

Strategie de negociere pentru tranzactii on-line

Ec. Ciprian TALABA
 Universitatea "Dunarea de Jos" Galati

Price negotiation is a new technique for electronic commerce used for increasing the number and the satisfaction of the customers. A fixed algorithm for negotiating would not accomplish this task. In this article will be presented a multiagent based negotiating algorithm. It uses an architecture with three agents: a pattern generalization agent, a pattern matching agent and a dynamic price issuing agent. For this we have used J2EE which is a set of specifications for developing distributed systems.

Keywords: negotiation, multiagent systems, data -mining, J2EE.

Introducere

IO forma frecventa de interactiune între agenti cu obiective diferite este negocierea. Negocierea este procesul prin care doi sau mai multi agenti ajung la o decizie comuna, fiecare încercând sa își atinga propriile obiective [4]. Agentii își comunica pozitiile, care pot fi conflictuale, si încearca sa ajunga progresiv la o înțelegere prin concesii sau prin cautare de alternative. Obiectivul negocierii îl reprezinta identificarea situatiilor în care pot apare interactiuni. Daca acestea sunt conflictuale este necesara modificarea planurilor locale ale agentilor. Elementele principale ale negocierii sunt: limbajul utilizat de catre agentii participanti, protocolul utilizat pentru a negocia si procesul de decizie utilizat de catre fiecare agent în scopul determinarii pozitiei sale si a concesiilor. Au fost dezvoltate o serie de algoritmi de negociere dintre care mentionam: Contract Net [3] (utilizat în special pentru alocarea task-urilor, dar care cu unele modificari poate fi utilizat si pentru piete electronice), modelul de negociere BAZAR [5], modelul de negociere al bunurilor si serviciilor [1]. În cazul tranzactiilor on-line

negocierea reprezinta de fapt schimbul valorilor de preturi între vânzator si cumparator, preturile formând serii care trebuie sa converge la un acelasi pret final.

Arhitectura sistemului SNegTO

Un proces de negociere consta în schimbul unor serii de preturi între vânzator si cumparator. O tranzactie este efectuata când ambele parti sunt de acord asupra valorii unui pret. Seria de preturi ale vânzatorului este notata $S = \{S_0, S_1, S_2, \dots, S_n\}$, iar cea a cumparatorului este notata $B = \{B_0, B_1, B_2, \dots, B_n\}$, unde S_i si B_i reprezinta preturile emise la pasul i de catre vânzator respectiv cumparator. În figura 1 este prezentat un exemplu de negociere, incluzând câștigul si prima care sunt detaliate în sectiunea de fata. Un sablon de negociere generalizeaza ambele serii de preturi (S si B). Lista pantelor preturilor de vânzare este notata cu $S_0S = \{K_0, K_1, K_2, \dots, K_n\}$, unde $K_0 = (B_0 - S_0)/S_0$, $K_1 = (S_0 - S_1)/S_0$, $K_2 = (S_1 - S_2)/S_0 \dots$ si $K_n = (S_{n-1} - S_n)/S_0$. Lista pantelor preturilor de cumparare este notata cu $S_0B = \{A_0, A_1, A_2, \dots, A_n\}$, unde $A_0 = (B_0 - S_0)/S_0$, $A_1 = (B_0 - B_1)/S_0$, $A_2 = (B_1 - B_2)/S_0 \dots$ si $A_n = (B_{n-1} - B_n)/S_0$.

