2014 33rd International Conference of the Chilean Computer Science Society

VirtualMarket: extending *ChileCompra* with agent capabilities for identifying providers associativity opportunities and negotiate alliance participation

Romina Torres, Diana Biscay
Facultad de Ingeniería
Universidad Andrés Bello
Viña del Mar, Chile

Rodrigo Salas
Escuela de Ingeniería Biomédica
Universidad de Valparaíso
Valparaíso, Chile

Oscar Cornejo, Marcelo Aliquintuy, Hernán Astudillo
Departamento de Informática
Universidad Técnica Federico Santa María
Valparaíso, Chile

***Abstract*—*Mercado Público* is a *ChileCompra's* platform for tenders, an environment where Chilean companies offer and demand goods and services of all kinds. The main aim of this platform is to make more efficient and transparent the contracting processes. Nowadays, tenders asking more than one good and/or service are answered separately. Thus, after receiving the offers from the sellers, the buyer must select for each part of the tender the best offer. Unfortunately, this is an inefficient and non optimal process. On the one hand, buyers spend much time in building a joint contract where not necessarily they obtained the best deal and on the other hand, small providers are less likely to be selected (if they apply) when competing with bigger providers or alliances of them (which already offer most products and typically at better prices). To address these issues, we presented *VirtualMarket*, a multi-agent platform, which extends the *ChileCompra's* platform representing providers as software agents (with willingness to cooperate), which, through the creation of alliances, are capable of naturally identifying associativity opportunities, when tenders are published, and negotiate pre-agreements which make them more competitive. In this ongoing work, we propose to use a slightly modified version of the Zeuthen Strategy in order to support providers agents during the negotiation of the terms of the alliance formation without revealing their utility functions. Our preliminary results about the feasibility and performance of our proposal are encouraging.**

Keywords*—*Zeuthen Strategy, negotiation, multi-agent system, virtual markets, electronic commerce.

I. Introduction

In general, a market coordinates the exchange of goods/services and/or money, providing a mechanism to match demand and supply and to determine price (if it is not explicit). Markets are economical systems, which consist of (i) (human) agents acting in the market, (ii) commodities being exchanged, and (iii) a set of rules defining the market. Auctions have been typically used to efficiently allocate goods/services and to determine prices because they may ensure buyers obtain lower prices due to the increased competition and transparency. Even when sellers expect to sell their goods/services to higher prices, this open environment facilitates price regulation. The increasing use of the Internet has led to the emergence of electronic markets such as eBay[1] and Amazon[2], which are well known Internet auction platforms. According to Viamonte et al. [10] the need of having software agents supporting buyers and sellers in trading their goods and services has grown quickly.

Electronic market mediated by software agents is not a new research topic. Mingua et al. [2] already presented a survey that described several works on this area. Dasgupta et al. [1] described MagNet, a system to electronically trade goods or services, which allows negotiation between buyers and providers. Viamonte et al. [10] presented a simulator of a multi-agent market designed to analyze different market strategies, preference models, price algorithms and risks' preferences. Lopez-Carmona et al. [4] presented ANEGSYS, a recommender system for the acquisition of products that uses automatic bilateral negotiation to generate pre-agreements between buyers and providers that are represented as agents. Later, Rau et al. [5] also proposed a negotiation model between buyers and providers. Zulkernine et al. [11] proposed a system that allows a consumer agent to negotiate with multiple providers at the same time to find the most appropriate.

However, most of the negotiation strategies have the disadvantages that (i) software agents need to know the utility function of its opponents (which is admissible for simulation purposes) and (ii) negotiations are possible only between agents of different kind [7].

Therefore, in this ongoing work, which reuses ideas of our previous work [8], [9] we have represented companies of the *ChileCompra* platform with software agents. Thus, providers are in our system active entities seeking complementary providers in order to present joint offers as alliance to the published tenders. Agents implement a slightly modified version of the Zeuthen Strategy [9], in which agents decrease prices in proportion to the time spent since the negotiation started. Thus, we address both issues: agents (of the same kind) forming an alliance are able to negotiate their participation (prices of their products/services, quantity, to name a few) without knowing the opponents' utility function.

This paper has the following structure: Section II describes original Zeuthen Strategy, Section III describes the *VirtualMarket* platform and the negotiation algorithm used by the providers, in Section IV we study the feasibility of the proposal showing results of the study and in Section V we highlight the drawbacks of this ongoing work presenting the future work.

