comitet interantional, iar lucrările au fost indexate. Pagina seminarului: <http://cs-gw.utcluj.ro/~adrian/CFAUAS2013.html>.
- (D1.4) Lucrările seminarului disponibile la <http://cs-gw.utcluj.ro/~adrian/projects/asdec/CFAUAS2013.pdf>
- (D1.5) Unelte software dezvoltate: Sistemul *OntowebStore* disponibil la: <http://ontowebstore.apphb.com/>
- (D1.6) Raport tehnic disponibil la <http://cs-gw.utcluj.ro/~adrian/projects/asdec/raporttehnic2013.pdf>

Posibilitățile de valorificare economică includ:

- Optimizarea canalelor de distribuție
- Monitorizarea proceselor de fabricație, identificarea abaterilor și generarea de explicații accesibile agentului uman.

În derulare avem:

1. Redactare lucrare cu autori comuni pentru conferința DAS, Suceava, 2014.
2. Extindere lucrare cu autori comuni pentru revista ACAM.

Capitolul 2

Descrierea contribuțiilor științifice

Contribuțiile științifice relevante din aceste lucrări sunt:

- metodă de raționare non-monotona pe fluxuri de date continue [1].
- utilizarea raționării semantice pe fluxuri de date continue cu aplicație în suportul deciziilor medicale [1].
- suport decizional în canale de distribuție prin optimizarea liniilor de asamblare [7].
- suport decizional în cazul procesor adaptive de producție a microfiredelor cu reglaj fuzzy [6].
- suport decizional și sistem de recomandare în domeniul turismului [5].
- suport decizional orientat e-business în cazul încălcării contractelor între companii [2].
- utilizarea tehnologiei multi-agent în monitorizarea traficului în contextul rețelelor vehiculare [3].
- utilizarea limbajului natural în explicarea deciziilor recomandate de sistem [4].

2.1 Suport decizional orientat e-business în cazul încălcării contractelor între companii

Propunem un sistem de gestiune a contractelor între agenții economici. După formalizarea contractelor, acestea sunt monitorizate și la încălcarea lor se activează mecanismul de remediere a disputelor. Platforma colaborativă

dezvoltată permite generarea de explicații privind cauzele încălcării prin interogarea ontologiei pe baza căreia contractele au fost instanțiate. Serviciile de raționare ale Logicii Descriptive sunt utilizate pentru verificarea formală a consistenței clauzelor contractuale.

2.1.1 Motivație

Nu întotdeauna un contract decurge conform regulilor stabilite. Adesea, un contract se execută fără a avea informații cu privire la alte contracte ale părților implicate [17]. În practică, agenții economici se află în diverse dependențe contractuale, îndeplinirea cu succes a unui contract fiind condiționată de alte contracte. Prin analiza situației de ansamblu, obiectivul lucrării este de a identifica cauzele încălcării clauzelor și de a genera explicații părților cu privire la aceste cauze.

Sistemele multi-agent constituie instrumentația tehnică adecvată problemei adresate. Fiecare agent păstrează o bază de cunoștințe iar apoi execută acțiuni. În momentul în care se execută o acțiune, o regulă din contract este activată. În caz că regula este încălcată, va rezulta o consecință datorită căreia una din părți are de suferit. Agentul care reprezintă partea victimă are dreptul de a cere o explicație partenerului care a încălcat clauza. Explicația este generată pe baza istoricului evenimentelor petrecute de la momentul inițierii contractului.

Având propria bază de cunoștințe, cu informații la care ceilalți agenți nu au acces, agenții pot să argumenteze sau să explice [9] validitatea sau invaliditatea încălcării unei clauze contractuale. Similar, agentul acuzat aduce dovezi extrase din analiza propriei baze de cunoștințe și poate să-și construiască argumente în apărarea lui. Agentul acuzator poate la rândul lui să ofere argumente care să dovedească falsitatea argumentelor aduse anterior de agentul acuzat. Dialogul argumentativ continuă până când cei doi agenți cad de comun acord asupra evenimentelor petrecute.

Un exemplu este cazul în care o firmă care produce calculatoare comandă piese de la o altă companie. Motivul pentru care un calculator nu a putut fi construit la timp poate fi faptul că piesele necesare nu au fost livrate în timpul stabilit. Acest lucru poate fi cauzat fie de nerealizarea la timp a pieselor comandate, fie de cei responsabili cu livrarea pieselor. Urmărind firul de evenimente apărute, se pot extrage explicații complexe numai dacă părțile contractuale sunt legate printr-o rețea. De aici apare ideea modelării părților cu ajutorul agenților și urmărirea acțiunilor dintre aceștia.

2.1.2 Reprezentarea contractelor în logica descriptivă

În această secțiune se prezintă modul în care contractele au fost formalizate în sintaxa Racer [7]. Figura 2.1 arată modul de reprezentare al conceptele și relațiile de la nivelul general al ontologiei. Figura 2.2 exemplifică instanțe




```

;;concepte
(implies Contract *top*)
(implies Actor *top*)
(implies Right *top*)
(implies Obligation *top*)
(implies Timeframe *top*)
(implies SellingContract Contract)
(implies RentContract Contract)
;;relatii
(define-primitive-role has-contracting-party :domain Contract :range Actor)
(define-primitive-role has-timeframe :range Timeframe)

```

Figura 2.1: Concepte generale in ontologia contractelor.

```

;;instante de concepte
(instance AComputerManufacturer Actor)
(instance ACPUManufacturer Actor)
(instance AComputerBuyer Actor)
(instance SellingContract1 SellingContract)
(instance SellingContract2 SellingContract)
;;instante de relatii
(related SellingContract1 ACPUManufacturer has-contracting-party)
(related SellingContract1 AComputerManufacturer has-contracting-party)
(related SellingContract2 AComputerManufacturer has-contracting-party)
(related SellingContract2 AComputerBuyer has-contracting-party)

```

Figura 2.2: Instante de contracte.