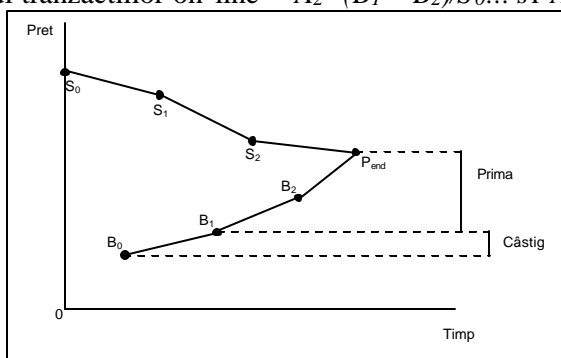


Fig.1. Exemplu de schimburi de preturi într-un proces de negociere

În figura [2] este prezentata structura conceptuala a unui sistem multiagent care suporta procesul dinamic de negociere online [2]. Acest model a stat la baza dezvoltarii sistemului SNegTO, care functioneaza la nivelul vânzatorului si include :

- 1.un agent care utilizeaza tehnica data-mining pentru a generaliza sabloane de negociere;
- 2.un agent care verifica similaritatea dintre tranzactia curenta si un anumit sablon;
- 3.un agent care genereaza dinamic pretul pe baza teoriei utilitatii.

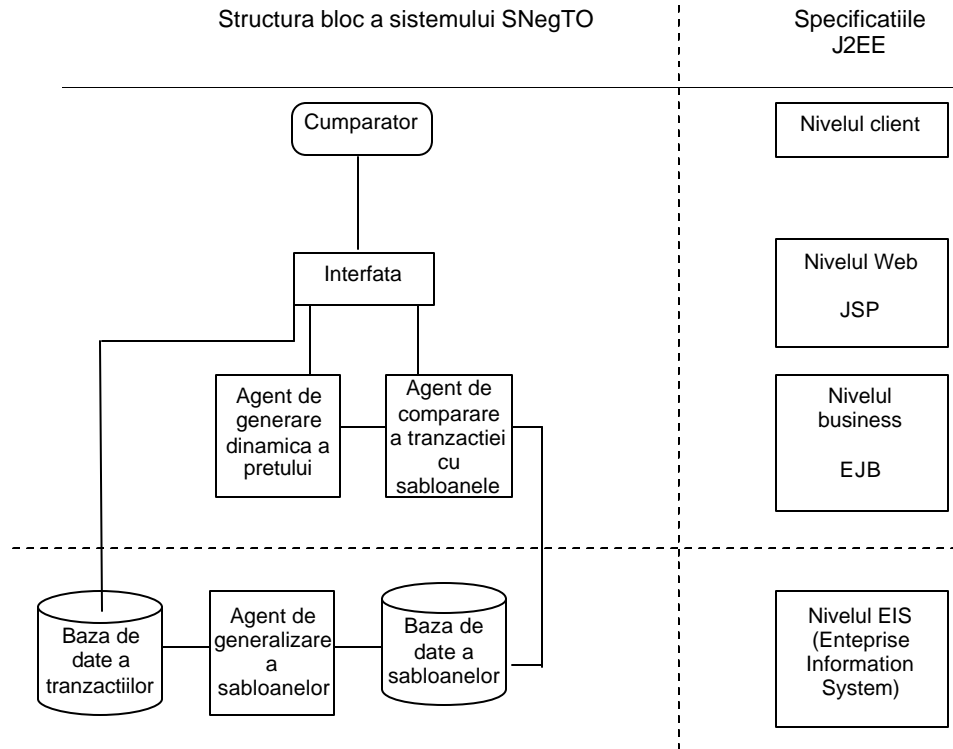


Fig.2. Arhitectura sistemului SNegTO în comparație cu specificațiile J2EE

Ordinea de execuție a agenților în cadrul sistemului SNegTO este reprezentată în figura 3.

