An Agent-based Commodity Trading Simulation

Shih-Fen Cheng and **Yee Pin Lim**
School of Information Systems
Singapore Management University
{sfcheng, sharonlim}@smu.edu.sg

Abstract

In this paper, an event-centric commodity trading simulation powered by the multiagent framework is presented. The purpose of this simulation platform is for training novice traders. The simulation is progressed by announcing news events that affect various aspects of the commodity supply chain. Upon receiving these events, market agents that play the roles of producers, consumers, and speculators would adjust their views on the market and act accordingly. Their actions would be based on their roles and also their private information, and collectively they shape the market dynamics. This simulation has been effectively deployed for several training sessions. We will present the underlying technologies that are employed and discuss the practical significance of such platform.

Introduction

Commodity trading is probably one of the most ancient economic activity, and the spot markets for commodities are in existence since the dawn of the human history. Over the centuries, the scope for the so-called "commodity" has grown from agricultural commodities to include metals and energy; in recent years, even "virtual entities" like carbon credits for emission are considered as commodities and are traded actively just like traditionally defined commodities. The nature of trading has also evolved from primitive *barter* exchange (direct exchange of goods or services without monetary instrument) to more sophisticated *forward* contracting between producers and consumers (agreement to buy or sell at a fixed price at a future period), to formal Futures exchanges with clearing houses guaranteeing the transactions. Since the scope of the commodities is casted so wide, and the available financial instruments are so rich, managing the trade of commodities effectively has become more and more important but also extremely challenging.

What makes trading in commodity markets unique and challenging, despite its similarity to equity and bond markets, is the physical transactions that are behind all the financial trades in any form of commodity market. Although the volume in the financial trades (commodity derivatives) has already overtaken the physical trades, physical transactions are still critically important. This is because the balance of supply and demand and the resulting *spot prices* in the physical transactions are still the fundamental forces that are behind the commodity market, and no matter how sophisticated the used financial instruments are, all of them still need to closely reference these spot prices.

This is why trading in commodity market is challenging: physical transactions are affected by all the physical elements that link together the supply and demand sides. For example, transportation disruption or freight rate changes will propagate through the supply chain and generate regional imbalances in supply or demand; the resulting impact could then be felt in all the related industries and commodity classes. Other factors, like new legislations, abnormal weather, or even political events could also exert complicated and significant impacts on the commodity market. Therefore, to trade successfully in a particular commodity market, one need to be very familiar with the physical properties and the supply chain of that commodity. These requirements are the primary barriers in training successful commodity traders. On the other hand, these sophisticated requirements probably also help to explain why fully automated trading has not taken over the commodity trading yet.

This is what motivates our research in the commodity trading simulation. On one hand, we would like to create a commodity trading simulation that is realistic enough so that novice commodity traders could be trained effectively. On the other hand, we are also interested in studying human trader's trading behaviors in the face of complicated environment, with the ultimate goal of making software agents trade just as human traders in the commodity market.

Despite the fact that there is a vast amount of literature in economics and finance on commodity price modeling, we find them not suitable for our purposes. One of the major missing features in these models is the link between physical events and the price dynamics. These links are important because one of the highly valued skills in trading commodity is the correct readings of the physical events and also the ability to carry out appropriate trades with these understandings. To address this need, we thus propose an *event-centric* simulation model in which the price dynamics is created by a series of user-defined events. By allowing events to be user-defined, we also grant ourselves the ability of creating scenarios that are rarely seen but important, e.g., the recent commodity boom and the subsequent market crash.

To create an event-centric commodity trading simulation, we adopt a constructive approach which is widely studied by researchers in the area of agent-based computational economics. Stated conceptually, our idea is to introduce multiple *market agents* with different physical roles into the system. When an event is announced in the simulation, each agent would trade depending on its role, private information, and also the event properties. The market dynamics would then be shaped by their joint actions. As demonstrated in our experiment, it is shown that complicated price dynamics could be generated with fairly simple agent strategies.

System Overview

There are three important components in our commodity trading simulation (see Figure 1): (a) human traders; (b) market server (servicing market mechanism and dispatching events); (c) market agents (including hedgers and speculators, which will be described in detail later).

