CCFEA Working paper WP038-10, revised November 2012;
further revised version to appear in Quantitative Finance, 2013

# A Formalization of Double Auction Market Dynamics

Edward Tsang
*Centre for Computational Finance and Economic Agents (CCFEA), University of Essex*

Richard Olsen
*Olsen Ltd. and Centre for Computational Finance and Economic Agents (CCFEA), University of Essex*

Shaimaa Masry
*Centre for Computational Finance and Economic Agents (CCFEA), University of Essex*

Correspondence:
Edward Tsang, CCFEA, University of Essex, Colchester, Essex, CO4 3SQ UK;
edward@essex.ac.uk

Biographical notes on contributors:

**Edward Tsang** has a first degree in Business Administration (Major in Finance) and a PhD in Computer Science. He has broad interest in applied artificial intelligence, in particularly computational finance, heuristic search, constraint satisfaction and scheduling. He is currently a professor in computer science at the University of Essex where he leads the Computational Finance Group and Constraint Satisfaction and Optimization Group. He is also the Director of the Centre for Computational Finance and Economic Agents (CCFEA), an interdisciplinary centre. He founded and chaired the Technical Committee for Computational Finance under the IEEE Computational Intelligence Society in 2004-2005.

**Richard Olsen** has a Master in Economics from Oxford University and a PhD in law from the University of Zurich. He has specialized in high frequency finance and has been a pioneer of this discipline. In 1995, he co-organized the first conference in the field. In 2001, he and his team published a book, 'Introduction to High Frequency Finance', Academic Press. He is CEO of Olsen Ltd, a systematic asset management company based in Zurich and co-founder of OANDA, an Internet market maker of foreign exchange. He is Visiting Professor at the Centre for Computational Finance and Economic Agents (CCFEA) since 2007.

**Shaimaa Masry** has a first degree in Management Information Systems and an MBA (Major IT). She is currently a PhD student in Computational Finance at the Centre for Computational Finance and Economic Agents (CCFEA) at University of Essex. She is working on the High Frequency Finance Project. She has wide interest in computational finance, artificial intelligence and business intelligence.

Submission to Quantitative Finance, 20 November 2012

# A Formalization of Double Auction Market Dynamics

To understand financial markets and prevent crisis we need to analyze market microstructure. This paper formalizes the market process in the context of a simple double auction market. The purpose of this calculus is to analyze market dynamics and feedback loops of for example cascading margin calls with the objective to get a better understanding of risk scenarios, not to forecast exogenous order flow. The price trajectory is determined by the present market state and new orders arriving in the market. By studying the market microstructure, we can compute the impact of an order of any size, or how big a sell order has to be to cause the market to fall by a certain percentage. Using a definite formalism reduces ambiguity and enables rigorous reasoning. An algorithm for assessing risk is proposed. Real markets are more complex than the models presented in this paper and this paper is a step towards building a solid foundation for studying market models.

Keywords: high-frequency Finance, financial markets.

## 1. Introduction

Financial markets are complex. Classical economics have been under serious challenge (e.g. see Olsen, 2005; Shleifer, 2000) to explain price action and volume flows in financial markets. One novel approach to market studies is to model the micro-behaviour of markets (Shleifer, 2000; Solomon, et al., 2000). The attempt is to observe micro-behaviour in the market with the aim to discover general dynamics (Acerbi and Scandolo, 2008; Glattfelder, et al., 2008). This approach is data-driven. Unlike classical economics, it does not depend on stringent assumptions, such as perfect rationality by the traders (Tsang, 2008). This new approach is still in its infancy. This paper looks at simple market models, and attempts to define the market dynamics formally. The intended contribution of this paper is not in modelling micro-behaviour, but in formalizing such models and analyzing their properties, to examine what can be usefully inferred from market information.

The market can be described by states. The state of the market can be changed by events. In this paper, we limit our attention to buy and sell events initiated by

market participants. Even though behaviour of the market participants may in general be unpredictable, certain inferences can be made. Given a set of buy and sell orders, the calculus can define state transitions. We can make an analogy with weather forecasts, where we may not know the long term weather changes, but we can predict the immediate future given the current state; e.g. air flows from high pressure to low pressure regions.

Event calculus is useful for reasoning (Chen, 2009; Kowalski and Sergot, 1986; Mueller, 2009). Shanahan states: *"The event calculus is a logical mechanism that infers what's true when given what happens when and what actions do"* (Shanahan, 1999). Although we have not adopted conventional event calculus, this paper formalises the components relevant to the calculus of market transitions. It highlights the fact that the consequences of an order (orders are the only events considered in this paper) can be complex: the consequences are dependent on the positions and margins held by market participants. With this analysis, one can determine, for example, how big orders need to be to cause market crashes.

This paper formalises the obvious. But it is better to state the obvious with mathematical rigor rather than allowing ambiguity, which needs repeated clarification later in our research. Besides, what is obvious to some may not be obvious to others. Stating the obvious through a formal description enables us to study micro-behaviour rigorously.