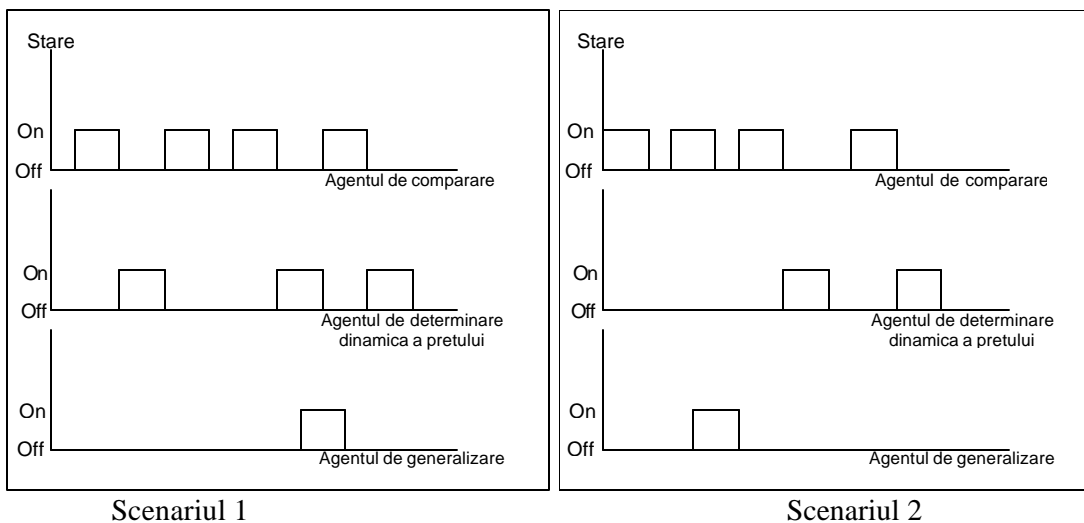


Fig.3. Ordinea de execuție a agenților componenti

Se observa ca agentul de determinare dinamica a pretului se executa în anumite cazuri (când agentul de comparare nu gaseste un sablon pentru tranzactia curenta), dupa executia agentului de comparare care îl apeleaza. Agentul de generalizare nu are restrictii din punct de vedere al activarii sale. El opereaza la anumite momente de timp bine stabilite de catre administratorul sistemului SNegTO. Aceste momente sunt determinate de numarul de tranzactii efectuate într-o perioada de timp (de exemplu o zi).

Modul de functionare al sistemului SNegTO se poate împarti în doua faze: învățare si executie.

În faza de învățare agentul de generalizare va genera sabloane de negociere reprezentate de pantele schimburilor de preturi între vânzator si cumparator. Generalizarea sabloanelor se realizeaza în urmatoarele patru etape:

1. se fragmenteaza seriile de preturi. Un fragment al listei de pante a preturilor de cumparare este definit ca un segment SoB cu trei preturi alaturate. Se calculeaza distanta dintre doua pante ale cumparatorului notata cu e si apoi distanta dintre doua fragmente SoB, $E_n = e_{n-1} + e_n + e_{n+1}$. s este o limita pentru E , si daca E este mai mare decât s cele doua serii de pante sunt considerate diferite. De exemplu daca $SoB_1 = \{A_{10}, A_{11}, A_{12}, \dots, A_{1n}\}$ si $SoB_2 = \{A_{20}, A_{21}, A_{22}, \dots, A_{2n}\}$ doua serii de pante, putem calcula $E_1 = [(A_{10} - A_{20}) + (A_{11} - A_{21}) + (A_{12} - A_{22})]$, $E_2 = [(A_{11} - A_{21}) + (A_{12} - A_{22}) + (A_{13} - A_{23})]$, si $E_{n-1} = [(A_{1,n-2} - A_{2,n-2}) + (A_{1,n-1} - A_{2,n-1}) + (A_{1,n} - A_{2,n})]$. Daca oricare E_i este mai mare decât s atunci cele doua serii de preturi sunt considerate diferite. Acelasi proces se aplica si în cazul SoS.

2. se compara doua stari din procesul de negociere. Daca nici un E_i nu depaseste limita s , se poate considera ca cele doua serii sunt identice. Mai este necesar sa comparam si seriile de preturi ale vânzatorului pentru a putea generaliza un sablon. Daca nici în acest caz nu este depasita limita s se poate generaliza un sablon.