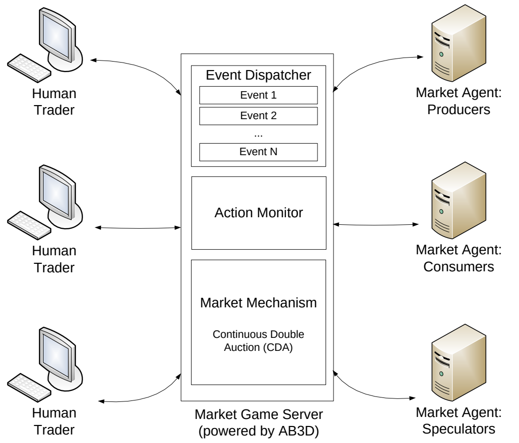


Figure 1: System architecture of the simulation.

Since one of our design goals is to provide an intuitive and straightforward trading simulation for training novices, we have proposed a highly simplified scenario that has only one futures market for some commodity (the exact type of the commodity, as we demonstrate later, can be specified by the user). Although the spot market is purposely hidden from the human traders, it is included in our consideration when creating market agents. To streamline the trading, we drop all the tedious steps in finalizing a transaction and assume that the matchings of all transactions are instantaneous without default and are handled by a standard Continuous Double Auction (CDA). In most simulations, human participants are assumed to be pure speculators that trade only for profit. For simplicity, we assume that for now we will ignore the daily settlement of futures contracts. This implies that *margin calls* will not be modeled and the cash flows will be computed only when a transaction is made (establish or close out certain position). However, we do require that agents not exceeding their position limits at all times.

The market mechanism, event dispatcher, action monitor, and all the required communication infrastructures are developed based on the AB3D (Lochner, Cheng, and Wellman 2007), a generic market game server. The list of events is predetermined by the scenario designer, where each event is defined by the following parameters (a sample event can be seen in Figure 2):

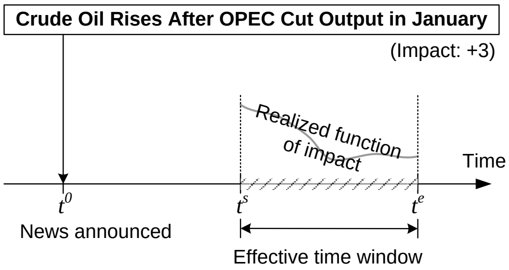

Crude Oil Rises After OPEC Cut Output in January
(Impact: +3)
Realized function of impact
Time
t^0
News announced
t^s
t^e
Effective time window

Figure 2: A typical event with all the important parameters.

Title and content: This information provides qualitative event information and is mainly for human traders.

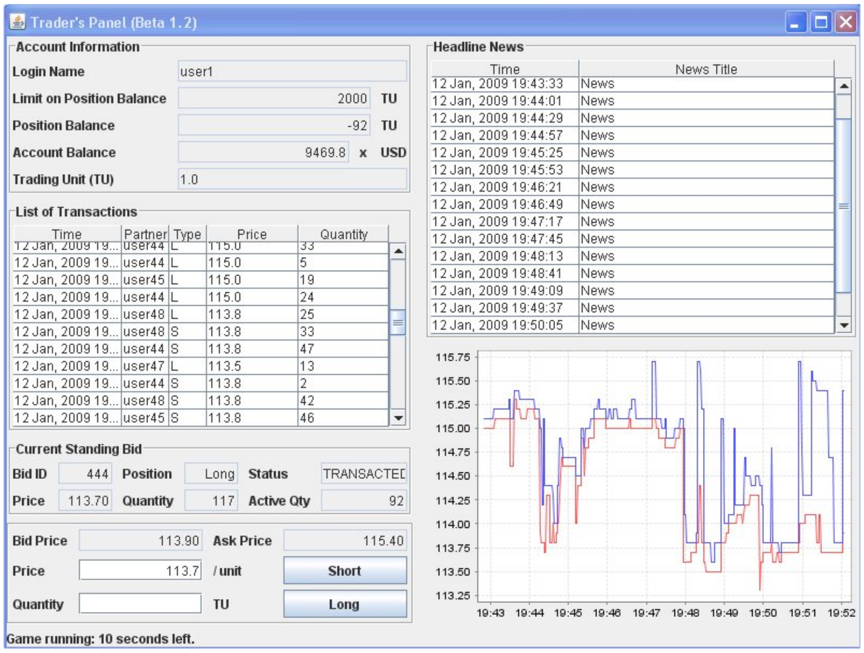
Arrival time: The time (t^0) when the event is delivered and visible to all agents (both human traders and market agents). It is assumed that all agents receive event at the same time, without discrimination.