la nivelul general al ontologiei. În plus, în RacerPro se pot defini reguli cu ajutorul limbajului nRQL. Metoda după care se aplică o regulă este următoarea: dacă pentru variabilele din corpul regulii, *precondiția*, există indivizi în ABox care se pot lega la variabile astfel încât condiția să fie îndeplinită, atunci aserția prezentă în capul regulii, *concluzia* este adăugată la ABox cu variabilele instanțiate cu indivizii care satisfac condițiile din corpul regulii. Pentru fiecare regulă se procedează la fel până când nu mai rămâne nimic de adăugat în ABox. Pentru sistemul automat de executare a contractelor o regulă validă este următoarea:

$$\begin{aligned}
 &(\text{define-rule } (?contract-var (= is-violated "true"))) \\
 & \quad (?contract-var ?clause-var has-violated-clause))
 \end{aligned}$$

Precondiția regulii de mai sus este $(?contract-var ?clause-var has-violated-clause)$ iar concluzia regulii de mai sus este $(?contract-var (= is-valid "false"))$, adică, în limbaj natural, *Un contract este încălcat dacă și numai dacă are o clauză încălcată*. Se observă că este vorba de o relație *dacă și numai dacă*, așadar regula definită este incompletă, reprezentând numai prima parte a teoremei. În continuare se prezintă o a doua regulă care să se asigure că teorema a fost declarată în totalitate.

$$\begin{aligned}
 &(\text{define-rule } (?contract-var ?clause-var has-violated-clause)) \\
 & \quad (\text{and } (?contract-var ?clause-var has-clause) \\
 & \quad \quad (?clause-var (= is-clause-violated "true")))
 \end{aligned}$$


Această regulă reprezintă următoarea propoziție: *Dacă o clauză face parte dintr-un contract și clauza a fost încălcată, atunci contractul se consideră a fi violat.*

Generare explicații. La acuzația unui agent că o regulă a fost încălcată, agentul victimă ar putea cere o explicație deoarece el nu consideră că ar fi violat contractul. Agentul acuzator va oferi o explicație bazată pe informațiile deținute în baza proprie de cunoștințe și anume în ABox-ul personal. Este posibil ca agentul victimă să nu dispună de informațiile necesare pentru formarea explicației pe care agentul acuzator le deține. În acest caz, pentru fiecare axiomă pe care agentul victimă nu o poate deduce, acesta poate cere o nouă explicație de la agentul acuzator:

(retrieve-with explanation (and Actor (some has-violated Contract) (some contracted X)))

În exemplul de mai sus se cere un răspuns însoțit de explicații pentru a justifica de ce există un actor care a încheiat un contract cu X și care a violat contractul.

Utilizarea de evenimente. În unele aplicații se vrea ca anumite aserții să fie valide numai într-un anumit interval de timp. În cazul contractelor se poate menționa că o parte contractantă, cumpărătorul, beneficiază de o reducere de preț sau un bonus din partea altei părți contractante, vânzătorul, pe o perioadă de cincisprezece zile dintr-un abonament pe un an în care se stipulează că timp de un an vor fi achiziționate bunuri de către cumpărător de la vânzător. RacerPro suportă inferență temporală ca și parte a sistemului nRQL. Se utilizează comanda (*define-event-assertion*) care ia ca parametri aserția ce trebuie făcută, timpii de start și de final pentru care aserția este valabilă:

*(implies Actor *top*)
(instance X Actor)
(define-event-assertion (take-bonus X) 1 15))*

Aserția subliniază faptul că un actor X poate beneficia de bonus de la timpul de start 1, prima zi, până la timpul de final 15, a cincisprezecea zi de la formarea abonamentului. Pentru aplicarea de reguli bazate pe evenimente, se utilizează comanda (*define-event-rule*). De exemplu, o regulă ar putea menționa că atunci când contractul/abonamentul dintre două părți nu a expirat încă, adică este valabil între un timp de start și unul de final, o parte contractantă poate beneficia de bonusul pentru cincisprezece zile:

*(define-event-rule ((valid-bonus ?actor ?contract) ?t1 ?t2)
((take-bonus ?actor) ?t1 ?t2)
((contract-executes ?contract) ?t1 ?t2)*

Pentru a se verifica dacă există un actor care să beneficieze de un bonus valid, se utilizează următoarea interogare:



(timener-retrieve ((valid-bonus ?actor ?contract) ?t1 ?t2)

Se returnează o listă cu perechi (*variabilă, valoare-variabilă*) pentru a se sublinia instanțele ce pot fi asociate interogării în caz că există, sau *nil* în caz ca interogarea nu găsește instanțe ce pot lua locul variabilei.

2.1.3 Gestiunea colaborativă a contractelor

Pentru stocarea și vizualizarea contractelor în limbaj natural s-a utilizat Google Drive. Google permite adăugarea de comentarii dacă se dorește purtarea unei conversații pe marginea unui document stocat pe Google Docs/ Google Drive, la care un utilizator poate lucra în prezent. Mai multe comentarii sunt grupate în discuții. Astfel, comentariile ce aparțin unei discuții anume pot fi adresate unui grup restrâns de utilizatori, pot avea o temă specifică, pot avea legătură cu un grup restrâns de documente. O discuție poate urmări comentariile adăugate prin intermediul e-mailului.

În lucrarea [17] contractele sunt păstrate pe Google Docs în format *doc*. Prin urmare, comentariile ar putea fi utilizate pentru a se purta o discuție pe marginea contractelor. Un exemplu este cazul când un agent cere explicații unui alt agent. Explicațiile ar putea fi redactate ca și comentarii. O viitoare direcție în dezvoltarea lucrării de față este traducerea comentariilor din limbajul RacerPro în limbaj natural.