## 2. Market Models

### *2.1 Model 1*

This model is defined under a double auction market.

$$\text{State} + \text{Orders} \rightarrow \text{State}$$

Where:

$$\text{State} = \text{Queue\_Profile} = (\text{Bid\_Queue}, \text{Offer\_Queue})$$

$Bid\_Queue = ((order_1, price_1, volume_1), (order_2, price_2, volume_2),$
$\ldots, (order_n, price_{bq}, volume_{bq}))$
$Where\ price_1 > price_2 > \ldots price_{bq}$
$Offer\_Queue = ((order_1, price_1, volume_1), (order_2, price_2, volume_2),$
$\ldots, (order_n, price_{oq}, volume_{oq}))$
$Where\ price_1 < price_2 < \ldots price_{oq}$

The Bid_Queue comprises the bids to buy. The Offer_Queue comprises offers to sell. Buy (sell) orders having the same price are not merged.

Orders refer to a sequence of orders, where each order is either a bid or an offer, together with its volume.

$Orders = (Order_1, Order_2, \ldots, Order_n)$

We assume that the orders are processed in sequence:

$State + (Order_1, Order_2, \ldots, Order_n) \rightarrow (State + Order_1) + (Order_2, \ldots, Order_n)$

For simplicity, we assume only two types of orders. A market order is to buy or sell at the market price. A limit order is to buy a certain volume up to a price specified, or to sell a certain volume above a price specified. For notional convenience, we write a market buy order as a limit buy order with the price set at infinity; a market sell order sets its price to minus infinity.

$Order = (Order\_No, Order\_Type, Price, Volume)$
$Order\_Type = bid \mid offer$
$Order\_No = O_i$

We define a symbol Inf, which stands for both infinity and minus infinity. We write a market buy order as (buy, Inf, Volume), a market sell order as (sell, Inf, Volume).

The calculus for clearance of a limit sell order can be defined below.

Let $Bid\_Queue_1 = ((O_1, P_1, V_1), (O_2, P_2, V_2) \ldots)$
$Offer\_Queue_1 = ((O_3, P_3, V_3), (O_4, P_4, V_4), \ldots)$
$Limit\_Order = (O_n, sell, P_n, V_n)$.

The calculus for a limit order is very simple. If the price of the sell order is less than or equal to at least the bid order at the head of the bid queue, the limit order can be fully or partially fulfilled. The sell order of volume $V_n$ removes from the head of the

Bid_Queue ($P_1$, $V_1$) the minimum of $V_n$ or $V_1$. If $V_n$ is greater than $V_1$, then the head of the Bid_Queue is removed. If the limit price is reached, clearing stops and the remaining unfulfilled sell order joins the offer queue. If the limit price is not yet reached, clearing continues with the remaining Bid_Queue until $V_n$ is reduced to 0. If the price of the limit sell order is larger than the first bid order in the bid queue, then the sell limit order joins the offer queue. This can be formalised as follows.

$$(((O_1, P_1, V_1), (O_2, P_2, V_2),\ldots), ((O_3, P_3, V_3), (O_4, P_4, V_4), \ldots)) + (O_n, \text{sell}, P_n, V_n) \rightarrow$$
$$(((O_1, P_1, V_1), (O_2, P_2, V_2),\ldots), ((O_3, P_3, V_3), (O_4, P_4, V_4), \ldots) \oplus (O_n, \text{sell}, P_n, V_n)) \quad \text{if } P_1 < P_n$$
$$(((O_1, P_1, V_1 - (\min(V_1, V_n)), (O_2, P_2, V_2), \ldots), ((O_3, P_3, V_3), (O_4, P_4, V_4), \ldots)) +$$
$$(O_n, \text{sell}, P_n, V_n - (\min(V_1, V_n))); \text{Transaction Price (TP)} = P_n \text{ is defined} \quad \text{if } P_1 \geq P_n$$

The + operation is recursive when $P_1 \geq P_n$, in which case transaction takes place; it stops when $P_1 < P_n$ or $V_n$ is reduced to 0. Here $\oplus$ is the queue joining operator which simply put the orders in ascending order according to their prices.[1] Cleared orders are removed from the bid queue:

$$((O_1, P_1, 0), (O_2, P_2, V_2),\ldots) \rightarrow ((O_2, P_2, V_2), \ldots)$$

In the above rule, we highlight Transaction Price (TP) at the point where it is defined. We shall refer to it later.

Limit buy orders are handled symmetrically.

In the calculus above, the clearing of a market order is exactly the same as the limit order, except that market orders do not have limit prices and hence are always completely fulfilled as long as there are buyers (sellers). They do not join the bid or offer queues. Generally, the handling of unmatched large market orders depends on the order book configuration of the trading system.

