Designing an Effective E-Market: An Overview of the CAT Agent

Ana Petric, Vedran Podobnik, Andrej Grguric, Marijan Zemljic

University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Telecommunications
Unska 3, HR-10000 Zagreb, Croatia
{ana.petric, vedran.podobnik, andrej.grguric, marijan.zemljic}@fer.hr

Abstract

Markets are complex economic systems that function as intermediaries between consumers and providers. Traditional methods for studying market mechanisms, utilized by economists and mathematicians, use analytical approaches. Today, the rapid proliferation of the Internet is stimulating the transformation of classical markets into electronic markets (e-markets). Traditional theories cannot adequately cover the dynamics of complex computing systems, such as the e-market, creating the need for new methods. Consequently, computing scientists and experts have joined their efforts with economists and mathematicians to create a novel approach for modeling and evaluating market processes, referred to as agent-based computational economics. The Trading Agent Competition Mechanism Design Tournament, also known as the CAT Tournament, is one of latest efforts in that direction. In the CAT Tournament, trading agents are created by the organizers of the competition and entrants compete by defining market rules. In this paper, a description of the CAT Tournament is given, followed by a look at the main features of our entrant for 2007, the CrocodileAgent.

Introduction

In the past decade, the accelerated economic globalization trend is redefining the trading process. Namely, functions of supply and demand are becoming more and more dynamic and the possibilities of choice have risen to amazing levels (Milicic et al. 2007). At the same time, the rapid proliferation of the Internet provokes companies and individual consumers, to use the Internet infrastructure as a trading platform. Electronic markets (e-markets) are Internet-based systems that serve as digital intermediaries which create value by bringing consumers and providers together (Podobnik, Trzec, and Jezic 2007; He, Jennings, and Leung 2003). Moreover, e-markets are complex adaptive economic and computational systems (Tesfatsion 2006) characterized by numerous interactions of many stakeholders that can be represented by *software agents* (Podobnik, Petric, and Jezic 2007; Fasli 2007). Consequently, two major issues regarding e-markets arise: "How to design an e-market?", and "How to understand their processes and measure their effectiveness?". The first question is addressed in Section 2, while the second is discussed in Sections 3 and 4. Namely, Section 3 describes the Trading Agent Competition (TAC) Market Design Game (i.e., the CAT Game), while Section 4 presents the CrocodileAgent, our entrant for CAT Tournament 2007. Section 5 concludes the paper.

Designing an E-Market

As a subfield of game theory, *mechanism design* is dedicated to achieving specific goals by manipulating the rules of a designed mechanism (Niu et al. 2006). When designing a market, the actual goal is to uphold a particular set of desired behavioral rules by imposing constraints on participants' behavior and the transactions carried out between them (Subrahmanian and Talukdar 2004). The explosive development of electronic business (e-business) and appropriate tools that can embed complex market rules have contributed to the importance of studying mechanism design over the last few decades (Bourbeau et al. 2005).
Auctions, due to their well defined protocols, are suitable enablers of negotiations in e-markets. Auctions are defined as a market institution that acts in pursuit of a set of predefined rules in order to compute the desired economic outcome (i.e., high allocation efficiency) of social interactions (Wurman, Wellman, and Walsh 2002). Most auctions can be distinguished with respect to how they manage three core activities (Wurman, Wellman, and Walsh 1998). The first activity includes receiving and processing bids. Clearing is the second and most important activity since it determines the transaction price for a matched buyer-seller pair. The third activity concerns revealing information about the status of active bids.

CAT Tournament - The Game

In the past, economists and mathematicians have analyzed market mechanisms by applying traditional analytical methods from game theory (Niu et al. 2006) which focus on „the rules of the game". The evolution of computers has enabled scientists to run simulations of the designed mechanisms in order to compare and improve them. Such computational studies, where economic processes are modeled as dynamical systems with interacting agents, are referred to as *agent-based computational economics*

