

# Sistem semantic pentru monitorizarea contractelor electronice

**Gabriela Vișinari**

Universitatea Tehnică din Cluj-Napoca  
Departamentul de Calculatoare  
Gabriela.Visinari@cs.utcluj.ro  
Cluj-Napoca, Romania

**Adrian Groza**

Universitatea Tehnică din Cluj-Napoca  
Departamentul de Calculatoare  
Adrian.Groza@cs.utcluj.ro  
Cluj-Napoca, Romania

## REZUMAT

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.

## INTRODUCERE

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 [4]. 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 [3] 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

```
;;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 1. Concepte generale în 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. Instante de contracte.

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 acestea.

## REPREZENTAREA CONTRACTELOR ÎN LOGICA DESCRIPTIVĂ

În această secțiune se prezintă modul în care contractele au fost formalizate în sintaxa Racer [2]. Figura 1 arată modul de reprezentare al conceptelor și relațiilor de la nivelul general al ontologiei. Figura 2 exemplifică instanțe 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 preconditionia 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:

(define-rule (?contract-var (= is-violated "true"))  
(?contract-var ?clause-var has-violated-clause))

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.

(define-rule (?contract-var ?clause-var  
has-violated-clause))  
(and (?contract-var ?clause-var has-clause)  
(?clause-var (= is-clause-violated "true")))

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 cincisprezece 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)

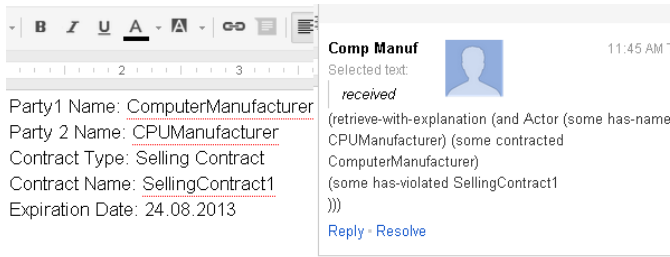
Se retunează o listă cu perechi (variabilă, valoare-variabilă) pentru a se sublinia instanțele ce pot fi asociate interogării în caz că există, sau nil în caz ca interogarea nu găsește instanțe ce pot lua locul variabilei.

## GESTIUNEA COLABORATIVĂ A CONTRACTELOR

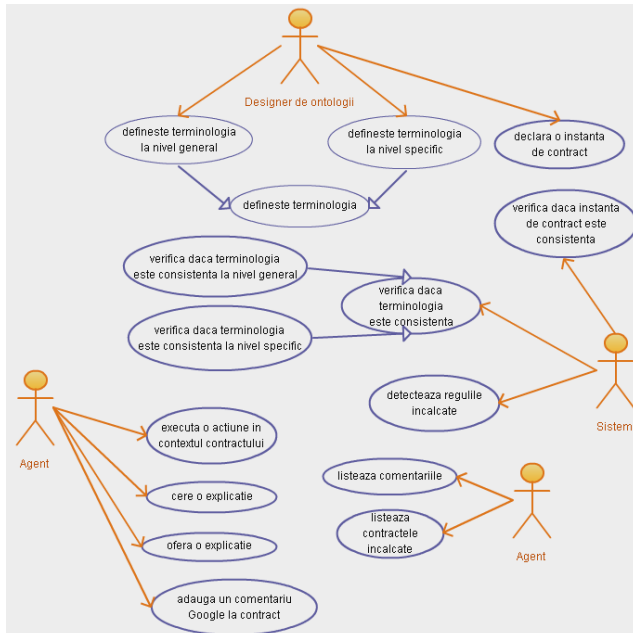
Pentru stocarea și vizualizarea contractelor în limbaj natural s-a utilizat Google Drive. Google permite adăugarea de comentarii dacă se dorește purtarea unei conversații pe marginea unui document stocat pe Google Docs/ Google Drive, la care un utilizator poate lucra în prezent. Mai multe comentarii 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 [4] 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 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 ex-



**Figura 3. Generarea de explicatii prin rationare semantica in platforma colaborativa Google Drive.**



**Figura 4. Cazurile de utilizare ale sistemului.**

istentă descrisă în [4]. Google pune la dispoziție un API care poate fi accesat din orice aplicație pentru adăugarea de comentarii printr-un alt produs software. Utilizând Google Drive SDK[1] pot fi accesate servicii de inserare, ștergere sau modificare a unui comentariu, conținutul cererilor fiind de tip JSON.