---

[1] In functional programming convention, $\oplus$ is defined below:

$$((P_1, V_1), (P_2, V_2), \ldots) \oplus (\text{sell}, P, V) \rightarrow$$
$$((P, V), (P_1, V_1), (P_2, V_2), \ldots) \quad \text{if } P < P1$$
$$((P_1, V_1), ((P_2, V_2), \ldots) \oplus (\text{sell}, P, V))) \quad \text{if } P \geq P1$$

### *2.2 Example 1 for Model 1*

With Model 1, the calculus for computing state transition is straight-forward. This example shows the state change for a given market order.

State 1.1 = (Bid_Queue1.1, Offer_Queue1.1)
Bid_Queue1.1 = (($O_1$, 1.60, 2500), ($O_2$, 1.59, 2000), ($O_3$, 1.58, 2500), ($O_4$, 1.57, 1500), ($O_5$, 1.56, 4000))
Offer_Queue1.1 = (($O_6$, 1.61, 3000), ($O_7$, 1.62, 2000), ($O_8$, 1.63, 1500))

Let Order1.1 = ($Order_9$, $Order_{10}$, $Order_{11}$), where
$Order_9$ = ($O_9$, sell, Inf, 5000)
$Order_{10}$ = ($O_{10}$, buy, 1.57, 1000)
$Order_{11}$ = ($O_{11}$, buy, 1.62, 6000)

With $Order_9$, which is a market order, the following transactions ensue:

2500 will be transacted at 1.60

This will result in the Bid_Queue being reduced to:

($O_2$, 1.59, 2000), ($O_3$, 1.58, 2500), ($O_4$, 1.57, 1500), ($O_5$, 1.56, 4000))

Next, the following two transactions will take place:

2000 will be transacted at 1.59
500 will be transacted at 1.58

The resulting state is:

State 1.2 = (Bid_Queue1.2, Offer_Queue1.2)
Bid_Queue1.2 = (($O_3$, 1.58, 2000), ($O_4$, 1.57, 1500), ($O_5$, 1.56, 4000))
Offer_Queue1.2 = Offer_Queue1.1

With $Limit\_Order_{10}$, the offer queue is not changed as the price of the buy limit order is less than the price of the head of the offer queue. Since $Limit\_Order_{10}$ is not matched; it is added to the bid queue.

The resulting state is:

State 1.3 = (Bid_Queue1.3, Offer_Queue1.3)
Bid_Queue1.3 = (($O_3$, 1.58, 2000), ($O_4$, 1.57, 1500), ($O_{10}$, 1.57, 1000), ($O_5$, 1.56, 4000))
Offer_Queue1.3 = Offer_Queue1.2

With Limit_Order$_{11}$ (to buy 6000 with limit price 1.62), the offer queue is changed. Since the price 1.62 is greater than or equal to the first two orders in the offer queue, the following transactions will take place:

3000 will be transacted at 1.61
2000 will be transacted at 1.62

The remaining 1000 units will join the bid queue. Therefore, the resulting state is:

State 1.4 = (Bid_Queue1.4, Offer_Queue1.4)
Bid_Queue1.4 = (($O_{11}$, 1.62, 1000), ($O_3$, 1.58, 2000), ($O_4$, 1.57, 1500), ($O_{10}$, 1.57, 1000), ($O_5$, 1.56, 4000))
Offer_Queue1.4 = (($O_8$, 1.63, 1500))

### *2.4 Model 2: When Positions and Margins are considered*

The market dynamics will change when traders trade with margins. A trader with margin m, where $0<m\leq1$, will pay up only proportion m of the value that it trades. We make the following assumptions in our analysis:

**Assumption 2.1.** *For a trader with a short (long) position with margin m, its position is closed automatically when the price rises (falls) by more than m.*

For example, a trader who trades with a margin of 4% will have its short position closed automatically when the price rises by 4% or more.

**Assumption 2.2.** *All consequences of an automatic position closure take place before any new event occurs.*

Today, market orders are cleared by computer programs, which will typically handle one order at a time. A program must clearly specify how orders are processed even if they reach the computer simultaneously with parallel hardware. A calculus can be written down for every clearly defined clearing mechanism. Without loss of generality, we assume in this paper that the market clearing process cannot be interrupted. We assume that the recursive application of the rule will not be interrupted before the clearing mechanism handles new orders.