3. generalizarea efectiva a sablonului: acest lucru se realizeaza prin înlocuirea listelor de

pante luate în discutie cu valorile medii ale lor. Astfel daca vom nota cu $Z = \{Z_0, Z_1, Z_2, Z_3\}$ noul sablon vom avea: $Z_0 = (A_{10} + A_{20})/2$, $Z_1 = (A_{11} + A_{21})/2$, $Z_2 = (A_{12} + A_{22})/2$ si $Z_3 = (A_{13} + A_{23})/2$. Pentru lista de pante a vânzatorului se aplica acelasi algoritim. Ambele liste de pante formeaza sablonul de negociere si sunt memorate în baza de date a sabloanelor.

4. se reiau etapele 2 si 3 pentru a verifica daca se mai pot generaliza si alte sabloane. Algoritmul se opreste atunci când nu mai este posibila generalizarea de noi sabloane.

Procesul de negociere începe când cumparatorii trimit preturi catre sistem prin intermediul unei interfete. Acest lucru initializeaza doua procese. Agentul de comparare verifica sabloanele din baza de date pentru a determina pretul care trebuie acceptat. Daca nici un sablon nu corespunde tranzactiei curente sistemul apeleaza agentul de determinare dinamica a pretului.

1. În cazul în care cumparatorul nu accepta pretul propus si face o contra-oferta, agentul de comparare a sabloanelor va returna pretul în conformitate cu urmatoarele trei etape:

2. compararea sabloanelor si selectia – când un cumparator începe procesul de negociere el citește pretul de pornire S_0 oferit de sistem. Când cumparatorul returneaza oferta sa B_0 agentul poate calcula $A_0 = (B_0 - S_0)/S_0$. Fie s limita lui E . Agentul calculeaza E_s bazându-se pe sabloanele memorate în baza de date si va selecta acele sabloane în care E_s nu depaseste limita s . Daca cel puțin un sablon îndeplineste conditia agentul selecteaza aleator unul dintre ele, salvându-le pe celelalte ca eventuale alternative.

3. calcularea pretului – agentul calculeaza pretul pe baza seriilor de preturi care formeaza sablonul selectat

4. repetarea etapelor 1 si 2 – dupa ce cumparatorul ofera un pret agentul de comparare poate obtine lista de preturi a acestuia (exemplu A_1 si A_2). Se calculeaza valoarea E (de exemplu $e_1 + e_2$) si daca aceasta nu depaseste limita s agentul trece la etapa 2. În cazul în care limita este depasita se trece la etapa 1 încercându-se gasirea unor noi sabloane care

sa îndeplineasca conditiile apoi trece la etapa 2. Daca nu se gaseste nici un sablon cores-punzator sistemul apeleaza agentul de deter-minare dinamica a pretului.

Procesul de determinare dinamica a pretului include doua etape:

1. stabilirea pretului dorit – în acest caz se uti-lizeaza datele tranzactiilor deja efectuate, selectându-se maximul preturilor obtinute pentru acel produs, sau în cazul în care nu exista tranzactii se utilizeaza metoda costului si a profitului dorit. De exemplu daca C este costul produsului si e este profitul asteptat atunci se poate stabili pretul dorit ca suma dintre C si e . Sistemul va accepta oferta daca ea este între costul produsului si pretul dorit, sau mai mare decât pretul dorit. În cadrul procesului de negociere sistemul determina daca oferta este mai mare decât $C+e$. Daca da, atunci tranzactia este efectuata. Da nu, sistemul ofera un nou pret astfel încât oferta sa se stabilizeze între C si $C+e$, preferabil cât mai aproape de $C+e$ posibil.

2. evaluarea ofertei - agentul ajunge la o deci-zie asupra pretului prin utilizarea câstigului si primei. În figura 1 este prezentat acest proces. Când cumparatorul ofera pretul B_1 , a-gentul de determinare dinamica a pretului fie accepta aceasta oferta fie, ofera un nou pret. În aceasta situatie vânzatorul câstiga $U_s(B_1) - U_s(B_0)$ si pierde prima $U_s(P_{end}) - U_s(B_1)$, unde U_s este functia de utilitate a vânzatorului. Pretul oferit de cumparator este acceptat daca câstigul este cel puțin egal cu prima si pretul este localizat în limita acceptata.