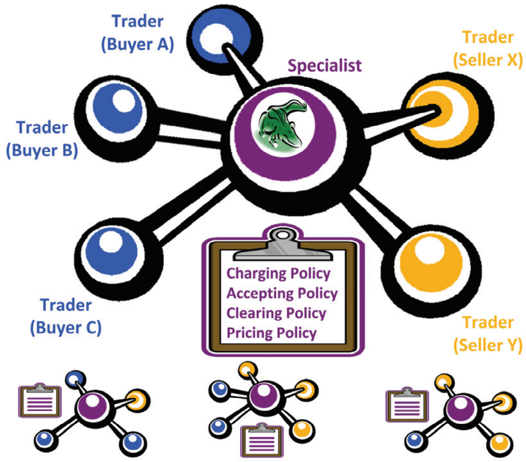

Fig. 1. Logical model of the CAT game

(ACE). An example of an ACE system is the CAT Competition (http://www.marketbasedcontrol.com/).
As shown in Figure 1, there are two types of agents participating in the CAT game: *traders* and *specialists*. Each contestant is in charge of a single specialist. Specialists represent the market and compete in defining effective rules for matching buyers and sellers. Furthermore, they determine fees for providing services with the objective to maximize their profit. Traders are intelligent software agents created by the organizers of the competition, where each is given the role of buyer or seller. Each trader is assigned a set of private values ***PV*** for traded goods and a market selection strategy ***MSS*** responsible for choosing a specialist and a trading strategy ***TS.*** The trading strategy is chosen from a set of strategies (i.e., GD, ZI-C, ZIP and RE) selected by the game organizers (Gerding et al. 2007) and used to generate shouts (i.e., bids and asks). Each game consists of multiple (virtual) trading days, where days are comprised of several rounds during which traders submit their shouts. During each round, specialists match shouts between buyers and sellers and determine a transaction price.

The CrocodileAgent

The CrocodileAgent is an intelligent software agent that plays the role of a specialist in the CAT Game. It was developed at the University of Zagreb, Croatia. Figure 2 illustrates the CrocodileAgent's basic architecture consisting of: a *communication component*, a *decision making component* and a *knowledge base component*.
Since specialists communicate by exchanging messages, the CrocodileAgent needs a communication component capable of interpreting incoming messages and extracting useful information from them. The decision making component evaluates incoming shouts and interprets game events. Furthermore, this component considers received shouts and decides which actions to take based on the agent's objectives, its information about the environment, and shouts history. The knowledge base component performs an analysis of relevant knowledge. Along with shout and transaction history, the CrocodileAgent's knowledge base holds information regarding the agent's environment, its preferences and objectives, and knowledge about its opponents.

The Charging Policy

Specialists earn money by charging fees for provided services to traders. There are five types of fees charged as follows: *registration fees* are charged when traders register with a specialist; *information fees* are charged for providing traders and other specialists with shout and/or transaction information; *shout fees* are charged for the successful placement of bids and asks; *transaction fees* are charged for each successfully executed transaction; and, finally, *profit fees* are charged as a share of the profit made by traders. The main challenge is to set these fees to values which achieve a good trade-off between attracting traders and profit. Consequently, the CrocodileAgent divides the game into the *luring phase* and the *charging phase*.
In the luring phase, the CrocodileAgent tries to increase its market share and attract traders by providing them with information and services free of charge. In the charging phase, the CrocodileAgent sets fees according to its acquired knowledge regarding the fees of other specialists, their estimated ranks, market shares, profit shares, etc. Determining an appropriate transition point at which to switch from the luring phase to the charging phase can be a difficult problem. Namely, in the absence of a market share increase, the agent can potentially stay in the luring phase longer than originally anticipated. In order to prevent this unfavorable situation from happening, the CrocodileAgent monitors the distribution of traders among other specialists. Key parameters which are monitored are the average number of traders registered with the specialist *up to* the

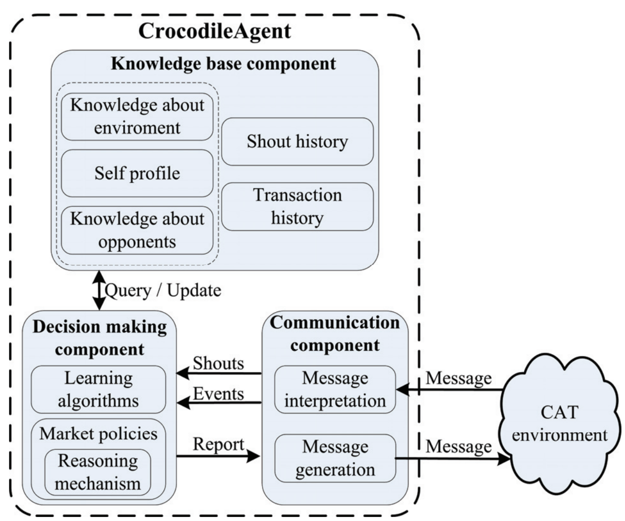

Fig. 2. The CrocodileAgent's architecture

current day (T_{avg}) and the number of traders registered with the specialist *on* the current day (T_{cur}). If T_{cur} is within the given boundaries (e.g., $T_{cur} \in [0.85\ T_{avg}, 1.15\ T_{avg}]$), we consider that T_{cur} is converging towards the average value. If this is true for a majority of the specialists (e.g., 75%), we assume that the distribution of traders among specialists will remain mostly unchanged in the following days. Thus, the CrocodileAgent shifts to the charging phase. Drawbacks of this strategy are as follows. In the starting phase of the game, there exists the possibility that the transition condition is accidentally fulfilled due to a higher frequency of traders shifting from one specialist to another. Furthermore, the condition for the transition might never be fulfilled, forcing the agent to stay in the luring phase for the entire game. In order to try to prevent this from occurring, we specify a lower and upper bound on the number of days the agent can stay in the luring phase. Throughout the entire game, the CrocodileAgent evaluates agents' performances in order to estimate the current winner. Listing 1 gives the relevant parameters used by the CrocodileAgent to calculate its fees.