Păstrarea comentariilor reprezintă o sursă pentru istoricul executării contractului. Pentru adăugarea unui comentariu este nevoie de trei etape. În prima etapă trebuie selectat textul, pagina dintr-un document sau documentul pe marginea căruia se va face comentariul. În cea de a doua etapă se alege să se insereze comentariul. În cea de a treia etapă se scrie comentariul în câmpul care apare lângă obiectul pe marginea căruia se comentează. Aceștia sunt pașii de utilizare a comentariilor utilizând interfața grafică oferită de Google. La inserarea unui comentariu, se poate decide dacă acesta să fie trimis printr-un email unei persoane, prin introducerea unei adrese de email, exemplu: *+contractAgentX@google.com*. Se poate da *reply* la un comentariu, se poate șterge sau edita un comentariu anterior, acesta fiind un caz util pentru momentul când unul din agenți, la primirea de noi informații, vrea să modifice o explicație anterioară. Dezavantajul îl reprezintă pierderea informațiilor cu privire la istoricul executării contractului. În figura 2.3 se ilustrează modul în care pot fi folosite comentariile. În partea stângă apare contractul pe marginea căruia se discută. În partea dreaptă apare spațiul în care pot fi adăugate comentarii și opțiunea de setare de notificări, în caz că, de exemplu, la adăugarea unui comentariu se vrea să se trimită un email la deținătorul documentului. Se va realiza integrarea comentariilor Google cu aplicația deja existentă descrisă în [17]. Google pune la dispoziție un API care poate fi accesat din orice aplicație pentru adăugarea de comentarii printr-un alt produs software. Utilizând Google



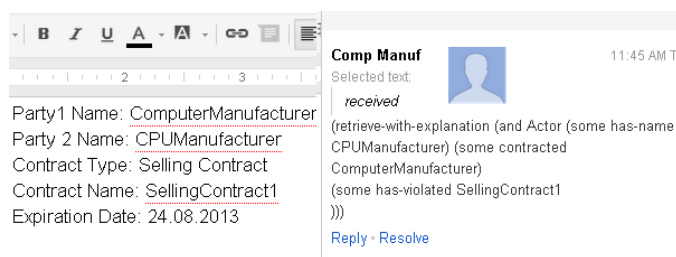


Figura 2.3: Generarea de explicatii prin rationare semantica in platforma colaborativa Google Drive.

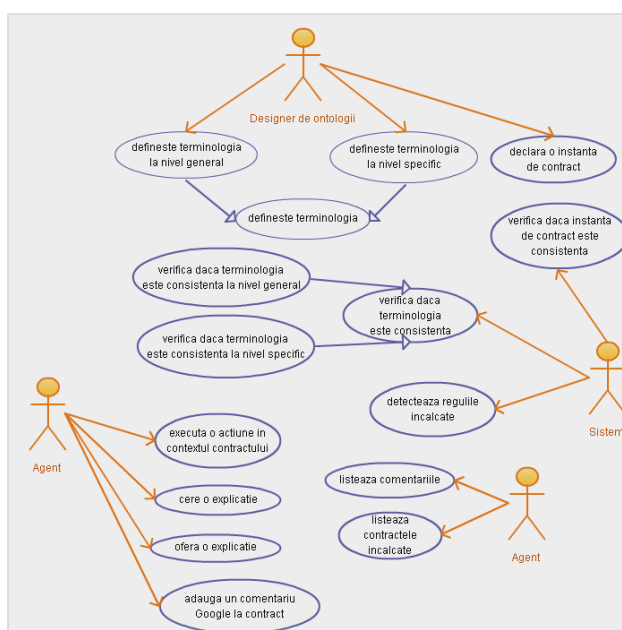


Figura 2.4: Cazurile de utilizare ale sistemului.

Drive SDK[6] pot fi accesate servicii de inserare, ștergere sau modificare a unui comentariu, conținutul cererilor fiind de tip JSON.

2.1.4 Arhitectura sistemului

În cele ce urmează se va face o descriere completă a fiecărui caz de utilizare din 2.4, detailindu-se în ce constă fiecare pas:

Definirea ontologiei de contracte generale: contractele împărtășesc noțiuni comune precum cea de parte contractantă, rol, clauză, date de contact pentru părțile contractante, momentul când a început executarea unui contract sau când acesta expiră. Toți termenii comuni pentru toate tipurile de contract trebuie definiți astfel încât aceleași reguli să poată fi aplicate indiferent



de tipul de contract.

Definirea ontologiei generice de contracte specifice: deși toate contractele au o parte comună, așa cum s-a menționat la cazul de utilizare anterior, partea specifică este de cele mai multe ori mai complexă cuprinzând detalii în funcție de tipul de contract. De obicei partea de clauze este specifică fiecărui tip de contract. Astfel, pentru un contract de achiziționare de componente pentru calculator, termenii pentru definirea entităților din contract și a relațiilor dintre ele vor fi diferiți de cei dintr-un contract de închiriere.

Verificarea dacă terminologia de la nivelulul general este consistentă: la definirea conceptelor, relațiilor sau regulilor dintr-un contract este posibil să apară inconsistențe. Astfel, unele definiții ar putea să se afle în relație de contradicție. Detectarea erorilor de genul acesta trebuie făcută automat prin verificarea consistenței taxonomie definite la primul caz de utilizare.

Verificarea dacă terminologia de la nivelulul specific este consistentă: la fel ca la cazul de utilizare anterior, și terminologia specifică tipului de contract care se execută poate suferi inconsistențe. De aceea este nevoie de un verdict care să confirme sau să infirme consistența terminologiei și eventual să returneze motivul pentru care terminologia s-a decis a fi inconsistentă.

Instanțierea unui contract: o dată ce noțiunile necesare pentru a descrie un contract au fost definite, se poate descrie un contract concret, utilizând atât terminologia de la nivelulul general cât și pe cea de la nivelulul specific. Astfel, fiecare din conceptele și relațiile definite anterior, vor fi instanțiate.

Verificarea dacă instanța de contract este consistentă: există posibilitatea să apară inconsistențe și la nivelul unei instanțe de contract. De exemplu, se poate declara că un contract expiră la o dată anterioară momentului de start a contractului. În acest caz se va semnala o eroare de inconsistență.