## ARHITECTURA SISTEMULUI

În cele ce urmează se va face o descriere completă a fiecărui caz de utilizare din 4, detaliindu-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 nivelul 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 taxonomiei definite la primul caz de utilizare.

*Verificarea dacă terminologia de la nivelul 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 nivelul general cât și pe cea de la nivelul 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 axiomei 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.

*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

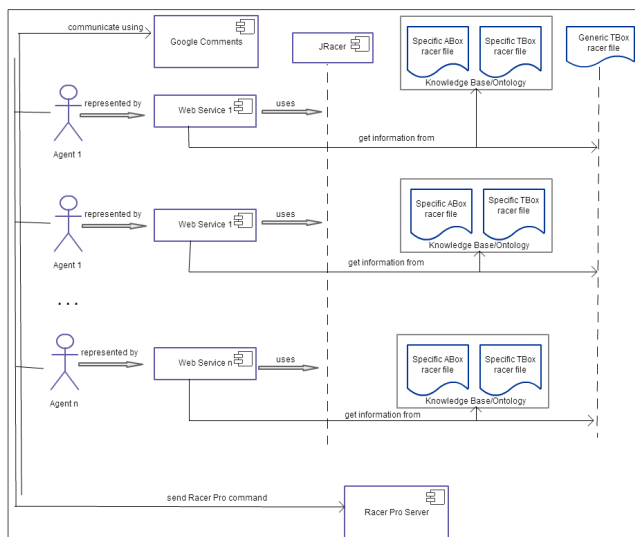


Figura 5. Arhitectura sistemului.

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

Table 1. Instanțe de contracte.

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 știi că în acele cazuri trebuie inițializat protocolul de rezolvare a conflictelor cu ajutorul explicațiilor. În figura 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 ultim motiv, foarte important, este evitarea *single point of failure*. Accesul la serviciile RacerPro se face prin intermediul bibliotecii Java, JRacer.

În figura 6 poate fi vizualizat modul în care vor fi păstrate bazele de cunoștințe în conformitate cu afirmațiile anterioare. Tabelul 1 exemplifică un caz de utilizare pentru contracte.

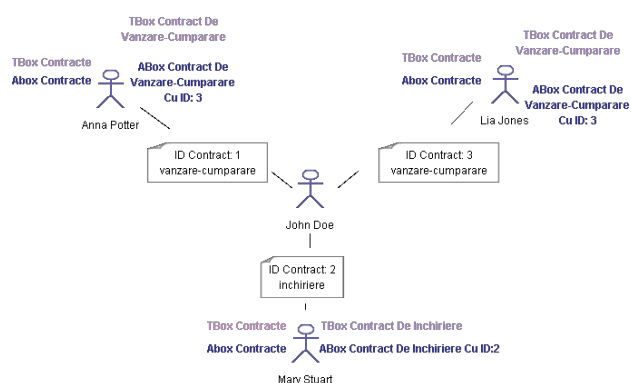


Figura 6. Organizarea ontologiei multi-nivel pentru cazul concret al contractelor din tabelul 1.

## MENTIUNI

Parte din cercetare s-a derulat în cadrul proiectului "Argumentare structurată pentru suport decizional cu constrangeri normative", programul PN-II Cooperari Bilaterale Romania-Republica Moldova, 2013-2014, UEFSCDI.

## CONCLUZII

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 pri. 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 [5]. Astfel, în anumite cazuri nu va mai fi necesară interogarea agenților, tipul de excepție fiind suficient pentru diagnosticarea unei probleme.

## REFERINȚE

1. Developers, G. Google drive sdk. <https://developers.google.com/drive/v2/reference/>.
2. Haarslev, V., Hidde, K., Möller, R., and Wessel, M. The RacerPro knowledge representation and reasoning system. *Semantic Web* 3, 3 (2012), 267–277.
3. Letia, I. A., and Groza, A. Interleaved argumentation and explanation in dialogue. In *The 12th Workshop on Computational Models of Natural Argument CMNA@ECAI, Montpellier, France* (2012), 44–52.
4. Visinari, G., and Groza, A. Building an e-contracts management tool using google docs. In *12th IEEE International Symposium on Computational, Intelligence and Informatics*, IEEE (21-22 November 2012).
5. Xu, D., Wijesooriya, C., Wang, Y., and Beydoun, G. Outbound logistics exception monitoring: A multi-perspective ontologies approach with intelligent agents. *Expert Syst. Appl.* 38, 11 (2011), 13604–13611.