Impact: This parameter specifies the type and the strength of the event. The strength of an event is specified by an integer in $[1, 5]$, where 5 indicates the strongest event and 1 indicates the weakest event. An event could be either bullish or bearish, and is indicated by the sign of the impact. Events with positive and negative impacts would be bullish and bearish respectively. The 'realized function of impact" is the unique response each market agent has to this event.

Effective time window: The event is only effective within the time window $[t^s, t^e]$. With these two parameters we could create events with short-term or long-term impacts. Also, by overlapping a series of events we could model the escalation of a major event (e.g., the impact on crude oil price exerted by the progression of the war in Iraq).

Note that both "impact" and "effective time window" are not revealed to the human traders and only the market agents can utilize these parameters. Besides qualitative event information, human traders also have access to the latest market information in the form of price quotes (however, they cannot peek into the order book). Both ask and bid quotes are provided to them in real-time.

By hiding the impact and the exact time when an event would be effective, human traders will be exposed to an uncertain environment similar to the real market, in which they have to estimate these two parameters solely by the qualitative content of the event. Of course, the realism of the simulation will be highly dependent on the behaviors of the market agents, which are elaborated in the next section.



Account Information

Login Name	user1	
Limit on Position Balance	2000	TU
Position Balance	-92	TU
Account Balance	9469.8	x USD
Trading Unit (TU)	1.0	

List of Transactions

Time	Partner	Type	Price	Quantity
12 Jan, 2009 19...	user44	L	115.0	33
12 Jan, 2009 19...	user44	L	115.0	5
12 Jan, 2009 19...	user45	L	115.0	19
12 Jan, 2009 19...	user44	L	115.0	24
12 Jan, 2009 19...	user48	L	113.8	25
12 Jan, 2009 19...	user48	S	113.8	33
12 Jan, 2009 19...	user44	S	113.8	47
12 Jan, 2009 19...	user47	L	113.5	13
12 Jan, 2009 19...	user44	S	113.8	2
12 Jan, 2009 19...	user48	S	113.8	42
12 Jan, 2009 19...	user45	S	113.8	46

Current Standing Bid

Bid ID	444	Position	Long	Status	TRANSACTED
Price	113.70	Quantity	117	Active Qty	92

Bid Price	113.90	Ask Price	115.40
Price	113.7	/ unit	Short
Quantity		TU	Long

Game running: 10 seconds left.

Headline News

Time	News Title
12 Jan, 2009 19:43:33	News
12 Jan, 2009 19:44:01	News
12 Jan, 2009 19:44:29	News
12 Jan, 2009 19:44:57	News
12 Jan, 2009 19:45:25	News
12 Jan, 2009 19:45:53	News
12 Jan, 2009 19:46:21	News
12 Jan, 2009 19:46:49	News
12 Jan, 2009 19:47:17	News
12 Jan, 2009 19:47:45	News
12 Jan, 2009 19:48:13	News
12 Jan, 2009 19:48:41	News
12 Jan, 2009 19:49:09	News
12 Jan, 2009 19:49:37	News
12 Jan, 2009 19:50:05	News

Figure 3: Human trader's interface.

The human trader's interface could be seen in Figure 3. As illustrated in Figure 3, the trading interface contains six major components: 1) account information; 2) list of transactions; 3) current standing bid; 4) bid panel; 5) list of events; 6) price chart. For the human trader, this interface provides access to both event-related information (double-clicking on any event would open up a pop-up window containing previously mentioned information) and also market information (bid and ask prices, and also the status of current valid bid submitted by this trader). At any given moment, long or short bid could be issued by clicking either "Long" or "Short" button, at the price and the quantity specified by the trader. Regardless of the status of the current bid, it will always be replaced by the newly submitted bid. Note that although the prices specified by the user's bid and reported by the market are in physical units (e.g., barrels for crude oil and bushels for grains), traders cannot trade in physical units directly; instead, traders have to trade in standard commodity contracts that contain different numbers of physical units for different commodity types. For example, one crude oil contract usually contains 1,000 barrels and one grain contract usually contains 5,000 bushels. In our trading simulation, the "Trading Unit" (TU), which represents the contract size, is introduced to accommodate different commodities. In the above examples, TU is set to 1,000 and 5,000 respectively for crude oil and grain.