Aspecte ale implementarii strategiei de ne-gociere

Sistemului SNegTO a fost implementat sub forma unui site de comert electronic care în-globeaza strategia de negociere a preturilor. În dezvoltarea acestuia a fost utilizata plat-forma Java 2 Enterprise Edition (J2EE), care reprezinta un set de specificatii pentru dez-voltarea de aplicatii distribuite. Este aplicabi-la în acest caz deoarece o aplicatie de comert electronic este în mod esential distribuita.

Componentele J2EE care pot fi utilizate pen-tru dezvoltarea de aplicatii Web sunt grupate

în patru categorii: Java applets; Java appli-cations; Java servlets, Java Server Pages (JSP); Enterprise Java Beans (EJB).

În cazul acestei aplicatii am optat pentru uti-lizarea componentelor JSP – pentru crearea interfeței Web utilizator – precum si a Enter-prise Java Beans (EJB) pentru construirea agentilor. Comunicarea dintre agenti (obiecte Java) este realizata în cazul platformei J2EE prin doua metode:

Remote Method Invocation (RMI) – utilizata pentru comunicarea sincrona (se asteapta raspunsul) si reprezinta de fapt apelarea unor metode publice ale obiectului cu care se do-reste a se comunica, metode ce vor returna un raspuns

Java Messaging Service (JMS) – utilizata pentru comunicarea asincrona.

Datorita faptului ca realizarea comunicarii în-tre agenti trebuie sa se realizeze într-un timp cât mai scurt pentru a se raspunde în timp util cererii clientului, solutia care a fost aleasa pentru comunicare este RMI.

În urma simularilor a rezultat faptul ca exista situatii pentru care modelul de negociere nu reuseste sa ajunga la o solutie acceptata atât de vânzator cât si de cumparator. Prima si-tuatie care poate apare este atunci când cum-paratorul porreste de la un pret foarte mic (uneori chiar 0) ceea ce determina scaderea pretului cerut de vânzator sub costul produ-sului. În acest caz negocierea este oprita de-oarece conditia este inacceptabila din punctul de vedere al vânzatorului (figura 4). Solutia în acest caz este impunerea unui pret minim oferit de cumparator pentru ca vânzatorul sa scada pretul cerut.

A doua situatie care poate duce la insuccesul strategiei de negociere este aceea în care cumparatorul scade pretul oferit la pasul an-terior (figura 5). În acest caz, conform mode-lului initial, vânzatorul va pastra acelasi pret deoarece nu este luata în discutie posibilita-tea ca utilitatea câstigata între doi pasi conse-cutivi sa fie negativa.

În acest caz pot fi identificate doua rezolvari: prima presupune interzicerea scaderii pretu-lui de catre cumparator, iar cea de-a doua presupune penalizarea cumparatorului care

adopta aceasta strategie prin cresterea pretului cerut cu un procent mai mare. În cazul de

fata am adoptat a doua solutie procentul de crestere fiind considerat 150%.

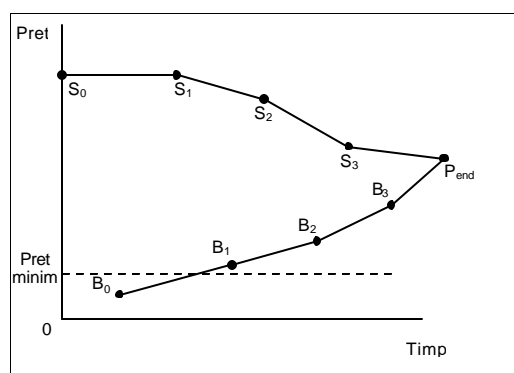
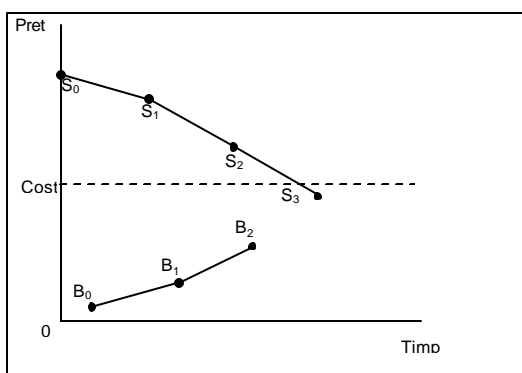