Executarea unei acțiuni legate de contract: Fiecare acțiune ce poate influența rezultatul final al executării contractului trebuie să poată fi executată în acest context, astfel încât rezultatul invocării ei să determine schimbări în evoluția contractului. Pentru obținerea ulterioară a unor explicații complete care să justifice anumite fapte, acțiunile trebuie păstrate într-un istoric.

Detectarea regulilor încălcate: Încălcarea unei reguli este evenimentul principal urmărit în cadrul unui contract. O dată cu violarea unei clauze, întreg contractul este nerespectat. Pentru aceasta este nevoie ca orice încălcare de clauză să fie detectată, iar conflictul generat de acest eveniment să fie rezolvat. Detectarea neregulilor se face pe baza axiomelor definite în primele două cazuri de utilizare și desigur, a valorilor concrete de la nivelul instanțelor.

Cererea de explicații pentru violarea contractului: La momentul încălcării unei reguli, agentul acuzator va cere o explicație. Agentul care a încălcat regula va trebui să formuleze o explicație care să justifice acțiunile executate sau ca să contrazică acuzația celeilalte părți cum că a încălcat o anumită regulă.



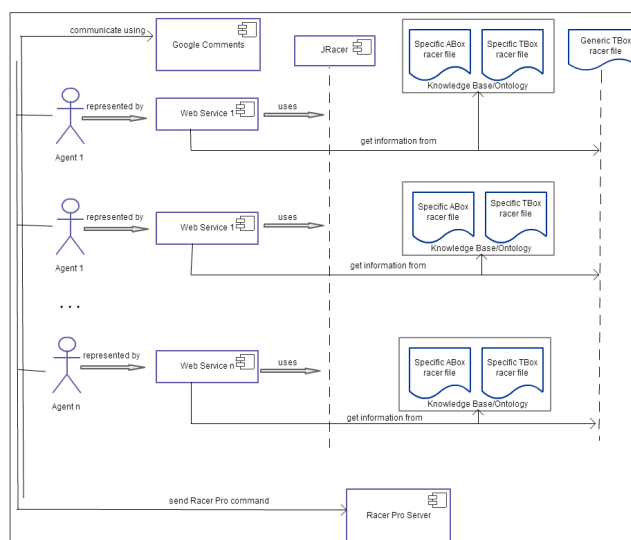


Figura 2.5: Arhitectura sistemului.

Furnizarea explicației pentru violarea contractului: generarea unei explicații pentru încălcarea unei clauze se va realiza prin consultarea propriei baze de cunoștințe și comunicarea cu agenții colaboratori.

adaugă un comentariu pe Google asociat contractului: Rezolvarea conflictelor în cazul contractelor se face de multe ori prin comunicare. De exemplu, conflictele pot apărea din cauza lipsei de informații. Astfel, o parte contractantă ar putea crede că o regulă a fost încălcată de partea cu care a încheiat contractul, acest lucru fiindu-i infirmat de partea acuzată prin aducerea de argumente sau explicații. În momentul în care se cere o explicație sau se generează una, un nou comentariu Google va fi adăugat.

listează comentariile pentru un anumit contract: Pentru înțelegerea modului în care a fost executat un contract și a rezolvării conflictelor, istoricul conversațiilor purtate în contextul contractului sunt relevante. De aici apare nevoia de a vizualiza toate comentariile adăugate pe parcursul desfășurării unui contract.

vizualizează lista cu contractele violate pentru o parte contractantă: Orice agent are nevoie de o listă cu contractele încălcate pentru a ști că în acele cazuri trebuie inițializat protocolul de rezolvare a conflictelor cu ajutorul explicațiilor. În figura 2.5 se observă că ABox-ul general și TBox-ul general este stocat la nivelul fiecărei părți contractante. La fel, pentru fiecare tip de contract în care este implicat, un agent păstrează și un TBox specific. Desigur, pentru fiecare contract încheiat va exista un ABox specific. Motivația utilizării acestei arhitecturi este executarea proceselor cât mai aproape de date pentru a reduce traficul, păstrarea confidențialității, cunoștințele deținute de o parte fiind ferite de accesul unei terțe părți. Un



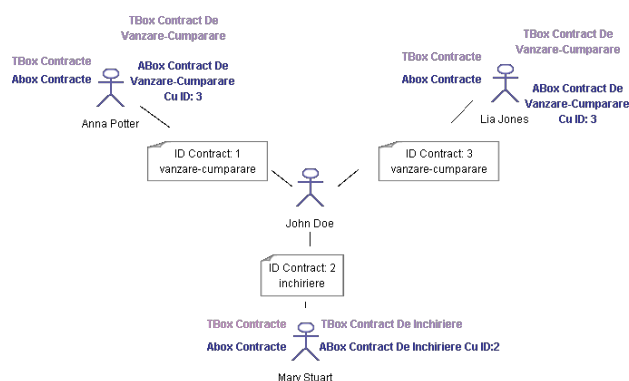


Figura 2.6: Organizarea ontologiei multi-nivel pentru cazul concret al contractelor din tabelul 2.1.

ID	TIP	PARTEA 1	PARTEA 2
1	vânzare-cumpărare	John Doe	Anna Potter
2	închiriere	John Doe	Mary Stuart
3	vânzare-cumpărare	John Doe	Lia Jones

Tabelul 2.1: Instanțe de contracte.

ultim motiv, foarte important, este evitarea *single point of failure*. Accesul la serviciile RacerPro se face prin intermediul bibliotecii Java, JRacer.

În figura 2.6 poate fi vizualizat modul în care vor fi păstrate bazele de cunoștințe în conformitate cu afirmațiile anterioare. Tabelul 2.1 exemplifică un caz de utilizare pentru contracte.