L_{phase} – If the specialist is in the luring phase, this parameter is *true* otherwise *false*;

W_{me} – If the CrocodileAgent is predicted as the current winner, this parameter is *true* otherwise *false*;

N_{wnd} – The size of the sliding window that determines how far in the past values are considered to be relevant;

W_{avg} – The average value of the winner's fees over the last N_{wnd} days (calculated separately for each type of fee);

O_{avg} – The average value of all specialists' fees over the last N_{wnd} days (calculated separately for each type of fee);

N_{win} – The number of days that the predicted winner has been winning;

T_{win} – The number of days that need to pass before we take winner fees under consideration;

Z – If all other specialists are currently charging zero prices for all types of fees this parameter is *true*, otherwise *false*;

C – If all other specialists have set the price for at least one type of fee above zero this parameter is *true*, otherwise *false*;

MS_{cur} – The average value of the CrocodileAgent's market share over the last ***wnd*** days;

$X, Y \in [0,1]$ – Heuristically determined parameters;

MS_{des} – The CrocodileAgent's desirable market share calculated as: $MS_{des} = 1.5 \times \frac{NumberOfTraders}{NumberOfSpecialist}$.

Listing 1. Parameters relevant for the calculation of fees

The algorithm used to set fees is presented in Listing 2. During its charging phase, the CrocodileAgent sets the price of all fees to O_{avg}. In case the currently leading specialist is winning by more than N_{win} and its W_{avg} is lower than O_{avg}, the fees are decreased by W_{avg}. If the CrocodileAgent is currently winning, its fees remain constant. This way, we ensure that the CrocodileAgent does not charge higher fees at times when it is not winning, since that could lead to a rapid loss of traders. Additionally, if all other specialists are charging zero, the CrocodileAgent sets its registration fee to a value slightly above zero to ensure the highest profit score of the day.

```
if (L_phase == true)
    set all fees to 0$;
else
    if (W_me == true)
        leave all fees at current values:
    else
        if ((N_win > T_win) && (W_avg < O_avg))
            set all fees to X * W_avg;
        else
            set all fees to O_avg;
    if (Z == true)
        registration fee = Y$ , other fees = 0$;
    if ((C == true) && (MS_cur < MS_des))
        set all fees 0$;
```

Listing 2. The algorithm for setting fees

The Pricing Policy

The pricing policy determines the price of a transaction for a matched ask-bid pair. The CrocodileAgent uses a slightly modified *n-pricing policy* (Niu et al. 2006) as follows. The original n-pricing policy maintains a sliding window by keeping the last *n* matched ask-bid pairs and uses their average price to set price p_t in the current transaction (i.e. the *t*th transaction, where *t* is the number of matched ask-bid pairs at a given point in time). If the resulting transaction price is outside the boundaries of the *t*th ask and bid prices, p_{a_t} and p_{b_t} respectively, then the price is simply set to p_{a_t} (if p_t is lower than p_{a_t}) or to p_{b_t} (if p_t is greater than p_{b_t}). In such cases, one of the traders has zero profit from the transaction. This outcome is unfavorable for the specialist since the profit from a transaction is a significant factor for most traders in their ***MSS***. In order to reduce the loss of traders and ensure that both sides obtain a profit from the transaction, we modified p_t with a heuristically determined factor $W \in [0,1]$, as shown in Listing 3.

```
if (p_t >= p_(b_t))
    p_t = p_(b_t) - (p_(b_t) - p_(a_t)) * W;
else if (p_t <= p_(a_t))
    p_t = p_(a_t) + (p_(b_t) - p_(a_t)) * W;
else
    p_t = p_t;
```

Listing 3. Calculation of the transaction price

The Accepting and Clearing Policy

The accepting policy determines which shouts will be accepted. The CrocodileAgent uses an implementation of the shout improvement rule (Niu et al. 2006) which requires bid prices to be above the estimated equilibrium price, while the ask prices must be below it. The specialist maintains a sliding window which contains a series of *m* transaction prices and uses their average value as the estimated price. The clearing policy determines when to execute transactions between matched ask-bid pairs. Since transaction efficiency greatly depends on matching functions, the CrocodileAgent clears the market every second round.