Fig.4. Representarea conflictului în cazul pretului initial foarte mic

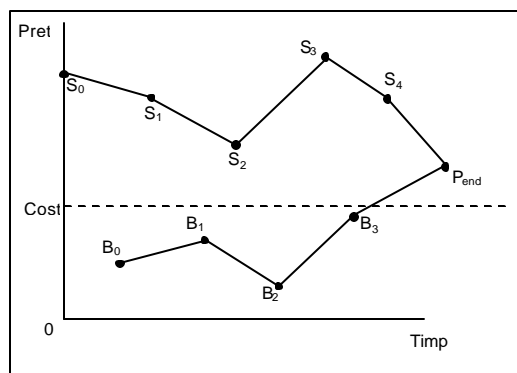
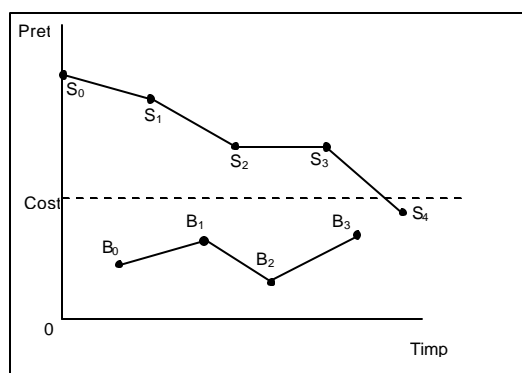


Fig.5. Representarea conflictului aparut în cazul scaderii pretului oferit

Concluzii

Negocierea preturilor reprezinta modalitatea efectiva de realizare a unei tranzactii convenabile pentru participanti, prin cresterea utilitatii aferente acestora. Este necesara introducerea de noi de tehnici care sa conduca la noi strategii apropiate de modul de negociere uman (subiectiv, dinamic, în prezenta unor cunostinte incomplete). Utilizarea tehnicii de data-mining pentru învatare poate reprezenta o solutie în acest caz

O prima directie de dezvoltare identificata se bazeaza pe structura unei aplicatii de comert electronic: un cumparator trebuie sa se autentifice pentru a putea utiliza aplicatia. Deci putem pune în legatura un anumit sablon cu un anumit utilizator. Astfel se va putea îmbunatati caracteristica inteligenta a negocierii, oferindu-se solutii orientate catre utilizator. O alta posibila directie de dezvoltare este implementarea unei functii de feedback.

Aceasta poate fi realizata prin punerea la dispozitia utilizatorilor a unui formular cu întrebări adresate lor care sa determine satisfactia în utilizarea sistemului.

Bibliografie

- 1.Huhns, M., N., Malhotra, A., K. – *Negotiating for Goods and Services*, IEEE Internet Computing, July-August 1999.
- 2.Lin F., Chang K. – *A Multiagent Framework for Automated Online Bargaining*, IEEE Intelligent Systems, vol. 16, nr. 4, iulie-august 2001, p. 41-47
- 3.Muller J.P. - *The Design of Intelligent Agents: A Layered Approach*, Springer, 1997
- 4.Weiss G. (editor) – *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 2000
- 5.Zeng, D., Sycara, K.: *How Can Agent Learn to Negotiate?*, Intelligent agents III, ECAI '96 Workshop, 1996, p.233-244