# JCAT: A Platform for the TAC Market Design Competition

Jinzhong Niu and Kai Cai
Simon Parsons
City University of New York
{jniu,kcai}@gc.cuny.edu
parsons@sci.brooklyn.cuny.edu

Enrico Gerding
University of Southampton
eg@ecs.soton.ac.uk

Peter McBurney and Thierry Moyaux
Steve Phelps and David Shield
University of Liverpool
{p.j.mcburney,t.moyaux}@liverpool.ac.uk
{sphelps,d.t.shield}@liverpool.ac.uk

## Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence: *Multiagent systems*

## General Terms

Algorithms, design, economics, experimentation, measurement

## Keywords

Mechanism design, trading agent competition, double auctions, trading strategies

## 1. INTRODUCTION

Auctions, when well designed, result in desirable economic outcomes and have been widely used in solving real-world resource allocation problems, and in structuring stock or futures exchanges. The field of auction mechanism design has drawn much attention in recent years from economists, mathematicians, and computer scientists. In traditional auction theory, auctions are viewed as games of incomplete information and traditional analytic methods from game theory have been successfully applied to some simple types of auctions. However, the assumption of prior common knowledge in the incomplete information approach may not hold in some auctions, and computing analytic solutions may be infeasible in other auctions. Both of these problems hold in the case of *continuous double auctions*[1].

As a result of these problems, researchers often use computer simulation of auctions in which traders are software agents. Such agents, armed with various learning algorithms and optimization techniques, have been shown to produce outcomes similar to those observed in auctions with human subjects [7]. Indeed, software traders are capable of outperforming human traders [3]. Along with the automation of traders, computer scientists have started to take evolutionary and adaptive approaches to automatically creating auction mechanisms [1, 9, 10]. Although this work has produced promising results, it has one common theme — the only comparisons that are made are indirect. The results from one lone market are compared with those of another lone market. In contrast, not only do traders in an auction compete against each other, but real market institutions compete against each other. In addition, existing work usually compares auction mechanisms in different settings which vary according to the availability of information, computational resources, and so on. The conclusions of these studies are thus difficult to compare and cumulate. It is therefore desirable to have a platform that allows multiple markets to compete against each other, and allows market mechanisms to be evaluated in a uniform way. The JCAT[2] system that we introduce in this paper addresses these concerns.

JCAT extends the Java Auction Simulator API, JASA[3] adding support for multiple parallel markets with trading agents moving between them. It has been used to conduct research on computational auction design [8] and was successfully used as the game server in the first Trading Agent Competition (TAC) Market Design Competition (CAT) [5].[4]

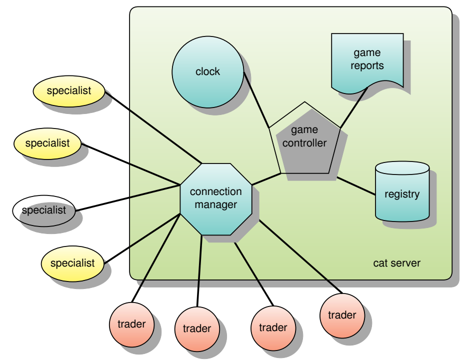


Figure 1: The topology of a CAT game.

## 2. WHAT JCAT PROVIDES

JCAT provides the ability to run what we will call "CAT games", each of which is an interaction between markets and traders. A typical CAT game consists of a CAT server and several CAT clients, which may be trading agents or specialists (markets). As illustrated in Figure 1, the CAT server works as a communication hub between CAT clients. A registry component records all game events and validates requests from traders and specialists. Various game report modules are available to process game events, calculate and output values of different measurements for post-game analysis.

[1] A continuous double auction involves both buyers and and sellers, and both kinds of trader are allowed to make or accept an offer at time during the auction.

[2] http://jcat.sourceforge.net/.

[3] http://www.csc.liv.ac.uk/~sphelps/jasa/.

[4] http://www.marketbasedcontrol.com/blog/index.php?page_id=5.

**Cite as:** Title, Author(s), *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

A CAT game lasts a certain number of *day*s, each day consists of *round*s, and each round lasts a certain number of *tick*s, or milliseconds. The game clock in the game server fires events to notify clients of opening and closing of each day and round intervals.

Each trading agent is assigned private values for the goods it will trade. For buyers the private value is the most it will pay for a good. For sellers, the private value is the least it will accept for a good. The private values and the number of goods to buy or sell make up the demand and supply of the markets. Private values remain constant during a day, but may change from day to day, depending upon the configuration of the game server.