Besides the purpose of trading, this interface is also used as an action monitor. Any action performed in this interface (viewing an event's detail or sending out a trade command) would be recorded and sent to the server for collection. This information will be available at the end of the simulation for further analysis (an example of such analysis is presented in the later section).

As hinted in the earlier discussion, our aim is to provide a completely flexible trading simulation. By modifying TU and also defining new stream of events, we could design trading scenarios for a wide variety of commodities.

Designing Market Agents

As in most other training-oriented trading simulations, fidelity and realism are some of the most important features we would like to achieve. To create simulations with high fidelity, a number of academic and commercial applications has deliberately created a linkage between trading simulations and the real market (e.g., see UMOO at `http://www.umoo.com/` and FACTSim at `http://www.factsim.org/`). In these cases, real-time market data feeds are what constitute the market dynamics. However, this type of design is not suitable for our purpose because an user-composed sequence of events is what drives the price movement. To realize our proposed model, we need to describe the occurred events quantitatively and to come up with the reaction model (to these events) for different market participants. More formally speaking, we break down the market by introducing independent agents as important market participants, embed appropriate trading strategy into each agent, and then let these agents interact in order to create market dynamics collectively. Therefore, the design of market agents is the most critical part of our commodity trading simulation.

Using agents in modeling complex economic or financial systems is not new, in fact, a large number of literature has been devote to the subject of "Agent-based Computational Economics" (ACE) (LeBaron 2001; Tesfatsion 2002; 2006). The ACE is probably best explained in Tesfatsion's own words (Tesfatsion 2006):

> The defining characteristic of ACE models is their constructive grounding in the interactions of agents, ... Starting from an initially specified system state, the motion of the state through time is determined by endogenously generated agent interactions.

Our model follows similar constructive principle, with the introduction of events as a way to guide agents' actions.

In our market model, we place the modeling emphasis on hedgers and speculators. Their internal respond models, which take event occurrences and market states as inputs, will determine how the market evolves (of course, actions from human traders will also influence the market evolution). The roles and the models of hedgers and speculators will be explained in detail in the following two subsections.

It should be noted that the framework in which we develop our market agents is quite open and flexible. Therefore if necessary, ourselves or any third-party could easily develop new types of agents and add them to the mix as appropriate.

Hedger Model

Hedgers are the original users of the futures market. They are usually producers or consumers of the commodity who would like to lock in at some specific prices and quantities well before the time of production (for producers) or usage (for consumers). These producers and consumers provide basic liquidity and are the main drivers of the supply and demand in the market.

To properly incorporate producers and consumers in our model, we assume that they exhibit stationary behaviors, i.e., the rate of their production and consumption will be stationary. Since we assume that only one futures market exists for this commodity, this assumption implies that all producers and consumers have to constantly establish new hedges in this market, and their collective actions will create the market dynamics accordingly. We further assume that all producers and consumers will employ a simple *hedge-and-forget* strategy, meaning that they will establish new hedges based on their own needs (new productions or usages), the current market condition, and their expectation; but once the hedges are established, they will hold them to the end (in other words, no *dynamic hedge* will be considered).

To construct market agents based on these simplifications, we assume that at a given interval (which could be stochastic and agent-specific), a new long (short) bid will be issued by the consumer (producer) agent. The quantity of the bid will be randomly drawn from a uniform distribution $U[Q_{\min}, Q_{\max}]$, in which $Q_{\min}$ and $Q_{\max}$ represent lower and upper bounds on agent's capacity, and again these two parameters are agent-specific. By manipulating $Q_{\min}$ and $Q_{\max}$ across all market agents, human traders could experience different level of market power (larger Q values correspond to smaller market power, and vice versa).

The price of the bid will depend on individual market agent's latest price forecast, which is computed based on the recent price trend, current spot price (it's assumed to be internal to the agent and not explicitly modeled), and also the estimated impact of the latest event. We first assume that a perfect forecast can be made and derive the perfect forecast. We will then discuss how agents could approximate this perfect forecast.