**Assumption 2.3**[2]**.** *We assume that a position cannot be adjusted and is only opened by a market or limit order. Position closure takes place automatically through margin calls. The relaxation of this assumption does not affect the generality of the results shown in our paper.*

**Assumption 2.4.** *We assume that the orders, positions and margins are available.*

Under this model, the description of a state must include traders' position profiles:

State = (Queue_Profile, Position_Profile)

Where:

Queue_Profile = (Bid_Queue, Offer_Queue)

Position_Profile = {Position | Position = (Position_Code, Position_Type, Volume, Value, Price, Margin)}

Position_No= $P(O_i)$, where $O_i$ is the Order_No of the order opening the position, given Assumption 2.3

Position_Type = long | short

Value = the value of the order(s) against which the opening position order has been matched. Given:

Bid_Queue = $((O_1, P_1, V_1), (O_2, P_2, V_2), .., (O_{n-1}, P_{n-1}, V_{n-1}))$

Order= $(O_n$, sell, Inf, $V_n)$

$P(O_n)$Value = $(P_1 * \min(V_1,V_n)) + (P_2 * \min(V_2, (V_n - \min(V_n,V_2))) + ...$

$+ (P_{n-1} * \min(V_{n-1}, V_n - \min(....)))$

Price= Unit Price= Value/Volume

The clearance calculus is exactly the same as in Model 1, except that new events, namely new orders, can be triggered by state transitions.

The last transaction price (TP) is defined by the order clearing rule described in Section 2.1. TP may trigger margin calls, which force some positions to be closed. The margin-triggered set of new orders is NO:

NO = $\{(O_i$, buy, Inf, V) | $(P(O_i)$, short, Vol, Val P, m) $\in$ Position_Profile such that $P\times (1+m) < TP\} \cup \{(O_i$, sell, Inf, V) | $(P(O_i)$, long, Vol, Val P, m) $\in$ Position_Profile such that $P\times (1-m) > TP\}$

Orders = Orders + NO

[2] In a real market, a position is constructed via a set of orders. It can be opened, adjusted and closed by market and limit orders. Position closure takes place as a result of either a margin call or the trader's decision.

Here we make no assumption on how the set of new orders (NO) join the Orders queue; i.e. the "+" operator between orders is yet to be defined. This is left to future refinement of the model.

### *2.5 Example 2 for Model 2: The effect of margin constraints*

The following example shows the state transitions and how new events (which are limited to market orders in this model) are triggered.

Let State 2.1 = ((Bid_Queue2.1, Offer_Queue2.1), Positions2.1)
BidQueue2.1 = (($O_4$, 1.60, 2500), ($O_5$, 1.59, 2000), ($O_6$, 1.58, 2500), ($O_7$, 1.57, 1500), ($O_8$, 1.56, 4000))
Offer_Queue2.1= (($O_9$, 1.61, 3000), ($O_{10}$, 1.62, 2000), ($O_{11}$, 1.63, 1500))
Positions2.1 = (($P(O_1)$, long, 4000, 6600, 1.65 4%),
($P(O_2)$, long, 2000, 3280, 1.64, 4%),
($P(O_3)$, long, 2000, 3280, 1.64, 5%))

For illustration purposes let us assume the following:

(1) The position profile (Positions2.1) represents the current positions in the market created from previous orders.

(2) Any new position in the market has a margin of 4%

(3) Only one market order in the queue:

Order 2.1 = (($O_{12}$, sell, Inf, 5000))

This is the same order that we used in Example 1. When it is cleared, as explained above, the bid queue will be changed. The state will be changed to:

State 2.2 = (Bid_Queue2.2, Offer_Queue2.2, Positions2.2)
Bid_Queue2.2 = (($O_6$, 1.58, 2000), ($O_7$, 1.57, 1500), ($O_8$, 1.56, 4000))
Offer_Quene2.2 = Offer_Queue2.1
Positions2.2 = (($P(O_1)$, long, 4000, 6600, 1.65 4%),
($P(O_2)$, long, 2000, 3280, 1.64, 4%),
($P(O_3)$, long, 2000, 3280, 1.64, 5%),
($P(O_4)$, long, 2500, 4000, 1.60, 4%),
($P(O_5)$, long, 2000, 3180, 1.59, 4%),
($P(O_6)$, long, 500, 790, 1.58, 4%),
($P(O_{12})$, short, 5000, 7970, 1.594, 4%))
Where:
$P(O_{12})$ Value = (1.6*2500)+(1.59*2000)+(1.58*500)= 7970
LastTP = 1.58 (the price of the last matched order in the Queue_Profile)

At this point, the bid queue and the position $P(O_1)$ together will trigger a new market order. This is because $1.65\times(1-4\%) = 1.584$, which is above the last transaction price, which was 1.580. Therefore, the margin is exceeded, and this position must be closed (Assumption 2.1). That means the order queue will be changed to:

Order 2.2 = (($O_{13}$, sell, Inf, 4000))

The following transactions take place:

2000 will be transacted at 1.58
1500 will be transacted at 1.57
500 will be transacted at 1.56

This will change the state to:

State 2.3 = ((Bid_Queue2.3, Offer_Queue2.3), Positions2.3)
Bid_Queue2.3 = (($O_8$, 1.56, 3500))
Offer_Quene2.3 = Offer_Queue2.2
Positions2.3 = (($P(O_2)$, long, 2000, 3280, 1.64, 4%),
($P(O_3)$, long, 2000, 3280, 1.64, 5%),
($P(O_4)$, long, 2500, 4000, 1.60, 4%),
($P(O_5)$, long, 2000, 3180, 1.59, 4%),
($P(O_6)$, long, 2500, 3950, 1.58, 4%),
($P(O_{12})$,short, 5000, 7970, 1.594, 4%),
($P(O_7)$, long, 1500, 2355, 1.57, 4%),
($P(O_8)$, long, 500, 780, 1.56, 4%))
Where:
LastTP = 1.56

Note that order $O_6$ has opened a new position $P(O_6)$ in State2.2. However, it was only partially matched. In State2.3, $O_6$ is fully matched. Thus, we do not open a new position but we update the already opened position $P(O_6)$.

The long position $P(O_2)$ must be closed when the last transaction price (1.56 in this case) falls below its margin, which is $1.64\times(1-4\%) = 1.574$. This means the order queue will be updated by the new market order:

Order 2.3 = ($O_{14}$ sell, Inf, 2000)

When the order (sell, Inf, 2000) is matched, 2000 will be transacted at 1.56. This will reduce the state to:

State 2.4 = ((Bid_Queue2.4, Offer_Queue2.4), Positions2.4)

Bid_Queue2.4 = (($O_8$, 1.56, 1500))
Offer_Queue2.4 = Offer_Queue2.3
Positions2.4 = (($P(O_3)$, long, 2000, 3280, 1.64, 5%),
($P(O_4)$, long, 2500, 4000, 1.60, 4%),
($P(O_5)$, long, 2000, 3180, 1.59, 4%),
($P(O_6)$, long, 2500, 3950, 1.58, 4%),
($P(O_{12})$,short, 5000, 7970, 1.594, 4%),
($P(O_7)$, long, 1500, 2355, 1.57, 4%),
($P(O_8)$, long, 2500, 3900, 1.56, 4%))

Where:
LastTP = 1.56

Note that order $O_8$ has opened a new position $P(O_8)$ in State2.3. However, $O_8$ was only partially matched. In State2.4, $O_8$ is fully cleared. Thus, we update the already opened position $P(O_8)$. The position $P(O_3)$ will only be closed when the last transaction price falls below $1.64\times(1-5\%) = 1.558$.

To summarize, a single market order of 5000 units led to the closure of two positions, which led to a total clearance of 11000 units, and a drop of 2.5% (from $\geq$1.60 to 1.56) in the market. It should be useful to compute, given a particular state of the market, how big an order is needed to drop the price by, say, 10%.

Besides, what would happen if the ($P(O_3)$, long, 2000, 3280, 1.64, 5%) position has a 4% margin, instead of 5%? This will mean that this position has to be closed, but only 1500 of the 2000 will be bought (by the last bid in the queue); the remaining 500 units will not be cleared. The analysis of these properties goes beyond the scope of this simple calculus

**3. Consequential Closure**

One can compute the consequential closure with respect to margin constraints. By doing so, one can evaluate the final state of any given event. For example, one would be able to say that "a market order to sell 6 million will lead to a price drop of 4%". One may also compute the condition for minimum price changes, e.g. "What is the minimum size of a market sell order to lead to a price drop of r%?" Answering

questions like this would help to assess the stability of the market and value at risk. It could provide early warnings.

An algorithm as outlined below returns the volume of a market sell order that would lead the price to drop to or below price $P_{drop}$. This function traverses the bid queue and examines the effect of hypothetical market sell orders on the underlying market state, with respect to traders' positions and their margin constraints. The function takes three inputs; the Queue_Profile and the Positions_Profile of the underlying market state and the desired $P_{drop}$. In each iteration of the function, a new market sell order is placed to walk through the bid queue. This continues until $P_{drop}$ is reached.

Function MinDrop(Queue_Profile, Position_Profile, $P_{drop}$)

/* Let Queue_Profile = (Bid_Queue, Offer_Queue)
If Bid_Queue is not empty, let it be (($P_1$, $V_1$), ($P_2$, $V_2$), …, ($P_{bq}$, $V_{bq}$)) */

i ← 1; Volume ← 0;
Bid_Queue' ← Bid_Queue;
/* Bid_Queue' is a working structure; if it is not empty, then let its head be ($P_1$', $V_1$') */

While $P_1$' > $P_{drop}$ and Bid_Queue' is not empty
If $V_i \leq V_1$' /* $V_i$ is the volume at index i of Bid_Queue*/
Then {Volume ← Volume + $V_i$; i ← i + 1}
Else Volume ← Volume + $V_1$'; /* See if incrementing Volume by $V_1$' makes any difference */
Queue_Profile' ← closure(Queue_Profile, Position_Profile, (offer, Inf, Volume));
($P_1$', $V_1$') ← Head of the bid queue in Queue_Profile'
End While