2.1.5 Contribuții

S-au implementat șabloane pentru diferite tipuri de contracte, fiind posibilă în același timp și personalizarea lor. Prin împărțirea ontologiei pe nivele s-a acordat importanță performanței sistemului, fiind suficient ca TBox-ul și ABox-ul general să fie încărcate o singură dată în RacerPro. Informațiile din contracte au fost decuplate, încurajându-se reutilizarea lor iar cele confidențiale nu au fost distribuite altor părți. Situațiile conflictuale dintre părți au fost rezolvate cu ajutorul explicațiilor. S-a realizat un mediu colaborativ pentru părțile contractante prin integrarea aplicației cu Google Drive. Cercetare în desfășurare se focalizează pe monitorizarea excepțiilor apărute în sistem, construirea unei taxonomii de excepții și luarea deciziilor în funcție de tipul excepțiilor [19]. Astfel, în anumite cazuri nu va mai fi necesară interogarea agenților, tipul de excepție fiind suficient pentru diagnosticarea unei probleme.



2.2 Utilizarea limbajului natural în explicarea deciziilor recomandate de sistem.

2.2.1 Motivație

Linked Data înseamnă utilizarea Web-ului pentru conectarea datelor care nu au fost conectate în prealabil în vederea facilitării accesului integrat și uniform la date. Unul dintre cei mai mari jucatori ai acestui domeniului încă tânăr este *DBpedia*[1] - un efort al comunității de a extrage informații structurate din Wikipedia. Domeniul guvernamental sau medical vin să completeze plaja de domenii în care Linked Data a luat un avânt deosebit. Problema interogării acestor date devine astfel complicată nu doar datorită diferitelor formalisme de reprezentare a datelor ci și datorită dimensiunii enorme atât a datelor cât și a ontologiilor de descriere.

Ne propunem să investigăm interogarea acestor date în limbaj natural, pornind de la premisa Web-ului Semantic de acceptare a limbajului RDF(Resource Description Framework) drept standard de reprezentare a datelor. În ciuda numeroaselor rezultate în domeniul prelucrării limbajului natural[10], [4], parsarea sintactică și mai ales cea semantică ridică încă o multitudine de probleme care vor limita expresivitatea sistemului nostru. Sistemul propus construiește interogări SPARQL (Protocol and RDF Query Language) din interogări în limbaj natural prin îmbinarea metodelor existente de parsare sintactică cu o formă proprie de parsare semantica. Domeniul ales în prezent pentru testare este cel al datelor medicale, și mai precis proiectul Bio2RDF.

2.2.2 Linked Data și proiectul Bio2RDF

În contextul acceptării tot mai largi a conceptului de Linked Data (Date interconectate), proiectul Bio2RDF se remarcă printre alte proiecte ce încearcă integrarea colecțiilor bioinformatică [3]. Cu Bio2RDF, documente din bazele de date bioinformatică publice, precum Kegg (Kyoto Encyclopedia of Genes and Genomes), PDB (Protein Data Bank), MGI (Mouse Genome Informatics), HGNC (HUGO Gene Nomenclature) și baze ale centrului NCBI (National Center for Biotechnology Information) sunt accesibile în formatul RDF printr-un URI (Identificator Uniform de Resursă) unic.

DrugBank este o componentă a proiectului Bio2RDF care oferă informații despre medicamente, conținând în jur de 766,920 triplete și 4,800 medicamente. În DrugBank, fiecare medicament este o resursă care are proprietăți șiruri de caractere precum *toxicitatea*, *categoria*, *indicațiile*, *mecanismul de acțiune*, *sinonim*, *descriere*, *absorbție*. Un alt tip de proprietăți sunt cele care au drept valori alte resurse în loc de șiruri de caractere, precum *ddi-interactor-in*, *patent*, *dosaj*, *target*, *tip*, *la fel ca*. Conectarea resurselor intra și inter vocabular se poate observa în exemplele din figura ??, unde o resursă



din vocabularul bio2rdf din DrugBank este în relație cu o resursă din vocabularul Pharmacogenomics Knowledge Base.

În acest exemplu, se observă triplete ce descriu resursa asociată medicamentului cu eticheta Lepirudin: organisme afectate, absorbție și ruta. Dacă valorile primelor două proprietăți sunt șiruri de caractere, în cazul ultimei proprietăți, valoarea este o altă resursă a cărei proprietate *label* descrie modul de administrare ‘Intravenous’.

2.2.3 Accesul la date În Linked Data

Dincolo de problemele de integrare, accesul la date din Linked Data este greu datorită necesității cunoașterii prealabile a ontologiilor sau a vocabularului utilizat. Ceea ce propunem în aceasta lucrare este interogarea acestor date în limbaj natural. Considerăm că o astfel de abordare este justificată în ciuda limitărilor procesării limbajului natural prin faptul că utilizatorul uman nu este străin de domeniul interogat și se poate referi la entitățile domeniului în limbaj natural. Ceea ce nu cunoaște utilizatorul uman și considerăm că nici nu ar trebui să cunoască, sunt ontologiile utilizate pentru reprezentarea domeniului. Ontologia ar trebui să fie destinată mașinii, nu omului, iar modul în care este reprezentată semantica datelor ar trebui să fie cât mai transparent utilizatorului uman.

Figura 2.7 prezintă principalele module ale sistemului. Modulul *Parsare*

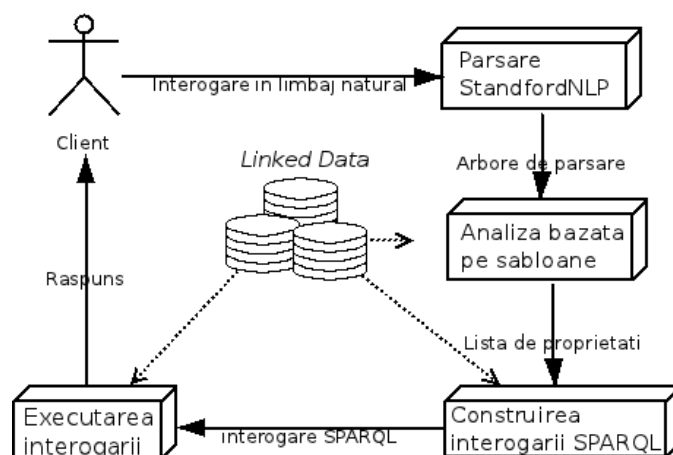


Figura 2.7: Modulele sistemului

determină arborele de parsare pentru interogarea furnizată de utilizator în limbaj natural în limba engleza. În primul rând, interogarea este preprocesată în vederea evitării dependențelor false, după care, folosindu-se API-ul Stanford Parser [12], se realizează analiza morfologică și sintactică a frazei, returnându-se arborele de parsare.