Commodity prices tend to be mean-reverting at the level of marginal cost of production. This has been shown to be true in a wide variety of commodities (e.g., see Geman and Nguyen (2005)). To model commodity prices, we thus have to look for models that are mean-reverting. One such model is the Ornstein-Uhlenbeck (OU) process, which is originally proposed by Vasicek (1977) for modeling interest rates. Because of its mean-reverting feature, the OU process has been a popular choice for modeling commodity prices in recent years. Besides mean-reversion, events also play an important role in our simulation. To incorporate the event impacts into the model, we follow the *jump diffusion model* proposed by Merton (1976). Finally, to properly implement the resulting price evolution model in our simulation, we prefer a discrete-time model (as opposed to the original continuous-time model). Our choice is a discrete-time variant described by Blanco and Soronow (2001).

In our simulation, the price forecast at time t is determined by:

$$P_f(t) = P_e(t) + n(t) + \epsilon(t),$$

where $P_e(t)$ is the equilibrium price derived from the market price information, and it follows the *mean reversion* model:

$$P_e(t) = \mu + \lambda(\frac{P_a(t) + P_b(t)}{2} - \mu).$$

μ is the mean price modeled in the simulation, λ is the weight for mean correction, and the current market price is estimated as the average of $P_b(t)$ (bid price, which is the price of the highest untransacted long bid) and $P_a(t)$ (ask price, which is the price of the lowest untransacted short bid). $n(t)$ is the estimated impact of the most recent event at time t, and is both event and agent specific. $n(t)$ could be defined as arbitrary functions and its simplest form (the one we adopted) exerts constant impact within an event's effective time range (i.e., $n(t) = c, t_0 \leq t \leq t_1$, and $n(t) = 0$ otherwise). $\epsilon(t)$ is the white noise (follows $N(0,1)$, the standard normal distribution).

There is a long list of reasons why such perfect prediction could not be obtained for all market agents. It could be that some agents have limited capabilities, both in terms of modeling or information acquisition. It could also be that the established hedge is not perfect (i.e., the desired maturity date and the commodity category don't match futures contract exactly). In our simulation, such approximation on the agent side is accounted for by introducing randomized bidding within the range of the perfect price prediction and the market price. Since $P_a(t)$ and $P_b(t)$ are deemed as the commodity market price by the producers and consumers respectively, the prices submitted by producers are uniformly distributed in $[\min\{P_f(t), P_a(t)\}, \max\{P_f(t), P_a(t)\}]$, while the prices submitted by consumers are uniformly distributed in $[\min\{P_f(t), P_b(t)\}, \max\{P_f(t), P_b(t)\}]$.

Speculator

While "mean-reverting" hedgers constitute the "fundamental" part of the simulated market, most of the liquidity and volatility, on the other hand, is generated by the speculator agents. In our simulation, we adopt the classical *zero intelligence* (ZI) strategy (Gode and Sunder 1993) in constructing our speculator agent. However, to prevent ZI agents from destroying the market trend (generated by producer and consumer agents), we limit the price range to be $[P_b(t) - \delta, P_a(t) + \delta]$, where $P_b(t)$ and $P_a(t)$ are bid and ask prices as defined previously, and δ is an agent parameter, controlling how aggressive this ZI agent should be in creating volatility. In most of our simulations, we simply set δ to be the bid increment.

After the price is randomly decided, the ZI agent will choose to take either long or short positions with equal probability. Since the ZI agent us constrained by a given position limit, it will randomly decide how much remaining position allowance it would devote to the new trade. As speculators, ZI agents are required to exit all positions at the end, and they are programmed to gradually exit their positions when the end draws near.

Validating Event Dynamics

The multiagent model presented in the previous section is pretty general and can be used in a wide variety of scenarios. As long as the scenario designer can come up with a list of events that follow the event specification, the above multiagent model could then generate the desired market dynamics.

The challenge here is: how do we know whether the generated market dynamics is what should be expected from the designed scenario? This is not a straightforward task since the scenario we plan to execute might be completely fictitious and it does not need to have a real-world counterpart. Without benchmark data, establishing the credibility of the simulation would not be easy. Of course, if the generated market dynamics together with the market scenario is carefully reviewed by a commodity expert, we probably could come up with an assessment qualitatively. However, this would not be feasible if we plan for large-scale and frequent deployments. Therefore, we need some method that is quantitative and could be automated.