If $P_1$' > $P_{drop}$ Then report that $P_{drop}$ cannot be reached in this market as Bid_Queue is exhausted

Return Volume;

The market sell order is fed into to the procedure *closure (Queue_Profile, Position_Profile,Order)*. The only variable input to *closure* is the market order, as it has a different volume in each iteration. The procedure computes the resulting Queue_Profile' after consequential closure is maintained using the calculus shown in

the Model 2 Section[3]. This involves matching the market sell order with the bid queue; updating the market positions profile; updating and sorting the queue profile; checking for margin calls and its consequential forced positions closures while keeping record of the last transaction price. The Queue_Profile' is a working structure, which is discarded on exit. It is used to define the potential price $P_1'$ (head of the bid queue in Queue_Profile'), the market would reach after executing the market sell order. If $P_1' > P_{drop}$, the colure procedure is called again to evaluate the impact of a bigger market sell order. The algorithm increments i (which has the effect of increasing volume) until enough volume is accumulated to see the price drop to $P_{drop}$. The function will terminate when $P_{drop}$ is reached or when the bid queue is completely cleared. Once terminated, the function returns the Volume required to reach $P_{drop}$., giving a preview of the potential multiplied effect on the underlying market once a market sell order of a specific volume is placed.

If the market does not have enough depth, all the buy orders will be exhausted before $P_{drop}$ is reached. Otherwise, there exists a minimum k such that, for all the orders $(P_i, V_i)$ at the front of the Bid_Queue, $P_{drop} \leq P_i$ and $Volume \leq V_1+V_2+...+V_k$. In the worst case, Function MinDrop has to go through all such $(P_i, V_i)$s[4]. Volume increases monotonically in Function MinDrop. Therefore this function must terminate.

Let M be the list of positions in the Position_Profile which margin calls are above $P_{drop}$. In the worst case, the procedure has to go through all of them. So each cycle of the Repeat loop will have complexity of |M|. Each "Then" part in each cycle

---

[3]Strictly speaking, the termination condition $P_1' \leq P_{drop}$ should be replaced by $LTP \leq P_{drop}$, where LTP is the Last transaction price which could be returned by the closure function. This is simplified for clarity. When the head of the queue in Queue_Profile' is below $P_{drop}$, any market order to sell will drop the price below $P_{drop}$. Therefore, the Volume returned is correct, which is our justification for the compromise.

[4] This is an upper-bound because any margin calls that might be triggered will absorb some of the volume.

of the Repeat loop would increase Volume to include one ($P_i$, $V_i$) pair. It is more complex to analyse the number of times that the "Else" part could be entered. In the worst case, each of the positions could bring the loop into the Else part through a margin call. Therefore, the complexity of the algorithm is bounded by $O(k \times |M|^2)$.

**4. Market Making**

The market maker is an aggregator who nets the flow of buyers and sellers. His profit is a reward for managing the uncertainty of this process. He manages the flow by dynamically skewing bid and ask prices. The market maker sets the "bid" and "ask" price on a tick by tick basis. The bid price is the price at which the market maker offers to buy; the ask price is the price at which the market maker offers to sell.

State = (Bid_price, Ask_price, MaxVol, Queue_Profile, Position_Profile)

Where:

Bid_price and Ask_price are the bid and ask prices quoted by the market maker;

MaxVol is the maximum volume that the market maker is willing to deal per order;

Queue_Profile and Position_Profile are the same as those defined in Model 2.

Here we assume that the clearing mechanism is completely automated. The key to the clearing mechanism is in the way that the market maker updates its bid and ask prices. In this paper, we make no assumption on f, which could vary from market maker to market maker; f should be a complex function.

Let Bid_price and Ask_price be the bid and ask prices in the current state, and Bid_price' and Ask_price' be the bid and ask prices in the next state. We generalize that the market maker sets the Bid_price' and Ask_price' with a function f, without specifying exactly what f is. f is a function that involves Bid_price, Ask_price, Queue_Profile, Position_Profile and many other factors, which may include the market maker's own

position, bid and ask prices by the other market makers, the balance of payment between countries, interest rates, news and other economic indicators of the countries involved.

$$(\text{Bid\_price}, \text{Ask\_price}, \text{MaxVol}, (\text{Bid\_Queue}, \text{Offer\_Queue}), \text{Positions}) + (\text{sell}, P, V) \rightarrow (\text{Bid\_price}', \text{Ask\_price}', \text{MaxVol}, (\text{Bid\_Queue}, \text{Offer\_Queue}), \text{Positions}) \quad \text{if } P \leq \text{Bid\_price} \ \& \ V \leq \text{MaxVol}$$

$$(\text{Bid\_price}', \text{Ask\_price}', \text{MaxVol}, (\text{Bid\_Queue}, \text{Offer\_Queue}), \text{Positions}) + (\text{sell}, P, V - \text{MaxVol}) \quad \text{if } P \leq \text{Bid\_price} \ \& \ V > \text{MaxVol}$$

$$(\text{Bid\_price}', \text{Ask\_price}', \text{MaxVol}, (\text{Bid\_Queue}, \text{Offer\_Queue} \oplus (\text{sell}, P, V)), \text{Positions}) \quad \text{if } P > \text{Bid\_price}$$

The queue joining operator $\oplus$ is defined in the Model 1 Section. For any well specified f, we should be able to formalize market making.