Modulul de *Analiza* identifică tipul de întrebare prin suprapunerea arborelui peste un set de șabloane predefinite. Scopul acestor șabloane este de a extrage semantica asociată frazei și de a identifica variabilele și proprietățile ce urmează a fi folosite în interogarea SPARQL. Lista de proprietăți și variabile astfel construită este preluată de către al treilea modul, cel de *Construire* a interogării SPARQL. În cele din urmă, interogarea generată este executată de către modulul de *Executare* care se conectează la serviciul Bio2RDF de tip endpoint¹ care furnizează date despre medicamente din colecția DrugBank.

2.2.4 Analiza interogării bazată pe șabloane structurale

Semantica interogării utilizatorului este extrasă pe baza unora șabloane structurale corelate cu ontologia curentă. Vom prezenta câteva dintre ele împreună cu interogarea SPARQL generată de sistem, urmând ca algoritmul de construire a interogării să îl detaliem în secțiunea următoare.

Rezultatul modulului *Analiză* este o listă de variabile și proprietăți, mai exact o **listă de tuple** de forma $\langle \text{coloana}, \text{proprietate}, \text{inSelect}, \text{deCăutat} \rangle$. Fiecărui aspect identificat în interogare îi corespunde o tuplă care în cele din urmă va deveni un pattern de triplet SPARQL. Componenta *Coloana* reprezintă numele variabilei corespunzătoare unei proprietăți cerute prin textul de intrare, *proprietate* este numele proprietății corespunzătoare în vocabularul Bio2RDF, în timp ce *deCăutat* poate avea valoare nulă sau numele unei anumite entități pe baza căreia se va face o filtrare. *inSelect* indică dacă o valoare a unei proprietăți i) corespunde unui aspect solicitat de către client, caz în care variabila dată de *coloana* este inclusă în clauza *SELECT* sau ii) este o proprietate auxiliară ce trebuie inclusă doar în clauza *WHERE*.

Reamintim că o interogare *SPARQL* constă din două părți: clauza *Select* care identifică variabilele ce vor apărea în rezultat, respectiv clauza *Where* care descrie graful șablon care restricționează datele rezultat.

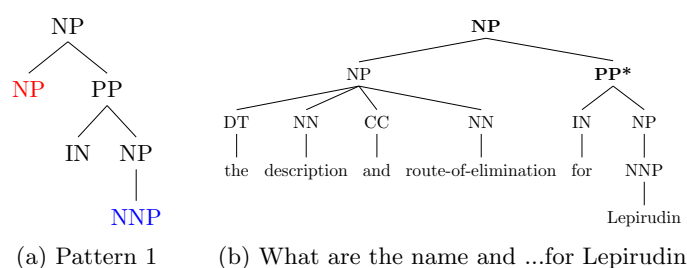


Figura 2.8: Șablonul 1 împreună cu un exemplu

¹<http://drugbank.bio2rdf.org/sparql>



Cel mai simplu șablon identificat este descris de structura din figura 2.8a. Interogările care respectă acest șablon sunt de forma **What is/are the list of properties for/of drug name ?** or **Find the list of properties for drug name**, unde proprietățile pot fi binare sau n-are cu $n \geq 3$. Exemple de întrebări: i) *What is the substructure of Lymecycline?* sau ii) *Find the composition for Lymecycline.*, iii) *Find the description and toxicity for Lepirudin*, iv) *What is the description and interactor drugs for Lepirudin?*

Numele entității căutate este extras din structura **NNP** (proper noun phrase), în timp ce proprietățile căutate sunt extrase din substantivele sau adjectivele identificate în subarboarele structurii **NP** aflat în relație de determinare cu structura prepozițională **PP**.

Pentru interogarea "What is the description and route of elimination for Lepirudin?" se obține un arbore sintactic din care componenta relevantă este cea inclusă în figura 2.8b. Pe baza acestui arbore, modulul de *Analiza* construiește următoarea listă de tuple din care ulterior modulul de *Construire* generează interogarea SPARQL.

coloana	proprietate	inSelect	deCăutat
description	$\langle \text{http://purl.org/dc/terms/description} \rangle$	true	nul
route-of-elimination	$\langle \text{http://bio2rdf.org/drugbank_vocabulary:route-of-elimination} \rangle$	true	nul
name	$\langle \text{http://www.w3.org/2000/01/rdf-schema\#label} \rangle$	false	Lepirudin

```

SELECT ?description ?routeofelimination
WHERE {
  ?a <http://www.w3.org/2000/01/rdf-schema#label> ?name.
  ?a <http://purl.org/dc/terms/description> ?description.
  ?a <http://bio2rdf.org/drugbank_vocabulary:route-of-elimination>
    ?routeofelimination
  FILTER REGEX( str(?name), "Lepirudin", "i")
}

```

Al doilea șablon structural (figura 2.9) permite ca proprietățile solicitate să nu fie ale unui anumit medicament ci ale unei clase de medicamente. Această clasă poate fi determinată de

- 2.1 apartenența sau lipsa apartenenței la o categorie, respectiv prezența interacțiunii: *drugs that are (not) în Anticoagulant category*
- 2.2 sinonimia cu alt medicament: *drugs that are synonymous with Lepirudin*
- 2.3 lipsa interacțiunii cu alt medicament: *(do not) interact with Lepirudin*
- 2.4 combinație ale primelor trei criterii: *drugs that interact with Ginkgo Biloba, are în anticoagulants category, do not interact with Lepirudin, and are synonymous with Hirudin?*