Fortunately, this is the type of question economists are repeatedly asked in various settings. And in accounting and financial economics, researchers have long been studying statistical approaches that can measure the effects of an economic event. This type of approaches is called *event studies*. There are many variants of event studies, and we will adopt the version proposed by MacKinlay (1997). According to MacKinlay, the procedure for an event study is as follows (some steps not necessary for our application are dropped):

- Define the event of interest and identify the event window. In our case, the event of interest and its occurrence is straightforward to define, and we will take 20 days before and after the event occurrence as the event window.
- Measure the abnormal return (AR), which is the actual return minus the normal return of the commodity over the event window. Normal return could be obtained by simply assuming *constant mean return model*. In the commodity market setting, it refers to the mean price of the commodity from the beginning of the horizon to just before the beginning of the event window..
- From AR, compute cumulative abnormal return (CAR) to draw overall inferences for the event of interest.
- Define null hypothesis H_0 and perform statistical tests over multiple sample instances. For all events, the null hypothesis H_0 can be defined as $\overline{CAR} = 0$. As for the alternative hypothesis, H_1, it can be define as $\overline{CAR} > 0$ for bullish events, $\overline{CAR} < 0$ for bearish events, and $\overline{CAR} \neq 0$ for neutral events (or no event).

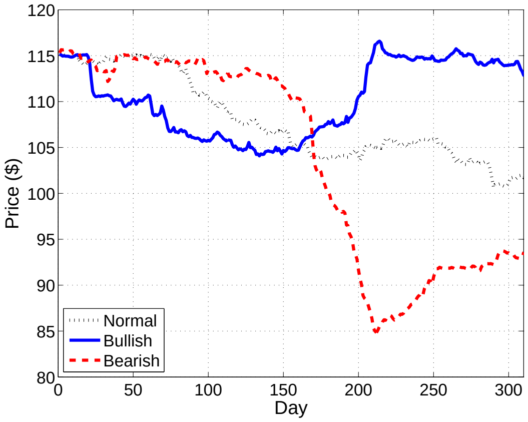


Figure 4: A sample price dynamics of the one-event scenario. The event is announced at around day 160.

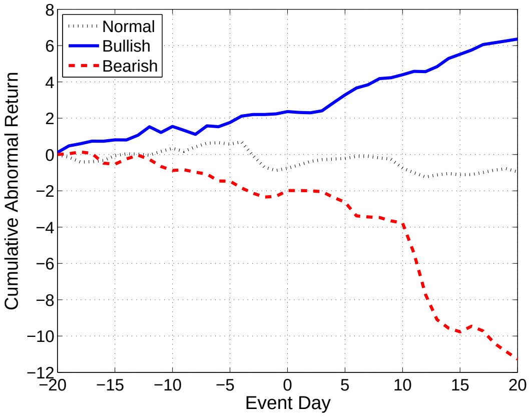


Figure 5: The cumulative abnormal returns (CAR) for bullish, bearish, and normal scenarios.

To validate that our multiagent model indeed creates detectable market dynamics in response to events, we create a special scenario with only one event (a sample price evolution of this scenario is shown in Figure 4). For the market agents, we include 12 producers, 13 consumers, and no speculator (to avoid introducing unnecessary noise). To collect enough sample data points, the same scenario is executed 15 times. Following the above procedures, we test the null hypothesis with three CAR series (one of the sample is shown in Figure 5). For both the bullish and the bearish cases, the p-values $\sim$ 0, implying that strong positive/negative abnormal returns are significant. For the normal (no-event) case, p-value $\sim$ 0.065, indicating that no abnormal return is detected in the event window.

Compared to the case of validating event *occurrence*, validating the *strength* of an event is much more difficult. This is because the absolute level of response that should be triggered by an event cannot be determined straightforwardly. Therefore, instead of trying to validate the absolute response strength, we choose to validate the relative response strengths. The objective of this is to ensure that higher impact levels indeed generate larger market responses when compared to events with lower levels. To establish this, we could simply perform statistical comparisons between adjacent strength levels (i.e., compare levels 1 and 2, 2 and 3, and so on). With this validation, we could at least be assured of the consistency in market responses throughout the simulation.