CAT Tournament 2007

The CrocodileAgent finished the Qualification Stage in the leading position, while taking third place out of ten in the Final Stage. CAT Tournament 2007 was the first of its kind to take place and proved very successful. This competition provides a fruitful and entertaining means of addressing important issues in the field of ACE and has the potential to significantly increase its popularity.

Conclusions and Future Work

The complexity and dynamics of auctions makes it difficult to apply standard game-theoretic methods to achieve their specific goals. Consequently, new experimental approaches are needed, such as the CAT Competition. With software agents becoming increasingly prevalent, a better understanding of their interactions in the marketplace is highly desirable. In the CAT Game, traders' decision making policies play an important role in the potential success of a specialist, since they determine which specialist traders register and trade with on a certain day. The initial distribution of traders also proved to be very important. It is our opinion that an uneven distribution of traders at the start of the game is not entirely fair since specialists with more traders initially assigned have a certain advantage. Furthermore, if no fixed expenses exist, a specialist can provide services free of charge during the entire game and still be awarded the top score. In this paper, we presented the CAT Competition and described the main features of the CrocodileAgent, our entry in the 2007 competition. The CrocodileAgent's charging strategy in the Qualification Stage proved to be very effective. This was recognized by other teams which then applied similar strategies to their specialists in Final Stage. For future work, we plan to analyze the CrocodileAgent's performance through a series of controlled experiments with other specialists and entrants in the CAT Competition 2007. Based on the performed analysis, improvements will be implemented in the CrocodileAgent's trading strategy.

References

Bourbeau, B.; Gabriel Crainic, T.; Gendreau, M. and Robert, J. 2005. Design for optimized multi-lateral multi-commodity markets. *European Journal of Operational Research* 127(2): 503-529.

Fasli, M. 2007. *Agent Technology for E-Commerce*. Chichester: Wiley & Sons.

Gerding, E.; McBurney, P.; Niu, J.; Parsons, S.; and Phelps, S. 2007. Overview of CAT: A market design competition, Technical Report ULCS-07-006, Department of Computer Science, University of Liverpool, Liverpool, UK. Version 1.1.

He, M.; Jennings, N.R.; and Leung, H. 2003. On Agent-Mediated Electronic Commerce. *IEEE Transactions on Knowledge and Data Engineering* 15(4): 985-1003.

Milicic, T.; Podobnik, V.; Petric, A.; and Jezic, G. 2007. The CrocodileAgent: A Software Agent for SCM Procurement Gaming. In N.T. Nguyen et al. (eds.): IEA/AIE 2008, *Lecture Notes in Computer Science* 5027: 865-875. Berlin Heidelberg: Springer-Verlag.

Niu, J.; Cai, K.; Parsons, S.; and Sklar. E. 2006. Reducing price fluctuation in continuous double auctions through pricing policy and shout improvement rule. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, 1143-1150. Hakodate, Japan. ACM.

Niu, J.; Cai, K.; Gerding, B.; McBurney,P.; Parsons, S. 2008. Characterizing Effective Auction Mechanisms: Insights from the 2007 TAC Market Design Competition. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems*. Estoril, Portugal. ACM. Forthcoming.

Podobnik, V.; Petric, A.; Jezic, G. 2007. An Agent-Based Solution for Dynamic Supply Chain Management. *Journal of Universal Computer Science*. Forthcoming.

Podobnik, V.; Trzec, K.; Jezic, G. 2007. Context-Aware Service Provisioning in Next-Generation Networks: An Agent Approach. *International Journal of Information Technology and Web Engineering* 2(4): 41-62.

Satterthwaite, M.; and Williams, S. R. The Bayesian theory of the k-double auction. In Friedman, D.; and Rust, J. eds. 1993. The Double Auction Market: Institutions, Theories and Evidence. Cambridge, MA.: Perseus Publishing.

Subrahmanian, E.; and Talukdar, S. 2004. Engineering of markets and artifacts. *Electronic Commerce Research and Applications* 3(4): 369-380.

Tesfatsion, L. 2006. Agent-Based Computational Economics: A Constructive Approach to Economic Theory. In: Tesfatsion, L.; and Judd, K. L. eds. 2006. Handbook of Computational Economics. Amsterdam, Netherlands: Elsevier.

Wurman, P.; Wellman, M.; and Walsh, W. 1998. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems* 24(1): 17-27.

Wurman, P.; Wellman, M.; and Walsh, W. 2002. Specifying Rules for Electronic Auctions. *AI Magazine* 23(3): 15-23.