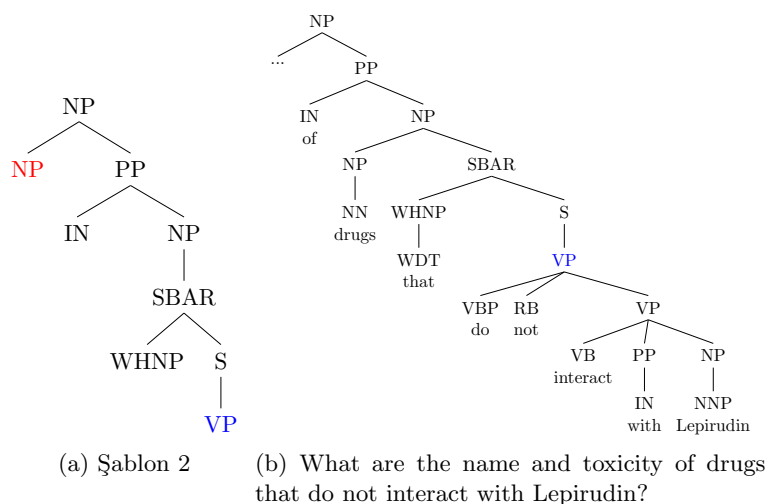


Figura 2.9: Șablonul 2 împreună cu un exemplu

Șablonul 2 identifică structuri de genul **What are/is the list of properties for drugs that** una dintre cele patru clase mai sus menționate. Similar șablonului 1, proprietățile solicitate sunt identificabile pe baza substantivelor sau a adjectivelor din structura **NP**. Clasa medicamentelor este furnizată de subarboarele structurii verbale **VP** pentru care în figura 2.10 se detaliaza câteva dintre tiparele posibile. Menționăm că arborii includ cuvinte de instanțiere doar pentru o mai bună înțelegere, șabloanele conținând doar structuri sintactice sau morfologice, nu și cuvinte concrete.

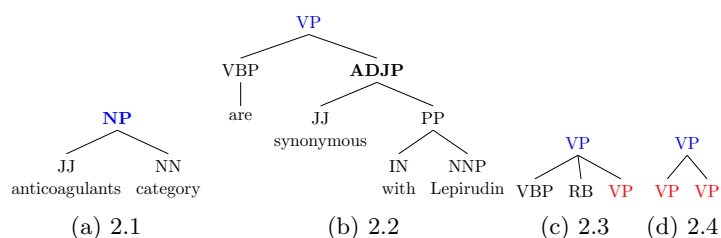


Figura 2.10: Clase de entitati - Sub-șabloane ale șablonului 2

Din motive de spațiu nu vom prezenta celelalte șabloane utilizate în sistem, dar menționăm câteva tipuri de întrebări acoperite de acestea:

- What type of drug is Lepirudin?
- What are the drugs from analgesics category ?
- Is there an interaction between Lepirudin, Thrombin, Hirudin?



- Is it safe/unsafe/dangerous to combine Lepirudin with Thrombin?

2.2.5 Construirea interogării SPARQL

Odata extrase variabilele și relațiile dintre ele, modulul de *Construire* generează interogarea SPARQL. [1] $L \leftarrow analyse(arboareledeparsare)$ $select \leftarrow "SELECT"$

$where \leftarrow "WHERE\{ "$ el în L if $el.inSelectcolumn \leftarrow el.getColumn()$ if $column$ contains $"interact"$ $column \leftarrow "nameb"$ $select+ \leftarrow "?" + column + " "$ Construirea clauzei Select

testcont

[H] [1] $testcont$ el în L $column \leftarrow el.getColumn()$ $select+ \leftarrow "?a" + el.getProperty +$
 $"?" + column + "."$ if $column$ contains $"interact"$ $select + \leftarrow "?b" + el.getProperty +$
 $"?" + column + "."$ $select+ \leftarrow "?b rdfs:label ?"+nameb+"." filter (?name != ?nameb) ."$
 if $el.searchingForwhere+ \leftarrow "FILTER REGEX(str("+ "?" + column+"), "+el.getSearchingFor+$
 $","i)"$ REGEX for drug name $where+ \leftarrow "}"$ construirea clauzei Where

Pentru construirea clauzei *select* se parcurge lista de tuple L și pentru fiecare tuplă cu componenta *inSelect* setată la valoarea *true* se adaugă în clauză o variabilă cu numele dat de componenta *coloana* a tuplei. În cazul clauzei *Where*, pentru fiecare tuplă se adaugă câte un triplet format dintr-o variabilă $?a$, proprietatea *property* și variabila din *coloana*. Dacă componenta *deCautat* a tuplei este *true*, se va adăuga o filtrare după proprietatea din tuplă.

Se poate remarca tratarea specială a proprietății *interact*, motivul fiind faptul că această proprietate nu există între două medicamente, ci între fiecare dintre cele două și o instanță a conceptului *Drug-Drug-Interaction* (figura ?? - interacțiunea dintre Lepirudin și Ginkgo Biloba). Pentru o întrebare de forma "What is the description and interactor drugs for Lepirudin?", rezultatul va fi

```
SELECT ?description ?nameb
WHERE {
  ?a <http://purl.org/dc/terms/description> ?description .
  ?a <http://bio2rdf.org/drugbank_vocabulary:ddi-interactor-in>
      ?interactor .
  ?b <http://bio2rdf.org/drugbank_vocabulary:ddi-interactor-in>
      ?interactor .
  ?b <http://www.w3.org/2000/01/rdf-schema#label> ?nameb .
  FILTER( ?name != ?nameb) .
  ?a <http://www.w3.org/2000/01/rdf-schema#label> ?name .
  FILTER REGEX( str(?name), "Lepirudin", "i") .
}
```

2.2.6 Contribuții

Potențialul uriaș al colecțiilor din Linked Data a fost demonstrat și de câștigarea premiului al treilea în cadrul competiției Semantic Web Challenge 2012 de către aplicația Open Self Medication[5]. În ce privește construirea interogărilor, menționăm metoda descrisă în [21]. Este vorba de



o metodă incrementală în care se pornește de la cuvinte cheie urmând ca apoi, pe baza conceptelor din ontologia domeniului, utilizatorul să poate opta pentru anumite extinderi. În cazul nostru, ontologia nu e utilizată la nivelul utilizatorului, ci facilitează maparea pe un domeniu particular a unor metode generale, adică a șabloanelor structurale propuse.