[1]http://www.ebay.com
[2]http://www.amazon.com

1522-4902/15 $31.00 © 2015 IEEE
DOI 10.1109/SCCC.2014.30

II. Theoretical framework

The monotonic concession protocol defines a set of rules for regulating negotiation between software agents. First of all, each agent has its own utility function. The negotiation process is made in several rounds, where at first round, all agents propose offers simultaneously and the negotiation ends when an agent determines that the offering of other agent is the same or better than its own offer. The negotiation also ends if no new concessions are made during a new round. It is not allowed that agents make a meaner offer than those made at previous rounds. These offers are evaluated by each agent using their utility functions where the aim of the negotiation is to increase the utility at each round.

The original Zeuthen strategy was designed to support the negotiation only between two opponent agents at the same time. It is been proved that solutions obtained with this strategy are Pareto optimal and the Nash's product is maximized. The Zeuthen strategy [6] formalizes during the negotiation process, who should concede and how much it should at least be conceded at each round. It allows agents to agree upon a deal by balancing the risk of agents during the negotiation. In each step, the agent with less risk should make a concession, until they make an agreement. This concession should be one that changes the balance of the risks between the agents. After the concession, the agent has done, should have higher risk than its opponent. The willingness of agent i to risk conflict is defined as follows, $Z_i^t = risk_i^t(\delta_i^t, \delta_j^t) =$

$$\begin{cases} 1, & \text{if } U_i(\delta_i^t) = 0 \\ \frac{U_i(\delta_i^t) - U_i(\delta_j^t)}{U_i(\delta_i^t)} & \text{otherwise} \end{cases} \tag{1}$$

where t denotes the $t-th$ round of the negotiation, i and j denote the agents i and j respectively, δ_i^t is the offer made by the agent i at round t, $U_i(\delta_i^t)$ is the utility of agent i at round t according to its own offer, and $U_i(\delta_j^t)$ is the utility of agent i at round t according to the offer of the agent j.

Due to the Zeuthen strategy uses the monotonic concession protocol, the risk computations are made simultaneously at each round in order to determine who must concede, and how much it should be conceded, implying that agents must know the utility functions of the other agents. The agent with smaller value of risk should concede. The value of this concession is the minimum value that invert the risk [3]. For instance, if agent j has risk smaller than agent i, then agent j can determine the minimum value that it should concede by isolating δ^{t+1} from the following inequality (inverting risks).

$$risk_i^{t+1} < risk_j^{t+1}$$

$$\frac{U_i(\delta_i^t) - U_i(\delta_j^{t+1})}{U_i(\delta_i^t)} < \frac{U_j(\delta_j^{t+1}) - U_j(\delta_i^t)}{U_j(\delta_j^{t+1})}$$

III. Proposal

In this Section, we present an overview of the *VirtualMarket* platform [3], where providers agents form alliances, negotiate the terms of their participation, and bid the tender as a group. During the negotiation process, agents use a slight modification of the Zeuthen strategy, which implements the idea that agents only knows the proposal and risk from others co-habitants agents from the system and not their utility functions.

VirtualMarket is a multi-agent system capable of identifying associativity opportunities between several companies that besides of notifying to different branch offices[4] when a relevant tender for them starts (according to their sector), it also alerts the convenience of presenting in a eventual business as an alliance with other complementary providers.

We have chosen to build the multi-agent systems of this platform over JADE[5]. We have built a Web client using Grails, which is a Web application framework. In architectural terms, the JADE container is embedded as a plugin inside of the Java Virtual Machine that Grails provides. Figure 1 shown the interaction of the main components and the information flow with the agent container in the system. It is very important to say that components shown are exposed through an API, which are accessible as demo through a Web client that consumes this API.

The main interactions in the JADE container are (i) for activating agents who represent the system providers, (ii) for alliance generation, i.e. when a tender starts, there is a specific service that notifies the coalition (alliance formation) service, and when this happens it interacts with the JADE container and communicates that every active agent in the system has to start to dialog so they should generate alliance in order to satisfy tender conditions, and (iii) for supporting the negotiation process, which is for creating, modifying, deleting negotiation instances and most important, to coordinate the dialogs between the container agents.

As we can see in Algorithm 1, in this ongoing work, each provider agent implements the same negotiation protocol (monotonic concession protocol) and a modified version of the Zeuthen strategy. Due that agents only negotiate price we represent offers by P instead of δ. During the negotiation process, an agent i evaluates the utility that it would get accepting the counterpart offer. Then, each agent calculates its own utility with the counterpart offer and also each one estimates the risk associated with that operation. Consequently, the agent with less risk has to concede and therefore in this case to reduce its price. Especially, the new price (P_i') offered by that agent will be the previous price multiplied by a conversion factor. This conversion factor is established with the idea that opponents do not need to know the utility function of the other agents. This conversion factor is the minimum value between $\{0.99,\ 1 - 0.5 \cdot (risk_j - risk_i)\}$.