Case Study: A Crude Oil Trading Simulation

The prototype of the described system has been deployed for a number of trading courses since April, 2007. A recent deployment at one such course on crude oil trading is described here as an illustrative example.

The selected trading course was designed to give undergraduate business students an overview on crude oil trading. The course is 8-week long and it covers all the fundamental topics related to the oil industry. Since the primary objective of the course is to introduce students to the career as an oil trader, significant amount of time has been spent on topics related to effective oil trading. This course was concluded with a trading simulation powered by our prototype system.

Background

Each participant in the simulation plays the role of a speculator and trades for profit. The targeted commodity is the Western Texas Intermediate (WTI) and each trader can trade up to a maximum of 2,000,000 barrels (bbls). A trader can take either long or short positions, but once he reaches his trading limit, he cannot take further positions in the same direction (however, he could still trade to reduce his positions). To encourage participation, we explicitly asked participants to trade at least 100,000 bbls per trade and to transact at least 2,000,000 bbls during the simulation. The contract size is 1,000 bbls, thus the trading unit (TU) of this scenario is set to be 1,000.

To exaggerate profits and losses and to encourage participants to be decisive, the prices are set in the interval of 10 cents. Initial bid and ask prices are set to be $50.00/bbl and $50.10/bbl respectively.

The scenario is designed to have 48 events, with event intervals set to be somewhere between 1 to 2 minutes. Each event begins to affect the market roughly 20 seconds after its announcement. The resulting length of the simulation is thus around 72 minutes. All traders are asked to return their position balance to zero at the end of the simulation. They will be penalized with the lowest possible mark-to-market values if they hold any position at the end of the simulation.

The flow of events are composed by the course instructor. The expected price trend that ought to be generated by the flow of events is plotted in Figure 6. Note that no hint of any form is given to participants regarding the price trend. Participants are only briefed on the *initial market conditions* before the simulation.

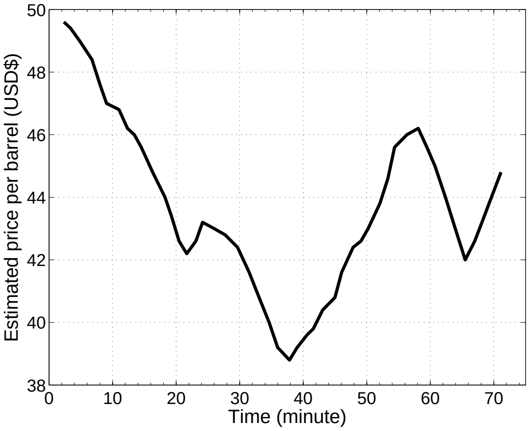


Figure 6: The expected market price evolution.

Summary and Analysis

40 students participated in this trading simulation and most of them ended up losing money. In fact, the average loss was 21 million dollars and only 8 out of 40 ended up earning profits. Since simulation session like this one is meant for training novices, participants must be given a chance to review their trading strategies after the simulation so that they can be debriefed on what might have gone wrong. After the simulation, each participant could review his trading history and also the trading chart characterizing his trading strategy. This trading chart is composed of three important time series: (a) expected market prices (to be generated by the flow of events), (b) observed price quotes, and (c) player's position balance. (c) is the realization of the trading strategy, and it might take (a) and/or (b) as inputs. In some cases, such connections are obvious; e.g., the trading chart for one of the best player is plotted in Figure 7, and it apparently follows the trend hinted by both the event disclosure and market price. In some other cases, completely wrong interpretation of the news events could happen, and in the extreme case, some traders might even choose to bet arbitrarily disregarding the market trend. Figure 8 shows trading chart for one such player, and not surprisingly, he is also one of the worst traders in the simulation.

It should be emphasized that we do not intend to offer a comprehensive analysis on all trading strategies. Instead, we present these cases so that we could illustrate the potential of our system in helping individuals spotting flaws in their strategies. Such observations would be extremely valuable in improving their trading skills.