**5. Liquidity**

Artzner, et al. (1999) proposed coherent measures of risk. This was scrutinized by Acerbi and Scandolo (2008), for not taking full consideration of liquidity risk. Acerbi and Scandolo (2008) introduced the marginal supply-demand curves (MSDCs), which defines at any time instance the available prices of a given asset in the market. The attractiveness of their formalism is that liquidity risk is measured by market data; no assumptions are required. Fig. 1 shows the MSDC in State 2.1. After clearing of Order 2.1, the market loses a certain amount of liquidity. This is shown by MSDC in Fig. 2. Like Acerbi and Scandolo (2008), we are looking at the microstructure of illiquid markets, and free from hypotheses on the dynamics of the market.

The work by Acerbi and Scandolo (2008) is based on the concept of mark-to-market. When position and margin information are not considered (Model 1 above), the shape of the MSDC curve depends on queue profiles alone. When position and margin information are available, the mark-to-market values are changed. In fact, the

shape of the MSDC could be changed by the orders processing procedure above. Therefore, this paper complements Acerbi and Scandolo's work.

The queue profile defines how liquid an asset is at any given time. Liquidity of an asset is therefore determined by how steep one ascends or descends in the MSDC. Following the above example, suppose at State 2.1, two traders bid 1.60 for 500 shares, and 1.59 for another 500 shares. Although the highest bid price is still 1.60, the new MSDC is actually steeper than the one shown in Fig. 1. This means, to sell over 1000 shares in this market (as opposed to the market shown in Fig. 1) , the seller must be prepared to accept lower bids.

Unfortunately, the ordinary investors/traders who have no access to order books have no means of fully assessing their liquidity risk.[5] Therefore, market making provides investors/traders with market liquidity up to a certain limit (MaxVol in the Market Making Section). It also offers transparency in market liquidity. The MSDC under market making is shown in Fig. 3.

It is worth noting the obvious that, as a Queue Profile does not have to be symmetric, an asset could be highly liquid when one wants to buy, but illiquid when one wants to sell (and vice versa).

## 6. Conclusion

In this paper, we have defined a calculus for describing state changes in a market as a consequence of new orders being posted. We base our analysis on simple market models. This is no attempt to predict what new exogenous orders will arrive. The purpose of this paper is to lay the foundation for analyzing market states. We show that even with the simple calculus defined, we can ask important questions such as

[5] OANDA provides information on trader positions 0. This could help conjecturing (with low confidence) marginal supply and demands (because eventually those in long positions have to sell, and those in short positions have to buy).

"*how big a sell order would push the price down by 10%?*" This research also supports Acerbi and Scandolo's call to measure liquidity risks with market data.

We acknowledge the fact that state changes in real markets are far more complex than what is described in this paper. It is up to the participants, including governing bodies, market makers and traders, to define the rules in an unambiguous mathematical and mechanical way. The aim is to create markets with properties that can be studied formally as well as extensively in simulations and validated empirically. Eliminating black boxes and laying the foundations for extensive scientific analysis may be the best way to ensure stability and prevent financial crises. We intend to extend the calculus to cover more sophisticated market mechanisms and trading strategies.

**Acknowledgements**

The authors wish to thank Monira Aloud for providing us with feedback and formatting the paper for us and the anonymous referees for providing thorough and insightful feedback.

**References**

Acerbi, C. and Scandolo, G., 2008. Liquidity Risk Theory and Coherent Measures of Risk, *Quantitative Finance*, 8(7), pp.681-692.

Acerbi, C., 2008. Portfolio Theory in Illiquid Markets. In: Resti, A. (ed.), Pillar II in the new Basel Accord, Riskbooks, pp. 241-272

Artzner,P., Delbaen,F. Eber, J-M. and Heath, D., 1999. Coherent measures of risk, *Math. Fin.*, 9(3), pp. 203-228.

Chen, C-C., 2009. *Complex Event Types for Agent-Based Simulation*. PhD, College of London University.

Dacorogna, M.M., Gencay, Muller, R.U., Olsen, R.B. and Picktet, O.V., 2001. *An Introduction to High Frequency Finance*. Academic Press.

Glattfelder, J.B., Dupuis, A. and Olsen R., 2008. An Extensive Set of Scaling Laws and the FX Coastline", *Working Paper WP025-08*, Centre for Computational Finance and Economic Agents (CCFEA), University of Essex.