Once agents have determined the new prices, the offers are sent and a new negotiation cycle begins. Agents receiving an offer will refuse it if it reaches the minimum boundary (the reserved price).

IV. Experiments

In this work we use to study the performance and properties of our proposal two tenders requesting two products of

[3]http://dev2.toeska.cl/virtualmarket/

[4]It is important to notice that providers can have multiples branch offices whose business rules aren't necessarily the same, therefore in this work, each provider's branch office is represented by an independent agent.

[5]http://jade.tilab.com/

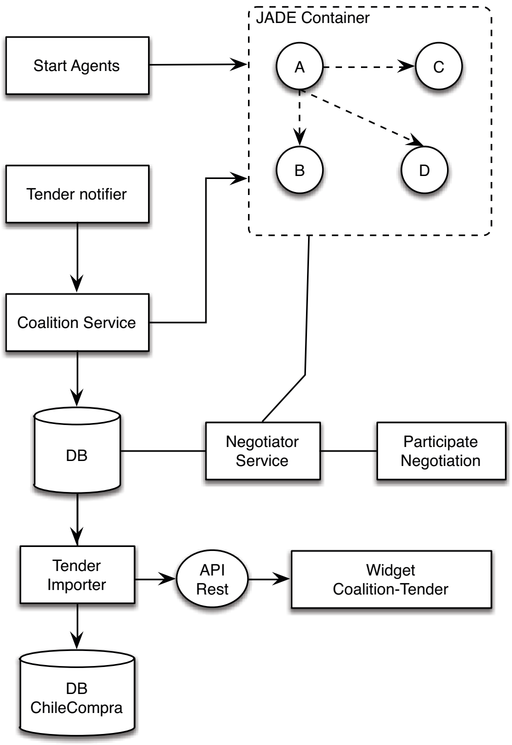

Figure 1: Agent's conceptual diagram in VirtualMarket.

different nature.

- **Tender ID :** 656236-1-LE13
- **Name:** Acquisition of a drying oven and conductivity
  - **Product 1**
    - **Name:** Conductivity meters
    - **Sector:** (37036) Instruments and accessories for electrochemical measurements
  - **Product 2**
    - **Name:** Air dryers
    - **Sector:** (37333) Industrial drying equipment
- **Tender ID:** 1088-2-LP13
- **Name:** Design of a Regional Botanic Park
  - **Product 1**
    - **Name:** Temporary services of architecture
    - **Sector:** (38231) Temporary staff services
  - **Product 2**
    - **Name:** Studies for project location
    - **Sector:** (38223) Project management

These experiments were executed using a machine with the following characteristics:

- **RAM:** 8 GB
- **OS:** OS X 10.9.3 (13D65)
- **Processor:** Intel(R) Core(TM) i5-3330S CPU @ 2.70GHz

Algorithm 1 Modified version of the Zeuthen strategy

Agent i propose P_i
if Agent j accept or refuse offer P_i **then**
  Negotiation is closed
  return
end if
while $risk_i < risk_j$ **do**
  Agent i receives proposal P_j
  if $U_i(P_j) \geq U_i(P_i)$ **then**
    Accept P_j
    return
  end if
  if $U_i(P_i) == 0$ **then**
    $risk_i = 1$
  else
    $risk_i \leftarrow \frac{U_i(P_i) - U_i(P_j)}{U_i(P_i)}$
  end if
  Agent i sends $risk_i$ to Agent j
  Agent i receives $risk_j$ from Agent j
  if $risk_i < risk_j$ **then**
    $P_i \leftarrow P_i$'
    if $C_i == R_i$ **then**
      Agent i refuse offer P_j
      return
    end if
  end if
end while

A. First experiment

The objective of this first experiment is to study the performance of *VirtualMarket* in a real environment. Hence, we study for a tender how much time it takes to generate valid alliances when the amount of active agent is growing. To do so, it has to be performed the following tasks in the system:

1) Initiate n agents in the system.
2) Initiate a tender fixing dates, budget and evaluation criteria with their respective ponderation percentages.
3) The active agents starts to dialog generating alliances, according to the sectors that are being satisfied in the tender for each branch office.

Through this process, it will be measured the amount of alliances generated, the time that this process takes and the effective amount of agents participating in the activity.