QUEPY² este o aplicație în dezvoltare care încearcă interogarea datelor Web-ului Semantic în limbaj natural. Din ce cunoaștem noi, la momentul actual permite doar interogări simple, fără posibilitatea de a descrie clase de entități precum cele permise de Șablonul 2 al sistemului nostru. Direcția de cercetare care a condus însă la abordarea propusă aici este cea a limbajelor de reprezentare a înțelesului (Meaning Representation Language) urmată în [4] sau [18]. În aceste două lucrări nu este vorba despre interogarea datelor RDF ci se prezintă metode de învățare a parsării semantice sau a translatării către un MRL. Considerăm că în absența învățării, nu se poate ajunge la un sistem de parsare semantică de un nivel acceptabil, aceasta fiind una dintre direcțiile următoare de lucru.

În concluzie, s-au prezentat primele rezultate obținute în direcția interogării colecțiilor LinkedData în limbaj natural. Sistemul nostru are drept input un text în limbaj natural din care, pe baza analizei sintactice, a unor șabloane structurale și a vocabularului de descriere a datelor, se obține interogarea SPARQL. Șabloanele structurale facilitează extragerea semanticii interogării, de aceea considerăm că reprezintă cea mai importantă componentă a sistemului nostru. În mod cert, setul de șabloane trebuie extins, rafinat și îmbogățit cu metode de compunere și de adaptare dinamică la ontologii.

²<http://quepy.machinalis.com/>



Bibliografie

- [1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *ISWC/ASWC*, pages 722–735, 2007.
- [2] Radu Balaj and Adrian Groza. Detecting influenza epidemics based on real-time semantic analysis of twitter streams. In *Modelling and Development of Intelligent Systems, 10-12 October 2013, Sibiu, Romania.*, page to appear, 2013.
- [3] François Belleau, Marc-Alexandre Nolin, Nicole Tourigny, Philippe Rigault, and Jean Morissette. Bio2rdf: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics*, 41(5):706–716, 2008.
- [4] David L. Chen and Raymond J. Mooney. Learning to interpret natural language navigation instructions from observations. In *the Twenty-Fifth AAAI Conference on Artificial Intelligence, USA*, 2011.
- [5] Olivier Curé. Open Self Medication on LOD. In *Proceedings of the Semantic Web Challenge co-located with ISWC2012*, Boston, US, November 2012.
- [6] Google Developers. Google drive sdk. <https://developers.google.com/drive/v2/reference/>.
- [7] Volker Haarslev, Kay Hidde, Ralf Möller, and Michael Wessel. The RacerPro knowledge representation and reasoning system. *Semantic Web*, 3(3):267–277, 2012.
- [8] Oxana Hotea and Adrian Groza. Reasoning on semantic sensor streams for smart city. In *International Conference on Intelligent Information Systems, 22-25 August 2013, Chisinau, Moldova, 2013, Ed. C Gaindric and S. Cojocaru, ISBN 978-9975-4237-1-7*, pages 219–222, 2013.
- [9] Ioan ALfred Letia and Adrian Groza. Interleaved argumentation and explanation in dialogue. In *The 12th Workshop on Computational Models of Natural Argument CMNA@ECAI, Montpellier, France*, pages 44–52, 2012.

- [10] Christopher D. Manning. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing - 12th International Conference, Japan*, pages 171–189, 2011.
- [11] Anca Marginean and Oana Marc. Querying linkdata collection with natural language. In *Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013, Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti.*, pages 165–168, 2013.
- [12] Bill MacCartney Marie-Catherine de Marneffe and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. 2006.
- [13] Adrian Popescu and Anca Marginean. Multi-agent system for traffic monitoring. In *Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013, Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti.*, pages 181–184, 2013.
- [14] Radu Razvan Slavescu and Adrian Coapsi. Detecting tourist preferences for a recommendation system. In *Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013, Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti.*, pages 169–172, 2013.
- [15] Sergiu Zaporojan Vasile Moraru. Relaxarea semidefinita pentru problema echilibrarii liniilor de asamblare. In *Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013, Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti.*, pages 177–180, 2013.
- [16] Gabriela Visinari and Adrian Groza. Semantic-based monitoring of e-contracts. In *Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013, Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti.*, pages 161–164, 2013.
- [17] Gabriela Visinari and Adrian Groza. Builing an e-contracts management tool using google docs. In *12th IEEE International Symposium on Computational, Intelligence and Informatics*. IEEE, 21-22 November 2012.
- [18] Yuk Wah Wong and Raymond J. Mooney. Learning for semantic parsing with statistical machine translation. In *HLT-NAACL*, 2006.



- [19] Dongming Xu, Chinthake Wijesooriya, Yonggui Wang, and Ghasan Beydoun. Outbound logistics exception monitoring: A multi-perspective ontologies approach with intelligent agents. *Expert Syst. Appl.*, 38(11):13604–13611, 2011.
- [20] Plotnic Constantin Carbune Viorel. Zaporojan Sergiu, Calmicov Igor. Monitoring the fabrication process of microwires. In *Proceedings of the 10th National Conference on Human Computer Interaction, Cluj-Napoca, Romania, 2-3 September 2013*, Ed. Teodor Stefanut and Cristian Rusu, ISSN 2344-1690, Matrix, Bucuresti, pages 173–176, 2013.
- [21] Gideon Zenz, Xuan Zhou, Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. From keywords to semantic queries - incremental query construction on the semantic web. *Journal Web Semantic*, 7(3):166–176, 2009.