Kowalski, R. and Sergot, M., 1986. A Logic-Based Calculus of Events, *New Generation Computing*, 4(1), pp. 67-95.

Mueller, E.T., 2009. Automated Commonsense Reasoning Using The Event Calculus",
*Communications of the ACM*, 52(1), pp. 113-117.

OANDA FxTrade. OANDA Forex Order Book [online] Available at: http://fxtrade.oanda.com/tools/statistical_information/fx_open_position_summary [Accessed 9 March 2011].

Olsen, R., 2005 Classical Economics: An Emperor with No Clothes, *Wilmott Magazine,* 15, pp. 84-85.

Shanahan, M.P., 1999. The Event Calculus Explained, in Artificial Intelligence Today, ed. M.J.Wooldridge and M.Veloso, *Springer Lecture Notes in Artificial Intelligence*, 1600, pp. 409-430.

Shleifer, A.,2000. *Inefficient Markets: An Introduction to Behavioral Finance*. Oxford University Press.

Solomon, S., Levy, H. and Levy, M., 2000. *Microscopic Simulation of Financial Markets from Investor Behaviour to Market Phenomena*. Academic Press

Tsang, E.P.K., 2008. Computational Intelligence Determines Effective Rationality", *International Journal on Automation and Control*, 5(1), pp. 63-66.

**Figure 1.** The Marginal Supply-Demand Curve defined by the Queue Profile at State 2.1

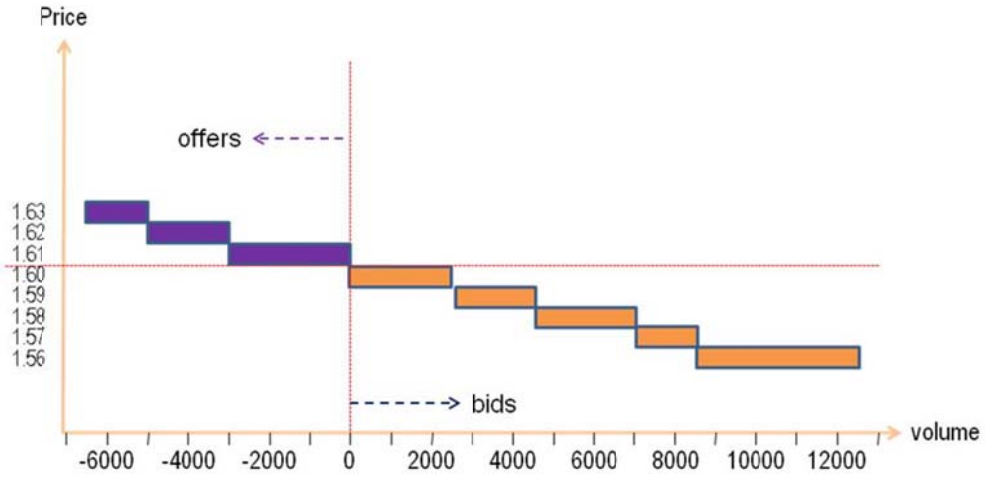

**Figure 2.** The Marginal Supply-Demand Curve defined by the Queue Profile at State 2.2

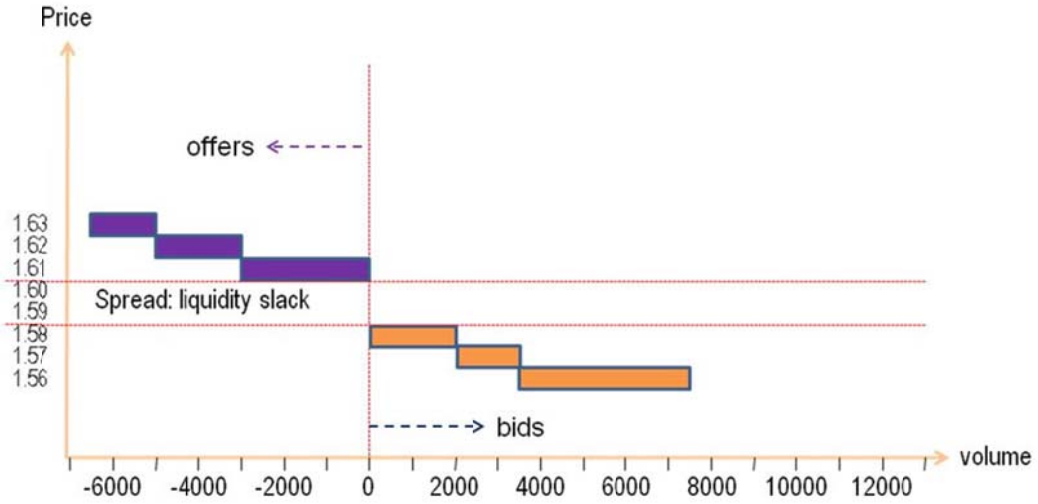


**Figure 3.** The Marginal Supply-Demand Curve under market making