Each trading agent is endowed with a *trading strategy* and a *market selection strategy*. The first specifies how to make offers, the second specifies which market to choose to make offers in. Trading strategies provided in JCAT include those that have been extensively researched in the literature and some of them have shown to work well in practice, e.g., ZI-C [7], RE [4], ZIP [2], and GD [6]. A typical class of market selection strategies treats the choice of market as an $n$-armed bandit problem where daily profits are used as rewards when updating the value function.

Specialists facilitate trade by matching offers and determining the trading price in an exchange market. Each specialist operates its own exchange market and may choose its own auction rules — the aim of the CAT competition is to create a specialist that optimizes a particular set of measures [5]. Specialists may have adaptive strategies such that the policies change during the course of a game in response to market conditions for desired outcomes. JCAT provides a reference implementation of a parameterizable specialist that can be easily configured and extended to use policies regulating different aspects of an auction.

A specialist typically includes components that regulate aspects of its market. The following components were common in entrants to the first CAT competition [8]. *Matching policies* define the set of matching offers in a market at a given time. *Quoting policies* determine the ask quote and bid quote, which respectively specify the upper bound for offers to sell and the lower bound for offers to buy that may be placed in the market at a given time. *Shout accepting policies* judge whether a request by a trader to place an offer in the market should be accepted or rejected. *Clearing conditions* define when to clear the market and execute transactions between matched offers. A *pricing policy* is responsible for determining transaction prices for matched ask-bid pairs. The decision may involve only the prices of the matched offers, or more information including market quotes. *Charging policies* determine the charges a specialist imposes on a trading day. A specialist can set its fees, or *price list*, which are charged to traders and other specialists who wish to use the services provided by the specialist. Each specialist is free to set the level of the charges for registration with a specialist, for making an offer, for completing a transaction, and so on.

# 3. IMPLEMENTATION

JCAT is written in Java, and adopts a client/server scheme for high flexibility and scalability. Its socket-based communication and the use of a plain-text message language (CATP) permits clients to be written in virtually any popular programming language, and CAT games can run across the Internet as in the 2007 TAC CAT competition.

The JCAT class library provides an extensible framework in which new auction rules can be easily implemented and coupled with other policies. A variety of learning algorithms have been included to support adaptive strategies. JCAT has a user-friendly interface based on Java Swing to monitor an on-going game, and also supports a PHP interface allowing results to be displayed on the Web.

# 4. MARKET DESIGN COMPETITION

JCAT was successfully used as the game server during the first TAC CAT Competition, held in July 2007 at the AAAI conference. Prior TAC competitions featured competing trading agents that aimed to maximize their payoffs by interacting in a single market. The TAC CAT competition did just the opposite. Each entrant in the competition provided a specialist, and these specialists competed against each other for market share, profits (by levying fees on traders), and maintaining a high transaction success rate. Traders in CAT games were provided by the competition. The Market Design Competition, which will run at AAAI again this year, provides an ideal testbed for modeling competition among market institutions, comparing different auction mechanisms, and evaluating them in a uniform way.

In theory, a CAT game may have both traders and specialists submitted by entrants and two competitions, one for traders and the other for markets, thus coupled together. This would more closely simulate the real world where both traders and markets adapt quickly and effectively to either the changes of their competitors or those of potential business partners to achieve their economic goals.

# 5. REFERENCES


[1] D. Cliff. Evolution of market mechanism through a continuous space of auction-types. Technical report, Hewlett-Packard Research Laboratories, Bristol, England, 2001.

[2] D. Cliff and J. Bruten. Minimal-intelligence agents for bargaining behaviours in market-based environments. Technical report, Hewlett-Packard Research Laboratories, Bristol, England, 1997.

[3] R. Das, J. E. Hanson, J. O. Kephart, and G. Tesauro. Agent-human interactions in the continuous double auction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Seattle U.S.A., August 2001.

[4] I. Erev and A. E. Roth. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *The American Economic Review*, 88(4):848–881, September 1998.

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

[6] S. Gjerstad and J. Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22:1–29, 1998.

[7] D. K. Gode and S. Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, 1993.

[8] J. Niu, K. Cai, E. Gerding, P. McBurney, and S. Parsons. Characterizing effective auction mechanisms: Insights from the 2007 TAC Mechanism Design Competition. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*, Estoril, Portugal, May 2008.

[9] D. Pardoe and P. Stone. Developing adaptive auction mechanisms. *ACM SIGecom Exchanges*, 5(3):1–10, April 2005.

[10] S. Phelps, S. Parsons, E. Sklar, and P. McBurney. Using genetic programming to optimise pricing rules for a double auction market. In *Proceedings of the Workshop on Agents for Electronic Commerce,*, Pittsburgh, PA, 2003.