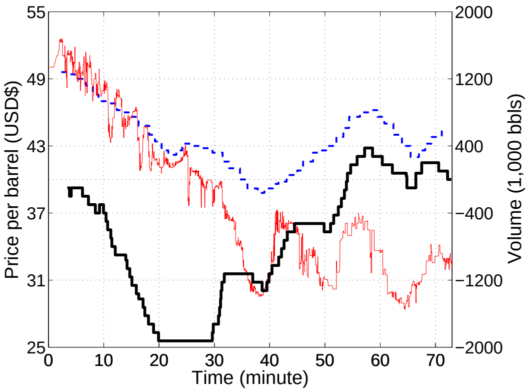


Figure 7: The best trader in the simulation: a trend follower. Thick line is player's position balance; thin line is real market price; dotted line is estimated market price dynamics.

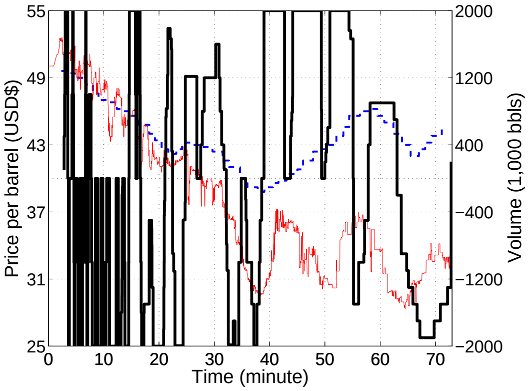


Figure 8: The worst trader in the simulation: no discernable trading strategy. See Figure 7 for legend.

Discussion

In this paper, we present our efforts in building an agent-based commodity trading simulation. The simulation scenario progresses by user-defined news events. Our framework is based on the constructive principle widely applied in the agent-based computational economics community, and the actors in our market are based on well-studied theoretical models. The fidelity of the market dynamics generated by the multiagent framework is statistically provable by using the event study method.

Our aim in this research is not to create new pricing models for commodities; instead, we have focused on how to construct a highly realistic commodity trading simulator from the building blocks that are well-founded theoretically. The primary application of such framework is currently in training human traders, and as demonstrated by a number of deployments, such exercises are highly valuable for both participants and us. The data that is collected during the simulation is extremely valuable in helping us better understand what makes a good trader. On one hand, these analysis results could be used to improve the market agents employed in our simulation. On the other hand, these analyses could also help educators in designing better educational programs that could more effectively teach the art of trading. Ultimately, our platform might one-day be used in benchmarking or diagnosing a trader's skill.

We do notice that a large number of systems have already been developed for the purpose of simulating commodity trading. However, to the best of our knowledge, none of them is truly flexible in terms of specifying scenario content (e.g., deciding properties of the target commodity and flow of events) and also the way the market dynamics is generated (in our system it could be changed if new market agents are introduced). This flexibility, granted both by the event-centric design and the multiagent-based approach, is our contribution to the area of commodity trading simulation.

Acknowledgements

This work was supported in part by the International Enterprise Singapore and the International Trading Institute at Singapore Management University. The authors would like to acknowledge the dedicated effort by Chao-Chi Liu in building the simulation infrastructure. The authors would also like to thank Samuel Owens and Anil Shamdasani for developing the crude oil and the soybean trading scenarios.

References

Blanco, C., and Soronow, D. 2001. Mean reverting processes — Energy price processes used for derivatives pricing & risk management. *Commodity Now* 68–72.

Geman, H., and Nguyen, V.-N. 2005. Soybean inventory and forward curve dynamics. *Management Science* 51(7):1076–1091.

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

LeBaron, B. 2001. A builder's guide to agent-based financial markets. *Quantitative Finance* 1:254–261.

Lochner, K. M.; Cheng, S.-F.; and Wellman, M. P. 2007. AB3D: A market game platform based on flexible specification of auction mechanisms.

MacKinlay, A. C. 1997. Event studies in economics and finance. *Journal of Economic Literature* 35(1):13–39.

Merton, R. C. 1976. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics* 3:125–144.

Tesfatsion, L. 2002. Agent-based computational economics: Growing economies from the bottom up. *Artificial Life* 8(1):55–82.

Tesfatsion, L. 2006. Agent-based computational economics: A constructive approach to economic theory. In Tesfatsion, L., and Judd, K. L., eds., *Handbook of Computational Economics, Volume 2: Agent-based Computational Economics*. Elsevier.

Vasicek, O. 1977. An equilibrium characterization of the term structure. *Journal of Financial Econometrics* 5(2):177–188.