We also measure the execution time of the algorithm, the number of formed alliances, the number of agents integrated alliances, and as an independent variable the number of agents used in each iteration.

The results can be appreciated in Table I. According to this data, it was decided to analyze the alliances generation versus the amount of agents used in each iteration; based on this, and observing the Figure 2, it was concluded that, if the **number of active agents** in the system grows, so does the **number of alliances** and **agents that participate in them** until the number of active agents is equal or greater than 100, when the **number of alliances** and **agents that participate in them** do not increase more (700 alliances formed of different combination of these 100 agents).

Table I: Alliance generation of a tender

Sector active agents	# agent that participates in an alliance	# alliance created	Exec time[s]
10	0	0	0
50	50	209	2,21
100	100	688	2,46
150	100	558	2,66
205	100	581	16,38

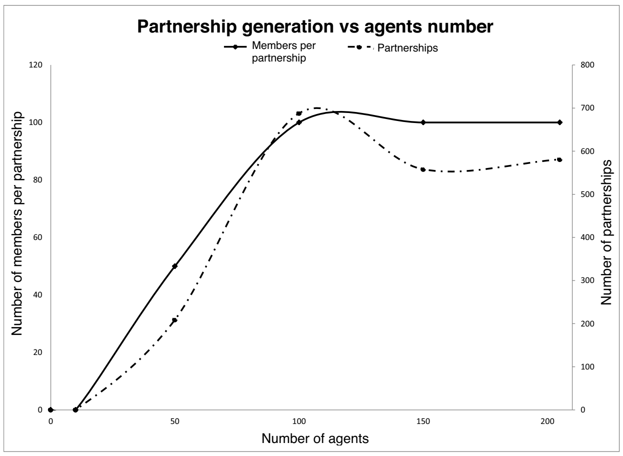

Figure 2: Alliance generation v/s number of agents

In the case of forming alliances of two tenders, 803 agents were initiated, i.e. all providers agents that accomplish the sectors of both tenders (656236-1-LE13 and 1088-2-LP13). 1.312 alliances were generated and the execution time was 5,75 seconds.

Table II: Data

Test	It Converged	Budget
1	Yes	300
2	No	120
3	Yes	320
4	No	110
5	Yes	1002
6	Yes	1000,5
7	Yes	20003
8	Yes	20010
9	No	30
10	Yes	55
11	Yes	2000
12	Yes	3000
13	Yes	4000

For the study of negotiations two kinds of tests were performed. In the first test, it was used just one tender. For the same tender, it was performed 13 iterations in which the goal was to check if varying the desired, reserved and budget prices different results would be obtained. In Table II we can see that the 77% of the cases converged.

In Table III, it can be observed the effect of the price variation in the round number of each test, and the relation existing between the desired prices, the risk and the value obtained finally in the several negotiations performed.

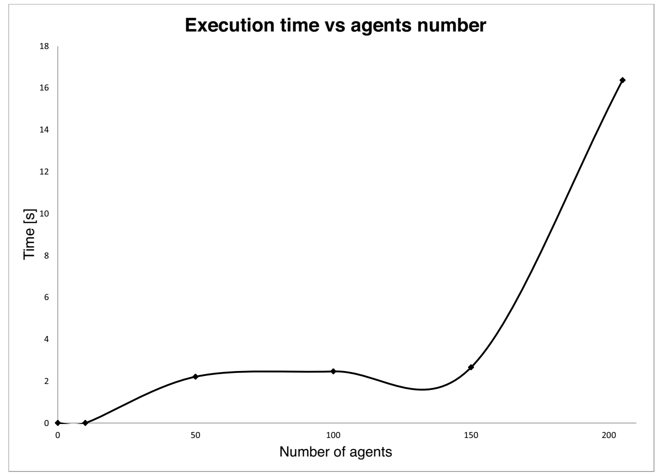

Figure 3: Alliances generation v/s Algorithm execution time.

Table III: Negotiation results

Test	#rounds	Utility P1-P2	Price P1-P2	Final price
1	1	1 - 0,35	60 - 500	60 - 240
2	19	0,0001 - 0,05	60 - 500	50 - 120
3	1	1 - 0,9647	6 - 300	6 - 314
4	110	0,0001 - 0	6 - 300	2 - 110
5	1	1 - 0,02	1 - 1050	1 - 1.001
6	70	0,5 - 0,0001	1 - 1050	0,5 - 1.000
7	121	0,2 - 0,0001	10 - 20030	2,99 -20000,01
8	1	1 - 0,0001	10 - 20030	10 - 20000
9	92	0,0001 - 0	0,5 - 60	0,2 - 30
10	1	1 - 0,45	0,5 - 60	0,5 - 54,5
11	138	0,0001	2000 - 2000	1000 - 1000
12	58	0,49 - 0,51	2000 - 2000	1494,35 - 1505,65
13	0	1	2000 - 2000	2000 - 2000

B. *Second experiment*

In our second experiment we measure the time the agents take to negotiate, varying the amount of simultaneous negotiations in the system observing the number of times that agents modify their prices and how they vary their utility and risk functions according to that. It is very important to notice that the negotiation process in *VirtualMarket* is as follows:

1) The process initiate when all alliances have been formed.
2) Each branch office of the alliance has to declare its reserve and desired prices.
3) When the last member of the alliance declare its prices the negotiation is triggered.
4) Each agent that represents a branch office of the alliance bargains the negotiation prices according to a function that allows it to estimate the utility and risk at each round. If both parties agree, the alliances is formed.
5) If alliances' participants do not reach an agreement, even after several intermediate negotiations, it is determined that the alliance was unsuccessful.

In the second kind of test it was decided to measure how the number of simultaneous negotiations affect the system performance. For this experiment, in each iteration was simulated a determined number of negotiations for different tenders, the execution time and the number of round of the process was measured. The results can be seen in Table IV.

It can be noticed that execution time does not excessively grow as the amount of simultaneous negotiations was increased.

Table IV: Times of simultaneous negotiations

Test	# simultaneous alliances	# rounds	Exec time[s]
1	1	15	2,51
2	2	30	0,21
3	4	48	0,35
4	5	72	0,39

V. Conclusions and future work

The limits of this proposal has to do with the algorithm inability to handle negotiations between more than two agents at the time, fundamental characteristic to implement this work in a real environment. Stated the above, as future work it will be developed a new proposal that includes the feature so offerers can negotiate their participation in a society when tenders asks more than two products.

Acknowledgment

This work has been funded by *FONDEF CA12i10380*.

References

[1] P. Dasgupta, N. Narasimhan, L.E. Moser, and P.M. Melliar-Smith. Magnet: mobile agents for networked electronic trading. *Knowledge and Data Engineering, IEEE Transactions on*, 11(4):509–525, Jul 1999.

[2] Minghua He, N.R. Jennings, and Ho fung Leung. On agent-mediated electronic commerce. *Knowledge and Data Engineering, IEEE Transactions on*, 15(4):985–1003, July 2003.

[3] Kivanc Karakas, Irem Dikmen, and M Talat Birgonul. Multiagent system to simulate risk-allocation and cost-sharing processes in construction projects. *Journal of Computing in Civil Engineering*, 27(3):307–319, 2012.

[4] Miguel A Lopez-Carmona, Ivan Marsa-Maestre, J.R. Velasco, and Bernardo Alarcos Alcazar. Anegsys: An automated negotiation based recommender system for local e-marketplaces. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 5(6):409–416, Oct 2007.

[5] Hsin Rau, Chao-Wen Chen, and Wei-Jung Shiang. Development of an agent-based negotiation model for buyer-supplier relationship with multiple deliveries. In *Networking, Sensing and Control, 2009. ICNSC '09. International Conference on*, pages 308–312, March 2009.

[6] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, MA, USA, 1994.

[7] Kwang Mong Sim. Complex and concurrent negotiations for multiple interrelated e-markets. *Cybernetics, IEEE Transactions on*, 43(1):230–245, Feb 2013.

[8] Romina Torres and Hernan Astudillo. A market-based approach to the dynamic reconfiguration problem of service-based systems. *International Journal of Innovative Computing Information and Control*, 10(1):115 –132, february 2014.

[9] Romina Torres, Denise Rivera, and Hernán Astudillo. Web service compositions which emerge from virtual organizations with fair agreements. In Gordan Jezic, Mario Kusek, Ngoc Thanh Nguyen, Robert J. Howlett, and Lakhmi C. Jain, editors, *KES-AMSTA*, volume 7327 of *Lecture Notes in Computer Science*, pages 34–43. Springer, 2012.

[10] M.J. Viamonte, C. Ramos, F. Rodrigues, and J.C. Cardoso. Isem: a multiagent simulator for testing agent market strategies. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(1):107–113, Jan 2006.

[11] Farhana H. Zulkernine and Patrick Martin. An adaptive and intelligent sla negotiation system for web services. *IEEE Transactions on Services Computing*, 4:31–43, 2011.